IDS — Assignment 1

Albin Lindqvist (13986236), Zeynep Mersinlioğlu (13616145)

2022-06-17

Question 1

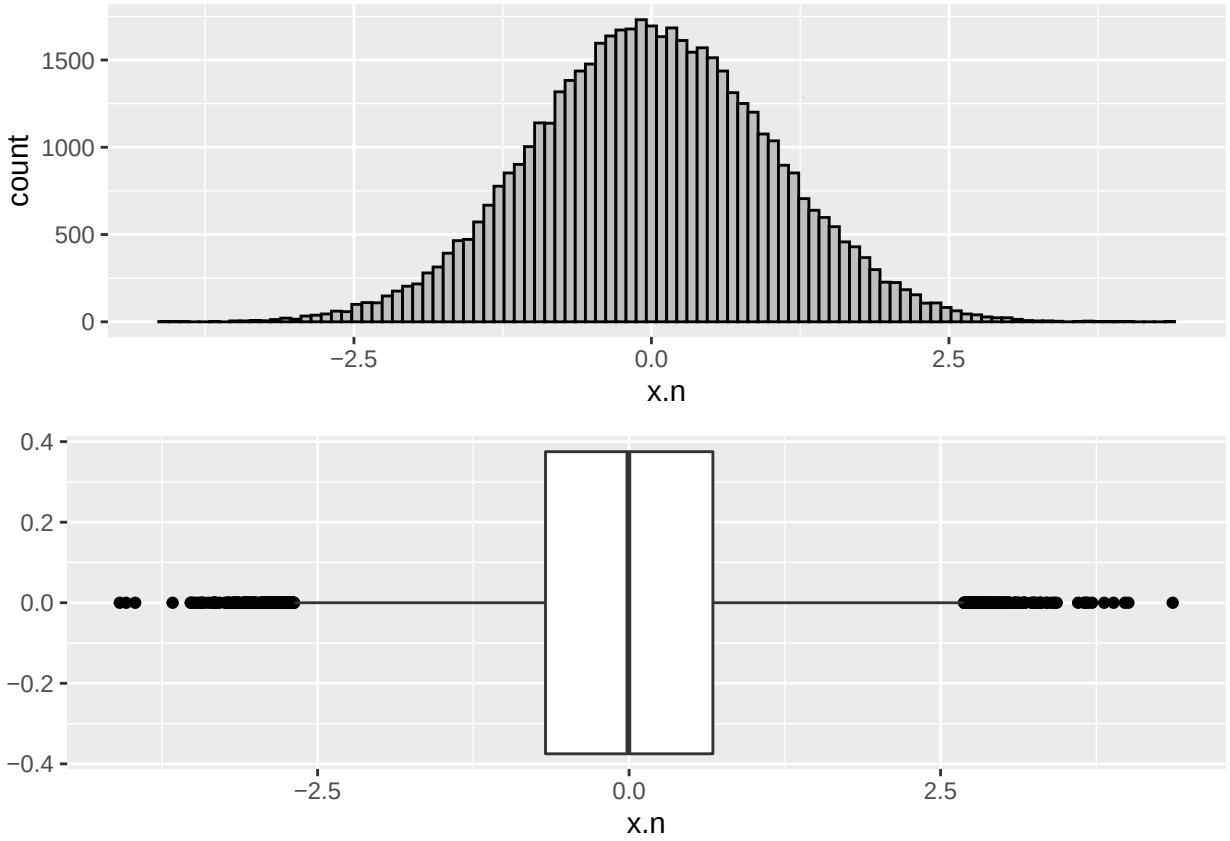
Setting up, generating, and saving the data.

```
library(Pareto)
set.seed(100)
Data=data.frame(x.n=rnorm(50000),x.p=rPareto(50000,t=1,alpha=2))
```

1.1

Creating the plot, using ggplot.

```
library(ggplot2)
library(gridExtra)
x_n_histo <- ggplot(Data, aes(x=x.n)) + geom_histogram(bins=100, col="black",
↪  fill="grey")
x_n_box <- ggplot(Data, aes(x=x.n)) + geom_boxplot( outlier.color = "black")
grid.arrange(x_n_histo, x_n_box)
```



1.2

```
(x_n_mean = mean(Data$x.n))
```

```
## [1] -0.0002084956
```

The mean is approximately zero, which is what we would expect from samples of a standard normal distributed variable.

```
(x_n_sd = sd(Data$x.n))
```

```
## [1] 0.9989658
```

The standard deviation is also very close to what we would expect from samples from standard normal distribution.

