MA40195 Project

Assessing the reliability of high flows records

Author :
Luke Shaw

Supervisor:
Dr. Ilaria Prosdocimi

Submitted
08/05/2017

Contents

Abstract

The purpose of this project was to assess the reliability of peaks over the threshold (POT) data for water gauging stations across the UK. This data comprises all peaks of a river's observed flow that exceed a predetermined threshold.

I developed multiple functions in the statistical package R to take the data file downloaded from the National River Flow Archive (NRFA), and clean it into a format suitable for analysis. Along the way multiple data quality checks were implemented, including cross-referencing another data set to check for missing values.

With the cleaned data, I performed statistical inference for two parameters of interest. The model used was an extended quasi-Poisson model, a Poisson fit in which I could test whether the dispersion parameter ϕ was statistically significant from 1, which is the value ϕ should be if the statistical assumptions are met.

However, when applying the analysis several abnormal results were encountered to suggest that the Poisson Distributional assumption was invalid. This was for multiple reasons, for example from poor quality data or because the threshold λ had been set incorrectly. The threshold was designed by the NRFA so the data had on average five POT events per year.

I identified 41 gauging stations out of the 263 that had poorly performing data, and set out guidelines for diagnosing whether a given data set is usable.

Introduction

When estimating and predicting the frequency of extreme events, there are two common forms of data sets:

1. **Annual Maxima (AM)** - containing information on the most extreme event for each year.
2. **Peaks Over Threshold (POT)** - containing information on all events that were more extreme than a pre-determined value.

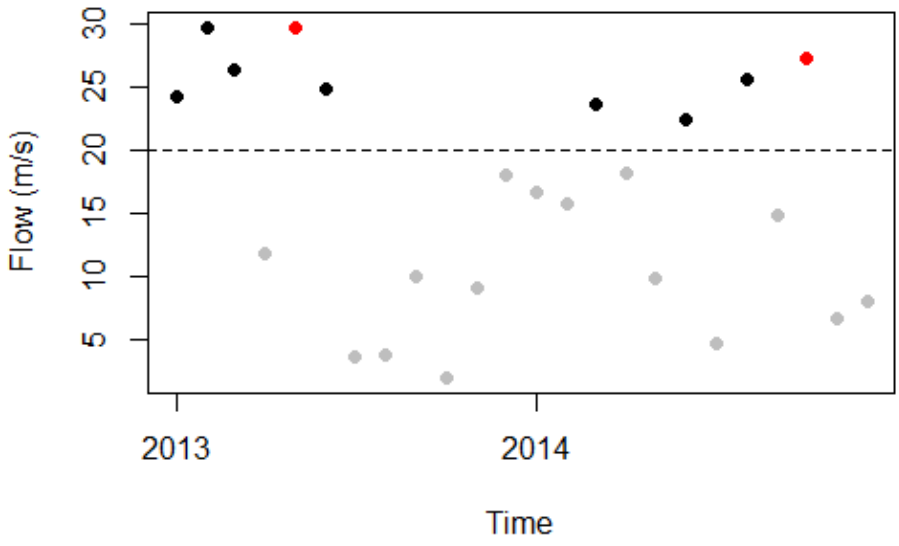

Figure 1: A simulated image showing how the POT and AM data sets are formed. The gauging station records all the points. The dashed line at 20 m/s is set as the threshold, and all nine points above it would be in the POT data set. The two red points are the annual maxima for each year, so are the only elements that would be recorded by the AM data set.

For flood data, AM data is usually used as it intuitively appears to make the most sense; determining the size and timing of the largest flood each year is valuable information. This

may not be the best approach, however, as it uses a fraction of the available data, and hence achieving significant results is difficult. It may also ignore other important features of the data. An example of the information that each of the two data sets would contain can be seen in Figure 1.

For this project, I looked at POT data for 263 water gauging stations across the United Kingdom. There is statistical theory concerning how to model Extreme Value data, which is the category POT data falls in to. I studied and adapted the theory to this specific situation, implemented the analysis, and identified gauging stations that were problematic.

This report is split into two distinct sections. The first section details how I cleaned the data from the version presented online into a format in which the analysis could be performed, as well as how I checked the validity of the data through various means involving rejecting incomplete information and cross-referencing the information to a data set cleaned by my supervisor Dr Prosdocimi. The second section gives the statistical theory, and discusses the results I found when applying the methods to the cleaned data sets. I have also attached all the R code that I created and implemented, which can be found in the Supplementary Section.

Chapter 1

Acquiring and Cleaning the Data

1.1 National River Flow Archive

The National River Flow Archive (NRFA) is an organisation that deals exclusively with water gauging stations in the UK[5]. It is involved from start to finish in the process, from collecting data from the gauging stations to quality control checking right through to publishing analytics. As part of transparency, anyone can request to use the data and the NRFA will make it available for free, which is how I have obtained all the data for this project.

Figure 1.1: The NRFA provided all the data used in this project

I initially had data for 956 gauging stations, as can be seen in Figure 1.2. However, data cleaning and analysis was not carried out on all of them due to known issues with data quality, as specified by the NRFA. I only cleaned data from the stations that were "Suitable for QMED", a criteria allocated by the NRFA. Data is of this quality if the observed annual maximum flow is confidently within 30% of the true value. The reason for only using the highest quality data to begin with was to do with time pressure, as cleaning the data is difficult enough and I still had over 200 stations to perform the analysis on.

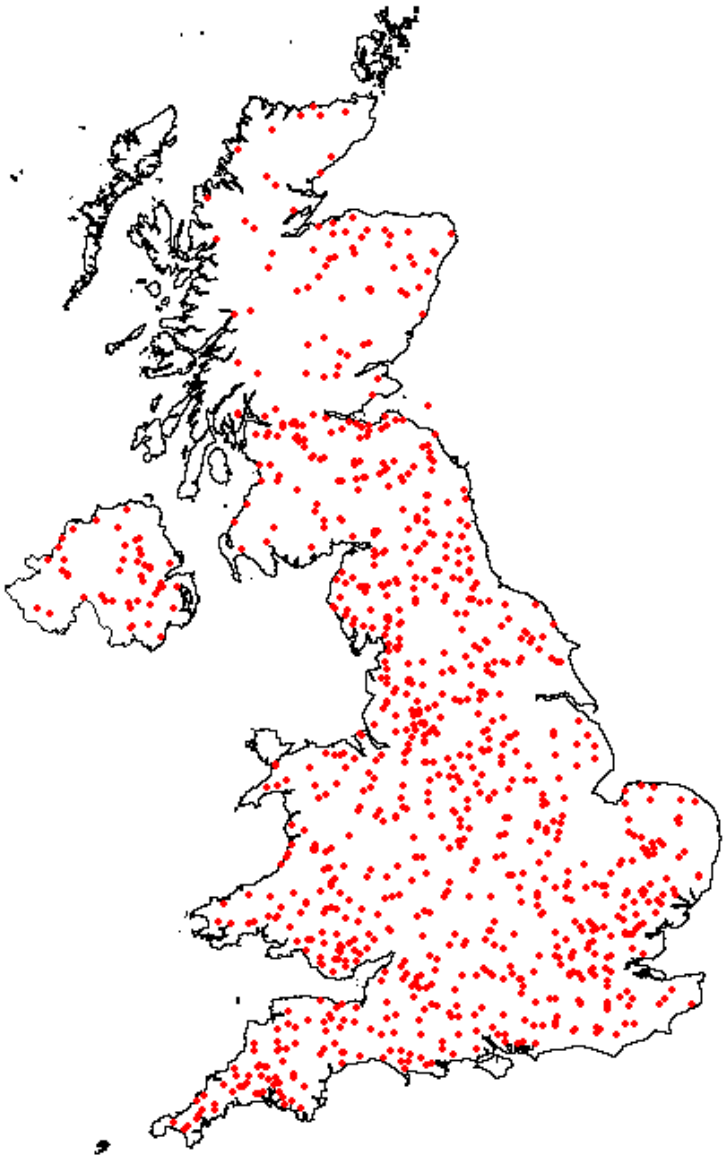

Figure 1.2: The locations of the 956 gauging stations in the UK. Note that the Republic of Ireland is excluded from the image.

1.2 The Initial Data

I downloaded the WINFAP-FEH_v4.1 data set from the NRFA website [5]. There are five separate file types for each station, but I focused on the .PT files as the data I was hoping to analyse was the POT values, not AM or location.

In Section 1.3.4 I discuss instances when data didn't follow the exact same pattern, causing errors in my code. Here I will discuss the standard format of the .PT files, an example of which can be seen in Figure 1.3.

	V1	V2	V3
1	[STATION NUMBER]		
2	17001		
3	[END]		
4	[POT Details]		
5	Record Period	01 Nov 1969	28 Mar 2006
6	Threshold	48.987	
7	[End]		
8	[POT Gaps]		
9	01 Jan 1983	06 Jan 1983	
10	11 Aug 1988	13 Sep 1988	
11	[End]		
12	[POT Values]		
13	02 Nov 1969	107.513	2.040
14	22 Dec 1969	50.762	1.400
15	30 Nov 1970	83.651	1.798
16	12 Feb 1971	49.320	1.380

Figure 1.3: An example of the format of the .PT files.

For each .PT the header contains the Station Number "[STATION NUMBER]", the start and end dates that the station has data for as well as the pre-defined threshold value "[POT DETAILS]", and the dates between which it is known the station wasn't recording data "[POT Gaps]".

Then, the file presented the POT occurrences themselves "[POT Values]", rows 13 onwards in Figure 1.3, in which each row contains the date of the POT, the flow at that time, and the height of the river in metres. The flow is estimated from the height of river and known properties of the river, and for this project I assumed all calculations were correct. For example, in Figure 1.3 we see that on 02nd November 1969 station 17001 observed a flow of 107.513 m/s, which is a POT because it is higher than the threshold set at 49.987 m/s.

Dr Prosdocimi had already cleaned the AM data and had an appropriate data set with all the information on annual maxima that I needed for checking data consistency, so I personally never dealt with the AM data.

1.3 The Cleaning Process

I spent at least half of this project cleaning the data, involving removing years with incomplete data and checking that the data was consistent. Here I document the process, and include the functions I created in the Supplementary Section 3.

1.3.1 Dates

Each station has the date on which a POT event occurred, in "DD MMM YYYY" format, for example "02 Nov 1969" is the first event in Figure 1.3. I needed to convert this into a more manageable format, and initially created my own functions in RStudio to manually separate the character string then output a numerical value that is easier to work with. However, after researching online I discovered the "lubridate" package [3], which does all the hard work for you. It even has multiple useful in-built capabilities such as calculating the number of days between the dates by simply using the "-" symbol.

1.3.2 Water Year

Due to the water cycle, the Water Year starts on 01st October of the calendar year. Thus, the theory of setting a threshold for an average of five POT is with reference to Water Year, as opposed to calendar year. The quickest way to deal with this in coding was, I found, by creating a separate function called "Water_Year" (see Supplementary Section 3.1), that outputs the Water Year when given a date in lubridate format. This kept my code clean and avoided needless repetition within other functions.

1.3.3 Getting from a .PT file to the number of POT events per Valid Water Year

All functions are provided in Supplementary Section 3, but here I shall detail the process for a given .PT file that takes all the POT events and then adds the events for each valid Water Year to the overarching data set events_table. In Figure 1.4 the format of the data frame that was required to make the events_table is shown, and then in Figure 1.5 the final information can be seen, as this is station 17001 in events_table.

