Outlier Detection: US Crime

Alena Fedash

2022-09-13

Contents

Summary of Results **1**

Solution in R **2**
Data exploration 2
Check data for normal distribution 3
Outlier detection 6

Summary of Results

I started the task with data exploration. I noticed that the data is right-skewed and that the median is significantly lower-bound, while the the distance from it to the max value is almost twice as big as to the min value.

In order to perform Grubb's test, I check the data for normal distribution, as Grubb's is usually used on normally distributed data.

First, I performed a visual inspection. I built a histogram and compared it with a normal curve, constructed a boxplot of the Crime variable and a QQ-plot. The distribution seemed to be very right-skewed, which again hinted that there would be outliers on the right table. However, the distribution was not flat and rather bell-shaped, which means that even if the data distr. is not normal, the core of it is normally distributed. The QQ plot and the Boxplot supported that and showed some point on the right which were far from the the rest of the data. I noticed at least 1 or 2 point outliers in the data.

Second, I checked for normal distribution using Shapiro-Wilk's test, since the sample data is less than 50 point and Shapiro-Wilk's is one of the most powerful tests for normality. As expected, I got a p-value less than 0.05 (with a 95% decision interval).

Despite the results of the test, I continued with Grubb's test. All metrics of normality showed something exception on the right tail - the outliers that we are looking for.

The first round of outlier detection tests showed that the min values is not an outlier, and the max probably is. I got a p value a bit higher than 0.05, but with a non-normal distribution that was expected, so I concluded that the max point of 1993 might be and outlier, and decided to remove it and test the second highest point.

After removing the first outlier, I tested for normality again - p value increase, but the distribution was still not normal. I reran the Grubb's test on the second high (max in the new data frame) value - and it was an outlier too. On the left the min value was again not an outlier.

I removed the second outlier, and this time Shapiro-Wilk's declared the data to be normally distributed. It was still skewed to the right, but the density on both ends was the same. THere was a point on a QQ plot, however, which could be a third outlier, so I ran Grubb's again. It showed that there was no outlier on either end this time.

I stopped at this point, as we should not over-clean the data and over-perfect the distribution - we might leave ourselves without some valuable insights in the situation that we're studying. The new points outside boxplot's whiskers (there were two) were not so far from the plot's ends now, so I consider them to be part of the data. Perhaps, those are high numbers of murders which are related to some other factors in the data.

All in all, the median of the data after two outliers' removal remained near the initial one (still lower-bound), but the difference in the distance from mean to min and from mean to max value has significantly reduced. If we had to build a model with this data, we would be able to use it now, as there are no more outliers.

Grubb's test was very helpful in detecting outliers, especially in telling the true ones from false ones (point looking like outliers on the graph may not be such). However, it is important to check the data for normality and keep the results in mind when performing the test - in my case, due to non-normality there was slight confusion in the first test round - I had to accept a point as an outlier, although the p value was a bit bigger than needed for that. Hence, in such situations it is important to have some possible error in mind and look at the data as a whole.

Below is the step-by-step solution in R with more comments.

Solution in R

First, load (install) necessary libraries.

```
library(dplyr)
library(tidyverse)
library(dslabs)
library(data.table)
library(ggplot2)
library(plotly)
library(outliers)
```

Read the file:

```
data <- read.table("uscrime.txt",
                   header = TRUE,
                   stringsAsFactors = FALSE,
                   sep = "",
                   dec = ".")
head(data)
```

```
##      M So   Ed  Po1  Po2    LF   M.F Pop   NW    U1  U2 Wealth Ineq     Prob
## 1 15.1  1  9.1  5.8  5.6 0.510  95.0  33 30.1 0.108 4.1   3940 26.1 0.084602
## 2 14.3  0 11.3 10.3  9.5 0.583 101.2  13 10.2 0.096 3.6   5570 19.4 0.029599
## 3 14.2  1  8.9  4.5  4.4 0.533  96.9  18 21.9 0.094 3.3   3180 25.0 0.083401
## 4 13.6  0 12.1 14.9 14.1 0.577  99.4 157  8.0 0.102 3.9   6730 16.7 0.015801
## 5 14.1  0 12.1 10.9 10.1 0.591  98.5  18  3.0 0.091 2.0   5780 17.4 0.041399
## 6 12.1  0 11.0 11.8 11.5 0.547  96.4  25  4.4 0.084 2.9   6890 12.6 0.034201
##      Time Crime
## 1 26.2011   791
## 2 25.2999  1635
## 3 24.3006   578
## 4 29.9012  1969
## 5 21.2998  1234
## 6 20.9995   682
```

Data exploration

Step 1 - check for NA values:

```r
is.null(data)
```

```
## [1] FALSE
```

None NA values in our data set.

