Durham E-Theses

Temporal dynamics and projected future changes in nitrate leaching in a small river catchment dominated by under-drained clay soil grasslands: analysis of high-frequency monitoring data

SUSLOVAITE, VAIDA

Temporal dynamics and projected future changes in nitrate leaching in a small river catchment dominated by under-drained clay soil grasslands: analysis of high-frequency monitoring data

Vaida Suslovaite

Abstract

Water pollution has long been an issue due to its adverse effect on human health and the environment. This research has concentrated on a small river catchment dominated by grassland underlain by drained clay soil. It analysed temporal dynamics and projected future changes in nitrate leaching from the catchment. The types of nitrate response were classified according to the type of hysteresis loop obtained when concentration was plotted against discharge for each storm event. To narrow down the variable list and focus on the most relevant variables, stepwise multiple regression analysis was used. This work informed the understanding of how nitrate leaching patterns depend on source and transport, and the effect of antecedent conditions on hydrology and leaching patterns. The Newby Beck catchment is dominated by clockwise hysteresis events. Clockwise hysteresis storm events are source (or supply) limited. Nitrate concentrations within the catchment are governed by preparatory processes of nitrate production at source in the soil. Projected climate change may act to alter these preparatory processes through increasing temperatures and more severe droughts in summer, which would result in even more extreme nitrate leaching events. The transport from the catchment is driven by shallow subsurface flow through the drained clay soil and increasing precipitation with future climate change in winter will act to further facilitate the rapid transportation of nitrate to stream water. It was estimated that there is a 71% probability of future nitrate flux increasing by the 2050s and an 82% probability of future nitrate flux increasing by the 2080s. Mitigation measures that intercept water that is transporting this nitrate, combined with reduction of nitrate supplied by anthropogenic activities, will be the best choices.

Temporal dynamics and projected future changes in nitrate leaching in a small river catchment dominated by under-drained clay soil grasslands: analysis of high-frequency monitoring data

Vaida Suslovaite

Masters by Research

Department of Geography

at

Durham University

2016

Table of Contents

List of Illustrations

Figure 1.12 Water flows during infiltration into a block of soil with macropores. $P(t)$, overall input (precipitation, irrigation); $I_1(t)$, infiltration into the matrix from the surface; $I_2(t)$, infiltration into the matrix from the walls of the macropores; $S_1(t)$, seepage into the

Declaration

I confirm that no part of the material presented in this thesis has been
previously submitted by me or any other person for a degree in this or any other
university. In all cases, where it is relevant, material from the work of others has
been acknowledged.

Chapter 1: Introduction and Literature Review

1.1 Introduction

Water pollution has long been an issue due to its adverse effect on human health and the environment. Water pollution issues can be recognized at a number of different scales that range from local to global, with regional and continental scales being in between these two extremes (Burt et al. 1993). Of most relevance are the local and regional scales as this is where management of rivers, lakes and aquifers is mostly undertaken and it is also the scale at which pollution sources are easier to pin-point and mitigate. This chapter outlines main issues caused by water: main issues and policies on water pollution (section 1.2); projected future climate change in UK (section 1.3); nitrate cycle (section 1.4); and, aims and objectives of this research (section 1.5).

1.2 Main Issues and Policies on Water Pollution

Excessive nutrient levels can have impacts on both human health and aquatic ecology. Drinking water contaminated with high concentrations of nitrate can be detrimental to human health (WHO, 2011). The main concerns are *methaemoglobinaemia* and stomach cancer. although these health concerns have divided opinions on whether they are indeed caused solely by consumption of high levels of nitrate polluted drinking water (Burt et al. 1993; WHO, 2011). The Council Directive on the quality of water intended for human consumption 80/778/EEC (updated version 98/83/EC) set the maximum nitrate concentration in drinking water at 50 mg NO_3 l^{-1} (equivalent to 11.3 mg NO_3-N l^{-1}).

In some freshwater and many marine environments, phosphorus and nitrate are the limiting nutrients for algal growth and hence the addition of large amounts of these nutrients can lead to eutrophication (Smith et al., 1999; O'Sullivan, 1995). Nutrients that are leached from the catchment and delivered to rivers, lakes and coastal waters, can be easily utilised by plants. Although slightly higher nutrient levels will initially promote growth, excessive amounts of it will favour fast-growing aquatic plants which can take advantage of this surplus of nutrients faster than other species. Then, one or two species of plant will dominate the water body, lowering the species diversity and food availability for animals. This effect can lead to toxic algal blooms that contaminate the water with toxins. Also, when the large amounts of organic matter produced in algal blooms die, its decomposition uses up even more oxygen in the water column, killing animals that require high oxygen levels to respire, further lowering biodiversity and contaminating water (Brady and Weil, 2008, p.677). Water contaminated with toxins and decomposing organic matter can be difficult and expensive to clean up (Pretty et al. 2003). Therefore, it makes sense for water

pollution to be mitigated before it reaches water bodies that are utilised for drinking water supply to avoid eutrophication and its by-products.

Eutrophication can lead to damage to natural ecosystems and the services they provide (Smith et al., 1999). Lowering of species diversity within a water body can have adverse effect on any terrestrial species that depend on it for food or spend part of their lives in water. Food supply for human consumption can also be affected through damage to estuarine, sea and ocean ecosystems (Vitousek et al. 1997).

To help achieve good status for all waters, the EU Water Framework Directive (2000/60/EC) set out a number of objectives for every water body within the EU. The main objectives are general protection and improvement of the aquatic ecology and specific protection of unique and valuable habitats. The management of the Water Framework Directive is implemented within individual river basins - the natural geographical and hydrological unit. Surface waters should achieve good ecological and chemical status while groundwater should achieve good chemical and quantitative status by 2015 and then be re-assessed every six years. In addition to the WDF, the Drinking Water Directive (98/83/EC) and Bathing Waters Directive (2006/7/EC) concern protection of drinking water resources, and protection of bathing water.

The WFD is complemented by the Habitats Directive (92/43/EEC) on the conservation of natural habitats and of wild fauna and flora aims to promote the maintenance of biodiversity, taking account of economic, social, cultural and regional requirements. It ensures the conservation of over a 1000 of rare, threatened or endemic animal and plant species as well as 200 rare and characteristic habitat types. Reduction of water pollution is relevant to both of these directives in order to improve water quality and protect the listed animals and habitats. The objective of good status for all waters in the EU was not reached by the first deadline of 2015 (European Commission, 2015). Improvement of knowledge of pollutant behaviour and underlying catchment hydrology is still needed as well as improved methods of water pollution mitigation. Future projected climate change complicates the planning and implementation of mitigation measures as it may impact the amount of nutrients leaving the river catchment in the future. Good mitigation measures will take into account effect of climate change as well as current nutrient leaching patterns.

1.3 Projected Climate Change in UK

Future climate change may affect UK climate to not only impact the temperatures but also the precipitation received (Met Office, 2011). Information from the United Kingdom Climate Projections 2009 (UKCP009) is the current standard for assessing possible future impacts from climate change. The projections are presented for future climates represented by 30-year periods centred on each decade of the 21st century. The projections presented here are for the 2050s (2040 – 2069) and the 2080s (2070-2099). The UK Met Office projects the temperature warming to be greater for summer (Figure 1.1) than winter (Figure 1.2). Mean summer temperature at the central estimate (50% probability) is projected to increase 2-3 °C by the 2050s and 3-4 °C by the 2080s with some southern parts of the country experiencing temperature increases of up to 4-5 °C, as can be seen in Figure 1.1 (b). The temperature increase (at 33% probability level) is unlikely to be less than 1-2 °C and 2-3 °C by the 2050s, and 2-3 °C and 3-4 °C by the 2080s. The magnitude of the change is dependent on location (Figure 1.1 (a)). The temperature increase (at 67% probability level) is unlikely to greater than 2-3°C and 3-4°C by the 2050s, and 3-4°C and 4-5 °C by the 2080s. The magnitude of the change is again dependent on location (Figure 1.1 (c)).
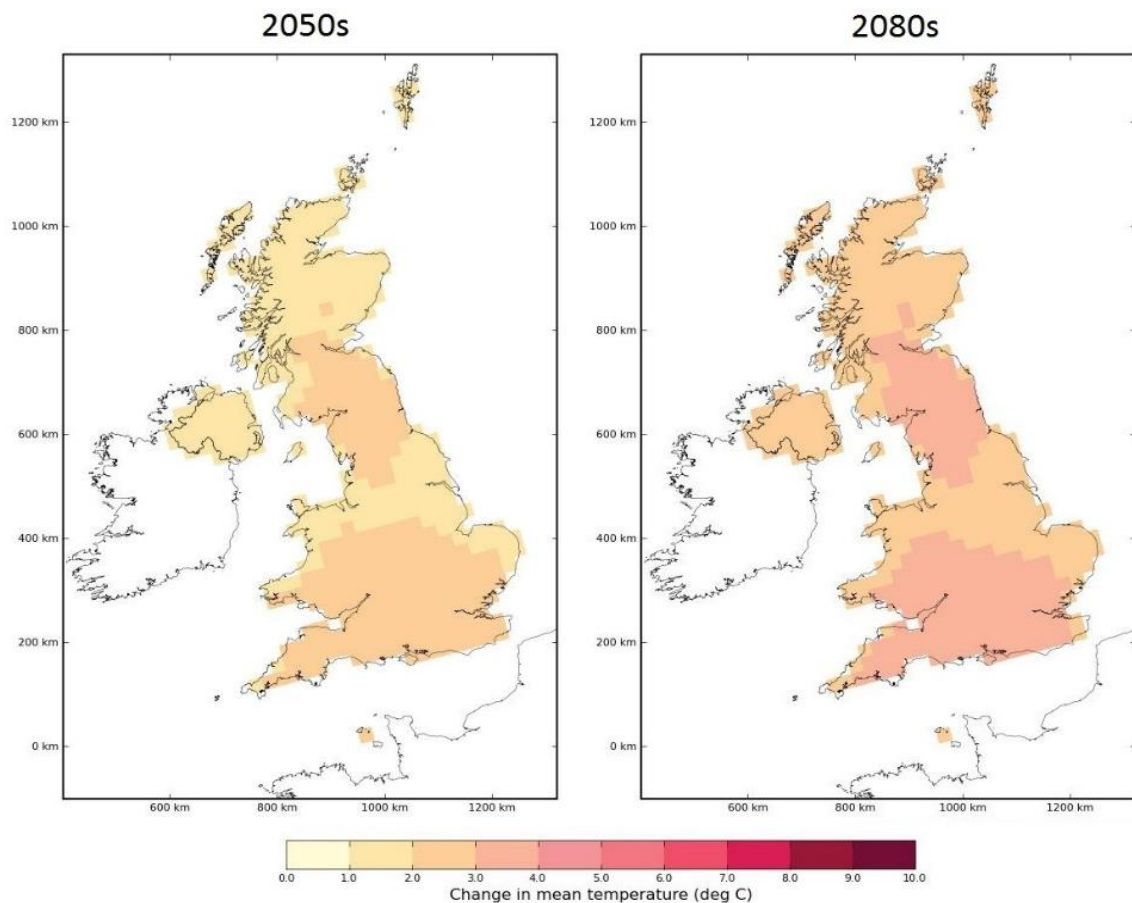
Figure 1.1 (a) Projected changes in mean summer temperatures (°C) at 33% probability level: unlikely to be less than the values shown, based on the UKCP09 projections. Maps are for medium emissions scenario. © UK Climate Projections 2009
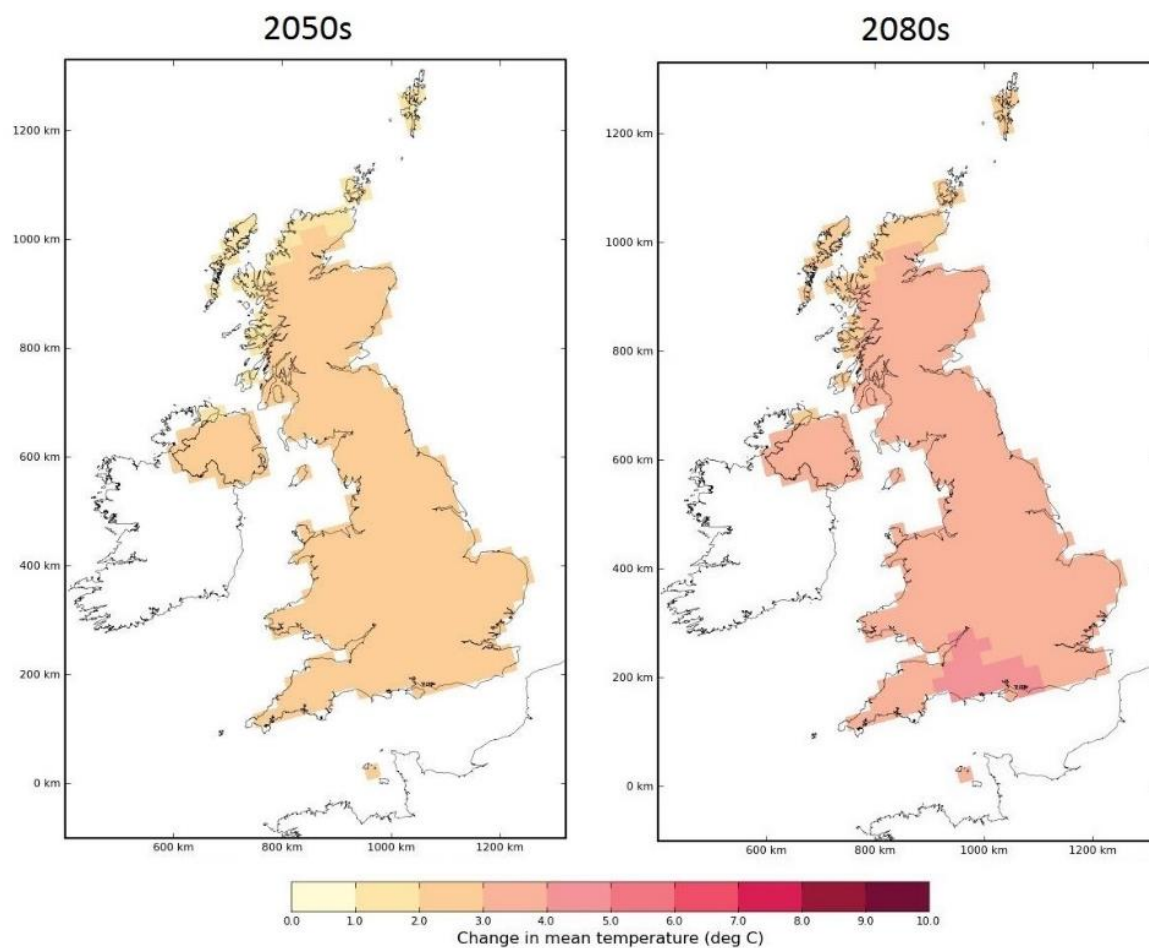
Figure 2.1 (b) Projected changes in mean summer temperatures (°C) at 50% probability level: central estimate, based on the UKCP09 projections. Maps are for medium emissions scenario. © UK Climate Projections 2009
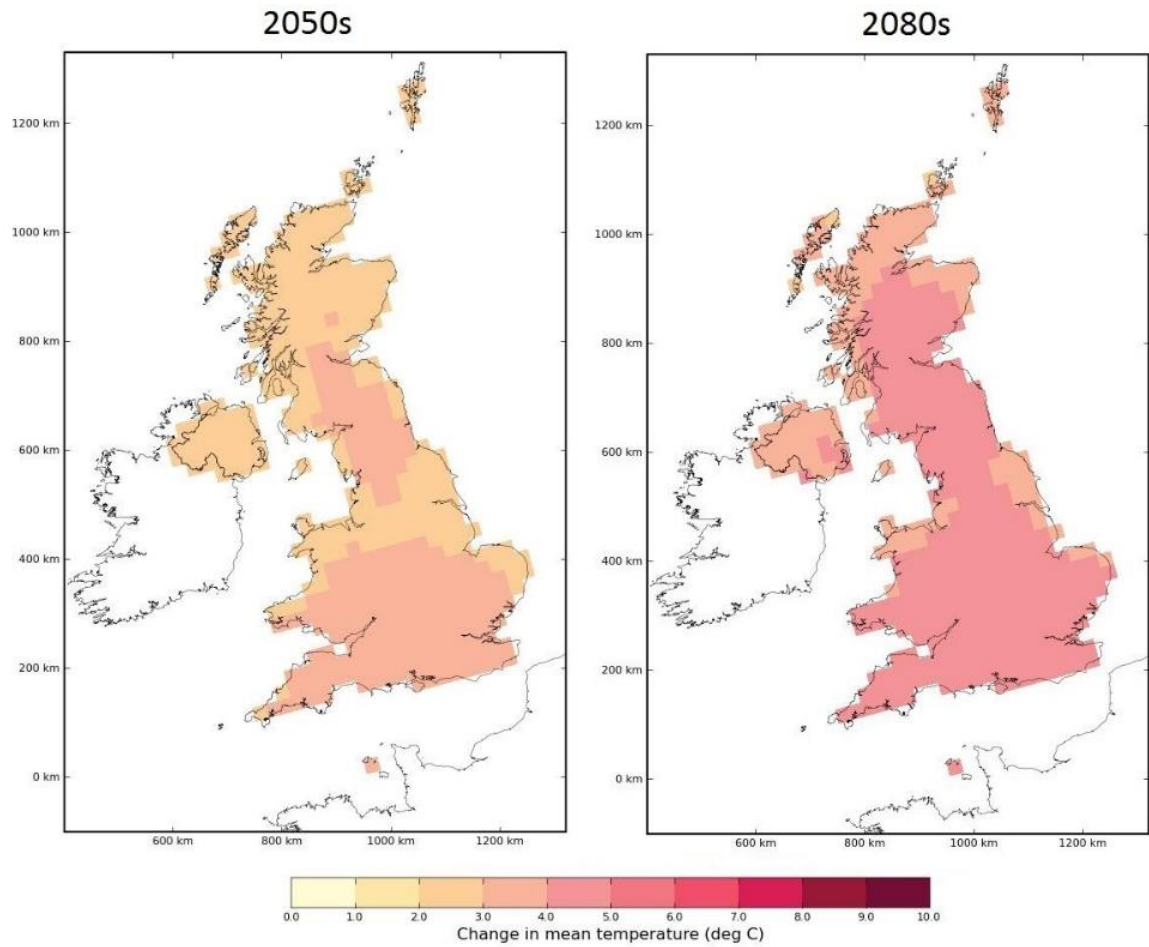
Figure 3.1 (c) Projected changes in mean summer temperatures (°C) at 67% probability level: unlikely to be greater than the values shown, based on the UKCP09 projections. Maps are for medium emissions scenario. © UK Climate Projections 2009

The projected increases in winter mean temperatures for the central estimate (50% probability level) are expected to be smaller at 1-2 °C in the north of the country and 2-3 °C in the south by the 2050s. By the 2080s UK is expected to experience increases in mean winter temperatures of 2-3 °C, with 3-4 °C in the south-east of the country (Figure 1.2 (b)). The increase in the mean temperatures will lead to higher soil temperatures. The temperature increase (at 33% probability level) is unlikely to be less than 1-2 °C by the 2050s, and 1-2 °C in the far north of the country and 2-3 °C in the rest of the country by the 2080s (Figure 1.2 (a)). The temperature increase (at 67% probability level) is unlikely to greater than 2-3°C by the 2050s, and 2-3°C in the north of the country and 3-4 °C in the rest of the country by the 2080s (Figure 1.2 (c)).
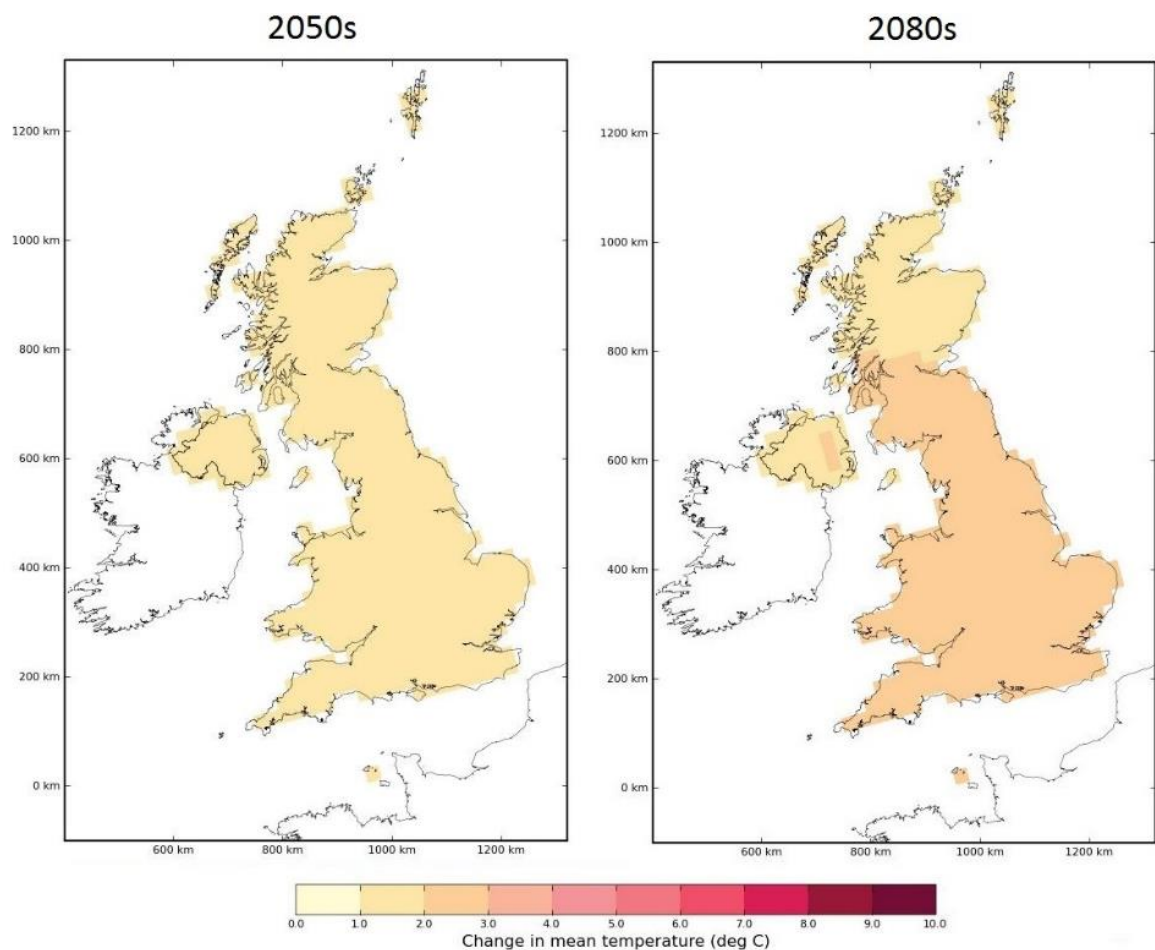
Figure 1.4 (a) Projected changes in mean winter temperatures (°C) at 33% probability level: unlikely to be less than the values shown, based on the UKCP09 projections. Maps are for the medium emissions scenario. © UK Climate Projections 2009
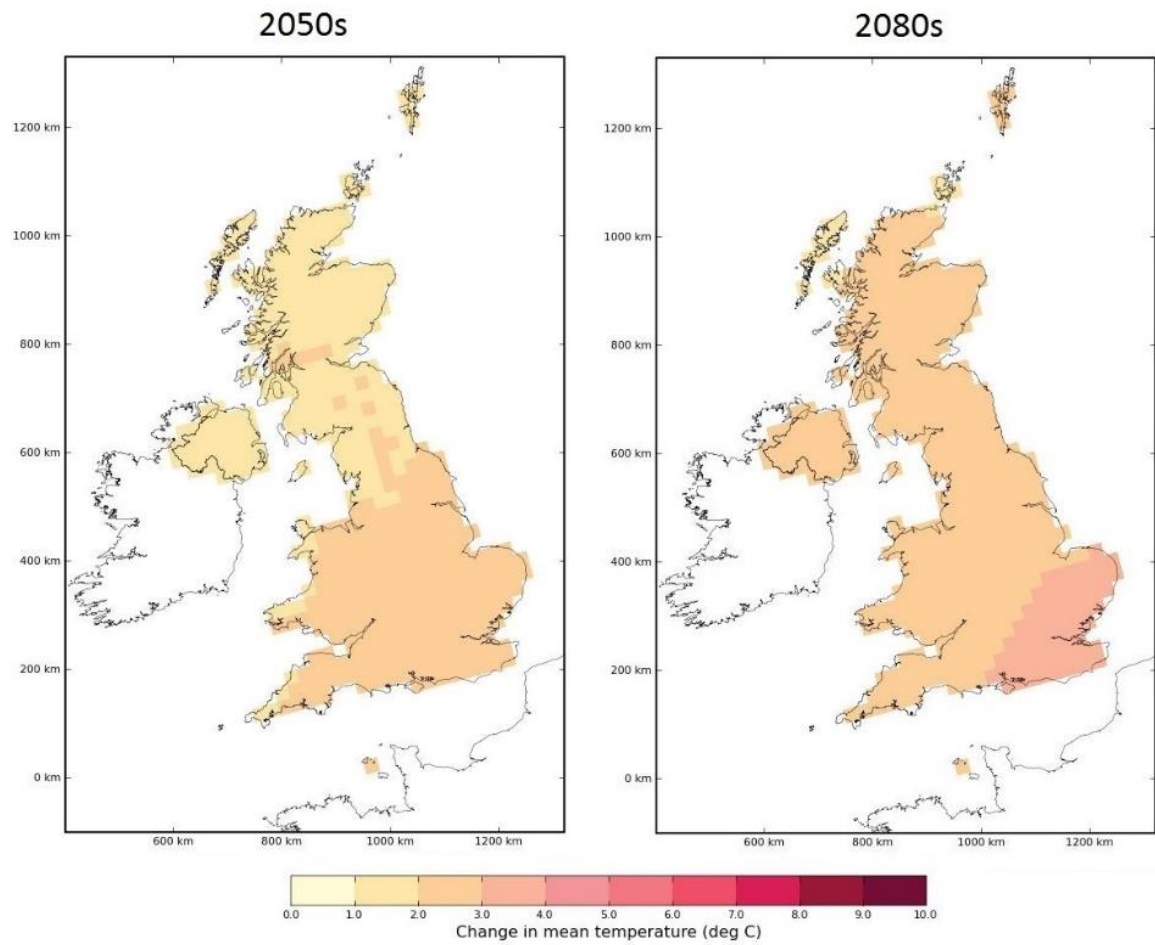
Figure 1.5 (b) Projected changes in mean winter temperatures (°C) at 50% probability level: central estimate, based on the UKCP09 projections. Maps are for the medium emissions scenario. © UK Climate Projections 2009
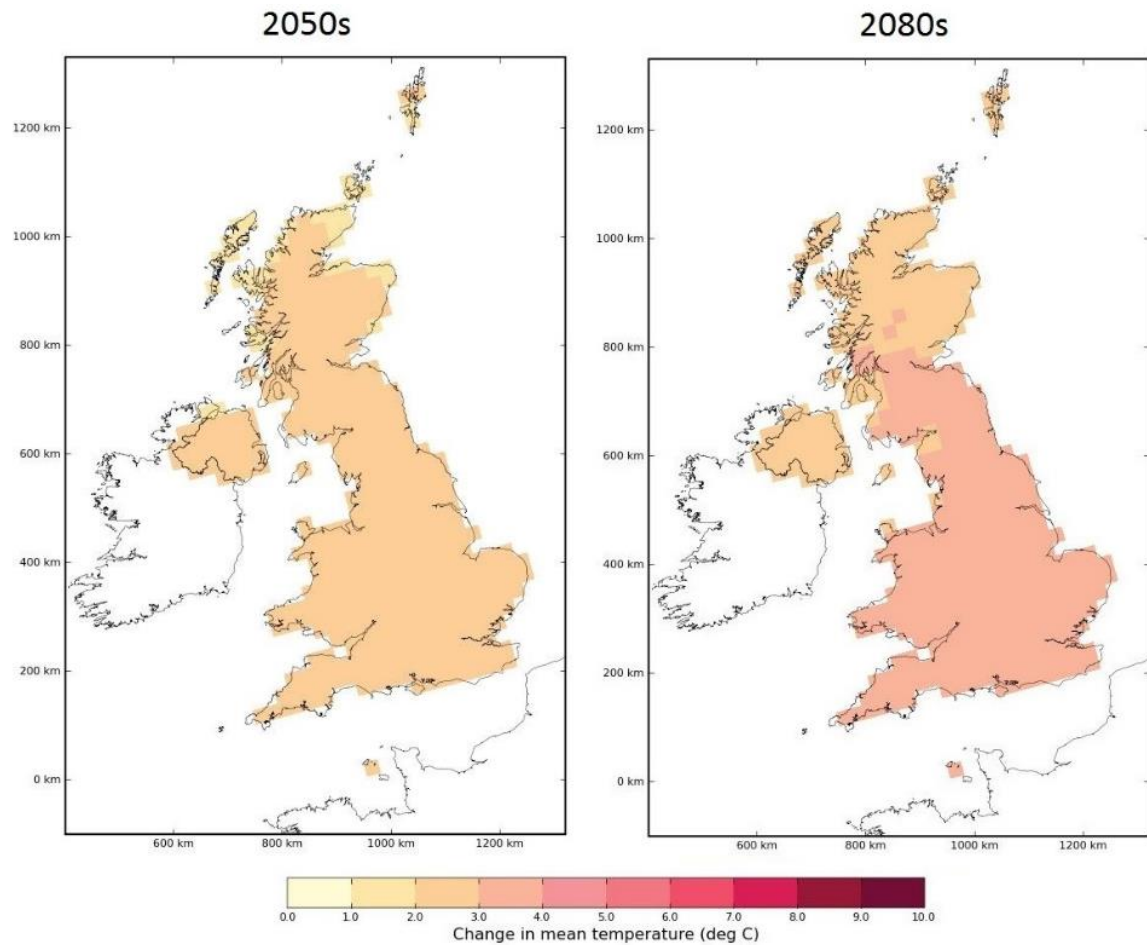
Figure 1.6 (c) Projected changes in mean winter temperatures (°C) at 67% probability level: unlikely to be greater than the values shown, based on the UKCP09 projections. Maps are for the medium emissions scenario. © UK Climate Projections 2009

There are stark differences in the type of precipitation response to climate change between seasons. Mean summer precipitation (Figure 1.3) is expected to decrease while mean winter precipitation is expected to increase (Figure 1.4). Mean summer precipitation for central estimate (50% probability level) is expected to decrease by up to 20% by the 2050s, with more extreme changes in the south-western parts of UK. By the 2080s changes should be even more severe with large parts of UK projected to experience a decrease in precipitation of up to 30%. Some south-western parts could again see a more extreme decrease in mean summer precipitation. The precipitation decrease (at 33% probability level) is unlikely to be less than -10 to -20% up north, -20 to -30% to the south and up to -50% in south west of the country by the 2050s. By the 2080s the patterns should be similar but will have shifted north (Figure 1.3 (a)). The precipitation decrease (at 67% probability level) is unlikely to be greater than 0 to -10% up north, -10 to -20% to the south and up to -30% in south west of the country by the 2050s. By the 2080s the patterns should be similar but will have shifted north (Figure 1.3 (c)).
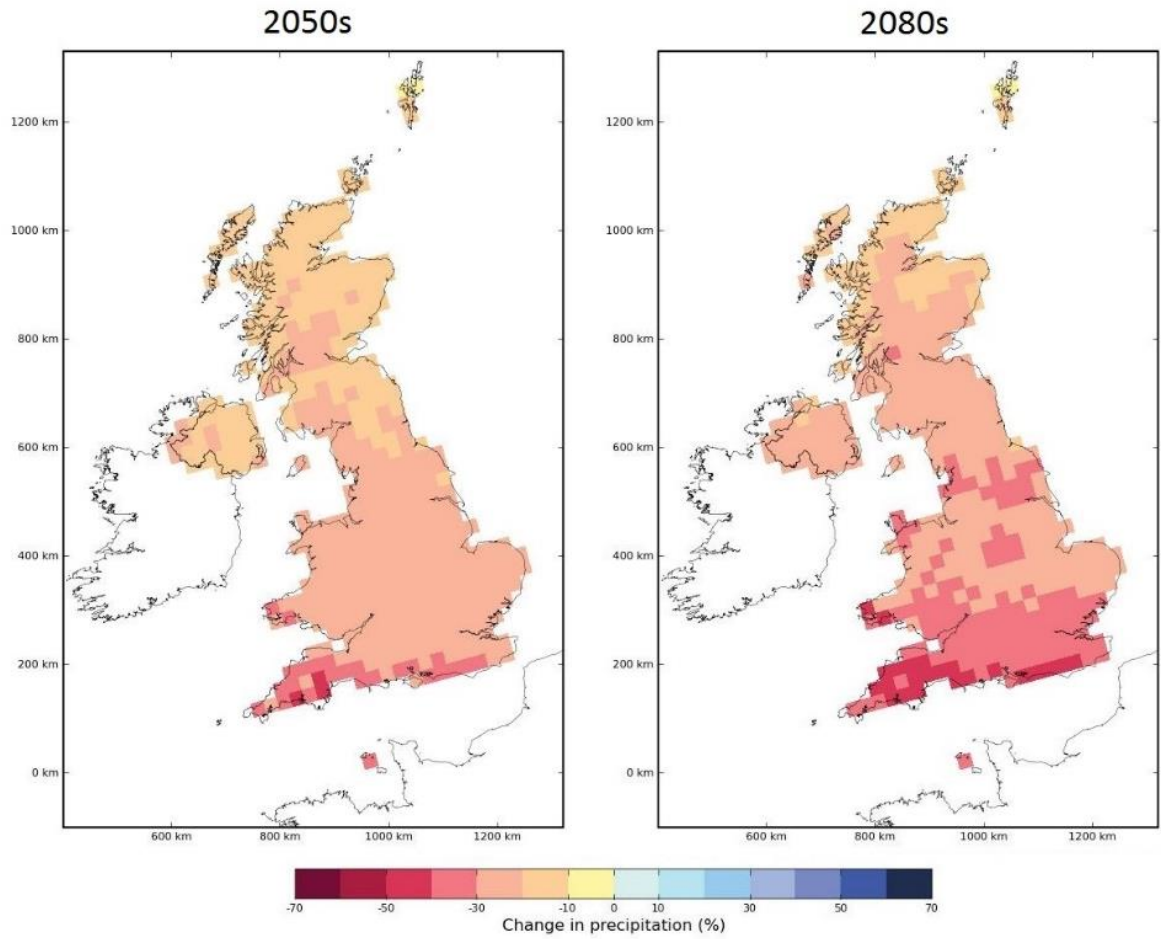
Figure 1.7 (a) Projected mean summer precipitation changes 33% probability level: unlikely to be less than the values shown, based on the UKCP09 projections. Maps are for medium emission. © UK Climate Projections 2009
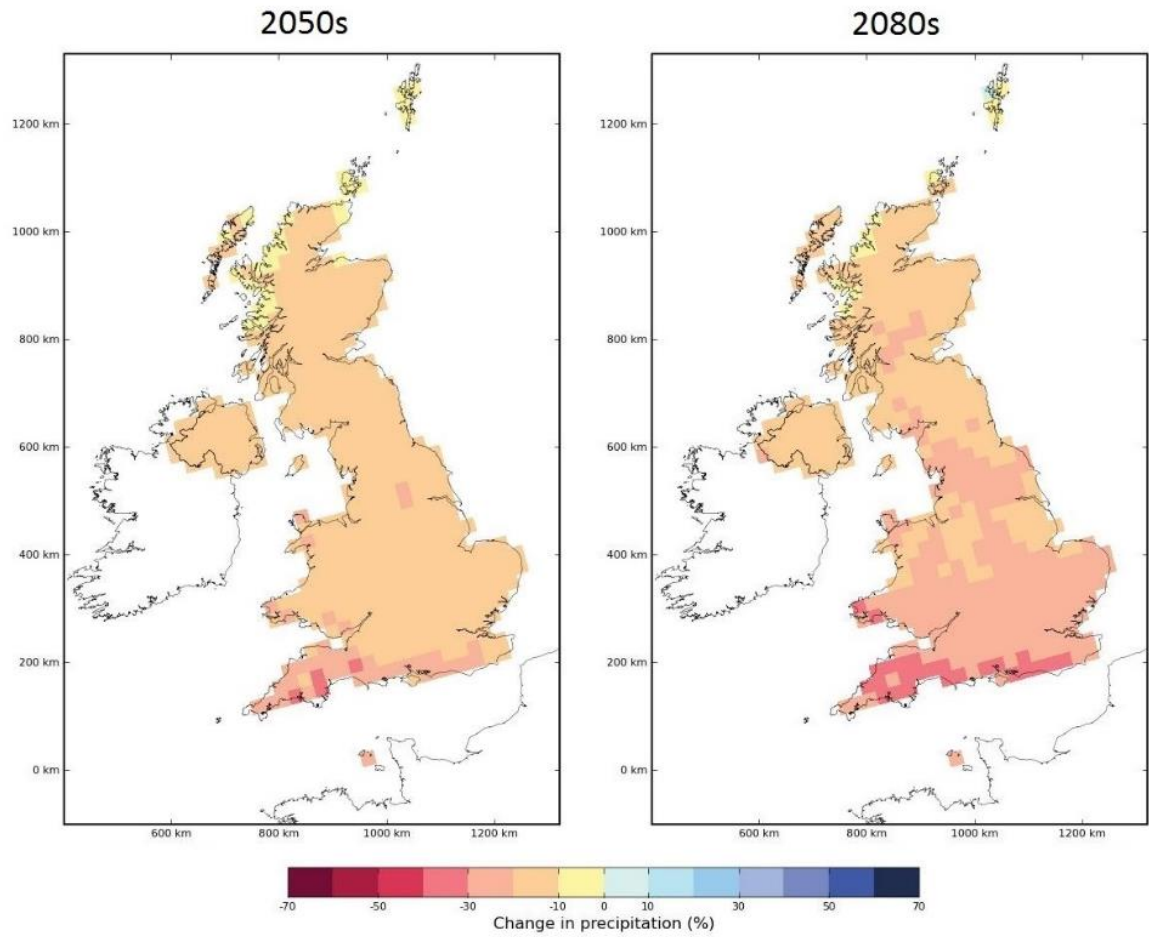
Figure 1.8 (b) Projected mean summer precipitation changes at 50% probability level: central estimate, based on the UKCP09 projections. Maps are for medium emission. © UK Climate Projections 2009
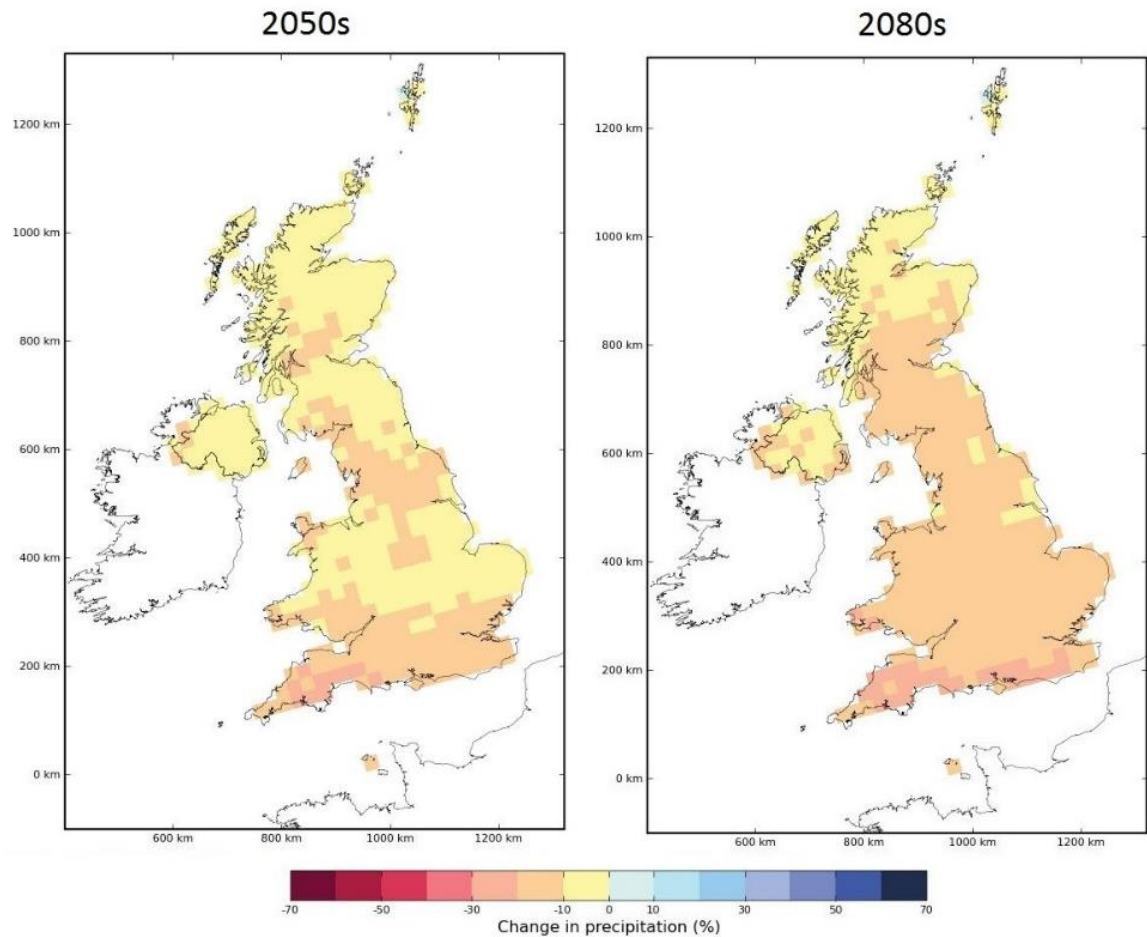
Figure 1.9 (c) Projected mean summer precipitation changes at 67% probability level: unlikely to be greater than the values shown, based on the UKCP09 projections. Maps are for medium emission. © UK Climate Projections 2009

Mean winter precipitation increases for the central estimate (50% probability level) are expected to be up to 20% by the 2050s and up to 30% by the 2080s (Figure 1.4 (b)). The precipitation increase (at 33% probability level) is unlikely to be less than 0 to -10% up north, 0 to 10% for the rest of the country and up to 20% in coastal and south of the country by the 2050s. By the 2080s precipitation change is projected to be 0 to -10% up north, 0 to 20% for the rest of the country and up to 30% in coastal and south of the country (Figure 1.4 (a)). The precipitation increase (at 67% probability level) is unlikely to greater than 0 to 10% up north, 10 to 20% to the south and up to 30% in south and coastal areas of the country by the 2050s. By the 2080s precipitation change is projected to be 0 to 10% in the north, 10 to 20% for the rest of the country and up to 50% in coastal and south of the country (Figure 1.4 (c))
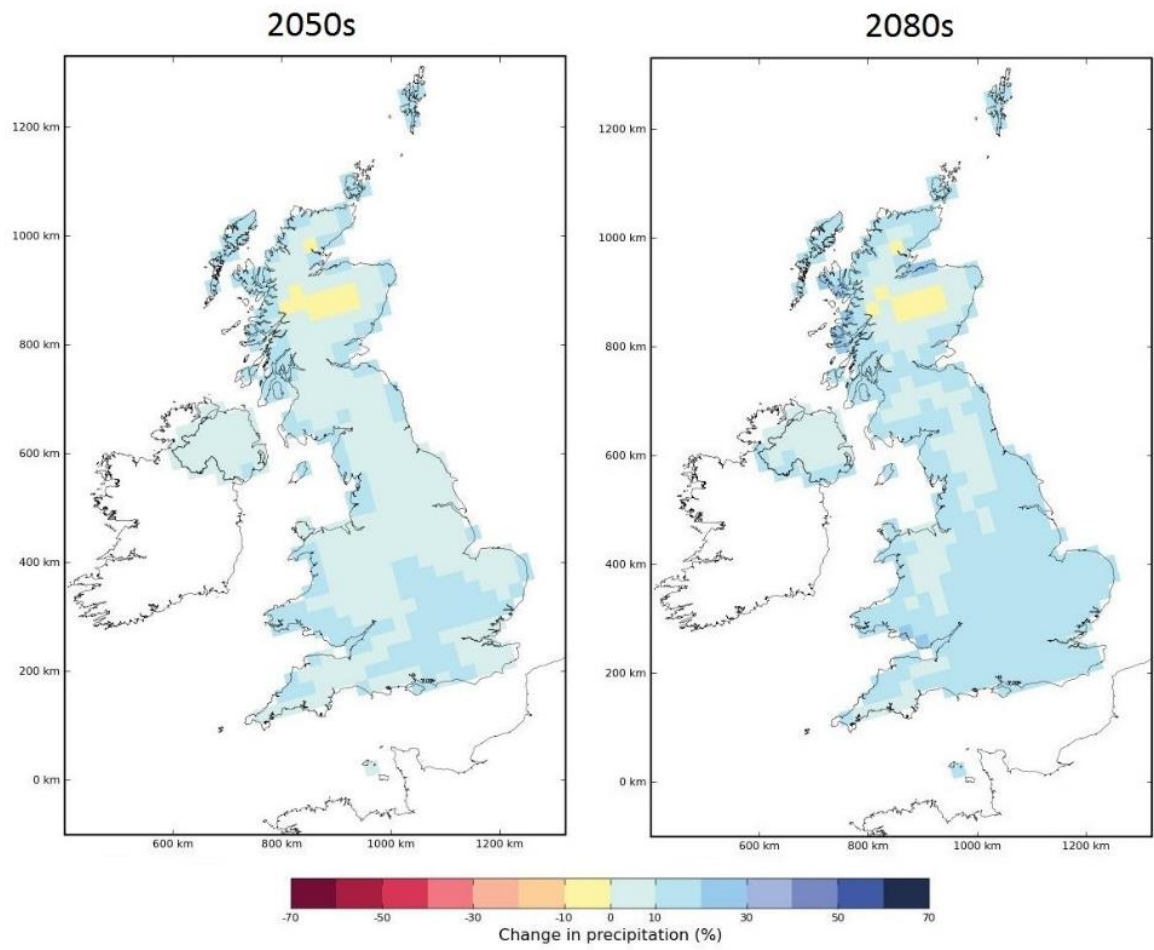
Figure 1.4 (a) Projected mean winter precipitation changes (%) at 33% probability level: unlikely to be less than the values shown. Maps are for medium emission scenario. © UK Climate Projections 2009
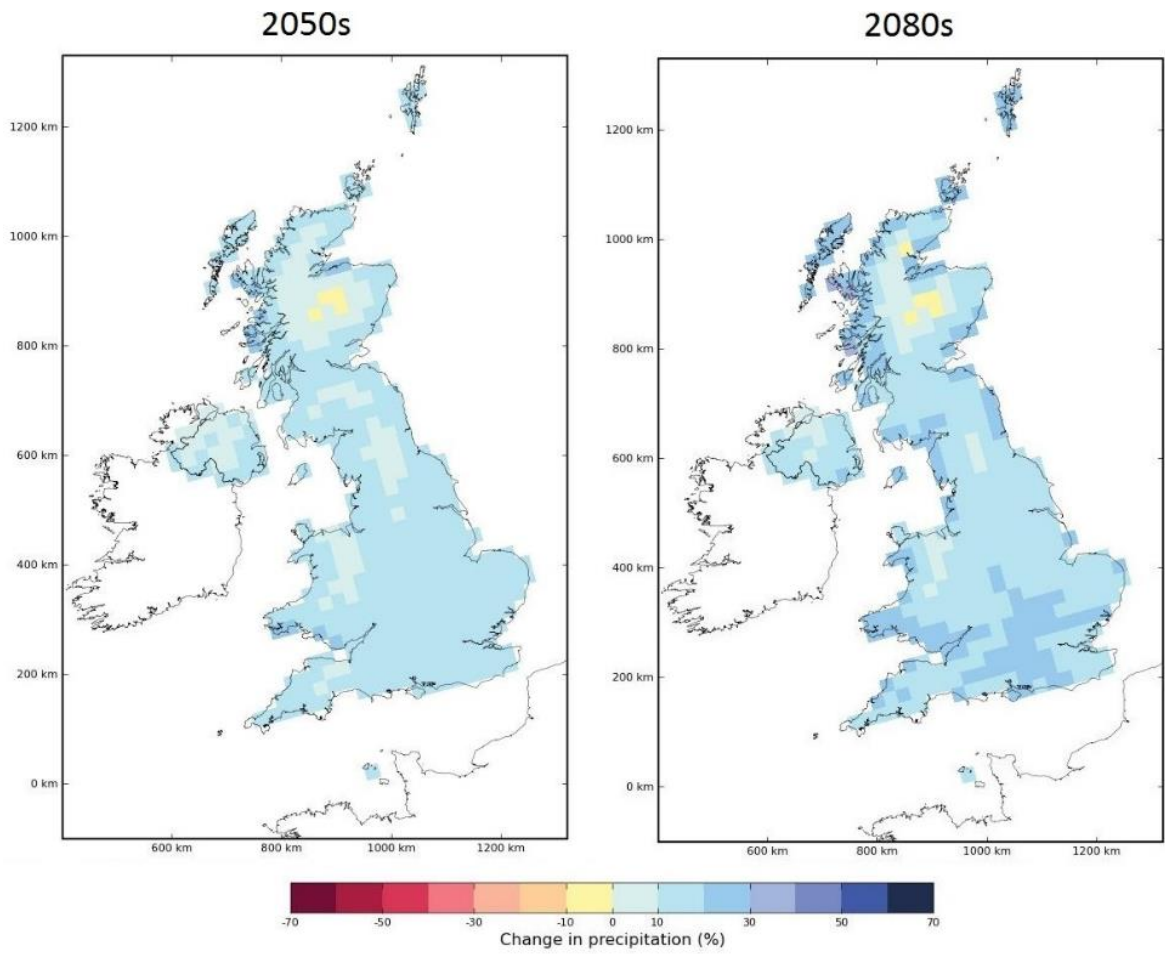
Figure 1.4 (b) Projected mean winter precipitation changes (%) at 50% probability level: central estimate. Maps are for medium emission scenario. © UK Climate Projections
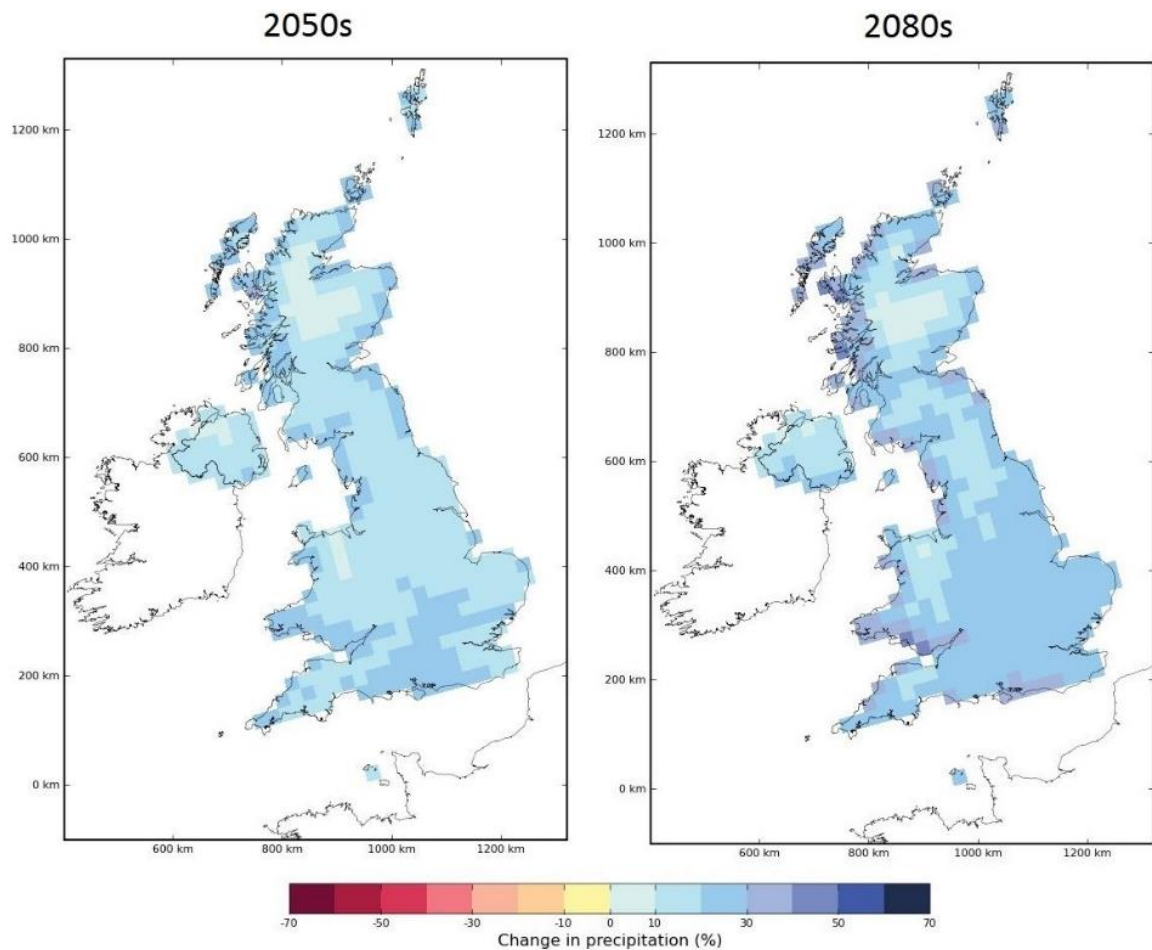
Figure 1.10 (c) Projected mean winter precipitation changes (%) at 67% probability level: unlikely to be greater than the values shown. Maps are for medium emission scenario. © UK Climate Projections 2009

1.4 Nitrogen Cycle

Soil can be seen as where nitrate starts its journey to the water body. See Figure 1.5 shows the nitrate cycle centred on processes within the soil. The source of nitrate could be livestock, organic or inorganic fertilisers, atmospheric deposition or nitrate mineralised from being previously stored within the soil. Although the majority of nitrogen (N) in soils is present in organic forms, it is the inorganic forms, such as nitrate, that are of more importance because in this form N is highly soluble and mobile, the exception being soluble organic nitrogen that is easily leached. As Figure 1.5 shows, ammonium is released from soil organic matter by the process of mineralisation. Mineralisation is the conversion of an element from an organic form to an inorganic state as a result of microbial decomposition. Immobilisation is the opposite process to mineralisation. Ammonium supplied either by mineralisation or fertilizers is readily nitrified to nitrate thus increasing the amount of nitrate available for plant uptake and leaching. Nitrification (Figure 1.5)

first oxidizes ammonium to nitrate and then, in a separate reaction, oxidises nitrite to nitrate. The process requires well aerated soil for oxidation reactions to be carried out and the rate of nitrification increases with increasing temperatures with best performance at temperatures between 20 and 30 °C (Brady and Weil, 2008, p. 626).
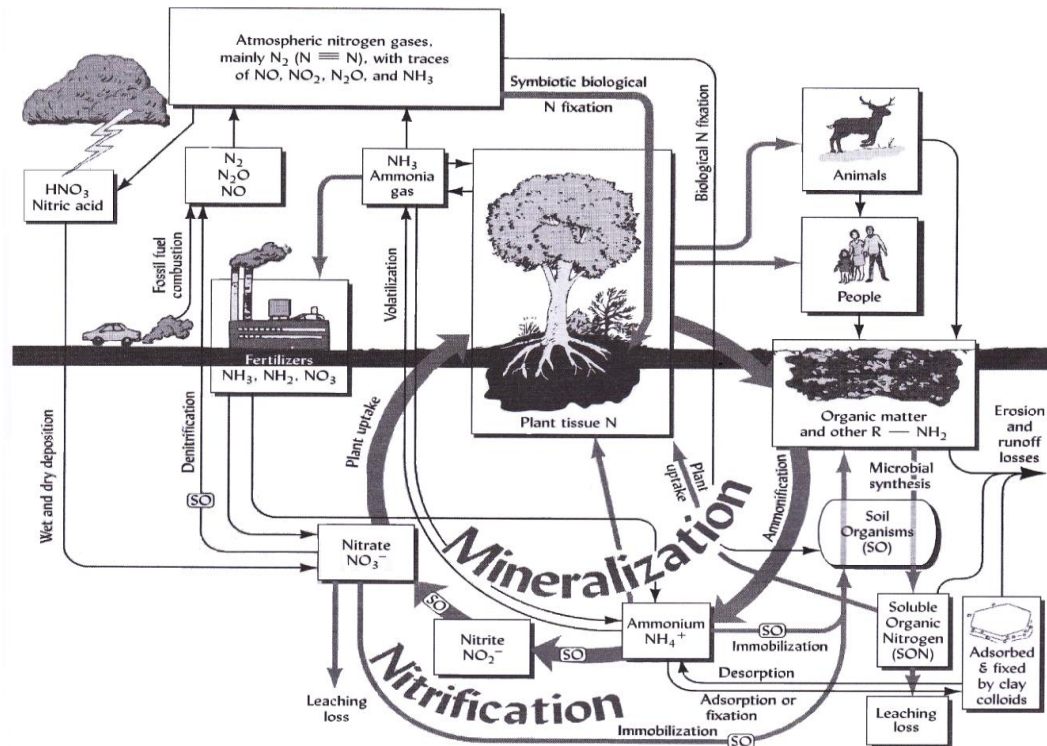
Figure 1.11 The nitrogen cycle. The boxes represent various forms of nitrogen; the arrows represent processes by which one form is transformed into another. Soil organisms, whose enzymes drive most of the reactions in the cycle, are represented as rounded boxes labelled 'SO'. [Diagram courtesy of R.Weil](Brady and Weil, 2008)

The denitrification process (Figure 1.5) acts to reduce the amount of nitrate in the soil through the reduction reaction which convert nitrate to gases (NO, N_2O, and N_2). Denitrification need anoxic conditions and is thus dominant in waterlogged soils. Optimum temperatures for denitrification are 25-35 °C but the process will occur at 2-50 °C. Although denitrification helps lower the amount of nitrate available for leaching, one of the gases it produces, nitrous oxide (N_2O) is a greenhouse gas (Brady and Weil, 2008, p. 631).