	WY	percent.complete	AM.vs.Thresh	WYvalid
1	1969	91.23288	TRUE	1
2	1970	100.00000	TRUE	1
3	1971	100.00000	TRUE	1
4	1972	100.00000	FALSE	1
5	1973	100.00000	TRUE	1
6	1974	100.00000	TRUE	1
7	1975	100.00000	TRUE	1
8	1976	100.00000	TRUE	1
9	1977	100.00000	TRUE	1
10	1978	100.00000	TRUE	1
11	1979	100.00000	TRUE	1
12	1980	100.00000	TRUE	1
13	1981	100.00000	TRUE	1
14	1982	98.63014	TRUE	1
15	1983	100.00000	TRUE	1
16	1984	100.00000	TRUE	1

Figure 1.4: Continued Example for Station 17001. This is the Water Year table that is used to create the number of POT per valid Water Year, as inputted into events_table. The "FALSE" entry in 1972 is because the Annual Maximum for that year was 43.494m/s, which is lower than the threshold. Hence that Water Year has no POT but it is not due to missing data.

	statno	validWY	num_events
1	17001	1969	2
2	17001	1970	2
3	17001	1971	1
4	17001	1972	0
5	17001	1973	2
6	17001	1974	2
7	17001	1975	3
8	17001	1976	1
9	17001	1977	3
10	17001	1978	2
11	17001	1979	5

Figure 1.5: Continued Example for Station 17001. This is the first ten of the 9602 rows that events_table ended up containing after my code was run on all 263 data sets.

1. **Amount of data per Water Year**

   Firstly, inside the function valid_data (3.1) I used a for loop to cycle over each Water Year in which the gauging station had data recorded. In the continued example of Station 17001 in Figure 1.3 this would be from 1969 to 2005, seeing as the start date "01 Nov 1969" is in the 1969 Water Year and the end date "02 Mar 2006" is in the 2005 Water Year (remembering that Water Year X starts on 01 Oct X).

   One of the criteria I imposed for a Water Year to count as valid is that it had at least 75% of the year with data being recorded. This is a reasonable measure to prevent having a low POT count purely on the basis that the gauging station wasn't recording any data. To implement this I cycled through the known POT gaps and used my bespoke function Gap_Percent (3.1) to calculate the percentage of the Water Year that the gap length was over. In Station 17001 the first gap happens for the first 6 days of January in 1983, which only amounts to 1.36% of the 1982 Water Year, so that Water Year was still counted as valid as can be seen in Figure 1.4.

   However, I had to be careful with the known gaps as there were different scenarios that could occur. My solution was to have three distinct cases, and treat them accordingly.

   Case 1: When the beginning and end dates are in the same Water Year. This is easiest to deal with as I needed only count the percentage of year they span and then take that away from the row corresponding to that Water Year. The aforementioned first gap for Station 17001 falls into this category.

   Case 2: When the beginning and end dates are in adjacent Water Years. In this scenario, I split the dates into 2 pieces separated by the 01st of October, and added their gaps to the corresponding Water Years in two separate steps.

   Case 3: When the beginning and end dates span multiple Water Years. In this instance, I found all the Water Years strictly inside those dates and made the percentage of complete data equal 0, and treat the end Water Years as in case 2.

   The final case had to be added after my code was not running as intended. I had not anticipated the third scenario because I presumed gaps in data collection would only last for a month or two before being fixed. In fact this was far from the case, and there were stations with over 20 years of reported missing data.

2. **Comparing against Annual Maximum**

   Still inside the valid_data function, the next step was to compare the POT results I had to the Annual Maximum for that year. This was a sanity check to ensure the reported data was consistent with itself. Dr Prosdocimi had already cleaned the data for the annual maxima, so I was comparing against that data file.

   If the .PT file said the Water Year was valid but had no POT for that year, which can understandably happen, then my code checks the corresponding AM value for

that year. If the annual maximum was higher than the threshold then something had gone wrong, because that peak should have been included in the .PT file. If the Annual Maximum was lower than the threshold, then the response of no POT for that year was justified. An example of this occurring can be seen and is discussed in Figure 1.4

This check was important because it brought confidence to a Water Year with no entries NOT simply due to the .PT file having incorrect information, as it was backed up by the AM file. However, as I discuss later in the report in Section 2.3.3, there is still no way of knowing if the station was outputting incorrect information consistently across both the .PT and AM files, as in that case they would match and were both incorrect.

3. **Adding to "events_table"**

   The penultimate stage was to add the cleaned data for the individual station to the overarching Events Table. This Table had columns of Station Number (statno), valid Water Year (validWY), and the number of POT (num_events). After all 263 stations were added, events_table had dimensions of 9602 by 3, which means from basic calculations it can be shown the average number of valid Water Years per station is $9602/263 = 36.5$. This is a healthy number, and proved adequate for analysis that was carried out later on.

4. **Adding to disp_table**

   The last thing that needed to be done was to calculate the dispersion and mean parameters for each Station. I used the inbuilt R function tapply to carry out this step in just a few lines, see Supplementary Section 3.1, and then created a Dispersion Table with each row containing a unique Station Number and the mean and dispersion estimates. Most of the analysis in Chapter 2 was done from either the dispersion table or the events table.

Though a difficult and somewhat complicated process, the end result was worthwhile. The number of POT events for the cleaned data set for Station 17001 can be seen in Figure 1.6.

1.3.4 Pitfalls along the way

As is often the case when coding, there were multiple instances where errors needed correcting and bugs needed fixing. Here I document some of my errors, and how I resolved them.

Firstly, I had multiple stations throwing up errors when being compared against the AM data set. Initially I could not see why the error was happening, as when I went into the

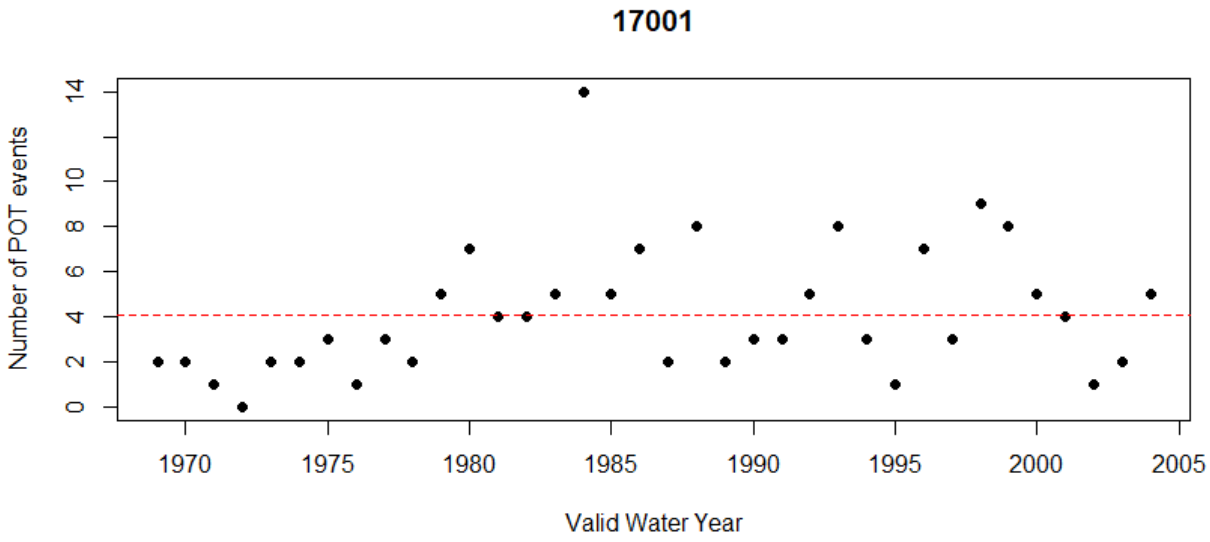

Figure 1.6: The POT cleaned data for station 17001. The horizontal dashed line is the mean of the data points, and was our point estimate for λ as is discussed in the following Section.

.PT file the troublesome Water Years should have been rejected. In fact, I hadn't realised that, in very rare cases, the missing gauge station data was recorded as "[POT Rejected]" instead of "[POT Gaps]", and so wasn't being picked up by my code to remove those gaps in data from the "percent.complete" column in the Water Year table. This is an example of when the code was working well, it was just being given a scenario that's unpredictable in advance.

Calculating the percentage of complete data per Water Year was extremely challenging. Most notably, I had no way of knowing whether a Water Year would be affected by 0, 1, or multiple known gaps in data. My way around this was as described above - instead of looking at each Water Year for gaps that covered it I looked at each gap and took that away from the corresponding Water Years. This meant that a Water Year on 56% complete data when adding a gap of 10% used that previous value and calculated 56 - 10 = 46% complete, as opposed to trying to add all the gaps for days in a specific Water Year THEN take it off 100%.

Once I'd ran my code over all files I noticed the peculiar case that often between one and two Water Years were being considered invalid per gauging station. On the face of it this is an alarming result as it says that for the majority of stations two Water Years out of the usually around 30 are invalid due to incomplete data. However, what was actually happening was that the first and final Water Years were often being rejected due to the fact the data didn't start and stop on 01st October. After going through an example station I realised what the cause was, and adjusted my summary to look for anomalies in number

of discarded Water Years within the middle of the record periods, excluding the first and last Water Years.

Finally, I had some issues with inconsistencies between the AM data provided by Dr Prosdocimi and the POT data I was using. The corresponding tables for the same gauging station were sometimes different lengths, meaning that there must be some Water Years appearing in one but not both of the tables. To deal with this, I altered my functions to include a second input and output variable "inconsist", which was a data frame that documented these scenarios. The inconsist data frame ended up with nearly 3000 rows. The size of the data frame was alarmingly large, so to make the problem more manageable I cut down inconsist into only valid Water Years - ones that had at least 75% complete data. This reduced the data frame to only 348 rows, which, whilst still concerning, was less distressing as it meant the problem was only affecting a small fraction of the data. On closer inspection, this was only happening in 56 of the 263 gauging stations. Furthermore, the AM tables were always shorter than the POT tables, mainly due to times when the annual maxima wasn't over the threshold for that year. When this happened, it is safe to assume the POT data is correct, which is what my code was doing anyway. It took a lot of effort and care to create the "inconsist" data frame, but was worth it to reassure myself that my code worked and to explore all avenues of potential issues. It is worth noting that this fix still does not account for the situation when both the AM and the .PT files contain the same missing information, and this is an issue that I discovered but could not find a comprehensive way of dealing with, and in the Further Steps Section 2.4 I discuss why it is such a difficult problem to overcome.

Chapter 2

Statistical Analysis

In this Chapter I assume the reader is comfortable with GLM theory, and if not I recommend Julian Faraway's textbook "Extending the Linear Model with R" [2], that explains the theory as well as the appropriate packages that I utilised in RStudio. Whilst the theory is interesting, only a knowledge of the concepts for mean and variance, as well as an understanding of confidence intervals, is required to understand the results in Section 2.2.

2.1 Statistical Theory

The cleaned data is in the format of number of POT events per year. In Chapter 4 of "An Introduction to Statistical Modeling of Extreme Values" by Stuart Coles [1], the author discusses the overarching theory for data that is over a set threshold. Under the assumption that the POT events per year are independently distributed from the same distribution, given an adequate threshold the magnitude of the POT data should follow a distribution that asymptotically becomes a member of the generalised Pareto Family. Grouping these events as number of threshold exceedances per Water Year, the data should asymptotically follow a Poisson Distribution. Whilst this is slightly esoteric, for this specific example the result is that:

> Given data $X_1, X_2, ..., X_n$, for a specific value of threshold u the POT data will approximately follow a Poisson Distribution with constant mean λ. The same applies for threshold $u + \epsilon$ for any $\epsilon > 0$, although with a different (smaller) λ.

This theory can also be extended to the modelling of annual maxima [6], however I did not perform any analysis with the AM data set and so do not discuss that theory. It is worth commenting that if the modelling of the POT data is in someway invalid, dependent on the cause any modelling applied to the AM data set could also be invalid for the same reason.

There are two conditions required for the theory to be valid:

1. The data points must be independent. Intuition about flood data may suggest that one would expect years with significantly many POT to happen at the same time, in a wave-like effect.