Step 2 - See summary of important metrics for our target column - Crime

```r
summary(data[ncol(data)])
```

```
##      Crime
##  Min.   : 342.0
##  1st Qu.: 658.5
##  Median : 831.0
##  Mean   : 905.1
##  3rd Qu.:1057.5
##  Max.   :1993.0
```

We can see that the average number of crimes per 100 000 people is 905. It is obvious that our data is right-skewed, since the maximum value is 900 points apart from the 3rd quartile (compare to around 300 point difference between min value and 1st quartile).

My first expectation here would be that if we do find outliers, they will most likely be on the right tail (near the max).

Now, we can move on to the main task. For convenience, I will create a Crime variable to store the Crime column of the data:

```r
#since we're learning about crime data - extract it to a variable

crime <- data$Crime
crime
```

```
##  [1]  791 1635  578 1969 1234  682  963 1555  856  705 1674  849  511  664  798
## [16]  946  539  929  750 1225  742  439 1216  968  523 1993  342 1216 1043  696
## [31]  373  754 1072  923  653 1272  831  566  826 1151  880  542  823 1030  455
## [46]  508  849
```

Check data for normal distribution

Grubb's test is commonly used to find outliers in a **normally distributed data set**. Hence, it is important that before running the model we check the data for normality.

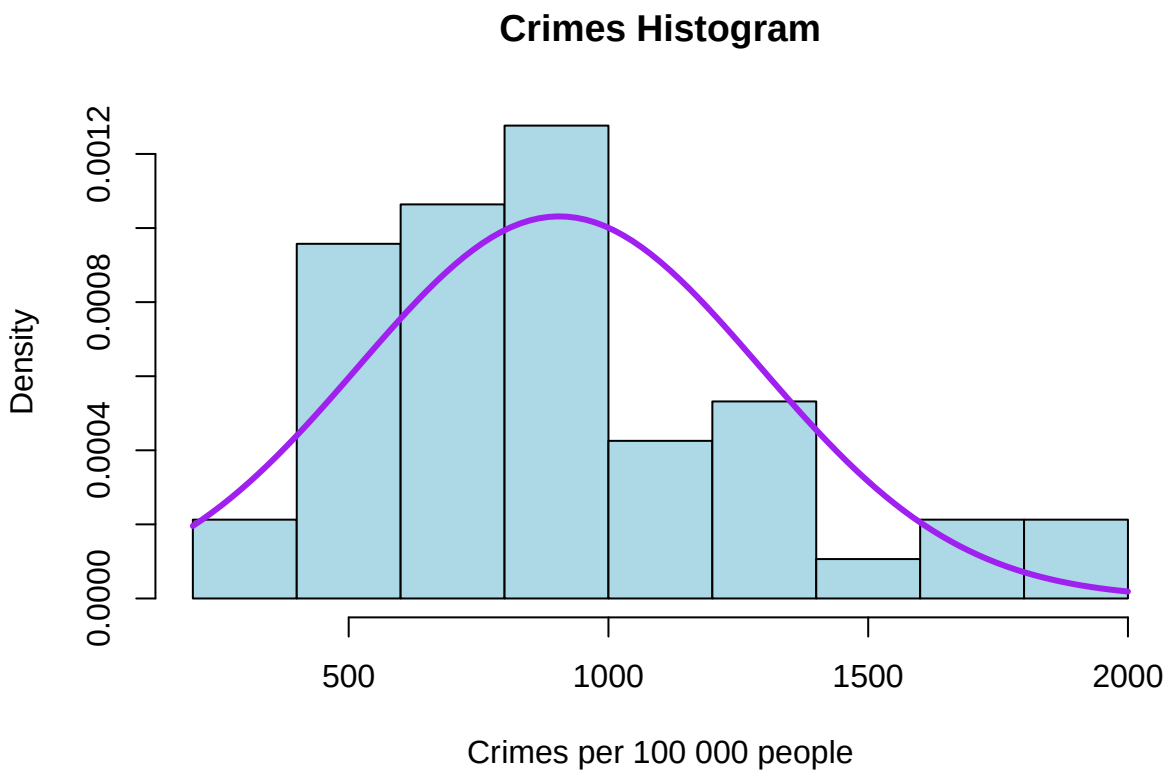
Visual inspection

Before running any normality tests, it is important to perform a visual inspection based on a histogram of the dependent variable. Ideally, we want it to be bell-shaped (not flat), with no skewness to any side (no long tails).