High nitrate solubility leads to it being easily carried from soil to water bodies such as rivers and underground aquifers. Spatial and temporal differences in nitrate leaching from land are associated with differences in climate, land use and soil type. Climate will influence nitrate leached through temperature and rainfall (especially average rainfall: Scholefield et al. 1996). As for land use, different activities will result in different amounts of nitrate leached. Burt and Arkell (1987) found that wooded areas will leach least nitrate with grasslands following second and arable land leaching

the most. However, for grasslands the quantity and timing of fertilizer applied, density of stocking, the age of the sward and drainage status can each exert a large influence on the amount of nitrate leached from the soil (Scholefield et al. 1993; Haigh and White. 1986). For soil type, higher leaching losses were observed from sandy soils, due to their good drainage, than from clay soils under similar climatic and cropping conditions (Vinten et al. 1994).

1.4.1 Nitrate in Soils

Clay has a somewhat different effect on soil hydrology and nitrate leaching compared to silt or sand (Jarvis, 2007; Castellano and Kaye, 2009). The higher the clay content in the soil, the poorer the soil structure will be. When a clay soil is fully saturated and its structure is poor as a result of swelling, the water flow will mainly occur by slow capillary flow, resulting in either accumulation of water at the surface, or surface runoff when water supplied exceeds the amount of water soil is able to infiltrate (Bouma and Dekker, 1978). Wormholes are thought to be the only significant route of water transfer under such conditions (Heppell et al. 2000).

As the clay soils dry out in spring, large cracks develop which, together with existing wormholes, root channels and natural soil pipes comprise a network of soil macropores (Figure 1.6). Beven and Germann (1982) give a useful and extensive review of literature on macropores. Macropores improve soil structure and facilitate rapid lateral flow, thus highlighting the importance of macropores to the hydrology of clay soils (Kneale, 1986).
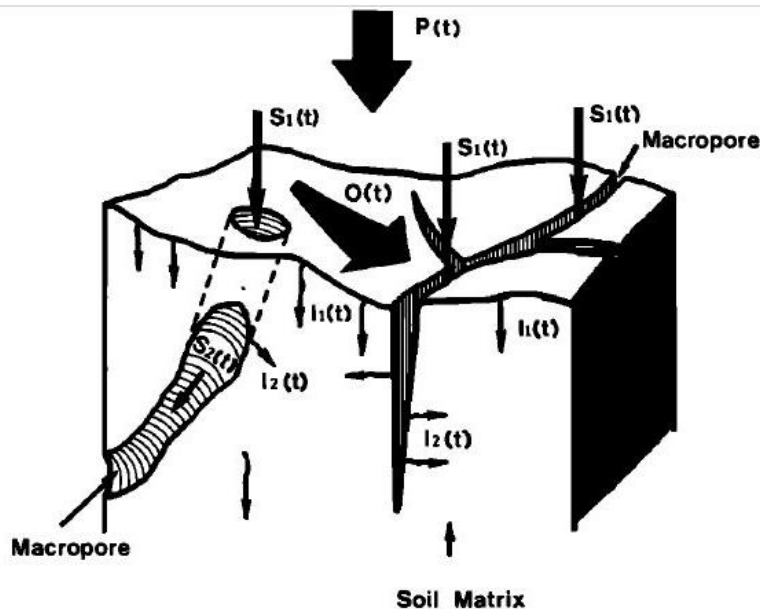
Figure 1.12 Water flows during infiltration into a block of soil with macropores. $P(t)$, overall input (precipitation, irrigation); $I_1(t)$, infiltration into the matrix from the surface; $I_2(t)$, infiltration into the matrix from the walls of the macropores; $S_1(t)$, seepage into the macropores at the soil surface; $S_2(t)$, flow within the macropores; $O(t)$, overland flow (Germann, 1980)

1.4.2 Nitrate Leaching in Undrained Clay Soil

As nitrate leaching patterns are very much dependent on soil structure, clay soils will have different effects on leaching depending on whether the soil structure is poor or well developed. If the soil is poor in structure, then nitrate will mainly be transported via surface runoff due to very low permeability of clay soil in the absence of macropores. Rainfall water in surface runoff has little chance to interact with nitrate in the soil which will affect the amount of nitrate that can be transported depending on nutrient availability at the surface. Soil that has dried out and developed cracks will rapidly leach nitrate to the groundwater as rainfall will pick up any nutrients available at the surface and short-circuit the rest of the soil (Bouma and Dekker, 1978). If no large amounts of nutrients are available at the soil surface, large storm events which deliver intense rainfall can generate enough water discharge to have a 'dilution' effect on nitrate concentrations within river water (Burt and Arkell, 1987). The presence of macropores will also mean that N locked in blocks of soil matrix will mostly be bypassed. Nitrate can be retained within these blocks of soil matrix in the topsoil for long periods of time (Vinten et al. 1994). Low-intensity rainfall is by far more effective at removing solutes from the clay soil and making them available for macropore transport as it encourages maximum water flow through the soil matrix (Heppell et al. 2000).

1.4.3 Nitrate Leaching in Drained Clay Soils

In the UK, drainage has been introduced to clay soils to make these soils more productive and easier to access for arable work such as ploughing. There was a large expansion in the amount of underdrainage in the 1950s through to the 1970s. To increase the productivity of the soils, the addition of inorganic fertilizers is often used, the majority of which may be leached if amounts in excess of the crop requirement are applied or if it is applied at periods of high or rainfall (Booltnik, 1995). Drained clay soils tend to develop a network of macropores that never fully close due to removal of excess water by drains, thus improving the soil structure even throughout wet winters by never allowing the cracks in the soil to fully close up due to swelling (Robinson and Beven, 1983). Drained clay soil will never swell up to its full potential thus not storing as much water as drainage from this type of soil starts well before it reaches field capacity (Reid and Parkinson, 1984). The drainage of a particular clay field will vary and deteriorate with time. For example, the more fragile near-surface mole drains are said to last 3-10 years while larger, deeper pipe drains last for 40-50 years (Robinson et al. 1987).

According to Robinson and Beven (1983), underdrainage reduces the magnitude of winter discharge peaks because of increased infiltration, but increases the magnitude of summer and autumn discharge peaks due to the efficiency of the drainage through macropores before the re-wetting of the soil. Robinson et al. (1987) found the same summer/autumn discharge pattern with

low flows being higher from the drained field. Soils with large drainable porosity have slow rates of rise in water table levels in response to rainfall: this is an additional factor operating to limit peak flow rates in dry, cracked soils as the drain outflow is more sensitive to the height of the water table above the drainage than it is to the hydraulic conductivity (Leeds-Harrison et al. 1986). However, there are exceptions to these patterns as Figure 1.7 shows. Hydrographs for drained and undrained plots (Figures 1.7 (a-d) show the drained plot to have higher flows during large storm events and lower flows during smaller events. The drained plot in Figure 1.7 (e) and (f) displays lower and less peaky discharges during large storm events and larger discharges during low flow conditions, compared to the undrained plot. Also, at the end of hydrograph in Figure 1.7 (b), the flow of drained plot changes from lower flows during small events to higher flows than undrained plot.
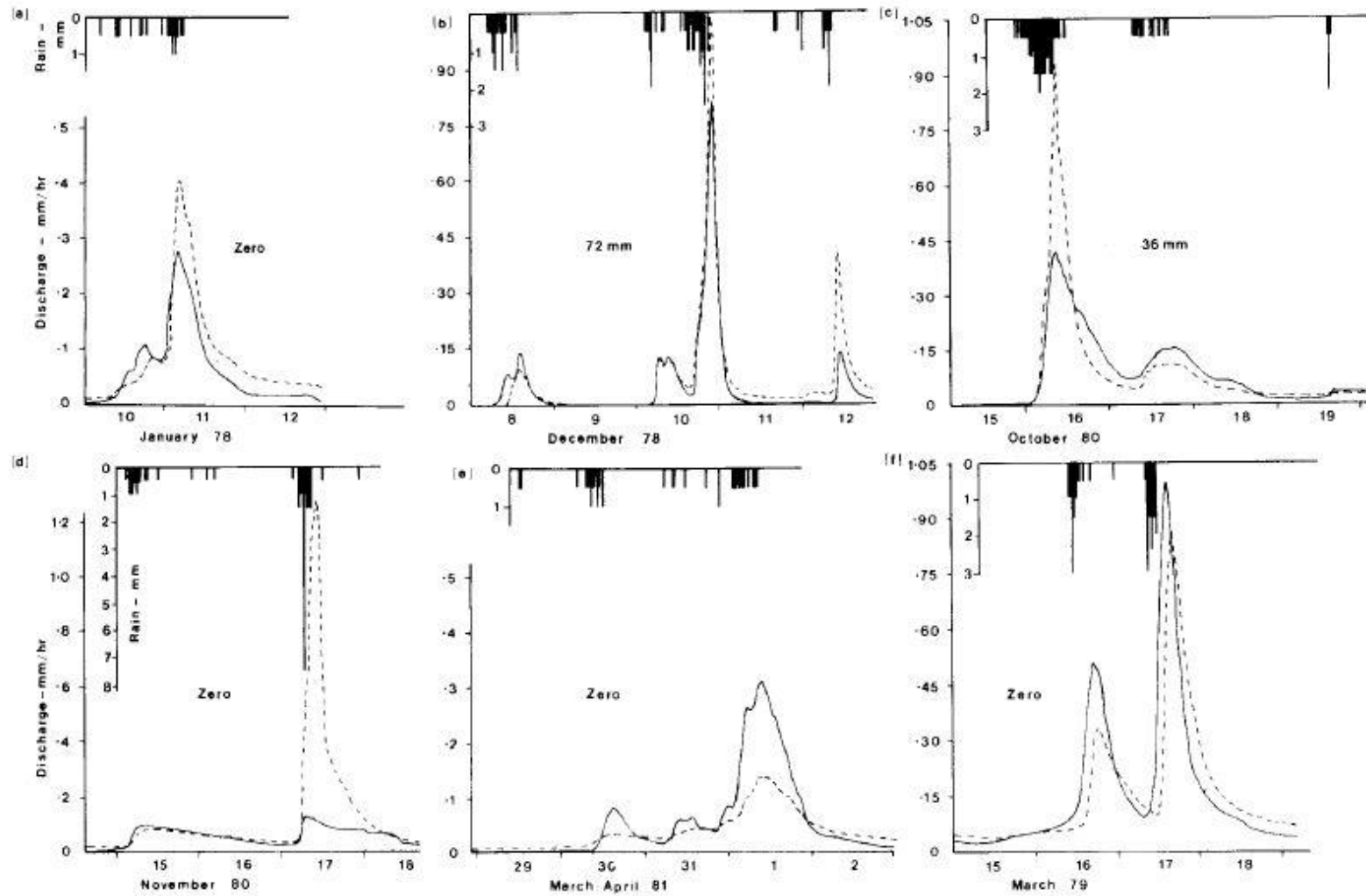
Figure 1.13 Hydrographs showing the various possible responses of the drained (dashed line) and undrained (solid line) clay soil plots

19

With efficient subsurface drainage, water movement through cracked clay soils is very rapid and this can lead to considerable losses of nutrients and pesticides (Beven and Germann, 1982; Anderson and Burt, 1990, Figure 11.5; Harris and Catt, 1999). What is more, during winter, when the soil matrix will be storing water rich in nutrients, macropores will be available for rapid transport of this water. It does increase the amount of nitrate leached as compared with the leaching through limited and slow capillary flow of undrained soils. Figure 1.8 (a) provides an insight into the displacement of old nutrient-rich water by new rain water. This scenario would mainly be observed during wet seasons, when soil water content is high. This type of event will result in almost instantaneous increase of nitrate concentrations in stream water. If the soil is initially dry, it will need to undergo thorough rewetting first, as depicted in Figure 1.8 (b). Although the presence of macropores will increase the volume of soil in contact with water, it will also provide a major route for water to short-circuit the soil matrix. Prolonged rainfall will be needed to fully rewet the soil matrix. Once the soil is rewetted, the scenario in Figure 1.8 (a) will come into action to push the now nutrient-rich soil water out by new rainwater. This type of event will result in delayed increase of nitrate concentrations in stream water.
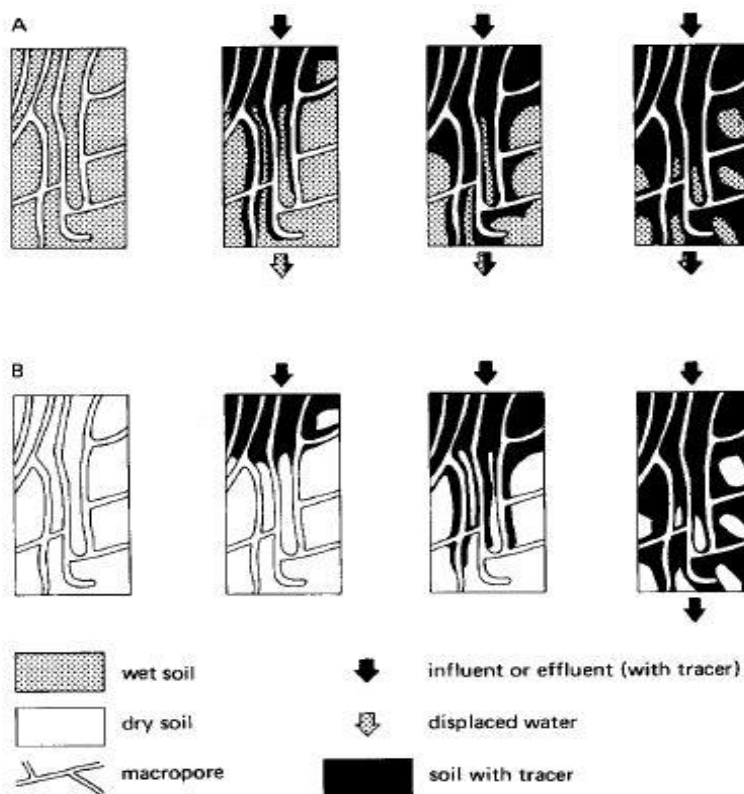
Figure 1.14 Hydrology of clay soils with a well-developed macropore network. When the soil is initially wet (A) there is both displacement of the initially present water (gray arrow) and preferential movement along the macropores (not shown). If the soil is initially dry (B) , there is no displacement but only infiltration at the soil surface and preferential movement along the macropores (Bouma, 1977).

20

Drained soils have changed from previously being the sites for nitrogen accumulation to being sites of significant sources of nitrogen (Findlay and Fisher, 2013). This change is due to the land use change facilitated by improved soil structure and consequentially, hydrological behaviour. For example, when woodland is converted into arable land, the very efficient use of nutrients and low rate of nitrate leaching is replaced with enhanced leaching rates (Williams et al. 1995). Even a far less pronounced land use change, such as a conversion from cut grassland to grazing by cattle or sheep, may result in higher amounts of nitrate leached. Ryden et al. (1984) found the annual loss of nitrate from grazed grassland to be 5.6 times greater than that from comparable cut grassland, despite a common input fertilizer N and defoliation frequency. The difference is due to the nitrogen in cut grasslands being removed with hay while grazed grasslands have high nitrogen returns from livestock in urine and faeces. Williams et al. (1995) found that the trend of increased nutrient leaching with land use change could be reversed by changing land use back to one with low-nutrient output such as planting a forest.

1.4.4 Soil Drainage and the Riparian Zone

The riparian zone is the area both above and below the ground surface that borders a river. Dense strips of riparian vegetation along the banks of the river facilitate the removal of nutrients from groundwater before it reaches the stream water through the process of denitrification and uptake by the vegetation (Brady and Weil, 2008 p.21, p.634). Riparian zones are often implemented as mitigation measures to reduce nutrient supply to stream water from the catchment in order to comply with the EU Nitrates Directive (91/676/EEC).

Drainage can provide short-circuiting pathways for delivery of shallow soil water and surface runoff high in nutrients collected on its way to streams, which may minimize the opportunities for removal of nitrogen through the riparian-zone processes as it short-circuits the soil (Tomer et al. 2003). It should therefore be noted that implementation of riparian zones as mitigation measures for under-drained clay soils that drain directly into stream might not be effective and should be carefully assessed before implementation.

1.4.5 Nitrate Leaching From Grazed Improved Grasslands

Agricultural land accounts for 71% of total land area in the UK; 72% of this agricultural land is grassland (Defra, 2015). The amount of nutrients leached from grassland tend is usually lower than from other land uses (Lazenby, 1981, Johnes et al., 1996). Within grassland, the presence of dense growth of plants lowers the leaching of nutrients through absorption and uptake in the organic matter of the plant which later decomposes slower, thus delaying the release of nutrients by storing it in the soil. However, it has been shown that grassland can be a significant source of nutrients

(Greene et al., 2013) and the amount leached will depend on the particular use of the grassland (Ryden et al. 1984,)

Grazed grasslands have different nitrate leaching patterns compared to cut grasslands. Cut grassland may have large amounts of N removed with hay while grazed grassland will have most of the N consumed by the animals returned back into the soil within excreta. However, Ryden et al (1984) argue that removal of N with hay only acts to merely delay the leaching of nitrate after the disposal of slurry and farmyard manure. Within intensively managed grazed grasslands, nitrate leaching will be also dependent on the type of grassland planted (Stout et al. 2000). There are other factors further complicating the patterns and amounts of nitrate leached. Firstly, animals might not graze the field all year round or there could be a field rotation for the grazed fields and cattle are kept indoors and fed on hay during cold winter months. Secondly, the compaction of the soil by trampling can result in a degraded soil structure followed by reduced drainage and a less productive field which will increase surface runoff (Oenema et al., 1997). The surface runoff might bypass riparian zones where some of the nitrate might be removed by plants or through denitrification. The nitrate can then reach the water body as a significant source of pollution.

1.4.6 The Effect of Projected Climate Change on Nitrate Leaching

Temperature, soil moisture and water movement are key for understanding the N cycle. Projected climate change, as mentioned in section 1.3, will affect these variables through fluctuation in temperatures and rainfall. Warmer temperatures will result in increased nutrient cycling while wetter conditions will result in more nutrient leaching. As section 1.3 identifies, summers are projected to be hotter and drier while winters are projected to be warmer but wetter. This will result in even more extreme leaching events in summer due to accumulation of nutrients in soil after prolonged periods of dry weather. Although winters might not see such extreme leaching events, the overall amount of nutrients leached would increase due to increased nutrient availability with increasing temperatures and increased transport availability.

1.5 The Demonstration Test Catchment Platform for Nitrogen Research

DTC is a UK government-funded project designed to provide robust evidence regarding how diffuse pollution from agriculture can be cost-effectively controlled to improve and maintain water quality in rural river catchment areas. The project investigates four English catchments (National Demonstration Test Catchments Network, 2016): the River Eden in Cumbria, the River Wensum in Norfolk, the River Avon in Hampshire and the River Tamar on the Devon/Cornwall border. These four catchments are representative of 80% of UK soil/ rainfall combinations and the major farm types in England and Wales. The catchments were selected in order to build on existing

infrastructure, datasets, knowledge and farming contacts developed through previous and ongoing initiatives, which have not previously been well linked. (National Demonstration Test Catchments Network, 2016). The Demonstration Test Catchments (DTC) project has developed a large-scale research platform to test measures for reducing diffuse pollution from agriculture whilst maintaining agricultural productivity. The project monitors the stream water quality and biology and researches how the catchment responds to storm events. Many aspects of water quality are measured including the fluxes of nitrogen, phosphorus, sediment and total amount of water leaving the catchment at a high temporal resolution of 30 minutes. There is close collaboration with farmers in the catchment to test a range of mitigation measures. Eden DTC comprises 'communities' of practitioners that link researchers, policy-makers, delivery bodies (e.g. EA, NE, Rivers Trusts), farmers, advisors, water companies and NGOs (EdenDTC, 2016). Eden DTC has three focus sub-catchments. The different sub-catchments were chosen to best represent the various land use types, soil types and hydrology of the Eden river catchment. The study site is one of these sub-catchments.

The Demonstration Test Catchment (DTC) project can provide much more information than traditional approaches using automatic water samplers, so there will be very many more storm events to consider than would be possible without the latest high-frequency measurements. Kirchner et al (2004) highlight the importance of high-frequency data in enabling the detailed understanding of catchment hydrology, especially of small catchments, where hydrological response timescales are much shorter. Low-frequency measurements would not provide an accurate picture of hydrological responses of a catchment this size. However, they also point out that long-term measurements are needed in order to obtain data under a wide range of hydrological conditions. Long-term measurements would maximise the chances of recording any unusual events and hydrological responses as well as help establishing what events are more 'typical' to the hydrology of that particular catchment.

1.6 Aims and Objectives of This Research

After many years of research into nitrate dynamics, there are still issues with the management of nitrate leaching and it is still leading to eutrophication. The recent development of high temporal resolution monitoring of nitrate flux in river channels enables old ideas to be tested and has the potential for new insights into the nitrate dynamics at the catchment scale. These new insights have the potential to drive new management techniques and policies of the use of N in the agricultural environment. Therefore, the aim of this thesis is to analyse the nitrate leaching patterns of grassland underlain by drained clay soil and how the N export from landscape to the river system may be

affected by future climate change. The aim of this research will be achieved by carrying out this set of objectives:

1. Create an automated analysis approach to analyse high temporal frequency data spanning a the period of several years to enable the identification and analysis of all the storm events occurring throughout the length of the dataset and associated storm data such as identification of the type of hysteresis for each storm event.

2. Characterise the hydrological dynamics of a drained clay soil grassland catchment, taking advantage of the high-frequency DTC measurements to gain new insights into nitrate leaching;

3. Analyse how nitrate leaching patterns depend on transport (rainfall, soil hydrology) and source (supply);

4. Investigate the effect of antecedent conditions on hydrology and leaching patterns of drained clay soil grassland;

5. Investigate the potential effect that projected climate change might have on N leaching through changes in temperature and precipitation;

6. Develop possible mitigation measures and solutions.

The overview of the research approach to be taken to answer these objectives is below. The automated analysis approach is achieved by using the Python programming language. Python enables visualization and analysis of large datasets and is able to carry out multiple complex calculations/actions very quickly. Characterisation of the hydrological dynamics of a drained clay soil grassland catchment and gaining an understanding of how nitrate leaching patterns depend on transport (rainfall, soil hydrology) and source (supply) can be addressed by observing river water discharge and nitrate concentrations in response to storm events. To classify the types of nitrate response, concentration will be plotted against discharge for each storm event. Depending on the direction and shape of the resulting graph show the patterns of nitrate leaching. This approach has been used by Siwek et al (2013), Bieroza and Heathwaite (2015), Perks et al. (2015) and Bowes et al (2015) to deduce the nutrient delivery mechanisms to the river water.

Investigation of the effect of antecedent conditions on hydrology and leaching patterns of drained clay soil grassland can be carried out by analysis with Pearson's correlation matrix and stepwise multiple regression to find the most influential antecedent conditions. Investigation of the effect that projected climate change might have on N leaching through an increase in temperature and precipitation can be carried out by simulating the river discharge in response to future weather from

the UKCP09 weather generator using a catchment hydrological model. These simulation results will then be used to drive the stepwise multiple regression model to predict changes in N for the future climate scenarios. The development of possible mitigation measures and solutions will also include the results from this projected climate change scenario.

Chapter 2: Methodology

2.1 Introduction

This chapter describes the methods used in this research. It starts with site description and the description of the project that supplied the data used and description of said data (section 2.2). It then moves on to description of measures taken to assure data quality (section 2.3), event classification and the methods of statistical analyses undertaken to analyse the data associated with the storm events (section 2.4). Methods of calculation of nitrate concentration and flux (section 2.5), hysteresis loops (section 2.6), seasonal regression (section 2.7), stepwise multiple regression (section 2.8), and future changes in nitrate concentrations (section 2.9) are also described. Finally, section 2.10 describes how automated analysis was used to enable the analysis of large amounts of storm events.

2.2 Site Description

The Newby Beck sub-catchment is located south-east of Penrith town in Cumbria, in the North West of England (Figure 2.1). The sub-catchment, part of River Eden catchment (Figure 2.1, Figure 2.2), is 12.5 km^2 in area. The study site is part of the River Eden Demonstration Test Catchments (DTC) project which in turn is part of National Demonstration Test Catchments Network. The total annual rainfall ranges from 602 mm to 1264mm (for years 2011-2015, average of 1042 mm). Average annual temperature 7.9°C to 8.9°C (for years 2011-2015, average of 8.3°C). Land use is mainly improved grassland with small areas of arable and acid grassland (Perks at al., 2015). Soil of most of the catchment is loamy and clayey with impeded drainage (Cranfield University, 2016). The main water protection issues associated with this type of soil are associated with overland flow from compacted or poached fields. Organic slurry, dirty water, fertiliser, pathogens and fine sediment can all move in suspension or solution with overland flow or drain water. The rest of the catchment has loamy freely draining soils. The main water protection issues associated with loamy soil are associated with groundwater contamination with nitrate; plus siltation and nutrient enrichment of streams from soil erosion on certain areas of these soils (Cranfield University, 2016).

Analysis of the catchment by Allen et al. (2010) showed that aquifers in the Newby Beck catchment are likely to be confined. The catchment hydrogeology is dominated by the superficial deposits and hence a significant proportion of infiltration may result in relatively rapid shallow groundwater flow to the streams.
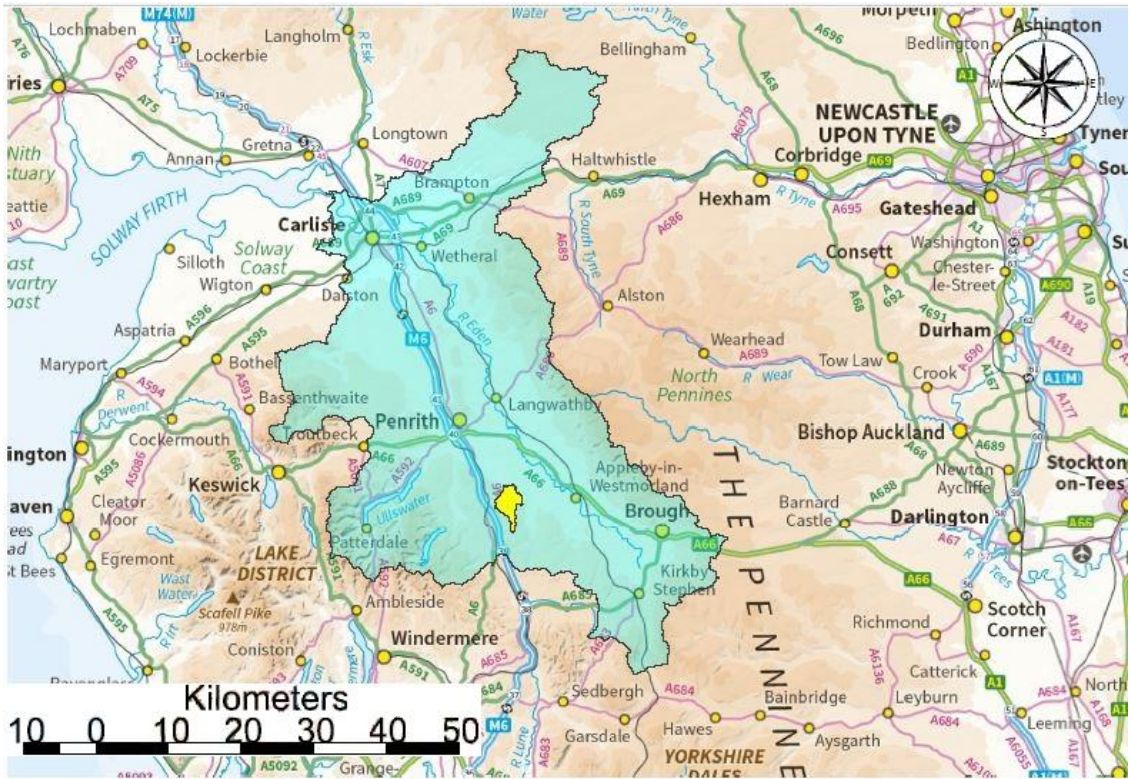
Figure 2.1 Eden DTC catchment in light blue with Newby Beck sub-catchment marked in yellow (Contains OS data © Crown copyright 2016)
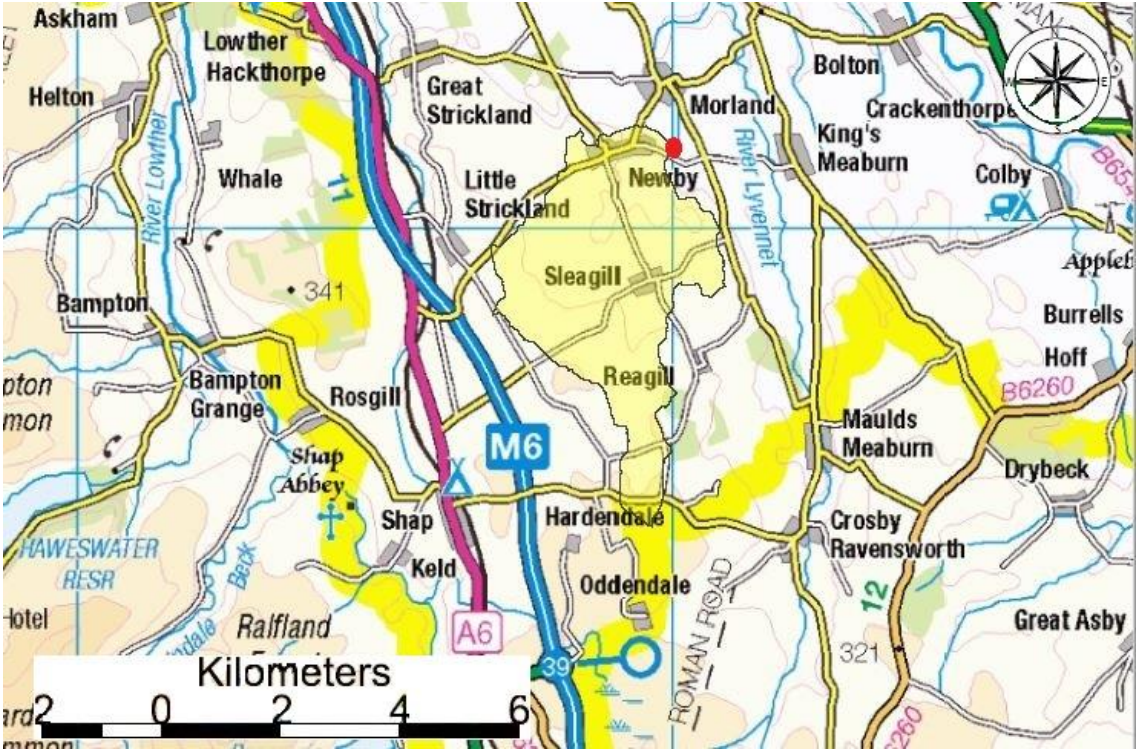
Figure 2.2 Newby Beck sub-catchment. Water from across the entire catchment is conveyed through the river networks and monitored at the catchment outlet. The outlet monitoring station is represented by the red dot. (Contains OS data © Crown copyright 2016).

Water from across the catchment is conveyed through the river networks and monitored at the catchment outlet. This monitoring station is represented by the red dot on Figure 2.2. This station is equipped with a Hach Lange Nitratax SC sensor which measures the nitrate concentration of the river on a 30 minute temporal resolution (EdenDTC, 2016). An automatic weather station measures rainfall, air temperature, net radiation, relative humidity, wind speed, wind direction; two separate tipping bucket rain gauges measure rainfall. The high frequency measurements taken at this station form the basis for addressing the proposed objectives. The data will be used in automated analysis and the outcome of the analysis will then be interpreted to address the rest of the objectives.

2.2.1 Data

Rainfall (EdenDTC, 2016[a]), discharge (EdenDTC, 2016[b]) and nitrate (EdenDTC, 2016[c]) data for the entire record can be seen in Figure 2.3. The hydrograph is flashy, with low base flow and easily distinguished storm events. The maximum measured concentration throughout the record was 30 mg NO_3 l^{-1}. Nitrate concentrations can be seen to have different responses to rainfall events. A spike in nitrate concentrations in response to rainfall can be observed on some occasions while others result in a fall in nitrate concentrations. A few examples of different responses of nitrate concentrations to rainfall are shown and discussed in more detailed below in Figures 2.4 and 2.5.
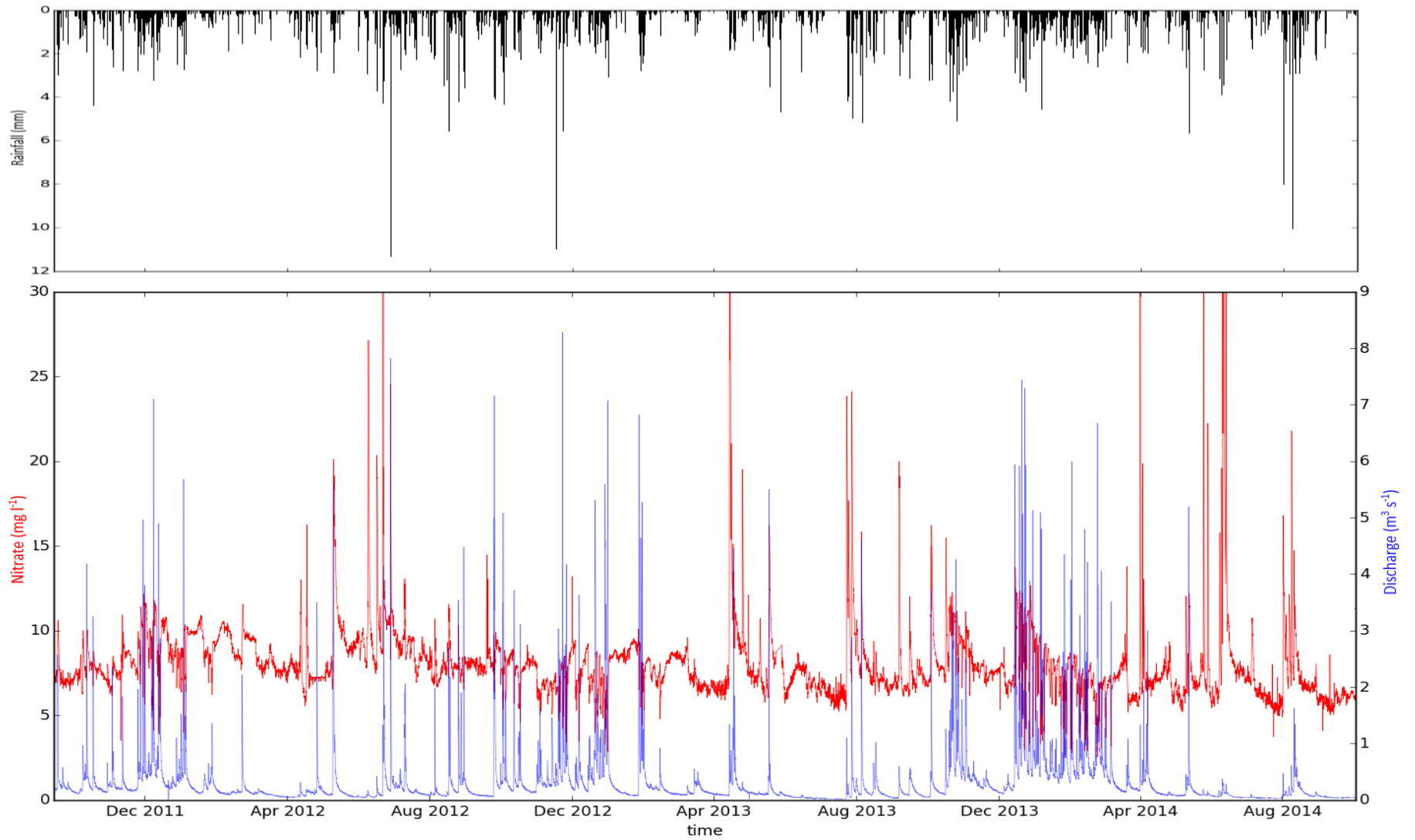
Figure 2.3 Half hourly rainfall (top) nitrate concentration (red) and discharge (blue), for 14/09/2011 12:00 and 02/10/2014 13:00

Figure 2.4 shows rainfall, discharge and nitrate concentrations after a prolonged period of no or very little rainfall. The nitrate concentrations are shown to be low during the dry period with large spikes during the first large rainfall event. The first spike in nitrate concentrations is soon followed by another, similar magnitude, increase during the second rainfall event. The subsequent third rainfall event delivers a spike in nitrate concentrations that is not as large in magnitude.
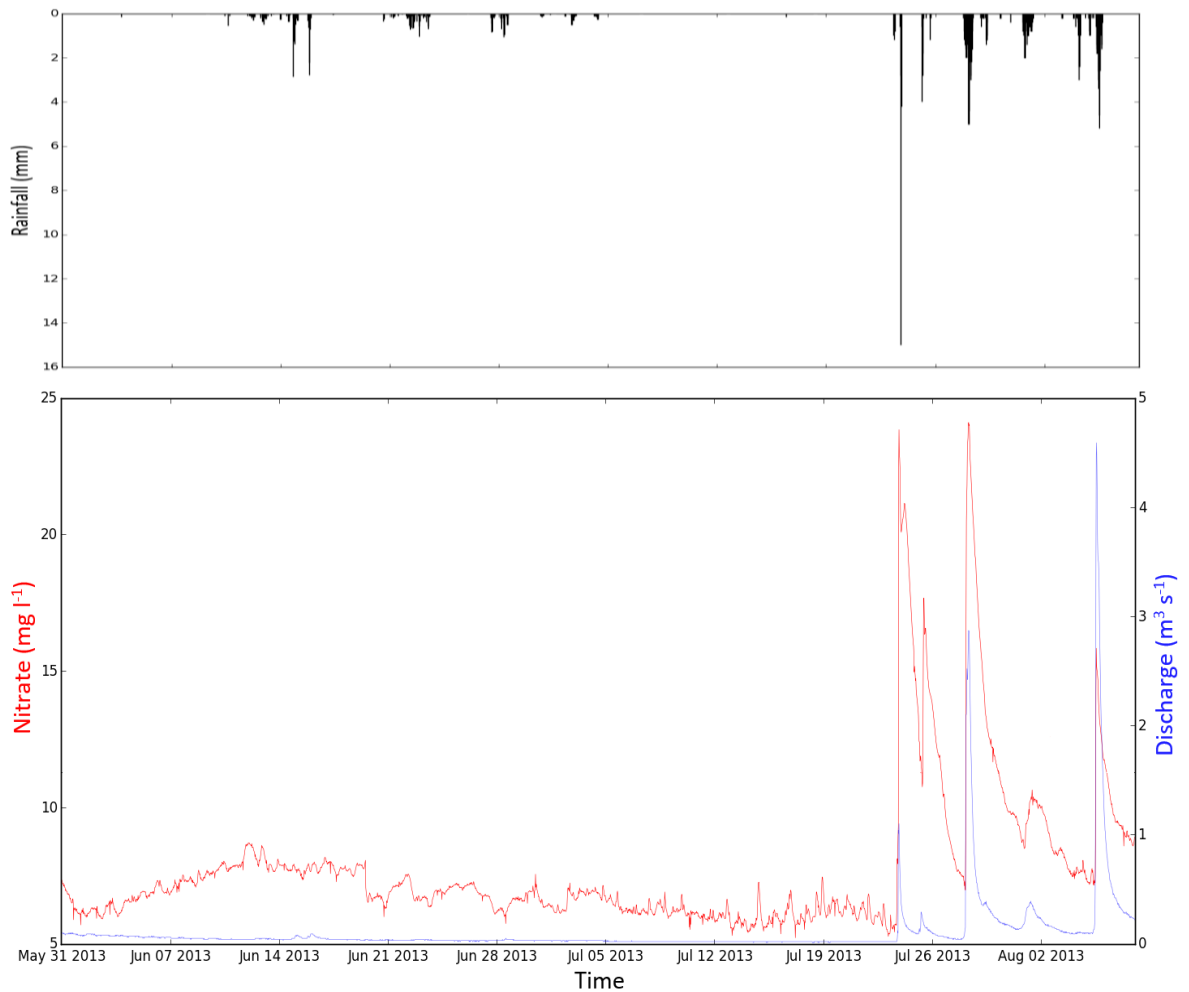
Figure 2.4 Nitrate concentrations after a prolonged dry period. Rainfall is displayed on the top graph, discharge in transparent blue, nitrate in red on the bottom graph.

Figure 2.5 shows rainfall, discharge and nitrate concentrations after a prolonged period of rainfall. The nitrate concentrations are high at the start of the period and thereafter the rainfall events result in a dilution effect as nitrate concentrations plunge with increase discharge and bounce back after the event. The overall trend is downward as maximum nitrate concentrations fall, most likely due to exhaustion of nitrate supply within the soil.
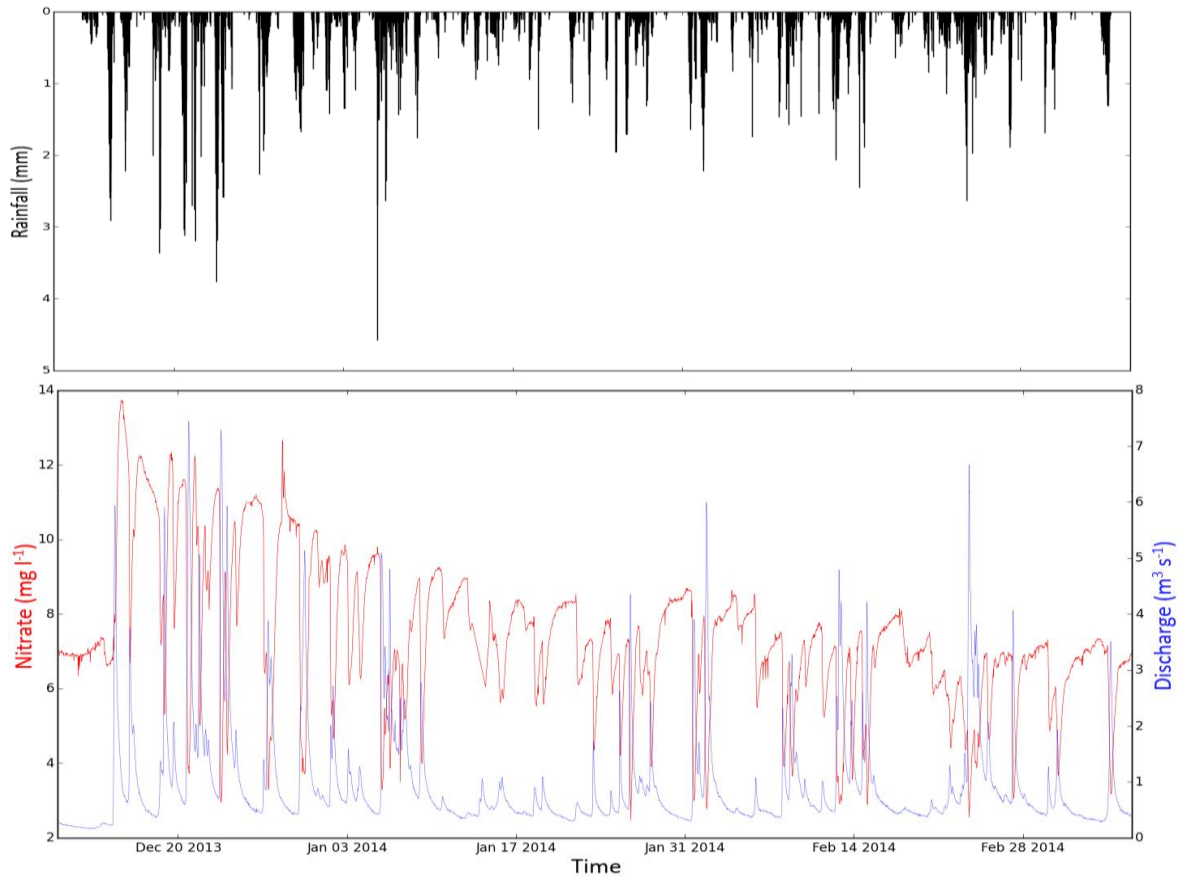
Figure 2.5 Nitrate concentrations after prolonged wet period. Rainfall is displayed on the top graph, discharge in transparent blue, nitrate in red on the bottom graph. Note the seasonal decline in baseflow nitrate concentrations across the winter.

2.2.2 Climate Change Projections for the Area

As can be derived from section 1.3, the region will, overall experience hotter, drier summers and wetter, warmer winters. As mentioned in section 1.3, the UK Met Office projects the temperature warming to be greater for summer (Figure 1.1) than winter (Figure 1.2). Section 1.3 also indicate mean summer precipitation (Figure 1.3) to be expected to decrease while mean winter precipitation to be expected to increase (Figure 1.4). This section will therefore concentrate on a more detailed climate change projections for the region for mean summer temperatures and mean winter precipitation.

Figure 2.6 show projected changes in mean summer temperatures for Newby Beck for 2050s (Figure 2.6 (a)) and 2080s (Figure 2.6 (b)). The increase in summer mean temperature for the medium emissions scenario is (at 33% probability level) unlikely to be less than 2.1 °C by the 2050s. Central estimate (at 50% probability level) for the same time period is 2.6 °C. The temperature increase by 2050s (at 67% probability level) is unlikely to be greater than 3.1 °C (Figure 2.6 (a)). The increase in summer mean temperature at medium emission scenario by the

31

2080s is unlikely to be less than 3.2 °C. Central estimate (at 50% probability level) for the same time period is 3.8 °C. The temperature increase by 2050s (at 67% probability level) is unlikely to be greater than 4.5 °C (Figure 2.6 (b)).
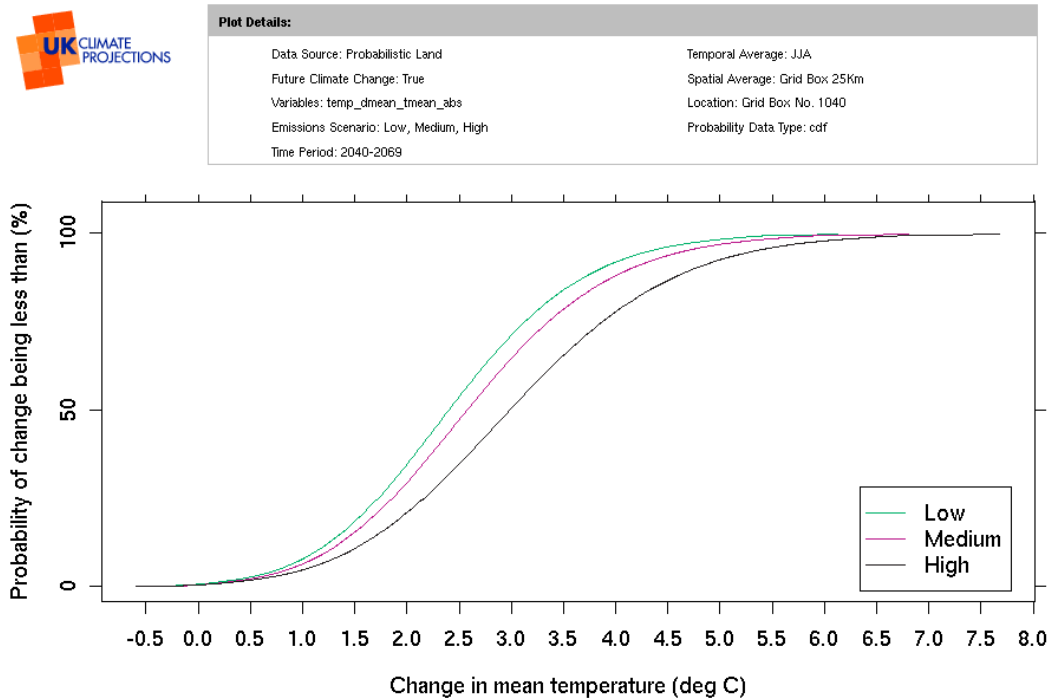
(a) 2050s
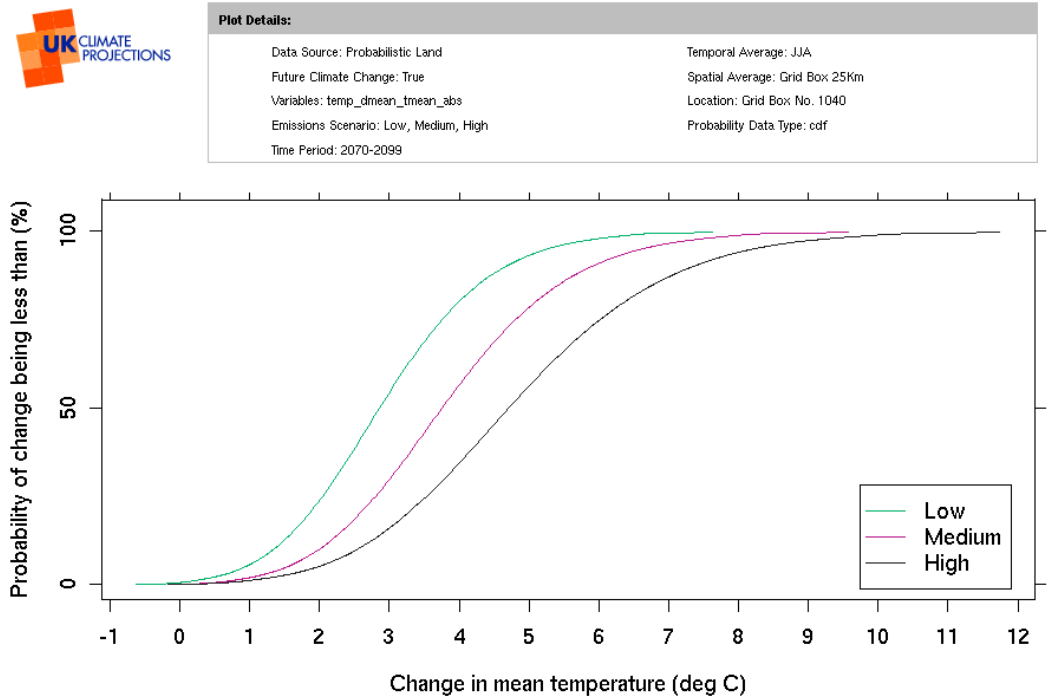
(b) 2080s

Figure 2.6 Projected changes in Newby Beck mean summer temperatures. © UK Climate Projections 2009

Figure 2.7 show projected changes in mean winter precipitation for Newby Beck for 2050s (Figure 2.7 (a)) and 2080s (Figure 2.7 (b)). The increase in mean winter precipitation for the medium emissions scenario is (at 33% probability level) unlikely to be less than 6% by the 2050s. Central estimate (at 50% probability level) for the same time period is 9%. The precipitation increase by 2050s (at 67% probability level) is unlikely to be greater than 12% (Figure 2.7 (a)). The increase in mean winter precipitation for the medium emissions scenario by the 2080s is (at 33% probability level) unlikely to be less than 7%. Central estimate (at 50% probability level) for the same time period is 11%. The precipitation increase by 2080s (at 67% probability level) is unlikely to be greater than 15% (Figure 2.7 (b)).
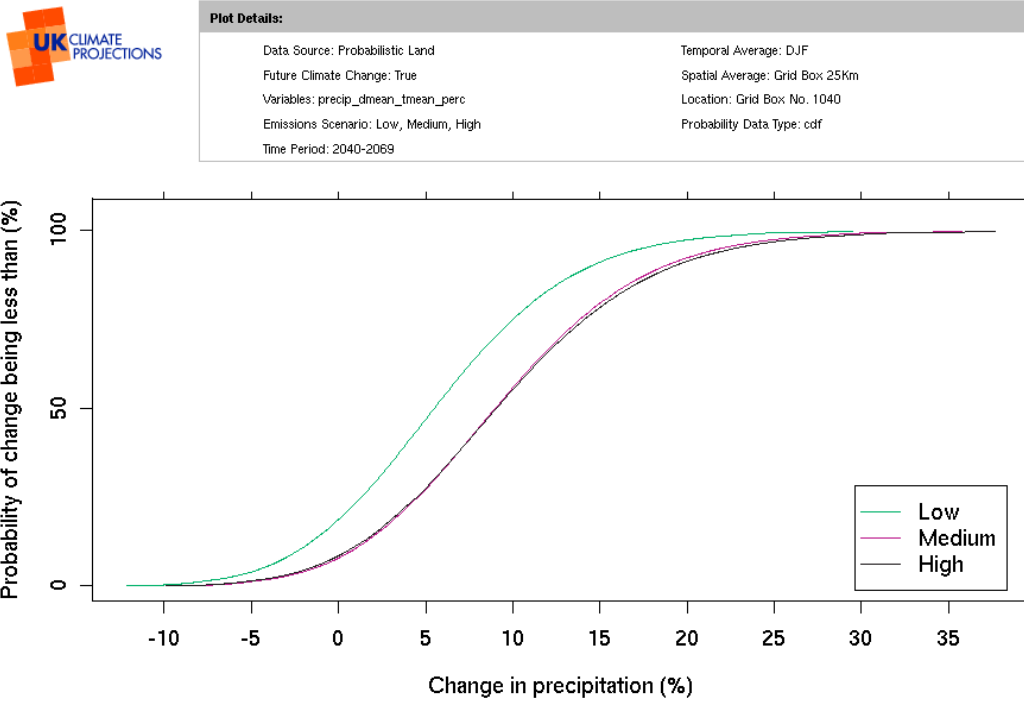
(a) 2050s
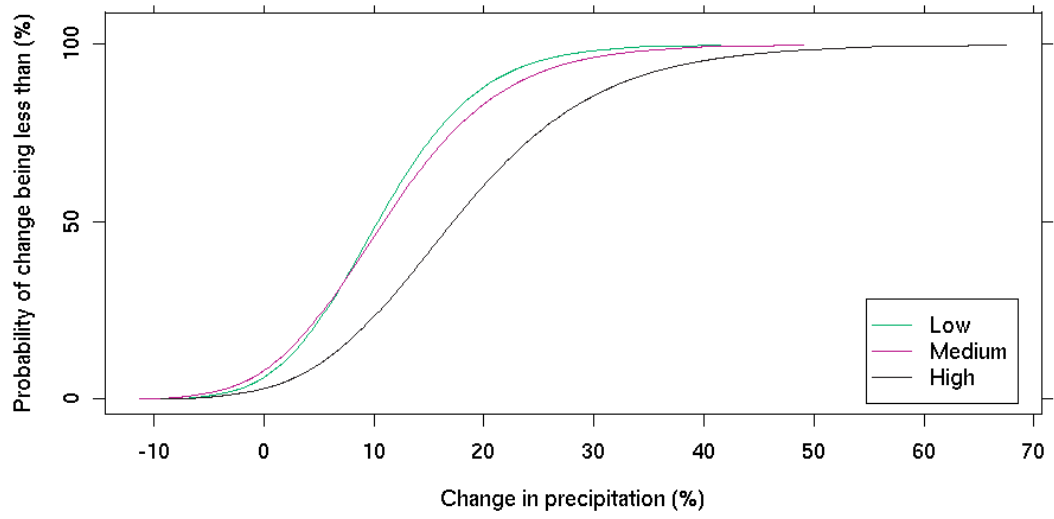
(b) 2080s

Figure 2.7 Projected changes in Newby Beck mean winter precipitation. © UK Climate Projections 2009

2.3 Data Quality

The dataset (Figure 2.8) comprises rainfall, discharge, dissolved oxygen, electrical conductivity, pH, turbidity, total phosphate, total reactive phosphate, nitrate and water temperature records available from 14/09/2011 to 02/10/2014. The dataset is of very high temporal resolution, with most of these measurements taken at a 30-minute interval. The data record has been checked for missing data. For nitrate, smaller gaps, of up to four data points, or larger gaps, of 5 or more data points, with no apparent change in discharge or rainfall have been linearly interpolated. Large gaps of missing nitrate data, where there were visible changes in antecedent conditions observed, were excluded from analysis together with associated storm events.