2. The threshold must be set appropriately. If the threshold is too low, the data will not follow the Poisson Distribution.

For the first point, it is important to know that the NRFA states their POT data accounts for such scenarios of excessive flooding. They do this by when multiple peak flows occur in a single event, the largest flow is used for the POT data file whereas the others are not added. For the second point, the threshold is determined by the NRFA to give on average five POT events per year, meaning λ is expected to be five. This result can be tested analytically, and when this was not the case I had to assess whether it was due to an incorrect threshold setting, a change of circumstance at the gauging station, or another reason altogether.

2.1.1 Extended Quasi-Likelihood Models

The first step when applying the theory to the data is to fit a Poisson GLM with identity link. As the time variable should not be a factor, this fit is merely a horizontal line with intercept that is the sole parameter λ of the Poisson Distribution. In the previous image, Figure 1.6, our λ estimate is 4.111 and can be shown by the horizontal dotted line. As the example shows, it is somewhat difficult to tell intuitively whether or not the average of five POT events per year is met; is an average of 4.111 significantly different? Obviously confidence intervals are required here, but I also needed to estimate whether the variance, which should be λ as well, was behaving as anticipated.

I instead used a quasi-likelihood model that also estimates the dispersion parameter, ϕ. As is standard GLM theory, for a Poisson Distribution the dispersion parameter is 1, because ϕ = dispersion = variance / mean and for Poisson data it follows that variance = mean = λ. A quasi-model instead estimates this parameter, reducing the number of degrees of freedom by one. However, this doesn't allow inference on ϕ, which is what is needed to confidently say the dispersion parameter is different from one and hence the Poisson assumption is invalid.

This naturally leads to applying an extended quasi-likelihood model, which does allow for inference on ϕ. Deriving the extended quasi-likilihood model from the quasi-likelihood model is outlined in Chapter 9 of McCullagh and Nelder [4]. However, I needed to use prior GLM knowledge to derive the corresponding Fisher Information matrix, the details of which are below.

Often, when checking for over-dispersion a statistician would fit a Negative Binomial model to the data, instead of an extended quasi-likelihood model. This would not allow for inference on whether the data is under-dispersed, and hence I opted with the extended quasi-likelihood approach.

2.1.2 Deriving Confidence Intervals

In Chapter 9 of the main textbook I was using [4], the extended quasi-likelihood model for a distribution in the Exponential Family has the following form:

$$Q^+(\mu, \sigma^2; y) = -\frac{1}{2\sigma^2} D(y; \mu) - \frac{1}{2} log\sigma^2 \tag{2.1}$$

where the parameters in Equation (2.1) are given in the Table below.

Symbol	Meaning
Q^+	(log) Extended Quasi-Likelihood function
μ	mean
σ^2	variance
y	response (observation)
D	quasi-deviance

Now, the extended quasi-likelihood function behaves in the same way as a normal likelihood function, in the sense of attaining confidence intervals for parameters. So, by differentiation of the function with respect to the two parameters μ and σ^2, I derived the quasi-score function as seen in Equation (2.2). Note from now on the equations are vectored, so Y is an nx1 vector in the scenario where there are n POT values.

$$u^+ = \begin{bmatrix} \frac{\delta}{\delta\mu}(Q^+) \\ \frac{\delta}{\delta\sigma^2}(Q^+) \end{bmatrix} = \begin{bmatrix} \frac{Y-\mu}{\sigma^2 V(\mu)} \\ \frac{D(Y;\mu)}{2\sigma^4} - \frac{1}{2\sigma^2} \end{bmatrix} \tag{2.2}$$

In Chapter 9 of McCullagh and Nelder [4] the authors comment that σ^2 can be assumed to be small enough to justify that the expectation of the quasi-deviance is approximately σ^2, which then gives the expectation of the quasi-score being 0, parallel to standard GLM theory. The variance function for μ appears when differentiating the quasi-deviance, and whilst this all seems confusing it simplifies when specified to our model, as happens in the second equality of Equation (2.3). This is because instead of having to think about variance functions and extra parameters there are now only the two parameters λ and ϕ,

with $V(\mu) = \mu = \lambda$, and once the (quasi) Fisher Information Matrix $\mathcal{I}^+$ is calculated things are in a much more manageable form.

$$\mathcal{I}^+ = \begin{bmatrix} \frac{1}{\sigma^2 V(\mu)} & 0 \\ 0 & \frac{1}{2\sigma^4} \end{bmatrix} = \begin{bmatrix} \frac{1}{\phi\lambda} & 0 \\ 0 & \frac{1}{2\phi^2} \end{bmatrix} \tag{2.3}$$

The reason for calculating $\mathcal{I}^+$ is to invert it to extract the variance of the parameters, in line with GLM theory. This can be seen below.

$$|\mathcal{I}^+| = \frac{1}{2\phi^3\lambda} \tag{2.4}$$

$$(\mathcal{I}^+)^{-1} = |\mathcal{I}^+|^{-1} \begin{bmatrix} \frac{1}{2\phi^2} & 0 \\ 0 & \frac{1}{\phi\lambda} \end{bmatrix} = 2\phi^3\lambda \begin{bmatrix} \frac{1}{2\phi^2} & 0 \\ 0 & \frac{1}{\phi\lambda} \end{bmatrix} = \begin{bmatrix} \phi\lambda & 0 \\ 0 & 2\phi^2 \end{bmatrix} \tag{2.5}$$

It is known for a vector of parameters β from which the Information Matrix $\mathcal{I}$ is derived that the true values of those parameters follow the distribution:

$$\beta \sim N_2(\beta, (\mathcal{I}^+)^{-1}) \tag{2.6}$$

In this case $\beta = [\lambda, \phi]^t$ and so combining this with (2.5) and (2.6) I found that:

$$\lambda \sim N(\hat{\lambda}, \hat{\phi}\hat{\lambda}) \tag{2.7}$$

$$\phi \sim N(\hat{\phi}, 2\hat{\phi}^2) \tag{2.8}$$

where parameters with hats on are estimated values from the data. So I could now get the desired confidence intervals for the mean and dispersion. Applying the Central Limit Theorem to the data of size n I arrived at the following confidence intervals:

$$95\% \text{ C. I. for } \lambda = \hat{\lambda} \pm t_{n-2,0.975} \frac{\sqrt{\hat{\phi}\hat{\lambda}}}{\sqrt{n}} \tag{2.9}$$

$$95\% \text{ C. I. for } \phi = \hat{\phi} \pm t_{n-2,0.975} \frac{\sqrt{2}\hat{\phi}}{\sqrt{n}} \tag{2.10}$$

Where the t in the above equations is the appropriate quantile from a t distribution, with the n-2 degrees of freedom being due to the 2 parameters being estimated, and the 0.975 being to estimate a 95% confidence interval.

2.2 Applying to the Data

I applied the aforementioned theory to the cleaned data sets, and analysed the results. All code I created and ran can be seen in the Supplementary Section 3.2. Certain gauging stations gave abnormal results, and they are discussed later in Section 2.3.

2.2.1 Dispersion Parameter (ϕ) Confidence Intervals

I added 2 columns to disp_table to have the lower and upper limits for a 95% confidence interval for ϕ, the dispersion parameter. The main reason extended quasi-likelihood was employed as it allows inference on the dispersion, to search for both over and under-dispersion.

Out of the 263 analysed stations, with 95% confidence 132 (50.2%) had dispersion estimates significantly greater than 1. That is to say for over half the stations the lower limit of the dispersion 95% confidence interval was strictly greater than 1. This brings doubt into the assumptions made about the data, and is a significant result. I looked at the most over-dispersed stations individually, shown in Section 2.3, but having over 100 meant checking all was not possible.

However, only 3 of the 263 had significantly under-dispersed data, found by their upper confidence limit being strictly less than 1.

In Figure 2.1 these results are displayed visually. An advantage of plotting a geographical map is that it shows there appears to be a trend in more over-dispersed gauging stations being in the southern regions of England, as well as a clump in the middle of Scotland. This is probably due to the geological make-up of the soil in different areas, and so a high dispersion doesn't immediately say the gauging station data is invalid, only that one must be wary during analysis that there will be natural over-dispersion.

2.2.2 Rate Parameter (λ) Confidence Intervals

Using the same process as in the subsection 2.2.1 above, I calculated the λ confidence intervals for the 263 cleaned gauging stations. It is more difficult to classify what counts as an abnormal result for the λ values, which in the model are the average POT per Water Year. This is because though the threshold is set to produce on average five POT, it is difficult to claim that 4.9 is truly significant, and it is probable that the threshold determination has changed over time from how many POT *should* have been assigned.

Instead, this can be thought of as a quality control measure; is the data consistently close enough to five POT for it to prove accurate to claim that is how the threshold functions? In parallel to the previous analysis, I created the equivalent location map which

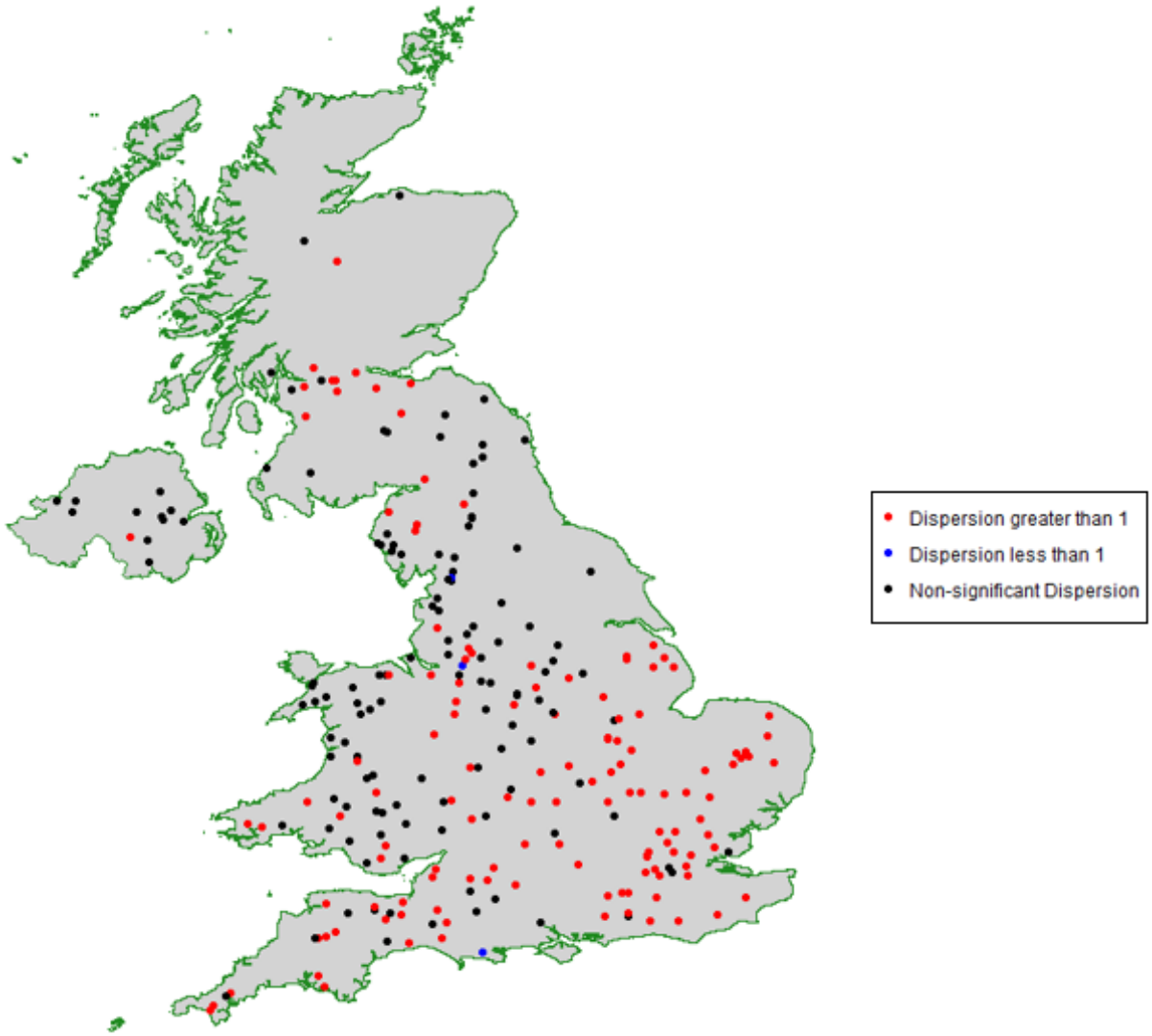

Figure 2.1: Identifying stations that were over-dispersed and under-dispersed.

can be seen in Figure 2.2. As a comparison, only 7 (2.7%) stations had average POT values greater than 6, and 34 (12.9%) had average POT values less than 4.