I will visualize the distribution of the Crime variable with a histogram and add a normal distribution curve to compare it with:

```r
#Histogram
hist(crime,
     main = "Crimes Histogram",
     xlab = "Crimes per 100 000 people",
     col = "lightblue",
     probability = TRUE)
#add norm distr curbv
```

```
curve(dnorm(x,
            mean=mean(crime),
            sd=sd(crime)),
      add=TRUE,
      col = "purple",
      lwd = 3)
```



From the plot, it is obvious that crime data is **skewed to the right**, as it has a long right tail. This is what we first noticed comparing mean, the 3rd quartile and the max value. It looks like there are a few values that are quite high. This might give us a hint that the normality test may rule this data to be not normally distributed because of those few high values.

However, the histogram clearly shows that the distribution is bell-shaped, so no matter the test result we could still run Grubb's outlier detection.

In general, it looks like there might be outliers on the right - which is exactly what we need to find.

Shapiro-Wilk test

Although the histogram hints that we may not get a normal distribution from the normality test, it is still usefull to run it.

I will use **Shapiro-Wilk test** for that purpose, as it is the most widely used method that is more powerful in non-normality detection. It is more appropriate for samples with less than 50 data points, which is exactly what we have - 47 data entries.

H0: the data is normally distributed **Ha**: the distribution is not normal

I will use a confidence interval of 95%, so we would say that data is normally distributed if the p value is over 0.05. Lets run the test:

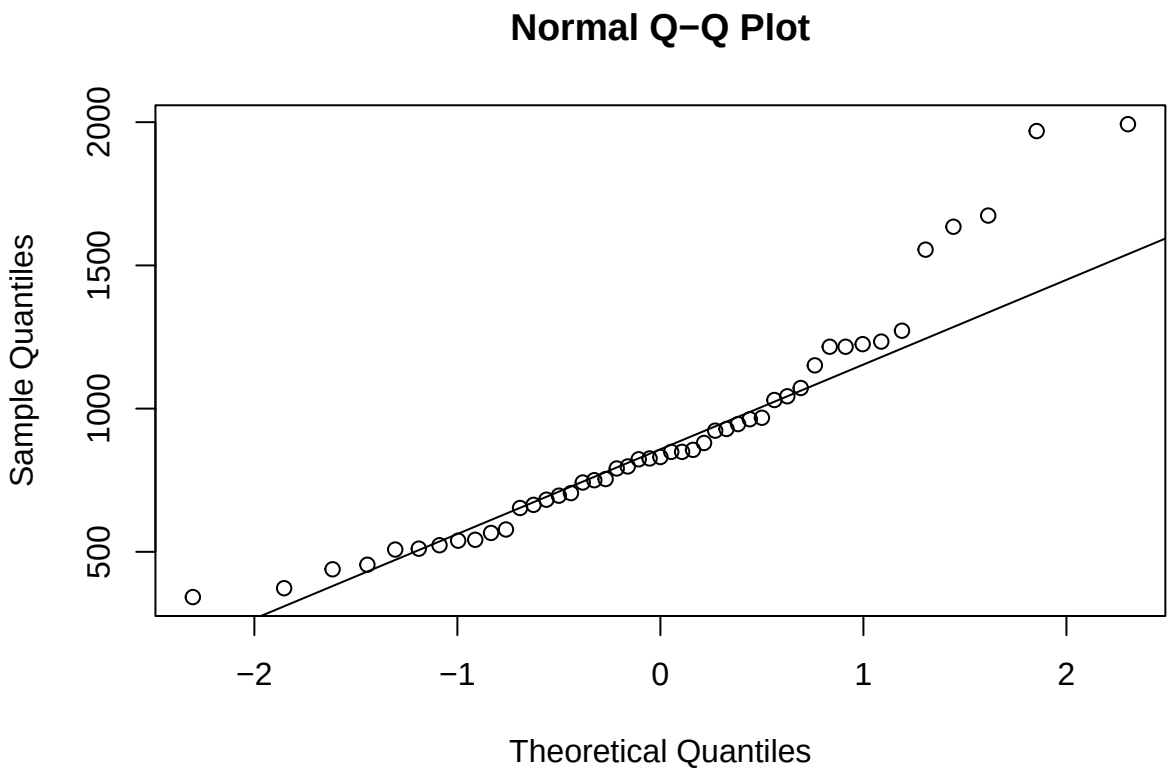
```
shapiro.test(crime)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  crime
## W = 0.91273, p-value = 0.001882
```