DateTime	Rainfall	Discharge	Dissolved oxygen	Electrical conductivity	pH	Turbidity	Total Phosphate	Total Reactive Phosphate	Nitrate	Water temperature
	mm	ms/s	%	us	pH	NTU	mg/l	mg/l	mg/l	Degrees C
14/09/2011	0	0.25	111.6	550	8.34	6.1	0.08	0.04	7.33	12.1
14/09/2011	0	0.25	111.7	548	8.35	4.6	0.09	0.05	7.3	12.19
14/09/2011	0	0.26	112	547	8.36	4	0.07	0.04	7.22	12.27
14/09/2011	0	0.25	111.9	545	8.36	4.2	0.06	0.04	7.18	12.38
14/09/2011	0	0.25	111.5	545	8.35	3.8	0.06	0.03	7.16	12.43
14/09/2011	0	0.24	111.4	544	8.36	3.8	0.06	0.03	7.2	12.61
14/09/2011	0	0.24	111.1	543	8.35	3.9	0.06	0.03	7.15	12.76
14/09/2011	0	0.24	110.8	543	8.35	4.2	0.06	0.03	7.17	12.82
14/09/2011	0	0.24	110.7	542	8.35	3.9	0.06	0.03	7.09	12.9
14/09/2011	0	0.24	109.8	542	8.35	3.8	0.06	0.03	7.09	12.89
14/09/2011	0	0.24	108.2	543	8.33	3.8	0.06	0.03	7.1	12.88
14/09/2011	0	0.24	106.8	543	8.33	3.9	0.06	0.03	7.09	12.83
14/09/2011	0	0.24	105.7	543	8.31	3.8	0.06	0.03	7.08	12.75
14/09/2011	0	0.24	104.8	543	8.3	3.8	0.06	0.03	7.07	12.68
14/09/2011	0	0.24	104	544	8.29	4.4	0.05	0.03	7.08	12.56
14/09/2011	0	0.23	103.8	544	8.29	4.9	0.06	0.03	7.09	12.44
14/09/2011	0	0.24	103.8	545	8.27	4.6	0.06	0.03	7.11	12.3
14/09/2011	0	0.23	103.7	546	8.27	4.5	0.05	0.03	7.1	12.13
14/09/2011	0	0.23	103.7	546	8.26	4.6	0.05	0.03	7.11	11.98
14/09/2011	0	0.23	103.4	548	8.25	4.7	0.06	0.03	7.11	11.79

Figure 2.8 Dataset excerpt

2.4 Event Classification

The event classification started with separation of the entire hydrograph into base flow and storm flow. Hydrograph separation was done using a local-minimum method (Sloto and Crouse, 1996). Local minimum method checks each day to determine if it is the lowest discharge in one half the interval minus one day [0.5(2D-1) days] before and after the day being considered. If it is, then it is a local minimum and is connected by straight lines (using linear interpolation) to adjacent local minima. D is the duration of surface runoff, given by $D= A^{0.2}$ where D is the number of days after which surface runoff ceases, A is the drainage area in square miles. The interval $2D*$ is the odd integer between 3 and 11 nearest to $2D$ (Pettyjohn and Henning, 1979; Linsley et al. 1982)

A storm event is defined within this research project as follows (Figure 2.9):

- A storm event starts when base flow is equal to discharge and ends when base flow equals the initial discharge again. As the storm events are selected automatically rather than manually by eye, they may need to be refined.

- A storm event can have local peaks; this happens when discharge falls before the main storm event peak or rises after it. If discharge for the local peak rises by one third and then falls one third below its local peak, it is counted as a separate storm event.

- To refine the storm events, each event had to match certain criteria: be at least 5 measurements long (Evans et al. 1999) to facilitate a reasonable comparison of falling and rising limb (2 measurements for each limb and one for the peak); the main peak should have risen at least 50% above the start discharge (based on Bowes et al (2015) setting similar thresholds).
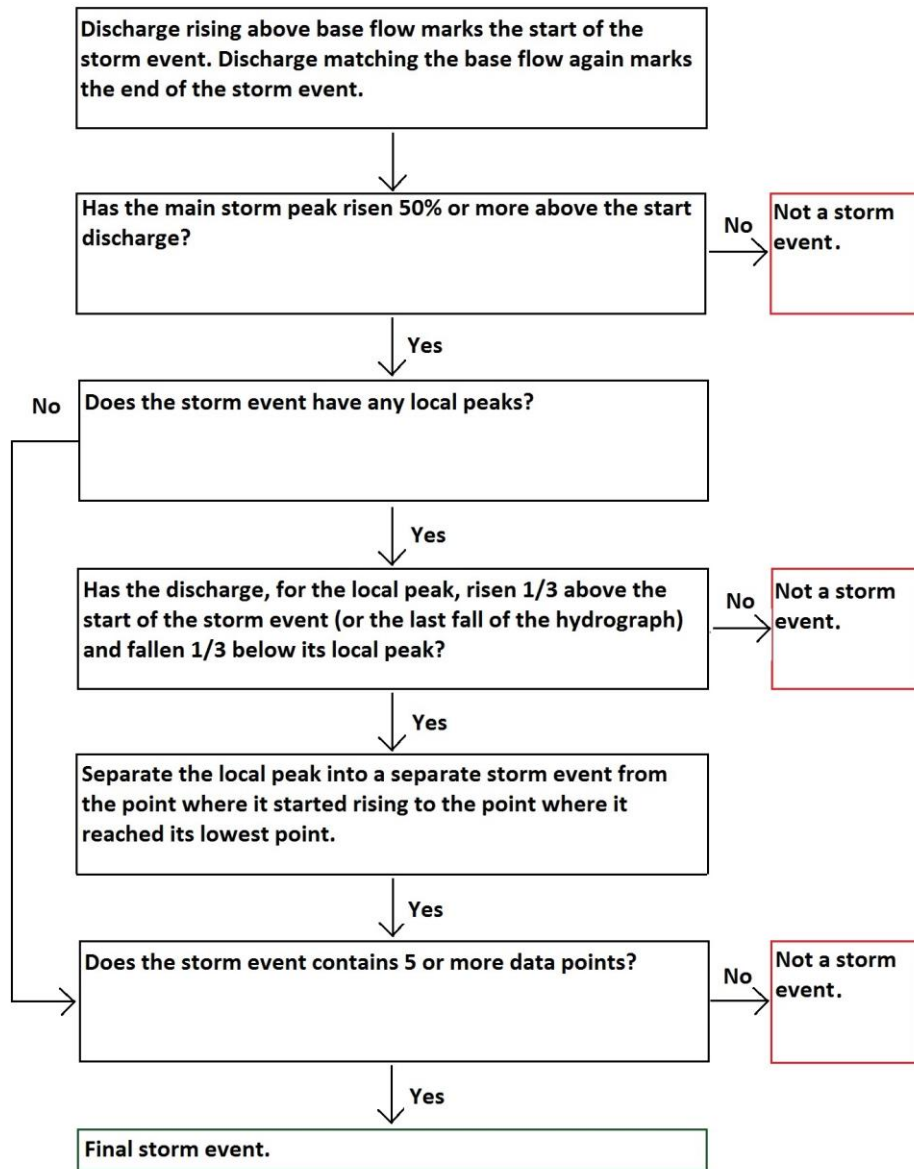
Figure 2.9 The classification process for storm events.

2.5 Nitrate Concentration and Flux

Mean nitrate concentration was calculated for each storm event. Also, each storm event was checked for its minimum and maximum nitrate concentrations. Total nitrate flux in each half-hour was also calculated for each storm. The nitrate flux for each half an hourly measurement was calculated first (Equation 2.1) and then summed to give the total event N. *F* stands for nitrate flux, *Q* for discharge and *C* for concentration. The multiplication by 1800 is due to the time interval (60 seconds * 30 minutes).

$F\ (kg) = Q\ (m^3\!/s) * C\ (mg/l) * 1800$ (Eq. 2.1)

2.6 Hysteresis

Each storm event was checked for hysteresis direction and magnitude applying methodology as used by Siwek et al. (2013). To start with, residuals were calculated for each storm event. A residual is the difference between the observed Y value and the estimated Y value ($e_i = \hat{Y} - Y_i$). The estimated values are obtained using linear regression, Equation (2.2), where $\hat{Y}$ stands for estimated Y (nitrate in this case), a – the intercept, b – the slope of the regression line and X – the independent variable (discharge):

$$\hat{Y} = a + bX \qquad \text{(Eq. 2.2)}$$

Each storm event was then divided into rising limb (from the start up to the peak discharge of the storm event) and falling limb (from peak discharge to the end of the storm event). A mean was then calculated for the residuals of the rising and falling limb of each storm event; see Figure 2.10 for the visualisation of this calculation. If the average residual of the rising limb is larger than the average residual of the falling limb ($\bar{e}_r > \bar{e}_f$), the hysteresis direction is clockwise. If $\bar{e}_r < \bar{e}_f$, then hysteresis direction is anticlockwise. Hysteresis magnitude is described by the range between the average residuals of the rising and falling limbs ($|\bar{e}_r| + |\bar{e}_f|$).
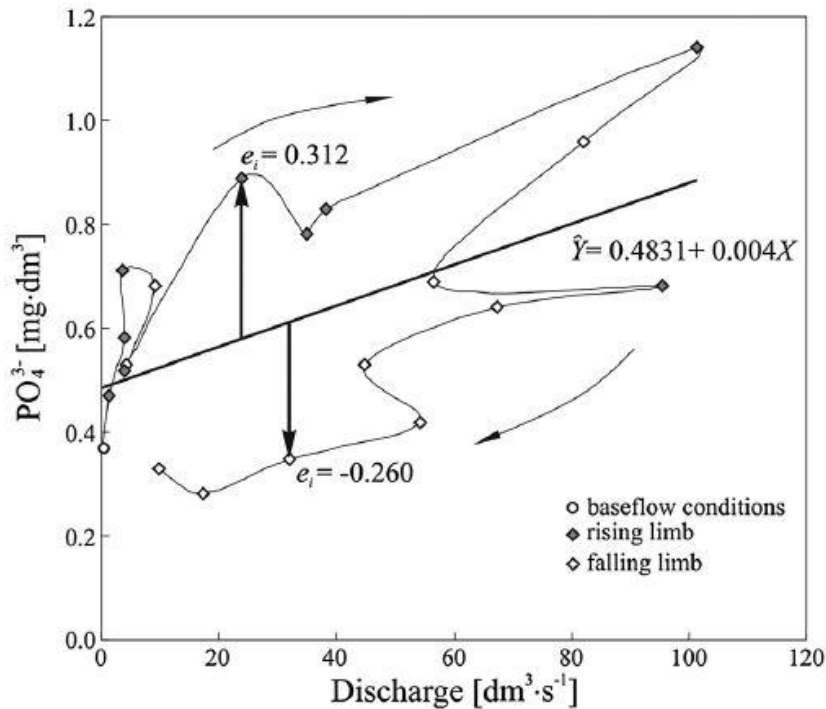
Figure 2.10 Visual representation of the approached used for the calculation of regression residuals for the falling and rising limb of the storm hydrograph (Siwek et al., 2013).

A concentration-discharge plot resulting in a straight line indicates a direct response of concentration to discharge, both peaking at the same time (Figure 2.11 (a)). It does not display hysteretic behaviour and does not give an additional source of information for interpretation. All the other responses are called hysteresis loops and provide an additional source of information about nitrate source and transport. A clockwise loop indicates immediate response to a storm event with concentrations peaking before the peak in discharge (Figure 2.11 (b)); for dilution events, peak discharge precedes minimum concentration. This gives high immediate concentrations with decline as the storm event progresses. An anti-clockwise loop represents an event with concentration peaking after the discharge peak (Figure 2.11 (c)); or for a dilution event, minimum discharge precedes peak discharge. This results in low immediate concentrations with an increase as the storm event progresses. A figure-8 loop represents a complex response of concentrations to discharge (Figure 2.11 (d)). (Williams, 1989).
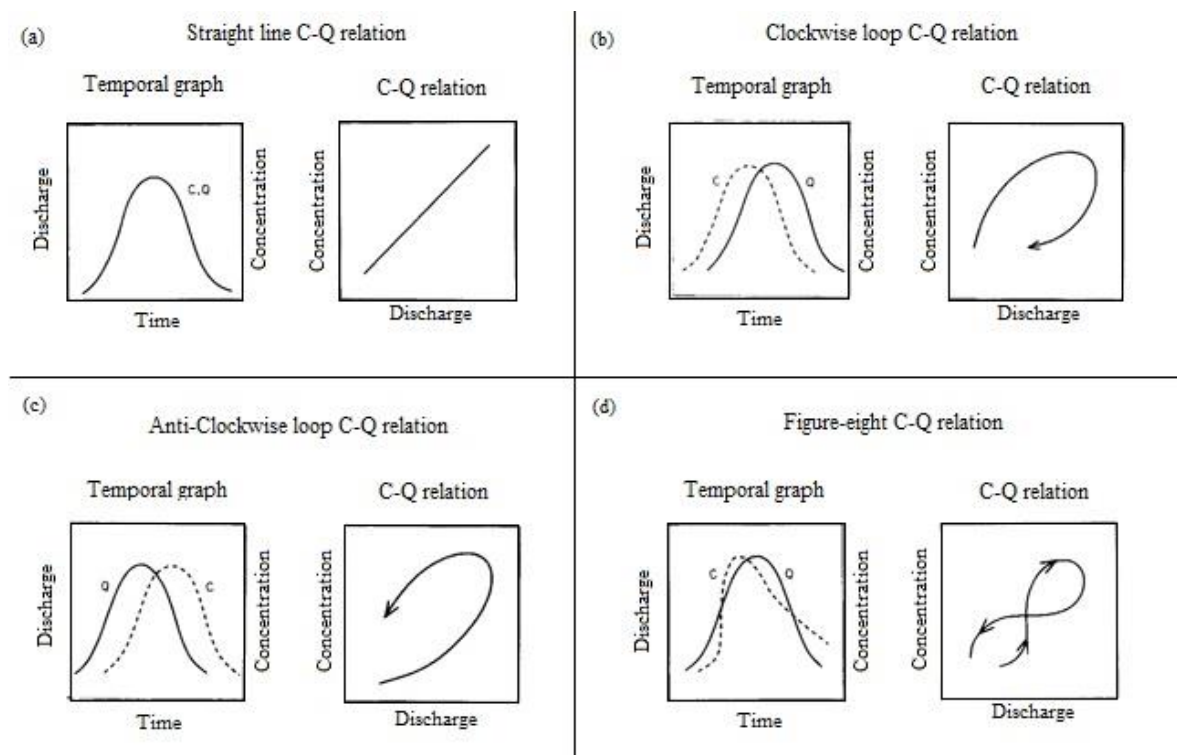
Figure 2.11 Types of hysteresis loop and their temporal graphs (Williams, 1989).

To gain a better understanding of nitrate leaching patterns and the effect of antecedent conditions, a correlation matrix will be used to investigate the antecedent variables that are correlated to each other and to nitrate. Average, maximum and minimum concentrations as well as total nitrate flux can be derived from the raw nitrate data. Minimum and maximum nitrate concentrations will help with understanding about what variables are influencing the extremes of nitrate concentration. Total nitrate flux indicates the total amount of nitrate leaving the catchment in an event. Investigation of the correlation of these derived parameters to other variables will facilitate better

understanding of nitrate leaching patterns. To focus on the most relevant variables, stepwise multiple regression analysis will be used.

2.7 Seasonal Regression

The standard meteorological description of season was used: the seasons were defined as Spring (March, April, May), Summer (June, July, August), Autumn (September, October, November) and Winter (December, January, February) (Met Office 2016). Each storm event was then assigned a season depending on which month it belonged to at the start of the storm event. Storm events were then grouped by season and simple linear regression was used to check whether there is a particular season when any relationship between nitrate and discharge during storm events is more pronounced. The results will be used to show whether the changes in nitrate concentration are related to transport (discharge) more in some seasons than in others and will facilitate interpretation of results from hysteresis analysis and stepwise multiple regression.

2.8 Stepwise Multiple Regression

Stepwise regression is an automated tool used to identify a useful, smaller set of variables when there are large numbers of possible variables available. Stepwise multiple regression either systematically adds the most significant variable or removes the least significant variable during each step (Minitab Inc., 2015). However, the results must be interpreted with caution as stepwise multiple regression can pose some problems as pointed out by Minitab Inc. (2015):

- 'If two predictor variables are highly correlated, only one might end up in the model even though both might be important.

- Because the procedure fits many models, it could be selecting ones that fit the data well because of chance alone.

- Stepwise regression might not always stop with the model with the highest R^2 value possible for a specified number of predictors.

- Automatic procedures cannot consider special knowledge the analyst might have about the data. Therefore, the model selected might not be the best from a practical point of view.'

To counteract some of these problems and facilitate a more balanced interpretation of results, a correlation matrix including all of the variables will be created beforehand. Variables investigated include antecedent variables and variables derived from data spanning the length of each storm event. A full list of these is included in Table 2.1. Each variable entered in the correlation matrix and stepwise multiple regression analysis is per storm event. Dependent variables are: average

nitrate concentration, maximum nitrate concentration, minimum nitrate concentration, total nitrate flux, hysteresis type and magnitude. The rest of are the independent variables.

c	Description	Units
NAverageConc	*Average nitrate concentration*	mg NO$_3$ l^{-1}
TotalNFlux	*Total nitrate flux*	kg
MaxNConc	*Maximum nitrate concentration*	mg NO$_3$ l^{-1}
MinNConc	*Minimum nitrate concentration*	mg NO$_3$ l^{-1}
Hysteresis	*Hysteresis type; 1 for clockwise, -1 for anti-*	-
HMagnitude	*Hysteresis magnitude*	-
TimeSinceLastPeak	*Time since last discharge peak*	days
TotalRainfall	*Total rainfall*	mm
stormAPI	*Storm antecedent precipitation index*	-
TotalQ	*Total discharge*	m^3/s
StartQ	*Start discharge*	m^3/s
PeakQ	*Discharge peak discharge*	m^3/s
Qp-Qo	*Difference between peak and start discharge*	m^3/s
TMedian3	*Medium temperature of 3 days prior to storm*	℃
TMedian5	*Medium temperature of 5 days prior to storm*	℃
TMedian7	*Medium temperature of 7 days prior to storm*	℃
TMedian10	*Medium temperature of 10 days prior to storm*	℃
TMedian15	*Medium temperature of 15 days prior to storm*	℃
seasonID	*Seasonality; 1-spring, 2-summer, 3-autumn, 4-*	-
seasonCos	*Seasonality; +1 in January, -1 in July; represents variation in wetness*	-

Table 2.1 Variables used for stepwise analysis. Each variable is per storm event

A cosine curve was used to quantify seasonality because it mimics the overall seasonal variation in temperature. It peaks at 1 on 1st January and reaches -1 on 1st July. An antecedent precipitation index (API) was calculated ($I=kI_0+rainfall$ (Eq. 2.3)) using methodology employed by Kohler and Linsley (1951). The antecedent precipitation index (*I*) of any time period is equal to that of the previous time period multiplied by recession factor *k*. If rain occurs in that time period, it is added to the index (Equation 2.3). *k=0.85* was used with $I_0 = 1.8$. These numbers were taken from an example in the original paper (Kohler and Linsley 1951). As the value of API levels out within the first few weeks (if not days) of the record and as the purpose was to characterise variability rather than define an absolute value, these values of *k* and I_0 were deemed acceptable.

The results of stepwise multiple regression will be checked for the most significant variables that influence average nitrate concentration, minimum and maximum nitrate concentrations, total nitrate flux, and hysteresis type and magnitude.

2.9 Future Changes in Nitrate Concentrations

To enable consideration of the potential impacts of projected climate change on nitrate concentrations in stream water, the UKCP09 weather generator coupled with a simple catchment model will be used, to calculate the variables needed for the stepwise multiple regression question to predict changes in nitrate concentrations. The UKCP09 Weather Generator creates synthetic time series of weather variables at 5 km resolution, which are consistent with the underlying climate projections (Jones et al, 2009). It is based around a Newman Scott rectangular pulse stochastic rainfall model that simulates future rainfall sequences. Other weather variables are then generated according to the rainfall state. Statistical measures within the Weather Generator are then modified according to the probabilistic projections developed in UKCP09 (Jones et al, 2009). To represent the predictive uncertainty when assessing the potential climate change effects many series of projected weather are produced for the same time period. The uncertainty arises from natural climate variability, both internal and external, incomplete understanding of Earth System processes, equifinality and imperfect representation in climate models (termed model uncertainty) and uncertainty in future emissions (Murphy et al, 2009).

The HBV-light catchment model (Seibert, J. and Vis, M . 2012) will be used to predict future discharges using the temperature and precipitation data provided by the UKCP09 Weather generator. HBV-Light is a lumped parameter model and hence represents the catchment as a single spatial unit. The model includes hydrological process representation for snow melt, water percolation between different vertical hydrological stores and the lateral movement of water to the river along both soil and groundwater pathways. This model was selected due to its proven ability to simulate river catchments (Driessen et al. 2010) and the fit between the observed hydrological flow pathways and dynamics at the Newby Beck catchment and the process representation within the model. The model structure is shown in Figure 2.12.
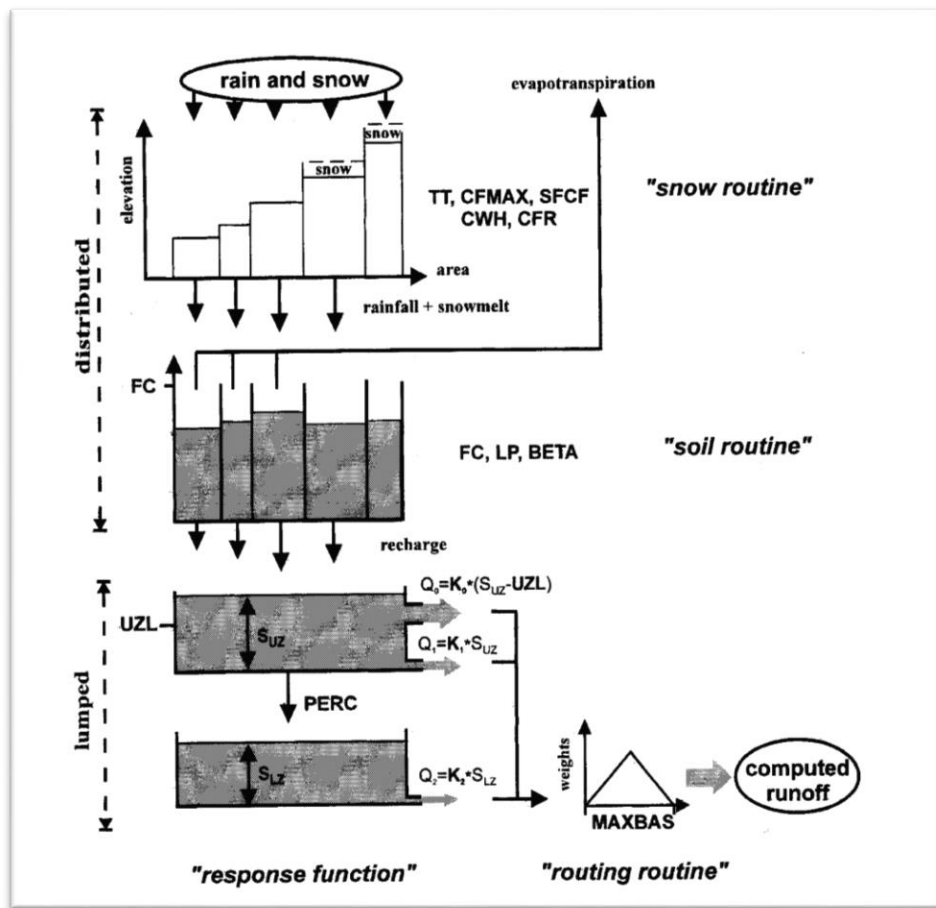
Figure 2.12 Structure of the HBV-Light hydrological simulation model Seibert (2000)

Modelling catchment discharge presents further uncertainties and thus a single simulation would be insufficient. The Generalised Likelihood Uncertainty Estimation (GLUE) method (Beven and Bingley 1992) will be used to generate a large number of behavioural model parameter sets with top 30 chosen for simulations based on the model performance compared to the observed dataset. This incorporation of the model predictive uncertainty will multiply the series for predicted discharge by 30 and hence when combined with the 100 different stochastically generated time series from the UKCP09 weather generator for temperature and rainfall results in 3000 modelled series of discharge.

The predicted nitrate concentrations per storm event will be calculated for each of these simulated weather and discharge series. The nitrate concentrations will be calculated using the stepwise multiple regression model results by choosing the model with highest level of explanation provided. A set of antecedent variables per storm event will be calculated alongside. Then the regression formula combined with the coefficients of the chosen model will be used to calculate

nitrate per storm event. Then a sum of nitrate concentrations for that series will be calculated. The storm events within the projected future simulations will be identified as described in the above methodology.

Nitrate concentrations can then be compared between different modelled scenarios. To enable these comparisons the change factor will be calculated by finding the change in nitrate concentrations between the baseline and modelled future scenarios (Eq 2.4). Plotting the change factors as a histogram will indicate the probability of change taking place.

CF = (scenario-baseline)/baseline   Eq. 2.4

CF stands for change factor, *scenario* stands for sum of nitrate concentrations for a series calculated for future scenario, and *baseline* for sum of nitrate concentrations for a series calculated for baseline weather.

2.10 Automated Analysis

The analysis of high-frequency data spanning over the period of 3 years was done by the use of Python programming language. The exact code written can be seen in Appendix 1. The use of Python enabled the consistent and efficient classification of large amounts of storm events using the methodology as described in section 2.4. The entire dataset was analysed for presence of storm events. The code was written to analyse the nitrate data associated with storm events. For every storm event identified, the storm nitrate concentration and discharge relationship was investigated. The code was written to identify hysteresis type and magnitude using methodology described in section 2.6. Simple regression analysis of storm event nitrate concentration and discharge by season was carried out (section 2.7). The antecedent data associated with each storm event was later used in stepwise multiple regression analysis (section 2.8). Future changes in nitrate concentrations were calculated as outlined in section 2.9 and was only possible because of the use of automated analysis.

Chapter 3: Results and Discussion

3.1 Introduction

This chapter presents the results of the methodology carried and discusses the findings of this research project. Results and discussion for storm events are presented in section 3.2, hysteresis in section 3.3, simple regression in section 3.4, and stepwise multiple regression in section 3.5. Discussion of influence of future climate change and advise on mitigation measures are presented in sections 3.6 and 3.7.

3.2 Storm Events

The catchment hydrograph displays flashy behaviour with storm events being well pronounced and easily identified as is consistent with under-drained clay soil hydrology (Granger et al. 2010). The low flows in between storm events are consistent with the catchment hydrology being governed by superficial deposits and lack of influence of large aquifers. A total of 248 storm events have been identified. Some storm events were excluded due to missing data that could not be successfully interpolated. The excluded storm events are as follows: event number 108 (missing temperature and rainfall data); event numbers 126, 127 and 189 (missing nitrate data). The discharge record had no missing data. This gives a total of 4 storm events that were excluded due to missing data. After the storm events were plotted, the hydrograph was visually inspected to detect any storm events that should not have been identified as such due to the nature of automatized separation of base flow and the identification of storm events. A number of storm events were identified that should be excluded due to either not being a complete storm event or the peak discharge rise above start discharge was insufficient. Nine storm events have been identified as unsuitable (event numbers 25, 130, 131, 137, 140, 141, 142, 238 and 240) leaving 235 storm events. Figure 3.1 displays a hydrograph with the storm events identified and excluded storm events highlighted.
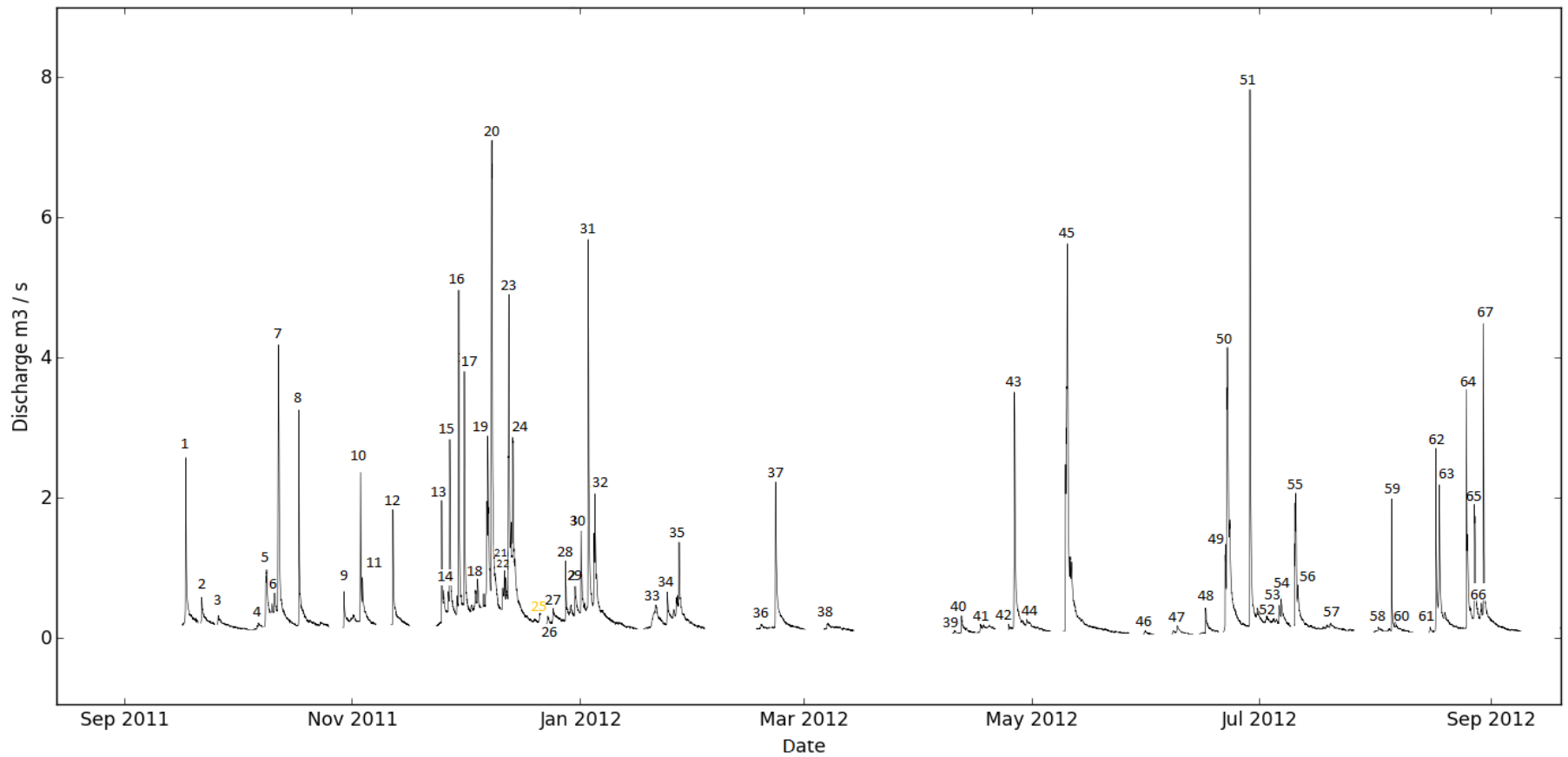
Figure 3.1 (a) Storm events September 2011 to September 2012. Excluded storm events highlighted in red and yellow. Events in red were excluded due to missing data. Events in yellow were excluded after visual inspection.
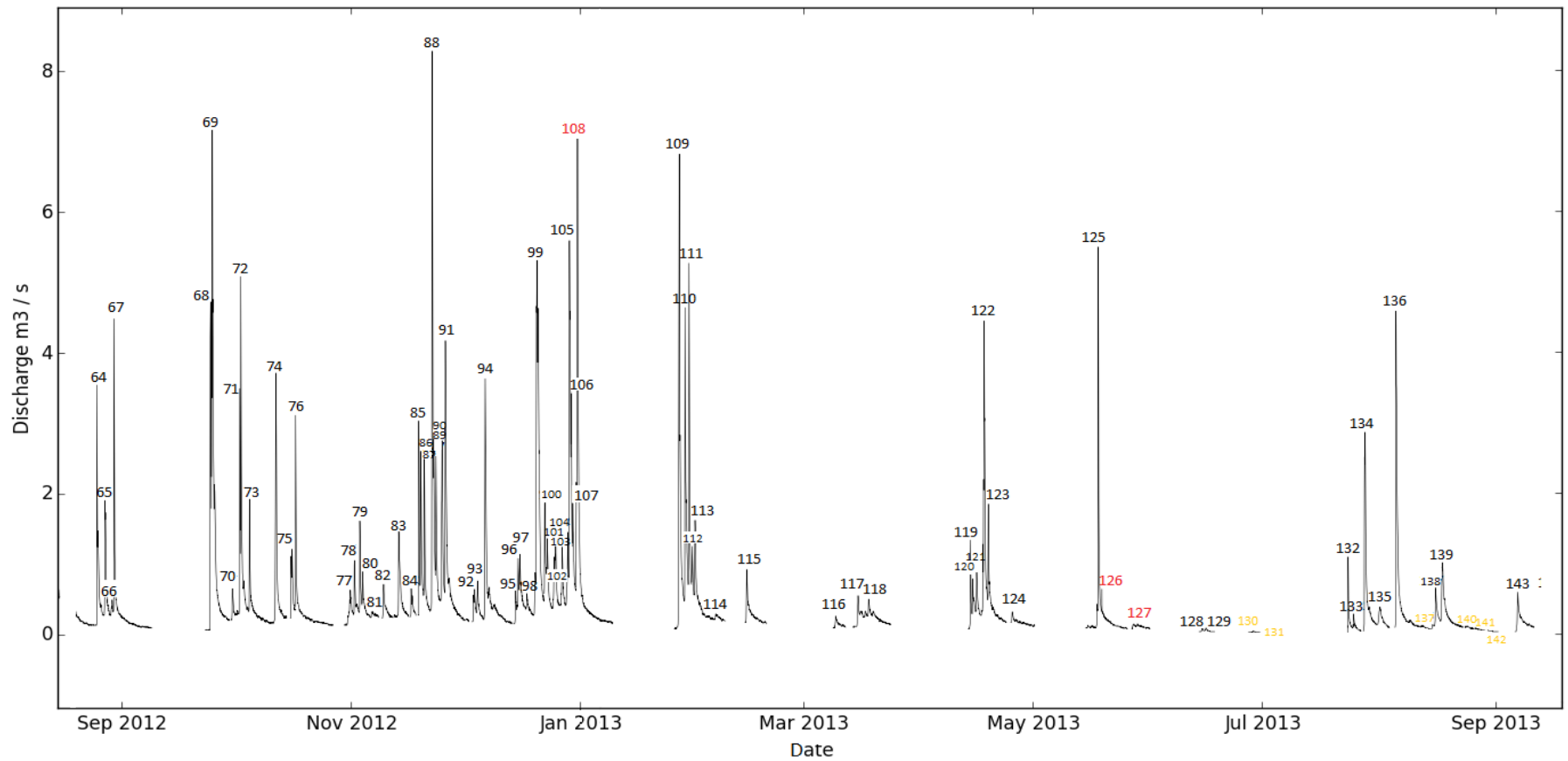
Figure 3.1 (b) Storm events September 2012 to September 2013. Excluded storm events highlighted in red and yellow. Events in red were excluded due to missing data. Events in yellow were excluded after visual inspection.

46
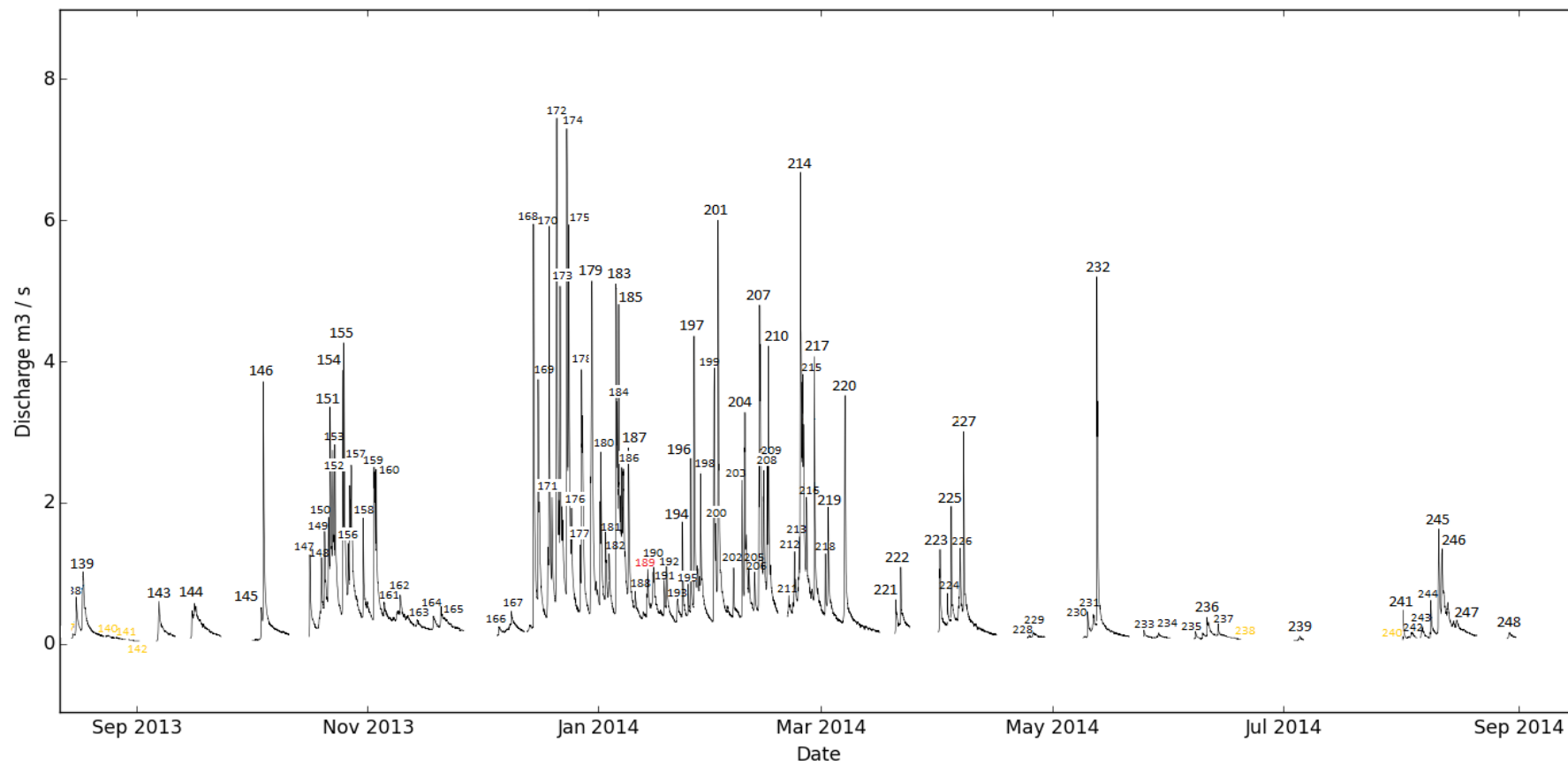
Figure 3.1 (c) Storm events September 2013 to September 2014. Excluded storm events highlighted in red and yellow. Events in red were excluded due to missing data. Events in yellow were excluded after visual inspection.

When the data were reviewed, after the identification of storm events, it emerged that on various occasions high nitrate levels were observed during non-storm periods. The non-storm high-nitrate leaching periods might indicate a large and sustained addition of nutrients from the soil to the stream, even after a storm event has ended. These nutrients continue to be leached without any significant increase in the discharge, probably indicating wet ground conditions and sustained drainage to the stream. It thus indicates that the nitrate is leached into shallow groundwater that forms the base flow of stream discharge during these non-storm periods. The leaching events could also be initiated by application of high amounts of liquid fertiliser such as slurry or rainfall events that do not produce much outflow from the clay soil (Anderson and Burt, 1990) thus only acting to increase soil moisture and facilitate nitrate release from the soil. This ability of the catchment to leach high amounts of nitrate, without storm event taking place, calls for further investigation into non-storm events to obtain a full image of catchment hydrology and nitrate leaching patterns. It will be important to determine whether such occurrences happen because of changes in supply (source-related) such as fertiliser addition, or whether they are predominantly transport-related. These 'non-events' were excluded from subsequent analysis.

3.3 Hysteresis

235 storm events were assessed for hysteresis type using the automated analysis approach. Figure 3.2. displays the storm events marked by the type of the storm event. Red storm events are for anti-clockwise hysteresis and blue events for clockwise hysteresis. The discharge-concentration plots for each of these storm events can be found in the Appendix 4.

A variety of hysteresis magnitudes both of clockwise and anti-clockwise hysteresis were visually assessed. It has been concluded that events of magnitude smaller than 0.27 should be considered either to be too close to 0 and hence display no hysteresis or be too complex in nature thus resulting in overall very small magnitudes. Upon this review of hysteresis plots, 45 further storm events (storm events number: 2, 3, 4, 6, 8, 9, 11, 18, 20, 24, 34, 36, 39, 42, 43, 44, 46, 53, 54, 69, 78, 83, 86, 97, 105, 109, 117, 118, 128, 129, 138, 154, 165, 180, 190, 198, 207, 210, 212, 214, 220, 222, 229, 242, 248) were considered unsuitable for further analysis of hysteresis. This removal of events left 190 storm events to be analysed. The events excluded during this step were analysed separately. After reconsideration of storm events, the remainder of storm events contained 67 (35% of total) anticlockwise hysteresis loops and 123 (65% of total) clockwise loops.
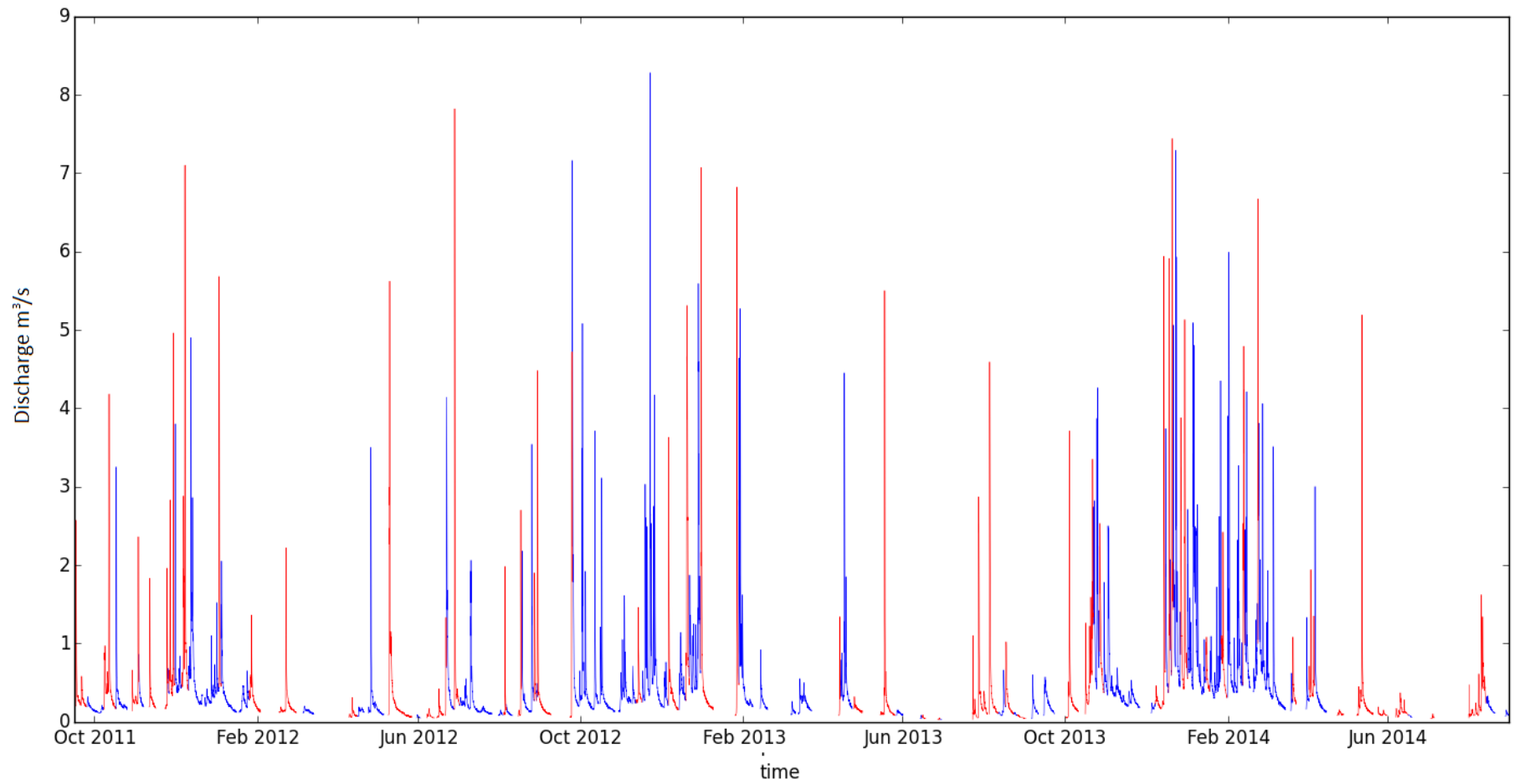
Figure 3.2 Storm events classified by hysteresis type. Blue events are clockwise and red events are anticlockwise.

3.3.1 Hysteresis Distribution by Season

There is a clear seasonal distribution of clockwise to anti-clockwise hysteresis storm events as can be seen in Figure 3.2 above (see Table 3.1 for hysteresis distribution by season). Summer has more anti-clockwise storm events than any other season. Over half (69%) of all summer storm events were anticlockwise. Autumn is dominated by clockwise storm events with 64% of total number of storm events. Winter has the largest total number of storm events of all seasons and the highest number of clockwise storm events. Clockwise storm events account for 86% of total winter storm events. The number of anticlockwise storm events increase again in spring with 52 % of total spring storm events being anticlockwise.

Season	Anticlockwise	Clockwise hysteresis	Total (100%)
Autumn	18 (36%)	32 (64%)	50
Winter	11 (14%)	67 (86%)	78
Spring	14 (52%)	13 (48%)	27
Summer	24 (69%)	11(31%)	35
Total:	67 (35%)	123 (65%)	190

Table 3.1 Hysteresis type distribution by season. Numbers in brackets are percentages of total number of events for that season.

3.3.2 Hysteresis Loop Magnitudes

Hysteresis loop magnitude ranges from 0.28 (event No. 23) to 12.84 (event No. 119). There was a visible difference in magnitude ranges displayed by clockwise and anticlockwise hysteresis loops. Anticlockwise hysteresis ranged from 0.323 (event No. 31) to 12.843 (event No. 119) while clockwise hysteresis had magnitudes that were much smaller, ranging from 0.277 (event No. 23) to 3.250 (event No. 50).

3.3.3 Clockwise Hysteresis Storm Events

During the clockwise hysteresis storm events, the Newby Beck stream water nitrate concentrations respond to storm events with high concentrations to start with that decline as the storm progresses (Figure 3.3). The dominant hysteresis type of this catchment is clockwise at 65% of total. Clockwise events start with high amounts of nitrate being leached from the catchment; concentrations then decline either because the stream it is diluted by excess rainfall and runoff /drain-flow or because the nitrate source is depleted, or both. Bowes et al (2015) state that clockwise hysteresis indicates a quick delivery of nitrate to river in response to rainfall with potential sources being mobilisation along river margin and delivery through drains. The catchment under investigation is small and well connected through the well-developed structure of clay soil and the presence of under-drainage. Nitrate availability and connectivity of the catchment is what produces a clockwise storm event within this catchment. As a storm event starts, old nutrient-rich

soil water is readily available for transport and is immediately moved out of the soil. This diffuse source is rapidly depleted as, due to the large amounts of storm events during wet seasons 'flushing the system' (House and Warwick, 1998) and lower temperatures, no large amounts of nitrate can accumulate in the soil. Nitrate concentrations in stream water of this type of event would therefore be source-limited (sometimes called 'supply-limited'). Further rainfall will act to dilute stream water. As the storm proceeds, it returns to just slightly lower nitrate concentrations as compared to concentrations at the start of event. The slight fall in concentration post-storm could be the effect of exhaustion of diffuse sources of nitrate (Webb and Walling, 1985, House and Warwick, 1998). So, the overall effect of clockwise hysteresis storm event is the dilution of stream water. Smaller magnitudes of clockwise hysteresis storm events (section 3.3.2) show the disruption to pre-storm nitrate concentrations to be of smaller magnitude during these storm events as the nitrate concentrations neither fall nor rise sufficiently to separate the falling and rising limb of hysteresis loop to the extent of anti-clockwise storm events.
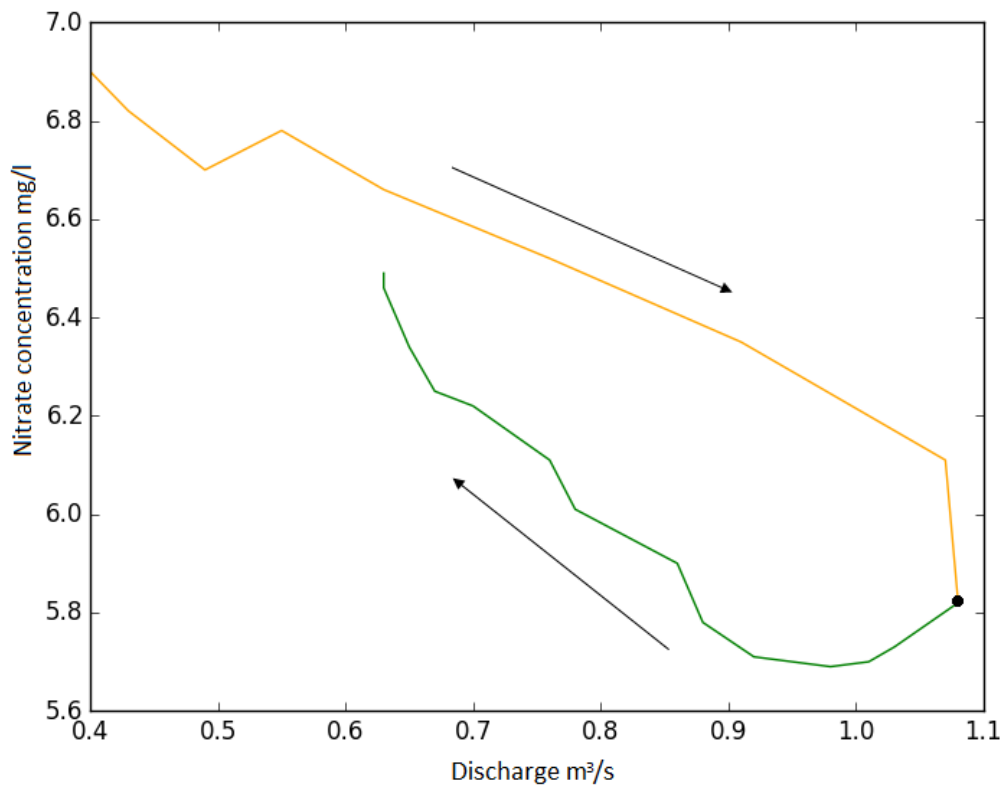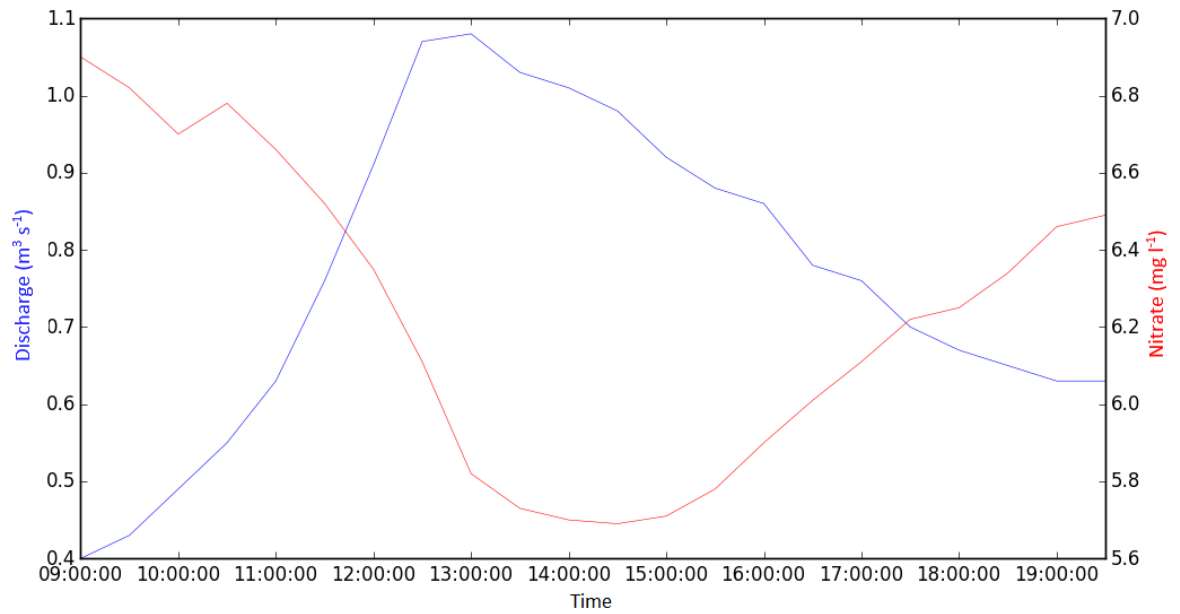
Figure 3.3 Storm event 96 with clockwise hysteresis loop. Storm hydrograph (top) in blue with nitrate concentration superimposed in red. Concentration-discharge plot (bottom) of the same storm event, orange represents a rising limb, green represents a falling limb and the black circle marks the peak of the storm event.