1.3

With the mean we can tell what the average number is. In other words, we can tell in what region the variable should be. So, if we take a sample from the same population, it is most likely we will find the mean, or something that is close to it.

The standard deviation on the other hand is useful to tell how concentrated the variable is, or another way to describe it: it is a measure of the spread of the variable. If we have a high standard deviation we would expect our variable to include a lot of values, and vice versa.

Thus, with this information we can predict, approximately, what the new variable could be. We can see in 1.1 that the samples are symmetric around the mean, which is a reason the mean is good for predicting the

new variable. As a matter of fact, we know that the interval $[\mu - 2\sigma, \mu + 2\sigma]$ should contain around 95% of the samples.

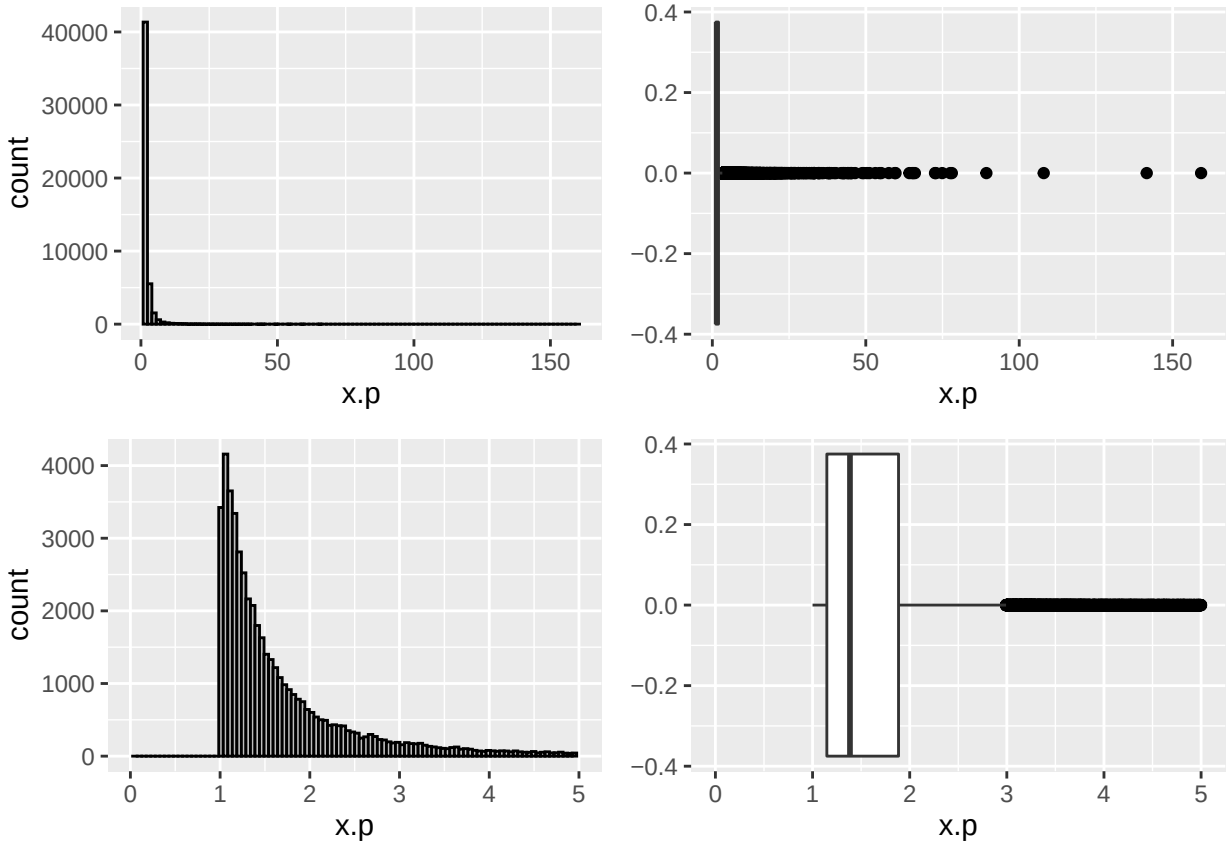
1.4

```
x_p_histo <- ggplot(Data, aes(x=x.p)) + geom_histogram(bins=100, col="black",
↪  fill="grey")
x_p_box <- ggplot(Data, aes(x=x.p)) + geom_boxplot( outlier.color = "black")
grid.arrange(x_p_histo, x_p_box, x_p_histo + xlim(0,5), x_p_box + xlim(0,5))
```

```
## Warning: Removed 1988 rows containing non-finite values (stat_bin).
```

```
## Warning: Removed 2 rows containing missing values (geom_bar).
```

```
## Warning: Removed 1988 rows containing non-finite values (stat_boxplot).
```




```
cat("Mean of x.p: ", mean(Data$x.p), " and standard deviation of x.p: ", sd(Data$x.p), "
↪  the median of x.p: ", median(Data$x.p), ".\n", sep = "")
```

```
## Mean of x.p: 1.993904 and standard deviation of x.p: 2.601173 the median of x.p: 1.41199.
```

```
summary(Data$x.p)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   1.154   1.412   1.994   1.992 159.275
```