I could be confident that my λ parameter values were correct because I could check them exactly using the "quasipoisson" family when fitting the appropriate generalised linear model in R. I could not do the same for the dispersion inference, but the mean being correct was reassuring as it meant I had not made a calculation error in deriving the (quasi) Fisher Information matrix, and hence the dispersion confidence intervals are likely also correct.

There doesn't appear to be a connection between whether the data is over-dispersed and whether the mean (average POT) is significantly different from five. The correlation between the two variables is only 0.18, which suggests a positive association but not a huge one. The $\hat{\phi}$ parameter does appear in the confidence interval calculation for λ as can be seen in Equation (2.10), but otherwise I can see no real reason why the two parameters should affect each other.

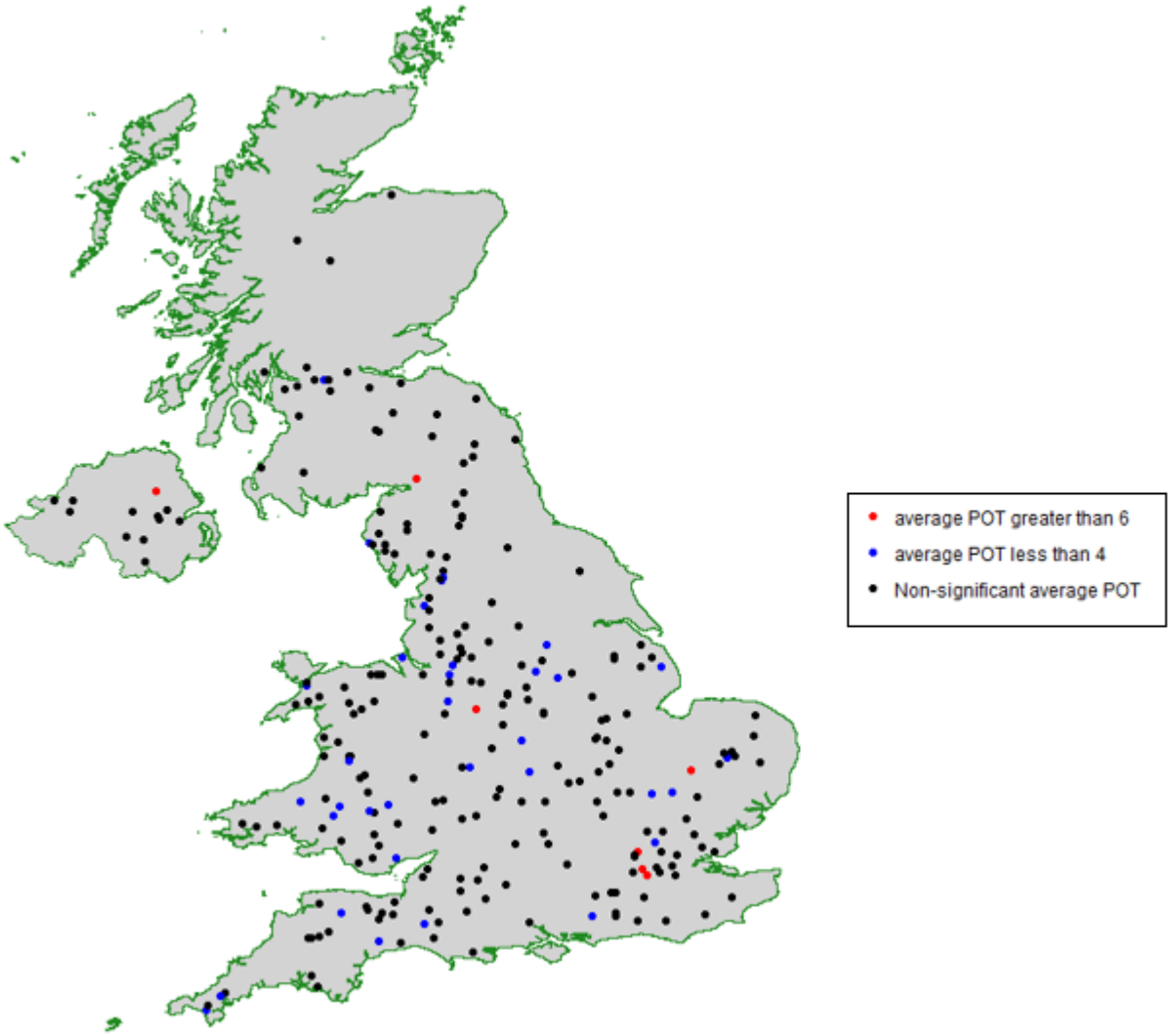

Figure 2.2: Identifying stations that had, with 95% confidence, average POT values more than 1 different than the expected value of five that they were designed to have.

2.3 Problematic Stations

In this section I explore the gauging stations that in some sense were extreme. It is also discussed whether the reason for the extreme values are due to the data being of poor quality, for example too many missing values for reliable results, or a different reason such as incorrect assumptions or inappropriate setting of the threshold.

For stations where I was 95% confident that the true average POT value was more than one away from the desired value of five, it is clear that the stations are problematic. There were 41 stations that fell into this category and have problems with either the threshold or unreported missing data, and they are listed in Table 2.1.

For stations with over-dispersion, the same conclusions can not necessarily be drawn about the data being problematic. This is because it could be down to a situation geographically that is not to do with the data being incorrect. However, where the dispersion is significantly high it is indeed worth examining what could be the cause.

Another point is that I decided to classify stations with fewer than five valid Water Years to contain insufficient data for analysis, and of the 263 the two where this was the case were station numbers 25810 and 32029, that only had three valid Water Years.

2.3.1 Low average POT

There was no significant association between the threshold set by the NRFA and whether or not the POT was low.

I looked at the three gauging stations with the lowest POT confidence intervals to see if there was a problem with data recording. These all had average POTs significantly less than 2.5, and were station number 27030, 68003, and 68010. Figure 2.3 shows the cleaned data for these stations, all on the same time x-axis and y-axis scale for comparison.

The first thing that one notices in Figure 2.3 is that station 68010 had only five valid Water Years, and they were 1973 then 1976-1979. Two comments I have are that this data is old, and that there are very few valid Water Years - five was the minimum I was allowing to justify analysis. On further investigation I discovered the gauging station only was recording between 12th September 1973 and 03 June 1981, and had 26 instances of known gaps in the data of that 8 year period. Hence an abnormal POT estimate is almost certainly due to the data being of poor quality and throws up questions, the first being "why isn't there any data after 1981?". A probable answer is that the gauging station has stopped being used.

For the stations 27030 and 68003 there were 51 and 32 valid Water Years respectively, so the problem was not caused by too few data points. For both stations the dispersion wasn't significantly different from 1. The maximum POT in any one year was only 3

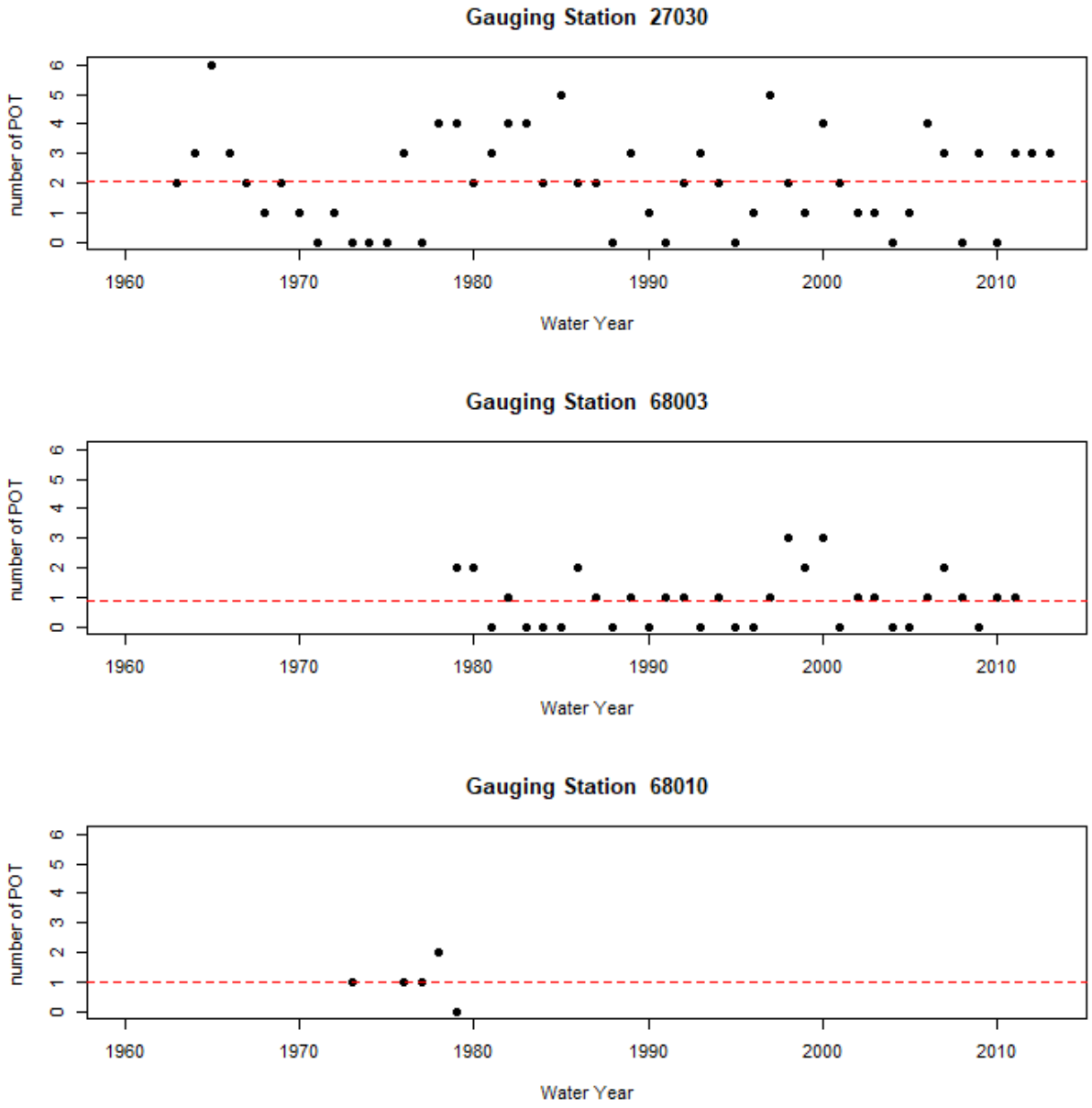

Figure 2.3: The three gauging stations with the lowest average POT, shown as a red dotted line. All had confidence intervals stating the true average POT for that gauging station and given threshold is less than 2.5, when the threshold is designed to produce an average of 5.

for 68003, and this gives strong evidence that the threshold has been set incorrectly. An advantage of plotting is that you can look for a time trend, and it doesn't appear that this is the case for these stations. Hence the problem probably isn't due to the fact the parameter λ is changing over time. Having said that, it should be noted there could still be dependence of events leading to clusters of POT happening at the same time; information which cannot be explicitly determined from these plots.