3.3.4 Anti-clockwise Hysteresis Storm Events

Within this particular catchment, anticlockwise hysteresis accounts for 35% of all the events. Although the catchment is not dominated by this type of event, they still comprise a large proportion of the total. Bowes et al. (2015) and Lloyd et al. (2016) found the proportion of anti-clockwise storm events smaller in their catchments. The progress of the anticlockwise storm event acts to increase nitrate concentrations in stream water. A storm event starts with low concentrations which increase within the second half of the storm event (Figure 3.4). The larger magnitudes of these storm events (section 3.3.2) would indicate large increases in the amounts of nitrate released as compared to pre-storm concentrations. These events are more common during summer and spring (Table 3.1 and Figure 3.2), and after prolonged dry periods (Figure 2.3 and 2.4), as this increase the time nitrate can accumulate in the soil. This type of storm event happening in winter could indicate large release of nutrients from farms during events such as poorly timed slurry spreading or simply the high concentrations associated with exceptionally high shallow groundwater flow (cf. Burt and Arkell, 1987). The anti-clockwise events are therefore of particular interest as they can drive high nitrate concentrations in stream water for a small amount of time.

Bowes et al. (2015) state that anti-clockwise hysteresis indicates slowly mobilised catchment-wide diffuse sources. However, the size of the Newby Beck catchment would point to the distance of sources and slow response being less relevant here. This leaves the sources that are slowly mobilised. When clay soil dries out, it accumulates nutrients due to reduced soil moisture and lack of transport. Clay soil that has dried out would only release the nutrients from its soil structure after thorough re-wetting (Figure 1.8 (b)). The re-wetting of the soil can take some time (Granger et al. 2010) and macropore flow will occur only after certain thresholds of rainfall intensity and soil moisture content are passed (Anderson and Burt, 1990). For nutrients to be released after soil is finally re-wetted, the new rainfall water needs to push the 'old' water out (Figure 1.8 (a)) which can happen very quickly (Anderson and Burt, 1982) and can generate large increases in nitrate concentrations (White et al. 1983) for anti-clockwise storm events. The large increases in nitrate concentration during anticlockwise storm events are responsible for the large magnitudes of these storm events (section 3.3.2). Due to the amount of time nitrate has had to accumulate in the soil and higher temperatures prevalent in summer and spring, the amounts of nitrate available for transport can be high in these seasons, with anti-clockwise storm events accounting for more than half of total events of those seasons (Table 3.1). Total number of anti-clockwise storm events is also high in autumn when warm soils are wetting up. Nitrate leaching of this type of storm event is transport-limited with plenty of nitrate available for its leaching to depend on flow.
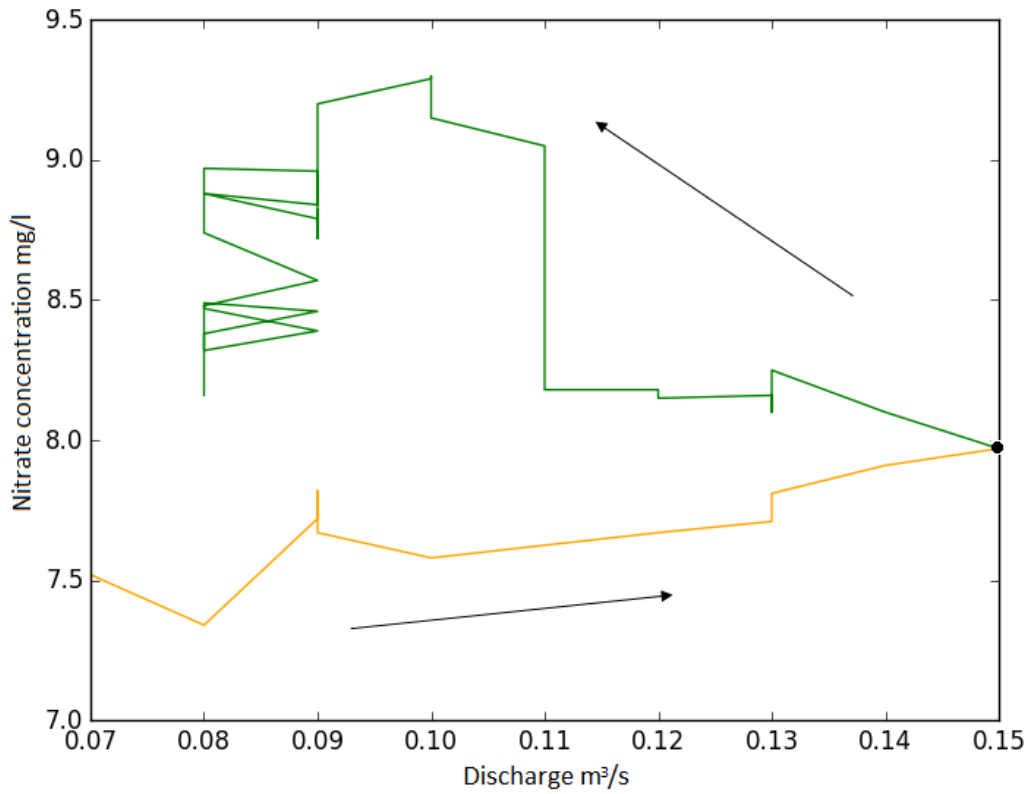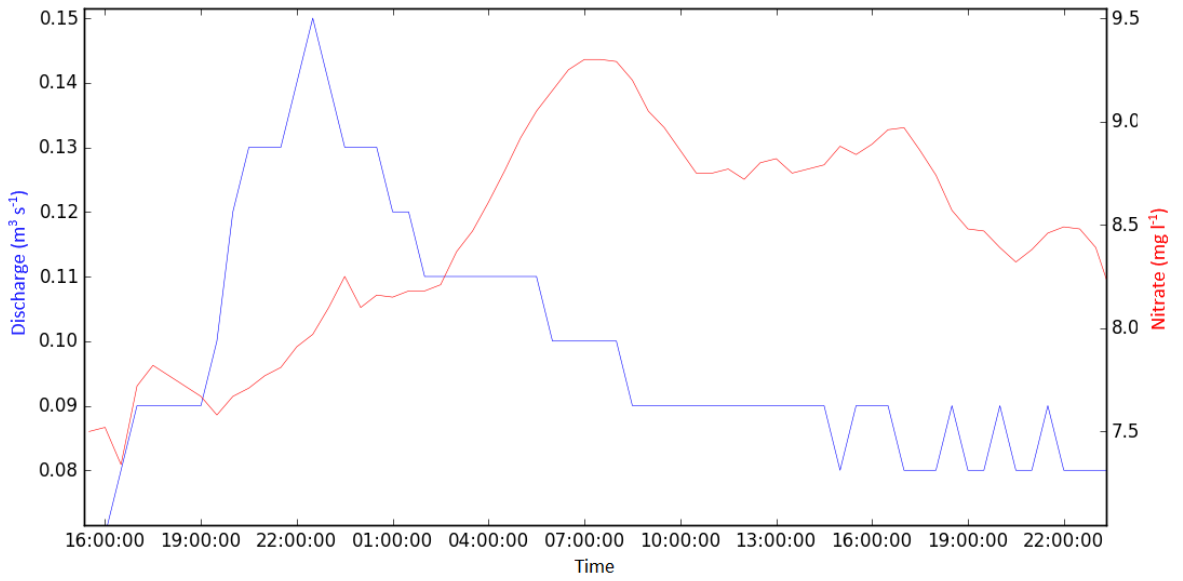
Figure 3.4 Storm event 61 with an anti-clockwise hysteresis loop. Storm hydrograph (top) in blue with nitrate concentration superimposed in red. Concentration-discharge plot (bottom) of the same storm event, orange represents a rising limb, green represents a falling limb and the black circle marks the peak of the storm event.

3.4 Simple Linear Regression

When a simple linear regression analysis for storm event discharge and nitrate concentration as grouped by season has been carried out, it has shown a varying degree of explanation of variation in nitrate concentration that discharge can provide. Discharge explains variation in nitrate concentration at 2.13% in autumn, 7.66% in winter 1.07% in spring and 0.01% in summer. Figure 3.5 shows a distribution of storm events by the size of the total storm discharge and average nitrate concentration indicated by season. Autumn and winter storm events have the highest flows with lower nitrate concentrations. Spring and summer events have highest nitrate concentrations with lower flows. Such seasonal variability in response of nitrate concentrations to levels of discharge could indicate it being dictated by something more than just discharge. Other variables, such as antecedent conditions should be investigated in future work.
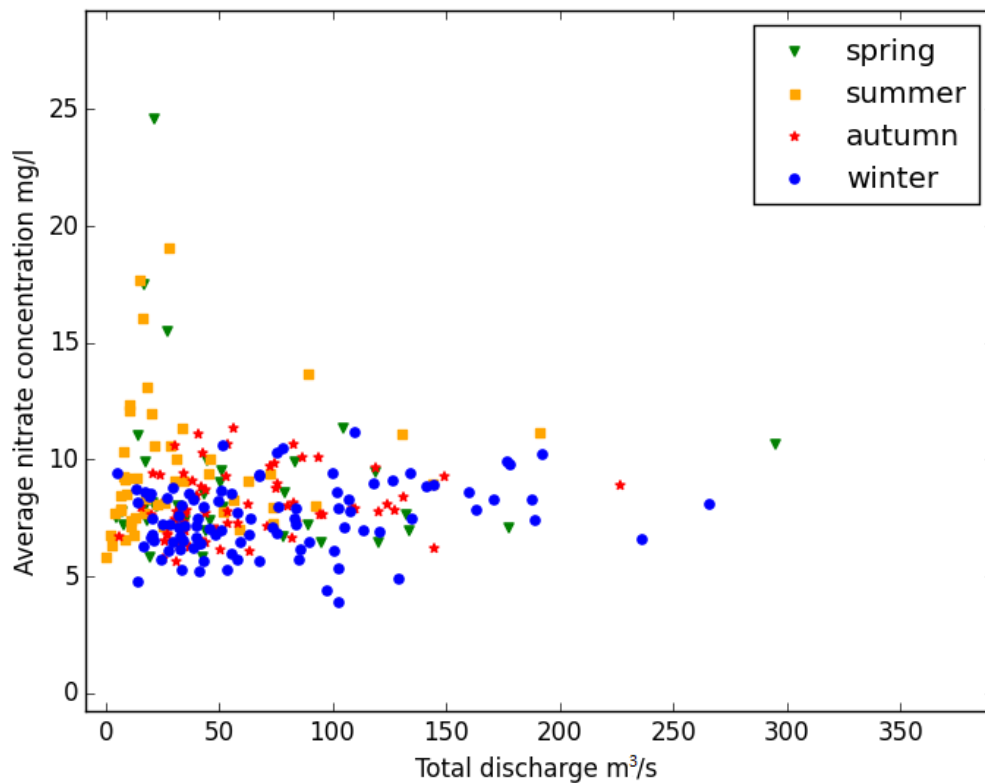
Figure 3.5 Total storm discharge and average storm nitrate concentration by season.

Autumn and winter storm events show some large storm events with majority of nitrate concentrations falling between 5 and 11 mg NO_3 l^{-1} (Figure 3.5). This result is consistent with smaller hysteresis magnitudes exhibited by clockwise storm events that dominate autumn and winter seasons (Table 3.1). Summer storm events show lower total discharges with higher nitrate concentrations. As discussed above in section 3.3.4, the nature of anti-clockwise storm events, that

dominate this season, is that it releases large amounts of nitrate thus accounting for its larger hysteresis magnitudes. Spring can be seen to fall in between these two distinct seasonal patterns.

3.5 Stepwise Multiple Regression

3.5.1 Pearson's Correlation Analysis of Storm Event Data

Prior to the stepwise multiple regression analysis (Full output of SPSS Stepwise multiple regression analysis can be found in Appendix 3.), a Pearson's correlation matrix was produced (Table 3.2) to show how each variable is correlated to the rest. The data for these variables, as calculated per storm event, can be found in Appendix 2. The most significant correlations are:

- Average nitrate concentration is correlated to time since last peak, total rainfall, and inversely to storm API (all at $P < 0.05$). It is highly correlated to seasonID and inversely to start discharge (both at $P < 0.01$), all the median temperatures prior to storm event and inversely to cosine curve (all at $P < 0.001$).

- Total nitrate flux is highly correlated to stormAPI (at $P < 0.01$), total rainfall, total discharge, peak discharge and [Qp-Qo] (all at $P < 0.001$).

- Maximum nitrate concentration is correlated to seasonID (at $P<0.05$). It is highly correlated to stormAPI (at $P = 0.01$), time since last peak, total rainfall, start discharge, all the median temperatures and cosine curve (all at $P < 0.001$).

- Minimum nitrate concentration is highly correlated to stormAPI, total discharge, start discharge, peak discharge, Qp-Qo, all the median temperatures, seasonID and cosine curve (all at $P < 0.001$).

- Hysteresis type is correlated to the hydrograph rise, Qp-Qo (at $P<0.05$). It is highly correlated to time since last peak, total rainfall, stormAPI, start discharge, all the median temperatures, seasonID and cosine curve (all at $P < 0.001$).

- Hysteresis magnitude is correlated to total discharge and TMedian15 (both at $P=0.05$). It is highly correlated to total rainfall (at $P < 0.01$), time since last peak, stormAPI, start discharge, the rest of median temperatures and cosine curve (all at $P < 0.001$).

	NAverageConc	TotalNFlux	MaxNConc	MinNConc	Hysteresis	HMagnitude
TimeSinceLastPeak	0.189	-0.069	0.292	0.133	-0.242	0.435
TotalRainfall	0.188	0.672	0.342	-0.059	-0.385	0.194
stormAPI	-0.182	0.215	-0.239	-0.299	0.305	-0.267
TotalQ	-0.046	0.879	-0.038	-0.260	-0.086	-0.171
StartQ	-0.215	0.049	-0.289	-0.240	0.339	-0.237
PeakQ	-0.106	0.574	-0.064	-0.351	-0.070	-0.092
QpMinusQo	-0.052	0.597	0.014	-0.305	-0.171	-0.031
TMedian3	0.418	0.083	0.396	0.484	-0.438	0.203
TMedian5	0.380	0.081	0.365	0.450	-0.447	0.201
TMedian7	0.360	0.079	0.348	0.420	-0.451	0.198
TMedian10	0.344	0.083	0.347	0.388	-0.462	0.203
TMedian15	0.321	0.075	0.330	0.362	-0.456	0.189
SeasonID	0.209	0.023	0.160	0.388	-0.255	0.042
seasonCos	-0.448	-0.090	-0.556	-0.447	0.469	-0.377

	P = 0.1	P = 0.05	P = 0.01	P = 0.001
190.000	0.121	0.144	0.189	0.239

Table 3.2 Pearson's correlation matrix for antecedent variables derived for each storm event

The apparent lack of correlation with discharge and variables derived from discharge data (with exception of the inverse correlation to start discharge) supports the small degree of explanation the variation in discharge provides for variation in nitrate concentration (section 3.4). Seasonality and especially temperature prior to storm event are positively correlated to nitrate concentrations. Median temperature three days before the event exerts most influence on nitrate concentrations. The inverse correlation of the antecedent precipitation (or soil wetness) and positive correlation of time since last peak to nitrate concentration point to nitrate levels in stream water influenced by the amount of time nitrate had to accumulate in the soil. In contrast to the positive correlation of the average, maximum and minimum nitrate concentrations to the total event rainfall, they are either not correlated or correlated negatively to discharge variables. The influence of rainfall could be due to the nitrate being made available through soil re-wetting and transportation of readily available 'old' soil water.

Maximum nitrate concentrations are correlated to the same variables as average nitrate concentrations but have stronger correlation with time since last peak, total rainfall, antecedent precipitation and start discharge. The inverse correlation of minimum nitrate concentrations to discharge could be due to dilution – the higher the discharge the smaller the minimum nitrate concentration. The only variable the minimum nitrate concentration is not correlated to is total rainfall. The nitrate flux being highly correlated to discharge is due to discharge being directly involved in the calculation of nitrate flux and should be interpreted with care.

Temperatures, seasonID, times since last peak, total rainfall, how fast the storm event peaked (Q0-Qp) all correlated to hysteresis type inversely. Therefore, the increase in the antecedent variable would be followed by a decrease in the hysteresis type, which in turn would favour anti-clockwise hysteresis (value of -1). Hence, decrease in these variables would favour clockwise hysteresis (value of 1). Antecedent precipitation, start discharge and seasonCos all had a positive correlation and thus the increase in those variables would result in an increase in the hysteresis type, favouring clockwise hysteresis.

Storm API, total discharge, start discharge and seasonCos all inversely correlated to hysteresis magnitude. Increase in these variables will result in decrease of hysteresis magnitude. Temperatures, time since last peak and total rainfall are directly correlated to hysteresis magnitude. Increase in these variables would result increase in hysteresis magnitude These are part of same variables favouring the anticlockwise hysteresis type that has larger magnitudes as mentioned above.

3.5.2 Average Nitrate Concentration per Storm Event

Median temperature 3 days before the storm event and time since last peak were the only variables selected by stepwise multiple regression analysis to create an average nitrate model (Table 3.3). This model explains 19.9% of total variance in average nitrate concentrations per storm event. Median temperature 3 days before the storm event accounts for 88% of the total explanation this model provides. When seasonality was added to the variable list as a cosine curve, it was the only variable selected by stepwise multiple regression. It created a model that explains 20.2% of variance in average nitrate concentration per storm event. Median temperature 3 days before the storm event, time since last peak and SeasonCos are all source related variables.

Model Summary		
Model	**Variables Entered**	**R Square**
1	TMedian3	0.175
2	TimeSinceLastPeak	0.199
Model Summary(SeasonCos included)		
Model	**Variables Entered**	**R Square**
1	SeasonCos	0.202

Table 3.3 Model summary for average nitrate concentration

3.5.3 Maximum Nitrate Concentration per Storm Event

Maximum nitrate concentration model (Table 3.4) comprises temperature 3 days prior to the storm event, time since last peak, total rainfall and total discharge. Temperature 3 days prior to storm event and time since last peak are source-related variables. Total rainfall and total discharge are transport-related variables. This model explains 28.1% of variance in the dependent variable; temperature 3 days prior to storm event account for 55% of total variation explained by the model. When seasonality was added to the variable list as a cosine curve, it has altered the list of variables selected. SeasonCos topped the model list followed by total rainfall, temperature 15 days prior to storm event and total discharge. SeasonCos and temperature 15 days prior to storm event are source related variables. Total rainfall and total discharge are transport-related variables. It created a model that explains 40.2% of variance in maximum nitrate concentration per storm event, with SeasonCos accounting for 77% of the total explanation provided by the model.

Model Summary		
Model	**Variables Entered**	**R Square**
1	TMedian3	0.156
2	TimeSinceLastPeak	0.225
3	TotalRainfall	0.263
4	TotalQ	0.281

Model Summary(SeasonCos included)		
Model	**Variables Entered**	**R Square**
1	SeasonCos	0.310
2	TotalRainfall	0.349
3	TMedian15	0.385
4	TotalQ	0.402

Table 3.4 Model summary for maximum nitrate concentration

3.5.4 Minimum Nitrate Concentration per Storm Event

Temperature 3 days prior to storm event, peak discharge, median temperature 15 days prior to storm event and seasonID were the variables making up model for minimum nitrate concentration (Table 3.5). Temperature 3 days prior to storm event, median temperature 15 days prior to storm event and seasonID are source related variables. Peak discharge is a transport-related variable. It explains 37% of total variation in minimum nitrate concentration, with median temperature 3 days prior to storm event accounting for 63% of the total explanation provided by the model. Adding cosine curve to variable list did not alter the model in any way.

Model Summary		
Model	**Variables Entered**	**R Square**
1	TMedian3	0.234
2	PeakQ	0.326
3	TMedian15	0.341
4	SeasonID	0.370

Table 3.5 Model summary for minimum nitrate concentration

3.5.5 Total Nitrate Flux per Storm Event

The stepwise multiple regression model for total nitrate flux (Table 3.6) explains more total variability than models for any other dependent variable. Total rainfall, time since last peak and hydrograph rise (Qp-Qo) were the variables chosen by stepwise multiple regression for this model. Total rainfall is a transport related variable. Time since last peak and Qp-Qo are source related variables. It explains 53.5% of the variance in total nitrate flux. Total rainfall account for 84% of total explanation provided by this model. Adding a cosine curve to variable list did not alter the model in any way.

Model Summary		
Model	**Variables Entered**	**R Square**
1	TotalRainfall	0.452
2	TimeSinceLastPeak	0.518
3	Qp-Qo	0.535

Table 3.6 Model summary for total nitrate flux

3.5.6 Hysteresis

Temperature 10 days prior to the storm event, total event rainfall, storm API and Qp-Qo make up the model for hysteresis loop type (Table 3.7). Temperature 10 days prior to the storm event, stormAPI and Qp-Qo are source related variables. Total event rainfall is a transport-related variable. The model explains 35.9% of total variation in hysteresis type, with Temperature 10 days prior to the storm event accounting for 60% of the explained variance. When a cosine curve is included in the variable list, it goes to the top of the selected variable list of the model, with the rest of the list being as follows: total rainfall, storm API and median temperature 15 days prior to storm event after which the SeasonCos was removed and Qp-Qo was added. Total rainfall is a transport related variable. Storm API, median temperature 15 days prior to storm event and Qp-Qo are source related variables. The model explains 36.1% of total variability in hysteresis type.

Model Summary		
Model	**Variables Entered**	**R Square**
1	TMedian10	0.214
2	TotalRainfall	0.283
3	StormAPI	0.345
4	Qp-Qo	0.359

Model Summary(SeasonCos included)		
Model	**Variables Entered**	**R Square**
1	SeasonCos	0.218
2	TotalRainfall	0.286
3	StormAPI	0.320
4	TMedian15	0.348
5	SeasonCos(Removed)	0.346
6	Qp-Qo	0.361

Table 3.7 Model summary for hysteresis type

3.5.7 Hysteresis Magnitude

Time since last peak, median temperature 3 days prior to the storm event and seasonID were the variables included in the hysteresis magnitude model (Table 3.8). These variables are all source-related. The model explained 23.6% of total variance in hysteresis magnitude. Time since last peak accounts for 80% of explanation this model provides. When cosine curve is added to the original list of variables, the model is as follows: time since last peak, cosine curve and seasonID. These variables are all source related. This model explains 28.1% of total variance in hysteresis magnitude. Time since last peak explains 67% of total explanation provided by this model.

Model Summary		
Model	**Variables Entered**	**R Square**
1	TimeSinceLastPeak	0.188
2	TMedian3	0.217
3	SeasonID	0.236

Model Summary(SeasonCos included)		
Model	**Variables Entered**	**R Square**
1	TimeSinceLastPeak	0.188
2	SeasonCos	0.257
3	SeasonID (or TMedian15)	0.281(or 0.274)

Table 3.8 Model summary for hysteresis magnitude

3.5.8 Variables Governing Stream Water Nitrate Concentration

Stepwise multiple regression has revealed the most important variables governing average nitrate concentrations being source related. Temperature (combined with soil moisture) helps release nitrate from soil organic matter (Burt et al. 1993, p. 40) while time since last peak allows time for the nitrate to accumulate in the soil. Maximum nitrate concentration is governed by the same source-related variables (temperature and time since last peak) plus a slight influence by transport related variables (rainfall and discharge). Anticlockwise storm events, that produce the majority of the highest nitrate concentrations during storm events, are mostly observed when temperatures are higher and after prolonged droughts and are transport limited. Minimum nitrate concentrations are also influenced by a mix of source (temperature and season ID) and transport (peak discharge) related variables. Minimum nitrate concentrations will be observed during the events with either low nitrate supply or dilution or a combination of both. It is most likely to be the result of clockwise storm events when supply is limited and dilution occurs. Total nitrate flux is influenced by transport-related variables (total rainfall and the difference between peak and start discharge) with some influence from a source-related variable (time since last peak). Total rainfall will exert large influence on discharge, which is one of the components of nitrate flux calculation, as the catchment hydrology is dominated by superficial deposits (Allen et al. 2010).

Hysteresis is predominantly governed by source-related antecedent conditions (temperature and antecedent precipitation index) with some influence from transport-related variable (total rainfall). Hysteresis magnitude is entirely dictated by source-related variables (time since last peak,

temperature and season ID). The high importance of preparatory processes is clear and is consistent with the findings of Perks et al (2015) who looked into suspended sediment and phosphorus transport in the same catchment. The importance of antecedent conditions and the length/intensity of rainfall were also mentioned as influencing factors by House and Warwick (1998).

3.6 Events not Included in the Analysis

Storm events with double hysteresis loops or more complex patterns are not included in the analysis. This is due to the nature of automated analysis that can only differentiate between either simple clockwise or anticlockwise events. These excluded events will be breafly discussed here.

As shown in Figure 2.11 (d), figure-eight hysteresis loops indicate a more complex response of nutrient concentrations to storm events. The complexity of these events can be attributed to multiple delivery flow pathways (Bieroza and Heathwaite, 2015), unsynchronised inputs of substantial runoff contributions from several upstream source areas for large and heterogenous drainage basin (Walling and Webb, 1980), and during large events ususally disconnected depositional zones may become connected (Perks et al, 2015). These events are also often rather complex in terms of of the rainfall input with several bursts of rainfall producing a complex runoff response.

The catchment itself is small sub-catchment that would not be expected to have the heterogenous properties of large catchments. It also does not have large variations of soil types or a large variety of land uses. The aquifers within the catchment being confined and the catchment hydrogeology dominated by the superficial deposits may make a significant proportion of infiltration result in relatively rapid shallow groundwater flow to the streams (section 2.2). This combined with the complex hydrological response of the underdrained clay soil to rainfall (section 1.4.3) added to a complex pattern of rainfall input can result in the complicated nitrate concentration responses observed during these storm events.

3.7 Projected Climate Change

3.7.1 Seasonality and Climate Change

The dependence of nitrate concentrations on seasonal variables shows how sensitive the catchment will be to future climate change. The most important variables for average and maximum nitrate concentrations are temperatures and time since last peak (Table 3.3 and 3.4). Drier and hotter summers (section 2.2.2) act to increase these variables thus to increase nitrate concentrations in stream water. Although nitrate concentrations are predominantly dictated by source availability, transport still exert some influence, especially on total nitrate flux (Table 3.6). Wetter and warmer winters (section 2.2.2) will increase soil wetness and together with higher temperatures make more

nitrate available from the soil while increased precipitation will enable the transportation of this nitrate out of the catchment.

3.7.2 Model Predictive Performance

The HBV-Light was calibrated for the hydrological years 2013-14 and 2014-15 on a daily basis. The validated model achieved a Nash Sutcliffe model performance statistics of 0.87. The model was then subjected to a blind performance assessment on the 2013-14 and 2014-15 hydrological years and achieved a Nash Sutcliffe model performance statistics of 0.83, indicating it was a good fit to the observed data. The measured and modelling discharge hydrographs from the model are shown in Figure 3.6 below.
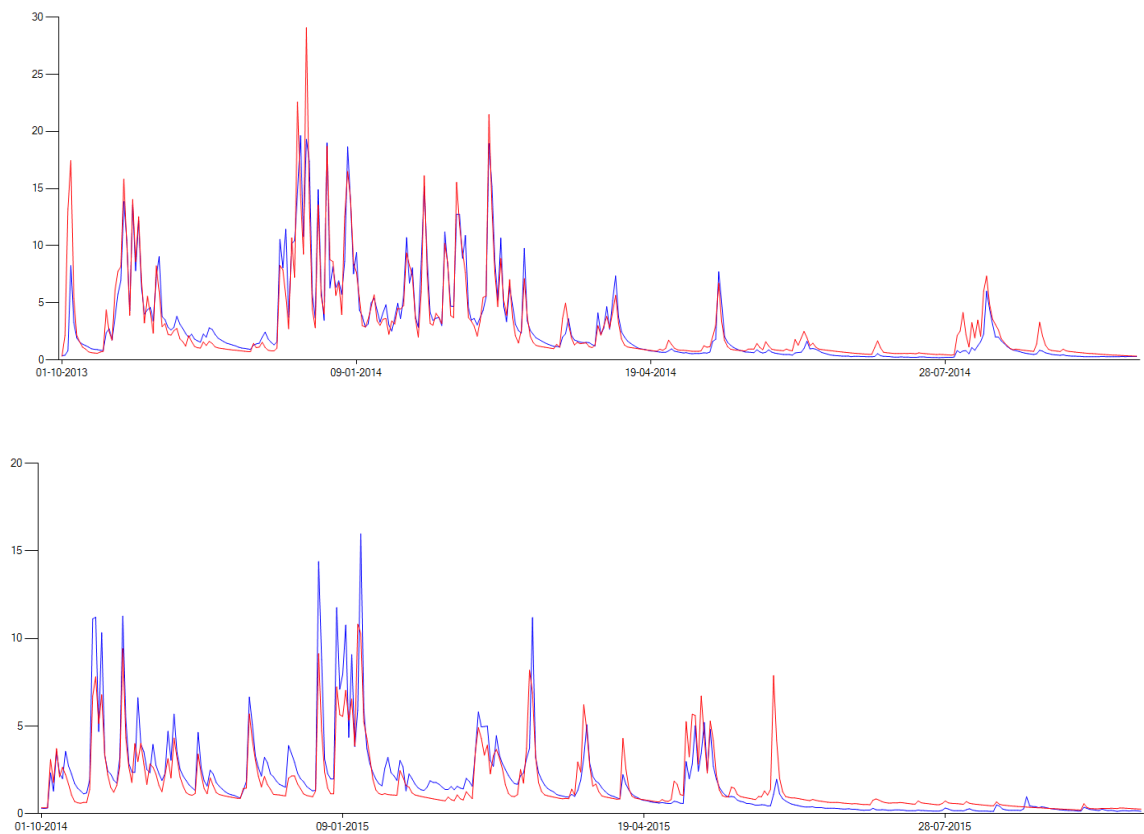
Figure 3.6 Measured (blue) and modelled (red) discharge hydrographs showing the validation of the HBV-Light model

3.7.3 Changes in Discharge Under Projected Climate Change

The HBV-Light model was used within the GLUE framework to predict changes in total discharge. The change factors graphs for the 2050s and 2080s compared to the baseline period are shown in Figure 3.7 below.
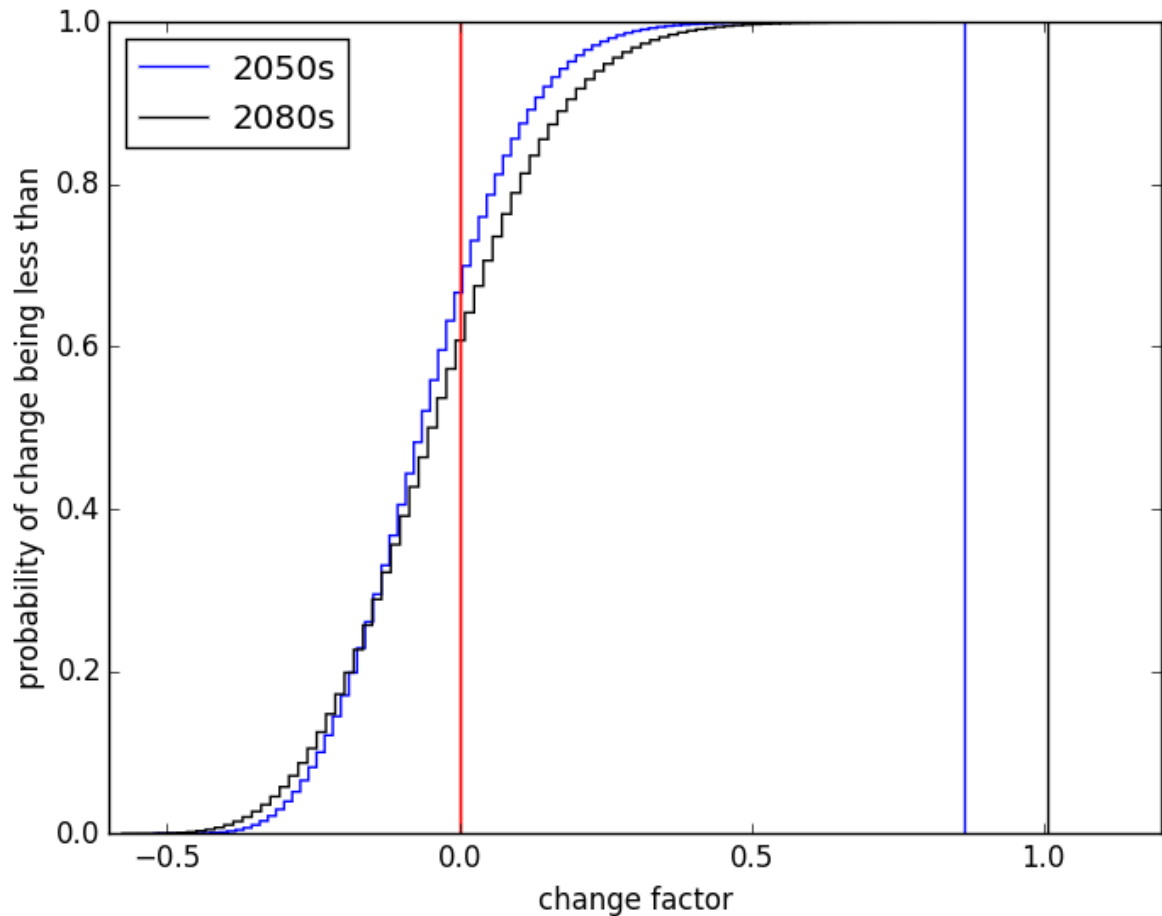
Figure 3.7 Change factor graphs for river discharge in the 2050s and 2080s

The probability of there being no change in discharge by 2050s is 0.67. Hence, there is 33% probability of there being a positive change factor and thus a 33% probability of increase in discharge by the 2050s. Central estimate (probability of 0.5) for the 2050s shows the change factor being -0.06. There is 0.9 probability of change factor for discharge being less than 0.13. The probability of there being no change in discharge by 2080s is 0.61. Hence, there is 39% probability of there being a positive change factor and thus a 39% probability of increase in discharge by the 2080s. Central estimate (probability of 0.5) for the 2080s shows the change factor being -0.05. There is 0.9 probability of change factor for discharge being less than 0.18.

3.7.4 Nitrate Concentrations and Climate Change

The stepwise multiple regression model for total nitrate flux has been chosen for calculation of future changes in nitrate response as it contains the largest R^2. Total nitrate flux was calculated per storm event and then summed for each of the series for baseline and 2050s and 2080s scenarios. Factor change was calculated and histograms plotted for each scenario.

Figure 3.8 shows the histogram of change factors for the 2050s and 2080s. The probability of there being no change in nitrate flux by 2050s is 0.29. Hence, there is 71% probability of there being a positive change factor and thus a 71% probability of increase in nitrate flux by the 2050s. The central estimate (probability of 0.5) for the 2050s shows the change factor being 0.03. There is 0.9 probability of change factor for total nitrate flux being less than 0.1.
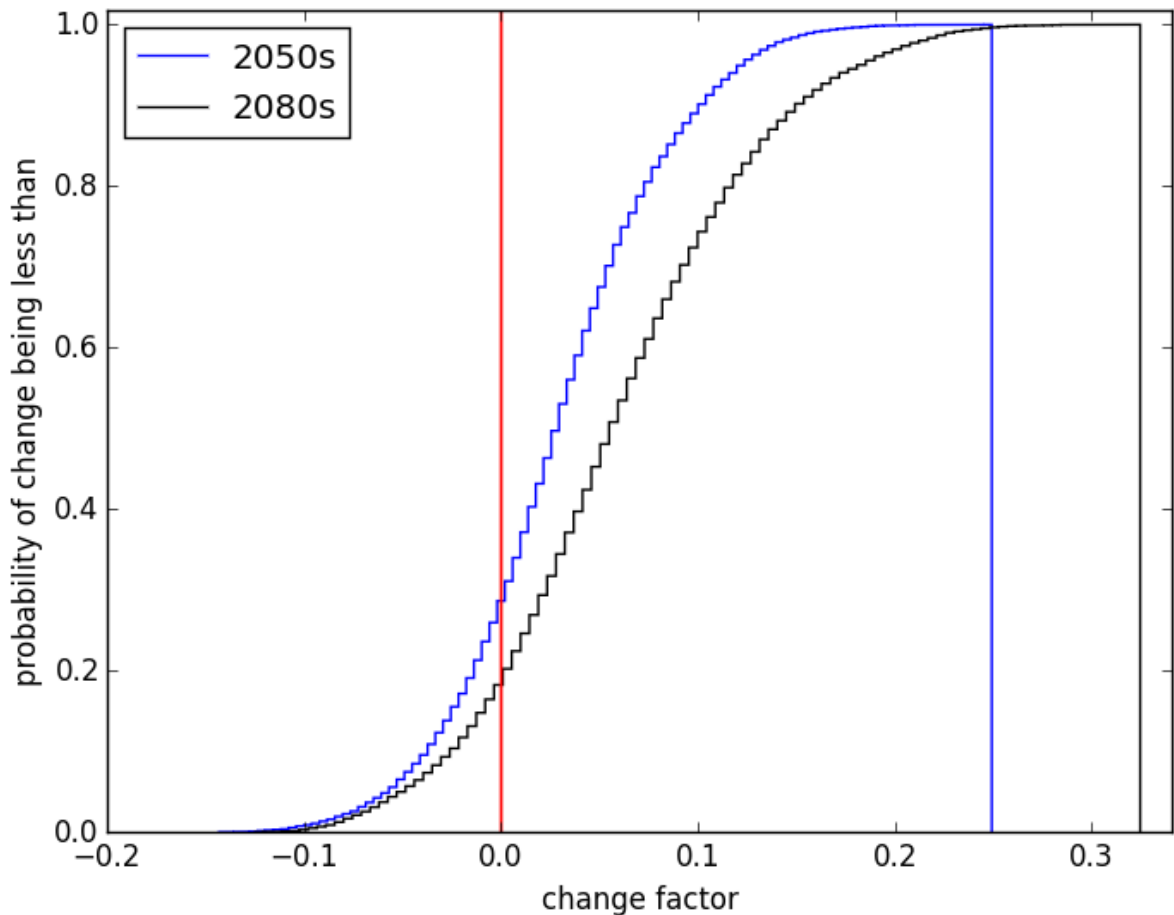
Figure 3.8 Histogram of factor change for nitrate flux in the 2050s and 2080s

The probability of there being no change in nitrate flux by the 2080s is 0.18. Hence, there is 82% probability of there being a positive change factor and thus an 82% probability of increase in nitrate flux by the 2080s. The central estimate (probability of 0.5) shows the change factor being 0.06. There is 0.9 probability of it being less than 0.15. The magnitude of nitrate flux increase will be larger by the 2080s than 2050s. The future mitigation measures should take this into account when preparing to counteract current nitrate concentrations.

3.8 Mitigation Measures

An Inventory of Mitigation Methods and Guide to their Effects on Diffuse Water Pollution, Greenhouse Gas Emissions and Ammonia Emissions from Agriculture is a user guide on mitigation measures prepared as part of Defra Project WQ0106 (Newell Price et al., 2011). The guide lists 83 mitigation measure options. The impact of each mitigation measure is assessed on nitrogen losses (nitrate, nitrite, ammonium), phosphorus (total and soluble), sediment, biological oxygen demand (BOD) and faecal indicator organism (FIO) losses to water, and gaseous emissions (i.e. ammonia, nitrous oxide, methane and carbon dioxide) to air. This guide was used to choose some specific nitrate leaching mitigation measures. Note should be taken that, while these mitigation measures reduce nitrate leaching losses, it can increase losses of other pollutants. It should be carefully considered, before implementing such measures, whether the decrease in target pollutant will outweigh the possible increases in other pollutants. The mitigation measures with medium (range of 20 to 80%) to high (range of 50 to 90%) effectiveness of nitrate leaching loss reduction were chosen. These methods are summarised below.

Land use conversion offers some of the largest potential reductions in nitrate leaching. Method 1A – Convert arable land to unfertilised and ungrazed grass reduces nitrate leaching losses by around 90% but the take-up of this method is expected to be low due to high costs. Method 1B – Arable reversion to low fertiliser input extensive grazing offers reduction of nitrate losses by around 80-90% but the uptake is likely to be low due to high costs and a significant change in farm business outlook and stockmanship skills. Method 2 – Convert arable /grassland to permanent woodlands result in reduction of nitrate losses of around 90%. Although this method would offer earnings in long run, the likely uptake is expected to be low due to dramatic change in land use and a short-term negative cash flow in the farming business. Method 3 – Convert land to biomass cropping (i.e. willow, poplar, miscanthus) would reduce nitrate leaching losses by around 50% but is unlikely to be taken up due to short-term negative cash flow and changes to the farming business.

The mitigation measures requiring change in the method of farming business rather than complete change in the land use are more likely to be taken up. These methods offer less effective reduction in nitrate leaching losses but this is outweighed by the willingness of more farm businesses to take it up. Method 4 – Establish cover crops in the autumn reduce nitrate leaching losses by 30-60% and is expected to have low to moderate uptake by farm businesses. Method 6 – Cultivate land for crops in spring rather than autumn reduce nitrate leaching losses of 20-50% and is expected to have low to moderate uptake by farm businesses. Method 16 – Allow field drainage systems to deteriorate offer reduction in nitrate leaching losses of 10-50%. This method is unlikely to be taken up without financial incentives due to costs based on loss of production due to poor drainage.

Method 65 – Change from a slurry to solid manure handling system reduce nitrate leaching losses of up to 50% and is expected to have low uptake by farm businesses due to high costs of building conversion and cost/limited availability of straw. Method 69 – Do not spread slurry or poultry manure at high-risk times reduce nitrate leaching losses of up to 20% of total manure N applied and expected moderate to high uptake by farm businesses. Method 74 – Transport manure to neighbouring farms reduce nitrate leaching losses (on exporting farm) by up to 10% on the dairy farm and up to 50% of pig/poultry farms and is expected to have low/moderate uptake on dairy farms and moderate/high on pig/poultry farms within Nitrate Vulnerable Zones (low outside Nitrate Vulnerable Zones). Method 75 – Incinerate poultry litter for energy recovery reduces nitrate leaching losses. The exact effectiveness/ magnitude of reduction was not stated for this method. Method 82 – Irrigate crops to achieve optimum yields reduce nitrate leaching losses by around 40% and is expected to have low uptake due to high costs of implementation.

Within Newby Beck cachment the effect of increasing temperatures with climate change will act to increase the amounts of nitrate leaving the catchment (Section 3.6.1 and 3.6.4). Nitrate concentrations are influenced mainly by supply-related variables (section 3.5.8), therefore mitigation measures that intercept water that is transporting this nitrate combined with reduction of nitrate supplied by anthropogenic activities will be the best choices. The best combination of mitigation options for the Newby Beck catchment to reduce nitrate supplied by anthropogenic sources are land use change (Methods 1A, 1B, 2 and3) and change in the method of farming (Methods 4, 6, 69 and 82). As it is unlikely that the option of land use change will be taken up widely, the rest of the land should be treated by changing current farming methods. Limitation of N application to soil given higher production of inorganic N compounds in warmer soils would also be useful to combat changes brought upon by future temperature increases. This catchment would also benefit from a combination of mitigation measures that involve interception of the overland flow and particularly subsurface drain outflow by retention ponds (as is already implemented at Eden DTC) and interception of shallow groundwater by riparian zones by altering the drainage system. Ng et al (2000) found that controlled drainage and subsurface irrigation also showed reduced nitrate leaching.

Chapter 4: Conclusions

This research has concentrated on a small river catchment dominated by grassland underlain by drained clay soil. The types of nitrate response were classified according to the type of hysteresis loop obtained when concentration was plotted against discharge for each storm event. This approach has helped with understanding of catchment hydrology and possible nitrate sources. Correlation matrices were used to see how the antecedent variables are correlated to each other and to nitrate. To narrow down the variable list and focus on the most relevant variables, stepwise multiple regression analysis was used. This work informed the understanding of how nitrate leaching patterns depend on transport and source and the effect of antecedent conditions on hydrology and leaching patterns. Future nitrate flux was also estimated. High-frequency observations of nitrate concentration and stream discharge provided a large database of storm events which could be analysed. The analysis conducted here involved several hundred storm events over a three-year period, compared to the relatively few that could be captured using traditional water sampling equipment in a short fieldwork campaign. For example, in a 3-year period in the Slapton Wood catchment, 1979-82, data for only 26 storms were collected using manual sampling (Burt, pers. comm.). Thus, the conclusions here about the frequency of different types of hysteresis events and about the variables controlling nitrate concentrations and fluxes can be very much more convincing than are likely to have emerged hitherto (cf. Burt and Arkell, 1987).

The Newby Beck catchment is dominated by clockwise hysteresis events but anti-clockwise storm events still account for over one third of total storm events. Clockwise hysteresis storm events are source (or supply) limited. These events display small initial increase in nitrate concentration due to readily available nitrate being mobilised through quick delivery pathways. It is then followed by a dilution of stream water. Anticlockwise storms are generally the events when large amounts of nitrate are leached. This type of event is transport-limited. Nitrate concentrations are governed by preparatory processes of nitrate production at source in the soil. Projected climate change may act to alter these preparatory processes through increasing temperatures and more severe droughts in summer which would result in even more extreme nitrate leaching events. The transport from the catchment is driven by shallow subsurface flow through the drained clay soil and increasing precipitation with future climate change in winter will act to further facilitate the rapid transportation of nitrate to stream water. Combined with projected increase in winter temperatures, the increase in transport availability could act to increase the amount of nitrate leaving the catchment. It was estimated that there is a 71% probability of future nitrate flux increasing by the 2050s and an 82% probability of future nitrate flux increasing by the 2080s.

Mitigation measures that intercept water that is transporting this nitrate combined with reduction of nitrate supplied by anthropogenic activities will be the best choices. Change in land use and alteration of the method of farming would help reduce anthropogenic sources of nitrate. These methods should be combined with mitigation measures that involve interception of the overland flow and particularly subsurface drain outflow by retention ponds and interception of shallow groundwater by riparian zones by altering the drainage system. Controlled drainage and subsurface irrigation might also be useful in reducing nitrate leaching.

The main strength of this research is in the automated analysis that has provided the opportunity to analyse a high frequency dataset over 3-year span with a number of antecedent variables. It allowed fast and consistent identification of storm events. The calculation of future changes in nitrate concentrations (as outlined in section 2.9) was also only possible because of the use of automated analysis. However, this research lacks the complementary analysis of individual source areas that would have provided a more detailed understanding of catchment dynamics; future research should seek to identify the soil hydrological processes that drive the hysteresis responses seen in the Newby Beck stream.

References

Allen, D.J., Newell, A.J. and Butcher, A.S. (2010) 'Preliminary review of the geology and hydrogeology of the Eden DTC sub-catchments', *British Geological Survey Open Report,* OR/10/063, pp. 1-35.

Anderson, M.G. and Burt, T.P. (1990) '*Process studies in hillslope hydrology'*, John Wiley & Sons Ltd., Chichester, pp.371-372.

Beven, K., and Germann, P. (1982) 'Macropores and Water Flow in Soils', *Water Resources Research,* vol. 18, no. 5, pp. 1311-1325.

Beven, K.J. and Binley, A.M. (1992) 'The future of distributed models: model calibration and uncertainty prediction', *Hydrological Processes*, vol. 6, pp. 279–298.

Bieroza, M.Z. and Heathwaite, A.L. (2015) 'Seasonal variation in phosphorus concentration-discharge hysteresis inferred from high-frequency in situ monitoring', *Journal of Hydrology,* vol. 524, pp.333-347.

Booltink, H.W.G (1995) 'Field monitoring of nitrate leaching and water flow in a structured clay soil', *Agriculture, Ecosystems and Environment,* vol. 52, pp.251-261.

Bouma, J. and Dekker, L. W. (1978) 'A case study on infiltration into dry clay soil I. Morpological observations', *Geoderma*, vol 20, pp. 27-40.

Bouma, J. (1981) 'Soil morphology and preferential flow along macropores', *Agricultural water management,* vol. 3, pp. 235-250.

Bowes, M.J., Jarvie, H.P., Halliday, S.J., Skeffington, R.A., Wade, A.J., Lowenthal, M., Gozzard, E., Newman, J.R. and Palmer-Felgate, E.J. (2015) 'Characterising phosphorus and nitrate inputs to a rural river using high-frequency concentration-flow relationships', *Science of the Total Environment,* vol. 511, pp.608-620.

Brady, N.C. and Weil, R.R. (2008) 'The nature and properties of soils' (14th ed.), Pearson Prentice Hall, p. 677.

Burt, T.P. and Arkell, B.P (1987) 'Temporal and spatial patterns of nitrate losses from an agricultural catchment', Soil Use and Management, vol 3, No. 4, pp. 138-142.

Burt, T.P., Heathwaite, A.L. and Trudgill S.T. (1993) *Nitrate: Processes, Patterns and Management,* Wiley, Chichester.

Castellano, M.J. and Kaye, J.P. (2009) 'Global Within-Site Variance in Soil Solution Nitrogen and Hydraulic Conductivity Are Correlated to Clay Content', *Ecosystems,* vol. 12, pp.1343-1351.

Cranfield University 2016. The Soils Guide. Available: www.landis.org.uk. Cranfield University, UK. Last accessed 14/07/2016

Demonstration Test Catchments (2014) *Newsletter-Winter 2014* Available at: http://www.demonstratingcatchmentmanagement.net/?p=983 (Accessed 08 March 2016).

National Demonstration Test Catchments Network, *Demonstration Test Catchments – Overview,* Available at: http://www.demonstratingcatchmentmanagement.net/ (Accessed 22 September 2016).

Department for Environment, Food & Rural Affairs, Department for Agriculture and Rural Development (Northern Ireland), Welsh Assembly Government, The Department for Rural Affairs and Heritage The Scottish Government, Rural and Environment Research and Analysis Directorate (2015) 'Agriculture in the United Kingdom 2014', *National Statistics,* p. 70.

Driessen et al. (2010) 'The hydrological response of the Ourthe catchment to climate change as modelled by the HBV model'; *Hydrology and Earth System Science,* Vol.14, pp. 651-665

EdenDTC – A Defra Demonstration Test Catchment (2016) *What is Eden DTC?* [online], Available at: http://www.edendtc.org.uk/what-is-the-river-eden-demonstration-test-catchments-project/ (Accessed 10 February 2016)

EdenDTC (2016)[a]: Newby Beck Catchment Automatic Weather Station Dataset. Version:1. [dataset] Freshwater Biological Association [publisher]. doi:10.17865/dtceden211

EdenDTC (2016)[b]: Newby Beck Outlet Discharge Dataset. Version:1. [dataset] Freshwater Biological Association [publisher]. doi:10.17865/dtceden242

EdenDTC (2016)[c]: Newby Beck Outlet Analogues (nutrients) Dataset. Version:1. [dataset] Freshwater Biological Association [publisher]. doi:10.17865/dtceden240

European Commission (2015) 'The Water Framework Directive and the Floods Directive: Actions towards the 'good status' of EU water and to reduce flood risks', *Communication From The Commission To The European Parliament And The Council,* Brussels, COM(2015) 120 final

Evans, C., Davies, T.D. and Murdoch, P.S. (1999) 'Component flow processes at four streams in the Catskill Mountains, New York, analysed using episodic concentration/discharge relationships', *Hydrological Processes,* vol. 13, pp. 563-575.

Findlay, S. and Fisher, D. (2013) 'Ecosystem attributes related to tidal wetland effects on water quality', *Ecology*, vol. 94 No. 1, pp. 117-125.

Granger, S.J., Hawkins, J.M.B., Bol, R., White, S.M., Naden, P., Old, G., Bilotta, G.S., Brazier, R.E., Macleod, C.J.A and Haygarth, P.M. (2010) 'High temporal resolution monitoring of multiple pollutant responses in drainage from an intensively managed grassland catchment caused by a summer storm', *Water Air Soil Pollut*, vol. 205, pp. 377-393.

Haigh, R.A. & White, R.E. (1986) 'Nitrate leaching from small, underdrained grassland, clay catchment', *Soil Use and Management,* vol. 2., No. 2, pp 65-69

Harris, G.L. and Catt, J.A. (1999) 'Overview of the studies on the cracking clay soil at Brimstone Farm, UK', *Soil Use and Management,* vol. 15, pp. 1-239.

Heppell, C.M., Burt, T.P. and Williams, R.J. (2000) 'Variations in the hydrology of an underdrained clay hillslope', *Soil Use and Management*, vol. 227, pp.236-256.

House, W.A. and Warwick, M.S. (1998) 'Intensive measurements of nutrient dynamics in the River Swale', *The Science of the Total Environment,* vol. 210/211, pp. 111-137.

House, W.A. and Warwick, M.S. (1998) 'Hysteresis of the solute concentration/discharge relationship in rivers during storms', *Water Resources,* vol. 32, no. 8, pp. 2279-2290.

Jarvis N.J. (2007) 'A review of non-equilibrium water flow and solute transport in soil macropores: principles, controlling factors and consequences for water quality', *European Journal of Soil Science,* vol. 58, pp.523-546.

Jones, P. D., Kilsby, C. G., Harpham, C., Glenis, V., Burton, A. (2009), 'UK Climate Projections science report: Projections of future daily climate for the UK from the Weather Generator.', University of Newcastle, UK.

Kirchner, J.W., Feng, X., Neal, C. and Robson, A.J. (2004) 'The fine structure of water-quality dynamics: the (high-frequency) wave of the future', *Hydrological Processes,* vol. 18, pp.1353-1359.

Kneale, W.R. (1986) 'The hydrology of a sloping, structured clay soil at Wytham, near Oxford, England', *Journal of Hydrology,* vol. 85, pp. 1-14.
Kohler, M. A. and Linsley, R. K. (1951) 'Predicting the runoff from storm rainfall', *U.S. Department of Commerce,* Research paper no. 34. pp. 1-10.

Lazenby, A. (1981) 'Nitrogen relationships in grassland ecosystems', *Proceedings of the 14th International Grassland Congress,* pp. 56-63.

Leeds-Harrison, P.B., Shipway, C.J.P., Jarvis, N.J. & Youngs, E.G. (1986) The influence of soil macroporosity on water retention, transmission and drainage in a clay soil. *Soil Use and Management,* vol. 2, no. 2, pp 47-50.

Linsley, R.K., Kohler, M.A., and Paulhus, J.L. (1982) Hydrology for engineers (3rd ed.): New York, McGraw-Hill, p. 508.

Lloyd, C.E.M., Freer, J.E., Johnes, P.J. and Collins, A.L. (2016) 'Using hysteresis analysis of high-resolution water quality monitoring data, including uncertainty, to infer controls on nutrient and sediment transfer in catchments', *Science of Total Environment,* vol. 543, pp. 388-404.

Met Office (2011) 'Climate: observations, projections and impacts', *Met Office,* Exeter, pp. 1-147.

Met Office (2016) *When does spring start?* [online], Available at: http://www.metoffice.gov.uk/learning/learn-about-the-weather/how-weather-works/when-does-spring-start (Accessed 22 February 2016).

Minitab Inc. (2015) *Minitab 17 Support: Basics of stepwise regression* [online], Available at: http://support.minitab.com/en-us/minitab/17/topic-library/modeling-statistics/regression-and-correlation/basics/basics-of-stepwise-regression/ (Accessed 03 January 2016)

Murphy, J.M., Sexton, D.M.H., Jenkins, G.J., Boorman, P.M., Booth, B.B.B., Brown, C.C., Clark, R.T., Collins, M., Harris, G.R., Kendon, E.J., Betts, R.A., Brown, S.J., Howard, T. P., Humphrey, K. A., McCarthy, M. P., McDonald, R. E., Stephens, A., Wallace, C., Warren, R., Wilby, R., Wood, R. A. (2009), UK Climate Projections Science Report: Climate change projections. Met Office Hadley Centre, Exeter.

