Delft University of Technology
Faculty of Electrical Engineering, Mathematics and Computer Science
Delft Institute of Applied Mathematics

Outlier detection in non-Gaussian distributions

Uitschieter detectie in niet-Gauss verdelingen

Thesis submitted to the
Delft Institute of Applied Mathematics
in partial fulfillment of the requirements

for the degree of

BACHELOR OF SCIENCE
in
APPLIED MATHEMATICS

by

Y.R. Maas

Delft, The Netherlands
December 2019

BSc Thesis APPLIED MATHEMATICS

"Outlier detection in non-Gaussian distributions"

Y.R. Maas

Delft University of Technology

Supervisor

Dr.ir. J.J. Cai

Other committee members

Dr.ir. L. Meester,

Dr.ir. M. Keijzer

December, 2019 Delft

Abstract

In this thesis we are going to study outlier detection methods and propose a new method. Classical outlier detection is typically based on the assumption that the data is from a Gaussian/normal distribution. When the underlying distribution of a random sample is heavy tailed, so not normal , it is likely to have some extreme observations which would be identified as outlier using the classical procedure. This paper aims to address this issue by proposing a procedure to identify real 'outliers' for heavy tailed data set. We first dive in the some existing methods and see how they work, try to understand them, simulate them and see their shortcomings in the case of a heavy tailed distribution. Then we study Extreme Value Theory (EVT) which we shall use to set up our proposed method of detecting outliers. Once we have constructed the proposed method, we are going to simulate and compare it with the existing methods. The goal in the case of normality is that the new method is not worse than the existing ones, at least not extremely, and in the case of a heavy tailed function to work better.

Contents

1 Introduction

When we are analysing a data set, it can happen that there is an outlier in it. You want to detect which data point in your data set is really an outlier and remove it for further analysis or take a closer look at it. Questions like, why is this point in my data set? Has a mistake been made?

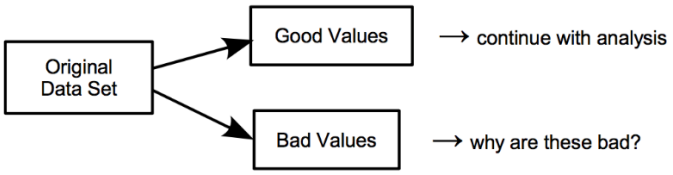


Figure 1: Steps for outlier detection. [1]

The algorithm is shown in figure 1, where we consider outliers the bad values. So there should be an method that detects when we really have an outlier or not. Most of the classical methods use the assumption that the data is from a normal distribution. This is a very strong assumption, as much data which is being analysed today does not come from a normal distribution.Then it can happen that some points are considered as outliers with the classical methods while they are in fact not and thus shouldn't be removed from the data set. We need to propose a new method that does not depend on the normality assumption and still works as good as the other classical methods if the data is indeed normally distributed or at least not much worse.

Let $X_1, \ldots, X_n$ be independently and identically distributed from an univariate distribution F. The definition which we will use for an outlier is as follows:

Definition 1.1. x^* is considered an outlier and should be removed from your data set if

$$\mathbb{P}_F(\max_{1\leq i\leq n} X_i > x^*) \tag{1}$$

is very small.

In other words "x^* is not likely to occur in a sample of size n and should therefore be deleted". Very small is too vague, so will need some threshold that says when the chance is small enough and make the definition a bit better. So definition 1.1 becomes:

Definition 1.2. x^* is considered an outlier and should be removed from your data set if

$$\mathbb{P}_F(\max_{1\leq i\leq n} X_i > x^*) \leq \alpha \tag{2}$$

for a predetermined significance level α.

For the case that we have an outlier at the left side of the data set:

$$\begin{aligned}&\mathbb{P}_F(\min_{1\leq i\leq n} X_i < x^*) \leq \alpha \\ &= \mathbb{P}_F(\max_{1\leq i\leq n} -X_i > -x^*) \leq \alpha\end{aligned}$$

So we can still use our definition 1.2 and don't have to change it when we want to look at the left side of the data set. Most common is that the significance level α has the value 10%, 5%

or 1%, but the user of the method can change that number how he pleases. From definition 1.2 we know x^*, $\max_{1\leq i\leq n} X_i$ and α, and not the distribution F, although in some cases we might know it. If we know F, then we can compute (1) directly, but in practice you most likely don't know the underlying distribution of your data set and thus have to use an outlier detection method.

We need to estimate chance (1) and will use extreme value theory (EVT) to do that. What we are going to do is finding the extreme value distribution G or at least make a good estimate. We we also study the tail probability to propose a method. But before we are looking at that, we can also compute it this way

$$\begin{aligned}
&\mathbb{P}(\max_{1\leq i\leq n} X_i > x^*) \\
&= 1 - \mathbb{P}(\max_{1\leq i\leq n} X_i < x^*) \\
&= 1 - \mathbb{P}(X_1 < x^*, X_2 < x^*, ..., X_n < x^*) \\
&= 1 - \mathbb{P}(X_1 < x^*)\mathbb{P}(X_2 < x^*) \cdots \mathbb{P}(X_n < x^*) \\
&= 1 - F^n(x^*) \\
&= 1 - [\mathbb{P}(X < x^*)]^n
\end{aligned}$$

Where we use that $X_1, \ldots, X_n$ are independently and identically distributed from distribution F and say $X \sim F$. If we know F, we can compute (1). However, in most cases you are not 100% sure what the underlying distribution is. Nevertheless we will use this to compare it to the proposed method.

2 Some existing methods

2.1 Theory

Now we are going to study some existing methods of outlier detection. There are many existing methods or criteria you can use to determine if a point x^* is an outlier or not. We will study these 6:

- Peirce's criterion;
- Chauvenet's criterion;
- Grubbs's test;
- Dixon's Q test;
- Z-score;
- MAD.

We will study these methods and compare them with the proposed method later in the paper.

Peirce's Criterion:
Like many older methods for determining outliers in a data set, Peirce's method is also derived for the Gaussian distribution, in other words a normal distribution. However it has a nice edge compared to some other criteria for removing outliers, it can be used to find and remove more than 1 outlier. These are the following steps you have to take with this method:

1. calculate the mean μ and standard deviation σ of the data set.
2. obtain R from the Peirce's criterion table and assume the case of 1 outlier (see figure 2).
3. for any weird point x^* calculate $|x^* - \mu|$.
4. remove the presumed outlier x^* if

$$\sigma R < |x^* - \mu| \tag{3}$$

5. if there is only 1 point removed from the data set, assume now the case of 2 outliers with the already calculated mean and standard deviation σ. Then go to step 8.
6. if there are more than 1 removed from the data set, assume the next highest case of outliers with the already calculated mean and standard deviation σ. Then go to step 8.
7. repeat steps 2 to 5 until no more outliers are to be found
8. obtain the new value of the mean and standard deviation of the reduced data set.

This is the table used to determine R (table contains max for $n = 50$ data points and up to 4 points who are considered abnormal):

	Number Of Suspected Outliers			
n	1	2	3	4
3	1.196			
4	1.383	1.078		
5	1.509	1.200		
6	1.610	1.299	1.099	
7	1.693	1.382	1.187	1.022
8	1.763	1.453	1.261	1.109
9	1.824	1.515	1.324	1.178
10	1.878	1.570	1.380	1.237
11	1.925	1.619	1.430	1.289
12	1.969	1.663	1.475	1.336
13	2.007	1.704	1.516	1.379
14	2.043	1.741	1.554	1.417
15	2.076	1.775	1.589	1.453
20	2.209	1.914	1.732	1.599
25	2.307	2.019	1.840	1.709
50	2.592	2.326	2.158	2.035

Figure 2: Table for R. [2]

To get the R values for n higher than 50, you can use for number of suspected outliers m:

$$R[m] = p_1[m] * log(n) + p_2[m] \tag{4}$$

where p_1 comes from the array [0.4094 0.4393 0.4565 0.4680 0.477 0.4842 0.4905 0.4973 0.5046] and p_2 from the array [0.991 0.6069 0.3725 0.2036 0.0701 -0.0401 -0.1358 -0.2242 -0.3079]. [3] So we now have a way to calculate R for all n, but we can only do it up to 9 points who are considered abnormal.

Chauvenet's Criterion:
Another criterion we can use is Chauvenet's criterion which also has the assumption of normality. This method is used in many educational institutions and laboratories to look for outliers. Although it's a good way to establish if a point is a true outlier or not, it makes an arbitrary assumption considering the rejection of the data. It also doesn't make a distinction between the case of 1 or more suspicious data points. Peirce's method doesn't make this arbitrary assumption and can easily be used in the case of several outliers.

The algorithm:

1. calculate the mean μ and standard deviation σ.
2. reject the suspicious point x^* if

$$\operatorname{erfc}\left(\frac{|x^* - \mu|}{\sigma}\right) < \frac{1}{2n} \tag{5}$$

3. repeat steps 1 and 2.
4. obtain final μ, σ and n.

In step 2 the assumption of the Chauvenet's criterion is used, if (5) is true than the suspicious point has to be removed. The erfc is the complementary error function and is $1 - \text{erf}$ where erf is the normal error function which is defined as:

$$\operatorname{erf}(x) = \frac{1}{\sqrt{\pi}} \int_{-x}^{x} e^{-t^2} \mathrm{d}t \tag{6}$$

Which we can write as $\mathbb{P}\left(Y \in [-x, x]\right)$ with $Y \sim N\left(0, \frac{1}{2}\right)$, so we see the assumption of normality back in the algorithm. We can re-write (5) to

$$n \cdot \operatorname{erfc}\left(\frac{|x^* - \mu|}{\sigma}\right) < \frac{1}{2} \tag{7}$$

From the assumption we now have the number $\frac{1}{2}$. But how can we interpret $\frac{1}{2}$? We give each point a 50% of survival. In other words, there must be as many points closer to the mean as there are further away and when it's too far away it is considered an outlier and should be removed from the database. Now we will look at a little example of the criterion used on a data set with 14 points.