2.3.2 High average POT

I decided to look at the four gauging stations with the highest POT confidence intervals; these were the ones with confidence intervals above the point 6.5. When carrying out the same process as in the previous section, I arrived at the plots seen in Figure 2.4.

The two plots on the left-hand side of the Figure, stations 39003 and 39049, appear to have a clear increase in average POT per year as time moves to the right. Most notably, station 39049 appears to have a significant shift between the 1985 and 1986 Water Years, from averaging 2 POT before that date to averaging 18.5 after it. This suggest a change in the situation in which the gauging station was placed, such as water from another river being redirected into the gauging station's river upstream.

What is interesting is that all four of these stations have over-dispersion. Whilst this behaviour can be seen in Figure 2.4 the statistics also back this up, with confidence intervals for the dispersion ϕ all being greater than 1. It is hard to believe that, if the model is correct, then there would be Water Years with up to 35 POT events per year, and so whilst it is difficult to pinpoint where the assumptions fail I can be confident that something is not right. In the next part, Section 2.3.3, I look at the most significant over-dispersed stations and similar conclusions are drawn.

2.3.3 Examining the most over-dispersed Stations

There were 12 gauging stations with dispersion confidence intervals greater than 2.5. On further investigation it appeared there were two distinct cases for the cause of these incredibly high over-dispersed values.

The first is where there appears to be a large period of time where there are no POT per year. This can be seen in Station 47011, in Figure 2.5. Initially, it appears that there is missing data. I checked both the .PT files and the Annual Maxima data given by Dr Prosdocimi and they are consistent in that there is no data over those periods, but no notification that this is due to the station not recording. However, in this instance it is clear that the station wasn't recording. Having said that, if the station had been recording it though the dispersion may be lower the number of POT, our λ, appears to be high, around the mark of 10 POT per year.

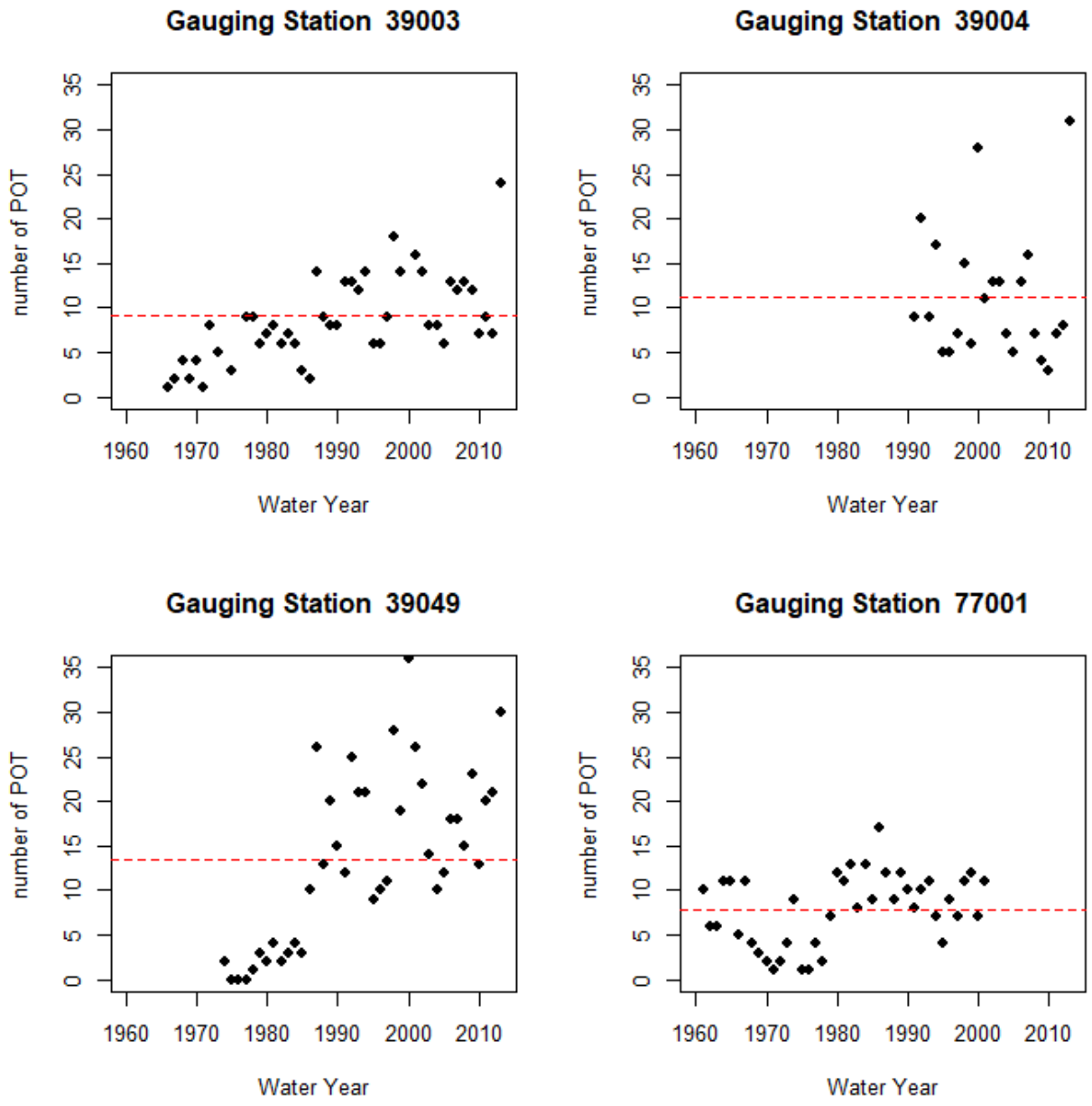

Figure 2.4: The four gauging stations with the highest average POT, shown as a red dotted line. Note the trend in the bottom left image, where it is clear that time is not independent.

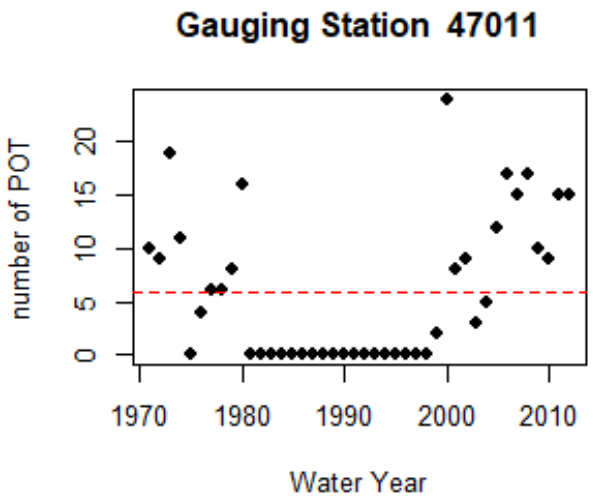

Figure 2.5: This station has dispersion confidence interval (4.39, 11.34). On investigation I discovered there were no POT nor annual maxima data points between September 1981 and April 2000. This is almost certainly due to there being missing data in both .PT and AM file that has not been identified.

The second cause for over-dispersion appears to be natural change in circumstance. An example of this form of change is Station 39003, which a slightly different representation is given in Figure 2.6 to show the effect. I did not use this kind of plot often, as it required delving back into the .PT file as opposed to using my cleaned data set, but it proved useful in this situation.

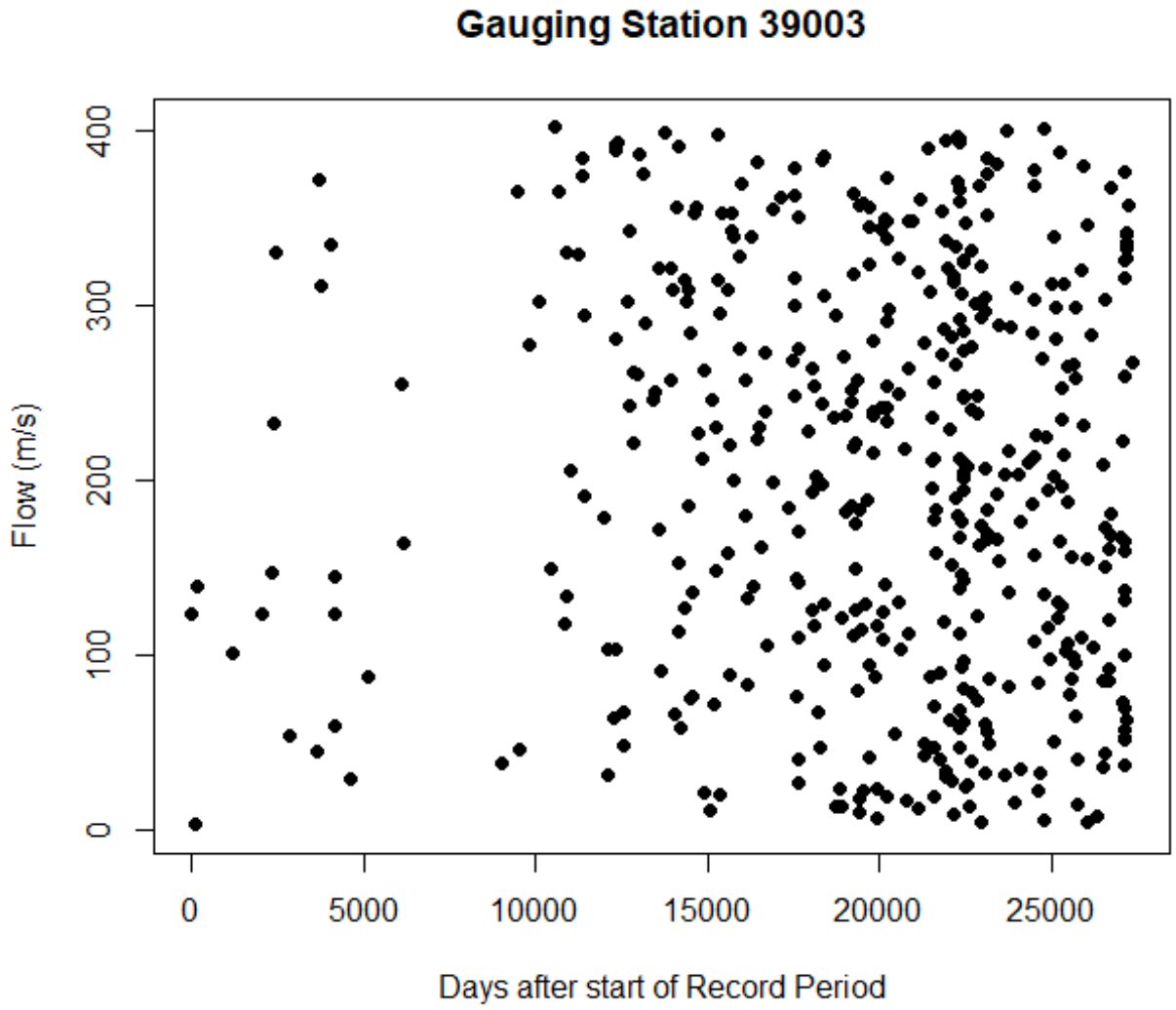

Figure 2.6: This station has dispersion confidence interval (2.62, 6.42). The image shows a marked increase in number of POT after roughly 10 000 days into the Record Period. This may be due to an environmental change. However, the Flow range does not have appeared to change before and after the 10 000 day mark, which is interesting.