National Demonstration Test Catchments Network (2016) *Demonstration Test Catchments – Overview* [online], Available at: http://www.demonstratingcatchmentmanagement.net/ (Accessed 10 February 2016).

Neptune Solutions (2016) *Morland Catchment* [online], Available at: https://neptunesolutions.wordpress.com/morland-catchment/ (Accessed 15 February 2016)

Ng, H.Y.F., Drury, C.F., Serem, V.K., Tan, C.S. and Gaynor J.D (2000) 'Modelling and testing of the effect of tillage, cropping and water management practices on nitrate leaching in clay loam soil', *Agricultural Water Management,* Vol. 43, pp. 111-131.

Oenema, O., Velthof, G.L., Yamulki, S. and Jarvis, S.C. (1997) 'Nitrous oxide emissions from grazed grassland', *Soil Use and Management,* vol. 13, pp. 288-295.

O'Sullivan, P.E. (1995) 'Eutrophication', *Intern. J. Environmental Studies,* vol. 47, pp. 173-195.

Perks, M.T., Owen, G.J., Benskin, C.McW.H., Jonczyk, J., Deasy, C., Burke, S., Reany, S.M. and Haygarth, P.M. (2015) 'Dominant mechanisms for the delivery of fine sediment and phosphorus to fluvial networks draining grassland dominated headwater catchments', *Science of the Total Environment,* vol. 523, pp.178-190.

Pettyjohn, W.A., and Henning, Roger, (1979) 'Preliminary estimate of ground-water recharge rates, related streamflow and water quality in Ohio: Ohio State University Water Resources Center Project Completion Report Number 552', p. 323.

Pilgrim, D.H., Huff, D.D. and Steele, T.D. (1979) 'Use of specific conductance and contact time relations for separating flow components in storm runoff', *Water Resources Research,* vol. 15, pp. 329-339.

Pretty, J.N., Mason, C.F., Nedwell, D. B. Hine, S. L., Leaf, S. and Dils, R. (2003) 'Environmental costs of freshwater eutrophication in England and Wales', *Environmental Science & Technology,* vol. 37, no. 2, pp. 201-208.

Reid, I. and Parkinson, R.J. (1984) 'The Nature of the Tile-drain Outfall Hydrograph in Heavy Clay Soils', *Journal of Hydrology,* vol. 72, pp.289-305.

Robinson, M. and Beven, K.J. (1983) 'The Effect of Mole Drainage on The Hydrological Response of a Swelling Clay Soil ', *Journal of Hydrology,* vol. 64, pp.205-223.

Robinson, M., Mulqueen, J. and Burke, W. (1987) 'On Flows From a Clay Soil – Seasonal Changes And The Effect of Mole Drainage ', *Journal of Hydrology,* vol. 91, pp.339-350.

Ryden, J.C., Ball, P.R. and Garwood, E.A. (1984) 'Nitrate leaching from grassland', *Nature,* vol. 311, pp. 50-53.

Scholefield, D. Tyson, K.C., Garwood, E. A., Armstrong, A. C. , Hawkins, J. & Stone, A.C. (1993) Nitrate leaching from grazed grassland lysimeters: effects of fertilizer input, field drainage, age of sward and patterns of weather. *Journal of Soil Science,* vol. 44, pp. 601-613

Scholefield, D., Lord, E.I., Rodda, H.J.E. and Webb, B. (1996) 'Estimating peak nitrate concentrations from annual nitrate loads', *Journal of Hydrology,* vol. 186, pp.355-373.

Seibert, J. and Vis, M. (2012) 'Teaching hydrological modeling with a user-friendly catchment-runoff-model software package', *Hydrology and Earth System Sciences*, vol. 16, pp. 3315–3325.

Seibert, J. (2000) 'Multi-criteria calibration of a conceptual runoff model using a genetic algorithm', *Hydrology and Earth System Sciences,* Vol. 4(2), pp. 215-224.

Siwek, J., Siwek, J.P. and Zelazny, M. (2013) 'Environmental and land use factors affecting phosphate hysteresis patterns of stream water during flood events (Carpathian Foothills, Polland)', *Hydrological Processes*, vol. 27, pp.3674-3684.

Sloto, R.A. and Crouse, M.Y. (1996) 'HYSEP: A Computer Program For Streamflow Hydrograph Separation and Analysis', *U.S. Geological Survey,* Water-Resources Investigations Report 96-4040.

Smith, V.H, Tilman, G.D., Nekola, J.C. (1999) 'Eutrophication: impacts of excess nutrient inputs on freshwater, marine, and terrestrial ecosystems', *Environmental Pollution*, vol. 100, pp. 179-196

Stout, W.L., Fales, S.L., Muller, L.D., Schnabel, R.R. and Weaver, S.R. (2000) 'Water quality implications of nitrate leaching from intensively grazed pasture swards in the northeast US', *Agriculture, Ecosystems and Environment,* Vol. 77, pp. 203-210.

Tomer, M.D., Meek, D.W, Jaynes, D.B and Hatfield J.L. (2003) 'Evaluation of Nitrate Nitrogen Fluxes from a Tile-Drained Watershed in Central Iowa', *J. Environ.Qual.,* vol. 32, pp. 642-653.

UK Climate Change Projections (2014) *Maps & key findings 2050s* [online], Available at: http://ukclimateprojections.metoffice.gov.uk/23847?emission=medium (Accessed 17 July 2016)

UK Climate Change Projections (2014) *Maps & key findings 2080s* [online], Available at: http://ukclimateprojections.metoffice.gov.uk/23848?emission=medium (Accessed 17 July 2016)

Vinten, A.J.A., Vivian, B.J., Wright, F. and Howard, R.S. (1994) 'A comparative study of nitrate leaching from soils of differing textures under similar climatic and cropping conditions', *Journal of Hydrology,* vol. 159, pp.197-213.

Vitousek, P.M., Aber, J.D., Howarth, R. W., Likens, G. E., Matson, P. A., Schindler, D.W., Schlesinger, W.H. and Tilman, D.G. (1997) 'Human alteration of the global nitrogen cycle: sources and consiquences', *Ecological applications,* vol. 7, no. 3, pp. 737-750.

Webb, B.W. and Walling, D.E. (1985) 'Nitrate behaviour in streamflow from a grassland catchment in Devon, U.K.', *Water resources,* vol. 19, no. 8, pp. 1005-1016.

White, R.E., Wellings, S.R. and Bell, J.P. (1983) 'Seasonal variations in nitrate leaching in structured clay soils under mixed land use', *Agricultural Water Management,* vol. 7, pp. 391-410.

WHO (2011) 'Nitrate and nitrite in drinking-water', WHO/SDE/WSH/07.01/16/Rev/1

Williams, G.P. (1989) 'Sediment concentration versus water discharge during single hydrologic events in rivers', *Journal of Hydrology,* vol. 111, pp.89-106.

Williams, J.R., Rose, S.C. and Harris, G.L. (1995) 'The impact on hydrology and water quality of woodland and set aside establishment on lowland clay soils', *Agriculture Ecosystems &Environment,* vol. 54, pp. 215-222.

Appendix 1: Python code for data analysis

```python
"""Created on Mon Feb 16 15:21:27 2015

@author: Vaida

"""

from matplotlib.pylab import *

import datetime

import scipy.interpolate as sp

from prettytable import PrettyTable

import statsmodels.api as sm

from numpy import *


def plotStormNQ(Q, N):

    scatter(springQ, springN, color='green', marker = 'v',
label = 'spring')

    scatter(summerQ, summerN, color='orange', marker
= 's', label = 'summer')

    scatter(autumnQ, autumnN, color='red', marker = '*',
label = 'autumn')

    scatter(winterQ, winterN, color='blue', label =
'winter')

    ax=gca()

    ax.legend(loc='best', scatterpoints=1)

    ax.set_ylabel("Average nitrate concentration mg/l")

    ax.set_xlabel("Total discharge m3/s")

    ax.set_ylim()

    ax.set_xlim()


def plotBaseFlow(times1, dischargem3s1,
baseflow30min):

    plot(times1, dischargem3s1, color='blue',
linewidth='0.4')

    ax=gca()

    ax.set_ylabel("discharge m3 / s", color='blue')

    ax.set_xlabel("time")

    ax.set_ylim(bottom=-1, top=9)

    ax2=ax.twinx()

    ax2.plot(times1, baseflow30min,
color='red',linewidth='0.4')

    ax2.set_ylabel("baseflow30min",color='red')
```

```python
    ax2.set_xlabel("time", color='red')

    ax2.set_ylim(bottom=-1, top=9)


def plotThroughFlow(times1, dischargem3s1,
throughFlow):

    plot(times1, dischargem3s1, color='blue',
linewidth='0.4')

    ax=gca()

    ax.set_ylabel("discharge m3 / s", color='blue')

    ax.set_xlabel("time")

    ax.set_ylim(bottom=-1, top=9)

    ax2=ax.twinx()

    ax2.plot(times1, throughFlow,
color='red',linewidth='0.4')

    ax2.set_ylabel("through flow",color='red')

    ax2.set_xlabel("date", color='red')

    ax2.set_ylim(bottom=-1, top=9)


def plotStormEvent(times1, dischargem3s1,
FinalStormEvent):

    plot(times1, dischargem3s1, color='blue',
linewidth='0.4')

    ax=gca()

    ax.set_ylabel("discharge m3 / s", color='blue')

    ax.set_xlabel("time")

    ax.set_ylim(bottom=-1, top=9)

    ax2=ax.twinx()

    ax2.plot(times1, FinalStormEvent,
color='red',linewidth='0.4')

    ax2.set_ylabel("Storm event",color='red')

    ax2.set_xlabel("date", color='red')

    ax2.set_ylim(bottom=-1, top=9)


def plotStormsOnly(times1, FinalStormEvent):

    plot(times1, FinalStormEvent, color='black',
linewidth='0.4')

    ax=gca()

    ax.set_ylabel("Discharge m3 / s", color='black')
```

```
    ax.set_xlabel("Date")

    ax.set_ylim(bottom=-1, top=9)


def plotPTQ(times, rainfallmm, dischargem3s,
nitratemgl, waterTemperatureC):

    subplot(211)

    xticks(rotation=45)

    tight_layout()

    plot(times, dischargem3s, color='blue',
linewidth='0.4')

    ax=gca()

    ax.set_ylabel("discharge m3 / s", color='blue')

    ax.set_xlabel("time")

    ax2=ax.twinx()

    ax2.plot(times, nitratemgl,
color='red',linewidth='0.4')

    ax2.set_ylabel("nitrate mg / l",color='red')


    subplot(212)

    xticks(rotation=45)

    tight_layout()

    bar(times, rainfallmm, color='blue', linewidth='0')

    ax=gca()

    ax.set_ylabel("rainfallmm mm", color='blue')

    ax.set_xlabel("time")

    ax2=ax.twinx()

    ax2.plot(times, waterTemperatureC,
color='red',linewidth='0.4')

    ax2.set_ylabel("water temp C",color='red')


times = []

times1 = []

timesDaily = []

decDay1 = []

rainfallmm = []

dischargem3s = []

dischargem3s1 = []

nitratemgl = []
```

```
waterTemperatureC =[]

dailyMin = []

localMin = []

baseFlow = []

baseflow30min = []

throughFlow = []

stormEvent = []

nitratemgl1 = []

N = []

Q = []

ID = []

stormID = []

season = []

seasonID = []


data = open("Morland_outlet_Copy_to_use.csv")

for i in range(0, 2):

    data.readline()


n = 0

for line in data.readlines():

    line = line.split(',')

    tim = datetime.datetime.strptime(line[0],
"%d/%m/%Y %H:%M")

    times.append(tim)


    if (len(line[1]) > 0):

        rainfallmm.append(float(line[1]))

    else:

        rainfallmm.append(0.0)


    if (len(line[2]) > 0):

        dischargem3s.append(float(line[2]))

    else:

        dischargem3s.append(0.0)


    if (len(line[9]) > 0):
```

```
        nitratemgl.append(float(line[9]))

    else:

        nitratemgl.append(0.0)


    if (len(line[10]) > 2):

        waterTemperatureC.append(float(line[10]))

    else:

        waterTemperatureC.append(0.0)

    n += 1


#plotPTQ(times, rainfallmm, dischargem3s, nitratemgl,
waterTemperatureC)


'''to find daily min and number of days'''


n = 0 # do i want the daily values to start with 0 or 1

currentMin = 9999999

tt = times[0].timetuple() #print tt reads from the second
measurement

currentDay = tt[2] # reads correct day

#print currentDay


for  i in range(0, len(times)): # when it reaches the last
day it does not append day number and dailyMin

    tt = times[i].timetuple()

    d = tt[2] #day

    if (d != currentDay):

        dailyMin.append(currentMin)

        currentMin = 9999999

        currentDay = d

        timesDaily.append(n)

        n=n+1

    else:

        if dischargem3s[i] < currentMin:

            currentMin = dischargem3s[i]

timesDaily.append(1114)

dailyMin.append(0.3)
```

```
#skips the last day, the last dailyMin should be 0.3 and
last timesDaily 1114

#so i added the values :(


#plot (dailyMin)

#print len(dailyMin) #= len 1115

#print dailyMin

#print len(timesDaily) # =len 1115  n=0 start with 0
end with 1114

#print timesDaily


'''to find local minimums at a 3 day window(the day
before and after current day)'''


for i in range(1, len(dailyMin)-1):

    minValue = min(dailyMin[i-1], dailyMin[i],
dailyMin[i+1])

    localMin.append(minValue)


#print len(localMin) #=1113

#plot(localMin)


'''to make interpolated arrays the same length'''


timesDaily2 = timesDaily[1:]

timesDaily2.pop()


#print timesDaily2

#print len(timesDaily2) #= 1113

#timesDaily2 starts with value 1 and ends with value
1113


'''linear interpolation'''


baseFlow = sp.interp1d(timesDaily2, localMin,
kind='linear')


#plot (timesDaily2, baseFlow(timesDaily2))
```

```python
#times starts with value datetime.datetime(2011, 9, 15,
0, 0)

#and ends with datetime.datetime(2014, 10, 1, 23, 30)

#print times


'''to convert the total number of days into half an hourly
data'''


for i in range(24, len(times)-27):

    tt = times[i].timetuple()

    decDay = (float(tt[0]) - 2011)*(365) #year 2012 feb
has 29days

#    print decDay

    decDay += float(tt[7] - 257) #day of year

#    print decDay

    decDay += float(tt[3]) / 24 #hour

#    print decDay

    decDay += float(tt[4]) / 1440 #minute !!!

#    print decDay

    decDay1.append(decDay)

#    print tt


#print decDay1

#print len(decDay1)

out = open("decDay.txt", 'w+')


out.write("decDay\n")

for i in range(0, len(decDay1)):

    out.write(str(decDay1[i]))

    out.write("\n")

    out.flush()


'''shorten the discharge array to be able to find the
throughflow'''


dischargem3s1 = dischargem3s[24:53448]

times1 = times[24:53448]

nitratemgl1 = nitratemgl[24:53448]
```

```python
#print len (dischargem3s)=53475

#print len (dischargem3s1) #= 53424

#print times1[53423]

#print len(times1)

'''interpolation to half an hourly values'''


baseflow30min = baseFlow(decDay1)


#plot (times1, baseflow30min)

#plotBaseFlow(times1, dischargem3s1,
baseflow30min)


'''to find throughflow'''


for i in range(0, len(times1)):

    throughFlow.append(dischargem3s1[i] -
baseflow30min[i])


#print throughFlow

#plotThroughFlow(times1, dischargem3s1,
throughFlow)


'''describe storm events'''


for i in range(0, len(decDay1)):

    if (dischargem3s1[i] > baseflow30min[i]):

        stormEvent.append(dischargem3s1[i])

    else:

        stormEvent.append(nan)


#print len(stormEvent)

#plot (times1, stormEvent)

#plotStormEvent(times1, dischargem3s1, stormEvent)


'''assign ID to storm events'''


n=1
```

```python
    for i in range(0, len(stormEvent)):

        if (stormEvent[i] is not (nan)):

            stormID.append(n)

            if (stormEvent[i+1] is (nan)):

                n=n+1

        else:

            stormID.append(nan)


#print stormID

#print len(stormID) #=53424 (1 to 1003)

#print len(nitratemgl1) #=53424


'''better description of storm events thestorm has to have more than

5 measurements and the peak must rise at least 50% above min flow'''


j = []    #temp list

jj = []   #temp list

stormEvent2 = [] # a list with storm events that have less than 5 measurements and peaks that risen less than 50% removed


n = 1

for i in range(0, len(stormID)):

    if (stormID[i] == n):

        j.append(dischargem3s1[i])

        jj.append(nan)

        if (stormID[i+1] is (nan)):

            maxDischarge = max(j)

            minDischarge = min(j)

            n = n + 1

            if ((jj.count(nan) < 5) or (maxDischarge < minDischarge*1.5)): #or (maxDischarge/2 < minDischarge)#count the number of measurements in storm event and append j (all nan) if number is less than 5

                stormEvent2.extend(jj) #the storm event replaced by nan

                del j[:] #empty the lists
```

```python
                del jj[:]

            else:

                stormEvent2.extend(j) #else append the storm event as usual

                del j[:] #empty the lists

                del jj[:]

    else:

        stormEvent2.append(nan)


#print stormEvent2

#print len(stormEvent2)


'''seperate storm events further, anything that falls 1/4 below'''


hydrograph = []

hydrograph2 = []


for i in range (0, len(stormEvent2)):

    if (stormEvent2[i] is not nan):

        if (stormEvent2[i] < stormEvent2[i-1]):

            hydrograph.append('f')

        else:

            hydrograph.append('R')

    else:

        hydrograph.append(nan)


#print stormEvent2


'''re-assign ID to storm events'''

stormID2 = []

n=1

for i in range(0, len(stormEvent2)):

    if (stormEvent2[i] is not (nan)):

        stormID2.append(n)

        if (stormEvent2[i+1] is (nan)):

            n=n+1
```

```
        else:

            stormID2.append(nan)


#print stormID2

# now all the measurements that are equal to falling
measurement are still marked as R

# need to change R's that are equal to the f's into f's

n = 1

fall = 0

for i in range(0, len(stormID2)):

    if (stormID2[i] == n):

        if (hydrograph[i] == 'f'):

            hydrograph2.append('f')

            fall = stormEvent2[i] #so the last 'f' before 'R' is
memorised

            if (stormID2[i+1] is (nan)):

                n = n + 1

                fall = 0

        else:

            if (stormEvent2[i] == fall): #now this
memorised 'f' is compared to 'R' and if it is == then 'R'
is replaced with 'f'

                hydrograph2.append('f')

                if (stormID2[i+1] is (nan)):

                    n = n + 1

                    fall = 0

            else:

                hydrograph2.append('R')

                if (stormID2[i+1] is (nan)):

                    n = n + 1

                    fall = 0

    else:

        hydrograph2.append(nan)


#print hydrograph2

#print len(hydrograph2)

# now all the R's and f's are correct

#need to decide which rises/falls are a beggining of a
new storm event or just a part of a complex storm event
```

```
hydrograph3 = []

hydrograph4 = []

k1 = [] #temp lists

k2 = []

k3 = []

k4 = []

rise = 0

fall = 0

#checking rises

for i in range (0, len(hydrograph2)): #check if each rise
that isnt the main peak is at least 1/3 (1/4?) above the
fall (except for the final rise which will be always
marked as a fall) (later will fill in the gaps or seperate
storm events)

    if (hydrograph2[i] is not (nan)):

        if (hydrograph2[i] is 'R'):

            k1.append(stormEvent2[i])

            k2.append('f')

            k3.append('R')

            if (hydrograph2[i+1] is ('f')):

                if (max(k1) >= rise):    #all the R's before
the main peak and including the main peak are rises

                    hydrograph3.extend(k3)

                    rise = max(k1)

                    del k1[:] #empty the lists

                    del k2[:]

                    del k3[:]

                else:

                    if (max(k1) >= (fall/3)*4): #old was
(max(k1)/3*2 >= fall) #is this correct? check# see if the
max of rise that is not the main peak is 1/3 (1/4?) more
than fall min

                        hydrograph3.extend(k3)

                        del k1[:] #empty the lists

                        del k2[:]

                        del k3[:]

                    else:

                        hydrograph3.extend(k2)

                        del k1[:] #empty the lists

                        del k2[:]
```

80

```
                    del k3[:]

            else:

                if (hydrograph2[i+1] is (nan)): #so the last
rise in the storm event will always be marked as a fall

                    hydrograph3.extend(k2)

                    rise = 0

                    fall = 0

                    del k1[:] #empty the lists

                    del k2[:]

                    del k3[:]

        else:

            hydrograph3.append(hydrograph2[i])# or 'f'

            k4.append(stormEvent2[i])

            if (hydrograph2[i+1] is ('R')):

                fall = min(k4)

                del k4[:]

            else:

                if (hydrograph2[i+1] is (nan)):

                    rise = 0

                    fall = 0

                    del k4[:]

    else:

        hydrograph3.append(nan)


#print hydrograph3

#print len(hydrograph3)


#check fall

rise = 0

for i in range (0, len(hydrograph3)): #check if each fall
is at least 1/3 (1/4?) below the previous rise (except for
the final fall which will be always marked as a fall)
(later will fill in the gaps or seperate storm events)

    if (hydrograph3[i] is not (nan)):

        if (hydrograph3[i] is 'f'):

            k1.append(stormEvent2[i])

            k2.append('f')

            k3.append('R')

            if (hydrograph3[i+1] is ('R')):

                if (min(k1) <= (rise/3)*2):     #all the R's
before the main peak and including the main peak are
rises

                    hydrograph4.extend(k2)

                    del k1[:] #empty the lists

                    del k2[:]

                    del k3[:]

                else:

                    hydrograph4.extend(k3)

                    del k1[:] #empty the lists

                    del k2[:]

                    del k3[:]

            else:

                if (hydrograph3[i+1] is (nan)): #so the last
rise in the storm event will always be marked as a fall

                    hydrograph4.extend(k2)

                    rise = 0

                    del k1[:] #empty the lists

                    del k2[:]

                    del k3[:]

        else:

            hydrograph4.append(hydrograph3[i])# or 'R'

            k4.append(stormEvent2[i])

            if (hydrograph3[i+1] is ('f')):

                rise = max(k4)

                del k4[:]

            else:

                if (hydrograph3[i+1] is (nan)):

                    rise = 0

                    del k4[:]

    else:

        hydrograph4.append(nan)


#print hydrograph4

#print len(hydrograph4)
```

```python
#final hydrograph with nan insesarted to seperate storm events
#final stormEvent with nan insesarted to seperate storm events

FinalStormEvent = []

FinalHydrograph = []

for i in range(0, len(stormEvent2)):

    if (hydrograph4[i] is not (nan)):

        if (hydrograph4[i] is 'R'):

            FinalStormEvent.append(stormEvent2[i])

            FinalHydrograph.append(hydrograph4[i])#or 'R'

        else:

            if (hydrograph4[i+1] is 'R'):

                FinalStormEvent.append(nan)

                FinalHydrograph.append(nan)

            else:

                FinalStormEvent.append(stormEvent2[i])

                FinalHydrograph.append(hydrograph4[i])#or 'f'

    else:

        FinalStormEvent.append(nan)

        FinalHydrograph.append(hydrograph4[i])#or 'nan'


#print FinalStormEvent
#print len(FinalStormEvent)


'''re-assign ID to storm events'''


FinalStormID = []


n=1
for i in range(0, len(FinalStormEvent)):

    if (FinalStormEvent[i] is not (nan)):

        FinalStormID.append(n)

        if (FinalStormEvent[i+1] is (nan)):

            n=n+1

    else:

        FinalStormID.append(nan)
```

```python
#print FinalStormID #was 110 events, after aditional 1:3 fall split its 229


'''04/06/16....check if there are storm events containing less than 5 data points again'''

LastCount = []

n = 1

j = 0 #to count data points within storm event


for i in range(0, len(FinalStormID)):

    if (FinalStormID[i] == n):

        j = j + 1

        if (FinalStormID[i+1] is (nan)):

            LastCount.append(j)

            n = n + 1

    else:

        j = 0


#print len(LastCount)
#print min(LastCount)


'''find season for each half an hourly measurement'''


for i in range(24, len(times)-27):

    tt = times[i].timetuple()

    m = tt[1]

    spring = [3,4,5]

    summer = [6,7, 8]

    autumn = [9,10,11]

    if (m in spring):

        season.append('spring')

    elif (m in summer):

        season.append('summer')

    elif (m in autumn):

        season.append('autumn')

    else:
```

```python
        season.append('winter')

#print season

#print len(season) #=53424


'''Calculate nitrate and discharge +season for each
storm event'''


n=1

s=0 # nitrate

ss=0 #discharge

sss=0 # season

z=0 # count the number of measurements


for i in range(0, len(FinalStormID)):

    if (FinalStormID[i] ==n):

        s += nitratemgl1[i]

        ss += dischargem3s1[i]

        z = z + 1 #count the amount of measurements to
find mean

        sss = season[i] #season is set to be the season last
day of the storm event was in?

        if (FinalStormID[i+1] is (nan)):

            mean = s/z

            N.append(mean)

            Q.append(ss)

            seasonID.append(sss)

            s=0

            ss=0

            sss=0

            n=n+1

            z=0


#print seasonID

#print len(seasonID)

#print N

#print len(N)

#print Q

#print len(Q)
```

```python
'''assign ID'''


n=1

for i in range(0, 248):

    ID.append(n)

    n=n+1

#print ID


'''a table to show ID, N and Q'''


x = PrettyTable()

x.add_column("ID",ID)

x.add_column("N",N)

x.add_column("Q",Q)

x.add_column("Season",seasonID)


out = open("resutls_table.csv", 'w+')

out.write(str(x))

out.write("\n")

#table difficult to view in console so outwrite to a file

#and can be easily viewed using notepad


#print x


'''linear regression to storm events by season'''


springN = []

springQ = []

for i in range (0, len(seasonID)):

    if (seasonID[i]=='spring'):

        springN.append(N[i])

        springQ.append(Q[i])


x = springQ

y = springN

x = sm.add_constant(x)
```

83

```python
        model = sm.OLS(y, x)

        fitted = model.fit()


        #print 'SPRING'

        #print fitted.summary()

        #scatter (springQ, springN)

        #plot (springQ, fitted.fittedvalues)


        summerN = []

        summerQ = []

        for i in range (0, len(seasonID)):

            if (seasonID[i]=='summer'):

                summerN.append(N[i])

                summerQ.append(Q[i])


        x = summerQ

        y = summerN

        x = sm.add_constant(x)

        model = sm.OLS(y, x)

        fitted = model.fit()


        #print 'SUMMER'

        #print fitted.summary()

        #scatter (summerQ, summerN)

        #plot (summerQ, fitted.fittedvalues)


        autumnN = []

        autumnQ = []

        for i in range (0, len(seasonID)):

            if (seasonID[i]=='autumn'):

                autumnN.append(N[i])

                autumnQ.append(Q[i])


        x = autumnQ

        y = autumnN

        x = sm.add_constant(x)

        model = sm.OLS(y, x)
```

```python
        fitted = model.fit()


        #print 'AUTUMN'

        #print fitted.summary()

        #scatter (autumnQ, autumnN)

        #plot (autumnQ, fitted.fittedvalues)


        winterN = []

        winterQ = []

        for i in range (0, len(seasonID)):

            if (seasonID[i]=='winter'):

                winterN.append(N[i])

                winterQ.append(Q[i])


        x = winterQ

        y = winterN

        x = sm.add_constant(x)

        model = sm.OLS(y, x)

        fitted = model.fit()


        #print winterN

        #print winterQ

        #print len(fitted.fittedvalues)

        #print len(winterN)

        #print 'WINTER'

        #print fitted.summary()

        #scatter (winterQ, winterN)

        #plot (winterQ, fitted.fittedvalues)


        '''find residuals for each storm event

        (observed y values minus estimated y values)'''


        h = [] #temp dischargem3s1

        hh = [] #temp nitratemgl1

        ye = [] #estimated y

        residuals = [] #

        peak = [] #max flow of each storm event
```

```python
n = 1

for i in range(0, len(FinalStormID)):

    if (FinalStormID[i] == n):

#       print str(stormID[i]) + " " + str(n)

        h.append(dischargem3s1[i]) #temporary list

        hh.append(nitratemgl1[i]) #temporary list

        x = h

        y = hh

        x = sm.add_constant(x)

        model = sm.OLS(y, x)

        fitted = model.fit()

        yp = fitted.fittedvalues #predicted y

        if (FinalStormID[i+1] is (nan)):

            peak.append(max(h)) #find a maximum
discharge to later find the falling and rising limb of
hydrograph

            ye.extend(yp)

            del h[:] # emty the list

            del hh[:] # emty the list

            n = n + 1

    else:

        ye.append(nan)


#print peak

#print len(peak)

#print ye

#print len(ye) #len 53424


for i in range(0, len(nitratemgl1)):

    residuals.append(nitratemgl1[i]-ye[i])


#print residuals

#print len(residuals)


'''calculate average residuals for a falling and rising
limb for each storm event'''


R= []
```

```python
F = []

r= [] #temp list

f = [] #temp list

n = 1


for i in range(0, len(FinalStormID)):

    if (FinalStormID[i] == n):

        if (FinalHydrograph[i] is 'R'):

            r.append(residuals[i]) #temporary list

        else:

            f.append(residuals[i]) #temporary list

        if (FinalStormID[i+1] is (nan)):

            R.append(float(sum(r))/len(r) if len(r) > 0
else float(nan)) #find average

            F.append(float(sum(f))/len(f) if len(f) > 0 else
float(nan))

            del r[:] # emty the list

            del f[:] # emty the list

            n = n + 1


#print R

#print len(R)

#print F

#print len(F)


'''R>F for clockwise hysteresis, R<F for anticlockwise
hysteresis'''

hysteresis = []


for i in range(0, len(ID)):

    if (R[i] > F[i]):

        hysteresis.append('C')

    elif(R[i] < F[i]):

        hysteresis.append('A')

    else:

        hysteresis.append('N')


#print hysteresis
```

```python
'''magnitude of hysteresis is the range between the average

residuals of the rising and falling limbs (|R[i]| + |F[i]|)'''

hMagnitude = []

for i in range(0, len(ID)):

    hMagnitude.append(abs(R[i]) + abs(F[i]))


#print hMagnitude


#for i in range(0, len(ID)):

#    if (hysteresis[i] == 'C'):

#        print hMagnitude[i]

'''a table to show summary'''


x = PrettyTable()

x.add_column("ID",ID)

x.add_column("Season",seasonID)

x.add_column("Hysteresis",hysteresis)

x.add_column("Magnitude",hMagnitude)


out = open("table_hysteresis.csv", 'w+')

out.write(str(x))

out.write("\n")


#print x


'''variables'''

maxN = []#max N for each storm event

minN = []#min N for each storm event

startDischarge = []

totalRainfallmm = []

s1 = []#temp. lists

s2 = []


rainfallmm2 = rainfallmm[24:53448]

#find total rainfall for each storm event


s3 = 0

n = 1

for i in range(0, len(FinalStormID)):

    if (FinalStormID[i] == n):

        s2.append(nitratemgl1[i]) #temporary list

        s3 += rainfallmm2[i]

        if (FinalStormID[i-1] is (nan)):

            startDischarge.append(dischargem3s1[i])

        else:

            if (FinalStormID[i+1] is (nan)):

                maxN.append(max(s2))

                minN.append(min(s2))

                totalRainfallmm.append(s3)

                del s1[:]# emty the list

                del s2[:] # emty the list

                s3 = 0

                n = n + 1


#print maxN

#print len(maxN)

#print minN

#print len(minN)

#print startDischarge

#print len(startDischarge)

#print totalRainfallmm

#print len(totalRainfallmm)


'''Qp-Qo   variable'''


QpMinusQo = []

for i in range(0, len(ID)):

    QpMinusQo.append(peak[i] - startDischarge[i])


#print QpMinusQo

#print len(QpMinusQo)]


#total N flux
```

```python
Ntotal = []


n = 1
s = 0
for i in range(0, len(FinalStormID)):
    if (FinalStormID[i] == n):
        s+= (dischargem3s1[i]*nitratemgl1[i]*1800)
        if (FinalStormID[i+1] is (nan)):
            Ntotal.append(s)
            n = n + 1
            s = 0


#print Ntotal


'''Antecedent precipitation index (API)'''
DailyRainfall = []


tt = times1[0].timetuple() # same problem with
skipping the last day
currentDay = tt[2]
rain = 0
for  i in range(0, len(times1)):
    tt = times1[i].timetuple()
    d = tt[2] #day
#    print tt
    if (d != currentDay):
        DailyRainfall.append(rain)
        currentDay = d
        rain = 0
#        print currentDay
    else:
        rain += rainfallmm2[i]
DailyRainfall.append(0.8)


#print len(DailyRainfall)
#print DailyRainfall
```

```python
#take Io = 1.8 (need check) take k = 0.85 (need check)
api = Io*k+rainfall
Io =  1.8
k = 0.85
dailyAPI = []


for i in range(0, len(DailyRainfall)):
    api = Io * k + DailyRainfall[i]
    dailyAPI.append(api)
    Io = api


#print dailyAPI
#print len(dailyAPI)


#interpolate
interpAPI =[]
API = []


interpAPI = sp.interp1d(timesDaily2, dailyAPI, kind =
'linear') #linear interpolation
API = interpAPI(decDay1) #interpolation to half an
hourly values


#print API
#print len(API)


#find API for the start of each storm event
#should it be API of the first day of the storm event
instead of the interpolated value of the exact value the
storm event starts on?
stormAPI = []
EventStartTime = []    #taken from decDay1 so not a
date but a decimal number


n = 1
for i in range(0, len(FinalStormID)):
    if (FinalStormID[i] == n):
        if (FinalStormID[i-1] is (nan)):
            EventStartTime.append(decDay1[i])
```

```
        stormAPI.append(API[i])

    else:

      if (FinalStormID[i+1] is (nan)):

        n = n + 1
```

#print EventStartTime

#print len(EventStartTime)

#print stormAPI

#print len(stormAPI)

''' Antecedent temperature at 3, 5, 7, 10 and 15 days'''

'''Time since the last peak'''

PeakTime = []

#will use the decimal numbers for that instead of the day

```
for i in range(0, len(FinalHydrograph)):

   if (FinalHydrograph[i] is 'R'):

     if (FinalHydrograph[i+1] is 'f'):

        PeakTime.append(decDay1[i])
```

#print PeakTime

#print len(PeakTime)

TimeSinceLastPeak = []

TimeSinceLastPeak.append(0)

```
for i in range(1, len(PeakTime)):

   TimeSinceLastPeak.append(PeakTime[i] -
PeakTime[i-1])
```


#print TimeSinceLastPeak

#print len(TimeSinceLastPeak)

'''temperature before storm event'''

temperature = []

date = []

data = open("Morland_temperature_to_use.csv")

```
for i in range(0, 2):

   data.readline()
```

n=0

```
for line in data.readlines():

   line = line.split(',')

   tim = datetime.datetime.strptime(line[0], "%Y-%m-
%d %H:%M")

   date.append(tim)


   if (len(line[1]) > 1):

     temperature.append(float(line[1]))

   else:

     temperature.append(0.0)

     n=n+1
```

print n

sum

#print temperature

#print len(date)

#print len(temperature)

#read every second measurement to get the half hourly measurement

AirTemperature = []

AirTemperature.extend(temperature[::2])

#print temperature[::2]

#print len(temperature[::2])

print AirTemperature

#print len(AirTemperature)

'''outwrite half hourly temp'''

out = open("Morland_temp_halfHour.csv", 'w+')

out.write("Date, Temperature, N\n")

```
for i in range(0, len(times1)):

   out.write(str(times1[i]))

   out.write(",")

   out.write(str(AirTemperature[i]))

   out.write("\n")
```

'''mean air temp 3, 5, 7, 10, 15 days before storm event'''

TMean3 = []

```python
TMean5 = []

TMean7 = []

TMean10 = []

TMean15 = []

TMedian3 = []

TMedian5 = []

TMedian7 = []

TMedian10 = []

TMedian15 = []


#3days  48(measurements in a
day)*3(days)=144(measurements)

n = 1

for i in range(0, len(FinalStormID)):

    if (FinalStormID[i] == n):

      if (FinalStormID[i-1] is (nan)):

        TMean3.append(np.mean(AirTemperature[i-
144:i]))

TMedian3.append(np.median(AirTemperature[i-
144:i]))

      else:

        if (FinalStormID[i+1] is (nan)):

          n = n + 1


#print TMean3

#print len(TMean3)


#5days  48(measurements in a
day)*5(days)=240(measurements)

#start with 3rd storm event

n = 1

for i in range(0, len(FinalStormID)):

    if (FinalStormID[i] == n):

      if (FinalStormID[i-1] is (nan)):

        TMean5.append(np.mean(AirTemperature[i-
240:i]))

TMedian5.append(np.median(AirTemperature[i-
240:i]))
```

```python
      else:

        if (FinalStormID[i+1] is (nan)):

          n = n + 1

#print TMean5

#print len(TMean5)


#7days  48(measurements in a
day)*7(days)=336(measurements)

#start with 4th storm event

n = 1

for i in range(0, len(FinalStormID)):

    if (FinalStormID[i] == n):

      if (FinalStormID[i-1] is (nan)):

        TMean7.append(np.mean(AirTemperature[i-
336:i]))


TMedian7.append(np.median(AirTemperature[i-
336:i]))

      else:

        if (FinalStormID[i+1] is (nan)):

          n = n + 1

#print TMean7

#print len(TMean7)


#10days  48(measurements in a
day)*10(days)=480(measurements)

#start with 4th storm event

n = 1

for i in range(0, len(FinalStormID)):

    if (FinalStormID[i] == n):

      if (FinalStormID[i-1] is (nan)):

        TMean10.append(np.mean(AirTemperature[i-
480:i]))


TMedian10.append(np.median(AirTemperature[i-
480:i]))

      else:

        if (FinalStormID[i+1] is (nan)):

          n = n + 1

#print TMean10
```

```
#print len(TMean10)


#15days  96(measurements in a
day)*15(days)=720(measurements)

#start with 8th storm event

n = 1

for i in range(0, len(FinalStormID)):

    if (FinalStormID[i] == n):

        if (FinalStormID[i-1] is (nan)):

            TMean15.append(np.mean(AirTemperature[i-
720:i]))


TMedian15.append(np.median(AirTemperature[i-
720:i]))

        else:

            if (FinalStormID[i+1] is (nan)):

                n = n + 1

#print TMean15

#print len(TMean15)


# storm events to exclude:

#108 (See event 03/01/2013 07:00 -09/01/2013 09:00)

#126-127 (24/05/2013 13:00 - 29/05/2013 13:00)

#189 (13/01/2014 10:00 - 14/01/2014 11:30)


'''trying to express seasons as cosine wave'''

#so i would express each years measurement as a
degree which

#i then convert to cos curve

normalYrDegrees = []

leapYrDegrees = []

fullRecord = []

CosFullRecord = []


normalYrDegrees.extend(np.linspace(0, 2*pi,
num=17520)) #where did i find pi? seem to work

leapYrDegrees.extend(np.linspace(0, 2*pi,
num=17568))

#print leapYrDegrees

#print len(leapYrDegrees)
```

```
#now biuld the whole period from these

# the measurements start with a normal year,

fullRecord.extend(normalYrDegrees[12336:17520])

#print len(fullRecord)

fullRecord.extend(leapYrDegrees)

fullRecord.extend(normalYrDegrees)

fullRecord.extend(normalYrDegrees[0:13152])

#print len(fullRecord)

#print fullRecord


CosFullRecord.extend(cos(fullRecord))

#print CosFullRecord

#print len(CosFullRecord)

#plot(times1, CosFullRecord)


# find the cosine(for seasonCos) for the beginning of
each storm event

seasonCos = []


n = 1

for i in range(0, len(FinalStormID)):

    if (FinalStormID[i] == n):

        if (FinalStormID[i-1] is (nan)):

            seasonCos.append(CosFullRecord[i])

        else:

            if (FinalStormID[i+1] is (nan)):

                n = n + 1


#print seasonCos

#print len(seasonCos)


'''save the variables onto spreadsheet'''

out = open("resutls_variables.csv", 'w+')

out.write("ID, EventStartTime, TimeSinceLastPeak,
TotalRainfall, stormAPI, StartQ, PeakQ, QpMinusQo,
TotalQ, SeasonID, seasonCos, NAverageConc,
TotalNFlux, MaxNConc, MinNConc, Hysteresis,
HMagnitude, TMean3, TMean5, TMean7, TMean10,
```

```python
    TMean15, TMedian3, TMedian5, TMedian7,
TMedian10, TMedian15\n")
for i in range(0, len(ID)):
    out.write(str(ID[i]))
    out.write(",")
    out.write(str(EventStartTime[i]))
    out.write(",")
    out.write(str(TimeSinceLastPeak[i]))
    out.write(",")
    out.write(str(totalRainfallmm[i]))
    out.write(",")
    out.write(str(stormAPI[i]))
    out.write(",")
    out.write(str(startDischarge[i]))
    out.write(",")
    out.write(str(peak[i]))
    out.write(",")
    out.write(str(QpMinusQo[i]))
    out.write(",")
    out.write(str(Q[i]))
    out.write(",")
    if (seasonID[i] is 'winter'):
        out.write(str(1))
        out.write(",")
    elif (seasonID[i] is 'spring'):
        out.write(str(2))
        out.write(",")
    elif (seasonID[i] is 'summer'):
        out.write(str(3))
        out.write(",")
    elif (seasonID[i] is 'autumn'):
        out.write(str(4))
        out.write(",")
    out.write(str(seasonCos[i]))
    out.write(",")
    out.write(str(N[i]))
    out.write(",")
    out.write(str(Ntotal[i]))
    out.write(",")
    out.write(str(maxN[i]))
    out.write(",")
    out.write(str(minN[i]))
    out.write(",")
    if (hysteresis[i] is 'C'):
        out.write(str(1))
        out.write(",")
    else:
        out.write(str(-1))
        out.write(",")
    out.write(str(hMagnitude[i]))
    out.write(",")
    out.write(str(TMean3[i]))
    out.write(",")
    out.write(str(TMean5[i]))
    out.write(",")
    out.write(str(TMean7[i]))
    out.write(",")
    out.write(str(TMean10[i]))
    out.write(",")
    out.write(str(TMean15[i]))
    out.write(",")
    out.write(str(TMedian3[i]))
    out.write(",")
    out.write(str(TMedian5[i]))
    out.write(",")
    out.write(str(TMedian7[i]))
    out.write(",")
    out.write(str(TMedian10[i]))
    out.write(",")
    out.write(str(TMedian15[i]))
    out.write("\n")
    out.flush()

'''save storm events onto spreadsheet'''
```

```python
out = open("resutls_storm.csv", 'w+')


out.write("Date, StormEvent, discharge, N\n")

for i in range(0, len(times1)):

    out.write(str(times1[i]))

    out.write(",")

    out.write(str(FinalStormID[i]))

    out.write(",")

    out.write(str(FinalStormEvent[i]))

    out.write(",")

    out.write(str(nitratemgl1[i]))

    out.write("\n")


'''save the events exceeding 11.3 mg/l Nitrate events on spreadsheet '''

out = open("resutls_NAbove11_3short.csv", 'w+')

out.write("Date, StormID, NAbove11_3, Discharge\n")

for i in range(0, len(times1)):

    if (nitratemgl1[i] >= 11.3):

        out.write(str(times1[i]))

        out.write(",")

        out.write(str(FinalStormID[i]))

        out.write(",")

        out.write(str(nitratemgl1[i]))

        out.write(",")

        out.write(str(dischargem3s1[i]))

        out.write("\n")

        out.flush()


#plotStormsOnly(times1, FinalStormEvent)

#plotPTQ(times, rainfallmm, dischargem3s, nitratemgl, waterTemperatureC)

#plotStormNQ(Q, N)


''' plot rising and falling limbs of discharge/concentration(hysteresis) plots with different colours of a specific storm event by changing finalStormID number '''

n=1
```

```python
for i in range(0, 248):

    NitrR = [] #temp when plotting hysteresis for N

    DischR = [] #temp when plotting hysteresis for Q

    NitrF = [] #temp when plotting hysteresis for N

    DischF = [] #temp when plotting hysteresis for Q

    for i in range (0, len(FinalStormID)):

        if (FinalStormID[i] == n):

            if (FinalHydrograph[i] is 'R'):

                NitrR.append(nitratemgl1[i])

                DischR.append(dischargem3s1[i])

                if (FinalHydrograph[i+1] is 'f'):

                    NitrF.append(nitratemgl1[i])

                    DischF.append(dischargem3s1[i])

            else:

                NitrF.append(nitratemgl1[i])

                DischF.append(dischargem3s1[i])




    plot(DischR, NitrR,color='orange')

    plot(DischF, NitrF,color='green')

    suptitle(str(n))

    savefig("storm_"+ str(n), dpi=300)

    close()

    del NitrR[:]# emty the list

    del DischR[:] # emty the list

    del NitrF[:]# emty the list

    del DischF[:] # emty the list

    n += 1


'''quick way to plot a temporal dicharge+nitrate graph of a

 specific storm event by changing finalStormID number '''


Nitr = [] #temp when plotting hysteresis for N

Disch = []
```

```
DAte = []
for i in range (0, len(FinalStormID)):
    if (FinalStormID[i] == 96):
        Nitr.append(nitratemgl1[i])
        Disch.append(dischargem3s1[i])
        DAte.append(times1[i])



#temporaldicharge+nitrate graph
#plot(DAte, Disch, color='blue', linewidth='0.4')
#ax=gca()
#ax.set_ylabel("discharge m3 / s", color='blue')
#ax.set_xlabel("time")
#ax2=ax.twinx()
#ax2.plot(DAte, Nitr, color='red',linewidth='0.4')
#ax2.set_ylabel("nitrate",color='red')
#ax2.set_xlabel("time", color='red')


'''----------'''


#the plot of discharge and rainfall for the whole record
#plot(times, dischargem3s, color='blue',
linewidth='0.8')
#ax=gca()
#ax.set_ylabel("Discharge m3 / s", color='blue')
#ax.set_xlabel("Time")
#ax2=ax.twinx()
#ax2.bar(times, rainfallmm, color='black',
linewidth='100')
#ax2.invert_yaxis()
#ax2.set_ylabel("Rainfall",color='black')
#ax2.set_xlabel("Time")


#the plot of nitrate for the whole record
#plot(times, nitratemgl, color='red', linewidth='0.8')
#ax=gca()
#ax.set_ylabel("Nitrate mg / l", color='red')
```

```
#ax.set_xlabel("Time")
#plotStormNQ(Q, N)


'''plot the discharge with storm and non storm events
exceeding 11.3 mg NO3-N l-1 highligted '''
QR = []
TR = []
for i in range(0, len(dischargem3s1)):
    if (nitratemgl1[i] >= 11.3):
        QR.append(dischargem3s1[i])
        TR.append(times1[i])
    else:
        QR.append(nan)
        TR.append(times1[i])


#plot (times1, dischargem3s1, color='grey')
#plot (times1, FinalStormEvent, color='black')
#plot (TR, QR, color='red')
#print timesDaily
#thold = []
#for i in range(0, len(times)):
#    thold.append(11.3)
#
#subplot(211)
#xticks(rotation=45)
#tight_layout()
#bar((times), (rainfallmm), color='black', width=0.01)
#ax=gca()
#ax.invert_yaxis()
#ax.set_ylabel("rainfall/mm", color='black')
#ax.set_xlabel("time")
#
#subplot(212)
#xticks(rotation=45)
#tight_layout()
#plot((times), (nitratemgl), color='red',linewidth='0.4')
#ax=gca()
```

```python
#ax.set_xlabel("time")

#ax.set_ylabel("nitrate/mgl",color='red')

#ax.axhline(y=11.3,xmin=0,xmax=3,c='black',linewidth='0.6',zorder=0)

#ax2=ax.twinx()

#ax2.plot((times), (dischargem3s), color='blue', linewidth='0.4', alpha=0.6)

#ax2.set_ylabel("discharge/m3s", color='blue')


Anticlockwise = [] #store only data for anticlockwise hysteresis data

Clockwise = []

n = 1

j = 0

for i in range(0, len(FinalStormEvent)):

    if (FinalStormID[i] == n):

        if (hysteresis[j] is 'A'):

            Anticlockwise.append(FinalStormEvent[i])

            Clockwise.append(nan)

            if(FinalStormEvent[i+1] is (nan)):

                n = n + 1

                j = j + 1

        else:

            Anticlockwise.append(nan)

            Clockwise.append(FinalStormEvent[i])

            if(FinalStormEvent[i+1] is (nan)):

                n = n + 1

                j = j + 1

    else:

        Anticlockwise.append(nan)

        Clockwise.append(nan)


#plot(times1, Clockwise, color='green', linewidth='0.4')

#plot(times1, Anticlockwise, color='orange', linewidth='0.4')

#print len(FinalStormEvent)

#print len(Anticlockwise)
```

Code for factor analysis

```python
# -*- coding: utf-8 -*-
"""Created on Tue Jul 05 19:34:07 2016
@author: Vaida
"""
from matplotlib.pylab import *
import scipy.interpolate as sp


def calcNitrate(Q, g):
    dataFile = open('P_series_2050.txt')
    rainfall = []


#Skip over the header
    for i in range(0, 2):
        dataFile.readline()


#Read in the data into 2D arrays,
    for line in dataFile.readlines():
        line = line.split('          ')
        rainfall.append(float(line[g+1]))
#    print len(rainfall)
#    print rainfall
#    day number
    day = []
    n = 0
    for i in range(0, 10957):
        day.append(n)
        n += 1
#    print len(day)
#    print day
    '''calculating baseflow'''
    '''cant find daily minimums as values are daily so will jump to
    finding 3 day minimums'''
    localMin = []
    for i in range(1, len(Q)-1):
        minValue = min(Q[i-1], Q[i], Q[i+1])
        localMin.append(minValue)
#    print len(Q)
#    print len(localMin)
    '''to make day and localMin arrays the same length'''
    day2 = day[1:]
    day2.pop()
#    print day2
#    print len(day2)


    '''interpolate to half hourly  values for code to work better?'''
    #decimal day for interpolation
    decDay = []
    n=1
    for i in range(0, 525793):
        decDay.append(n)
        n+=0.0208333333299999
#    print decDay
#    print len(decDay)
    '''linear interpolation'''
    baseFlow = []
    base = sp.interp1d(day2, localMin, kind='linear')
    baseFlow = base(decDay)


    '''shorten the discharge array to be able to find the throughflow'''
    Q2 = Q[1:]
    Q2.pop()
#    plot (Q2)
#    plot (baseFlow, color = 'red')
    '''interpolate discharge'''
    QInt = []
    Qinterpolation = sp.interp1d(day2, Q2, kind='linear')
    QInt = Qinterpolation(decDay)


#    consider making a for loop to append variable into a list?
```

```
'''describe storm events'''                          n = 0
stormEvent = []                                      for i in range(0, len(stormID)):
for i in range(0, len(decDay)):                          if (stormID[i] == n):
    if (QInt[i] > baseFlow[i]):                              j.append(QInt[i])
        stormEvent.append(QInt[i])                           jj.append(nan)
    else:                                                    if (stormID[i+1] is (nan)):
        stormEvent.append(nan)                                   maxDischarge = max(j)
```
\#   plot (QInt)

\#   plot (stormEvent)

\#   plot (baseFlow, color = 'red')

\#   plot (localMin, color = 'red')
```
                                                                 minDischarge = min(j)
                                                                 n = n + 1
```
'''assign ID to storm events'''
```
stormID = []
n=0
for i in range(0, len(stormEvent)-1):
    if (stormEvent[i] is not (nan)):
        stormID.append(n)
        if (stormEvent[i+1] is (nan)):
            n=n+1
    else:
        stormID.append(nan)
```

```
                                                                 if ((jj.count(nan) < 5) or (maxDischarge <
                                                         minDischarge*1.5)): #or (maxDischarge/2 <
                                                         minDischarge)#count the number of measurements in
                                                         storm event and append j (all nan) if number is less
                                                         than 5

                                                                     stormEvent2.extend(jj) #the storm event
                                                         replaced by nan

                                                                     del j[:] #empty the lists

                                                                     del jj[:]

                                                                 else:

                                                                     stormEvent2.extend(j) #else append the
                                                         storm event as usual

                                                                     del j[:] #empty the lists

                                                                     del jj[:]

                                                             else:

                                                                 stormEvent2.append(nan)
```

```
stormID.append(nan) # to make up for lost last
measurement in list
```

\#   print len(stormEvent2)

'''seperate storm events further, anything that falls 1/4
below'''

\#   print len(stormID)

```
hydrograph = []
hydrograph2 = []
```

'''better description of storm events thestorm has to
have more than

5 measurements and the peak must rise at least 50%
above min flow'''
```
                                                         for i in range (0, len(stormEvent2)):
                                                             if (stormEvent2[i] is not nan):
                                                                 if (stormEvent2[i] < stormEvent2[i-1]):
                                                                     hydrograph.append('f')
                                                                 else:
                                                                     hydrograph.append('R')
```
```
j = []    #temp list
jj = []   #temp list
stormEvent2 = [] # a list with storm events that have
less than 5 measurements and peaks that risen less than
50% removed
```

```python
        else:

            hydrograph.append(nan)


    '''re-assign ID to storm events'''


    stormID2 = []
    n=0
    for i in range(0, len(stormEvent2)):
        if (stormEvent2[i] is not (nan)):
            stormID2.append(n)
            if (stormEvent2[i+1] is (nan)):
                n=n+1
        else:
            stormID2.append(nan)


#print stormID2
# now all the measurements that are equal to falling
measurement are still marked as R

# need to change R's that are equal to the f's into f's


    n = 0
    fall = 0
    for i in range(0, len(stormID2)):
        if (stormID2[i] == n):
            if (hydrograph[i] == 'f'):
                hydrograph2.append('f')
                fall = stormEvent2[i] #so the last 'f' before 'R'
is memorised
                if (stormID2[i+1] is (nan)):
                    n = n + 1
                    fall = 0
            else:
                if (stormEvent2[i] == fall): #now this
memorised 'f' is compared to 'R' and if it is == then 'R'
is replaced with 'f'
                    hydrograph2.append('f')
                    if (stormID2[i+1] is (nan)):
                        n = n + 1
                        fall = 0
                else:
                    hydrograph2.append('R')
                    if (stormID2[i+1] is (nan)):
                        n = n + 1
                        fall = 0
        else:
            hydrograph2.append(nan)
#print hydrograph2
#print len(hydrograph2)
# now all the R's and f's are correct
#need to decide which rises/falls are a beggining of a
new storm event or just a part of a complex storm event
    hydrograph3 = []
    hydrograph4 = []
    k1 = [] #temp lists
    k2 = []
    k3 = []
    k4 = []
    rise = 0
    fall = 0
#checking rises
    for i in range (0, len(hydrograph2)): #check if each
rise that isnt the main peak is at least 1/3 (1/4?) above
the fall (except for the final rise which will be always
marked as a fall) (later will fill in the gaps or seperate
storm events)
        if (hydrograph2[i] is not (nan)):
            if (hydrograph2[i] is 'R'):
                k1.append(stormEvent2[i])
                k2.append('f')
                k3.append('R')
                if (hydrograph2[i+1] is ('f')):
                    if (max(k1) >= rise):      #all the R's before
the main peak and including the main peak are rises
                        hydrograph3.extend(k3)
                        rise = max(k1)
                        del k1[:] #empty the lists
                        del k2[:]
```