	Original	*Pass #1*	*Pass #2*	
	8.02	.	.	
	8.16	.	.	
	3.97	.	.	
	8.64	.	.	
	0.84	.	.	
	4.46	.	.	
	0.81	.	.	
	7.74	.	.	
	8.78	.	.	
	9.26	.	.	
	20.46	.	**outlier**	← Shielded outlier
	29.87	**outlier**	.	
	10.38	.	.	
	25.71	**outlier**	.	
avg:	10.51	7.63	6.46	
stdev:	8.77	5.17	3.38	
n:	14	12	11	

Figure 3: An Example of Chauvenet's criterion. [1]

We see in figure 3 that in the first iteration of the method we get 2 outliers (29.87 and 25.71). However when we do the second iteration the point 20.46 is also considered an outlier by the criterion. This outlier is called a shielded outlier. An outlier which in first instance was small enough or close enough to the mean to be excluded as an outlier, but by removing the other extreme values it is revealed to be an outlier as well. The shielding effect is the reason why you must use an outlier detection method multiple times.

Grubbs's test:
The Grubbs test, which is also referred to as maximum normalized residual test or extreme studentized deviate test, is an outlier detection test to find outliers in an univariate data set. This test has again the assumption of normality. The test detects 1 outlier at a time, unlike we saw in the method of Chauvenet, and therefore the points which are tested are always the maximum or the minimum value points of the data set. When the assumed outlier is to be considered as one, we remove it and apply Grubbs's test again. The test should not be used when the data set only has 6 or less points, because then it will most likely consider them all as outliers. The test works with the 2 hypothesis:

- H_0: There are no outliers in the data set.
- H_a: There is 1 outlier in the data set.

H_0 is called the null hypothesis and H_a the alternative hypothesis. The test looks if we should reject the null hypothesis or not and the test statistic is defined as:

$$G = \frac{\max_{i=1,\dots,n} |x_i - \mu|}{\sigma} \tag{8}$$

where μ and σ are the mean and standard deviation. This is the 2-sided version of the test, but there is also a 1-sided test. For the minimum value the statistic becomes:

$$G = \frac{\mu - x_{\min}}{\sigma} \tag{9}$$

where x_{min} is the minimum value of the data points. And for the maximum value it becomes:

$$G = \frac{x_{\max} - \mu}{\sigma} \tag{10}$$

The null hypothesis is rejected with a beforehand determined significance level α for the 2-sided test if

$$G > \frac{n-1}{\sqrt{n}} \sqrt{\frac{t^2_{\frac{\alpha}{2n},n-2}}{n-2+t^2_{\frac{\alpha}{2n},n-2}}} \tag{11}$$

where $t_{\frac{\alpha}{2n},n-2}$ is the upper critical value of the t-distribution with n-2 degrees of freedom and a significance level of $\frac{\alpha}{2n}$. For 1 sided test just replace $\frac{\alpha}{2n}$ by $\frac{\alpha}{n}$.

Dixon's Q-test:
Dixon's Q-test or just Q-test is a test that assumes a normal distribution. It is said to not use this test too often and if you use it, only use it once in your data set. The first thing you need to do is to arrange your data points from lowest to highest value. Then determine the Q-statistic:

$$Q = \frac{\text{gap}}{x_{\max} - x_{\min}} \tag{12}$$

where gap is the absolute difference between the suspicious point in question and the closet point to it. If $Q > Q_0$, where Q_0 is found in the Dixon's table with the corresponding number of observations n and confidence level, then we should remove the point from our data set. The table is for n up to 25:
We now only have a way for a very small data set, up to n is 25. This is why Dixon's Q-test is not such a good method to use on a large data set, for example with 2000 observations. The Q_0 is not easily calculated like the R in Peirce's method and it is already advised to only use the test once while in a large data set you most likely have more than 1 outlier.

N	10%	5%	2%	1%	0.5%
3	0.886	0.941	0.976	0.988	0.994
4	0.679	0.765	0.846	0.889	0.926
5	0.557	0.642	0.729	0.780	0.821
6	0.482	0.560	0.644	0.698	0.740
7	0.434	0.507	0.586	0.637	0.680
8	0.479	0.554	0.631	0.683	0.725
9	0.441	0.512	0.587	0.635	0.677
10	0.409	0.477	0.551	0.597	0.639
11	0.517	0.576	0.638	0.679	0.713
12	0.490	0.546	0.605	0.642	0.675
13	0.467	0.521	0.578	0.615	0.649
14	0.492	0.546	0.602	0.641	0.674
15	0.472	0.525	0.579	0.616	0.647
16	0.454	0.507	0.559	0.595	0.624
17	0.438	0.490	0.542	0.577	0.605
18	0.424	0.475	0.527	0.561	0.589
19	0.412	0.462	0.514	0.547	0.575
20	0.401	0.450	0.502	0.535	0.562
21	0.391	0.440	0.491	0.524	0.551
22	0.382	0.430	0.481	0.514	0.541
23	0.374	0.421	0.472	0.505	0.532
24	0.367	0.413	0.464	0.497	0.524
25	0.360	0.406	0.457	0.489	0.516

Figure 4: Table for Q_0.[4]

Z-score:
The Z-score, also called standard score, z-value, normal scores or standardized variables, is another criterion you can use to detect outliers. The score is defined as follows:

$$Z = \frac{x - \mu}{\sigma} \tag{13}$$

where x is the data point, μ is the mean and σ is the standard deviation. The outlier detection method is very straightforward . Calculate all the Z-scores of the data points. Then a point is considered outlier, and therefore should be removed from the data set, if the value of its z-score is higher than 3 or lower than -3. This rule of thumb is based on the empirical rule and we see from this rule that almost all data points should be within 3 standard deviation from the mean. So this it the algorithm you use:

1. look for the maximum or minimum value x^*.
2. determine the mean μ and standard deviation σ of the other values.
3. we consider x^* an outlier if it's 3 times the standard deviation of the mean, so if $x^* < \mu - 3\sigma$ or $x^* > \mu + 3\sigma$.
4. if x^* was indeed an outlier, remove it from your data set and repeat the steps above with the remaining data.

We say that x^* is an outlier because 99.7% of the data from a normal distribution is in the interval of $[\mu - 3\sigma, \mu + 3\sigma]$

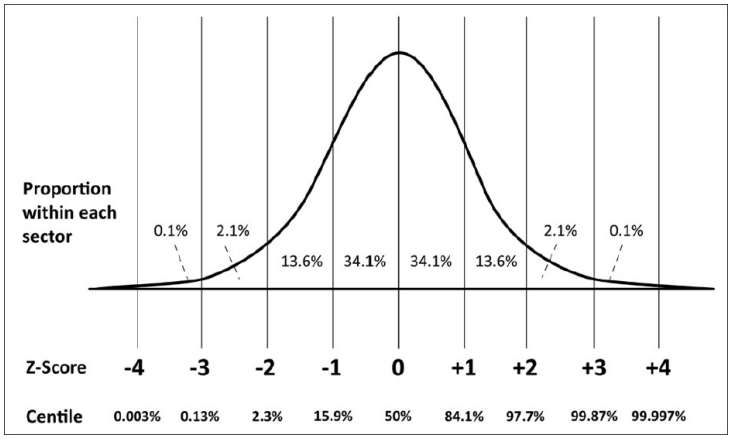


Figure 5: Z-score.[5]

MAD (Median absolute deviation):
The MAD method is one that can be used very well on the normal distribution, but it also works if the underlying distribution is not normal. It is defined as follows:

$$MAD = bM_i\left(|x_i - M_j\left(x_j\right)|\right) \tag{14}$$

where

- $b = \frac{1}{Q(0.75)}$, with Q(0.75) the 0.75 quantile of the distribution/data;
- M_i is the median of the series;
- x_i is the observation value;
- $M_j(x_j)$ is the median of the observation point x_j

In words the MAD is calculated in these steps:

1. the first thing you do is to subtract the median of all the value points in you data set and take the absolute value of it. Now we have a new series of these absolute values.
2. then rank these values from lowest to highest.
3. after that calculate the median of the that series, which is $M_i\left(|x_i - M_j\left(x_j\right)|\right)$.
4. lastly do b times the median to get the MAD.

When we have calculated the MAD, we must define a rejection criterion. Here lies the subjective nature of making a decision if a point is an outlier or not. There are 3 values, lets say C, which are most commonly used 3 (very conservative), 2.5 (moderately conservative) and 2 (poorly conservative). Then the decision criterion is defined as follows:

$$M - C \cdot MAD < x^* < M + C \cdot MAD \tag{15}$$

where M is the median of the original data set and x^* the data point. If x^* lies in the range of criterion (15), then it is kept in the data set and not considered an outlier. We use (15) to say we remove the suspicious point from the data set if:

$$\left|\frac{x^* - M}{MAD}\right| > C \tag{16}$$

2.2 Simulation Study

Now it's time to simulate these existing models. We want to test it with a data set where we know we will have an outlier for certain. So we need to calculate the equal part of (2) with a significance level α the value x^*.

$$\begin{aligned} 1 - F^n(x^*) &= \alpha \\ F^n(x^*) &= 1 - \alpha \\ F(x^*) &= \sqrt[n]{1-\alpha} \end{aligned}$$

So we get:

$$x^* = F^{-1}(\sqrt[n]{1-\alpha}) \tag{17}$$

where F^{-1} the inverse is of the cumulative distribution function. We have used $\alpha = 0.1$.

These data sets with n observations are without a point x^* or higher in it:

$N(0,1)$					
n	# of detected outliers (Peirce)	# of detected outliers (Chauvenet)	# of detected outliers (Grubb)	# of detected outliers (Z-score)	# of detected outliers (MAD)
500	0	11	0	3	1
2000	0	19	0	5	4
5000	0	30	0	13	16
Lognormal(0,1)					
500	12	16	373	8	146
2000	21	48	1873	43	535
5000	42	117	4873	98	1341
Cauchy					
500	9	3	397	1	118
2000	5	8	1475	6	419
5000	9	31	4475	18	1050

The first thing we notice is that we have 0 outliers in the normal situation without a point x^* or higher in it when the distribution is normal for Peirce. This makes sense as the method has the normality assumption. While in the situation with lognormal and Cauchy, we already have "outliers" in it according to method. But this shouldn't be the case as this is without a point x^* or higher in it. However we have outliers in the case of a normal distribution with Chauvenet. While this was not the case with Peirce. So maybe the arbitrary assumption from Chauvenet is not that good, as in the case of normality it detects outliers. As expected the number of outliers is higher in the case for lognormal, but what is strange is that the number of outliers in the case of a Cauchy distribution lower is than in the case of normality. When we have a normal distribution, Grubb's method works perfect in this case. Like Peirce, no outliers are detected when we put our point x^* in it and that is the one that gets picked out even though the μ and σ changes. However in the case of both our non-normal distributions Grubb's test detects a lot of outliers, more than half is considered an outlier. This can never be correct. Although the Z-score, like the other methods, is based on normality, it does not work as good as Peirce method and Grubb's test on a normal distribution. And is slightly better than Chauvenet in that case. But overall it is not bad. With the MAD method there are "outliers" spotted in the data set in the case of normality and also many in the case of the 2 non-normal distribution. Which was not the case for example when we used Peirce's criterion or the Z-score.