Our p value is 0.0019, less than 0.05, so we **reject H0** - the data is **not normally distributed, according to Shapiro-Wilk test**.

However, we can still run Grubb's test, if the 'core data' of our sample is normally distributed. We can see that from the histogram, but let's also check using a QQ plot:

```
qqnorm(crime)
qqline(crime)
```



Yes - **the core of our data set is normally distributed, according to the QQ plot**. We can see that the non-normality is due to the high points on the righ tail - at least one-wo, or even more outliers.

So, we can go on and do the Grubb's test to identify those outliers.

Outlier detection

Boxplot

I start with a boxplot to visually inspect where the outliers are located. I will use plotly package to be able to see the exact coordinates of each outlier. I will color the points located beyond the whiskers red.

```
plot_ly(y=crime,
        type="box",
        quartilemethod="inclusive",
        fillcolor="lightblue",
        boxpoints="outliers",
        #color for outliers
        marker=list(color="darkred")) %>%
  layout(title="Number of Crimes per 100 000 people",
         xaxis=list(title="Crimes"),
         yaxis=list(title="Number of commited crimes"))
```

We can see that there are likely no outliers on the left tail, and two or even three on the right - 1993, 1969 and 1674. The latter, however, is located close to the right whisker, so it may not be considered an outlier by Grubb's.

Grubb's test

Now we will run the Grubb's test. With normal data we would consider p>0.05 as a sign that the point is not an outlier. We can do this here too since we have a 95% decision interval. However, keeping in mind that our data is not normally distributet, we should understand that p value may be a little bit off, and we would need to evaluate it ourselves to rule whether it is an outlier or not.

H0 - point is not an outlier **Ha** - point is an outlier

First, I check the left tail to confirm there are no outliers, as we saw on the boxplot.

Grubb's tests the min or max point on chosen side, so we can only test one point at a time.

```
#for left tail
grubbs.test(crime,
            type=10,
            opposite = TRUE, #?
            )
```

```
##
##  Grubbs test for one outlier
##
## data:  crime
## G = 1.45589, U = 0.95292, p-value = 1
## alternative hypothesis: lowest value 342 is an outlier
```

We got a p value of 1, so we accept H0 - **there are no outliers on the left tail (min values)**, p=1 is a very definitive answer, so the min point definitely belongs to the data set and should not be removed.

Now, let's check the max point on the right:

```
#check max values (RIGHT  TAIL)
grubbs.test(crime,
            type=10,#test for one outlier on one tail
            opposite = FALSE #specify tail = right (max vals)
            )
```

```
##
```

```
##  Grubbs test for one outlier
##
## data:  crime
## G = 2.81287, U = 0.82426, p-value = 0.07887
## alternative hypothesis: highest value 1993 is an outlier
```

We got a p value of 0.07887. It is not less than 0.05, so normally we would accept H0 and say that this is not an outlier. However, since our data is not normally distributed and all visual inspections showed that at least one point on the right tail is an outlier - it is worth considering the max point an outlier. Moreover, p value is not at 1, so it is not as definitive as with the left tail.

Based on the QQ plot and boxplot, which both showed outliers on the right, we will say that the max point might be an outlier.

We will do the following:

- remove the max value, considering it an outlier
- run Shapiro normality test again
- check if the min and max points of the data set are outliers too.