97

```
                del k3[:]

            else:

                if (max(k1) >= (fall/3)*4): #old was
(max(k1)/3*2 >= fall) #is this correct? check# see if the
max of rise that is not the main peak is 1/3 (1/4?) more
than fall min

                    hydrograph3.extend(k3)

                    del k1[:] #empty the lists

                    del k2[:]

                    del k3[:]

                else:

                    hydrograph3.extend(k2)

                    del k1[:] #empty the lists

                    del k2[:]

                    del k3[:]

        else:

            if (hydrograph2[i+1] is (nan)): #so the last
rise in the storm event will always be marked as a fall

                hydrograph3.extend(k2)

                rise = 0

                fall = 0

                del k1[:] #empty the lists

                del k2[:]

                del k3[:]

    else:

        hydrograph3.append(hydrograph2[i])# or 'f'

        k4.append(stormEvent2[i])

        if (hydrograph2[i+1] is ('R')):

            fall = min(k4)

            del k4[:]

        else:

            if (hydrograph2[i+1] is (nan)):

                rise = 0

                fall = 0

                del k4[:]

    else:

        hydrograph3.append(nan)
```

```
#print hydrograph3

#print len(hydrograph3)


#check fall

    rise = 0

#check if each fall is at least 1/3 (1/4?) below the
previous rise (except for the final fall which will be
always marked as a fall) (later will fill in the gaps or
seperate storm events)

    for i in range (0, len(hydrograph3)):

        if (hydrograph3[i] is not (nan)):

            if (hydrograph3[i] is 'f'):

                k1.append(stormEvent2[i])

                k2.append('f')

                k3.append('R')

                if (hydrograph3[i+1] is ('R')):

                    if (min(k1) <= (rise/3)*2):       #all the R's
before the main peak and including the main peak are
rises

                        hydrograph4.extend(k2)

                        del k1[:] #empty the lists

                        del k2[:]

                        del k3[:]

                    else:

                        hydrograph4.extend(k3)

                        del k1[:] #empty the lists

                        del k2[:]

                        del k3[:]

                else:

                    if (hydrograph3[i+1] is (nan)): #so the last
rise in the storm event will always be marked as a fall

                        hydrograph4.extend(k2)

                        rise = 0

                        del k1[:] #empty the lists

                        del k2[:]

                        del k3[:]

            else:

                hydrograph4.append(hydrograph3[i])# or 'R'

                k4.append(stormEvent2[i])
```

```python
            if (hydrograph3[i+1] is ('f')):

                rise = max(k4)

                del k4[:]

            else:

                if (hydrograph3[i+1] is (nan)):

                    rise = 0

                    del k4[:]

        else:

            hydrograph4.append(nan)


#print hydrograph4

#print len(hydrograph4)


#final hydrograph with nan insesarted to seperate storm
events

#final stormEvent with nan insesarted to seperate storm
events

    FinalStormEvent = []

    FinalHydrograph = []

    for i in range(0, len(stormEvent2)):

        if (hydrograph4[i] is not (nan)):

            if (hydrograph4[i] is 'R'):

                FinalStormEvent.append(stormEvent2[i])

                FinalHydrograph.append(hydrograph4[i])#or
'R'

            else:

                if (hydrograph4[i+1] is 'R'):

                    FinalStormEvent.append(nan)

                    FinalHydrograph.append(nan)

                else:

                    FinalStormEvent.append(stormEvent2[i])

FinalHydrograph.append(hydrograph4[i])#or 'f'

        else:

            FinalStormEvent.append(nan)

            FinalHydrograph.append(hydrograph4[i])#or
'nan'


#print FinalStormEvent
```

```python
#print len(FinalStormEvent)

#   plot (Q2)

#   plot (FinalStormEvent, color = 'red')


    '''re-assign ID to storm events'''


    FinalStormID = []


    n=0

    for i in range(0, len(FinalStormEvent)):

        if (FinalStormEvent[i] is not (nan)):

            FinalStormID.append(n)

            if (FinalStormEvent[i+1] is (nan)):

                n=n+1

        else:

            FinalStormID.append(nan)


#print FinalStormID #was 110 events, after aditional
1:3 fall split its 229


    '''04/06/16....check if there are storm events
contaning less than 5 data points again'''

    LastCount = [] #consider removing if less than 5

    n = 0

    j = 0 #to count data points within storm event


    for i in range(0, len(FinalStormID)):

        if (FinalStormID[i] == n):

            j = j + 1

            if (FinalStormID[i+1] is (nan)):

                LastCount.append(j)

                n = n + 1

        else:

            j = 0


#print len(LastCount)

#print min(LastCount)
```

```python
    '''Peak for variables'''
    numberOfStorms = []
    h = [] #temp Q2
    peak = [] #max flow of each storm event
    n = 0
    for i in range(0, len(FinalStormID)):
        if (FinalStormID[i] == n):
#           print str(stormID[i]) + " " + str(n)
            h.append(QInt[i]) #temporary list
            if (FinalStormID[i+1] is (nan)):
                peak.append(max(h)) #find a maximum
discharge to later find the falling and rising limb of
hydrograph
                numberOfStorms.append(n)
                del h[:] # emty the list
                n = n + 1


    '''variables'''
    startDischarge = []
    totalRainfall = []


    rainfall2 = rainfall[1:]
    rainfall2.pop()


    '''interpolate rainfall'''
    PInt = []
    Pinterpolation = sp.interp1d(day2, rainfall2,
kind='linear')
    PInt = Pinterpolation(decDay)


    #find total rainfall for each storm event


    s3 = 0 # for total rainfall
    n = 0
    for i in range(0, len(FinalStormID)):
        if (FinalStormID[i] == n):
            s3 += PInt[i]
```

```python
            if (FinalStormID[i-1] is (nan)):
                startDischarge.append(QInt[i])
            else:
                if (FinalStormID[i+1] is (nan)):
                    totalRainfall.append(s3)
                    s3 = 0
                    n = n + 1


    '''Qp-Qo   variable'''


    QpMinusQo = []
    for i in range(0, len(numberOfStorms)):
        QpMinusQo.append(peak[i] - startDischarge[i])


#print QpMinusQo
#print len(QpMinusQo)]
#total N flux


    '''Time since the last peak'''
    PeakTime = []
    n = 0
    for i in range(0, len(FinalStormID)):
        if (FinalStormID[i] == n):
            if (QInt[i] == peak[n]):
                PeakTime.append(decDay[i])
#               print PeakTime
                n = n + 1


#   print len (numberOfStorms)
#   print len (PeakTime)


    TimeSinceLastPeak = []
    TimeSinceLastPeak.append(0)
    for i in range(1, len(PeakTime)):
        TimeSinceLastPeak.append(PeakTime[i] -
PeakTime[i-1])
#   print len(numberOfStorms)
```

```python
#   print len(totalRainfall)

#   print len(TimeSinceLastPeak)

#   print len(QpMinusQo)

#print TimeSinceLastPeak

#print len(TimeSinceLastPeak)

    NFlux = []

    for i in range(0, len(numberOfStorms)):

        NFlux.append(22241.968 +
49948.389*(totalRainfall[i]) -
35197.094*(TimeSinceLastPeak[i]) +
92306.823*(QpMinusQo[i]))


    return sum(NFlux)


'''Start'''

dataFile = open('BatchQsim_2050.txt')

flowm3s = zeros([3000, 10957])


#Skip over the header

for i in range(0, 1):

    dataFile.readline()


#Read in the data into 2D arrays,

i = 0

for line in dataFile.readlines():

    line = line.split(' ')

    for n in range (0, 3000):

        flowm3s[n][i] = (float(line[n+2]))

#   print i,

#   print flowm3s[0][i]

    i += 1

#   print i


#create a 1D Q array

SumNF = []

for j in range (0,3000):

    Q = []

    P = []
```

```python
#   to get the row number for P as 30 lines

#   in Q will have to read same one line in P

    g = int(round(j/30))

#   print g

    for i in range(0,10957):

        Q.append(flowm3s[j][i])

    SumNF.append(calcNitrate(Q, g))

#   print SumNF

    print len(SumNF)


out = open("SumNF_2050.txt", 'w+')

out.write("SumNF\n")

for i in range(0, 3000):

    out.write(str(SumNF[i]))

    out.write("\n")

    out.flush()


# -*- coding: utf-8 -*-

"""Created on Tue Jul 05 19:34:07 2016

@author: Vaida
"""

from matplotlib.pylab import *

import scipy.interpolate as sp


def calcNitrate(Q, g):

    dataFile = open('P_series_2080.txt')

    rainfall = []


#Skip over the header

    for i in range(0, 2):

        dataFile.readline()


#Read in the data into 2D arrays,

    for line in dataFile.readlines():

        line = line.split('          ')

        rainfall.append(float(line[g+1]))

#   print len(rainfall)
```

```python
#   print rainfall
#   day number
    day = []
    n = 0
    for i in range(0, 10957):
        day.append(n)
        n += 1
#   print len(day)
#   print day
    '''calculating baseflow'''
    '''cant find daily minimums as values are daily so will jump to
    finding 3 day minimums'''
    localMin = []
    for i in range(1, len(Q)-1):
        minValue = min(Q[i-1], Q[i], Q[i+1])
        localMin.append(minValue)
#   print len(Q)
#   print len(localMin)
    '''to make day and localMin arrays the same length'''
    day2 = day[1:]
    day2.pop()
#   print day2
#   print len(day2)


    '''interpolate to half hourly  values for code to work better?'''


    #decimal day for interpolation
    decDay = []
    n=1
    for i in range(0, 525793):
        decDay.append(n)
        n+=0.0208333333299999
#   print decDay
#   print len(decDay)
    '''linear interpolation'''
```

```python
    baseFlow = []
    base = sp.interp1d(day2, localMin, kind='linear')
    baseFlow = base(decDay)


    '''shorten the discharge array to be able to find the throughflow'''
    Q2 = Q[1:]
    Q2.pop()
#   plot (Q2)
#   plot (baseFlow, color = 'red')
    '''interpolate discharge'''
    QInt = []
    Qinterpolation = sp.interp1d(day2, Q2, kind='linear')
    QInt = Qinterpolation(decDay)


#   consider making a for loop to append variable into a list?
    '''describe storm events'''
    stormEvent = []
    for i in range(0, len(decDay)):
        if (QInt[i] > baseFlow[i]):
            stormEvent.append(QInt[i])
        else:
            stormEvent.append(nan)
#   plot (QInt)
#   plot (stormEvent)
#   plot (baseFlow, color = 'red')
#   plot (localMin, color = 'red')


    '''assign ID to storm events'''
    stormID = []
    n=0
    for i in range(0, len(stormEvent)-1):
        if (stormEvent[i] is not (nan)):
            stormID.append(n)
            if (stormEvent[i+1] is (nan)):
                n=n+1
```

```python
        else:

            stormID.append(nan)


    stormID.append(nan) # to make up for lost last measurement in list

#   print len(stormID)


    '''better description of storm events thestorm has to have more than

    5 measurements and the peak must rise at least 50% above min flow'''


    j = []     #temp list

    jj = []    #temp list

    stormEvent2 = [] # a list with storm events that have less than 5 measurements and peaks that risen less than 50% removed


    n = 0

    for i in range(0, len(stormID)):

        if (stormID[i] == n):

            j.append(QInt[i])

            jj.append(nan)

            if (stormID[i+1] is (nan)):

                maxDischarge = max(j)

                minDischarge = min(j)

                n = n + 1

                if ((jj.count(nan) < 5) or (maxDischarge < minDischarge*1.5)): #or (maxDischarge/2 < minDischarge)#count the number of measurements in storm event and append j (all nan) if number is less than 5

                    stormEvent2.extend(jj) #the storm event replaced by nan

                    del j[:] #empty the lists

                    del jj[:]

                else:

                    stormEvent2.extend(j) #else append the storm event as usual

                    del j[:] #empty the lists

                    del jj[:]
```

```python
        else:

            stormEvent2.append(nan)


#   print len(stormEvent2)

    '''seperate storm events further, anything that falls 1/4 below'''

    hydrograph = []

    hydrograph2 = []


    for i in range (0, len(stormEvent2)):

        if (stormEvent2[i] is not nan):

            if (stormEvent2[i] < stormEvent2[i-1]):

                hydrograph.append('f')

            else:

                hydrograph.append('R')

        else:

            hydrograph.append(nan)


    '''re-assign ID to storm events'''


    stormID2 = []

    n=0

    for i in range(0, len(stormEvent2)):

        if (stormEvent2[i] is not (nan)):

            stormID2.append(n)

            if (stormEvent2[i+1] is (nan)):

                n=n+1

        else:

            stormID2.append(nan)


#print stormID2

# now all the measurements that are equal to falling measurement are still marked as R

# need to change R's that are equal to the f's into f's


    n = 0

    fall = 0
```

```python
    for i in range(0, len(stormID2)):

        if (stormID2[i] == n):

            if (hydrograph[i] == 'f'):

                hydrograph2.append('f')

                fall = stormEvent2[i] #so the last 'f' before 'R'
is memorised

                if (stormID2[i+1] is (nan)):

                    n = n + 1

                    fall = 0

            else:

                if (stormEvent2[i] == fall): #now this
memorised 'f' is compared to 'R' and if it is == then 'R'
is replaced with 'f'

                    hydrograph2.append('f')

                    if (stormID2[i+1] is (nan)):

                        n = n + 1

                        fall = 0

                else:

                    hydrograph2.append('R')

                    if (stormID2[i+1] is (nan)):

                        n = n + 1

                        fall = 0

        else:

            hydrograph2.append(nan)


#print hydrograph2

#print len(hydrograph2)

# now all the R's and f's are correct

#need to decide which rises/falls are a beggining of a
new storm event or just a part of a complex storm event



    hydrograph3 = []

    hydrograph4 = []

    k1 = [] #temp lists

    k2 = []

    k3 = []

    k4 = []
```

```python
    rise = 0

    fall = 0

#checking rises

    for i in range (0, len(hydrograph2)): #check if each
rise that isnt the main peak is at least 1/3 (1/4?) above
the fall (except for the final rise which will be always
marked as a fall) (later will fill in the gaps or seperate
storm events)

        if (hydrograph2[i] is not (nan)):

            if (hydrograph2[i] is 'R'):

                k1.append(stormEvent2[i])

                k2.append('f')

                k3.append('R')

                if (hydrograph2[i+1] is ('f')):

                    if (max(k1) >= rise):      #all the R's before
the main peak and including the main peak are rises

                        hydrograph3.extend(k3)

                        rise = max(k1)

                        del k1[:] #empty the lists

                        del k2[:]

                        del k3[:]

                    else:

                        if (max(k1) >= (fall/3)*4): #old was
(max(k1)/3*2 >= fall) #is this correct? check# see if the
max of rise that is not the main peak is 1/3 (1/4?) more
than fall min

                            hydrograph3.extend(k3)

                            del k1[:] #empty the lists

                            del k2[:]

                            del k3[:]

                        else:

                            hydrograph3.extend(k2)

                            del k1[:] #empty the lists

                            del k2[:]

                            del k3[:]

                else:

                    if (hydrograph2[i+1] is (nan)): #so the last
rise in the storm event will always be marked as a fall

                        hydrograph3.extend(k2)

                        rise = 0
```

```
                fall = 0

                del k1[:] #empty the lists

                del k2[:]

                del k3[:]

        else:

            hydrograph3.append(hydrograph2[i])# or 'f'

            k4.append(stormEvent2[i])

            if (hydrograph2[i+1] is ('R')):

                fall = min(k4)

                del k4[:]

            else:

                if (hydrograph2[i+1] is (nan)):

                    rise = 0

                    fall = 0

                    del k4[:]

        else:

            hydrograph3.append(nan)


#print hydrograph3

#print len(hydrograph3)


#check fall

    rise = 0

    for i in range (0, len(hydrograph3)): #check if each
fall is at least 1/3 (1/4?) below the previous rise (except
for the final fall which will be always marked as a fall)
(later will fill in the gaps or seperate storm events)

        if (hydrograph3[i] is not (nan)):

            if (hydrograph3[i] is 'f'):

                k1.append(stormEvent2[i])

                k2.append('f')

                k3.append('R')

                if (hydrograph3[i+1] is ('R')):

                    if (min(k1) <= (rise/3)*2):      #all the R's
before the main peak and including the main peak are
rises

                        hydrograph4.extend(k2)

                        del k1[:] #empty the lists

                        del k2[:]
```

```
                    del k3[:]

                else:

                    hydrograph4.extend(k3)

                    del k1[:] #empty the lists

                    del k2[:]

                    del k3[:]

            else:

                if (hydrograph3[i+1] is (nan)): #so the last
rise in the storm event will always be marked as a fall

                    hydrograph4.extend(k2)

                    rise = 0

                    del k1[:] #empty the lists

                    del k2[:]

                    del k3[:]

        else:

            hydrograph4.append(hydrograph3[i])# or 'R'

            k4.append(stormEvent2[i])

            if (hydrograph3[i+1] is ('f')):

                rise = max(k4)

                del k4[:]

            else:

                if (hydrograph3[i+1] is (nan)):

                    rise = 0

                    del k4[:]

        else:

            hydrograph4.append(nan)


#print hydrograph4

#print len(hydrograph4)


#final hydrograph with nan insesarted to seperate storm
events

#final stormEvent with nan insesarted to seperate storm
events

    FinalStormEvent = []

    FinalHydrograph = []

    for i in range(0, len(stormEvent2)):

        if (hydrograph4[i] is not (nan)):
```

105

```
if (hydrograph4[i] is 'R'):

    FinalStormEvent.append(stormEvent2[i])

    FinalHydrograph.append(hydrograph4[i])#or
'R'

else:

    if (hydrograph4[i+1] is 'R'):

        FinalStormEvent.append(nan)

        FinalHydrograph.append(nan)

    else:

        FinalStormEvent.append(stormEvent2[i])


FinalHydrograph.append(hydrograph4[i])#or 'f'

    else:

        FinalStormEvent.append(nan)

        FinalHydrograph.append(hydrograph4[i])#or
'nan'


#print FinalStormEvent

#print len(FinalStormEvent)

#    plot (Q2)

#    plot (FinalStormEvent, color = 'red')


    '''re-assign ID to storm events'''

    FinalStormID = []


    n=0

    for i in range(0, len(FinalStormEvent)):

        if (FinalStormEvent[i] is not (nan)):

            FinalStormID.append(n)

            if (FinalStormEvent[i+1] is (nan)):

                n=n+1

        else:

            FinalStormID.append(nan)


#print FinalStormID #was 110 events, after aditional
1:3 fall split its 229


    '''04/06/16....check if there are storm events
contaning less than 5 data points again'''
```

```
LastCount = [] #consider removing if less than 5

n = 0

j = 0 #to count data points within storm event


for i in range(0, len(FinalStormID)):

    if (FinalStormID[i] == n):

        j = j + 1

        if (FinalStormID[i+1] is (nan)):

            LastCount.append(j)

            n = n + 1

    else:

        j = 0

#print len(LastCount)

#print min(LastCount)


    '''Peak for variables'''

    numberOfStorms = []

    h = [] #temp Q2

    peak = [] #max flow of each storm event

    n = 0

    for i in range(0, len(FinalStormID)):

        if (FinalStormID[i] == n):

#        print str(stormID[i]) + " " + str(n)

            h.append(QInt[i]) #temporary list

            if (FinalStormID[i+1] is (nan)):

                peak.append(max(h)) #find a maximum
discharge to later find the falling and rising limb of
hydrograph

                numberOfStorms.append(n)

                del h[:] # emty the list

                n = n + 1


    '''variables'''

    startDischarge = []

    totalRainfall = []
```

```python
    rainfall2 = rainfall[1:]

    rainfall2.pop()


    '''interpolate rainfall'''

    PInt = []

    Pinterpolation = sp.interp1d(day2, rainfall2,
kind='linear')

    PInt = Pinterpolation(decDay)


    #find total rainfall for each storm event

    s3 = 0 # for total rainfall

    n = 0

    for i in range(0, len(FinalStormID)):

        if (FinalStormID[i] == n):

            s3 += PInt[i]

            if (FinalStormID[i-1] is (nan)):

                startDischarge.append(QInt[i])

            else:

                if (FinalStormID[i+1] is (nan)):

                    totalRainfall.append(s3)

                    s3 = 0

                    n = n + 1


    '''Qp-Qo   variable'''

    QpMinusQo = []

    for i in range(0, len(numberOfStorms)):

        QpMinusQo.append(peak[i] - startDischarge[i])

#print QpMinusQo

#print len(QpMinusQo)]

#total N flux


    '''Time since the last peak'''

    PeakTime = []

    n = 0

    for i in range(0, len(FinalStormID)):

        if (FinalStormID[i] == n):

            if (QInt[i] == peak[n]):

                PeakTime.append(decDay[i])

#            print PeakTime

                n = n + 1


#   print len (numberOfStorms)

#   print len (PeakTime)


    TimeSinceLastPeak = []

    TimeSinceLastPeak.append(0)

    for i in range(1, len(PeakTime)):

        TimeSinceLastPeak.append(PeakTime[i] -
PeakTime[i-1])

#   print len(numberOfStorms)

#   print len(totalRainfall)

#   print len(TimeSinceLastPeak)

#   print len(QpMinusQo)

#print TimeSinceLastPeak

#print len(TimeSinceLastPeak)

    NFlux = []

    for i in range(0, len(numberOfStorms)):

        NFlux.append(22241.968 +
49948.389*(totalRainfall[i]) -
35197.094*(TimeSinceLastPeak[i]) +
92306.823*(QpMinusQo[i]))

    return sum(NFlux)


'''Start'''

dataFile = open('BatchQsim_2080.txt')

flowm3s = zeros([3000, 10957])


#Skip over the header

for i in range(0, 1):

    dataFile.readline()


#Read in the data into 2D arrays,

i = 0

for line in dataFile.readlines():

    line = line.split(' ')
```

```python
        for n in range (0, 3000):
            flowm3s[n][i] = (float(line[n+2]))
#    print i,
#    print flowm3s[0][i]
        i += 1
#    print i


#create a 1D Q array
SumNF = []
for j in range (0,3000):
    Q = []
    P = []
#    to get the row number for P as 30 lines
#    in Q will have to read same one line in P
    g = int(round(j/30))
#    print g
    for i in range(0,10957):
        Q.append(flowm3s[j][i])
    SumNF.append(calcNitrate(Q, g))
#    print SumNF
    print len(SumNF)


out = open("SumNF_2080.txt", 'w+')
out.write("SumNF\n")
for i in range(0, 3000):
    out.write(str(SumNF[i]))
    out.write("\n")
    out.flush()


# -*- coding: utf-8 -*-
"""Created on Tue Jul 05 19:34:07 2016
@author: Vaida
"""
from matplotlib.pylab import *
import scipy.interpolate as sp


def calcNitrate(Q, g):
```

```python
    dataFile = open('P_series_Baseline.txt')
    rainfall = []


#Skip over the header
    for i in range(0, 2):
        dataFile.readline()


#Read in the data into 2D arrays,
    for line in dataFile.readlines():
        line = line.split('          ')
        rainfall.append(float(line[g+1]))
#    print len(rainfall)
#    print rainfall
#    day number
    day = []
    n = 0
    for i in range(0, 10957):
        day.append(n)
        n += 1
#    print len(day)
#    print day
    '''calculating baseflow'''
    '''cant find daily minimums as values are daily so will jump to
    finding 3 day minimums'''
    localMin = []
    for i in range(1, len(Q)-1):
        minValue = min(Q[i-1], Q[i], Q[i+1])
        localMin.append(minValue)
#    print len(Q)
#    print len(localMin)
    '''to make day and localMin arrays the same length'''
    day2 = day[1:]
    day2.pop()
#    print day2
#    print len(day2)
```

```
    '''interpolate to half hourly  values for code to work
better?'''


    #decimal day for interpolation

    decDay = []

    n=1

    for i in range(0, 525793):

        decDay.append(n)

        n+=0.0208333333299999

#   print decDay

#   print len(decDay)

    '''linear interpolation'''

    baseFlow = []

    base = sp.interp1d(day2, localMin, kind='linear')

    baseFlow = base(decDay)


    '''shorten the discharge array to be able to find the
throughflow'''

    Q2 = Q[1:]

    Q2.pop()

#   plot (Q2)

#   plot (baseFlow, color = 'red')

    '''interpolate discharge'''

    QInt = []

    Qinterpolation = sp.interp1d(day2, Q2, kind='linear')

    QInt = Qinterpolation(decDay)


#   consider making a for loop to append variable into
a list?


    '''describe storm events'''

    stormEvent = []

    for i in range(0, len(decDay)):

        if (QInt[i] > baseFlow[i]):

            stormEvent.append(QInt[i])

        else:

            stormEvent.append(nan)

#   plot (QInt)
```

```
#   plot (stormEvent)

#   plot (baseFlow, color = 'red')

#   plot (localMin, color = 'red')


    '''assign ID to storm events'''

    stormID = []

    n=0

    for i in range(0, len(stormEvent)-1):

        if (stormEvent[i] is not (nan)):

            stormID.append(n)

            if (stormEvent[i+1] is (nan)):

                n=n+1

        else:

            stormID.append(nan)


    stormID.append(nan) # to make up for lost last
measurement in list


#   print len(stormID)


    '''better description of storm events thestorm has to
have more than

    5 measurements and the peak must rise at least 50%
above min flow'''

    j = []   #temp list

    jj = []   #temp list

    stormEvent2 = [] # a list with storm events that have
less than 5 measurements and peaks that risen less than
50% removed


    n = 0

    for i in range(0, len(stormID)):

        if (stormID[i] == n):

            j.append(QInt[i])

            jj.append(nan)

            if (stormID[i+1] is (nan)):

                maxDischarge = max(j)

                minDischarge = min(j)

                n = n + 1
```

109

```
            if ((jj.count(nan) < 5) or (maxDischarge <
minDischarge*1.5)): #or (maxDischarge/2 <
minDischarge)#count the number of measurements in
storm event and append j (all nan) if number is less
than 5

                stormEvent2.extend(jj) #the storm event
replaced by nan

                del j[:] #empty the lists

                del jj[:]

            else:

                stormEvent2.extend(j) #else append the
storm event as usual

                del j[:] #empty the lists

                del jj[:]

        else:

            stormEvent2.append(nan)


#    print len(stormEvent2)

    '''seperate storm events further, anything that falls 1/4
below'''


    hydrograph = []

    hydrograph2 = []


    for i in range (0, len(stormEvent2)):

        if (stormEvent2[i] is not nan):

            if (stormEvent2[i] < stormEvent2[i-1]):

                hydrograph.append('f')

            else:

                hydrograph.append('R')

        else:

            hydrograph.append(nan)


    '''re-assign ID to storm events'''


    stormID2 = []

    n=0

    for i in range(0, len(stormEvent2)):

        if (stormEvent2[i] is not (nan)):

            stormID2.append(n)
```

```
            if (stormEvent2[i+1] is (nan)):

                n=n+1

        else:

            stormID2.append(nan)


#print stormID2

# now all the measurements that are equal to falling
measurement are still marked as R

# need to change R's that are equal to the f's into f's


    n = 0

    fall = 0

    for i in range(0, len(stormID2)):

        if (stormID2[i] == n):

            if (hydrograph[i] == 'f'):

                hydrograph2.append('f')

                fall = stormEvent2[i] #so the last 'f' before 'R'
is memorised

                if (stormID2[i+1] is (nan)):

                    n = n + 1

                    fall = 0

            else:

                if (stormEvent2[i] == fall): #now this
memorised 'f' is compared to 'R' and if it is == then 'R'
is replaced with 'f'

                    hydrograph2.append('f')

                    if (stormID2[i+1] is (nan)):

                        n = n + 1

                        fall = 0

                else:

                    hydrograph2.append('R')

                    if (stormID2[i+1] is (nan)):

                        n = n + 1

                        fall = 0

        else:

            hydrograph2.append(nan)


#print hydrograph2
```

```
#print len(hydrograph2)

# now all the R's and f's are correct

#need to decide which rises/falls are a beggining of a
new storm event or just a part of a complex storm event


    hydrograph3 = []

    hydrograph4 = []

    k1 = [] #temp lists

    k2 = []

    k3 = []

    k4 = []

    rise = 0

    fall = 0

#checking rises

    for i in range (0, len(hydrograph2)): #check if each
rise that isnt the main peak is at least 1/3 (1/4?) above
the fall (except for the final rise which will be always
marked as a fall) (later will fill in the gaps or seperate
storm events)

        if (hydrograph2[i] is not (nan)):

            if (hydrograph2[i] is 'R'):

                k1.append(stormEvent2[i])

                k2.append('f')

                k3.append('R')

                if (hydrograph2[i+1] is ('f')):

                    if (max(k1) >= rise):     #all the R's before
the main peak and including the main peak are rises

                        hydrograph3.extend(k3)

                        rise = max(k1)

                        del k1[:] #empty the lists

                        del k2[:]

                        del k3[:]

                    else:

                        if (max(k1) >= (fall/3)*4): #old was
(max(k1)/3*2 >= fall) #is this correct? check# see if the
max of rise that is not the main peak is 1/3 (1/4?) more
than fall min

                            hydrograph3.extend(k3)

                            del k1[:] #empty the lists

                            del k2[:]
```

```
                            del k3[:]

                        else:

                            hydrograph3.extend(k2)

                            del k1[:] #empty the lists

                            del k2[:]

                            del k3[:]

                else:

                    if (hydrograph2[i+1] is (nan)): #so the last
rise in the storm event will always be marked as a fall

                        hydrograph3.extend(k2)

                        rise = 0

                        fall = 0

                        del k1[:] #empty the lists

                        del k2[:]

                        del k3[:]

            else:

                hydrograph3.append(hydrograph2[i])# or 'f'

                k4.append(stormEvent2[i])

                if (hydrograph2[i+1] is ('R')):

                    fall = min(k4)

                    del k4[:]

                else:

                    if (hydrograph2[i+1] is (nan)):

                        rise = 0

                        fall = 0

                        del k4[:]

        else:

            hydrograph3.append(nan)


#print hydrograph3

#print len(hydrograph3)


#check fall

    rise = 0

    for i in range (0, len(hydrograph3)): #check if each
fall is at least 1/3 (1/4?) below the previous rise (except
for the final fall which will be always marked as a fall)
(later will fill in the gaps or seperate storm events)
```

111

```python
        if (hydrograph3[i] is not (nan)):

            if (hydrograph3[i] is 'f'):

                k1.append(stormEvent2[i])

                k2.append('f')

                k3.append('R')

                if (hydrograph3[i+1] is ('R')):

                    if (min(k1) <= (rise/3)*2):      #all the R's
before the main peak and including the main peak are
rises

                        hydrograph4.extend(k2)

                        del k1[:] #empty the lists

                        del k2[:]

                        del k3[:]

                    else:

                         hydrograph4.extend(k3)

                         del k1[:] #empty the lists

                         del k2[:]

                         del k3[:]

                else:

                    if (hydrograph3[i+1] is (nan)): #so the last
rise in the storm event will always be marked as a fall

                        hydrograph4.extend(k2)

                        rise = 0

                        del k1[:] #empty the lists

                        del k2[:]

                        del k3[:]

            else:

                hydrograph4.append(hydrograph3[i])# or 'R'

                k4.append(stormEvent2[i])

                if (hydrograph3[i+1] is ('f')):

                    rise = max(k4)

                    del k4[:]

                else:

                    if (hydrograph3[i+1] is (nan)):

                        rise = 0

                        del k4[:]

        else:

            hydrograph4.append(nan)
```

```python
    #print hydrograph4

    #print len(hydrograph4)


    #final hydrograph with nan insesarted to seperate storm
events

    #final stormEvent with nan insesarted to seperate storm
events

    FinalStormEvent = []

    FinalHydrograph = []

    for i in range(0, len(stormEvent2)):

        if (hydrograph4[i] is not (nan)):

            if (hydrograph4[i] is 'R'):

                FinalStormEvent.append(stormEvent2[i])

                FinalHydrograph.append(hydrograph4[i])#or
'R'

            else:

                if (hydrograph4[i+1] is 'R'):

                    FinalStormEvent.append(nan)

                    FinalHydrograph.append(nan)

                else:

                    FinalStormEvent.append(stormEvent2[i])

FinalHydrograph.append(hydrograph4[i])#or 'f'

        else:

            FinalStormEvent.append(nan)

            FinalHydrograph.append(hydrograph4[i])#or
'nan'


    #print FinalStormEvent

    #print len(FinalStormEvent)

    #    plot (Q2)

    #    plot (FinalStormEvent, color = 'red')


    '''re-assign ID to storm events'''

    FinalStormID = []


    n=0

    for i in range(0, len(FinalStormEvent)):
```

```python
        if (FinalStormEvent[i] is not (nan)):

            FinalStormID.append(n)

            if (FinalStormEvent[i+1] is (nan)):

                n=n+1

        else:

            FinalStormID.append(nan)


#print FinalStormID #was 110 events, after aditional
1:3 fall split its 229


    '''04/06/16....check if there are storm events
contaning less than 5 data points again'''
    LastCount = [] #consider removing if less than 5

    n = 0

    j = 0 #to count data points within storm event


    for i in range(0, len(FinalStormID)):

        if (FinalStormID[i] == n):

            j = j + 1

            if (FinalStormID[i+1] is (nan)):

                LastCount.append(j)

                n = n + 1

        else:

            j = 0


#print len(LastCount)

#print min(LastCount)


    '''Peak for variables'''

    numberOfStorms = []

    h = [] #temp Q2

    peak = [] #max flow of each storm event

    n = 0

    for i in range(0, len(FinalStormID)):

        if (FinalStormID[i] == n):

#           print str(stormID[i]) + " " + str(n)

            h.append(QInt[i]) #temporary list
```

```python
            if (FinalStormID[i+1] is (nan)):

                peak.append(max(h)) #find a maximum
discharge to later find the falling and rising limb of
hydrograph

                numberOfStorms.append(n)

                del h[:] # emty the list

                n = n + 1


    '''variables'''

    startDischarge = []

    totalRainfall = []


    rainfall2 = rainfall[1:]

    rainfall2.pop()


    '''interpolate rainfall'''

    PInt = []

    Pinterpolation = sp.interp1d(day2, rainfall2,
kind='linear')

    PInt = Pinterpolation(decDay)


    #find total rainfall for each storm event


    s3 = 0 # for total rainfall

    n = 0

    for i in range(0, len(FinalStormID)):

        if (FinalStormID[i] == n):

            s3 += PInt[i]

            if (FinalStormID[i-1] is (nan)):

                startDischarge.append(QInt[i])

            else:

                if (FinalStormID[i+1] is (nan)):

                    totalRainfall.append(s3)

                    s3 = 0

                    n = n + 1


    '''Qp-Qo   variable'''
```

```python
    QpMinusQo = []
    for i in range(0, len(numberOfStorms)):
        QpMinusQo.append(peak[i] - startDischarge[i])


#print QpMinusQo
#print len(QpMinusQo)]
#total N flux

    '''Time since the last peak'''
    PeakTime = []
    n = 0
    for i in range(0, len(FinalStormID)):
        if (FinalStormID[i] == n):
            if (QInt[i] == peak[n]):
                PeakTime.append(decDay[i])
#           print PeakTime
                n = n + 1


#   print len (numberOfStorms)

#   print len (PeakTime)


    TimeSinceLastPeak = []
    TimeSinceLastPeak.append(0)
    for i in range(1, len(PeakTime)):
        TimeSinceLastPeak.append(PeakTime[i] -
PeakTime[i-1])
#   print len(numberOfStorms)

#   print len(totalRainfall)

#   print len(TimeSinceLastPeak)

#   print len(QpMinusQo)

#print TimeSinceLastPeak
#print len(TimeSinceLastPeak)
    NFlux = []
    for i in range(0, len(numberOfStorms)):
        NFlux.append(22241.968 +
49948.389*(totalRainfall[i]) -
35197.094*(TimeSinceLastPeak[i]) +
92306.823*(QpMinusQo[i]))
```

```python
    return sum(NFlux)


'''Start'''
dataFile = open('BatchQsim_Baseline.txt')
flowm3s = zeros([3000, 10957])

#Skip over the header
for i in range(0, 1):
    dataFile.readline()


#Read in the data into 2D arrays,
i = 0
for line in dataFile.readlines():
    line = line.split(' ')
    for n in range (0, 3000):
        flowm3s[n][i] = (float(line[n+2]))
#   print i,
#   print flowm3s[0][i]
    i += 1
#   print i


#create a 1D Q array
SumNF = []
for j in range (0,3000):
    Q = []
    P = []
#   to get the row number for P as 30 lines
#   in Q will have to read same one line in P
    g = int(round(j/30))
#   print g
    for i in range(0,10957):
        Q.append(flowm3s[j][i])
    SumNF.append(calcNitrate(Q, g))
#   print SumNF
    print len(SumNF)
```

```python
out = open("SumNF_Baseline.txt", 'w+')
out.write("SumNF\n")
for i in range(0, 3000):
    out.write(str(SumNF[i]))
    out.write("\n")
    out.flush()
#
# -*- coding: utf-8 -*-
"""
Created on Fri Jul 08 14:17:35 2016
@author: Vaida
"""


from matplotlib.pylab import *
import matplotlib.patches as mpatches
import pylab


#import Baseline SumNF
dataFile = open('SumNF_Baseline.txt')


#Skip over the header
for i in range(0, 1):
    dataFile.readline()


baseline = []
for line in dataFile.readlines():
    line = line.split(' ')
    baseline.append(float(line[0]))


#import 2050 SumNF
dataFile = open('SumNF_2050.txt')


#Skip over the header
for i in range(0, 1):
    dataFile.readline()


#Read in the data
```

```python
N2050 = []
for line in dataFile.readlines():
    line = line.split(' ')
    N2050.append(float(line[0]))


#import 2080 SumNF
dataFile = open('SumNF_2080.txt')


#Skip over the header
for i in range(0, 1):
    dataFile.readline()


#Read in the data into 2D arrays,
N2080 = []
for line in dataFile.readlines():
    line = line.split(' ')
    N2080.append(float(line[0]))


#calculate change factors
cf2050 = []
for i in range (0, len(baseline)):
    for j in range (0, len(N2050)):
        cf2050.append((N2050[j] - baseline[i]) /
baseline[i])
#
hist(cf2050, bins=100, cumulative=True,
histtype='step', normed=True, label = '2050s')
xlabel("change factor")
ylabel("probability of change being less than")
axvline(0, linewidth=1, color='r')
#savefig("cfN2050", dpi=300)


cf2080 = []
for i in range (0, len(baseline)):
    for j in range (0, len(N2080)):
        cf2080.append((N2080[j] - baseline[i]) /
baseline[i])
```

```python
hist(cf2080, bins=100, cumulative=True,
histtype='step', normed=True, color='black', label =
'2080s')

xlabel("change factor")

ylabel("probability of change being less than")

axvline(0, linewidth=1, color='r')

pylab.legend(loc='upper left')


#savefig("cfN2080", dpi=300)
#plot (cf2050)


# -*- coding: utf-8 -*-
"""Created on Sat Sep 24 18:04:27 2016
@author: Vaida
"""
from matplotlib.pylab import *
import matplotlib.patches as mpatches
import pylab


#sum of Q for 2050


dataFile = open('BatchQsim_2050.txt')
flowm3s = zeros([3000, 10957])


#Skip over the header
for i in range(0, 1):
    dataFile.readline()


#Read in the data into 2D arrays,
i = 0
for line in dataFile.readlines():
    line = line.split(' ')
    for n in range (0, 3000):
        flowm3s[n][i] = (float(line[n+2]))
#    print i,
#    print flowm3s[0][i]
    i += 1
```

```python
#    print i

#create a 1D Q array
SumQ = []
for j in range (0,3000):
    Q = []
    for i in range(0,10957):
        Q.append(flowm3s[j][i])
    SumQ.append(sum(Q))
#print SumQ
#print len(SumQ)


out = open("SumQ_2050.txt", 'w+')
out.write("SumQ\n")
for i in range(0, 3000):
    out.write(str(SumQ[i]))
    out.write("\n")
    out.flush()


#sum of Q for 2080
'''Start'''
dataFile = open('BatchQsim_2080.txt')
flowm3s = zeros([3000, 10957])


#Skip over the header
for i in range(0, 1):
    dataFile.readline()


#Read in the data into 2D arrays,
i = 0
for line in dataFile.readlines():
    line = line.split(' ')
    for n in range (0, 3000):
        flowm3s[n][i] = (float(line[n+2]))
#    print i,
#    print flowm3s[0][i]
    i += 1
```

116

```python
#    print i

#create a 1D Q array
SumQ = []
for j in range (0,3000):
    Q = []
    for i in range(0,10957):
        Q.append(flowm3s[j][i])
    SumQ.append(sum(Q))
#print SumQ
#print len(SumQ)


out = open("SumQ_2080.txt", 'w+')
out.write("SumQ\n")
for i in range(0, 3000):
    out.write(str(SumQ[i]))
    out.write("\n")
    out.flush()


#sum of Q for Baseline

dataFile = open('BatchQsim_Baseline.txt')
flowm3s = zeros([3000, 10957])


#Skip over the header
for i in range(0, 1):
    dataFile.readline()


#Read in the data into 2D arrays,
i = 0
for line in dataFile.readlines():
    line = line.split(' ')
    for n in range (0, 3000):
        flowm3s[n][i] = (float(line[n+2]))
#    print i,
#    print flowm3s[0][i]
    i += 1
```

```python
#    print i

#create a 1D Q array
SumQ = []
for j in range (0,3000):
    Q = []
    for i in range(0,10957):
        Q.append(flowm3s[j][i])
    SumQ.append(sum(Q))
print SumQ
print len(SumQ)


out = open("SumQ_Baseline.txt", 'w+')
out.write("SumQ\n")
for i in range(0, 3000):
    out.write(str(SumQ[i]))
    out.write("\n")
    out.flush()


# factors for Q
#import Baseline SumQ
dataFile = open('SumQ_Baseline.txt')


#Skip over the header
for i in range(0, 1):
    dataFile.readline()


baseline = []
for line in dataFile.readlines():
    line = line.split(' ')
    baseline.append(float(line[0]))


#import 2050 SumQ
dataFile = open('SumQ_2050.txt')


#Skip over the header
```

117

```python
for i in range(0, 1):

    dataFile.readline()


#Read in the data
Q2050 = []
for line in dataFile.readlines():

    line = line.split(' ')

    Q2050.append(float(line[0]))


#import 2080 SumQ
dataFile = open('SumQ_2080.txt')


#Skip over the header
for i in range(0, 1):

    dataFile.readline()


#Read in the data into 2D arrays,
Q2080 = []
for line in dataFile.readlines():

    line = line.split(' ')

    Q2080.append(float(line[0]))


#calculate change factors
cf2050 = []
for i in range (0, len(baseline)):

    for j in range (0, len(Q2050)):

        cf2050.append((Q2050[j] - baseline[i]) /
baseline[i])

#
hist(cf2050, bins=100, cumulative=True,
histtype='step', normed=True, label = '2050s')

xlabel("change factor")

ylabel("probability of change being less than")

axvline(0, linewidth=1, color='r')

#savefig("cfQ2050", dpi=300)


cf2080 = []

for i in range (0, len(baseline)):

    for j in range (0, len(Q2080)):

        cf2080.append((Q2080[j] - baseline[i]) /
baseline[i])


hist(cf2080, bins=100, cumulative=True,
histtype='step', normed=True, color='black', label =
'2080s')

xlabel("change factor")

ylabel("probability of change being less than")

axvline(0, linewidth=1, color='r')

pylab.legend(loc='upper left')


#savefig("cfQ2050-2080", dpi=300)

#plot (cf2050)
```

Appendix 2: Python output/Results

ID	NAverageConc	TotalNFlux	MaxNConc	MinNConc	Hysteresis	HMagnitude
1	8.048218	1225649	10.61	6.73	-1	0.602531
5	8.465679	647973.2	10	6.23	-1	1.667982
7	8.388924	1972573	10.05	7.06	-1	1.071646
10	6.937941	442637.3	8.44	6.13	-1	1.056774
12	8.763667	1186259	10.92	7.51	-1	0.730692
13	7.817263	507973.7	9.95	7.07	-1	1.124521
14	9.323333	399856	9.95	8.89	1	0.42111
15	9.832661	1226830	11.41	6.83	-1	1.849339
16	10.07777	1477278	12.16	6.06	-1	1.965477
17	10.28098	1189031	11.63	5.64	1	0.354279
19	8.254608	1441696	10.57	5.27	-1	1.086964
21	8.750652	456669.7	10.9	7.29	1	0.913995
22	8.414138	289944.9	9.16	7.57	1	0.523838
23	7.077273	1192270	9.21	3.93	1	0.277954
26	8.612466	267424.9	9.2	8.02	1	0.650063
27	8.49032	562085.3	9.18	7.5	1	0.351077
28	8.538581	836095.9	9.53	6.61	1	0.699638
29	8.031974	479700.2	8.96	6.55	1	0.546916
30	8.175056	707278.1	8.97	6.36	1	0.745381
31	8.574648	1211392	10.35	3.97	-1	0.323552
32	9.760613	2956048	10.43	6.37	1	0.282344
33	9.24901	1095940	10.92	7.82	1	0.68562
35	8.964719	1823058	9.96	6.21	-1	0.609905
37	9.478868	2065686	11.57	7.67	-1	1.526945
38	7.961641	744657.7	8.5	7.18	1	0.393762
40	8.028177	341596.8	13	5.99	-1	1.058014
41	8.022766	467303.2	16.25	5.54	1	2.044726
45	10.6283	7747932	20.11	6.52	-1	2.465367
47	12.00148	478379.3	27.13	7.77	-1	5.097084
48	10.57653	613672.6	20.34	7.45	-1	2.082601
49	16.02472	739956.6	30	8.64	-1	1.599666
50	11.18267	4289102	21.58	10.03	1	3.250976
51	11.10786	3367991	24.53	8.29	-1	1.121515
52	9.371164	764597.3	10.04	8.31	1	0.61472
55	10.03143	855390.6	13.06	8.7	1	1.465208
56	9.11539	1007234	9.71	8.26	1	0.474555
57	8.272506	828010.1	9.64	6.92	1	0.857731
58	7.728256	237264.3	9.06	6.89	1	0.960447
59	8.089109	371166.1	10.7	6.75	-1	1.489802
60	7.122939	321840.9	8.27	6.64	1	1.052931
61	8.446308	96897.06	9.3	7.34	-1	0.919991
ID	NAverageConc	TotalNFlux	MaxNConc	MinNConc	Hysteresis	HMagnitude

ID	NAverageConc	TotalNFlux	MaxNConc	MinNConc	Hysteresis	HMagnitude
62	9.998378	586933.6	11.55	7.2	-1	2.193007
63	8.022013	1393225	9.61	7.14	1	0.510201
64	7.967647	1145615	10.33	6.96	1	0.305327
65	7.757848	706088.3	8.42	6.9	-1	0.706032
66	7.53	175447.6	8.15	6.97	1	0.423686
67	8.075477	1725294	8.49	6.52	-1	0.442354
68	8.07939	933570.5	9.66	7.57	-1	0.628215
70	7.774842	437233	9.02	6.8	1	1.304018
71	6.81087	308681.3	7.51	5.86	1	0.466325
72	7.636857	1188277	8.84	5.36	1	0.404608
73	8.11947	1143772	8.83	5.87	1	0.79566
74	7.916261	1391706	8.91	5.69	1	0.831182
75	6.251803	398722	7.5	5.38	1	0.428281
76	7.827656	1655249	9.16	4.57	1	0.562797
77	7.320259	434037.2	8.19	5.99	1	0.700271
79	5.749412	251939	6.48	5.03	1	0.603484
80	6.587748	478204.6	7.09	5.24	1	0.388683
81	6.756979	328428	7.15	6.48	1	0.328384
82	6.42	502786.1	7.23	5.64	1	0.645725
84	6.526176	308679.1	7.51	6.18	1	0.924562
85	5.6376	298056.6	6.68	4.88	1	0.724959
87	7.245802	697223.7	8.1	5.16	1	0.356053
88	6.202407	1406234	8.08	4.21	1	0.543779
89	7.273	643233.1	8.49	4.9	1	1.391968
90	6.051961	602471	8.46	3.87	1	0.95859
91	8.241705	2055673	13.2	3.4	1	0.712851
92	7.463273	270325.6	9.43	5.92	1	1.427618
93	6.705625	389958.3	7.39	5.26	1	0.483467
94	7.814414	1975897	9.02	3.45	-1	0.419168
95	8.125	203295.4	9.24	6.95	1	1.110184
96	6.229545	185865.3	6.9	5.69	1	0.306287
98	7.142632	445980.1	7.59	6.45	1	0.337983
99	6.590038	2246342	8.77	3.96	-1	1.060079
100	6.221714	418517.8	8.62	4.9	1	1.539119
101	7.470149	522290	8.53	5.75	1	0.614165
102	6.762526	725578.4	8.58	5.07	1	0.540216
103	6.410938	455766.1	7.84	4.82	1	0.807292
104	6.087941	289813.5	7.46	4.82	1	0.505379
106	5.183684	369989.6	6.41	4.28	1	0.424743
107	6.404634	471889.4	7.34	5.32	1	0.768669
110	7.175106	1017346	8.72	5.47	1	0.590625
111	6.821429	805901.8	8.14	4.4	1	0.582975
ID	NAverageConc	TotalNFlux	MaxNConc	MinNConc	Hysteresis	HMagnitude
112	7.225161	323741	8.11	6.37	1	0.918372

ID	NAverageConc	TotalNFlux	MaxNConc	MinNConc	Hysteresis	HMagnitude
113	7.884781	1368394	8.44	5.31	1	0.514076
114	7.582676	446244.8	8.32	6.95	1	0.77177
115	7.056978	889889.9	8.07	4.78	1	0.471051
116	8.005597	327894.1	9.59	6.8	1	1.485822
119	17.47135	777190.1	30	6.76	-1	12.84347
120	24.59467	992531.3	30	16.92	1	2.166451
121	15.46097	796601.5	21.04	12.12	1	0.506191
122	11.34541	2031924	15.1	9	1	0.604346
123	9.876329	1504317	11.5	8.56	1	0.584148
124	9.515556	922310.1	19.52	7.25	-1	2.769711
125	8.572621	1601203	16.19	6.01	-1	6.162933
132	17.68333	531086.6	23.84	7.93	-1	7.518693
133	12.12167	248465.1	17.67	8.36	-1	1.593394
134	13.65646	2969512	24.1	6.97	-1	3.953762
135	9.099236	517492.8	10.64	7.61	-1	0.355122
136	8.976604	2974436	15.83	7.14	-1	1.217898
139	7.254516	1003005	8.72	6.18	-1	0.9205
143	8.710251	823650.3	19.99	6.1	1	0.356662
144	7.638133	1366164	12.01	6.12	1	0.402704
145	7.978947	287560.1	14.98	6.67	-1	2.061414
146	9.660317	2568016	16.21	7.29	-1	0.906116
147	10.25618	846427.3	15.48	6.93	-1	3.315084
148	9.412258	598720.3	11.5	7.55	-1	2.294097
149	11.12143	818241.8	12.01	9.95	-1	1.025538
150	10.616	576296.5	11.39	9.67	-1	0.845538
151	11.31966	1127868	12.26	9.72	-1	0.482209
152	10.63533	1014373	11.9	9.73	1	0.554065
153	10.65319	1553272	11.07	8.92	1	0.29193
155	10.10654	1473441	11.15	7.53	1	0.322508
156	9.374038	345407.1	10.76	8.54	1	1.701645
157	9.250218	2372108	10.29	7.43	-1	0.614749
158	8.972	1197565	9.55	6.6	1	0.484581
159	9.266061	901566.7	11.13	8.62	1	0.622706
160	9.689375	1227509	10.25	7.5	1	0.492252
161	8.862059	676622.5	10.04	6.14	1	0.584528
162	7.899753	1533151	9.29	6.98	1	0.462494
163	7.728101	615354.8	8.1	7.16	1	0.393723
164	7.464956	405011.2	8.29	6.85	1	0.754514
166	7.197987	368458.7	9.03	6.25	1	1.116625
167	6.79067	592029.7	7.12	5.99	-1	0.327435
168	9.064848	2255178	13.72	6.59	-1	4.842753
ID	NAverageConc	TotalNFlux	MaxNConc	MinNConc	Hysteresis	HMagnitude
169	11.12919	2022807	12.23	6.68	1	0.626282
170	9.401475	1472139	12.27	5.31	-1	1.684503

171	10.59058	952375.9	12.19	7.61	1	1.749587
172	8.605098	1960693	12.23	3.72	-1	1.102328
173	9.367805	2062950	11.37	5.01	1	1.508118
174	6.099565	855409.1	11.24	2.95	1	0.82197
175	7.782632	1338068	10.38	4.23	1	1.12074
176	10.48078	1418106	11.21	7.68	1	1.444948
177	8.704375	203885.8	10.81	6.41	1	2.652286
178	8.916446	1794442	12.65	3.3	-1	0.540116
179	8.248555	2214122	10.42	3.71	-1	0.4174
181	7.925319	599544.2	9.6	6.1	1	1.436996
182	8.63359	764476.4	9.79	6.26	1	0.835203
183	6.452917	473785	9.63	3.29	1	0.912872
184	5.618824	433277.8	6.36	4.77	1	0.356903
185	6.131579	892411.9	7.05	3.86	1	0.756795
186	6.928831	1328715	8.89	3.49	1	0.461277
187	7.914186	1063860	9.25	3.99	1	1.107799
188	8.27875	581595.8	9.18	7.34	1	1.081356
191	6.525882	245064.6	7.93	5.53	1	1.209561
192	7.688049	773889.3	8.35	5.58	1	0.73286
193	7.106892	412553.5	8.53	6.09	1	1.038715
194	6.667463	455218.4	7.84	4.35	1	0.901247
195	6.777436	248915.2	7.88	5.84	1	1.194123
196	6.147949	342087.8	7.33	4.2	1	0.962337
197	5.606667	570412.4	7.68	2.47	1	0.941389
199	5.247407	406688.4	8.61	2.99	1	1.19383
200	6.624444	239219.8	7.4	5.87	1	0.442733
201	7.413095	2034268	8.63	2.77	1	0.299441
202	6.974434	638959.3	8.53	5.49	1	1.486641
203	5.267241	296193.6	7.48	3.91	1	1.727419
204	5.667544	803107.4	7.12	3.65	1	0.483625
205	7.110484	507009.1	7.75	6.03	1	0.66965
206	6.513621	397825.6	7.74	5.24	1	1.256501
208	5.955349	553444.4	7.52	3.56	1	1.145901
209	5.70125	235847.9	7.54	4.08	1	2.027054
211	6.454754	345436.6	7.51	5.84	1	1.078485
213	4.7375	123786.4	5.2	4.35	1	0.601667
215	5.31	908473.7	6.91	3.86	1	0.500035
216	6.441895	1003620	6.97	4.64	1	0.937097
217	6.428511	1214023	7.22	3.05	1	0.78655
218	5.804643	202687	7.31	4.85	1	1.745883
ID	NAverageConc	TotalNFlux	MaxNConc	MinNConc	Hysteresis	HMagnitude
219	6.441828	1025994	7.34	3.67	1	0.282946
221	8.17127	320172.8	13.78	6	1	0.760517
223	9.990991	1116856	30	6.06	1	1.626674

224	9.890222	325024.9	19.86	6.35	-1	6.897068
225	8.530241	703690.6	13.03	6.56	-1	1.037146
226	6.956176	548617.3	7.55	6.4	-1	0.587074
227	6.922594	1571472	8.06	4.39	1	0.431606
228	7.528235	65348.28	8.22	7.08	-1	0.462949
230	7.99757	271535.4	12.04	5.7	-1	1.8251
231	7.41125	245991.1	8.65	6.28	-1	0.967412
232	7.655473	2017151	12.2	6.11	-1	3.071643
233	11.04608	309152.5	30	6.51	-1	6.027342
234	8.321226	274803.6	22.23	5.78	-1	1.883443
235	10.31467	147848.6	15.77	6.01	-1	4.393367
236	19.03374	1035450	30	6.77	-1	7.819869
237	13.113	466626.1	30	9.4	-1	3.227973
239	7.920894	98903.34	10.75	6.77	-1	2.507031
241	12.33352	222362.8	16.79	6.46	-1	4.366125
243	9.21717	233029.4	12.13	6.56	-1	2.88382
244	10.60518	475553.7	21.79	6.73	-1	2.895212
245	11.34625	756866.2	14.73	7.05	-1	3.66708
246	9.386917	1278657	11.46	8.38	-1	0.346738
247	7.026013	755567.8	8.32	6.39	1	0.292265