These are the values of x^* computed with (17) and an $\alpha = 0.1$:

x^*			
n	$N(0,1)$	Lognormal(0,1)	Cauchy
500	3.526	34	1511
2000	3.878	48	6042
5000	4.095	60	15105

These data sets with n observations are with a point x^* or higher in it:

$N(0,1)$					
n	# of detected outliers (Peirce)	# of detected outliers (Chauvenet)	# of detected outliers (Grubb)	# of detected outliers (Z-score)	# of detected outliers (MAD)
500	1	11	1	4	1
2000	1	19	1	6	4
5000	1	30	1	14	16
Lognormal(0,1)					
500	6	11	281	5	150
2000	14	39	1781	34	536
5000	34	102	4873	81	1341
Cauchy					
500	1	2	0	2	123
2000	2	5	793	4	426
5000	4	7	3793	7	1053

The number in Cauchy stays the low in Peirce's method, so maybe this method could be a good for a Cauchy distribution. While in the case of lognormal there are more outliers. What is weird is that in the case of lognormal and Cauchy we see that the number of outliers decrease instead of increase. In Chauvenet's case it also seems that by increasing the average μ and standard deviation σ the number of outliers decrease instead of increase which also happened in with Peirce's method with lognormal and Cauchy. We already said that Peirce's method was more general than Chauvenet's and so we could already assume that Peirce's method would be better. We really see this in the case of a normal distribution. It seems that with Grubb the number of outliers again decreases, in this case by a lot, when μ and σ change. Thus Grubb's test seems to work as good as Peirce in the normal case and better than Chauvenet's method, but is not so good on these 2 non-normal distributions. The good thing is that in the case of normality the change in the average and standard deviation doesn't increase the number of outliers by a lot. It only detects the 1 extra weird point we put in. In the case of a lognormal distribution, the Z-score method detects far more outliers and we see again a decrease in outliers when the average μ and standard deviation σ increase. What is surprising is that the Z-score works quite well in the case of a Cauchy distribution. We expected that in the normal case as Z-score is based on normality. With the MAD method it is interesting that, in the case of a normal distribution, the number of outliers doesn't change when we add the number higher than x^* in the data set. Nevertheless the values of the outliers do change. But the method is based on taking the median so adding 1 extra value will change it and thus our result.

2.3 Conclusions

We have seen that the existing methods don't work very good in the case of heavy tailed distribution like a lognormal or Cauchy. In some cases the number of outliers is very high and the method says we should remove more than half of our data points. Which can never be correct. This was to be expected as almost all are based on the assumption of normality. So we need to propose a method for outlier detection that works good in the case of data set that is not distributed normally but has is heavy tailed.

3 Proposed method

The definitions and theorems in this section follow the ones in Ferreira and de Haan. [6]

3.1 Extreme Value Theory

When we get some random data set, we don't know from which distribution it was from and thus don't know F. So to study the end of the tail, we will use Extreme Value Theory (EVT). Let $M_n = \max_{1\leq i\leq n} X_i$, we study this relation:

$$\lim_{n\to\infty} \mathbb{P}(\frac{M_n - a_n}{b_n} \leq x) = G(x) \tag{18}$$

where a_n a sequence of positive numbers and b_n a sequence of real numbers. We shall use this definition for the relation of (18):

Definition 3.1. Suppose there exists a sequence of constants $a_n > 0$ and b_n real such that for a non-degenerate distribution function G,

$$\lim_{n\to\infty} \mathbb{P}(\frac{M_n - a_n}{b_n} \leq x) = \lim_{n\to\infty} F^n\left(a_n x + b_n\right) = G(x). \tag{19}$$

G is the so called extreme value distribution. F is said in the max domain attraction of G. Notation: $F \in \mathcal{D}(G)$.

So we need to have a way to determine that G easily as taking the limit can be difficult in some cases. The theorem of Fisher, Tippet (1928) and Gnedenko (1943) can help us with that:

Theorem 3.1. The class of extreme value distribution functions is $G_\gamma\left(c_1 x + c_2\right)$ with $c_1 > 0$, c_2 real, where

$$G_\gamma\left(x\right) = \exp\left(-(1+\gamma x)^{\frac{-1}{\gamma}}\right) \tag{20}$$

with γ real and where for $\gamma = 0$ the right hand side is read as $\exp(-\exp(-x))$.

From the theorem of Fisher, Tippet and Gnedenko we know that the class of extreme value distribution functions is

$$G_\gamma(x) = e^{-(1+\gamma x)^{\frac{-1}{\gamma}}} \tag{21}$$

The γ is called the extreme value index and is a real number. When $\gamma = 0$, the right side is $G_\gamma(x) = e^{-e^{-x}}$. There are 3 situations for γ:

1) $\gamma > 0$, this means the distribution is heavy tailed at the right side, and thus the right endpoint can be infinite and the distribution does not have moments of order higher than $\frac{1}{\gamma}$. This is called Frechet domain.

2) $\gamma = 0$, this means it has a light right tail. So the right endpoint can be finite or infinite and there exists moment of any order. This is called Gumbel domain.

3) $\gamma < 0$, this means it has no tail and the right endpoint is finite. This is called reverse-Weibull domain.

We need to find a way to estimate a_n, b_n and γ . To find the estimators for a_n and b_n we will study this theorem:

Theorem 3.2. For $\gamma \in \mathbb{R}$, the following statements are equivalent:

1. $F \in \mathcal{D}(G_\gamma)$, for $\gamma \in \mathbb{R}$
2. There exists a positive function a such that for $x > 0$,

$$\lim_{t\to\infty} \frac{U(tx) - U(t)}{a(t)} = \frac{x^\gamma - 1}{\gamma}, \tag{22}$$

   where for $\gamma = 0$, the right hand side is read as $\log(x)$.

3. There exists a positive function a such that for all x with $1 + \gamma x > 0$,

$$\lim_{t\to\infty} t\left(1 - F\left(a(t)\, x + U(t)\right)\right) = (1 + \gamma x)^{\frac{-1}{\gamma}}. \tag{23}$$

   For $\gamma = 0$, the right hand side is read as $\exp(-x)$

From (23) we can see that we have to take $b_n = U(n)$. $U(t)$ is called the tail quantile function and for that we have:

$$\mathbb{P}(X > U(t)) = 1 - F(U(t)) = \frac{1}{t}, \quad t \geq 1 \tag{24}$$

From (24) we can get an expression for $U(t)$:

$$U(t) = F^{-1}\left(1 - \frac{1}{t}\right) \tag{25}$$

We already mentioned max domain of attraction, but what do we mean with that.

Example 3.1. Let $X \sim \text{EXP}(1)$, so $F(x) = 1 - e^{-x}$. From (25) we get that the tail quantile function $U(t) = \log(t), \quad t \geq 1$. Take $a(t) \equiv 1$, then we get $\frac{U(tx)-U(t)}{a(t)} = \log(tx) - \log(t) = \log(x)$. So (22) is satisfied if $\gamma = 0$. By theorem 3.2, $F \in \mathcal{D}(G_0)$. Then we get:

$$\begin{aligned} F^n(a_n x + b_n) &= \left(1 - e^{-(a_n x + b_n)}\right)^n \\ &= \left(1 - e^{-(x + \log(n))}\right)^n \\ &= \left(1 + \frac{-e^{-x}}{n}\right)^n \end{aligned}$$

Taking $n \to \infty$ gives $F^n(a_n x + b_n) \to e^{-e^{-x}}$. So the function $G(x)$ goes to $e^{-e^{-x}}$, in other words attracts to that function.

We can have that different distributions are in the same max domain of attraction. For example take $F_1 \sim$ Normal, $F_2 \sim$ Log-normal and $F_3 \sim$ Exp, we have that $F_1^n(a_n x + b_n) \to e^{-e^{-x}}$, $F_2^n(a_n x + b_n) \to e^{-e^{-x}}$ and $F_3^n(a_n x + b_n) \to e^{-e^{-x}}$. So they attract all to the same extreme value function G.

For now we will focus on the case of a heavy tailed distribution, so with $\gamma > 0$, and determine our estimators for a_n, b_n and γ. We start we approximating the quantile $U(t)$ and thus find our estimator for b_n. To do this we study this theorem:

Theorem 3.3. $F \in \mathcal{D}(G_\gamma$, for $\gamma > 0$, if and only if

$$\lim_{t\to\infty} \frac{U(tx)}{U(t)} = x^\gamma, \quad \text{for any } x > 0. \tag{26}$$

So for a distribution with a heavy tail we get that $U(tx) \approx x^\gamma U(t)$. Now we take $tx = \frac{1}{p}$ and $n = \frac{n}{k}$ where k is a large integer but much smaller than n. So we get that:

$$U\left(\frac{1}{p}\right) \approx U\left(\frac{n}{k}\right)\left(\frac{k}{np}\right)^\gamma \tag{27}$$

Now we need to estimate γ and $U\left(\frac{n}{k}\right)$. We can estimate γ with the Hill estimator and that is given by

$$\hat{\gamma}^H = \frac{1}{k}\sum_{i=1}^{k} \log\frac{X_{n-i+1,n}}{X_{n-k,n}} \tag{28}$$

Where we have ordered statistics $X_{1,n} \leq X_{2,n} \leq ... \leq X_{n-1,n} \leq X_{n,n}$ and k is a large integer but much smaller than n. To know which number for k we have to choose you can do that with visual inspection, just plot the estimates of (28) against k and see when it becomes stable. You will use the k from the moment the estimate looks to be in a stable area. For a smaller k, you have a smaller bias but a larger variance. So it is not possible to have a small bias and a small variance. There is an optimal k, but it is difficult to find that and as the estimator works good for a range of different k's there is not need to find the optimal one. So with the Hill estimator $\hat{\gamma}$ we have a way to estimate γ. $U\left(\frac{n}{k}\right)$ can be estimated with $X_{n-k,n}$. We already saw that $b_n = U(n)$, so we also have our estimator for b_n:

$$\hat{b}_n = X_{n-k,n} k^{\hat{\gamma}^H} \tag{29}$$

Now we only need a way to estimate a_n. From theorem 3.2 point 2 we have that:

$$\lim_{t\to\infty} \frac{U(tx) - U(t)}{a(t)} = \frac{x^\gamma - 1}{\gamma} \tag{30}$$