2.3.4 Listing Problematic Stations

Here I document the stations out of the 263 that I believe have poor data quality, either due to incorrect threshold setting or untold missing data. There are the 41 stations with significantly different λ confidence intervals from the five that the threshold is designed to have, and they can be seen in Table 2.1 on the following page. Stations with large dispersion parameters could also qualify, though I have not explicitly detailed them here as the cut-off point is not clear. In the scenario of checking data quality, I would advise individually examining station scenarios for the stations with largest dispersion parameters, as I have done and shown in the previous Section.

	Station Number	$\hat{\lambda}$	Confidence Interval
1	203027	7.10	(6.29, 7.91)
2	27030	2.04	(1.6, 2.48)
3	27052	2.69	(1.99, 3.39)
4	28026	3.11	(2.47, 3.75)
5	28032	2.58	(1.59, 3.56)
6	28083	7.37	(6.13, 8.61)
7	29002	2.52	(1.68, 3.36)
8	33023	7.77	(6.19, 9.36)
9	33027	1.98	(1.21, 2.75)
10	33044	2.64	(1.82, 3.47)
11	33055	2.50	(1.44, 3.56)
12	38021	2.74	(1.56, 3.92)
13	39003	9.22	(7.3, 11.13)
14	39004	11.26	(8.08, 14.44)
15	39049	13.55	(10.47, 16.63)
16	41011	3.04	(2.23, 3.86)
17	45013	3.11	(2.32, 3.91)
18	48006	1.95	(1.21, 2.7)
19	48803	1.89	(1.05, 2.73)
20	50012	3.20	(2.43, 3.97)
21	52007	3.36	(2.74, 3.98)
22	54006	2.04	(1.29, 2.8)
23	54019	2.92	(2.24, 3.59)
24	55010	2.94	(2.16, 3.72)
25	55025	3.03	(2.27, 3.78)
26	56013	3.23	(2.5, 3.96)
27	57008	3.17	(2.52, 3.82)
28	60007	2.53	(1.99, 3.07)
29	60009	2.66	(1.83, 3.49)
30	62002	2.55	(1.7, 3.4)
31	65004	3.25	(2.61, 3.89)
32	68003	0.88	(0.56, 1.2)
33	68005	2.93	(2.14, 3.73)
34	68010	1.00	(0,.00 2.01)
35	68011	2.17	(1.31, 3.02)
36	72002	2.72	(1.83, 3.62)
37	72003	2.43	(1.82, 3.03)
38	72009	3.12	(2.56, 3.68)
39	74005	3.10	(2.45, 3.75)
40	77001	7.88	(6.62, 9.14)
41	84015	3.22	(2.54, 3.91)

Table 2.1: The 41 gauging stations, ordered by Station Number, that had confidence intervals for λ strictly outside [4,6].

2.4 Further Steps

There is still much more interesting work and analysis to be carried out on the data files provided by the NRFA.

Firstly, I only cleaned the best-quality data that was on offer, the "Suitable for QMED" gauging stations. In these I still found errors in the data, but didn't find a consistent way to flag gauging stations with poor data. One of the common versions of misinformation occurs in the scenario where there are no POT for a number of consecutive Water Years, as well as no Annual Maxima, but the data hasn't been flagged as missing so one incorrectly assumes there just weren't any events that year.

Given more time, I would love to develop a data cleaning process that works for all Stations published by the NRFA, even the "Not Suitable for Pooling or QMED" ones. This would involve clearly defining criteria for what counts as probably incorrect data recording. I didn't personally look at the annual maxima data not the Stage values (that give the height in metres of the river), and these could prove an interesting piece of the puzzle to analyse.

Whilst my checks against the AM file ensured the .PT file was consistent and not outputting incorrect information, I have not managed to develop a check for when both files have missing data that has not been identified. An example when it is clear this has happened is for Station 47011, shown in Figure 2.5, where there are nearly 20 years of unidentified missing data. The reason I found this hard to classify in coding is that it is dependent on the surrounding data. For example, for station 47001 most other years have over five POT, and so multiple zeros is noticeable. But for a gauging station with a far lower average number of POT, it is very hard to know whether there genuinely weren't any POT for that period. Also, continuing with the Station 47001, if I decided to exclude those 19 years must I then also exclude the only other Water Year with zero POT values, 1975, because it may also be missing data? The answers to these questions I believe are very station-specific, but if I were to develop this project further I would try to formulate a more comprehensive checking of missing data that can spot such problems as those that occur in Station 47011.

It would be interesting to see for the gauging stations with λ values significantly different from 5, whether a change in the threshold would correct that situation. Given the data it is possible to increase the threshold, however I could not decrease the threshold seeing as only the POT events are published for the given threshold, so I didn't have the information on flow less than the preset threshold. Also, the threshold may not be appropriate due to a circumstantial change such as permanent changes being made to a river, and if these were known occurrences one could look at adjusting the threshold accordingly at these key times so the data still followed the desired distribution. However, if this is done multiple times then the model is meaningless and presumably there is some form of time association, so

	Average Rainfall	Urban	Permeability	Dispersion
Average Rainfall	1.00	-0.16	-0.36	-0.25
Urban	-0.16	1.00	0.16	0.35
Permeability	-0.36	0.16	1.00	0.28
Lower CI for Dispersion	-0.25	0.35	0.28	1.00

Table 2.2: Correlation Matrix for certain catchment descriptors and the lower confidence level for the dispersion parameter.

whilst an interesting concept one would need to be very cautious.

Another interesting piece of analysis that could be carried out would be to find the associations between certain aspects of the catchment area the gauging stations are in, and comparing it to the dispersion value ϕ. Using the lower limit of the ϕ confidence interval I looked at the correlation matrix of these criteria, shown in Table 2.2, but further analysis would be desirable. Initial analysis shows that a higher dispersion value is negatively correlated to the Standard-period Average Annual Rainfall (SAAR), and positively correlated to increased permeability of the soil in the catchment area and how urban the area is.

Conclusion

I initially set out to clean the POT data files for the "Suitable for QMED" gauging stations across the UK, and analyse whether or not the number of POT per Water Year followed a Poisson Distribution with rate $\lambda = 5$.

After developing a bespoke method for cleaning .PT files, flagging up potential problems, my code was created to then output the POT data in the appropriate format of number of POT per valid Water Year. On the way, I overcame many challenges from missing data to different formats. I created the following criteria for the data:

1. for a Water Year to be valid, it must have less than 25% of the year containing known missing data.

2. to be considered for analysis, a gauging station must have at least 5 Valid Water Years.

3. if there were no POT for a given Water Year, the corresponding result must be validated in the Annual Maxima data.

Once cleaned, I applied inference to the two parameters of interest: λ and ϕ. If the Poisson Distribution was appropriate, then the ϕ parameter would be 1. However, in approximately 50% (132 out of 263) of the gauging stations the data was over-dispersed. Conversely, only 3 stations had significantly under-dispersed data.

In many cases the cause for the over-dispersion was due to changes in circumstance at the gauging station, which is reasonable as some have been recording data for over 50 years. This invalidates the setting of the threshold to average five POT events per year, as in this scenario time is dependent. A potential solution would be to change the threshold appropriately if the cause for the change is known, however doing this often negates the threshold's purpose.

In other cases, the over-dispersion was due to misinformation about the data, such as long periods of gaps not mentioned as being due to missing data. This may confirm the perceptions that POT data can be of a poor quality.

Similarly, the λ inference showed that often the true number of POT events per year was different from 5, due to either change in circumstance, poor quality data, or peaks not being independent.

In summary: this project successfully developed a method for cleaning .PT files published by the NRFA into an analysable format, and performed it on 263 data sets. I also adapted statistical theory to perform inference on the parameter estimates, identifying 41 out of the 263 stations where the Poisson Distribution was invalid, as well as setting criteria for when stations should be checked individually.