We can clearly see that there are some very extreme values of `x.p`, and we need to zoom in a lot to see the box-plot and the histograms properties. We can see that a lot of the values are below 5, but still quite a lot

of values are above it. The standard deviation is around 2.6 but there are values which are more than 20 times the standard deviation plus the mean, so they are inadequate to describe the data accurately. We can also note that the distribution is not symmetric, there are, for example, no values that are below 1, which is logical because the support of `x.p` is $[1, \infty)$.

If we look at the range of the sample of `x.p` we will see that it is $159.275 - 1 = 158.275$, whilst the IQR is $1.992 - 1.154 = 0.838$ the difference is staggering, we can thus see that the mean and the standard deviation does not show the characteristics of the distribution.

Question 2

2.1

```
Data2 = read.csv("DataAssignment1.txt", sep="\n", header = FALSE)
Data2_log = log(Data2)
```

```
## Warning in FUN(X[[i]], ...): NaNs produced
```

As we can see we have some entries that are not a number ('NaN'), this means that our observations contains values of $x < 0$, which of course is not defined. So, we will use the following code to find which indexes that are not correct, and also remove them.

```
(idx_na_data2_log = which(is.na(Data2_log)))
```

```
## [1] 416
```

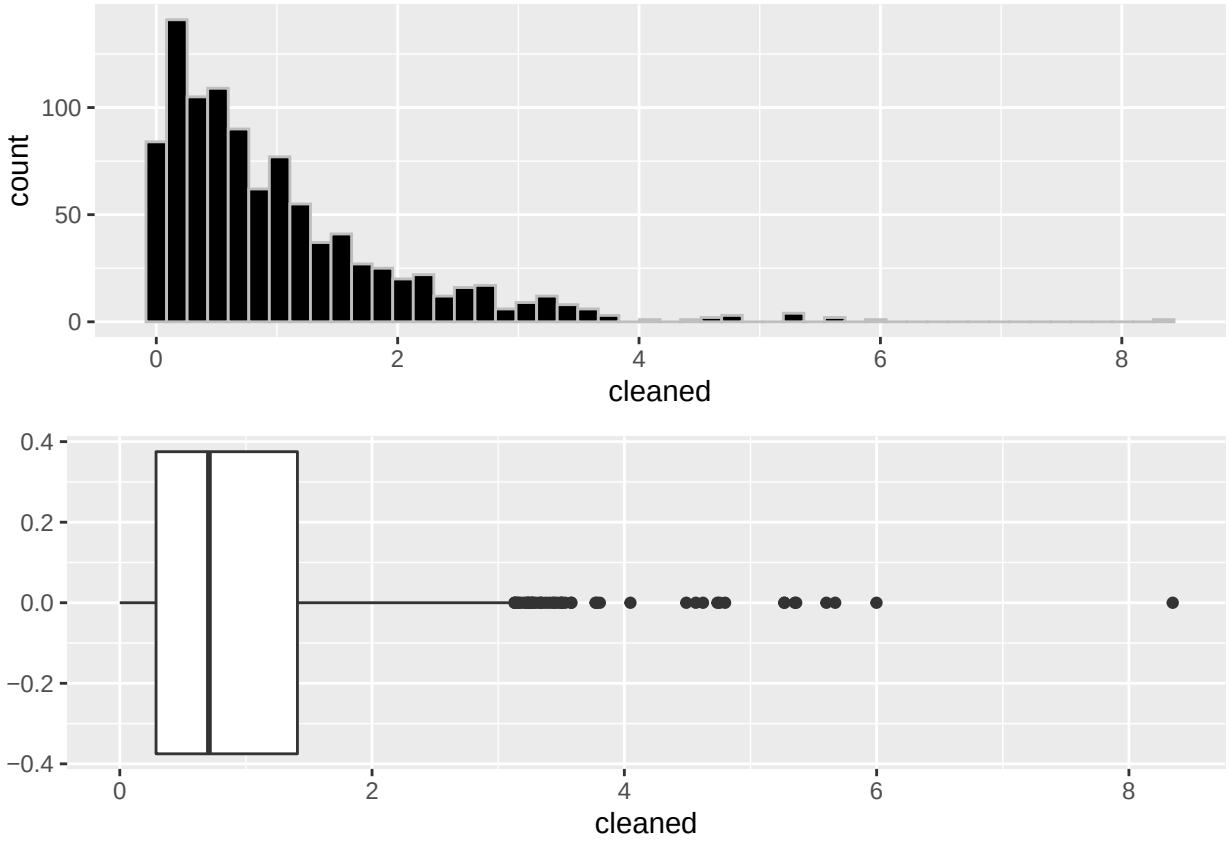
```
(Data2$V1[416])
```

```
## [1] -4.368772
```

```
cleaned = Data2_log$V1[-idx_na_data2_log]
Data2_log_clean <- data.frame(cleaned)
```