From (26) we get that $\frac{U(tx)-U(t)}{a(t)}$ is approximately $\frac{x^\gamma U(t)-U(t)}{a(t)}$. So we have get that:

$$\begin{aligned}
\frac{U(tx) - U(t)}{a(t)} &\approx \frac{x^\gamma - 1}{\gamma} \\
\Rightarrow \frac{x^\gamma U(t) - U(t)}{a(t)} &\approx \frac{x^\gamma - 1}{\gamma} \\
\Rightarrow a(t)(x^\gamma - 1) &\approx \gamma U(t)(x^\gamma - 1) \\
\Rightarrow a(t) &\approx \gamma U(t)
\end{aligned}$$

Thus our estimator of a_n is:

$$\hat{a}_n = \hat{\gamma}^H X_{n-k,n} k^{\hat{\gamma}^H} \tag{31}$$

Now we have our estimators for a_n, b_n and γ and can estimate (1) with the help of EVT. We have

$$\begin{aligned}
&\mathbb{P}(\max_{1\leq i\leq n} X_i > x^*) \\
&= \mathbb{P}(\frac{M_n - b_n}{a_n} > \frac{x^* - b_n}{a_n}) \\
&= 1 - \mathbb{P}(\frac{M_n - b_n}{a_n} \leq \frac{x^* - b_n}{a_n}) \\
&= 1 - G_\gamma(\frac{x^* - b_n}{a_n})
\end{aligned}$$

Where we use relation (19) to get the extreme value distribution G. So we need to use (20) and our estimators (28), (31) and (29) for γ, a_n and b_n to estimate

$$1 - G_\gamma(\frac{x^* - b_n}{a_n}) \tag{32}$$

So the estimator we will use for (1) will become

$$1 - e^{-(1+\hat{\gamma}\frac{x^* - \hat{b}_n}{\hat{a}_n})^{\frac{-1}{\hat{\gamma}}}} \tag{33}$$

Another way to determine (1) will be with the help of the tail probability:

$$p_0 = \mathbb{P}(X > x^*) \tag{34}$$

Because we can use that one as follows:

$$\begin{aligned}
&\mathbb{P}(\max_{1\leq i\leq n} X_i > x^*) \\
&= 1 - \mathbb{P}(\max_{1\leq i\leq n} X_i < x^*) \\
&= 1 - F^n(x^*) \\
&= 1 - [\mathbb{P}(X < x^*)]^n \\
&= 1 - [1 - \mathbb{P}(X > x^*)]^n
\end{aligned}$$

and thus get

$$\mathbb{P}(\max_{1\leq i\leq n} X_i > x^*) = 1 - (1 - p_0)^n \tag{35}$$

So we need a way to determine the tail probability p_0. We will use this theorem to estimate p_0.

Theorem 3.4. $F \in D(G_\gamma)$, for $\gamma > 0$, if and only if

$$\lim_{t\to\infty} \frac{1-F(tx)}{1-F(t)} = x^{\frac{-1}{\gamma}}$$

From theorem 3.4 we get $1-F(tx)\approx 1-F(t)x^{\frac{-1}{\gamma}}$. We now take $tx=x^*$ and $\mathrm{t}=U\left(\frac{n}{k}\right)$. Then we get

$$\begin{aligned}
&1-F(x^*)\\
&\approx 1-F\left(U\left(\frac{n}{k}\right)\right)\left(\frac{x^*}{U\left(\frac{n}{k}\right)}\right)^{\frac{-1}{\gamma}}\\
&=\frac{k}{n}\left(\frac{x^*}{U\left(\frac{n}{k}\right)}\right)^{\frac{-1}{\gamma}}
\end{aligned}$$

Now we only need to put the estimators of $U\left(\frac{n}{k}\right)$ and γ to get the estimator of p_0:

$$\hat{p}_0:=\frac{k}{n}\left(\frac{x^*}{X_{n-k,n}}\right)^{\frac{-1}{\hat{\gamma}^H}} \tag{36}$$

with $\hat{\gamma}^H$ the Hill estimator of γ (see (28)).

There are also other estimators we can use. Now we study these 2: Maximum Likelihood estimator (MLE) and Moment estimator (ME). First we study the Maximum Likelihood method. We consider the MLE of γ and $a\left(\frac{n}{k}\right)$. We denote $\sigma:=a\left(\frac{n}{k}\right)$. For $1+\frac{\gamma}{\sigma}x>0$, the density is given by:

$$h_{\sigma,\gamma}(x)=\frac{1}{\sigma}\left(1+\frac{\gamma}{\sigma}x\right)^{\frac{-1}{\gamma-1}} \tag{37}$$

The likelihood function which we have to maximize is as follows:

$$L(\sigma,\gamma)=\prod_{i=1}^{k}h_{\sigma,\gamma}\left(X_{n-i+1,n}-X_{n-k,n}\right)^{\frac{-1}{\gamma-1}} \tag{38}$$

$$=\sigma^{-k}\prod_{i=1}^{k}\left(1+\frac{\gamma}{\sigma}\left(X_{n-i+1,n}-X_{n-k,n}\right)\right)^{\frac{-1}{\gamma-1}} \tag{39}$$

We will maximize the function over the set $\{(\gamma,\sigma):\gamma>-1,\sigma>0\}$, as it will tend to ∞ for $\gamma<-1$ and $1+\frac{\gamma}{\sigma}\left(X_{n-i+1,n}-X_{n-k,n}\right)\to 0^+$. To change the product to a summation, we will look at the log-likelihood function which is defined as:

$$LL(\sigma,\gamma)=\log\left(L(\sigma,\gamma)\right) \tag{40}$$

$$=-k\log(\sigma)-\sum_{i=1}^{k}\left(\frac{1}{\gamma}+1\right)\log\left(1+\frac{\gamma}{\sigma}\left(X_{n-i+1,n}-X_{n-k,n}\right)\right) \tag{41}$$

The MLE are then obtained by solving this equation system with $Y_i=X_{n-i+1,n}-X_{n-k,n}$:

$$\begin{cases}\frac{\partial LL}{\partial\gamma}=\sum_{i=1}^{k}\left(\frac{1}{\gamma^2}\log\left(1+\frac{\gamma}{\sigma}Y_i\right)-\left(\frac{1}{\gamma}+1\right)\frac{\frac{Y_i}{\sigma}}{1+\frac{\gamma}{\sigma}Y_i}\right)=0\\ \frac{\partial LL}{\partial\sigma}=-\frac{k}{\sigma}+\sum_{i=1}^{k}\left(\frac{1}{\gamma}+1\right)\frac{\frac{Y_i}{\sigma}}{1+\frac{\gamma}{\sigma}Y_i}=0\end{cases} \tag{42}$$

As $\gamma = 0$ is not interesting, we will exclude that solution and we can therefore simplify (42) to:

$$\begin{cases} \frac{1}{k}\sum_{i=1}^{k}\log\left(1+\frac{\gamma}{\sigma}Y_i\right) = \gamma \\ \frac{1}{k}\sum_{i=1}^{k}\frac{1}{1+\frac{\gamma}{\sigma}Y_i} = \frac{1}{\gamma+1} \end{cases} \tag{43}$$

From (43) we can derive that:

$$\left(\frac{1}{k}\sum_{i=1}^{k}\log\left(1+\frac{\gamma}{\sigma}Y_i+1\right)\right)\frac{1}{k}\sum_{i=1}^{k}\frac{1}{1+\frac{\gamma}{\sigma}Y_i} = 1 \tag{44}$$

We define $f(x) := \frac{1}{k}\sum_{i=1}^{k}\log\left(1+\frac{\gamma}{\sigma}Y_i+1\right)$ and $g(x) := \frac{1}{k}\sum_{i=1}^{k}\frac{1}{1+\frac{\gamma}{\sigma}Y_i}$ and take $x = \frac{\gamma}{\sigma}$. Then we get

$$f\left(\frac{\gamma}{\sigma}\right)g\left(\frac{\gamma}{\sigma}\right) - 1 = 0 \tag{45}$$

We calculate the MLE in the following steps:

- Find root $\widetilde{x}$ for $f(x)\,g(x) - 1 = 0$;
- $\hat{\gamma}^{mle} = f(\widetilde{x}) - 1$;
- $\hat{\sigma}^{mle} = \frac{\hat{\gamma}^{mle}}{\widetilde{x}}$.

The moment estimator is the generalization of the Hill estimator. It's more general because the Hill estimator is base on the assumption that $\gamma > 0$, while the moment estimator is suitable for all γ. We have:

$$M_n = \frac{1}{k}\sum_{i=1}^{k}\left(\log\left(X_{n-i+1,n}\right) - \log\left(X_{n-k,n}\right)\right)^2 \tag{46}$$

Then the moment estimator of γ is given by:

$$\hat{\gamma}^M = \hat{\gamma}^H + 1 - \frac{1}{2}\left(1 - \frac{\left(\hat{\gamma}^H\right)^2}{M_n}\right)^{-1} \tag{47}$$

Define $\gamma_+ = \max(0,\gamma)$ and $\gamma_- = \min(0,\gamma)$. Then the Hill estimator $\hat{\gamma}^H$ estimates γ_+ and $1 - \frac{1}{2}\left(1 - \frac{\left(\hat{\gamma}^H\right)^2}{M_n}\right)^{-1}$ estimates γ_-. However if you get a random data set, you don't know beforehand what for sort distribution you have. Thus we don't know if γ should be positive or negative.

As we use have a different assumption for γ, we also need to change our method of estimating the tail probability p_0. The MLE of p_0 is as follows:

$$\hat{p}_0^{mle} = \max\left(0, \frac{k}{n}\left(1 + \hat{\gamma}^{mle}\frac{x^* - X_{n-k,n}}{\hat{a}^{mle}\left(\frac{n}{k}\right)}\right)^{-\frac{1}{\hat{\gamma}^{mle}}}\right) \tag{48}$$

with $a^{mle}\left(\frac{n}{k}\right) = \hat{\sigma}^{mle}$ and $\hat{\gamma}^{mle}$ the gamma you get from the MLE steps.