Chapter 3

Supplementary Section

3.1 R Code for Cleaning the Data

```
##############################
#This script takes data from the point I was at at the
#start of the project right through to the cleaned data sets
#"events_table" and "disp_table"
##############################



####Script 1.0 - preparing workspace####

##Set Working Directory
setwd("C:/Users/my laptop/Google Drive/Flood Folder")
##Load Workspace v1_outs (Dr Prosdocimi's code)
load("C:/Users/my laptop/Google Drive/Flood Folder/v41_outs.RData")
##POTfilenames contains as characters the directory path for all
#.PT files
POTfilenames <- c(list.files(path="WINFAP-FEH_v4.1/Suitable_for_QMED",
                             pattern="*.PT", full.names=TRUE))
NF<-length(POTfilenames)
NF #263 valid POT files to work with



####Script 1.1 - ave_POT and valid_data####
library(lubridate) #for simple date formatting
```

```
Water_Year <- function(date){
  #Given a date in ymd lubridate form, returns the WY that date is in.
  #The XXXX Water year starts 01st Oct XXXX and ends 30th Sep (XXXX+1)
  if (month(date) < 10){
    return(year(date) - 1)
  } else {year(date)}
}

Gap_Percent <- function(beg, end){
  #Given 2 dates in ymd lubridate form IN THE SAME WATER YEAR, returns
  #the percentage of that WY that they span
  if (Water_Year(beg) != Water_Year(end)){
    stop('Gap_Percent problem - dates not in same WY')
  } else if ((end - beg)[[1]] < 0){
    stop('Gap_Percent problem - beg is after end')
  } else {
    gap_days <- (end - beg)[[1]]
    leap <- leap_year(Water_Year(beg)+1) #check for leap year
    gap_percent <- 100 * gap_days / (365+leap)
  }
  return(gap_percent)
}

valid_data <- function(POTtable, inconsist){
  ##########
  ##Given a POT table outputs whether there is usable data for each
  ##individual WY

  ##Does this by checking whether no data for a year means (untold)
  ##missing data or there genuinely wasn't any POT that year.
  ##Creates new table with each row as the WY, and outputs each WY
  ##with whether there is valid data for that year.
  ##Returns as a table.
  ##########

  ########
  ##Create table shell using beginning and ending record periods
  beg <- POTtable$V2[POTtable$V1 == "Record Period"]
  end <- POTtable$V3[POTtable$V1 == "Record Period"]
  #make use of lubridate package
  beg <- dmy(beg); end <- dmy(end)
  WYvec <- c(year(beg) : year(end))
  if (month(beg) < 10){ #starts in previous WY
```

```
  WYvec <- append(year(beg) - 1 , WYvec)
}
if (month(end) < 10){ #ends in previous WY
  WYvec <- head(WYvec, -1)
}
#Create WY table shell
WYtable <- data.frame(WY = WYvec, "percent.complete" = 100,
                      "AM.vs.Thresh" = 0.5, "WYvalid" = 1)
#First and Last percent.complete's aren't 100% as recording is only
#for part of WY
WY_1 <- ymd(paste(Water_Year(beg)+1,"Sep",30))
WYtable$percent.complete[1] <- Gap_Percent(beg, WY_1)
WY_2 <- ymd(paste(Water_Year(end), "Oct", 01))
WYtable$percent.complete[length(WYtable$WY)] <-
  Gap_Percent(WY_2, end)
########

########
##Iterate through known missing data gaps to append
##"% complete data" column in WYtable

POT.pos <- match('[POT Gaps]',POTtable[,1]) #should be 8 or NA
if (!is.na(POT.pos)){ #There is at least one known gap
  num.gaps <- match('[End]',POTtable[-c(1:POT.pos),1]) - 1
  #Start the iteration over the known gaps in data
  for (i in 1 : num.gaps){
    beg <- POTtable[POT.pos + i,1]
    end <- POTtable[POT.pos + i,2]
    #make use of lubridate package
    beg <- dmy(beg); end <- dmy(end)
    #Use Water_Year function
    diff <- Water_Year(end) - Water_Year(beg)

    #3 cases here - when the gap  spans 0, 1, and >1 WY
    #CASE1
    if (diff == 0){
      gap_percent <- Gap_Percent(beg,end)
      #update WYtable
      temp <- WYtable$percent.complete[WYtable$WY ==
                                         Water_Year(beg)]
      temp <- temp - gap_percent
      WYtable$percent.complete[WYtable$WY == Water_Year(beg)] <-
        temp
```

```
}
#CASE2
if (diff == 1){
  WY_split <- ymd(paste(Water_Year(end),"Oct",01))
  #update First WY
  gap_percent <- Gap_Percent(beg,WY_split-1)
  #the "-1" above is because 1st Oct is in next WY
  temp <- WYtable$percent.complete[WYtable$WY ==
                                     Water_Year(beg)]
  temp <- temp - gap_percent
  WYtable$percent.complete[WYtable$WY == Water_Year(beg)] <-
    temp
  #update Second WY
  gap_percent <- Gap_Percent(WY_split, end)
  temp <- WYtable$percent.complete[WYtable$WY ==
                                     Water_Year(end)]
  temp <- temp - gap_percent
  WYtable$percent.complete[WYtable$WY == Water_Year(end)] <-
    temp
}
#CASE3
if (diff > 1){
  #Update First WY
  WY_1 <- ymd(paste(Water_Year(beg)+1,"Oct",01))
  gap_percent <- Gap_Percent(beg,WY_1-1)
  #the "-1" above is because 1st Oct is in next WY
  temp <- WYtable$percent.complete[WYtable$WY ==
                                     Water_Year(beg)]
  temp <- temp - gap_percent
  WYtable$percent.complete[WYtable$WY == Water_Year(beg)] <-
    temp
  #Update Last WY
  WY_2 <- ymd(paste(Water_Year(end),"Oct",01))
  gap_percent <- Gap_Percent(WY_2, end)
  temp <- WYtable$percent.complete[WYtable$WY ==
                                     Water_Year(end)]
  temp <- temp - gap_percent
  WYtable$percent.complete[WYtable$WY == Water_Year(end)] <-
    temp
  #Update all middle WYs (that have no data!!)
  WY_no_data <- c( (Water_Year(beg)+1) : (Water_Year(end)-1) )
  WYtable$percent.complete[WYtable$WY %in% WY_no_data] <- 0
}
```

```
    }
  }
  #We now repeat the process (previous 55 lines) but for
  #[POT Rejected] instead of [POT Gaps]. Can't use one or the
  #other in case both occur in one file.
  POT.pos <- match('[POT_Rejected]',POTtable[,1])
  if (!is.na(POT.pos)){ #There is at least one known gap
    num.gaps <- match('[END]',POTtable[-c(1:POT.pos),1]) - 1
    #^^^ for some reason is "END" not "End"
    #Start the iteration over the known gaps in data
    for (i in 1 : num.gaps){
      beg <- POTtable[POT.pos + i,1]
      end <- POTtable[POT.pos + i,2]
      #make use of lubridate package
      beg <- dmy(beg); end <- dmy(end)
      #Use Water_Year function
      diff <- Water_Year(end) - Water_Year(beg)

      #3 cases here - when the gap  spans 0, 1, and >1 WY
      #CASE1
      if (diff == 0){
        gap_percent <- Gap_Percent(beg,end)
        #update WYtable
        temp <- WYtable$percent.complete[WYtable$WY ==
                                           Water_Year(beg)]
        temp <- temp - gap_percent
        WYtable$percent.complete[WYtable$WY == Water_Year(beg)] <-
          temp
      }
      #CASE2
      if (diff == 1){
        WY_split <- ymd(paste(Water_Year(end),"Oct",01))
        #update First WY
        gap_percent <- Gap_Percent(beg,WY_split-1)
        #the "-1" above is because 1st Oct is in next WY
        temp <- WYtable$percent.complete[WYtable$WY ==
                                           Water_Year(beg)]
        temp <- temp - gap_percent
        WYtable$percent.complete[WYtable$WY == Water_Year(beg)] <-
          temp
        #update Second WY
        gap_percent <- Gap_Percent(WY_split, end)
        temp <- WYtable$percent.complete[WYtable$WY ==
```

```
                                                    Water_Year(end)]
      temp <- temp - gap_percent
      WYtable$percent.complete[WYtable$WY == Water_Year(end)] <-
        temp
    }
    #CASE3
    if (diff > 1){
      #Update First WY
      WY_1 <- ymd(paste(Water_Year(beg)+1,"Oct",01))
      gap_percent <- Gap_Percent(beg,WY_1-1)
      #the "-1" above is because 1st Oct is in next WY
      temp <- WYtable$percent.complete[WYtable$WY ==
                                                    Water_Year(beg)]
      temp <- temp - gap_percent
      WYtable$percent.complete[WYtable$WY == Water_Year(beg)] <-
        temp
      #Update Last WY
      WY_2 <- ymd(paste(Water_Year(end),"Oct",01))
      gap_percent <- Gap_Percent(WY_2, end)
      temp <- WYtable$percent.complete[WYtable$WY ==
                                                    Water_Year(end)]
      temp <- temp - gap_percent
      WYtable$percent.complete[WYtable$WY == Water_Year(end)] <-
        temp
      #Update all middle WYs (that have no data!!)
      WY_no_data <- c( (Water_Year(beg)+1) : (Water_Year(end)-1) )
      WYtable$percent.complete[WYtable$WY %in% WY_no_data] <- 0
    }
  }
}
########

########
##Fill in "AM.vs.Thresh" and WYvalid columns

WYtable$WYvalid[WYtable$percent.complete < 75] <- 0
#Find threshold
thresh <- POTtable$V2[POTtable$V1 == "Threshold"]
thresh <- as.numeric(as.character(thresh))
#^^^ so comparisson vs numerics can be made later
statno <- as.character(POTtable[2,1])
AMtable <- allAmax[allAmax$Station == statno, ]
#Trim AMtable if needed (in case it contains data outside
```

```
#of POT range)
firstWY <- head(WYtable$WY,1)
lastWY <- tail(WYtable$WY,1)
AMtable <- AMtable[AMtable$WaterYear >= firstWY,]
AMtable <- AMtable[AMtable$WaterYear <= lastWY,]
#Now ready to fill in AM.vs.Thresh column


if (dim(AMtable)[1] == dim(WYtable)[1]){
  WYtable$AM.vs.Thresh <- AMtable$Flow[AMtable$WaterYear ==
                                         WYtable$WY]
} else {#warning('WY inconsistency between AM and POT tables')
  for (k in 1 : dim(WYtable)[1]){
    if (WYtable$WY[k] %in% AMtable$WaterYear){#no problem
      WYtable$AM.vs.Thresh[k] <- AMtable$Flow[AMtable$WaterYear ==
                                                WYtable$WY[k]]}
    else{#case where problem occured - there is no AM for that year!
      WYtable$AM.vs.Thresh[k] <- 0} #
    #^^^we assume AM not higher than threshold
    #update inconsistency data frame, to show where the problem was
    inconsist <- rbind(inconsist,
                       c(statno, dim(AMtable)[1], dim(WYtable)[1],
                         WYtable$percent.complete[k]))
  }}

#Make AM.vs.Thresh an indicator vairable taking values:
#0 = AM not higher than Threshold
#1 = AM higher than threshold
WYtable$AM.vs.Thresh <- WYtable$AM.vs.Thresh > thresh
########

########
##Check WYs with no POT - are they correct or actually unknown
##missing data?

POTdates <- POTtable[-c(1:match('[POT_Values]',POTtable[,1])), 1]
POTdates <- head(POTdates,-1) #removes last element which is "[End]"
POTdates <- dmy(POTdates)
POTdates <- sapply(POTdates,Water_Year)
POTdates <- unique(POTdates)
noPOT <- WYtable$WY[!(POTdates %in% WYtable$WY)]
#Do the comparisson against "AM.vs.Thresh"
if (length(noPOT) != 0){ #there were WYs with no POT data
```

```
    for (i in 1 : length(noPOT)){
      if (WYtable$percent.complete[WYtable$WY == noPOT[i]] > 75) {
        #have enough valid data
        if (WYtable$AM.vs.Thresh[WYtable$WY == noPOT[i]]){
          #we are missing data
          WYtable$WYvalid[WYtable$WY == noPOT[i]] <- 0
          stop('AMax is higher than thresh but no POT for that year')
        } else { #this is a valid entry of no POT for that year
          WYtable$WYvalid[WYtable$WY == noPOT[i]] <- 1
        }
      }
    }
  }
  ##################
  return(list(WYtable, inconsist))
}

ave_POT <- function(POTtable, inconsist){
  ##########
  ##Given the POT data outputs the average POT over the valid WYs

  #Does this by utilising valid_data to get the WY table, then creates
  #the events_table which has the number of events for each valid WY.
  #events_table is outputted
  ##########


  #Get the WYtable using valid_data function
  WYtable <- valid_data(POTtable, inconsist)[[1]]

  statno <- as.character(POTtable[2,1])
  #create events_table shell
  events_table <- data.frame(statno = statno,
                             validWY =
                               WYtable$WY[WYtable$WYvalid == 1],
                             num_events = 0)

  dates_start <- which(POTtable$V1 == "[POT Values]")
  dates <- POTtable$V1[-c(1:dates_start)] #dates where POT happened
  dates <- head(dates, -1) #remove "[End]" which is always last entry
  WYs <- sapply(dmy(dates), Water_Year)

  #calculate how many events there were in each WY
```

```
  events_table$num_events <- sapply(events_table$validWY,
                                    function(x) sum(WYs == x))

  return(events_table)
}



####Script 1.2 - Creating events_table and disp_table####

#Initialise - first run i = 1 then use rbind to get large events_table
POTtable <-read.csv(POTfilenames[1],header=0)
statno <- as.character(POTtable[2,1])
#initialise inconsistency table - to look for missing AM values
inconsist <- data.frame(statno=0, numAM=0, numWYPOT=0,per.comp=0)
events_table <- ave_POT(POTtable, inconsist)

#loop over all .PT files
for (i in 2 : NF){
  POTtable <-read.csv(POTfilenames[i],header=0)
  statno <- as.character(POTtable[2,1])
  inconsist <- rbind(inconsist, valid_data(POTtable,data.frame(
    statno=0, numAM=0, numWYPOT=0,per.comp=0))[[2]])
  temp_events <- ave_POT(POTtable, inconsist) #temporary events_table
  events_table <- rbind(events_table, temp_events)
}

#Look at inconsistencies - is this something to worry about?
inconsist <- inconsist[inconsist$statno != 0,]
inconsist$per.comp <- round(as.numeric(inconsist$per.comp),2)
inconsist$diff <- as.numeric(inconsist$numAM) -
  as.numeric(inconsist$numWYPOT)
head(inconsist)
summary(inconsist$diff)
dim(inconsist) #2974

#I'm now looking if the fact they have <75% complete then we don't
#have to bother with them at all as those Water Years will be excluded
inconsist_comp <- inconsist[inconsist$per.comp >= 75,]
dim(inconsist_comp) #348
#So that has reduced our problem by a factor of 10 but it still occurs
length(unique(inconsist_comp$statno)) #56 gauging stations
```

```
summary(as.numeric(inconsist_comp$diff))
#So the difference is always negative, meaning there are more
#Water Years for the .PT files than for the .AM files.
#This fits intuition (see Report), although it relies on
#the AM data being correct, which isn't necessarily true.

#Now create the dispersion parameter table:
disp_table <- data.frame(statno = unique(events_table$statno),
                         mean = 0, var = 0, disp = 0)
#Calculate mean and variance using tapply
disp_table$mean <- tapply(
  events_table$num_events,events_table$statno, mean)
disp_table$var <- tapply(
  events_table$num_events,events_table$statno, var)
#dispersion is var/mean.
#A dispersion of 1 fits the Poisson assumptions.
disp_table$disp <- disp_table$var / disp_table$mean
```