We found out that it was only one value that became 'NaN', and it is indeed negative. Since it is only one value that are below 0, we can relatively safely remove it without altering the data too much. We can also confirm that it is indeed gone. We will now plot the log transformed data

```
log_histo <- ggplot(Data2_log_clean, aes(x=cleaned)) + geom_histogram(bins=50,
↪  fill="black", col="grey")
log_box <- ggplot(Data2_log_clean, aes(x=cleaned)) + geom_boxplot()
grid.arrange(log_histo, log_box)
```



2.2

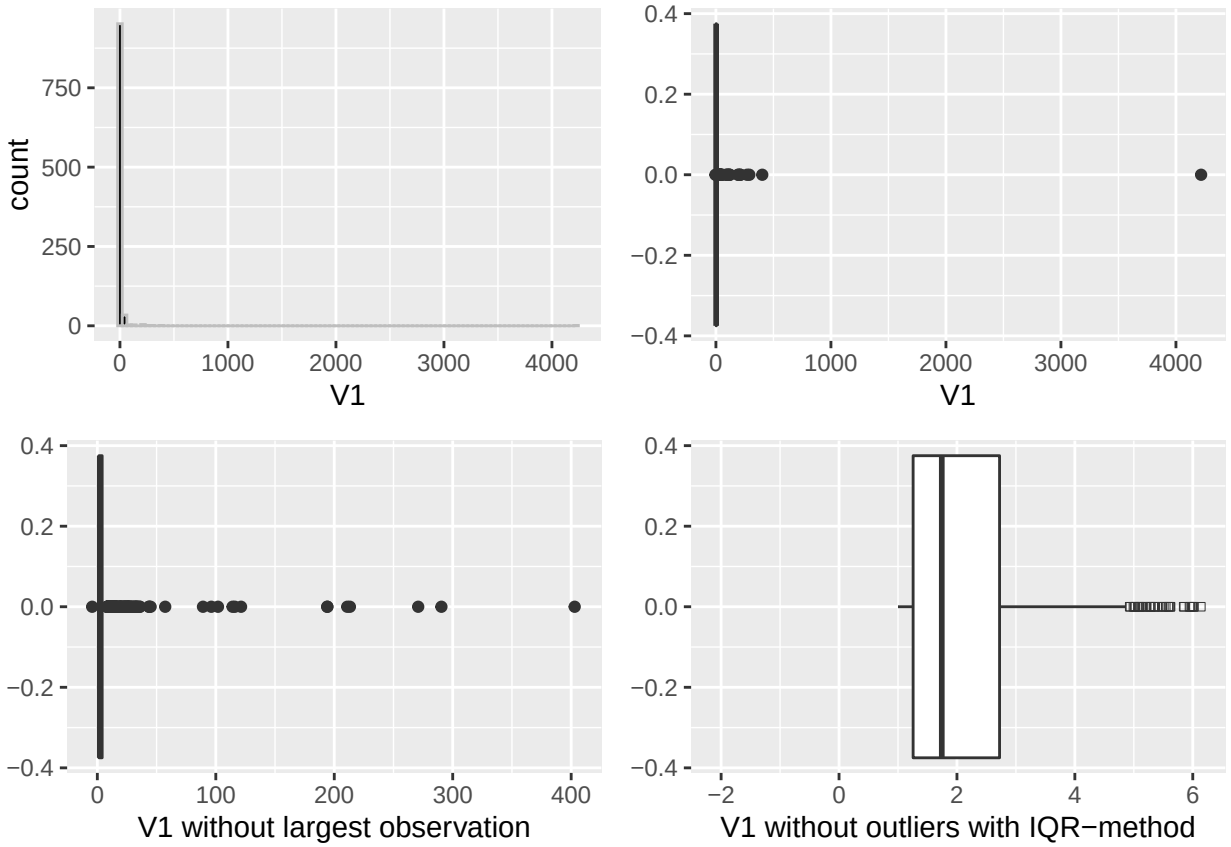
We will for the questions that follow use the original dataset. We start by plotting the data to see how it is distributed before any transformations.

```
data2_histo <- ggplot(data = Data2, aes(x=V1)) + geom_histogram(bins=100, fill="black",
↪  col="grey")
data2_box <- ggplot(data=Data2, aes(x=V1)) + geom_boxplot()
data2_box_iqr <- ggplot(data=Data2, aes(x=V1)) + geom_boxplot(outlier.shape = 0,
↪  outlier.stroke = 0) + xlim(median(Data2$V1)-1.5*IQR(Data2$V1),
↪  median(Data2$V1)+1.5*IQR(Data2$V1))+xlab("V1 without outliers with IQR-method")

grid.arrange(data2_histo, data2_box, data2_box+xlim(-5,405)+xlab("V1 without largest
↪  observation"),data2_box_iqr )
```

```
## Warning: Removed 1 rows containing non-finite values (stat_boxplot).
```

```
## Warning: Removed 168 rows containing non-finite values (stat_boxplot).
```