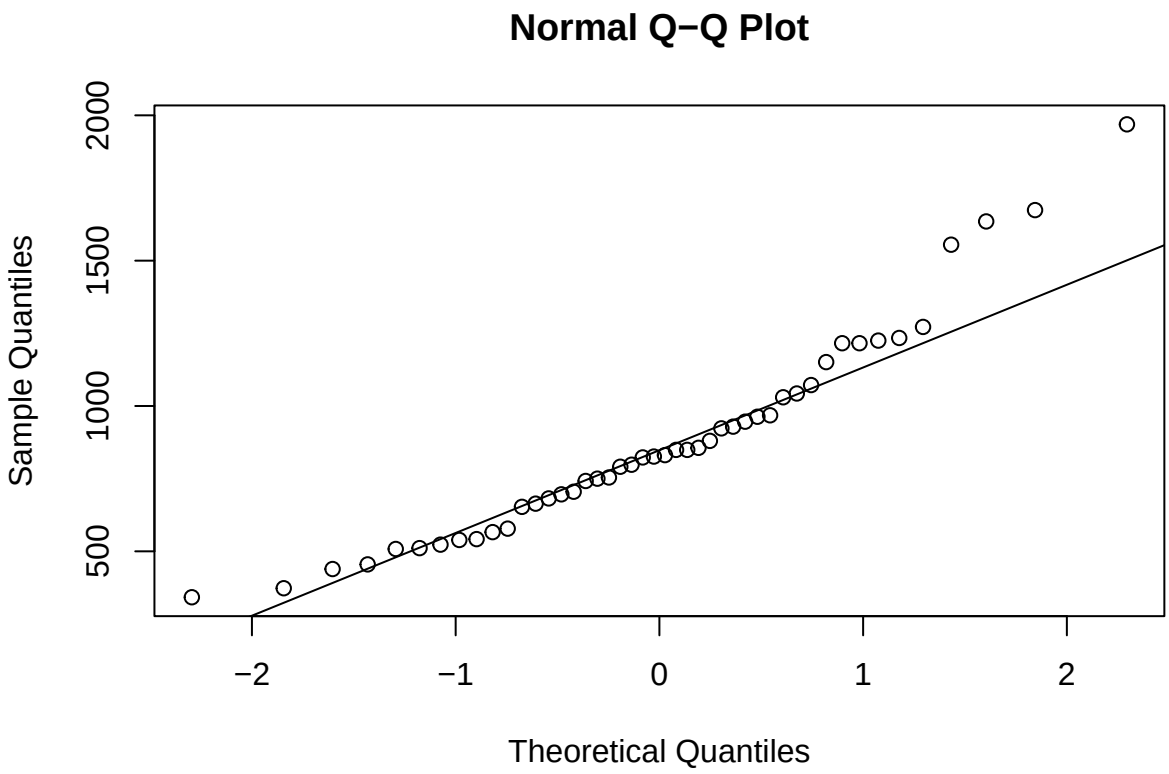
Removing the first outlier

Remove the outlier that we detected - max value of 1993:

```
#New ds without the outlier on the right
crime.1 <- crime[-which.max(crime)]
```

Check QQ plot to see if now the whole picture looks closer to normal distribution:

```
#see graphs to see if anythings changed
qqnorm(crime.1)
qqline(crime.1)
```



There is at least one more outlier on the right, although the whole picture looks a bit better now.

Let's run Shapiro-Wilk again to see if p has increased and risen above 0.05

```
shapiro.test(crime.1)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  crime.1
## W = 0.93207, p-value = 0.01001
```

P value has increased, however the distribution is still not normal. Perhaps, if there is one more outlier and we remove it, our data will be normally distributed.

Let's run Grubb's again:

```
#just in case check for left tail (min)
grubbs.test(crime.1,
            type=10,
            opposite = TRUE, #?
)
```

```
##
##  Grubbs test for one outlier
##
## data:  crime.1
## G = 1.51947, U = 0.94755, p-value = 1
## alternative hypothesis: lowest value 342 is an outlier
```

Still no outliers on the left (min)

Check the right tail - now the max value is 1969:

```
grubbs.test(crime.1,
            type=10,#test for one outlier on one tail
            opposite = FALSE #specify tail = right (max vals)
)
```

```
##
##  Grubbs test for one outlier
##
## data:  crime.1
## G = 3.06343, U = 0.78682, p-value = 0.02848
## alternative hypothesis: highest value 1969 is an outlier
```

p-value is less than 0.05, so 1969 is definitely an outlier.

We will remove the second outlier on the right and repeat the whole process again.