The moment estimator for p_0 equals to:

$$\hat{p}_0^M = \max\left(0, \frac{k}{n}\left(1 + \hat{\gamma}^M \frac{x^* - X_{n-k,n}}{\hat{a}^M\left(\frac{n}{k}\right)}\right)^{-\frac{1}{\hat{\gamma}^M}}\right) \tag{49}$$

with a^M defined as :

$$a^M\left(\frac{n}{k}\right) = \frac{1}{2} X_{n-k,n}\left(1 - \frac{\left(\hat{\gamma}^H\right)^2}{M_n}\right)^{-1} \tag{50}$$

where $\hat{\gamma}^H$ is the Hill estimator and M_n equal to (46).

3.2 Simulation Study

Now we simulate some distributions and want to know what happens when n becomes larger or other things that we may notice that are interesting. We will use $N(0,1)$, $U(0,1)$, EXP(3), Student's t with df = 3 and Lognormal(0,1) for this. The max-value is rounded at 2 decimal places and chance at 3. First we calculated it with $1-[\mathbb{P}(X<x^*)]^n$ as we now the distribution and can later compare that probability with what we get from our proposed method.

	$N(0,1)$	$U(0,1)$	EXP(3)	Student's t (df = 3)	Lognormal (0,1)
n	$X_{n,n}$, Prob.	$X_{n,n}$, Prob.	$X_{n,n}$, Prob.	$X_{n,n}$, Prob.	$X_{n,n}$, Prob.
500	3.04, 0.445	1, 0.104	1.91, 0.800	5.81, 0.921	41.66, 0.047
2000	3.73, 0.175	1, 0.708	3.07, 0.181	20.86, 0.214	50.89, 0.082
5000	3.93, 0.192	1, 0.697	3.47, 0.139	34.63, 0.124	46.93, 0.257
10000	3.93, 0.347	1, 0.853	3.33, 0.365	34.63, 0.233	95.61, 0.025

For normal distribution the max-value doesn't increase much as n becomes larger and the chance of an observation value larger than 3.04 is quite large, so it seems that the increase is even smaller. This is not so weird as the normal distribution has light right tail. Also in this case $n = 5000$ and $n = 10000$, we have the same max-value but the probability of that value or larger occurring is higher when $n = 10000$. So for larger n the change seems to be larger. That is not a surprise as you do $[P(X < x^*)]$ to the power n and a chance is between the values 0 and 1, the value of $[P(X < x^*)]^n$ will become smaller when n increases. x^* stays the same, thus $1-[P(X<x^*)]^n$ becomes larger.

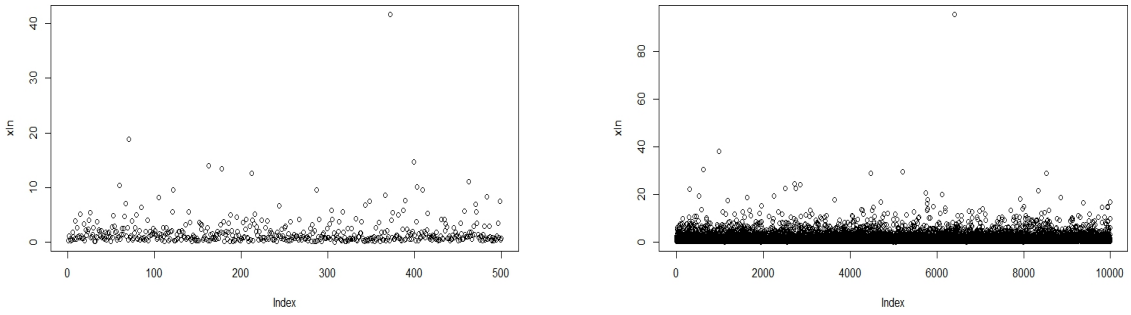
The max-value, in the case of an $U(0,1)$ case, is always 1.00 and we see that when n increases the change of that value occurring becomes larger. As the uniform distribution has a finite endpoint, here 1, the max-value cannot become higher than that endpoint value. The decrease in chance from $n = 2000$ to $n = 5000$ happens because I have rounded the max-value at 2 decimals while the max-value at 2000 observations is lower (0.99971...) than that of 5000 observations (0.99997...).

For the exponential distribution we notice that, as in the normal distribution case, the max-value doesn't increase much. Thus it seems that, like in the normal case, the exponential distribution also has a light right tail. We see again that as n grows larger the chance of some value larger or equal to the max-value increases if the max-value is almost the same.

With a student's t distribution something is happening that is different from what we saw in the other cases, the max-value increases a lot more when n becomes larger. This happens because a student's t distribution has a heavy right tail, so when n becomes larger the max-value will increase more. We see however that when n goes from 5000 to 10000 that the max-value doesn't increase, this is maybe a coincidence as I saw in other cases, using a different seed, the max-value did increase.

When we have a lognormal distribution we see, like in the student-t case, that the value increases when n becomes larger. Although it decreases from 50.89 to 46.93 when we do 3000 observations more, the probability of that value occurring or larger is lower in the case of $n = 2000$. As there is an increase in the max-value it seems that the lognormal distribution also has a heavy right tail. For this case we will look at some figures to clarify what we see in the table.

In both cases the chance of an observation like the max-value or larger was very small. When we look at figure 6, we see that is not so weird. The max-value is really far from the other points and it seems like an outlier. If we would have used $\alpha = 5\%$, the point would have been considered a real outlier and should have been removed from the data set.



(a) $n = 500$ (b) $n = 10000$

Figure 6: Points form a Lognormal(0,1)

As there is an increase in the max-value when n increases, we said that the lognormal might have a heavy right tail. That is easy to see in figure 7 when we make a QQ-plot. The points almost follow a normal distribution, but at the right tail it clearly doesn't. You can do the same for the other distributions and look at the QQ-plot to get an indication of the heaviness of the tail and instead of looking for outliers like the plots form figure 6, you can also make a box-plot.

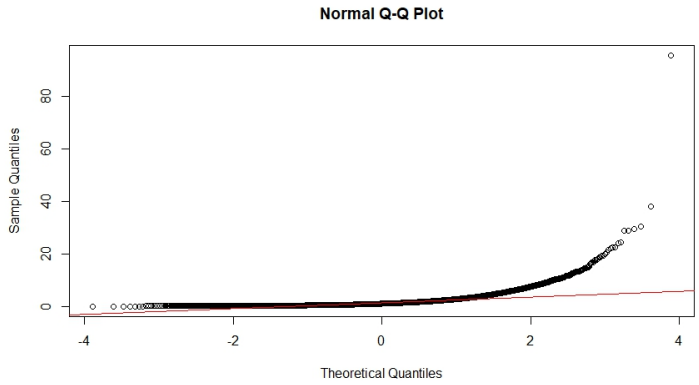


Figure 7: QQ-plot with $n = 10000$

We will use estimator (33) on Lognormal(0,1), because this is a heavy tailed distribution, and look if our estimate is close to the probability (1) we first got. The first thing we have to do before we estimate γ, a_n and b_n is to determine the right k. As this will also take some time, we will only look at $n = 500$ and $n = 2000$. After we have determined what k is suitable to use, the Hill estimator is used to estimate the value of γ. Then a_n and b_n are being estimated with (31) and (29). We will now look at the plots of the value of the Hill estimator for certain k. With these plots we can determine an usable k.

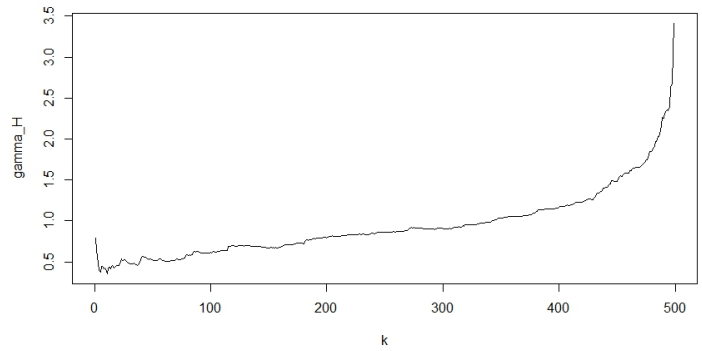


Figure 8: Hill estimator with $n = 500$

It seems that we can use $k = 50$. Then we have that $\hat{\gamma} \approx 0.523$ and $X_{450,500} \approx 3.996$. So $\hat{a}_n \approx 16.217$ and $\hat{b}_n \approx 30.979$.

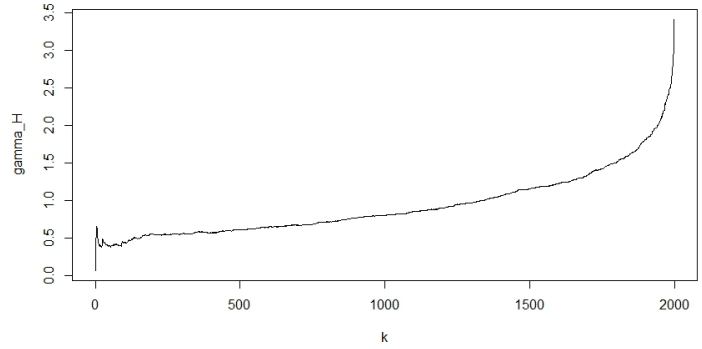


Figure 9: Hill estimator with $n = 2000$

Here we pick $k = 100$. The estimate of $\hat{\gamma} \approx 0.442$ and $X_{1900,2000} \approx 5.659$. So $\hat{a}_n \approx 19.175$ and $\hat{b}_n \approx 43.362$.

Lognormal(0,1):

n	x^*	Probability (1)	Estimate (33)
500	41.66	0.047	0.433
2000	50.89	0.082	0.502

The probability and estimate are not really close. As the γ of a lognormal distribution is zero we will simulate our method with a student's t distribution with degrees of freedom 3 as that distribution has a higher value of γ, $\gamma = \frac{1}{3}$. We will again use the estimators (28), (29) and (31) but now on the student t distribution. As the distribution has negative values, we will only look at some k and not all. This is not wrong because we know that k needs to be much smaller than n. If you would take k too large, you will also use many points that are very close to the mean and each other. While you would want to check with a select group of high value points, if the point x^* is indeed too large. And therefore should be considered an outlier and removed from the data set. After that we look if our estimate of (1) via G, so (33), is close to the probability we get via $1 - [P(X < x^*)]^n$. We first have to determine k again. With these plots we should be able to determine an usable k.