ID	TimeSinceLaPeak	TotRainfall	stormAPI	TotalQ	StartQ	PeakQ	Qp-Qo
1	0	24.26	14.04473	80.22	0.18	2.57	2.39
5	2.125	14.13	24.80203	40.03	0.16	0.97	0.81
7	1.041667	18.18	40.08945	131.06	0.38	4.18	3.8
10	4.458333	14.6	21.25597	35.32	0.24	2.36	2.12
12	8.1875	12	20.41326	75.05	0.19	1.83	1.64
13	13.0625	18.62	17.50744	35.39	0.17	1.96	1.79
14	0.520833	4.81	24.37407	23.78	0.55	0.68	0.13
15	1.625	11.69	29.94174	74.09	0.38	2.83	2.45
16	2.416667	21.72	40.53616	93.9	0.33	4.96	4.63
17	1.520833	10.98	41.31894	75.69	0.49	3.8	3.31
19	2.708333	24.51	38.07942	107.44	0.44	2.88	2.44
21	3.375	7.75	57.64256	29.56	0.4	0.96	0.56
22	0.4375	1.68	63.15475	19.22	0.55	0.86	0.31
23	0.75	24.02	66.94756	105.47	0.56	4.9	4.34
26	2.145833	6.31	24.99608	17.27	0.2	0.31	0.11
27	1.354167	5.44	25.20147	36.97	0.24	0.42	0.18
28	3.375	13.84	21.0939	55.54	0.24	1.1	0.86
29	2.479167	6.66	26.01691	33.76	0.31	0.73	0.42
30	1.583333	12.93	29.37965	49.92	0.36	1.52	1.16
31	1.916667	20.89	46.11507	102.05	0.44	5.68	5.24
32	1.791667	16.63	47.61742	178.17	0.47	2.05	1.58
33	16.41667	13.47	7.591235	67.73	0.12	0.46	0.34
35	3.166667	10.66	16.71162	118.3	0.28	1.36	1.08
37	3.8125	20.67	6.795283	118.83	0.13	2.22	2.09
38	14.125	8.46	10.04509	51.96	0.11	0.2	0.09
40	1.8125	12.93	22.73699	21.83	0.06	0.31	0.25
41	5.791667	16.88	16.48323	31.54	0.07	0.19	0.12
45	10.875	69.35	42.29436	295.19	0.09	5.62	5.53
47	8.5	26.43	28.74473	20.39	0.05	0.17	0.12
48	7.5625	29.09	24.4562	28.3	0.05	0.42	0.37
49	5.3125	19.88	48.56768	16.37	0.08	1.33	1.25
50	0.5	36.54	59.63124	191.03	0.69	4.14	3.45
51	6.041667	36.42	52.81503	130.19	0.16	7.82	7.66
52	4.541667	12.57	40.46343	45.36	0.22	0.31	0.09
55	3.875	17.96	39.03062	45.84	0.18	2.06	1.88
56	0.5625	6.32	36.22368	62.53	0.55	0.75	0.2
57	8.833333	24.12	22.72866	56.06	0.12	0.2	0.08
58	12.79167	12.35	19.10867	17	0.08	0.15	0.07
59	3.5	10.69	18.90492	22.91	0.09	1.98	1.89
60	1.0625	0.11	16.33644	25.12	0.22	0.22	0
61	9.3125	9.07	15.26858	6.39	0.07	0.15	0.08
62	1.479167	15.32	32.50939	31.27	0.09	2.7	2.61

ID	TimeSinceLaPeak	TotRainfall	stormAPI	TotalQ	StartQ	PeakQ	Qp-Qo
63	0.958333	16.05	31.72797	92.42	0.45	2.18	1.73
64	7.25	25.16	30.36033	73.95	0.14	3.54	3.4
65	2.166667	10.21	39.16263	51.48	0.27	1.9	1.63
66	1.875	9.05	41.13753	13.01	0.27	0.49	0.22
67	0.5	11.28	44.1729	123.93	0.33	4.48	4.15
68	25.6875	35.05	20.2516	62.43	0.06	4.72	4.66
70	5.3125	12.08	47.00328	31.42	0.2	0.65	0.45
71	1.979167	14.84	61.64645	27.13	0.29	3.49	3.2
72	0.270833	12.83	58.89771	95.82	1.48	5.08	3.6
73	2.291667	7.62	49.93861	82.48	0.34	1.92	1.58
74	7.041667	29.26	26.69484	110.04	0.16	3.71	3.55
75	4.3125	13.26	36.21454	36.86	0.23	1.21	0.98
76	0.895833	7.66	38.18185	127.42	0.39	3.11	2.72
77	14.5625	14.19	13.63322	34.14	0.13	0.63	0.5
79	1.416667	7.35	25.93806	25.08	0.32	1.61	1.29
80	0.729167	1.27	25.07772	41.31	0.44	0.89	0.45
81	2.541667	4.5	20.85336	27.01	0.24	0.32	0.08
82	3.0625	9.3	22.65447	44.06	0.23	0.71	0.48
84	3.3125	20.66	33.97757	26.25	0.25	0.65	0.4
85	2.020833	10.65	41.55721	31.39	0.28	3.03	2.75
87	0.9375	6.32	40.47255	58.48	0.56	2.49	1.93
88	2.083333	39.62	68.75135	144.92	0.34	8.28	7.94
89	1	4.07	63.41841	53.66	0.91	2.53	1.62
90	1.645833	13.29	63.59931	63.25	0.48	2.75	2.27
91	0.854167	19.35	70.93122	171.12	0.62	4.17	3.55
92	7.708333	5.45	32.57771	21.07	0.18	0.64	0.46
93	0.833333	3.01	29.23819	33.07	0.31	0.76	0.45
94	2.041667	17.7	36.35319	163.29	0.22	3.63	3.41
95	8.0625	9.5	22.61819	14.27	0.15	0.62	0.47
96	0.708333	5.98	26.47727	16.82	0.4	1.08	0.68
98	1.875	3.08	23.71577	34.93	0.36	0.58	0.22
99	2.666667	55.22	52.86242	235.99	0.27	5.31	5.04
100	2.0625	7.72	56.4702	38.56	0.58	1.87	1.29
101	0.666667	2.22	53.17478	40.86	0.89	1.36	0.47
102	2.229167	14.89	51.91226	63.59	0.37	1.25	0.88
103	1.75	7.65	47.34742	41.41	0.41	1.24	0.83
104	1.458333	7.34	54.52086	28.05	0.42	1.45	1.03
106	0.4375	3.24	71.9969	41.35	2.71	3.47	0.76
107	0.4375	5.56	72.05627	42.3	1.3	1.86	0.56
110	1.520833	17.54	39.00502	83.92	0.49	4.64	4.15
111	1.0625	16.75	49.3476	75.35	0.71	5.27	4.56
112	0.833333	2.42	52.17085	25.12	0.74	1.25	0.51
113	0.770833	13.15	51.68409	102.62	0.56	1.62	1.06
ID	TimeSinceLaPeak	TotRainfall	stormAPI	TotalQ	StartQ	PeakQ	Qp-Qo

ID	TimeSinceLaPeak	TotRainfall	stormAPI	TotalQ	StartQ	PeakQ	Qp-Qo
114	5.583333	3.12	29.74966	32.7	0.21	0.29	0.08
115	8.145833	9.47	17.53641	73.58	0.17	0.92	0.75
116	23.6875	7.74	5.183001	22.52	0.09	0.26	0.17
119	27.02083	18.79	3.984356	16.81	0.08	1.34	1.26
120	0.583333	6.76	22.61878	21.65	0.3	0.79	0.49
121	1.0625	4.34	25.22275	27.07	0.34	0.88	0.54
122	1.979167	27.98	28.9827	104.46	0.27	4.45	4.18
123	1.125	5.34	44.78867	83.52	0.53	1.85	1.32
124	6.375	9.31	24.20778	51.25	0.17	0.32	0.15
125	22.8125	33.2	14.02272	78.64	0.09	5.5	5.41
132	24.77083	22.4	20.30119	15.3	0.03	1.1	1.07
133	1.4375	10.6	23.98155	10.8	0.1	0.29	0.19
134	3.041667	50.2	30.61755	89.32	0.05	2.87	2.82
135	4	20.4	55.16944	30.61	0.14	0.39	0.25
136	4.229167	42.4	56.15732	144.19	0.1	4.59	4.49
139	1.791667	16.35	42.48423	73.97	0.2	1.02	0.82
143	7.895833	35.09	9.631414	43.85	0.04	0.6	0.56
144	9.354167	26.71	20.51802	94.63	0.08	0.57	0.49
145	17.6875	26.1	4.24545	15.48	0.05	0.51	0.46
146	0.5	17.6	31.57832	118.84	0.34	3.71	3.37
147	12.27083	19.06	18.17775	42.74	0.1	1.26	1.16
148	3.041667	12.81	20.08249	34.49	0.26	1.22	0.96
149	0.854167	9.21	32.84299	41.05	0.5	1.59	1.09
150	1.020833	8.84	39.05014	30.34	0.6	1.79	1.19
151	0.416667	11.84	42.19359	56.57	1.1	3.35	2.25
152	0.6875	11.88	51.94341	53.62	1.24	2.74	1.5
153	0.583333	6.23	57.81823	82.57	1.33	2.82	1.49
155	0.3125	9.88	61.74106	86.77	1.66	4.26	2.6
156	1.125	4.77	65.37932	20.53	0.71	1.42	0.71
157	0.4375	19.11	66.4576	148.94	0.9	2.53	1.63
158	3.541667	14.25	58.80273	75.92	0.37	1.78	1.41
159	2.854167	18.85	51.47586	53.27	0.34	2.5	2.16
160	0.5625	2.79	55.95964	72.51	1.42	2.47	1.05
161	2.083333	4.36	44.84509	42.21	0.4	0.59	0.19
162	4.270833	13.71	37.11346	108.26	0.32	0.69	0.37
163	4.458333	6.23	24.50777	44.34	0.25	0.34	0.09
164	4.375	6.64	14.92701	30.2	0.21	0.39	0.18
166	15.375	12.77	4.157869	28.42	0.11	0.24	0.13
167	3.208333	2.96	11.78315	48.69	0.24	0.46	0.22
168	5.833333	33.06	8.483797	126.59	0.17	5.94	5.77
169	1.229167	14.44	40.66311	109.52	0.67	3.74	3.07
170	2.875	25.9	39.22545	100.37	0.37	5.91	5.54
171	0.770833	5.53	53.49412	51.98	0.91	2.07	1.16
ID	TimeSinceLaPeak	TotRainfall	stormAPI	TotalQ	StartQ	PeakQ	Qp-Qo
172	1.25	37.87	68.52492	160.41	0.6	7.44	6.84

ID	TimeSinceLaPeak	TotRainfall	stormAPI	TotalQ	StartQ	PeakQ	Qp-Qo
173	0.875	18.06	81.00296	134.35	1.55	5.06	3.51
174	1.791667	29.24	89.40502	100.68	0.71	7.29	6.58
175	0.5	15.89	102.0019	107.68	2.5	5.93	3.43
176	0.729167	2.83	101.1591	78.02	1.27	1.92	0.65
177	2.3125	6.43	75.07745	13.72	0.45	1.4	0.95
178	0.354167	17.7	78.16488	144.43	0.86	3.88	3.02
179	3.041667	37.24	67.52609	187.67	0.42	5.13	4.71
181	1.25	8.87	67.83443	43.37	0.63	1.58	0.95
182	0.9375	3.29	66.03843	51.44	0.9	1.27	0.37
183	1.791667	21.09	64.21627	59.66	0.42	5.09	4.67
184	0.3125	4.29	69.71383	43.59	2.81	3.42	0.61
185	0.395833	9.78	72.70731	85.76	2.21	4.8	2.59
186	0.854167	15.65	77.51586	113.91	1.51	2.49	0.98
187	1.729167	12.4	74.03844	84.16	0.71	2.77	2.06
188	1.791667	3.07	65.74744	39.18	0.51	0.74	0.23
191	2.6875	3.82	37.9005	21.19	0.39	0.9	0.51
192	0.666667	2.03	37.24152	57.93	0.53	1.09	0.56
193	2.895833	9.27	31.85388	32.7	0.3	0.63	0.33
194	1.291667	6.65	30.92378	40.26	0.4	1.72	1.32
195	1.458333	4.9	32.77399	20.52	0.4	0.84	0.44
196	0.729167	5.92	33.24394	32.9	0.48	2.62	2.14
197	0.895833	13.95	36.94456	68.07	0.55	4.35	3.8
199	3.625	13.49	37.06476	53.92	0.31	3.9	3.59
200	0.395833	1.37	42.40098	20.24	1.17	1.7	0.53
201	0.604167	27.41	48.69404	188.96	0.74	5.99	5.25
202	4.083333	9.93	40.95795	51.33	0.38	1.07	0.69
203	2.25	7.02	34.66422	33.74	0.38	2.32	1.94
204	0.75	15.21	42.0832	85.16	0.69	3.27	2.58
205	1.083333	2.49	47.08443	40.05	0.72	1.05	0.33
206	1.479167	5.58	39.924	34.49	0.46	1.01	0.55
208	1	6.02	58.04896	55.53	0.98	2.45	1.47
209	0.958333	6.87	58.96308	24.45	0.77	2.61	1.84
211	5.354167	6.04	35.32756	29.82	0.39	0.68	0.29
213	1.25	2	41.44524	14.56	0.9	1.51	0.61
215	0.479167	12.18	61.72365	103.01	2.63	3.81	1.18
216	1.083333	9.41	63.64489	89.57	1.05	2.07	1.02
217	2.083333	15.48	53.40105	120.13	0.63	4.06	3.43
218	2.916667	6.13	46.75974	19.49	0.42	1.27	0.85
219	0.75	6.86	48.44248	95.23	0.55	1.93	1.38
221	13.39583	13.39	14.55653	20.82	0.14	0.62	0.48
223	10.35417	18.86	13.37684	44.55	0.16	1.33	1.17
224	1.958333	2.89	18.91135	17.56	0.41	0.71	0.3
ID	TimeSinceLaPeak	TotRainfall	stormAPI	TotalQ	StartQ	PeakQ	Qp-Qo
225	1.020833	5.53	22.30777	43.12	0.37	1.94	1.57
226	2.354167	10.57	22.63401	44.13	0.33	1.35	1.02

227	0.875	8.74	29.94636	133.89	0.47	3	2.53
228	17.4375	4.17	8.655765	4.82	0.08	0.12	0.04
230	14.1875	15.25	13.07829	17.35	0.08	0.45	0.37
231	1.5	7.99	23.28563	18.32	0.19	0.41	0.22
232	0.958333	14.83	37.2339	132.24	0.32	5.19	4.87
233	12.41667	6.69	15.81582	14.54	0.1	0.19	0.09
234	3.833333	9.86	16.86047	17.62	0.08	0.15	0.07
235	9.729167	7.92	16.36764	8.18	0.06	0.17	0.11
236	3.104167	18.66	21.07019	27.79	0.07	0.37	0.3
237	2.875	5.97	23.5278	18.22	0.11	0.28	0.17
239	18.41667	25.15	2.894229	6.89	0.04	0.1	0.06
241	0.083333	15.39	23.95073	10.18	0.47	0.47	0
243	2.854167	13.21	36.46233	13.66	0.08	0.24	0.16
244	2.229167	19.6	42.58766	21.63	0.08	0.61	0.53
245	2.104167	24.14	50.93042	33.79	0.15	1.62	1.47
246	0.9375	16.02	60.07773	72.7	0.45	1.34	0.89
247	3.875	16.28	53.18593	58.99	0.24	0.33	0.09

ID	TMed3	TMed5	TMed7	TMed10	TMed15	SeasonID	seasonCos
1						4	-0.25892
5	8.46	10.16	11.7	14.275	13.39	4	0.114783
7	12.685	11.23	11.135	11.745	13.63	4	0.176159
10	11.455	10.725	9.385	10.075	9.205	4	0.51724
12	9.55	8.715	8.53	8.95	9.355	4	0.645036
13	7.509	7.786	7.924	7.5705	8.4	4	0.788723
14	10.01	8.04	8.245	7.946	8.285	4	0.809417
15	10.01	7.8655	7.8495	8.29	8.045	4	0.820843
16	6.727	6.911	7.4655	7.591	7.509	4	0.842731
17	6.358	6.9605	7.0245	7.473	7.594	1	0.86058
19	1.8315	3.767	5.0285	5.7675	6.4275	1	0.902111
21	3.143	2.7785	2.13	2.916	4.6255	1	0.936623
22	2.6935	3.2195	2.471	2.9255	4.3775	1	0.942395
23	3.729	3.5575	2.9065	2.996	4.1515	1	0.94594
26	9.37	5.2255	3.455	2.551	3.1595	1	0.988568
27	9.03	6.066	4.9975	3.28	3.332	1	0.992221
28	9.48	9.29	9.195	6.187	4.2365	1	0.996832
29	5.153	6.5225	6.4995	6.8815	5.232	1	0.999667
30	5.536	6.257	7.4325	7.69	6.1775	1	0.999994
31	6.4755	4.606	5.7045	6.619	6.341	1	0.99936
32	3.7995	5.226	4.629	6.143	6.443	1	0.998108
33	-0.236	-0.6265	1.0555	3.149	4.309	1	0.957505
35	3.98	5.2015	5.0095	4.2645	3.841	1	0.910188
37	3.06	4.901	4.9945	3.3705	2.0895	2	0.648485
38	2.9825	4.9395	6.7225	6.8275	7.28	2	0.432583
40	5.619	6.5275	6.4865	5.8705	6.076	2	-0.18043
41	3.237	4.415	4.7575	5.6955	5.5595	2	-0.25304
45	6.9045	6.2145	6.3225	6.7215	6.487	2	-0.60829
47	10.725	9.335	9.62	10.78	12.37	3	-0.91436
48	9.695	10.045	10.75	10.81	10.65	3	-0.95807
49	11.745	11.245	11.095	10.73	11.05	3	-0.98383
50	12.07	11.215	11.215	10.965	11.07	3	-0.98611
51	14.305	13.595	12.48	12.34	11.53	3	-0.99821
52	12.075	13.135	13.38	12.38	12.06	3	-0.99998
55	13.145	14.225	14.675	14.095	14.07	3	-0.9894
56	11.815	12.85	14.06	14.07	14.035	3	-0.98705
57	13.19	12.765	12.28	12.41	13.57	3	-0.96457
58	10.96	11.96	13.23	14.01	13.87	3	-0.87211
59	14.59	13.875	13.145	13.43	14.095	3	-0.83961
60	13.805	13.935	13.835	13.1	14.125	3	-0.81923
61	16.765	16.145	15.895	14.885	14.62	3	-0.72023
62	15.465	15.905	15.77	15.325	14.725	3	-0.70085
ID	TMed3	TMed5	TMed7	TMed10	TMed15	SeasonID	seasonCos

ID	TMed3	TMed5	TMed7	TMed10	TMed15	SeasonID	seasonCos
63	15.92	16.315	15.92	15.625	14.935	3	-0.69109
64	13.11	13.97	14.425	14.74	14.98	3	-0.60659
65	12.885	12.865	13.325	13.695	14.355	3	-0.56486
66	13.28	13.115	13.07	13.43	14.09	3	-0.53981
67	13.275	12.88	12.935	13.33	13.99	4	-0.52893
68	8.013	8.34	9.12	10.375	10.61	4	-0.13948
70	10.065	9.585	9.435	9.345	9.505	4	-0.02048
71	10.68	10.3	9.965	9.605	9.47	4	0.013859
72	10.77	10.305	10.15	9.625	9.45	4	0.022442
73	8.545	9.565	9.745	9.425	9.315	4	0.06032
74	3.494	5.242	6.884	8.135	8.865	4	0.152751
75	4.66	6.6015	6.9135	6.1555	7.6985	4	0.246982
76	5.1905	5.1905	6.2085	5.652	7.019	4	0.268408
77	5.324	4.2125	6.2215	7.1155	7.3135	4	0.477168
79	4.442	4.9635	5.199	5.184	6.7005	4	0.534836
80	3.272	4.4645	5.1135	4.511	6.2985	4	0.545371
81	0.915	2.2095	3.2825	4.023	4.957	4	0.578502
82	8.12	7.58	5.238	5.3805	4.957	4	0.619727
84	8.49	8.73	6.9155	7.7525	6.1345	4	0.715872
85	4.989	5.643	6.475	6.034	6.14	4	0.73942
87	5.5335	5.438	6.7745	6.097	6.97	4	0.756276
88	8.2	6.3675	5.99	7.617	6.927	4	0.778515
89	6.3725	6.7035	5.99	6.5545	6.4325	4	0.79071
90	4.6555	6.1335	5.8745	5.7885	5.9305	4	0.804728
91	3.893	4.7315	5.8705	5.2575	5.8745	1	0.815629
92	-1.015	-1.012	0.841	1.837	3.8275	1	0.880979
93	0.94	-0.018	0.686	2.398	3.5395	1	0.890279
94	0.482	0.204	-0.2825	0.612	2.474	1	0.90554
95	-1.1015	-0.368	0.2915	0.309	0.204	1	0.955206
96	0.017	-0.368	0.6305	0.668	0.6085	1	0.959644
98	4.888	3.9625	0.8555	2.1885	1.3795	1	0.969614
99	4.589	4.741	4.3225	2.967	2.203	1	0.977892
100	4.072	4.1245	4.325	4.0775	3.7965	1	0.986671
101	4.1775	4.276	4.327	4.2735	3.9445	1	0.988684
102	6.373	4.378	4.417	4.6475	4.026	1	0.99204
103	5.225	5.4435	4.8615	4.7005	4.412	1	0.996037
104	4.6135	5.1145	4.956	4.654	4.47	1	0.997928
106	4.9945	5.0105	5.5175	4.8415	4.7755	1	0.999139
107	4.64	4.956	5.4305	4.8615	4.801	1	0.999411
110	3.467	1.197	-0.121	-0.321	-0.506	1	0.88936
111	4.2845	3.4605	1.197	-0.1195	-0.4715	1	0.881358
112	6.1265	4.1705	3.3615	0.1535	-0.382	1	0.873967
ID	TMed3	TMed5	TMed7	TMed10	TMed15	SeasonID	seasonCos
113	6.1265	5.0865	3.891	1.3185	-0.273	1	0.868332

ID	TMed3	TMed5	TMed7	TMed10	TMed15	SeasonID	seasonCos
114	3.185	3.9775	5.0305	4.762	3.256	1	0.819817
115	0.538	0.695	0.6785	1.169	1.7375	1	0.730509
116	3.58	3.517	3.491	3.4855	2.2235	2	0.40471
119	4.4105	3.708	3.1455	2.7855	2.191	2	-0.19923
120	8.09	5.2655	4.0115	3.4545	2.557	2	-0.21782
121	9.35	7.7995	5.1745	4.164	3.048	2	-0.23389
122	8.71	8.485	7.7845	5.2655	3.936	2	-0.25753
123	7.6235	8.455	8.185	6.9895	4.934	2	-0.28721
124	8.86	7.9665	7.5295	7.925	7.6235	2	-0.38577
125	6.4335	7.082	7.4205	7.898	7.9685	2	-0.67322
132	16.63	17.025	17.25	16.895	17.34	3	-0.93377
133	17.59	16.835	17.435	17.26	17.085	3	-0.9249
134	16.53	16.96	16.84	17.225	17.085	3	-0.90556
135	15.205	15.7	16.055	16.39	16.615	3	-0.877
136	15.34	16.095	15.905	16.02	16.25	3	-0.83502
139	15.68	15.015	14.305	14.21	14.4	3	-0.70237
143	13.665	14.05	13.835	13.885	14.085	4	-0.4251
144	10.535	10.975	10.655	10.36	11.67	4	-0.27999
145	10.505	10.88	11.25	12.03	10.855	4	-0.0026
146	13.645	12.615	12.285	12.175	12.17	4	0.045443
147	8.82	9.165	8.7	9.425	10.755	4	0.252159
148	8.915	8.67	9.15	8.83	9.865	4	0.294931
149	11.135	10.165	9.43	9.26	9.865	4	0.316443
150	11.22	10.8	10.19	9.65	9.87	4	0.333403
151	11.475	11.135	10.62	9.87	9.87	4	0.342179
152	11.755	11.39	10.9	10.24	9.87	4	0.352269
153	12.46	11.76	11.305	10.5	9.87	4	0.362653
155	10.555	11.315	11.415	10.86	9.92	4	0.400442
156	11.11	11.675	11.525	11.285	10.44	4	0.417786
157	11.11	11.36	11.45	11.285	10.455	4	0.424617
158	7.6875	9.17	9.17	10.47	10.36	4	0.47679
159	7.89	7.4435	8.18	8.79	10.18	4	0.52
160	6.8195	6.989	7.6115	8.725	9.78	4	0.530377
161	4.8805	5.8025	6.309	7.3735	8.705	4	0.559543
162	5.4765	5.5455	5.6515	6.141	7.481	4	0.590929
163	7	5.2015	5.8945	5.4885	5.954	4	0.673614
164	6.946	6.704	6.9455	6.291	5.8645	4	0.722623
166	6.0055	6.039	5.996	6.0585	4.649	1	0.895668
167	3.085	5.403	5.8975	5.8435	5.686	1	0.918994
168	7.093	7.9215	7.115	6.5365	6.286	1	0.947671
169	8.08	7.517	7.794	7.314	6.68	1	0.961816
ID	TMed3	TMed5	TMed7	TMed10	TMed15	SeasonID	seasonCos
170	4.66	5.8155	6.4965	7.421	6.4895	1	0.972031
171	4.0965	5.2965	6.471	6.753	6.4525	1	0.977012

ID	TMed3	TMed5	TMed7	TMed10	TMed15	SeasonID	seasonCos
172	3.7425	4.415	5.0575	6.1335	6.337	1	0.980888
173	6.106	4.45	5.4905	6.3285	6.897	1	0.984346
174	5.1735	4.86	4.309	5.068	6.514	1	0.989156
175	5.037	4.86	4.436	5.0495	6.4905	1	0.990383
176	4.12	4.917	4.5535	4.952	6.048	1	0.992221
177	3.297	3.5145	4.022	3.8155	4.9685	1	0.996267
178	3.297	3.445	4.3765	3.892	4.9685	1	0.996775
179	4.6215	4.1025	4.0985	4.3205	4.477	1	0.999344
181	5.0045	4.8955	4.9175	4.499	4.5805	1	0.999432
182	5.536	5.52	4.9505	4.8075	4.8665	1	0.998703
183	5	4.8475	4.909	4.8435	4.5105	1	0.99686
184	4.7115	5.2265	5.112	4.911	4.4625	1	0.99611
185	4.7115	5.396	5.112	4.911	4.5805	1	0.995521
186	6.2215	5.679	5.372	5.128	4.8665	1	0.994101
187	7.154	6.67	5.984	5.7425	5.1185	1	0.990679
188	5.257	6.67	5.3615	5.4405	5.2645	1	0.985947
191	4.936	4.374	3.938	4.1935	4.593	1	0.955492
192	4.817	4.768	4.346	4.245	4.548	1	0.951714
193	3.969	4.275	4.4865	4.0965	4.4095	1	0.937125
194	4.5555	4.497	4.53	4.462	4.3685	1	0.927399
195	4.29	3.759	4.061	4.438	4.1635	1	0.918002
196	4.3835	4.4055	4.399	4.511	4.2305	1	0.912218
197	3.3045	4.3865	3.9945	4.3745	4.1445	1	0.906247
199	2.2575	2.734	3.231	3.4255	3.795	1	0.865112
200	1.902	2.8275	3.1215	3.3045	3.738	1	0.860032
201	1.671	2.605	2.901	3.1375	3.621	1	0.856535
202	4.366	4.369	3.7645	3.701	3.821	1	0.818583
203	4.395	4.305	4.4405	3.8315	3.766	1	0.793549
204	4.721	4.152	4.554	3.956	3.8315	1	0.786956
205	4.4035	4.6215	4.4745	4.4255	3.8425	1	0.773952
206	4.235	4.036	4.218	4.3975	3.849	1	0.75898
208	2.247	2.7595	3.1045	3.6165	3.572	1	0.730509
209	2.9215	2.7595	3.3125	3.6465	3.6955	1	0.719641
211	6.2205	5.557	4.6055	3.7365	4.082	1	0.650501
213	5.218	5.662	5.509	5.1065	4.3365	1	0.610426
215	5.5695	5.8885	5.8875	5.495	4.447	1	0.598426
216	6.327	5.955	5.9715	5.7535	4.58	1	0.584545
217	5.7835	5.925	5.821	5.843	5.2675	2	0.555971
218	3.2685	4.628	5.2915	5.2675	5.362	2	0.512629
219	3.1115	4.2655	4.861	5.0815	5.2585	2	0.503671
ID	TMed3	TMed5	TMed7	TMed10	TMed15	SeasonID	seasonCos
221	7.6815	7.9555	7.7575	7.5305	7.457	2	0.217998
223	6.6105	5.2725	5.155	5.0105	5.563	2	0.022145
224	6.7825	6.6655	5.914	5.506	5.3725	2	-0.01802

225	6.3905	6.7575	6.3	5.694	5.37	2	-0.03451
226	7.3615	6.8845	6.992	6.3	5.6125	2	-0.0646
227	9.375	7.5765	7.514	6.992	6.056	2	-0.08927
228	9.03	8.93	8.76	8.765	8.41	2	-0.37249
230	9.47	9.475	9.105	9.13	8.99	2	-0.5931
231	9.235	9.36	9.41	8.84	9.07	2	-0.62383
232	8.91	9.315	9.315	9.07	9.02	2	-0.64188
233	8.43	9.62	10.285	11.06	10.135	2	-0.78927
234	11.315	10.07	9.96	10.68	10.95	3	-0.82156
235	11.86	11.63	11.86	11.045	10.815	3	-0.90933
236	13.325	12.18	12.045	12.065	11.225	3	-0.92242
237	13.945	14.075	13.54	12.685	12.225	3	-0.94761
239	14.745	13.915	12.995	13.035	13.555	3	-0.99976
241	14.365	14.38	14.885	15.425	15.73	3	-0.8686
243	13.545	13.65	13.98	14.075	14.745	3	-0.82744
244	14.555	13.9	13.8	14.05	14.32	3	-0.80529
245	12.775	13.765	13.72	13.615	13.98	3	-0.78106
246	12.235	12.67	13.295	13.23	13.68	3	-0.76996
247	12.21	12.23	12.29	12.89	13.325	3	-0.73793

Appendix 3: SPSS output

Regression NAverageC

```
REGRESSION
  /MISSING LISTWISE
  /STATISTICS COEFF OUTS R ANOVA
  /CRITERIA=PIN(.05) POUT(.10)
  /NOORIGIN
  /DEPENDENT NAverageConc
  /METHOD=STEPWISE TimeSinceLastPeak TotalRainfall stormAPI StartQ PeakQ
QpMinusQo TotalQ seasonCos TMedian3 TMedian5 TMedian7 TMedian10 TMedian15.
```

Notes

Output Created		24-FEB-2016 13:56:00
Comments		
Input	Data	C:\Users\dg\Documents\Durham University\EdenDTC\1.Final Results\Stepwise Final\data.sav
	Active Dataset	DataSet1
	Filter	<none>
	Weight	<none>
	Split File	<none>
	N of Rows in Working Data File	190
Missing Value Handling	Definition of Missing	User-defined missing values are treated as missing.
	Cases Used	Statistics are based on cases with no missing values for any variable used.
Syntax		REGRESSION /MISSING LISTWISE /STATISTICS COEFF OUTS R ANOVA /CRITERIA=PIN(.05) POUT(.10) /NOORIGIN /DEPENDENT NAverageConc /METHOD=STEPWISE TimeSinceLastPeak TotalRainfall stormAPI StartQ PeakQ QpMinusQo TotalQ seasonCos TMedian3 TMedian5 TMedian7 TMedian10 TMedian15.
Resources	Processor Time	00:00:00.06
	Elapsed Time	00:00:00.06
	Memory Required	14336 bytes
	Additional Memory Required for Residual Plots	0 bytes

```
[DataSet1] C:\Users\dg\Documents\Durham University\EdenDTC\1.Final
Results\Stepwise Final\data.sav
```

Variables Entered/Removed[a]

Model	Variables Entered	Variables Removed	Method
1	seasonCos	.	Stepwise (Criteria: Probability-of-F-to-enter <= .050, Probability-of-F-to-remove >= .100).

a. Dependent Variable: NAverageConc

Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.450[a]	.202	.198	2.23592007609167

a. Predictors: (Constant), seasonCos

ANOVA[a]

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	236.744	1	236.744	47.355	.000[b]
	Residual	934.876	187	4.999		
	Total	1171.620	188			

a. Dependent Variable: NAverageConc

b. Predictors: (Constant), seasonCos

Coefficients[a]

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	8.974	.177		50.593	.000
	seasonCos	-1.604	.233	-.450	-6.882	.000

a. Dependent Variable: NAverageConc

Excluded Variables[a]

Model		Beta In	t	Sig.	Partial Correlation	Collinearity Statistics
						Tolerance
1	TimeSinceLastPeak	.060[b]	.875	.382	.064	.912
	TotalRainfall	.069[b]	1.021	.308	.075	.923
	stormAPI	-.034[b]	-.492	.623	-.036	.883
	StartQ	-.043[b]	-.608	.544	-.045	.841
	PeakQ	-.007[b]	-.109	.914	-.008	.951
	QpMinusQo	.003[b]	.046	.963	.003	.985
	TotalQ	.021[b]	.317	.752	.023	.978
	TMedian3	.172[b]	1.662	.098	.121	.393
	TMedian5	.057[b]	.518	.605	.038	.359
	TMedian7	-.006[b]	-.058	.953	-.004	.350
	TMedian10	-.037[b]	-.339	.735	-.025	.367
	TMedian15	-.080[b]	-.764	.446	-.056	.386

a. Dependent Variable: NAverageConc

b. Predictors in the Model: (Constant), seasonCos

Regression NAverageC CosOUT

```
REGRESSION
  /MISSING LISTWISE
  /STATISTICS COEFF OUTS R ANOVA
  /CRITERIA=PIN(.05) POUT(.10)
  /NOORIGIN
  /DEPENDENT NAverageConc
  /METHOD=STEPWISE TimeSinceLastPeak TotalRainfall stormAPI StartQ PeakQ
QpMinusQo TotalQ TMedian3 TMedian5 TMedian7 TMedian10 TMedian15.
```

Notes

Output Created		24-FEB-2016 13:57:05
Comments		
Input	Data	C:\Users\dg\Documents\Durham University\EdenDTC\1.Final Results\Stepwise Final\data.sav
	Active Dataset	DataSet1
	Filter	<none>
	Weight	<none>
	Split File	<none>
	N of Rows in Working Data File	190
Missing Value Handling	Definition of Missing	User-defined missing values are treated as missing.
	Cases Used	Statistics are based on cases with no missing values for any variable used.
Syntax		REGRESSION /MISSING LISTWISE /STATISTICS COEFF OUTS R ANOVA /CRITERIA=PIN(.05) POUT(.10) /NOORIGIN /DEPENDENT NAverageConc /METHOD=STEPWISE TimeSinceLastPeak TotalRainfall stormAPI StartQ PeakQ QpMinusQo TotalQ TMedian3 TMedian5 TMedian7 TMedian10 TMedian15.
Resources	Processor Time	00:00:00.03
	Elapsed Time	00:00:00.04
	Memory Required	13056 bytes
	Additional Memory Required for Residual Plots	0 bytes

```
[DataSet1] C:\Users\dg\Documents\Durham University\EdenDTC\1.Final
Results\Stepwise Final\data.sav
```

137

Variables Entered/Removed[a]

Model	Variables Entered	Variables Removed	Method
1	TMedian3	.	Stepwise (Criteria: Probability-of-F-to-enter <= .050, Probability-of-F-to-remove >= .100).
2	TimeSinceLastPeak	.	Stepwise (Criteria: Probability-of-F-to-enter <= .050, Probability-of-F-to-remove >= .100).

a. Dependent Variable: NAverageConc

Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.418[a]	.175	.170	2.27403945087828
2	.446[b]	.199	.190	2.24613886489324

a. Predictors: (Constant), TMedian3

b. Predictors: (Constant), TMedian3, TimeSinceLastPeak

ANOVA[a]

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	204.596	1	204.596	39.564	.000[b]
	Residual	967.025	187	5.171		
	Total	1171.620	188			
2	Regression	233.224	2	116.612	23.114	.000[c]
	Residual	938.396	186	5.045		
	Total	1171.620	188			

a. Dependent Variable: NAverageConc

b. Predictors: (Constant), TMedian3

c. Predictors: (Constant), TMedian3, TimeSinceLastPeak

Coefficients[a]

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	6.591	.344		19.176	.000
	TMedian3	.254	.040	.418	6.290	.000
2	(Constant)	6.346	.355		17.884	.000
	TMedian3	.246	.040	.406	6.164	.000
	TimeSinceLastPeak	.076	.032	.157	2.382	.018

a. Dependent Variable: NAverageConc

Excluded Variables[a]

Model		Beta In	t	Sig.	Partial Correlation	Collinearity Statistics
						Tolerance
1	TimeSinceLastPeak	.157[b]	2.382	.018	.172	.994
	TotalRainfall	.094[b]	1.375	.171	.100	.942
	stormAPI	-.122[b]	-1.831	.069	-.133	.976
	StartQ	-.130[b]	-1.925	.056	-.140	.952
	PeakQ	-.063[b]	-.938	.349	-.069	.989
	QpMinusQo	-.031[b]	-.463	.644	-.034	.998
	TotalQ	-.001[b]	-.008	.994	-.001	.988
	TMedian5	-.462[b]	-1.650	.101	-.120	.056
	TMedian7	-.254[b]	-1.342	.181	-.098	.123
	TMedian10	-.171[b]	-1.119	.265	-.082	.188
	TMedian15	-.157[b]	-1.190	.236	-.087	.254
2	TotalRainfall	.056[c]	.798	.426	.059	.879
	stormAPI	-.051[c]	-.635	.526	-.047	.680
	StartQ	-.077[c]	-1.033	.303	-.076	.785
	PeakQ	-.033[c]	-.484	.629	-.036	.950
	QpMinusQo	-.016[c]	-.243	.808	-.018	.988
	TotalQ	.030[c]	.449	.654	.033	.953
	TMedian5	-.513[c]	-1.853	.065	-.135	.056
	TMedian7	-.318[c]	-1.692	.092	-.123	.121
	TMedian10	-.248[c]	-1.616	.108	-.118	.181
	TMedian15	-.213[c]	-1.619	.107	-.118	.247

a. Dependent Variable: NAverageConc

b. Predictors in the Model: (Constant), TMedian3

c. Predictors in the Model: (Constant), TMedian3, TimeSinceLastPeak

Regression NFlux

```
REGRESSION
  /MISSING LISTWISE
  /STATISTICS COEFF OUTS R ANOVA
  /CRITERIA=PIN(.05) POUT(.10)
  /NOORIGIN
  /DEPENDENT TotalNFlux
  /METHOD=STEPWISE TimeSinceLastPeak TotalRainfall stormAPI QpMinusQo TMedian3
TMedian5 TMedian7 TMedian10 TMedian15 seasonCos.
```

Notes

Output Created		24-FEB-2016 14:04:52
Comments		
Input	Data	C:\Users\dg\Documents\Durham University\EdenDTC\1.Final Results\Stepwise Final\data.sav
	Active Dataset	DataSet1
	Filter	<none>
	Weight	<none>
	Split File	<none>
	N of Rows in Working Data File	190
Missing Value Handling	Definition of Missing	User-defined missing values are treated as missing.
	Cases Used	Statistics are based on cases with no missing values for any variable used.
Syntax		REGRESSION /MISSING LISTWISE /STATISTICS COEFF OUTS R ANOVA /CRITERIA=PIN(.05) POUT(.10) /NOORIGIN /DEPENDENT TotalNFlux /METHOD=STEPWISE TimeSinceLastPeak TotalRainfall stormAPI QpMinusQo TMedian3 TMedian5 TMedian7 TMedian10 TMedian15 seasonCos.
Resources	Processor Time	00:00:00.03
	Elapsed Time	00:00:00.07
	Memory Required	10688 bytes
	Additional Memory Required for Residual Plots	0 bytes

```
[DataSet1] C:\Users\dg\Documents\Durham University\EdenDTC\1.Final
Results\Stepwise Final\data.sav
```

Variables Entered/Removed[a]

Model	Variables Entered	Variables Removed	Method
1	TotalRainfall	.	Stepwise (Criteria: Probability-of-F-to-enter <= .050, Probability-of-F-to-remove >= .100).
2	TimeSinceLastPeak	.	Stepwise (Criteria: Probability-of-F-to-enter <= .050, Probability-of-F-to-remove >= .100).
3	QpMinusQo	.	Stepwise (Criteria: Probability-of-F-to-enter <= .050, Probability-of-F-to-remove >= .100).

a. Dependent Variable: TotalNFlux

Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.672[a]	.452	.449	632017.1686509916000
2	.720[b]	.518	.513	594116.4345780500000
3	.732[c]	.535	.528	584959.3668826522000

a. Predictors: (Constant), TotalRainfall

b. Predictors: (Constant), TotalRainfall, TimeSinceLastPeak

c. Predictors: (Constant), TotalRainfall, TimeSinceLastPeak, QpMinusQo

ANOVA[a]

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	61505335337566.375	1	61505335337566.375	153.977	.000[b]
	Residual	74696346174818.190	187	399445701469.616		
	Total	136201681512384.560	188			
2	Regression	70548454674937.950	2	35274227337468.980	99.934	.000[c]
	Residual	65653226837446.610	186	352974337835.734		
	Total	136201681512384.560	188			
3	Regression	72898851245190.220	3	24299617081730.074	71.015	.000[d]
	Residual	63302830267194.340	185	342177460903.753		
	Total	136201681512384.560	188			

a. Dependent Variable: TotalNFlux

b. Predictors: (Constant), TotalRainfall

c. Predictors: (Constant), TotalRainfall, TimeSinceLastPeak

d. Predictors: (Constant), TotalRainfall, TimeSinceLastPeak, QpMinusQo

Coefficients[a]

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	165243.573	76669.075		2.155	.032
	TotalRainfall	54841.491	4419.587	.672	12.409	.000
2	(Constant)	258620.275	74395.006		3.476	.001
	TotalRainfall	60716.339	4313.638	.744	14.075	.000
	TimeSinceLastPeak	-44022.979	8697.448	-.268	-5.062	.000
3	(Constant)	222041.968	74566.134		2.978	.003
	TotalRainfall	49948.389	5909.184	.612	8.453	.000
	TimeSinceLastPeak	-35197.094	9201.744	-.214	-3.825	.000
	QpMinusQo	92306.823	35219.966	.184	2.621	.009

a. Dependent Variable: TotalNFlux

Excluded Variables[a]

Model		Beta In	t	Sig.	Partial Correlation	Collinearity Statistics Tolerance
1	TimeSinceLastPeak	-.268[b]	-5.062	.000	-.348	.928
	stormAPI	.202[b]	3.863	.000	.273	.999
	QpMinusQo	.282[b]	4.170	.000	.292	.591
	TMedian3	-.083[b]	-1.493	.137	-.109	.942
	TMedian5	-.090[b]	-1.626	.106	-.118	.939
	TMedian7	-.096[b]	-1.733	.085	-.126	.936
	TMedian10	-.112[b]	-2.007	.046	-.146	.923
	TMedian15	-.116[b]	-2.087	.038	-.151	.926
	seasonCos	.106[b]	1.893	.060	.137	.923
2	stormAPI	.078[c]	1.254	.211	.092	.661
	QpMinusQo	.184[c]	2.621	.009	.189	.512
	TMedian3	-.079[c]	-1.519	.130	-.111	.942
	TMedian5	-.083[c]	-1.589	.114	-.116	.938
	TMedian7	-.082[c]	-1.567	.119	-.114	.934
	TMedian10	-.091[c]	-1.714	.088	-.125	.917
	TMedian15	-.095[c]	-1.806	.073	-.132	.920
	seasonCos	.044[c]	.806	.421	.059	.870
3	stormAPI	.038[d]	.588	.558	.043	.613
	TMedian3	-.044[d]	-.817	.415	-.060	.864
	TMedian5	-.046[d]	-.855	.394	-.063	.852

TMedian7	-.045[d]	-.834	.406	-.061	.849
TMedian10	-.055[d]	-1.013	.313	-.074	.839
TMedian15	-.061[d]	-1.120	.264	-.082	.844
seasonCos	-.006[d]	-.107	.915	-.008	.762

a. Dependent Variable: TotalNFlux

b. Predictors in the Model: (Constant), TotalRainfall

c. Predictors in the Model: (Constant), TotalRainfall, TimeSinceLastPeak

d. Predictors in the Model: (Constant), TotalRainfall, TimeSinceLastPeak, QpMinusQo

Regression NFluxCosOUT

```
REGRESSION
  /MISSING LISTWISE
  /STATISTICS COEFF OUTS R ANOVA
  /CRITERIA=PIN(.05) POUT(.10)
  /NOORIGIN
  /DEPENDENT TotalNFlux
  /METHOD=STEPWISE TimeSinceLastPeak TotalRainfall stormAPI QpMinusQo TMedian3
TMedian5 TMedian7 TMedian10 TMedian15.
```

Notes

Output Created		24-FEB-2016 14:05:19
Comments		
Input	Data	C:\Users\dg\Documents\Durham University\EdenDTC\1.Final Results\Stepwise Final\data.sav
	Active Dataset	DataSet1
	Filter	<none>
	Weight	<none>
	Split File	<none>
	N of Rows in Working Data File	190
Missing Value Handling	Definition of Missing	User-defined missing values are treated as missing.
	Cases Used	Statistics are based on cases with no missing values for any variable used.
Syntax		REGRESSION /MISSING LISTWISE /STATISTICS COEFF OUTS R ANOVA /CRITERIA=PIN(.05) POUT(.10) /NOORIGIN /DEPENDENT TotalNFlux /METHOD=STEPWISE TimeSinceLastPeak TotalRainfall stormAPI QpMinusQo TMedian3 TMedian5 TMedian7 TMedian10 TMedian15.
Resources	Processor Time	00:00:00.05
	Elapsed Time	00:00:00.04
	Memory Required	9600 bytes
	Additional Memory Required for Residual Plots	0 bytes

```
[DataSet1] C:\Users\dg\Documents\Durham University\EdenDTC\1.Final
Results\Stepwise Final\data.sav
```

144

Variables Entered/Removed[a]

Model	Variables Entered	Variables Removed	Method
1	TotalRainfall	.	Stepwise (Criteria: Probability-of-F-to-enter <= .050, Probability-of-F-to-remove >= .100).
2	TimeSinceLastPeak	.	Stepwise (Criteria: Probability-of-F-to-enter <= .050, Probability-of-F-to-remove >= .100).
3	QpMinusQo	.	Stepwise (Criteria: Probability-of-F-to-enter <= .050, Probability-of-F-to-remove >= .100).

a. Dependent Variable: TotalNFlux

Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.672[a]	.452	.449	632017.1686509916000
2	.720[b]	.518	.513	594116.4345780500000
3	.732[c]	.535	.528	584959.3668826522000

a. Predictors: (Constant), TotalRainfall

b. Predictors: (Constant), TotalRainfall, TimeSinceLastPeak

c. Predictors: (Constant), TotalRainfall, TimeSinceLastPeak, QpMinusQo

ANOVA[a]

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	61505335337566.375	1	61505335337566.375	153.977	.000[b]
	Residual	74696346174818.190	187	399445701469.616		
	Total	136201681512384.560	188			
2	Regression	70548454674937.950	2	35274227337468.980	99.934	.000[c]
	Residual	65653226837446.610	186	352974337835.734		
	Total	136201681512384.560	188			
3	Regression	72898851245190.220	3	24299617081730.074	71.015	.000[d]
	Residual	63302830267194.340	185	342177460903.753		
	Total	136201681512384.560	188			

a. Dependent Variable: TotalNFlux

145

b. Predictors: (Constant), TotalRainfall

c. Predictors: (Constant), TotalRainfall, TimeSinceLastPeak

d. Predictors: (Constant), TotalRainfall, TimeSinceLastPeak, QpMinusQo

Coefficients^a

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	165243.573	76669.075		2.155	.032
	TotalRainfall	54841.491	4419.587	.672	12.409	.000
2	(Constant)	258620.275	74395.006		3.476	.001
	TotalRainfall	60716.339	4313.638	.744	14.075	.000
	TimeSinceLastPeak	-44022.979	8697.448	-.268	-5.062	.000
3	(Constant)	222041.968	74566.134		2.978	.003
	TotalRainfall	49948.389	5909.184	.612	8.453	.000
	TimeSinceLastPeak	-35197.094	9201.744	-.214	-3.825	.000
	QpMinusQo	92306.823	35219.966	.184	2.621	.009

a. Dependent Variable: TotalNFlux

Excluded Variables^a

Model		Beta In	t	Sig.	Partial Correlation	Collinearity Statistics Tolerance
1	TimeSinceLastPeak	-.268^b	-5.062	.000	-.348	.928
	stormAPI	.202^b	3.863	.000	.273	.999
	QpMinusQo	.282^b	4.170	.000	.292	.591
	TMedian3	-.083^b	-1.493	.137	-.109	.942
	TMedian5	-.090^b	-1.626	.106	-.118	.939
	TMedian7	-.096^b	-1.733	.085	-.126	.936
	TMedian10	-.112^b	-2.007	.046	-.146	.923
	TMedian15	-.116^b	-2.087	.038	-.151	.926
2	stormAPI	.078^c	1.254	.211	.092	.661
	QpMinusQo	.184^c	2.621	.009	.189	.512
	TMedian3	-.079^c	-1.519	.130	-.111	.942
	TMedian5	-.083^c	-1.589	.114	-.116	.938
	TMedian7	-.082^c	-1.567	.119	-.114	.934
	TMedian10	-.091^c	-1.714	.088	-.125	.917
	TMedian15	-.095^c	-1.806	.073	-.132	.920
3	stormAPI	.038^d	.588	.558	.043	.613

TMedian3	-.044[d]	-.817	.415	-.060	.864
TMedian5	-.046[d]	-.855	.394	-.063	.852
TMedian7	-.045[d]	-.834	.406	-.061	.849
TMedian10	-.055[d]	-1.013	.313	-.074	.839
TMedian15	-.061[d]	-1.120	.264	-.082	.844

a. Dependent Variable: TotalNFlux

b. Predictors in the Model: (Constant), TotalRainfall

c. Predictors in the Model: (Constant), TotalRainfall, TimeSinceLastPeak

d. Predictors in the Model: (Constant), TotalRainfall, TimeSinceLastPeak, QpMinusQo

Regression HMagnitude

```
REGRESSION
  /MISSING LISTWISE
  /STATISTICS COEFF OUTS R ANOVA
  /CRITERIA=PIN(.05) POUT(.10)
  /NOORIGIN
  /DEPENDENT HMagnitude
  /METHOD=STEPWISE TimeSinceLastPeak TotalRainfall stormAPI StartQ PeakQ
QpMinusQo TotalQ SeasonID TMedian3 TMedian5 TMedian7 TMedian10 TMedian15
seasonCos.
```

Notes

Output Created		01-MAR-2016 23:13:11
Comments		
Input	Data	C:\Users\dg\Documents\Durham University\EdenDTC\1.Final Results\Stepwise Final\data.sav
	Active Dataset	DataSet1
	Filter	<none>
	Weight	<none>
	Split File	<none>
	N of Rows in Working Data File	190
Missing Value Handling	Definition of Missing	User-defined missing values are treated as missing.
	Cases Used	Statistics are based on cases with no missing values for any variable used.
Syntax		REGRESSION /MISSING LISTWISE /STATISTICS COEFF OUTS R ANOVA /CRITERIA=PIN(.05) POUT(.10) /NOORIGIN /DEPENDENT HMagnitude /METHOD=STEPWISE TimeSinceLastPeak TotalRainfall stormAPI StartQ PeakQ QpMinusQo TotalQ SeasonID TMedian3 TMedian5 TMedian7 TMedian10 TMedian15 seasonCos.
Resources	Processor Time	00:00:00.05
	Elapsed Time	00:00:00.05
	Memory Required	15712 bytes
	Additional Memory Required for Residual Plots	0 bytes

Variables Entered/Removed[a]

Model	Variables Entered	Variables Removed	Method
1	TimeSinceLastPeak	.	Stepwise (Criteria: Probability-of-F-to-enter <= .050, Probability-of-F-to-remove >= .100).
2	seasonCos	.	Stepwise (Criteria: Probability-of-F-to-enter <= .050, Probability-of-F-to-remove >= .100).
3	SeasonID	.	Stepwise (Criteria: Probability-of-F-to-enter <= .050, Probability-of-F-to-remove >= .100).

a. Dependent Variable: HMagnitude

Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.434[a]	.188	.184	1.392950881954462
2	.507[b]	.257	.249	1.336128240565563
3	.530[c]	.281	.270	1.317814725344070

a. Predictors: (Constant), TimeSinceLastPeak

b. Predictors: (Constant), TimeSinceLastPeak, seasonCos

c. Predictors: (Constant), TimeSinceLastPeak, seasonCos, SeasonID

ANOVA[a]

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	84.230	1	84.230	43.410	.000[b]
	Residual	362.838	187	1.940		
	Total	447.068	188			
2	Regression	115.014	2	57.507	32.212	.000[c]
	Residual	332.054	186	1.785		
	Total	447.068	188			
3	Regression	125.790	3	41.930	24.144	.000[d]
	Residual	321.278	185	1.737		
	Total	447.068	188			

a. Dependent Variable: HMagnitude

b. Predictors: (Constant), TimeSinceLastPeak

c. Predictors: (Constant), TimeSinceLastPeak, seasonCos

d. Predictors: (Constant), TimeSinceLastPeak, seasonCos, SeasonID

Coefficients^a

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	.804	.128		6.283	.000
	TimeSinceLastPeak	.129	.020	.434	6.589	.000
2	(Constant)	1.084	.140		7.741	.000
	TimeSinceLastPeak	.105	.020	.352	5.324	.000
	seasonCos	-.606	.146	-.275	-4.153	.000
3	(Constant)	1.647	.265		6.222	.000
	TimeSinceLastPeak	.105	.019	.351	5.373	.000
	seasonCos	-.796	.163	-.361	-4.887	.000
	SeasonID	-.219	.088	-.178	-2.491	.014

a. Dependent Variable: HMagnitude

Excluded Variables^a

Model		Beta In	t	Sig.	Partial Correlation	Collinearity Statistics Tolerance
1	TotalRainfall	.087^b	1.271	.205	.093	.928
	stormAPI	-.045^b	-.567	.571	-.042	.693
	StartQ	-.067^b	-.924	.357	-.068	.820
	PeakQ	-.002^b	-.034	.973	-.003	.958
	QpMinusQo	.013^b	.202	.840	.015	.990
	TotalQ	-.089^b	-1.324	.187	-.097	.962
	SeasonID	-.014^b	-.209	.835	-.015	.982
	TMedian3	.170^b	2.620	.010	.189	.994
	TMedian5	.162^b	2.489	.014	.180	.991
	TMedian7	.149^b	2.264	.025	.164	.986
	TMedian10	.141^b	2.128	.035	.154	.977
	TMedian15	.127^b	1.922	.056	.140	.978
	seasonCos	-.275^b	-4.153	.000	-.291	.912
2	TotalRainfall	.030^c	.439	.661	.032	.885
	stormAPI	.027^c	.341	.733	.025	.659
	StartQ	.027^c	.363	.717	.027	.739
	PeakQ	.045^c	.681	.497	.050	.930
	QpMinusQo	.039^c	.616	.539	.045	.981