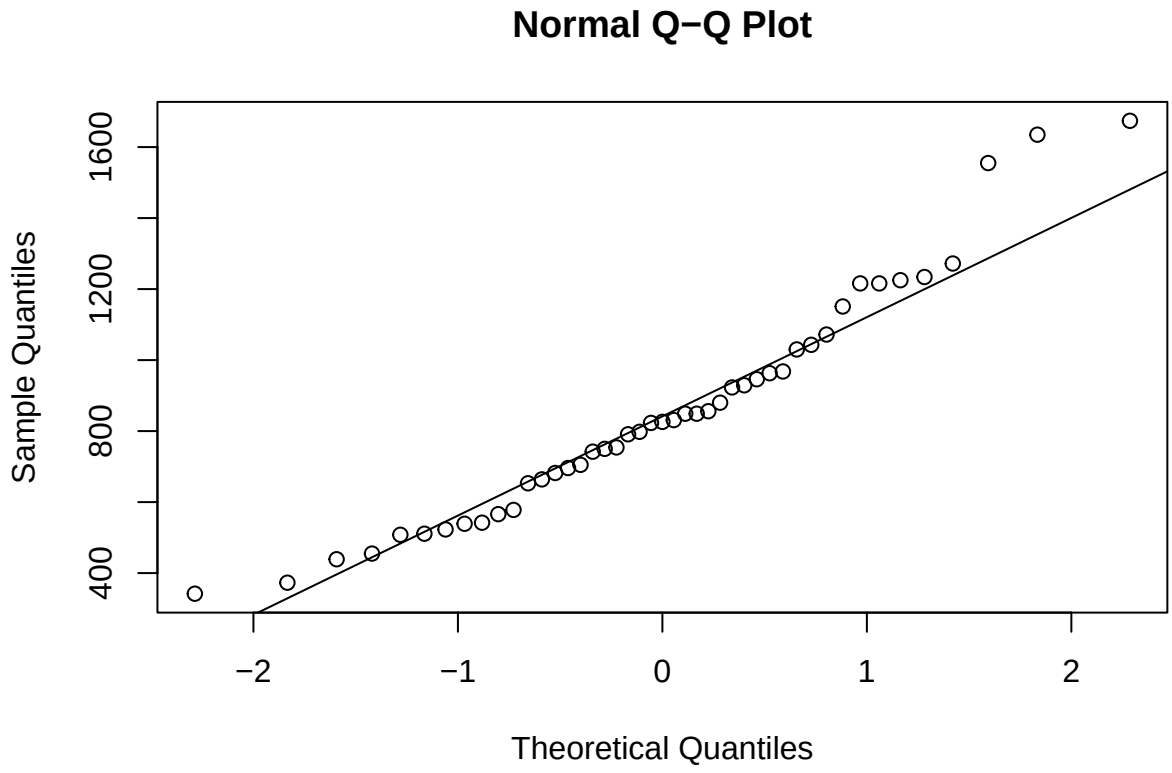
Removing the second outlier

Create a new df without the second and first outliers:

```
crime.2 <- crime.1[-which.max(crime.1)]
```

Check for normality:

```
qqnorm(crime.2)
qqline(crime.2)
```



Looks better, let's see if the distribution is normal now. There might be another outlier on the right.