3.2 R Code for Performing the Analysis

```
############################
#This script takes data from "events_table" and "disp_table"
#and performs statistical analysis
############################



####Script 2.1 - make parameter confidence intervals####

#pre-set columns to be filled
disp_table$disp_CI_lower <- 0; disp_table$disp_CI_upper <- 0
disp_table$lambda_CI_lower <- 0; disp_table$lambda_CI_upper <- 0
#very quick loop over each station as there's no downloading involved
for (i in 1 : dim(disp_table)[1]){
  #set up the parameters
  statno <- disp_table$statno[i]
  temp_events <- events_table[events_table$statno == statno, ]
  phi <- disp_table$disp[disp_table$statno == statno]
  n <- dim(temp_events)[1]
  lambda <- disp_table$mean[disp_table$statno == statno]
  #Calculate the confidence interval for dispersion
  disp_CI <- phi + c(-1,1) * qt(0.975, n-2) * sqrt(2) * phi / sqrt(n)
  disp_table$disp_CI_lower[i] <- disp_CI[1]
  disp_table$disp_CI_upper[i] <- disp_CI[2]
  #Calculate the confidence interval for lambda
  lambda_CI <- lambda + c(-1,1) * qt(0.975, n-2) *
    sqrt(phi * lambda) / sqrt(n)
  disp_table$lambda_CI_lower[i] <- lambda_CI[1]
  disp_table$lambda_CI_upper[i] <- lambda_CI[2]
}



####Script 2.2 - analyse results and produce figures####

#Get station numbers of over-dispersed places
overdisp_statno <- disp_table$statno[disp_table$disp_CI_lower >1]
length(overdisp_statno) #132
132/263 #50.2% of data overdispersed

#Get a geographical plot of this signficant result
```

```
library(rgdal)
ukShape <- readOGR(dsn = "C:\\Users\\my_laptop\\Google_Drive
_____________________\\Flood_Folder\\shapefiles", layer = "GBR_adm0")
ukShape <- spTransform(ukShape , CRS("+init=epsg:27700"))
#Reduce catchDesc to only stations that have been cleaned
catchDesc <- catchDesc[catchDesc$Station %in% disp_table$statno,]
#Make dispersion column for plotting
catchDesc$DispCI <- "Non-significant_Dispersion"
catchDesc$DispCI[catchDesc$Station %in% disp_table$statno[
  disp_table$disp_CI_lower > 1]] <- "Dispersion_greater_than_1"
catchDesc$DispCI[catchDesc$Station %in% disp_table$statno[
  disp_table$disp_CI_upper < 1]] <- "Dispersion_less_than_1"
table(catchDesc$DispCI)

require(ggplot2)
ggplot() +
  coord_cartesian(ylim = c(5000, 1018000), xlim = c(10000, 640000))+
  ### ^^^ limits to drop Shetland
  geom_polygon(data = ukShape, aes(x = long, y = lat, group = group),
               color = "forestgreen", fill = "light_grey")  +
  ### ^^^ plot the UK
  geom_point(data = catchDesc,aes(x=Easting,y=Northing, col = DispCI)) +
  ### ^^^ add points at stations location with coloring given DispCI
  scale_color_manual(values=c("red", "blue", "black")) +
  ### ^^^ change colour palette
  theme_void() ## no backgrond

#Get station numbers of over-dispersed places
overdisp_statno <- disp_table$statno[disp_table$disp_CI_lower >1]
length(overdisp_statno) #132
132/263 #50.2% of data overdispersed

#Replicate for the lambda parameter
catchDesc$LambdaCI <- "Non-significant_average_POT"
catchDesc$LambdaCI[catchDesc$Station %in% disp_table$statno[
  disp_table$lambda_CI_lower > 6]] <- "average_POT_greater_than_6"
catchDesc$LambdaCI[catchDesc$Station %in% disp_table$statno[
  disp_table$lambda_CI_upper < 4]] <- "average_POT_less_than_4"
table(catchDesc$LambdaCI)

ggplot() +
  coord_cartesian(ylim = c(5000, 1018000), xlim = c(10000, 640000)) +
  ### ^^^ limits to drop Shetland
```

```
  geom_polygon(data = ukShape, aes(x = long, y = lat, group = group),
               color = "forestgreen", fill = "light grey") +
  ### ^^^ plot the UK
  geom_point(data = catchDesc,aes(x=Easting,y=Northing,
                                  col = LambdaCI)) +
  ### ^^^ add points at station location with coloring given LambdaCI
  scale_color_manual(values=c("red", "blue", "black")) +
  ### ^^^ change colour palette
  theme_void() ## no backgrond

#Correlation between phi and lambda
cor(disp_table$disp, disp_table$mean) #0.18



####Script 2.3 - Look at low/high POT and overdispersion####

################
##Low POT
low_POT <- disp_table[round(disp_table$lambda_CI_upper) < 3,]
#Look at the POT plots - search for abnormal results:
par(mfrow = c(3,1))
for (i in 1 : dim(low_POT)[1]){
  statno <- low_POT$statno[i]
  #The following line pre-sets temp_events so that we have all 3
  #plots stacked nicely with the #same x-axis ranging from
  #1960 to 2013
  temp_events <- data.frame(statno = statno, validWY = c(1960:2013),
                            num_events=NA)
  validWY <- events_table$validWY[events_table$statno == statno]
  temp_events$num_events[temp_events$validWY %in% validWY] <-
    events_table$num_events[events_table$statno == statno]
  #^^^add in the known POT
  plot(temp_events$validWY, temp_events$num_events, main =
         paste("Gauging Station ",statno), pch=19,
       ylim = c(0,6),xlab = "Water Year",ylab = "number of POT")

  #Add the estimated mean (lambda)
  abline(disp_table$mean[disp_table$statno == statno],0, col = "red",
         lty = 2)
}
par(mfrow = c(1,1))
#Looking at station 68010 for bizareness -
```

```
#why only 5 valid Water Years?
statno <- 68010
POTtable <-read.csv(paste0("WINFAP-FEH_v4.1/Suitable_for_QMED/",
                           statno,".PT"),header=0)
#When viewing the POTtable see that there are 28 instances of known
#data gaps, and record ends in 1981

################
##High POT
high_POT <- disp_table[round(disp_table$lambda_CI_lower) > 6,]
dim(high_POT) #4 stations we're dealing with.
#The 4 with CI greater than 6.5
#Same as for low_POT - look at the POT plots search for odd results:
par(mfrow = c(2,2))
for (i in 1 : dim(high_POT)[1]){
  statno <- high_POT$statno[i]
  #The following line pre-sets temp_events so that we have all 3 plots
  #stacked nicely with the same x-axis ranging from 1960 to 2013
  temp_events <- data.frame(statno = statno, validWY = c(1960:2013),
                            num_events=NA)
  validWY <- events_table$validWY[events_table$statno == statno]
  temp_events$num_events[temp_events$validWY %in% validWY] <-
    events_table$num_events[events_table$statno == statno]
  #^^^ add in the known POT
  plot(temp_events$validWY, temp_events$num_events, main =
         paste("Gauging_Station_",statno), pch=19,
       ylim = c(0,35),xlab = "Water_Year",ylab = "number_of_POT")
  #Add the estimated mean (lambda)
  abline(disp_table$mean[disp_table$statno == statno],0, col = "red",
         lty = 2)
}
par(mfrow = c(1,1))

################
##Overdispersion
overdisp <- disp_table[round(disp_table$disp_CI_lower) > 2,]
dim(overdisp) #12 stations we're dealing with. The 12 with disp
#CI greater than 2.5
par(mfrow = c(2,2))
for (i in 1 : dim(overdisp)[1]){
  statno <- overdisp$statno[i]
  temp_events <- events_table[events_table$statno == statno, ]
  plot(temp_events$validWY, temp_events$num_events, main =
```

```
          paste("Gauging Station ",statno),
        xlab = "Water Year",ylab = "number of POT", pch=19)
  #Add the estimated mean (lambda)
  abline(disp_table$mean[disp_table$statno == statno],0, col = "red",
         lty = 2)
}
par(mfrow = c(1,1))

#Look at station 47001 (large missing data not reported)
statno <- 47011
POTtable <-read.csv(paste0("WINFAP-FEH_v4.1/Suitable_for_QMED/",
                           statno,".PT"),header=0)
#look at all the days of POT values
dates47011 <- dmy(POTtable$V1[9:282])
dates47011 <- dates47011 - dmy(POTtable$V2[POTtable$V1 ==
                                           "Record Period"])
#see jump in days after recording from 930 to 1000. This is due to
#large missing data period, as also recorded in allAMax.

#Look at station 39003 (large change in POT dates)
statno <- 39003
POTtable <-read.csv(paste0("WINFAP-FEH_v4.1/Suitable_for_QMED/",
                           statno,".PT"),header=0)
#look at all the days of POT values
dates39003 <- data.frame(date = dmy(POTtable$V1[34:491]),
                         POT = POTtable$V2[34:491])
dates39003$date <- dates39003$date -
  dmy(POTtable$V2[POTtable$V1 == "Record Period"])
plot(dates39003, main = paste("Gauging Station",statno),
     xlab = "Days after start of Record Period",
     ylab = "Flow (m/s)", pch=19)
#Can see change roughly 10,000 days into the Record Period.



#Make the table of 43 stations with mean outside of [4,6].
#Output in LATEX format
library(xtable)
options(xtable.floating = FALSE)
options(xtable.timestamp = "")
problem_table <- disp_table[disp_table$lambda_CI_upper < 4 |
                            disp_table$lambda_CI_lower > 6,
```

```
                                    c(1:2,7:8)]
problem_table$CI <- paste0(
  "(", round(problem_table$lambda_CI_lower,2), ", ",
  round(problem_table$lambda_CI_upper,2), ")")
problem_table <- problem_table[,c(1:2,5)]
names(problem_table) <- c("Station Number", "lambda",
                          "Confidence Interval")

row.names(problem_table) <- c(1:41)
xtable(problem_table)


##Brief look into correlation between overdispersion and other
##variables such as station location and environment:

#The next 3 lines ensure we have the same stations in the
#same order
catchDesc <- catchDesc[order(catchDesc$Station),]
disp_table$statno <- as.numeric(as.character(disp_table$statno))
disp_table <- disp_table[order(disp_table$statno),]
catchDesc$disp_CI_lower <- disp_table$disp_CI_lower
nums <- c("SAAR","URBEXT2000", "BFIHOST", "disp_CI_lower")
cor.dat <- as.data.frame(cor(cbind(catchDesc$SAAR,
                                   catchDesc$URBEXT2000,
    catchDesc$BFIHOST, catchDesc$disp_CI_lower)))
colnames(cor.dat) <- c("Average Rainfall", "Urban",
                       "Permeability", "Dispersion")
rownames(cor.dat) <- colnames(cor.dat)
xtable(round(cor.dat,2)) ##output in LATEX format
```

Bibliography

[1] Stuart Coles. *An Introduction to Statistical Modeling of Extreme Values*. Springer, 2001. ISBN: 978-1-84996-874-4.

[2] Julian J. Faraway. *Extending the linear model with R : Generalized Linear, Mixed Effects and Nonparametric Regression Models*. Taylor and Francis Group, 2006. ISBN: 158488424X.

[3] *lubridate R Package*. URL: `https://cran.r-project.org/web/packages/lubridate/lubridate.pdf` (visited on 04/16/2017).

[4] P. McCullagh and J. A. Nelder. *Generalised Linear Models*. Chapman and Hall, 1983.

[5] *National River Flow Archive Homepage*. URL: `https://nrfa.ceh.ac.uk/` (visited on 04/14/2017).

[6] Q.J. Wang. "The POT model described by the generalized Pareto distribution with Poisson arrival rate". In: *Journal of Hydrology* 129 (1991), pp. 263–280. DOI: `http://www.sciencedirect.com/science/article/pii/002216949190054L4`.