It is very obvious we have an outlier, that is very much above any of the others values. We can also note 167 other outliers, according to the IQR-method. However, these outliers are not as significant as the one above 4000, so we will thus discard that extreme sample as it does not fit the rest of the sample. It is nevertheless worth a short discussion on why it would be.

```
(mean(Data2$V1))
```

```
## [1] 10.76318
```

```
(mean(subset(Data2, V1<1000)$V1))
```

```
## [1] 6.551018
```

The data is from an insurance company, hence is it is not too unlikely that it was a day they received an extreme claim. Yet, we have to consider the impact this observation have. We can see that if we exclude this one extreme event our average becomes almost half of what it is if we include it, so it is not unreasonable to delete this outlier. We can also not completely rule out if it's a misplaced decimal.

We also have a negative value, which we found in 2.1. It is reasonable to delete this observation because it does not make sense to have a negative claim for an insurance company.

```
# Creating the cleaned dataset
Data2_clean = subset(Data2, V1 >= 0 & V1 < 1000)
```

We will also investigate how the z-score-method would perform. Given that the data contains a few observations that are not unreasonable, but much larger than others, it might make more sense to use the z-score instead of the IQR-method.

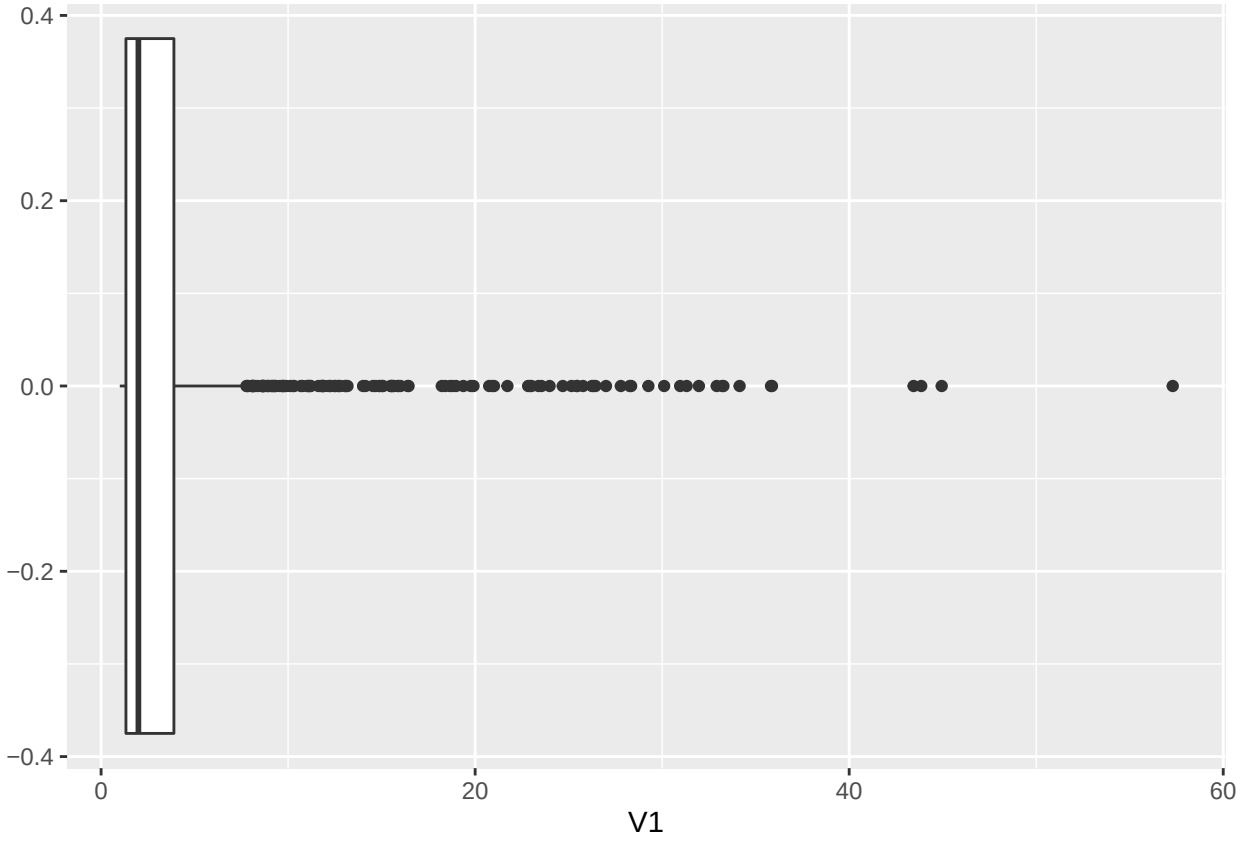
```
Data2_clean$z_score <- (Data2_clean$V1 - mean(Data2_clean$V1))/sd(Data2_clean$V1)
Data2_z_score_outliers = subset(Data2_clean, z_score <= 3)
length(Data2_z_score_outliers$z_score)
```

```
## [1] 985
```

```
summary(Data2_z_score_outliers)
```

```
##        V1            z_score
##  Min.   : 1.000   Min.   :-0.23355
##  1st Qu.: 1.330   1st Qu.:-0.21970
##  Median : 1.996   Median :-0.19175
##  Mean   : 4.196   Mean   :-0.09934
##  3rd Qu.: 3.895   3rd Qu.:-0.11201
##  Max.   :57.300   Max.   : 2.13054
```

```
ggplot(Data2_z_score_outliers) + geom_boxplot(aes(x=V1))
```



It is easily observed that the z-score-method keeps a more appropriate amount of data, we still have some values that are high, but they are not unreasonable. Thus, it appears that the appropriate method to use to find outliers is to use z-scores in the range $[-3, 3]$.

2.3

```
summary(Data2_clean$V1)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
```

```
##   1.000   1.333   2.022   6.562   4.067 402.944
```

```
summary(Data2_z_score_outliers$V1)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   1.330   1.996   4.196   3.895  57.300
```