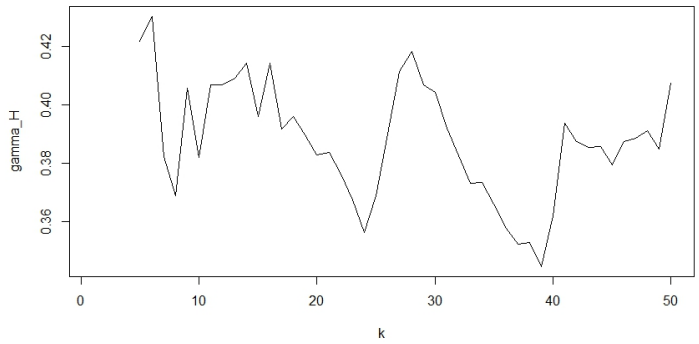


Figure 10: Hill estimator with $n = 500$

It seems that we can use $k = 38$. Then we have that $\hat{\gamma} \approx 0.353$, so close to the real value of γ and $X_{462,500} \approx 1.824$. Thus $\hat{a}_n \approx 2.319$ and $\hat{b}_n \approx 6.577$

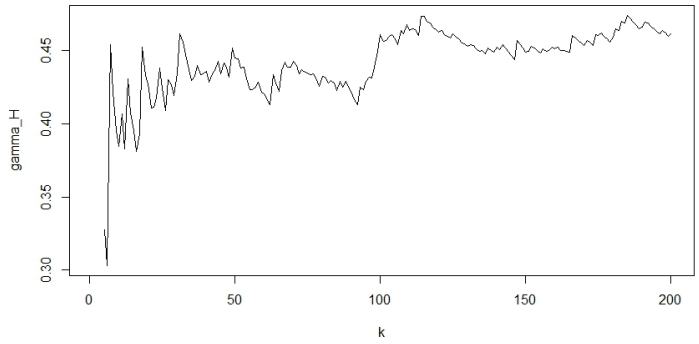


Figure 11: Hill estimator with $n = 2000$

Here we pick $k = 100$. The estimate of $\hat{\gamma} \approx 0.461$ and $X_{1900,2000} \approx 2.071$. So $\hat{a}_n \approx 7.975$ and $\hat{b}_n \approx 17.299$.

Student-t with df = 3:

n	x^*	Probability (1)	Estimate (35) with $\hat{p}_0$
500	5.81	0.921	0.307
2000	20.86	0.214	0.832

We see that for the chosen k the chances are not at all close to each other. Whereas the γ is close to the real one. We can do it in a different way via the tail probability p_0 which was defined as follows (36):

$$\hat{p}_0 := \frac{k}{n}\left(\frac{x^*}{X_{n-k,n}}\right)^{\frac{-1}{\hat{\gamma}^H}} \tag{51}$$

Same k as last time for 500, $k = 38$ and for 2000, $k = 100$. Student's t with df = 3:

n	x^*	Probability (1)	Estimate (35) with $\hat{p}_0$
500	5.81	0.921	0.283
2000	20.86	0.214	0.821

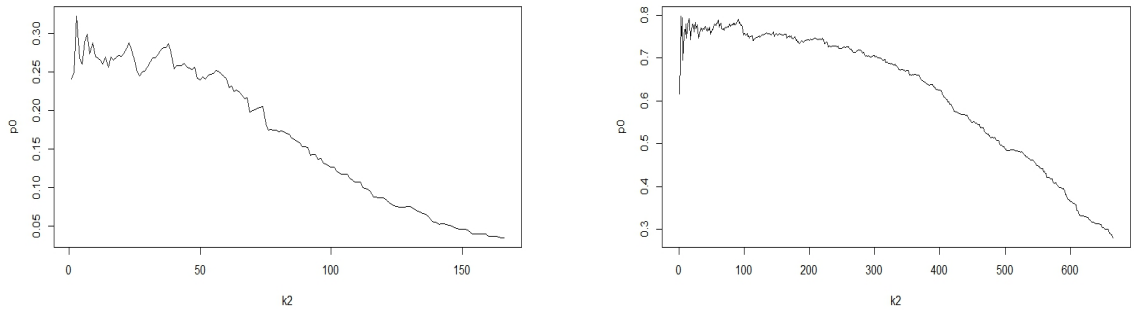
It seems it is not working as intended or not good for the distribution we used. To fix this we will use two k's instead of one. We will have a k_1 which determines the γ of the distribution and a parameter k_2 to make sure the model works as intended. So we then get:

$$\hat{p}_0 := \frac{k_2}{n}\left(\frac{x^*}{X_{n-k_2,n}}\right)^{\frac{-1}{\hat{\gamma}(k_1)^H}} \tag{52}$$

k_1 is again the same for both $n = 500$ and $n = 2000$. We have estimated for $n = 500$ that $k_2 = 6$ and $k_2 = 147$ for $n = 2000$ Student-t with df = 3:

n	x^*	Probability (1)	Estimate (35) with $\hat{p}_0$
500	5.81	0.921	0.921
2000	20.86	0.214	0.214

These k_2 are so chosen so that the (1) and (35) are equal, but we need to find a general way to estimate k_2 like we have for k_1. We shall do this in a similar way as we choose k_1.



(a) $n = 500$ (b) $n = 2000$

Figure 12: Choosing k_2

We see that p_0 will converge to 0, but we choose k_2 more in the beginning of the plot. At the moment when it seems to become stable.

Instead of using the Hill estimator for γ, we can use 2 other estimators: the maximum likelihood estimator (MLE) and the moment estimator (ME). As we also want our model to work for distribution who don't have a $\gamma > 0$, like the normal distribution, we use the more general estimator the moment estimator to determine γ. We got $\gamma \approx 0.582$ and $k_2 = 2925$ for $n = 500$ and $\gamma \approx 0.610$ and $k_2 = 9101$ for $n = 2000$

$N(0, 1)$:

n	x^*	Probability (1)	Estimate (35) with $\hat{p}_0^M$
500	3.04	0.445	0.445
2000	3.73	0.175	0.175

Now we will study the different proposed methods of estimating the probability with more different distributions:

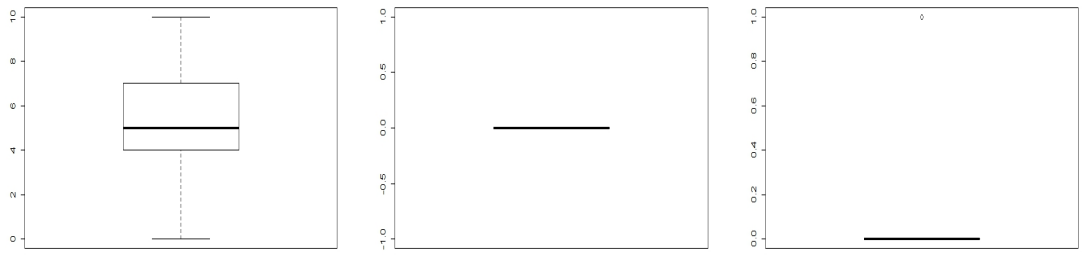
$N(0,1)$						
n	x^*	Probability (1)	Estimate (33)	Estimate (35) with $\hat{p}_0$	Estimate (35) with $\hat{p}_0^{mle}$	Estimate (35) with $\hat{p}_0^{M}$
500	3.04	0.445	0.684	0.511	1.000	0.662
2000	3.73	0.175	0.726	0.565	1.000	0.837
5000	3.93	0.192	0.800	0.676	1.000	1.000
$U(0,1)$						
500	0.998	0.630	0.665	0.496	-∞	1.000
2000	1.000	0.175	0.993	0.503	-∞	1.000
5000	1.000	0.381	1.000	0.994	-∞	1.000
EXP(3)						
500	1.98	0.733	0.882	0.763	1.000	1.000
2000	2.13	0.964	0.985	0.974	1.000	1.000
5000	3.02	0.445	0.777	0.588	1.000	1.000
Student's t (df = 3)						
500	9	0.475	0.323	0.329	0.918	0.747
2000	13	0.605	0.789	0.791	0.500	0.601
5000	21	0.453	0.648	0.624	0.477	0.468
Lognormal(0,1)						
500	21	0.445	0.616	0.527	0.030	0.060
2000	42	0.175	0.558	0.531	0.059	0.063
5000	51	0.192	0.742	0.750	0.015	0.031

Where we have used these parameter values:

$N(0,1)$					
n	Estimated γ^H	Estimated γ^{mle}	Estimated γ^M	k_1	k_2
500	0.181	0.000	-0.202	16	40
2000	0.167	0.000	-0.157	40	100
5000	0.149	0.000	-0.013	80	200
$U(0,1)$					
500	0.007	0.000	-0.449	12	30
2000	0.009	0.000	-0.803	40	80
5000	0.007	0.000	0.807	80	170
EXP(3)					
500	0.266	0.000	-0.347	25	50
2000	0.224	0.000	-0.159	60	130
5000	0.193	0.000	-0.128	80	240
Student's t (df = 3)					
500	0.356	0.462	0.376	24	60
2000	0.445	0.269	0.309	50	150
5000	0.397	0.335	0.345	150	250
Lognormal(0,1)					
500	0.410	0.397	0.256	33	75
2000	0.422	0.298	0.349	100	160
5000	0.428	0.319	0.302	210	300

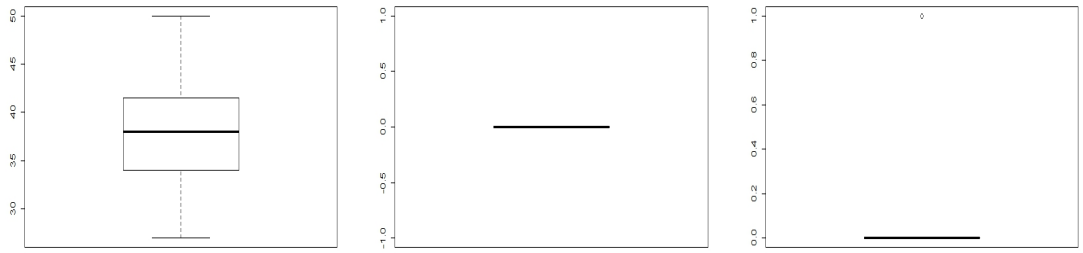
3.3 Comparison study

Now we are going to compare 1 existing method to 2 proposed models. From the existing ones we are going to use the Z-score. As that one is easily simulated and is the second best overall. For the proposed models we use the extreme value function G (33) and (35) with the tail probability p_0 estimated via $\hat{p}_0$ (36).