	TotalQ	-.064[c]	-.984	.327	-.072	.953
	SeasonID	-.178[c]	-2.491	.014	-.180	.765
	TMedian3	-.104[c]	-1.000	.319	-.073	.367
	TMedian5	-.153[c]	-1.410	.160	-.103	.335
	TMedian7	-.196[c]	-1.804	.073	-.132	.334
	TMedian10	-.193[c]	-1.838	.068	-.134	.359
	TMedian15	-.208[c]	-2.046	.042	-.149	.378
	TotalRainfall	.030[d]	.446	.656	.033	.885
	stormAPI	.011[d]	.139	.890	.010	.654
	StartQ	.015[d]	.207	.837	.015	.736
	PeakQ	.044[d]	.687	.493	.051	.930
	QpMinusQo	.042[d]	.661	.510	.049	.981
3	TotalQ	-.069[d]	-1.082	.281	-.080	.952
	TMedian3	-.010[d]	-.090	.929	-.007	.316
	TMedian5	-.054[d]	-.460	.646	-.034	.281
	TMedian7	-.098[d]	-.818	.414	-.060	.271
	TMedian10	-.093[d]	-.793	.429	-.058	.283
	TMedian15	-.110[d]	-.940	.349	-.069	.284

a. Dependent Variable: HMagnitude

b. Predictors in the Model: (Constant), TimeSinceLastPeak

c. Predictors in the Model: (Constant), TimeSinceLastPeak, seasonCos

d. Predictors in the Model: (Constant), TimeSinceLastPeak, seasonCos, SeasonID

Regression HMagnitude CosOUT

```
REGRESSION
  /MISSING LISTWISE
  /STATISTICS COEFF OUTS R ANOVA
  /CRITERIA=PIN(.05) POUT(.10)
  /NOORIGIN
  /DEPENDENT HMagnitude
  /METHOD=STEPWISE TimeSinceLastPeak TotalRainfall stormAPI StartQ PeakQ
QpMinusQo TotalQ SeasonID TMedian3 TMedian5 TMedian7 TMedian10 TMedian15.
```

Notes

Output Created		01-MAR-2016 23:17:38
Comments		
Input	Data	C:\Users\dg\Documents\Durham University\EdenDTC\1.Final Results\Stepwise Final\data.sav
	Active Dataset	DataSet1
	Filter	<none>
	Weight	<none>
	Split File	<none>
	N of Rows in Working Data File	190
Missing Value Handling	Definition of Missing	User-defined missing values are treated as missing.
	Cases Used	Statistics are based on cases with no missing values for any variable used.
Syntax		REGRESSION /MISSING LISTWISE /STATISTICS COEFF OUTS R ANOVA /CRITERIA=PIN(.05) POUT(.10) /NOORIGIN /DEPENDENT HMagnitude /METHOD=STEPWISE TimeSinceLastPeak TotalRainfall stormAPI StartQ PeakQ QpMinusQo TotalQ SeasonID TMedian3 TMedian5 TMedian7 TMedian10 TMedian15.
Resources	Processor Time	00:00:00.03
	Elapsed Time	00:00:00.08
	Memory Required	14368 bytes
	Additional Memory Required for Residual Plots	0 bytes

```
[DataSet1] C:\Users\dg\Documents\Durham University\EdenDTC\1.Final
Results\Stepwise Final\data.sav
```

Variables Entered/Removed[a]

Model	Variables Entered	Variables Removed	Method
1	TimeSinceLastPeak	.	Stepwise (Criteria: Probability-of-F-to-enter <= .050, Probability-of-F-to-remove >= .100).
2	TMedian3	.	Stepwise (Criteria: Probability-of-F-to-enter <= .050, Probability-of-F-to-remove >= .100).
3	SeasonID	.	Stepwise (Criteria: Probability-of-F-to-enter <= .050, Probability-of-F-to-remove >= .100).

a. Dependent Variable: HMagnitude

Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.434[a]	.188	.184	1.392950881954462
2	.466[b]	.217	.209	1.371610683843948
3	.486[c]	.236	.223	1.359024578350652

a. Predictors: (Constant), TimeSinceLastPeak

b. Predictors: (Constant), TimeSinceLastPeak, TMedian3

c. Predictors: (Constant), TimeSinceLastPeak, TMedian3, SeasonID

ANOVA[a]

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	84.230	1	84.230	43.410	.000[b]
	Residual	362.838	187	1.940		
	Total	447.068	188			
2	Regression	97.143	2	48.572	25.818	.000[c]
	Residual	349.925	186	1.881		
	Total	447.068	188			
3	Regression	105.383	3	35.128	19.019	.000[d]
	Residual	341.685	185	1.847		
	Total	447.068	188			

a. Dependent Variable: HMagnitude

b. Predictors: (Constant), TimeSinceLastPeak

c. Predictors: (Constant), TimeSinceLastPeak, TMedian3

d. Predictors: (Constant), TimeSinceLastPeak, TMedian3, SeasonID

Coefficients[a]

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	.804	.128		6.283	.000
	TimeSinceLastPeak	.129	.020	.434	6.589	.000
2	(Constant)	.342	.217		1.577	.117
	TimeSinceLastPeak	.125	.019	.421	6.468	.000
	TMedian3	.064	.024	.170	2.620	.010
3	(Constant)	.530	.232		2.280	.024
	TimeSinceLastPeak	.130	.019	.436	6.723	.000
	TMedian3	.100	.030	.266	3.377	.001
	SeasonID	-.207	.098	-.167	-2.112	.036

a. Dependent Variable: HMagnitude

Excluded Variables[a]

Model		Beta In	t	Sig.	Partial Correlation	Collinearity Statistics Tolerance
1	TotalRainfall	.087[b]	1.271	.205	.093	.928
	stormAPI	-.045[b]	-.567	.571	-.042	.693
	StartQ	-.067[b]	-.924	.357	-.068	.820
	PeakQ	-.002[b]	-.034	.973	-.003	.958
	QpMinusQo	.013[b]	.202	.840	.015	.990
	TotalQ	-.089[b]	-1.324	.187	-.097	.962
	SeasonID	-.014[b]	-.209	.835	-.015	.982
	TMedian3	.170[b]	2.620	.010	.189	.994
	TMedian5	.162[b]	2.489	.014	.180	.991
	TMedian7	.149[b]	2.264	.025	.164	.986
	TMedian10	.141[b]	2.128	.035	.154	.977
	TMedian15	.127[b]	1.922	.056	.140	.978
2	TotalRainfall	.049[c]	.707	.480	.052	.879
	stormAPI	-.018[c]	-.227	.821	-.017	.680
	StartQ	-.030[c]	-.405	.686	-.030	.785
	PeakQ	.014[c]	.206	.837	.015	.950
	QpMinusQo	.021[c]	.317	.752	.023	.988
	TotalQ	-.073[c]	-1.096	.274	-.080	.953
	SeasonID	-.167[c]	-2.112	.036	-.153	.659
	TMedian5	-.061[c]	-.220	.826	-.016	.056
	TMedian7	-.096[c]	-.514	.608	-.038	.121

	TMedian10	-.079[c]	-.517	.606	-.038	.181
	TMedian15	-.085[c]	-.649	.517	-.048	.247
	TotalRainfall	.044[d]	.635	.526	.047	.878
	stormAPI	-.039[d]	-.491	.624	-.036	.670
	StartQ	-.046[d]	-.633	.528	-.047	.777
	PeakQ	.009[d]	.133	.894	.010	.948
	QpMinusQo	.019[d]	.298	.766	.022	.988
3	TotalQ	-.076[d]	-1.158	.248	-.085	.952
	TMedian5	.035[d]	.128	.898	.009	.054
	TMedian7	-.004[d]	-.021	.984	-.002	.114
	TMedian10	.015[d]	.094	.925	.007	.166
	TMedian15	.014[d]	.100	.921	.007	.216

a. Dependent Variable: HMagnitude

b. Predictors in the Model: (Constant), TimeSinceLastPeak

c. Predictors in the Model: (Constant), TimeSinceLastPeak, TMedian3

d. Predictors in the Model: (Constant), TimeSinceLastPeak, TMedian3, SeasonID

Regression Hysteresis

```
REGRESSION
  /MISSING LISTWISE
  /STATISTICS COEFF OUTS R ANOVA
  /CRITERIA=PIN(.05) POUT(.10)
  /NOORIGIN
  /DEPENDENT Hysteresis
  /METHOD=STEPWISE TimeSinceLastPeak TotalRainfall stormAPI QpMinusQo TMedian3
TMedian5 TMedian7 TMedian10 TMedian15 seasonCos.
```

Notes

Output Created		24-FEB-2016 14:11:06
Comments		
Input	Data	C:\Users\dg\Documents\Durham University\EdenDTC\1.Final Results\Stepwise Final\data.sav
	Active Dataset	DataSet1
	Filter	<none>
	Weight	<none>
	Split File	<none>
	N of Rows in Working Data File	190
Missing Value Handling	Definition of Missing	User-defined missing values are treated as missing.
	Cases Used	Statistics are based on cases with no missing values for any variable used.

Syntax		REGRESSION
		  /MISSING LISTWISE
		  /STATISTICS COEFF OUTS R ANOVA
		  /CRITERIA=PIN(.05) POUT(.10)
		  /NOORIGIN
		  /DEPENDENT Hysteresis
		  /METHOD=STEPWISE TimeSinceLastPeak TotalRainfall stormAPI QpMinusQo TMedian3 TMedian5 TMedian7 TMedian10 TMedian15 seasonCos.

Resources	Processor Time	00:00:00.05
	Elapsed Time	00:00:00.07
	Memory Required	10688 bytes
	Additional Memory Required for Residual Plots	0 bytes

```
[DataSet1] C:\Users\dg\Documents\Durham University\EdenDTC\1.Final
Results\Stepwise Final\data.sav
```

Variables Entered/Removed[a]

Model	Variables Entered	Variables Removed	Method
1	seasonCos	.	Stepwise (Criteria: Probability-of-F-to-enter <= .050, Probability-of-F-to-remove >= .100).
2	TotalRainfall	.	Stepwise (Criteria: Probability-of-F-to-enter <= .050, Probability-of-F-to-remove >= .100).
3	stormAPI	.	Stepwise (Criteria: Probability-of-F-to-enter <= .050, Probability-of-F-to-remove >= .100).
4	TMedian15	.	Stepwise (Criteria: Probability-of-F-to-enter <= .050, Probability-of-F-to-remove >= .100).
5	.	seasonCos	Stepwise (Criteria: Probability-of-F-to-enter <= .050, Probability-of-F-to-remove >= .100).
6	QpMinusQo	.	Stepwise (Criteria: Probability-of-F-to-enter <= .050, Probability-of-F-to-remove >= .100).

a. Dependent Variable: Hysteresis

Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.467[a]	.218	.213	.848
2	.535[b]	.286	.278	.812
3	.566[c]	.320	.309	.795
4	.590[d]	.348	.334	.780
5	.588[e]	.346	.335	.779
6	.601[f]	.361	.347	.772

a. Predictors: (Constant), seasonCos

b. Predictors: (Constant), seasonCos, TotalRainfall

c. Predictors: (Constant), seasonCos, TotalRainfall, stormAPI

d. Predictors: (Constant), seasonCos, TotalRainfall, stormAPI, TMedian15

e. Predictors: (Constant), TotalRainfall, stormAPI, TMedian15

f. Predictors: (Constant), TotalRainfall, stormAPI, TMedian15, QpMinusQo

ANOVA[a]

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	37.394	1	37.394	52.022	.000[b]
	Residual	134.416	187	.719		
	Total	171.810	188			
2	Regression	49.136	2	24.568	37.251	.000[c]
	Residual	122.673	186	.660		
	Total	171.810	188			
3	Regression	54.974	3	18.325	29.016	.000[d]
	Residual	116.835	185	.632		
	Total	171.810	188			
4	Regression	59.832	4	14.958	24.579	.000[e]
	Residual	111.977	184	.609		
	Total	171.810	188			
5	Regression	59.416	3	19.805	32.600	.000[f]
	Residual	112.394	185	.608		
	Total	171.810	188			
6	Regression	62.039	4	15.510	25.998	.000[g]
	Residual	109.770	184	.597		
	Total	171.810	188			

a. Dependent Variable: Hysteresis

b. Predictors: (Constant), seasonCos

c. Predictors: (Constant), seasonCos, TotalRainfall

d. Predictors: (Constant), seasonCos, TotalRainfall, stormAPI

e. Predictors: (Constant), seasonCos, TotalRainfall, stormAPI, TMedian15

f. Predictors: (Constant), TotalRainfall, stormAPI, TMedian15

g. Predictors: (Constant), TotalRainfall, stormAPI, TMedian15, QpMinusQo

Coefficients^a

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	.108	.067		1.606	.110
	seasonCos	.637	.088	.467	7.213	.000
2	(Constant)	.486	.110		4.404	.000
	seasonCos	.534	.088	.391	6.061	.000
	TotalRainfall	-.025	.006	-.272	-4.220	.000
3	(Constant)	.179	.148		1.213	.227
	seasonCos	.432	.093	.316	4.668	.000
	TotalRainfall	-.027	.006	-.298	-4.676	.000
	stormAPI	.009	.003	.198	3.040	.003
4	(Constant)	.671	.226		2.962	.003
	seasonCos	.119	.143	.087	.827	.409
	TotalRainfall	-.026	.006	-.286	-4.575	.000
	stormAPI	.011	.003	.239	3.649	.000
	TMedian15	-.065	.023	-.279	-2.825	.005
5	(Constant)	.786	.178		4.421	.000
	TotalRainfall	-.027	.006	-.294	-4.747	.000
	stormAPI	.013	.003	.261	4.335	.000
	TMedian15	-.080	.015	-.342	-5.474	.000
6	(Constant)	.757	.177		4.280	.000
	TotalRainfall	-.015	.008	-.168	-1.965	.051
	stormAPI	.015	.003	.314	4.848	.000
	TMedian15	-.088	.015	-.377	-5.880	.000
	QpMinusQo	-.103	.049	-.183	-2.097	.037

a. Dependent Variable: Hysteresis

Excluded Variables[a]

Model		Beta In	t	Sig.	Partial Correlation	Collinearity Statistics
						Tolerance
1	TimeSinceLastPeak	-.122[b]	-1.806	.073	-.131	.912
	TotalRainfall	-.272[b]	-4.220	.000	-.296	.923
	stormAPI	.158[b]	2.318	.022	.168	.883
	QpMinusQo	-.229[b]	-3.623	.000	-.257	.985
	TMedian3	-.190[b]	-1.852	.066	-.135	.393
	TMedian5	-.204[b]	-1.898	.059	-.138	.359
	TMedian7	-.213[b]	-1.958	.052	-.142	.350
	TMedian10	-.248[b]	-2.354	.020	-.170	.367
	TMedian15	-.235[b]	-2.287	.023	-.165	.386
2	TimeSinceLastPeak	-.069[c]	-1.038	.301	-.076	.874
	stormAPI	.198[c]	3.040	.003	.218	.868
	QpMinusQo	-.085[c]	-.967	.335	-.071	.494
	TMedian3	-.173[c]	-1.763	.079	-.129	.393
	TMedian5	-.185[c]	-1.796	.074	-.131	.358
	TMedian7	-.191[c]	-1.834	.068	-.134	.349
	TMedian10	-.208[c]	-2.044	.042	-.149	.363
	TMedian15	-.199[c]	-1.999	.047	-.145	.383
3	TimeSinceLastPeak	.056[d]	.720	.472	.053	.612
	QpMinusQo	-.187[d]	-2.074	.039	-.151	.444
	TMedian3	-.236[d]	-2.432	.016	-.176	.379
	TMedian5	-.263[d]	-2.574	.011	-.186	.342
	TMedian7	-.266[d]	-2.577	.011	-.187	.334
	TMedian10	-.278[d]	-2.760	.006	-.199	.349
	TMedian15	-.279[d]	-2.825	.005	-.204	.364
4	TimeSinceLastPeak	.045[e]	.586	.559	.043	.611
	QpMinusQo	-.200[e]	-2.261	.025	-.165	.443
	TMedian3	-.105[e]	-.842	.401	-.062	.226
	TMedian5	-.106[e]	-.674	.501	-.050	.145
	TMedian7	-.061[e]	-.299	.766	-.022	.086
	TMedian10	-.083[e]	-.260	.795	-.019	.035
5	TimeSinceLastPeak	.040[f]	.532	.596	.039	.613
	QpMinusQo	-.183[f]	-2.097	.037	-.153	.455
	TMedian3	-.126[f]	-1.066	.288	-.078	.252
	TMedian5	-.135[f]	-.915	.361	-.067	.164
	TMedian7	-.104[f]	-.539	.590	-.040	.095
	TMedian10	-.128[f]	-.407	.685	-.030	.036

	seasonCos	.087^f	.827	.409	.061	.322
6	TimeSinceLastPeak	.013^g	.175	.861	.013	.595
	TMedian3	-.129^g	-1.096	.274	-.081	.252
	TMedian5	-.144^g	-.987	.325	-.073	.163
	TMedian7	-.114^g	-.595	.553	-.044	.095
	TMedian10	-.128^g	-.413	.680	-.030	.036
	seasonCos	.125^g	1.186	.237	.087	.314

a. Dependent Variable: Hysteresis

b. Predictors in the Model: (Constant), seasonCos

c. Predictors in the Model: (Constant), seasonCos, TotalRainfall

d. Predictors in the Model: (Constant), seasonCos, TotalRainfall, stormAPI

e. Predictors in the Model: (Constant), seasonCos, TotalRainfall, stormAPI, TMedian15

f. Predictors in the Model: (Constant), TotalRainfall, stormAPI, TMedian15

g. Predictors in the Model: (Constant), TotalRainfall, stormAPI, TMedian15, QpMinusQo

Regression Hysteresis CosOUT

```
REGRESSION
  /MISSING LISTWISE
  /STATISTICS COEFF OUTS R ANOVA
  /CRITERIA=PIN(.05) POUT(.10)
  /NOORIGIN
  /DEPENDENT Hysteresis
  /METHOD=STEPWISE TimeSinceLastPeak TotalRainfall stormAPI QpMinusQo TMedian3
TMedian5 TMedian7 TMedian10 TMedian15.
```

Notes

Output Created		24-FEB-2016 14:11:31
Comments		
Input	Data	C:\Users\dg\Documents\Durham University\EdenDTC\1.Final Results\Stepwise Final\data.sav
	Active Dataset	DataSet1
	Filter	<none>
	Weight	<none>
	Split File	<none>
	N of Rows in Working Data File	190
Missing Value Handling	Definition of Missing	User-defined missing values are treated as missing.
	Cases Used	Statistics are based on cases with no missing values for any variable used.
Syntax		REGRESSION /MISSING LISTWISE /STATISTICS COEFF OUTS R ANOVA /CRITERIA=PIN(.05) POUT(.10) /NOORIGIN /DEPENDENT Hysteresis /METHOD=STEPWISE TimeSinceLastPeak TotalRainfall stormAPI QpMinusQo TMedian3 TMedian5 TMedian7 TMedian10 TMedian15.
Resources	Processor Time	00:00:00.05
	Elapsed Time	00:00:00.10
	Memory Required	9600 bytes
	Additional Memory Required for Residual Plots	0 bytes

```
[DataSet1] C:\Users\dg\Documents\Durham University\EdenDTC\1.Final
Results\Stepwise Final\data.sav
```

Variables Entered/Removed[a]

Model	Variables Entered	Variables Removed	Method
1	TMedian10	.	Stepwise (Criteria: Probability-of-F-to-enter <= .050, Probability-of-F-to-remove >= .100).
2	TotalRainfall	.	Stepwise (Criteria: Probability-of-F-to-enter <= .050, Probability-of-F-to-remove >= .100).
3	stormAPI	.	Stepwise (Criteria: Probability-of-F-to-enter <= .050, Probability-of-F-to-remove >= .100).
4	QpMinusQo	.	Stepwise (Criteria: Probability-of-F-to-enter <= .050, Probability-of-F-to-remove >= .100).

a. Dependent Variable: Hysteresis

Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.462[a]	.214	.210	.850
2	.532[b]	.283	.275	.814
3	.587[c]	.345	.334	.780
4	.599[d]	.359	.345	.773

a. Predictors: (Constant), TMedian10

b. Predictors: (Constant), TMedian10, TotalRainfall

c. Predictors: (Constant), TMedian10, TotalRainfall, stormAPI

d. Predictors: (Constant), TMedian10, TotalRainfall, stormAPI, QpMinusQo

ANOVA^a

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	36.734	1	36.734	50.854	.000^b
	Residual	135.076	187	.722		
	Total	171.810	188			
2	Regression	48.588	2	24.294	36.671	.000^c
	Residual	123.222	186	.662		
	Total	171.810	188			
3	Regression	59.219	3	19.740	32.435	.000^d
	Residual	112.590	185	.609		
	Total	171.810	188			
4	Regression	61.743	4	15.436	25.804	.000^e
	Residual	110.066	184	.598		
	Total	171.810	188			

a. Dependent Variable: Hysteresis

b. Predictors: (Constant), TMedian10

c. Predictors: (Constant), TMedian10, TotalRainfall

d. Predictors: (Constant), TMedian10, TotalRainfall, stormAPI

e. Predictors: (Constant), TMedian10, TotalRainfall, stormAPI, QpMinusQo

Coefficients^a

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	1.122	.131		8.591	.000
	TMedian10	-.110	.015	-.462	-7.131	.000
2	(Constant)	1.335	.135		9.901	.000
	TMedian10	-.092	.015	-.386	-5.979	.000
	TotalRainfall	-.025	.006	-.273	-4.230	.000
3	(Constant)	.808	.181		4.470	.000
	TMedian10	-.082	.015	-.342	-5.440	.000
	TotalRainfall	-.027	.006	-.292	-4.699	.000
	stormAPI	.012	.003	.252	4.180	.000
4	(Constant)	.780	.180		4.343	.000
	TMedian10	-.090	.015	-.376	-5.830	.000
	TotalRainfall	-.015	.008	-.169	-1.966	.051
	stormAPI	.015	.003	.304	4.683	.000
	QpMinusQo	-.101	.049	-.179	-2.054	.041

a. Dependent Variable: Hysteresis

Excluded Variables[a]

Model		Beta In	t	Sig.	Partial Correlation	Collinearity Statistics Tolerance
1	TimeSinceLastPeak	-.184[b]	-2.855	.005	-.205	.977
	TotalRainfall	-.273[b]	-4.230	.000	-.296	.923
	stormAPI	.232[b]	3.654	.000	.259	.976
	QpMinusQo	-.190[b]	-2.982	.003	-.214	.998
	TMedian3	-.114[b]	-.759	.449	-.056	.188
	TMedian5	-.066[b]	-.310	.757	-.023	.092
	TMedian7	.086[b]	.256	.798	.019	.037
	TMedian15	-.072[b]	-.214	.831	-.016	.037
2	TimeSinceLastPeak	-.128[c]	-1.987	.048	-.145	.921
	stormAPI	.252[c]	4.180	.000	.294	.971
	QpMinusQo	-.020[c]	-.240	.811	-.018	.537
	TMedian3	-.128[c]	-.892	.374	-.065	.188
	TMedian5	-.117[c]	-.570	.569	-.042	.092
	TMedian7	-.065[c]	-.200	.841	-.015	.037
	TMedian15	-.074[c]	-.228	.820	-.017	.037
3	TimeSinceLastPeak	.034[d]	.442	.659	.033	.613
	QpMinusQo	-.179[d]	-2.054	.041	-.150	.456
	TMedian3	-.110[d]	-.803	.423	-.059	.188
	TMedian5	-.114[d]	-.580	.563	-.043	.092
	TMedian7	-.032[d]	-.102	.919	-.008	.037
	TMedian15	-.218[d]	-.698	.486	-.051	.037
4	TimeSinceLastPeak	.006[e]	.082	.935	.006	.593
	TMedian3	-.110[e]	-.805	.422	-.059	.188
	TMedian5	-.122[e]	-.626	.532	-.046	.092
	TMedian7	-.044[e]	-.143	.886	-.011	.037
	TMedian15	-.252[e]	-.815	.416	-.060	.036

a. Dependent Variable: Hysteresis

b. Predictors in the Model: (Constant), TMedian10

c. Predictors in the Model: (Constant), TMedian10, TotalRainfall

d. Predictors in the Model: (Constant), TMedian10, TotalRainfall, stormAPI

e. Predictors in the Model: (Constant), TMedian10, TotalRainfall, stormAPI, QpMinusQo

Regression NMax

```
REGRESSION
  /MISSING LISTWISE
  /STATISTICS COEFF OUTS R ANOVA
  /CRITERIA=PIN(.05) POUT(.10)
  /NOORIGIN
  /DEPENDENT MaxNConc
  /METHOD=STEPWISE TimeSinceLastPeak TotalRainfall stormAPI StartQ PeakQ
QpMinusQo TotalQ seasonCos TMedian3 TMedian5 TMedian7 TMedian10 TMedian15.
```

Notes

Output Created		24-FEB-2016 14:06:59
Comments		
Input	Data	C:\Users\dg\Documents\Durham University\EdenDTC\1.Final Results\Stepwise Final\data.sav
	Active Dataset	DataSet1
	Filter	<none>
	Weight	<none>
	Split File	<none>
	N of Rows in Working Data File	190
Missing Value Handling	Definition of Missing	User-defined missing values are treated as missing.
	Cases Used	Statistics are based on cases with no missing values for any variable used.
Syntax		REGRESSION /MISSING LISTWISE /STATISTICS COEFF OUTS R ANOVA /CRITERIA=PIN(.05) POUT(.10) /NOORIGIN /DEPENDENT MaxNConc /METHOD=STEPWISE TimeSinceLastPeak TotalRainfall stormAPI StartQ PeakQ QpMinusQo TotalQ seasonCos TMedian3 TMedian5 TMedian7 TMedian10 TMedian15.
Resources	Processor Time	00:00:00.03
	Elapsed Time	00:00:00.09
	Memory Required	14336 bytes
	Additional Memory Required for Residual Plots	0 bytes

```
[DataSet1] C:\Users\dg\Documents\Durham University\EdenDTC\1.Final
Results\Stepwise Final\data.sav
```

Variables Entered/Removed[a]

Model	Variables Entered	Variables Removed	Method
1	seasonCos	.	Stepwise (Criteria: Probability-of-F-to-enter <= .050, Probability-of-F-to-remove >= .100).
2	TotalRainfall	.	Stepwise (Criteria: Probability-of-F-to-enter <= .050, Probability-of-F-to-remove >= .100).
3	TMedian15	.	Stepwise (Criteria: Probability-of-F-to-enter <= .050, Probability-of-F-to-remove >= .100).
4	TotalQ	.	Stepwise (Criteria: Probability-of-F-to-enter <= .050, Probability-of-F-to-remove >= .100).

a. Dependent Variable: MaxNConc

Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.557[a]	.310	.307	4.43690
2	.591[b]	.349	.342	4.32258
3	.621[c]	.385	.375	4.21209
4	.634[d]	.402	.389	4.16611

a. Predictors: (Constant), seasonCos

b. Predictors: (Constant), seasonCos, TotalRainfall

c. Predictors: (Constant), seasonCos, TotalRainfall, TMedian15

d. Predictors: (Constant), seasonCos, TotalRainfall, TMedian15, TotalQ

ANOVA[a]

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	1656.875	1	1656.875	84.165	.000[b]
	Residual	3681.293	187	19.686		
	Total	5338.169	188			
2	Regression	1862.820	2	931.410	49.849	.000[c]
	Residual	3475.349	186	18.685		
	Total	5338.169	188			
3	Regression	2055.959	3	685.320	38.628	.000[d]
	Residual	3282.209	185	17.742		
	Total	5338.169	188			
4	Regression	2144.577	4	536.144	30.890	.000[e]
	Residual	3193.592	184	17.356		
	Total	5338.169	188			

a. Dependent Variable: MaxNConc

b. Predictors: (Constant), seasonCos

c. Predictors: (Constant), seasonCos, TotalRainfall

Coefficients[a]

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	12.653	.352		35.949	.000
	seasonCos	-4.242	.462	-.557	-9.174	.000
2	(Constant)	11.071	.587		18.862	.000
	seasonCos	-3.810	.469	-.500	-8.124	.000
	TotalRainfall	.104	.031	.204	3.320	.001
3	(Constant)	14.485	1.182		12.253	.000
	seasonCos	-5.605	.711	-.736	-7.888	.000
	TotalRainfall	.114	.031	.223	3.695	.000
	TMedian15	-.401	.121	-.307	-3.299	.001
4	(Constant)	14.750	1.175		12.552	.000
	seasonCos	-5.166	.729	-.678	-7.085	.000
	TotalRainfall	.179	.042	.350	4.266	.000
	TMedian15	-.408	.120	-.313	-3.395	.001
	TotalQ	-.020	.009	-.180	-2.260	.025

a. Dependent Variable: MaxNConc

Excluded Variables[a]

Model		Beta In	t	Sig.	Partial Correlation	Collinearity Statistics Tolerance
1	TimeSinceLastPeak	.138[b]	2.199	.029	.159	.912
	TotalRainfall	.204[b]	3.320	.001	.237	.923
	stormAPI	-.057[b]	-.887	.376	-.065	.883
	StartQ	-.080[b]	-1.215	.226	-.089	.841
	PeakQ	.062[b]	.993	.322	.073	.951
	QpMinusQo	.083[b]	1.358	.176	.099	.985
	TotalQ	.046[b]	.747	.456	.055	.978
	TMedian3	-.098[b]	-1.009	.315	-.074	.393
	TMedian5	-.226[b]	-2.257	.025	-.163	.359
	TMedian7	-.288[b]	-2.862	.005	-.205	.350
	TMedian10	-.263[b]	-2.666	.008	-.192	.367

Model		Beta In	t	Sig.	Partial Correlation	Tolerance
	TMedian15	-.276[b]	-2.877	.004	-.206	.386
	TimeSinceLastPeak	.101[c]	1.599	.111	.117	.874
	stormAPI	-.086[c]	-1.364	.174	-.100	.868
	StartQ	-.051[c]	-.777	.438	-.057	.824
	PeakQ	-.114[c]	-1.444	.150	-.106	.561
	QpMinusQo	-.113[c]	-1.350	.179	-.099	.494
2	TotalQ	-.172[c]	-2.111	.036	-.153	.515
	TMedian3	-.110[c]	-1.171	.243	-.086	.393
	TMedian5	-.241[c]	-2.472	.014	-.179	.358
	TMedian7	-.306[c]	-3.121	.002	-.224	.349
	TMedian10	-.298[c]	-3.100	.002	-.222	.363
	TMedian15	-.307[c]	-3.299	.001	-.236	.383
	TimeSinceLastPeak	.069[d]	1.107	.270	.081	.850
	stormAPI	-.043[d]	-.678	.498	-.050	.825
	StartQ	-.025[d]	-.396	.693	-.029	.812
	PeakQ	-.098[d]	-1.279	.202	-.094	.559
	QpMinusQo	-.107[d]	-1.304	.194	-.096	.493
3	TotalQ	-.180[d]	-2.260	.025	-.164	.514
	TMedian3	.150[d]	1.244	.215	.091	.227
	TMedian5	.009[d]	.060	.952	.004	.146
	TMedian7	-.106[d]	-.537	.592	-.040	.086
	TMedian10	.026[d]	.083	.934	.006	.035
	TimeSinceLastPeak	.011[e]	.153	.878	.011	.690
	stormAPI	.012[e]	.170	.865	.013	.709
	StartQ	.027[e]	.397	.692	.029	.716
	PeakQ	-.019[e]	-.218	.828	-.016	.427
4	QpMinusQo	-.035[e]	-.388	.698	-.029	.404
	TMedian3	.148[e]	1.239	.217	.091	.227
	TMedian5	.015[e]	.100	.921	.007	.146
	TMedian7	-.087[e]	-.448	.655	-.033	.086
	TMedian10	.007[e]	.023	.982	.002	.035

a. Dependent Variable: MaxNConc

b. Predictors in the Model: (Constant), seasonCos

c. Predictors in the Model: (Constant), seasonCos, TotalRainfall

d. Predictors in the Model: (Constant), seasonCos, TotalRainfall, TMedian15

e. Predictors in the Model: (Constant), seasonCos, TotalRainfall, TMedian15, TotalQ

Regression NMax CosOUT

```
REGRESSION
  /MISSING LISTWISE
  /STATISTICS COEFF OUTS R ANOVA
  /CRITERIA=PIN(.05) POUT(.10)
  /NOORIGIN
  /DEPENDENT MaxNConc
  /METHOD=STEPWISE TimeSinceLastPeak TotalRainfall stormAPI StartQ PeakQ
QpMinusQo TotalQ TMedian3 TMedian5 TMedian7 TMedian10 TMedian15.
```

Notes

Output Created		24-FEB-2016 14:07:44
Comments		
Input	Data	C:\Users\dg\Documents\Durham University\EdenDTC\1.Final Results\Stepwise Final\data.sav
	Active Dataset	DataSet1
	Filter	<none>
	Weight	<none>
	Split File	<none>
	N of Rows in Working Data File	190
Missing Value Handling	Definition of Missing	User-defined missing values are treated as missing.
	Cases Used	Statistics are based on cases with no missing values for any variable used.
Syntax		REGRESSION /MISSING LISTWISE /STATISTICS COEFF OUTS R ANOVA /CRITERIA=PIN(.05) POUT(.10) /NOORIGIN /DEPENDENT MaxNConc /METHOD=STEPWISE TimeSinceLastPeak TotalRainfall stormAPI StartQ PeakQ QpMinusQo TotalQ TMedian3 TMedian5 TMedian7 TMedian10 TMedian15.
Resources	Processor Time	00:00:00.03
	Elapsed Time	00:00:00.13
	Memory Required	13056 bytes
	Additional Memory Required for Residual Plots	0 bytes

```
[DataSet1] C:\Users\dg\Documents\Durham University\EdenDTC\1.Final
Results\Stepwise Final\data.sav
```

170

Variables Entered/Removed[a]

Model	Variables Entered	Variables Removed	Method
1	TMedian3	.	Stepwise (Criteria: Probability-of-F-to-enter <= .050, Probability-of-F-to-remove >= .100).
2	TimeSinceLastPeak	.	Stepwise (Criteria: Probability-of-F-to-enter <= .050, Probability-of-F-to-remove >= .100).
3	TotalRainfall	.	Stepwise (Criteria: Probability-of-F-to-enter <= .050, Probability-of-F-to-remove >= .100).
4	TotalQ	.	Stepwise (Criteria: Probability-of-F-to-enter <= .050, Probability-of-F-to-remove >= .100).

a. Dependent Variable: MaxNConc

Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.396[a]	.156	.152	4.90721
2	.474[b]	.225	.217	4.71604
3	.513[c]	.263	.251	4.61172
4	.530[d]	.281	.266	4.56606

a. Predictors: (Constant), TMedian3

b. Predictors: (Constant), TMedian3, TimeSinceLastPeak

c. Predictors: (Constant), TMedian3, TimeSinceLastPeak, TotalRainfall

d. Predictors: (Constant), TMedian3, TimeSinceLastPeak, TotalRainfall, TotalQ

ANOVA[a]

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	835.079	1	835.079	34.678	.000[b]
	Residual	4503.090	187	24.081		
	Total	5338.169	188			
2	Regression	1201.328	2	600.664	27.007	.000[c]
	Residual	4136.841	186	22.241		
	Total	5338.169	188			
3	Regression	1403.592	3	467.864	21.999	.000[d]
	Residual	3934.576	185	21.268		
	Total	5338.169	188			
4	Regression	1501.972	4	375.493	18.010	.000[e]
	Residual	3836.197	184	20.849		
	Total	5338.169	188			

a. Dependent Variable: MaxNConc

b. Predictors: (Constant), TMedian3

c. Predictors: (Constant), TMedian3, TimeSinceLastPeak

d. Predictors: (Constant), TMedian3, TimeSinceLastPeak, TotalRainfall

e. Predictors: (Constant), TMedian3, TimeSinceLastPeak, TotalRainfall, TotalQ

Coefficients[a]

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	7.535	.742		10.159	.000
	TMedian3	.512	.087	.396	5.889	.000
2	(Constant)	6.657	.745		8.936	.000
	TMedian3	.486	.084	.375	5.795	.000
	TimeSinceLastPeak	.271	.067	.263	4.058	.000
3	(Constant)	5.843	.775		7.540	.000
	TMedian3	.427	.084	.329	5.063	.000
	TimeSinceLastPeak	.217	.068	.210	3.210	.002
	TotalRainfall	.106	.034	.208	3.084	.002
4	(Constant)	7.015	.938		7.479	.000
	TMedian3	.354	.090	.273	3.939	.000
	TimeSinceLastPeak	.133	.077	.130	1.731	.085
	TotalRainfall	.190	.051	.372	3.690	.000
	TotalQ	-.024	.011	-.210	-2.172	.031

a. Dependent Variable: MaxNConc

Excluded Variables[a]

Model		Beta In	t	Sig.	Partial Correlation	Collinearity Statistics Tolerance
1	TimeSinceLastPeak	.263[b]	4.058	.000	.285	.994
	TotalRainfall	.264[b]	3.955	.000	.279	.942
	stormAPI	-.184[b]	-2.761	.006	-.198	.976
	StartQ	-.213[b]	-3.167	.002	-.226	.952
	PeakQ	-.023[b]	-.334	.739	-.024	.989
	QpMinusQo	.034[b]	.502	.616	.037	.998
	TotalQ	.006[b]	.083	.934	.006	.988
	TMedian5	-.347[b]	-1.222	.223	-.089	.056
	TMedian7	-.179[b]	-.935	.351	-.068	.123
	TMedian10	-.050[b]	-.324	.746	-.024	.188
	TMedian15	-.046[b]	-.343	.732	-.025	.254
2	TotalRainfall	.208[c]	3.084	.002	.221	.879

172

	stormAPI	-.055[c]	-.706	.481	-.052	.680
	StartQ	-.122[c]	-1.684	.094	-.123	.785
	PeakQ	.031[c]	.470	.639	.035	.950
	QpMinusQo	.059[c]	.914	.362	.067	.988
	TotalQ	.057[c]	.868	.386	.064	.953
	TMedian5	-.430[c]	-1.576	.117	-.115	.056
	TMedian7	-.283[c]	-1.526	.129	-.111	.121
	TMedian10	-.170[c]	-1.122	.263	-.082	.181
	TMedian15	-.134[c]	-1.032	.303	-.076	.247
	stormAPI	-.124[d]	-1.570	.118	-.115	.636
	StartQ	-.101[d]	-1.422	.157	-.104	.778
	PeakQ	-.188[d]	-2.168	.031	-.158	.521
	QpMinusQo	-.174[d]	-1.901	.059	-.139	.469
3	TotalQ	-.210[d]	-2.172	.031	-.158	.417
	TMedian5	-.466[d]	-1.749	.082	-.128	.055
	TMedian7	-.309[d]	-1.709	.089	-.125	.120
	TMedian10	-.218[d]	-1.471	.143	-.108	.180
	TMedian15	-.172[d]	-1.355	.177	-.099	.245
	stormAPI	-.094[e]	-1.177	.241	-.087	.611
	StartQ	-.065[e]	-.883	.378	-.065	.722
	PeakQ	-.133[e]	-1.408	.161	-.104	.434
	QpMinusQo	-.118[e]	-1.208	.229	-.089	.409
4	TMedian5	-.479[e]	-1.814	.071	-.133	.055
	TMedian7	-.312[e]	-1.743	.083	-.128	.120
	TMedian10	-.231[e]	-1.571	.118	-.115	.179
	TMedian15	-.185[e]	-1.466	.144	-.108	.245

a. Dependent Variable: MaxNConc

b. Predictors in the Model: (Constant), TMedian3

c. Predictors in the Model: (Constant), TMedian3, TimeSinceLastPeak

d. Predictors in the Model: (Constant), TMedian3, TimeSinceLastPeak, TotalRainfall

e. Predictors in the Model: (Constant), TMedian3, TimeSinceLastPeak, TotalRainfall, TotalQ

173

Regression NMin

```
REGRESSION
  /MISSING LISTWISE
  /STATISTICS COEFF OUTS R ANOVA
  /CRITERIA=PIN(.05) POUT(.10)
  /NOORIGIN
  /DEPENDENT MinNConc
  /METHOD=STEPWISE TimeSinceLastPeak TotalRainfall stormAPI StartQ PeakQ
QpMinusQo TotalQ SeasonID seasonCos TMedian3 TMedian5 TMedian7 TMedian10
TMedian15.
```

Notes

Output Created		01-MAR-2016 23:01:16
Comments		
Input	Data	C:\Users\dg\Documents\Durham University\EdenDTC\1.Final Results\Stepwise Final\data.sav
	Active Dataset	DataSet1
	Filter	<none>
	Weight	<none>
	Split File	<none>
	N of Rows in Working Data File	190
Missing Value Handling	Definition of Missing	User-defined missing values are treated as missing.
	Cases Used	Statistics are based on cases with no missing values for any variable used.
Syntax		REGRESSION /MISSING LISTWISE /STATISTICS COEFF OUTS R ANOVA /CRITERIA=PIN(.05) POUT(.10) /NOORIGIN /DEPENDENT MinNConc /METHOD=STEPWISE TimeSinceLastPeak TotalRainfall stormAPI StartQ PeakQ QpMinusQo TotalQ SeasonID seasonCos TMedian3 TMedian5 TMedian7 TMedian10 TMedian15.
Resources	Processor Time	00:00:00.06
	Elapsed Time	00:00:00.09
	Memory Required	15712 bytes
	Additional Memory Required for Residual Plots	0 bytes

```
[DataSet1] C:\Users\dg\Documents\Durham University\EdenDTC\1.Final
Results\Stepwise Final\data.sav
```

Variables Entered/Removed[a]

Model	Variables Entered	Variables Removed	Method
1	TMedian3	.	Stepwise (Criteria: Probability-of-F-to-enter <= .050, Probability-of-F-to-remove >= .100).
2	PeakQ	.	Stepwise (Criteria: Probability-of-F-to-enter <= .050, Probability-of-F-to-remove >= .100).
3	TMedian15	.	Stepwise (Criteria: Probability-of-F-to-enter <= .050, Probability-of-F-to-remove >= .100).
4	SeasonID	.	Stepwise (Criteria: Probability-of-F-to-enter <= .050, Probability-of-F-to-remove >= .100).

a. Dependent Variable: MinNConc

Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.484[a]	.234	.230	1.5674619176610
2	.571[b]	.326	.319	1.4747531656269
3	.584[c]	.341	.330	1.4625138375517
4	.608[d]	.370	.356	1.4332585695653

a. Predictors: (Constant), TMedian3

b. Predictors: (Constant), TMedian3, PeakQ

c. Predictors: (Constant), TMedian3, PeakQ, TMedian15

d. Predictors: (Constant), TMedian3, PeakQ, TMedian15, SeasonID

ANOVA[a]

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	140.632	1	140.632	57.239	.000[b]
	Residual	459.447	187	2.457		
	Total	600.080	188			
2	Regression	195.549	2	97.774	44.956	.000[c]
	Residual	404.531	186	2.175		
	Total	600.080	188			
3	Regression	204.374	3	68.125	31.850	.000[d]
	Residual	395.705	185	2.139		
	Total	600.080	188			
4	Regression	222.101	4	55.525	27.030	.000[e]
	Residual	377.978	184	2.054		
	Total	600.080	188			

a. Dependent Variable: MinNConc

b. Predictors: (Constant), TMedian3

c. Predictors: (Constant), TMedian3, PeakQ

175

d. Predictors: (Constant), TMedian3, PeakQ, TMedian15

e. Predictors: (Constant), TMedian3, PeakQ, TMedian15, SeasonID

Coefficients[a]

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	4.679	.237		19.748	.000
	TMedian3	.210	.028	.484	7.566	.000
2	(Constant)	5.416	.267		20.294	.000
	TMedian3	.196	.026	.452	7.467	.000
	PeakQ	-.302	.060	-.304	-5.025	.000
3	(Constant)	5.539	.271		20.403	.000
	TMedian3	.286	.051	.660	5.566	.000
	PeakQ	-.306	.060	-.309	-5.135	.000
	TMedian15	-.105	.052	-.241	-2.031	.044
4	(Constant)	5.263	.282		18.656	.000
	TMedian3	.280	.050	.645	5.553	.000
	PeakQ	-.298	.058	-.300	-5.097	.000
	TMedian15	-.164	.055	-.375	-3.002	.003
	SeasonID	.324	.110	.226	2.938	.004

a. Dependent Variable: MinNConc

Excluded Variables[a]

Model		Beta In	t	Sig.	Partial Correlation	Collinearity Statistics
						Tolerance
1	TimeSinceLastPeak	.097[b]	1.519	.130	.111	.994
	TotalRainfall	-.187[b]	-2.900	.004	-.208	.942
	stormAPI	-.229[b]	-3.651	.000	-.259	.976
	StartQ	-.140[b]	-2.148	.033	-.156	.952
	PeakQ	-.304[b]	-5.025	.000	-.346	.989
	QpMinusQo	-.283[b]	-4.652	.000	-.323	.998
	TotalQ	-.210[b]	-3.352	.001	-.239	.988
	SeasonID	.162[b]	2.092	.038	.152	.667
	seasonCos	-.178[b]	-1.755	.081	-.128	.393
	TMedian5	-.370[b]	-1.370	.172	-.100	.056
	TMedian7	-.273[b]	-1.497	.136	-.109	.123
	TMedian10	-.256[b]	-1.745	.083	-.127	.188
	TMedian15	-.219[b]	-1.737	.084	-.126	.254

		Beta In	t	Sig.	Partial Correlation	Tolerance
2	TimeSinceLastPeak	.038^c	.618	.537	.045	.954
	TotalRainfall	-.008^c	-.107	.915	-.008	.620
	stormAPI	-.106^c	-1.523	.130	-.111	.748
	StartQ	-.040^c	-.606	.545	-.044	.844
	QpMinusQo	.140^c	.606	.545	.044	.068
	TotalQ	.015^c	.175	.862	.013	.483
	SeasonID	.142^c	1.937	.054	.141	.665
	seasonCos	-.075^c	-.758	.449	-.056	.374
	TMedian5	-.435^c	-1.715	.088	-.125	.056
	TMedian7	-.324^c	-1.893	.060	-.138	.122
	TMedian10	-.275^c	-1.999	.047	-.145	.188
	TMedian15	-.241^c	-2.031	.044	-.148	.254
3	TimeSinceLastPeak	.059^d	.959	.339	.071	.930
	TotalRainfall	.022^d	.278	.781	.021	.599
	stormAPI	-.103^d	-1.492	.137	-.109	.747
	StartQ	-.056^d	-.860	.391	-.063	.832
	QpMinusQo	.199^d	.860	.391	.063	.067
	TotalQ	.024^d	.278	.781	.021	.482
	SeasonID	.226^d	2.938	.004	.212	.577
	seasonCos	-.165^d	-1.587	.114	-.116	.328
	TMedian5	-.186^d	-.571	.568	-.042	.034
	TMedian7	-.128^d	-.445	.657	-.033	.044
	TMedian10	-.113^d	-.304	.762	-.022	.026
4	TimeSinceLastPeak	.051^e	.837	.404	.062	.928
	TotalRainfall	.030^e	.402	.688	.030	.598
	stormAPI	-.070^e	-1.017	.310	-.075	.724
	StartQ	-.039^e	-.606	.546	-.045	.825
	QpMinusQo	.138^e	.606	.546	.045	.066
	TotalQ	.030^e	.360	.719	.027	.482
	seasonCos	-.190^e	-1.868	.063	-.137	.326
	TMedian5	-.110^e	-.345	.731	-.025	.034
	TMedian7	-.058^e	-.205	.838	-.015	.043
	TMedian10	-.002^e	-.005	.996	.000	.025

a. Dependent Variable: MinNConc

b. Predictors in the Model: (Constant), TMedian3

c. Predictors in the Model: (Constant), TMedian3, PeakQ

d. Predictors in the Model: (Constant), TMedian3, PeakQ, TMedian15

e. Predictors in the Model: (Constant), TMedian3, PeakQ, TMedian15, SeasonID

Regression NMin CosOUT

```
REGRESSION
  /MISSING LISTWISE
  /STATISTICS COEFF OUTS R ANOVA
  /CRITERIA=PIN(.05) POUT(.10)
  /NOORIGIN
  /DEPENDENT MinNConc
  /METHOD=STEPWISE TimeSinceLastPeak TotalRainfall stormAPI StartQ PeakQ
QpMinusQo TotalQ SeasonID TMedian3 TMedian5 TMedian7 TMedian10 TMedian15.
```

Notes

Output Created		01-MAR-2016 23:10:03
Comments		
Input	Data	C:\Users\dg\Documents\Durham University\EdenDTC\1.Final Results\Stepwise Final\data.sav
	Active Dataset	DataSet1
	Filter	<none>
	Weight	<none>
	Split File	<none>
	N of Rows in Working Data File	190
Missing Value Handling	Definition of Missing	User-defined missing values are treated as missing.
	Cases Used	Statistics are based on cases with no missing values for any variable used.
Syntax		REGRESSION /MISSING LISTWISE /STATISTICS COEFF OUTS R ANOVA /CRITERIA=PIN(.05) POUT(.10) /NOORIGIN /DEPENDENT MinNConc /METHOD=STEPWISE TimeSinceLastPeak TotalRainfall stormAPI StartQ PeakQ QpMinusQo TotalQ SeasonID TMedian3 TMedian5 TMedian7 TMedian10 TMedian15.
Resources	Processor Time	00:00:00.03
	Elapsed Time	00:00:00.06
	Memory Required	14368 bytes
	Additional Memory Required for Residual Plots	0 bytes

```
[DataSet1] C:\Users\dg\Documents\Durham University\EdenDTC\1.Final
Results\Stepwise Final\data.sav
```

178

Variables Entered/Removed[a]

Model	Variables Entered	Variables Removed	Method
1	TMedian3	.	Stepwise (Criteria: Probability-of-F-to-enter <= .050, Probability-of-F-to-remove >= .100).
2	PeakQ	.	Stepwise (Criteria: Probability-of-F-to-enter <= .050, Probability-of-F-to-remove >= .100).
3	TMedian15	.	Stepwise (Criteria: Probability-of-F-to-enter <= .050, Probability-of-F-to-remove >= .100).
4	SeasonID	.	Stepwise (Criteria: Probability-of-F-to-enter <= .050, Probability-of-F-to-remove >= .100).

a. Dependent Variable: MinNConc

Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.484[a]	.234	.230	1.5674619176610
2	.571[b]	.326	.319	1.4747531656269
3	.584[c]	.341	.330	1.4625138375517
4	.608[d]	.370	.356	1.4332585695653

a. Predictors: (Constant), TMedian3

b. Predictors: (Constant), TMedian3, PeakQ

c. Predictors: (Constant), TMedian3, PeakQ, TMedian15

d. Predictors: (Constant), TMedian3, PeakQ, TMedian15, SeasonID

ANOVA[a]

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	140.632	1	140.632	57.239	.000[b]
	Residual	459.447	187	2.457		
	Total	600.080	188			
2	Regression	195.549	2	97.774	44.956	.000[c]
	Residual	404.531	186	2.175		
	Total	600.080	188			
3	Regression	204.374	3	68.125	31.850	.000[d]
	Residual	395.705	185	2.139		
	Total	600.080	188			
4	Regression	222.101	4	55.525	27.030	.000[e]
	Residual	377.978	184	2.054		
	Total	600.080	188			

a. Dependent Variable: MinNConc

b. Predictors: (Constant), TMedian3

c. Predictors: (Constant), TMedian3, PeakQ

d. Predictors: (Constant), TMedian3, PeakQ, TMedian15

e. Predictors: (Constant), TMedian3, PeakQ, TMedian15, SeasonID

Excluded Variables[a]

Model		Beta In	t	Sig.	Partial Correlation	Collinearity Statistics Tolerance
1	TimeSinceLastPeak	.097[b]	1.519	.130	.111	.994
	TotalRainfall	-.187[b]	-2.900	.004	-.208	.942
	stormAPI	-.229[b]	-3.651	.000	-.259	.976
	StartQ	-.140[b]	-2.148	.033	-.156	.952
	PeakQ	-.304[b]	-5.025	.000	-.346	.989
	QpMinusQo	-.283[b]	-4.652	.000	-.323	.998
	TotalQ	-.210[b]	-3.352	.001	-.239	.988
	SeasonID	.162[b]	2.092	.038	.152	.667
	TMedian5	-.370[b]	-1.370	.172	-.100	.056
	TMedian7	-.273[b]	-1.497	.136	-.109	.123
	TMedian10	-.256[b]	-1.745	.083	-.127	.188
	TMedian15	-.219[b]	-1.737	.084	-.126	.254
2	TimeSinceLastPeak	.038[c]	.618	.537	.045	.954
	TotalRainfall	-.008[c]	-.107	.915	-.008	.620
	stormAPI	-.106[c]	-1.523	.130	-.111	.748
	StartQ	-.040[c]	-.606	.545	-.044	.844
	QpMinusQo	.140[c]	.606	.545	.044	.068
	TotalQ	.015[c]	.175	.862	.013	.483
	SeasonID	.142[c]	1.937	.054	.141	.665
	TMedian5	-.435[c]	-1.715	.088	-.125	.056
	TMedian7	-.324[c]	-1.893	.060	-.138	.122
	TMedian10	-.275[c]	-1.999	.047	-.145	.188
	TMedian15	-.241[c]	-2.031	.044	-.148	.254
3	TimeSinceLastPeak	.059[d]	.959	.339	.071	.930
	TotalRainfall	.022[d]	.278	.781	.021	.599
	stormAPI	-.103[d]	-1.492	.137	-.109	.747
	StartQ	-.056[d]	-.860	.391	-.063	.832
	QpMinusQo	.199[d]	.860	.391	.063	.067
	TotalQ	.024[d]	.278	.781	.021	.482
	SeasonID	.226[d]	2.938	.004	.212	.577
	TMedian5	-.186[d]	-.571	.568	-.042	.034
	TMedian7	-.128[d]	-.445	.657	-.033	.044
	TMedian10	-.113[d]	-.304	.762	-.022	.026
4	TimeSinceLastPeak	.051[e]	.837	.404	.062	.928

TotalRainfall	.030[e]	.402	.688	.030	.598
stormAPI	-.070[e]	-1.017	.310	-.075	.724
StartQ	-.039[e]	-.606	.546	-.045	.825
QpMinusQo	.138[e]	.606	.546	.045	.066
TotalQ	.030[e]	.360	.719	.027	.482
TMedian5	-.110[e]	-.345	.731	-.025	.034
TMedian7	-.058[e]	-.205	.838	-.015	.043
TMedian10	-.002[e]	-.005	.996	.000	.025

a. Dependent Variable: MinNConc

b. Predictors in the Model: (Constant), TMedian3

c. Predictors in the Model: (Constant), TMedian3, PeakQ

d. Predictors in the Model: (Constant), TMedian3, PeakQ, TMedian15

e. Predictors in the Model: (Constant), TMedian3, PeakQ, TMedian15, SeasonID

Coefficients[a]

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	4.679	.237		19.748	.000
	TMedian3	.210	.028	.484	7.566	.000
2	(Constant)	5.416	.267		20.294	.000
	TMedian3	.196	.026	.452	7.467	.000
	PeakQ	-.302	.060	-.304	-5.025	.000
3	(Constant)	5.539	.271		20.403	.000
	TMedian3	.286	.051	.660	5.566	.000
	PeakQ	-.306	.060	-.309	-5.135	.000
	TMedian15	-.105	.052	-.241	-2.031	.044
4	(Constant)	5.263	.282		18.656	.000
	TMedian3	.280	.050	.645	5.553	.000
	PeakQ	-.298	.058	-.300	-5.097	.000
	TMedian15	-.164	.055	-.375	-3.002	.003
	SeasonID	.324	.110	.226	2.938	.004

a. Dependent Variable: MinNConc

Appendix 4: Hysteresis plots

153

154

155

156

157

158

159

160