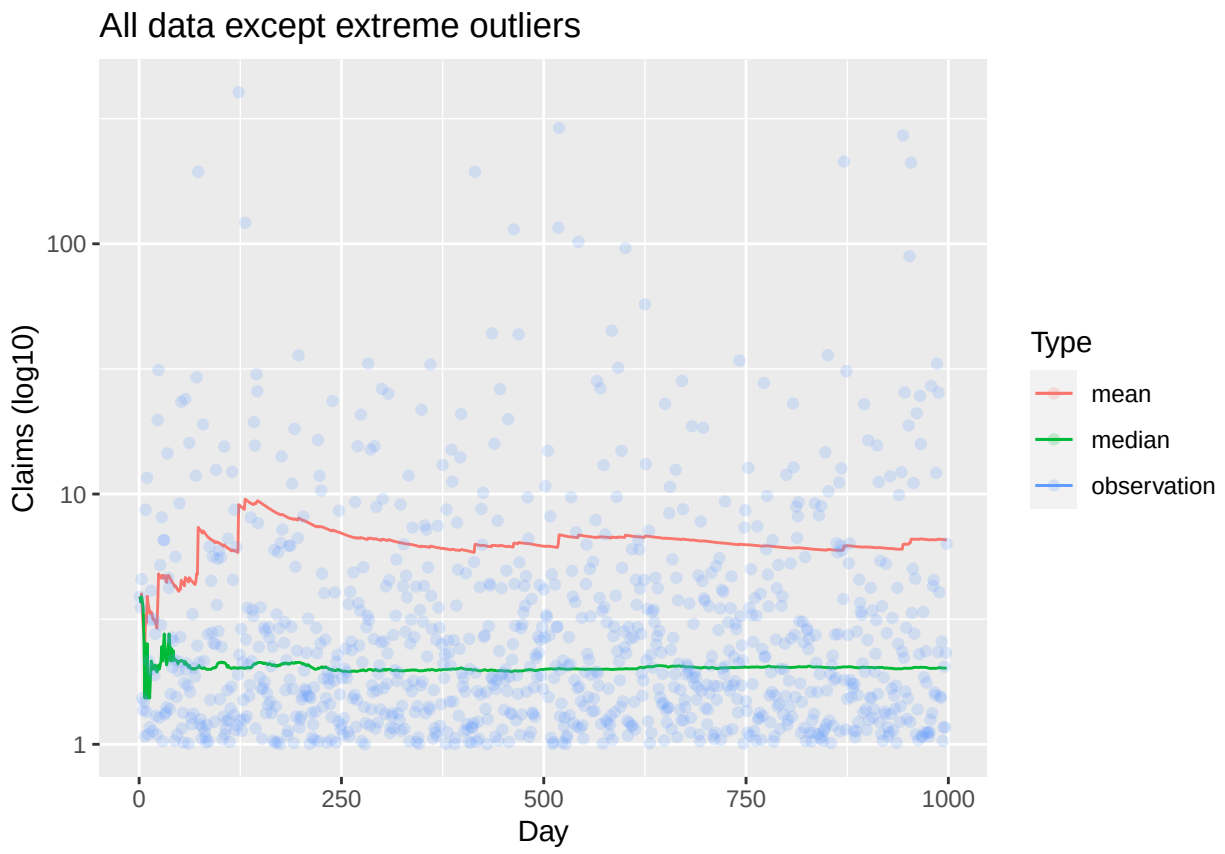
It can be seen that in the "original" (still excluding negative and > 4000 values) dataset there is a difference between the mean and median. This of course has to do with the behavior of these two methods. Much of the data is indeed close to 2, which the median indicate. However, we have a quite large spread towards the right tail, which can be observed in the mean, that is over 3 times higher than the median. Looking at the cleaned data (removing outliers with z-score-method) the median is almost the same as in the "original" data. There is however, a difference in the mean. The cleaned data is now just 2 times higher than the median. This is expected given that we removed some of the outliers, so the mean changes significantly.

2.4

```
rolling_mean <- data.frame(sapply(1:length(Data2_clean$V1),
↪  function(i)mean(Data2_clean$V1[1:i])))

rolling_median <- data.frame(sapply(1:length(Data2_clean$V1),
↪  function(i)median(Data2_clean$V1[1:i])))

colnames(rolling_mean) <- c("V1")
colnames(rolling_median) <- c("V1")
ggplot() + geom_line(data=rolling_mean, mapping=aes(x=1:length(V1),y=V1,colour="mean")) +
↪  geom_line(data=rolling_median, mapping=aes(x=1:length(V1),y=V1,colour="median")) +
↪  geom_point(data=Data2_clean, aes(x=1:length(V1),y=V1, colour="observation"),
↪  alpha=0.2) + labs(x="Day",y="Claims (log10)",title = "All data except extreme
↪  outliers", colour="Type") + scale_y_log10()
```