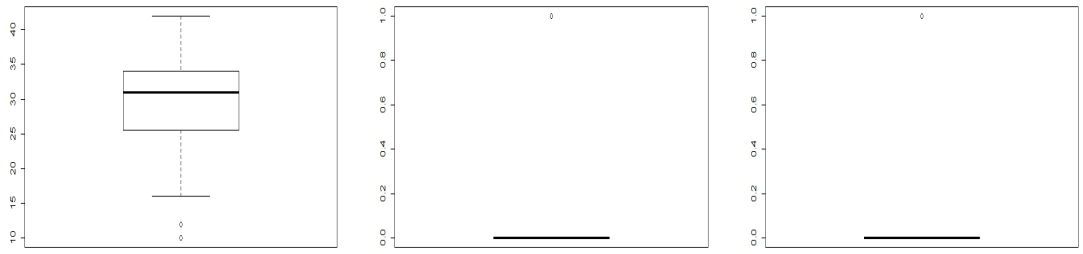
(a) Z-score, $n = 2000$ (b) (33), $n = 2000$ (c) (35) with (36), $n = 2000$

Figure 13: Number of detected outliers of 100 data sets $N(0, 1)$.



(a) Z-score, $n = 2000$ (b) (33), $n = 2000$ (c) (35) with (36), $n = 2000$

Figure 14: Number of detected outliers of 100 data sets Lognormal(0,1).



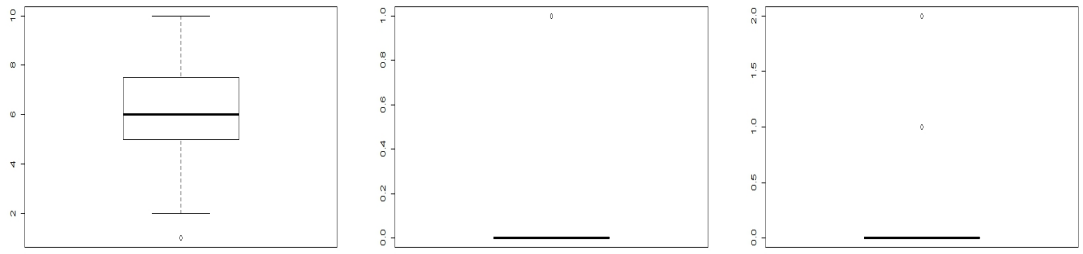
(a) Z-score, $n = 2000$ (b) (33), $n = 2000$ (c) (35) with (36), $n = 2000$

Figure 15: Number of detected outliers of 100 data sets student's t (df = 3).

In the case of a normal distribution and lognormal it seems that the method of using the extreme value function works the best. We see that in figures 13 and 14. In this data set there isn't an

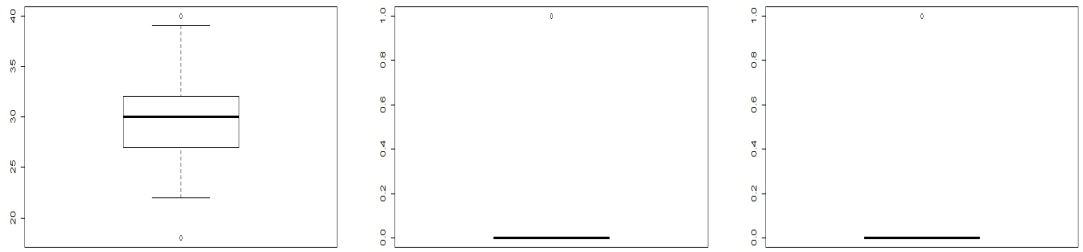
outlier and estimator (33) doesn't detect any in all the 100 different data sets. We see that our (35) sometimes detects 1 outlier, but in most cases it also detects 0. In figure 15 we see that holds for both the 2 proposed methods. In all 3 cases, we see that the proposed methods work better than the Z-score method.

Now we are going to put an outlier in the data set ourselves in the same way as we did before. So as we did in the simulation of the existing methods via (17).



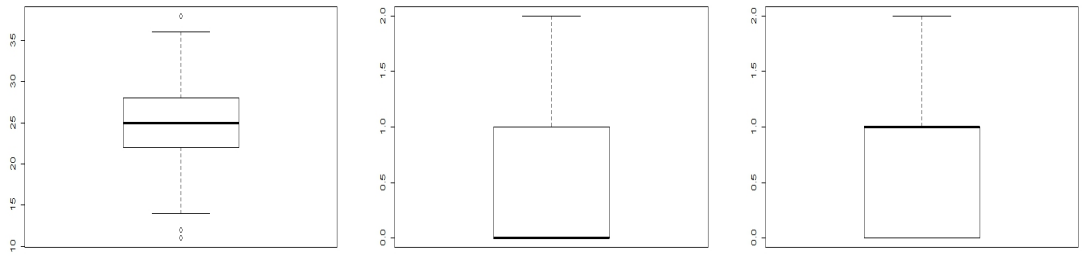
(a) Z-score, $n = 2000$ (b) (33), $n = 2000$ (c) (35) with (36), $n = 2000$

Figure 16: Number of detected outliers of 100 data sets $N(0, 1)$.



(a) Z-score, $n = 2000$ (b) (33), $n = 2000$ (c) (35) with (36), $n = 2000$

Figure 17: Number of detected outliers of 100 data sets Lognormal(0,1).



(a) Z-score, $n = 2000$ (b) (33), $n = 2000$ (c) (35) with (36), $n = 2000$

Figure 18: Number of detected outliers of 100 data sets student's t (df = 3).

We notice that, in the case of estimator (33), there are more times that it detects an outlier in the data set, but this doesn't happen very often. The same can be said about (35). Thus the proposed methods are not as consistent as we would have liked. However compared to the existing method the 2 proposed methods work well and those 2 methods detect far less points that are not-outliers. The Z-score detects a lot of "outliers" compared to the 2 proposed, while most of those points are in fact not outliers.

3.4 Conclusions

The first thing that is of notice is that estimate of (35) with (48) and (49) is not very consistent. It is sometimes very high and gives 1.000 as estimate. Or doesn't give a proper probability and gives -∞. This cannot be right as probability (1) is never so high or low. We also take $X_{n,n}$ as our x^*, not one we put in the data set ourselves, so naturally you would think there is at least some chance of a larger value then that coming in your data set. This is clearly shown by probability (1). It is true that in the case of distribution with a light tail, for example a $N(0,1)$ or $U(0,1)$, this change would be high. Secondly, estimates (33) and (35) with $\hat{p}_0$ are not very close to probability (1), which would be good, but they are close to each other. So it seems something is a little of with (33) and (35) with $\hat{p}_0$, as most of the time the probability is higher than the actual probability. But if we look at the comparison study those 2 methods are not so bad after all. Lastly, the estimations of the γ are good. It is pretty close to the real value of γ and thus the choice of k_1 seems correct. So to get the probability closer to the real one maybe a different choice of k_2 should be made or the model has to be changed a little.

4 Conclusions and discussion

The goal was a method that would be better than the existing ones we studied. Based on the results of the comparison study this has been achieved. The problem however is that it is not consistent enough. It sometimes detects outliers while there are not and sometimes it doesn't detect them while they are. From the estimations of γ with the Hill estimator, we know that the choice of k_1 is not the problem, as the values are close to the real value of γ. Thus it could be that k_2 is not correctly chosen or that there is something wrong with the method. As we hadn't enough time to test it more, we cannot say for certain that the 2 proposed method in the comparison study work all the time. We only tested it on a few distributions, one sample size and only compared it to 1 existing method. For example Peirce's and Grubb's method would most likely be better in the case of normality. However, the 2 proposed methods look to work decent in the case of a heavy tailed distribution. The main problem is that the methods are not consistent enough. With more simulations and changes in the method, this might be solved.