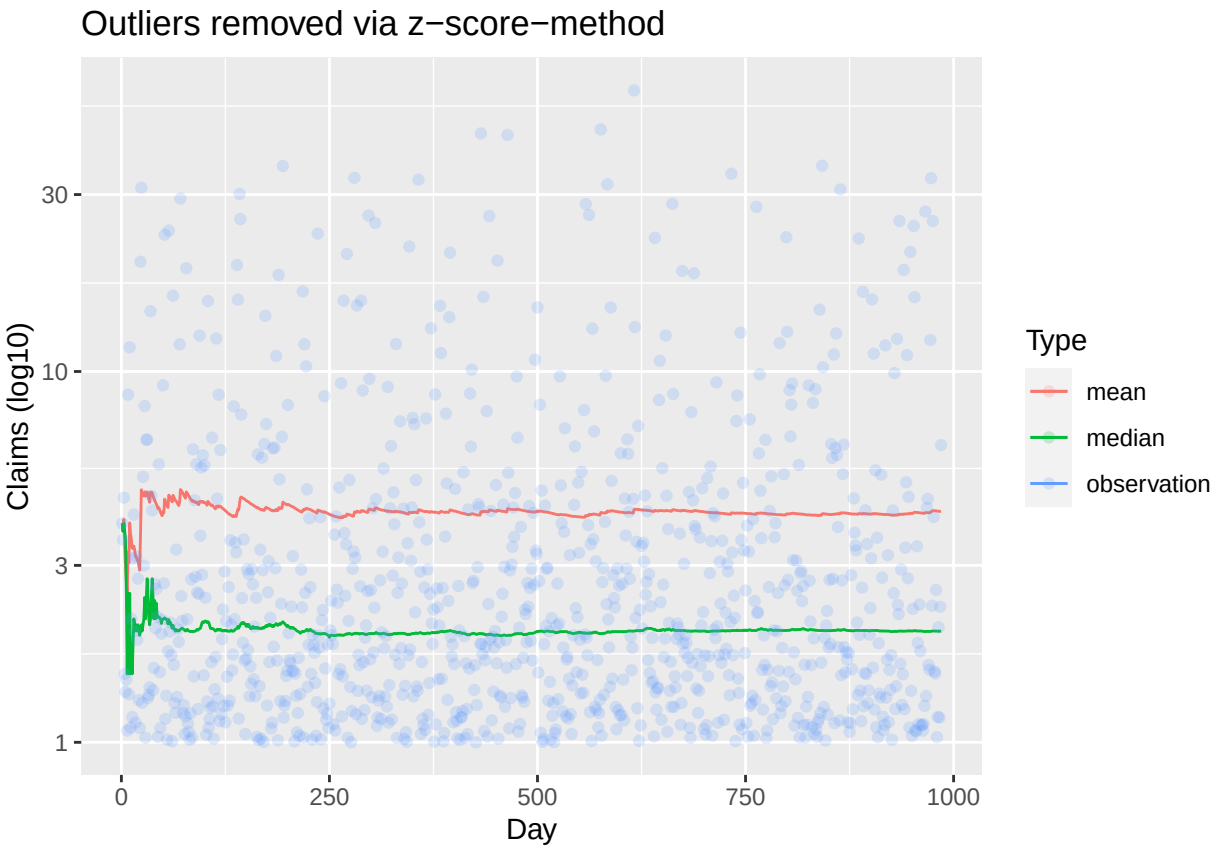
These plots shows the clean data (only removing the two extreme outliers), and the rolling mean (blue) and median (red). We use a logarithmic y-axis to be able to represent some of the larger observations. One can note a larger discrepancy between the median and mean and it can be seen that the median stabilizes much quicker than the mean, something we will bear in mind.

```
rolling_mean <- data.frame(sapply(1:length(Data2_z_score_outliers$V1),
↪  function(i)mean(Data2_z_score_outliers$V1[1:i])))

rolling_median <- data.frame(sapply(1:length(Data2_z_score_outliers$V1),
↪  function(i)median(Data2_z_score_outliers$V1[1:i])))

colnames(rolling_mean) <- c("V1")
colnames(rolling_median) <- c("V1")

ggplot()+geom_line(data=rolling_mean, mapping=aes(x=1:length(V1),y=V1,colour="mean")) +
↪  geom_line(data=rolling_median, mapping=aes(x=1:length(V1),y=V1,colour="median")) +
↪  geom_point(data=Data2_z_score_outliers, aes(x=1:length(V1),y=V1,
↪  colour="observation"),alpha=0.2) + labs(x="Day",y="Claims (log10)",title = "Outliers
↪  removed via z-score-method", colour="Type") + scale_y_log10()
```



In this graph, which also have a $\log_{10}$ y-axis, the dataset is a bit smaller and disregards all observations not contained in range $[-3, 3]$. One can see that the mean and median, in principle, stabilizes at the same time. As a matter of fact the mean appear to be more stable than the median in the beginning, something we did not notice in the previous graph.

With these remarks in mind we can start to summarize which method and which measure is preferred. The median appears to be the best when considering both scenarios. When having almost all the data still in use it is still a good measure on the claim amounts. The median tells us that half of it is above, and the other is below this value. However, the median does not clearly reflect days where there is a larger claim amount, but it is the most stable with and without outliers.