A R-code

```
#Peirce normal distribution
set.seed(349)

n = 500;
x = rnorm(n,0,1);
mu = mean(x);
sigma = sd(x);
alpha = 0.1;
xc = qnorm(nthroot(1-alpha,n))

a = c(0.4094, 0.4393, 0.4565, 0.4680, 0.477, 0.4842, 0.4905,
   0.4973, 0.5046);
b = c(0.991, 0.6069, 0.3725, 0.2036, 0.0701, -0.0401, -0.1358,
   -0.2242, -0.3079);
R_values = a*log(n)+b

mad1 = sigma*R_values[1]

v1 <- vector()
for (i in 1:n){
  v1[i] <- abs(x[i]-mu)
}

v2 <- vector()
for (i in 1:n){
  if (mad1 < v1[i]){
    v2[i] <- x[i]
  }
}
outliers = na.omit(v2)

#then do it again with assuming i outliers.Then madi = sigma*R-
   values[i], i is based on how many outliers there were
#detected with the assumption that there was 1 outlier. i can be
   max 9
```

```
#Grubb normal distribution
set.seed(212)

n = 500;
x = rnorm(n,0,1);
alpha = 0.1;
xc = qnorm(nthroot(1-alpha,n))
#x[n+1] <- xc*1.1
st = n;

for (t in 1:n){
  mu = mean(x);
  sigma = sd(x)
  v1 <- vector()
  for (i in 1:st){
    v1[i] = abs(x[i]-mu)
  }
  abnormalpoint = max(v1)+mu;

  G = max(v1)/sigma;
  alpha = 0.1;
  p = 1-alpha/(2*n);
  df = n-2;
  t = qt(p,df)

  G0 = ((n-1)/sqrt(n))*sqrt(t^2/(n-2+t^2))

  if (G > G0){
    outlier = abnormalpoint;
    setdiff(x,outlier);
    st = st - 1
  }
}
```

```r
#Chauvenet normal distribution
set.seed(501)

n = 500;
x = rnorm(n,0,1);
mu = mean(x);
sigma = sd(x)

v1 <- vector()
for (i in 1:n){
  v1[i] = (abs(x[i]-mu))/sigma
}

v2 <- vector()
for (i in 1:n){
  if (n*erfc(v1[i])< 0.5){
    v2[i] <- x[i]
  }
}

outliers = na.omit(v2)
```

```r
#Z-score normal distribution
set.seed(66)

n = 500;
x = rnorm(n,0,1);
mu = mean(x);
sigma = sd(x);

Zscores <- vector()
for (i in 1:n){
  Zscores[i] <- (x[i]-mu)/sigma
}

v <- vector()
for (i in 1:n){
if (abs(Zscores[i]) > 3){
    v[i] <- x[i]
    }
}
outliers = na.omit(v)
```

```r
#MAD normal distribution
set.seed(666)

n = 500;
x = rnorm(n,0,1)
xs = sort(x)
b = 1/(quantile(xs, 0.75))
M = median(xs)

ab <- vector()
for (i in 1:n){
  ab[i] <- abs(xs[i]-M)
}

abs = sort(ab)

Mi = median(abs)
MAD = b*Mi
C = 2.5

v <- vector()
for (i in 1:n){
  if (abs((xs[i]-M)/MAD) > C){
    v[i] <- xs[i]
  }
}

outliers = na.omit(v)
```

```r
set.seed(60)

#Student t
n = 5000
xt = rt(n,3)
xts = sort(xt)
xtm = max(xt)
Pt = 1 - (pt(xtm,3))^n
plot(xt)
boxplot(xt)
qqnorm(xt);qqline(xt, col='red')

gammah <- function(k){
  sum = 0
  for (i in 1:k){
    sum = sum + log(xts[n-i+1]/xts[n-k])
  }
  gammah <- (1/k)*sum
  return(gammah)
```

```
}

gammaH <- vector()
for (i in 5:(n/10)){
  gammaH[i]<- gammah(i)
}
plot(gammaH, typ = "l", xlab = "k", ylab = "gamma_H")

k = 150
gamma = gammaH[k]
X = xts[n-k]
a = gamma*xts[n-k]*(k^gamma)
b = xts[n-k]*(k^gamma)
chanceEVT = 1-exp(-(1+gamma*(xtm-b)/a)^(-1/gamma))
```

```
set.seed(60)

#Student t
n = 2000
xt = rt(n,3)
xts = sort(xt)
xtm = max(xt)
Pt = 1 - (pt(xtm,3))^n

gammah <- function(k){
  sum = 0
  for (i in 1:k){
    sum = sum + log(xts[n-i+1]/xts[n-k])
  }
  gammah <- (1/k)*sum
  return(gammah)
}

gammaH <- vector()
for (i in 5:n/10){
  gammaH[i]<- gammah(i)
}
plot(gammaH, typ = "l", xlab = "k", ylab = "gamma_H")

k1 = 150
gamma = gammaH[k1]

k2 <- function(y){
  1-(1-(y/n)*(xtm/xts[n-y])^(-1/gamma))^n
}

K2 <- vector()
for (i in 5:n/3){
  K2[i]<- k2(i)
}
```

```
plot(K2, typ = "l", xlab = "k2", ylab = "p0")

k_2 = 250

chanceH = 1-(1-(k_2/n)*(xtm/xts[n-k_2])^(-1/gamma))^n
```

```
set.seed(60)

#Student t
n = 2000
xt = rt(n,3)
xts = sort(xt)
xtm = max(xt)
Pt = 1 - (pt(xtm,3))^n

gammah <- function(k){
  sum = 0
  for (i in 1:k){
    sum = sum + log(xts[n-i+1]/xts[n-k])
  }
  gammah <- (1/k)*sum
  return(gammah)
}

gammaH <- vector()
for (i in 5:(n/5)){
  gammaH[i]<- gammah(i)
}
plot(gammaH, typ = "l", xlab = "k1", ylab = "gamma_H")

k1 = 50

f <- function(x){
  sum = 0
  for (i in 1:k1){
    sum = sum + log(1+x*(xts[n-i+1]-xts[n-k1]))
  }
  f <- ((1/k1)*sum +1)
  return(f)
}

g <- function(x){
  sum = 0
  for (i in 1:k1){
    sum = sum + 1/(1+x*(xts[n-i+1]-xts[n-k1]))
  }
  g <- (1/k1)*sum
  return(g)
}

```

```
mle <- function(x){
  mle <- f(x)*g(x) - 1
  return(mle)
}

#step 1
rx = uniroot(mle, lower = 0.1, upper = 1000000000000000000)

#step 2
gammamle = f(rx$root) - 1

#step 3
sigmamle = gammamle/rx$root

k2 = 150
chancetail = max(0, (k2/n)*(1+gammamle*(xtm-xts[n-k2]/sigmamle))
   ^(-1/gammamle))
chanceMLE = 1-(1-chancetail)^n
```

```
set.seed(60)

#student
n = 500
xt = rt(n,3)
xts = sort(xt)
xtm = max(xt)
Pt = 1 - (pt(xtm,3))^n
plot(xt)
boxplot(xt)
qqnorm(xt);qqline(xt, col='red')

gammah <- function(k){
  sum = 0
  for (i in 1:k){
    sum = sum + log(xts[n-i+1]/xts[n-k])
  }
  gammah <- (1/k)*sum
  return(gammah)
}

gammaH <- vector()
for (i in 5:(n/10)){
  gammaH[i]<- gammah(i)
}
plot(gammaH, typ = "l", xlab = "k1", ylab = "gamma_H")

k1 = 24
gamma = gammaH[k1]
X = xts[n-k1]

```

```
S = 0
for (i in 1:k1){
  S = S + (log(xts[n-i+1])-log(xts[n-k1]))^2
}

M = S/k1

gammaM = gamma + 1 -(1/2)*(1-(gamma^2/M))^(-1)

k2 = 60
aM = 0.5*xts[n-k2]*(1-(gamma^2/M))^(-1)
chancetail = max(0, (k2/n)*(1+gammaM*(xtm-xts[n-k2]/aM))^(-1/
   gammaM))
chanceME = 1 - (1-chancetail)^n
```

```
totalnumberoutliers <- vector()

for (m in 1:100){
  set.seed(m)

  n = 2000;
  x = rt(n,3);
  alpha = 0.1;
  xc = qt(nthroot(1-alpha,n),3)
  x[n+1] <- xc*1.1
  mu = mean(x);
  sigma = sd(x);

  Zscores <- vector()
  for (i in 1:n+1){
    Zscores[i] <- (x[i]-mu)/sigma
  }

  v <- vector()
  for (i in 1:n+1){
    if (abs(Zscores[i]) > 3){
      v[i] <- x[i]
    }
  }
  outliers = na.omit(v)
  numberoutliers = length(outliers)
  totalnumberoutliers[m] <- numberoutliers
}

totalnumberoutliers
boxplot(totalnumberoutliers)
```

```
totalnumberoutliers <- vector()

for (m in 1:100){
  set.seed(m)

  n = 2000
  xt = rt(n,3)
  xts = sort(xt)
  xtm = max(xt)
  alpha = 0.1;
  xc = qt(nthroot(1-alpha,n),3)
  xts[n+1] <- xc*1.1


  gammah <- function(k){
    sum = 0
    for (i in 1:k){
      sum = sum + log(xts[n-i+1]/xts[n-k])
    }
    gammah <- (1/k)*sum
    return(gammah)
  }

  gammaH <- vector()
  for (i in 5:(n/10)){
    gammaH[i]<- gammah(i)
  }

  k = 50
  gamma = gammaH[k]
  X = xts[n-k]
  a = gamma*xts[n-k]*(k^gamma)
  b = xts[n-k]*(k^gamma)


  chance <- vector()
  for (i in 1:n+1){
    chance[i] = 1-exp(-(1+gamma*(xts[i]-b)/a)^(-1/gamma))
  }

  chance2 = na.omit(chance)

  alpha = 0.1
  numberoutliers = 0

  for (i in 1:length(chance2)){
    if (chance2[i] < alpha){
      numberoutliers = numberoutliers + 1
    }
  }
```

```
51
52    totalnumberoutliers[m] <- numberoutliers
53  }
54
55  totalnumberoutliers
56  boxplot(totalnumberoutliers)
```

```
1   totalnumberoutliers <- vector()
2
3   for (m in 1:100){
4     set.seed(m)
5
6     n = 2000
7     xt = rt(n,3)
8     xts = sort(xt)
9     xtm = max(xt)
10    alpha = 0.1;
11    xc = qt(nthroot(1-alpha,n),3)
12    xts[n+1] <- xc*1.1
13
14    gammah <- function(k){
15      sum = 0
16      for (i in 1:k){
17        sum = sum + log(xts[n-i+1]/xts[n-k])
18      }
19      gammah <- (1/k)*sum
20      return(gammah)
21    }
22
23    gammaH <- vector()
24    for (i in 5:n/10){
25      gammaH[i]<- gammah(i)
26    }
27
28    k1 = 50
29    gamma = gammaH[k1]
30
31    k2 <- function(y){
32      1-(1-(y/n)*(xtm/xts[n-y])^(-1/gamma))^n
33    }
34
35    K2 <- vector()
36    for (i in 5:n/3){
37      K2[i]<- k2(i)
38    }
39
40    k_2 = 150
41
42    chance <- vector()
43    for (i in 1:n+1){
```

```
    chance[i] = 1-(1-(k_2/n)*(xts[i]/xts[n-k_2])^(-1/gamma))^n
  }

  chance2 = na.omit(chance)


  chance3 <- vector()

  for (i in 1:length(chance2)){
    if (chance2[i] < 1 & chance2[i]>0){
      chance3[i] <- chance2[i]
    }
  }

  chance4 = na.omit(chance3)
  alpha = 0.1
  numberoutliers = 0

  for (i in 1:length(chance4)){
    if (chance4[i] < alpha){
      numberoutliers = numberoutliers + 1
    }
  }

  totalnumberoutliers[m] <- numberoutliers
}

totalnumberoutliers
boxplot(totalnumberoutliers)
```

References

[1] A. F. Rochim. *Chauvenet's Criterion, Peirce's Criterion, and Thompson's Criterion (Literatures Review)*. University of Indonesia, Depok, 2016

[2] B. M. Tissue. *Basics of Analytical Chemistry and Chemical Equilibria*, John Wiley & Sons, Hoboken , 2013

[3] K. Porter, R. Hamburger and R. Kennedy. *Practical development and application of fragility functions*, American Society of Civil Engineers, Long Beach, 2007

[4] Sediment.uni-goettingen.de. (2019). Out. [online] Available at: http://www.sediment.uni-goettingen.de/staff/dunkl/software/o_l-help.html [Accessed 23 Sep. 2019].

[5] Tubbing, L. (2019). Z-score berekenen met de z-toets. [online] Available at: https://deafstudeerconsultant.nl/statistiek-met-spss/data-analyse/z-score-berekenen-met-de-z-toets/ [Accessed 22 Sep. 2019].

[6] A.F Ferreira and L. de Haan. *Extreme Value Theory: An Introduction*, Springer, New York, 2006