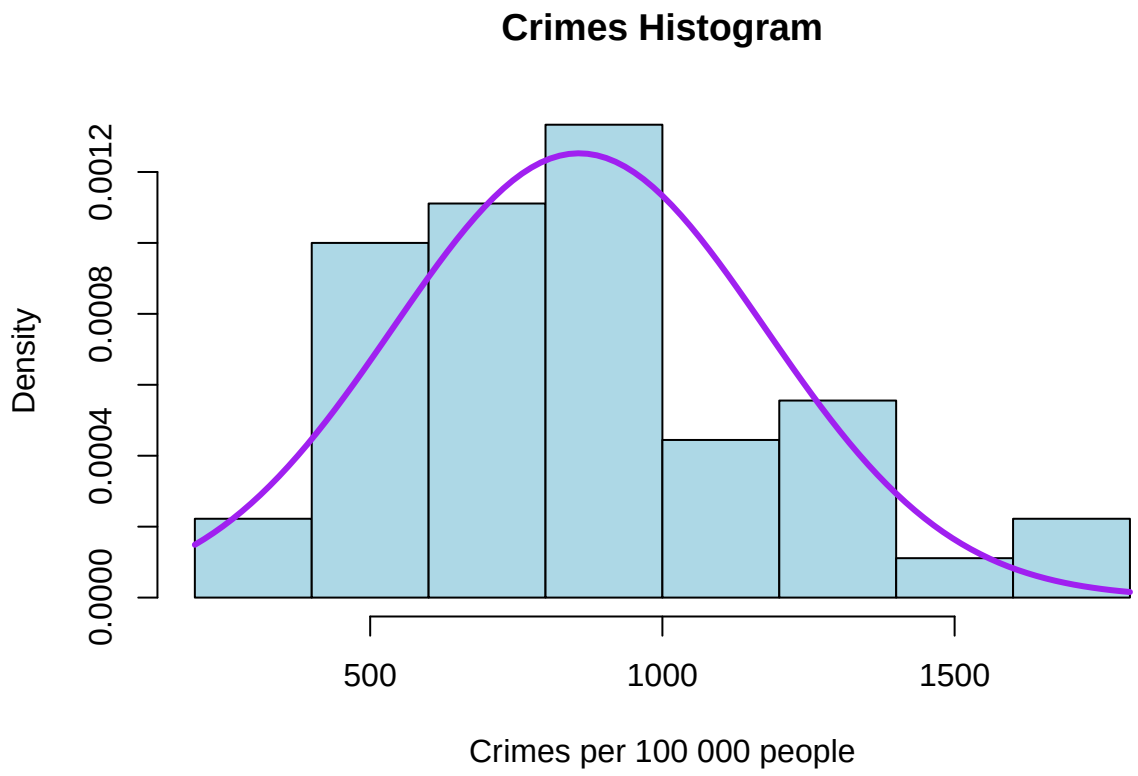
```
shapiro.test(crime.2)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  crime.2
## W = 0.95119, p-value = 0.05634
```

Yes, the distribution is finally normal now. This means that the removal of two outlier has helped a lot, and now the data would be more or less suitable to build a model based on it.

Let's see the new distribution on the histogram:

```
hist(crime.2,
     main = "Crimes Histogram",
     xlab = "Crimes per 100 000 people",
     col = "lightblue",
     probability = TRUE)
#add norm distr curve
curve(dnorm(x,
            mean=mean(crime.2),
            sd=sd(crime.2)),
      add=TRUE,
      col = "purple",
      lwd = 3)
```



Though the distribution still has a slightly longer right tail, it is not as big nowand the density is the same as

on the left tail.

Let's check if the max point now is the third outlier:

```
#just in case check for left tail (min dps)
grubbs.test(crime.2,
            type=10,
            opposite = TRUE, #?
)
```

```
##
##  Grubbs test for one outlier
##
## data:  crime.2
## G = 1.61796, U = 0.93915, p-value = 1
## alternative hypothesis: lowest value 342 is an outlier
```

Still no outliers on the left tail.

```
grubbs.test(crime.2,
            type=10,#test for one outlier on one tail
            opposite = FALSE #specify tail = right (max vals)
)
```

```
##
##  Grubbs test for one outlier
##
## data:  crime.2
## G = 2.56457, U = 0.84712, p-value = 0.1781
## alternative hypothesis: highest value 1674 is an outlier
```

Though we thought we might have a third outlier, Grubb's prooves otherwise (and it should not be wrong now that the data is normally distributed).

We should not remove this data point - although it seems to be excess here, it might provide valuable insights into the amount of Crimes. We should not overclean the data - it might make our future model too positive.

Let's check the boxplot for the new dataframe - see if there are any points outside the whiskers and how far they are:

```
plot_ly(y=crime.2,
        type="box",
        quartilemethod="inclusive",
        fillcolor="lightblue",
        boxpoints="outliers",
        #color for outliers
        marker=list(color="darkred")) %>%
  layout(title="Number of Crimes per 100 000 people (outliers removed)",
         xaxis=list(title="Crimes"),
         yaxis=list(title="Number of commited crimes"))
```

Yes, there are still two point outside the whiskers, but they might provide needed information. ALso, they are very close to the margins of the boxplot - not as far as the first ones were.

Finally, let's compare the five main metrics of our initial data and the one without outliers, and see the difference

```
after_removal <- data.table(
  c("minimum", "lower-hinge", "median", "upper-hinge", "maximum"),
  fivenum(crime),
```

```
  fivenum(crime.2))
colnames(after_removal) <- c("fivenum", "Initial data", "Data w/o 2 outliers")
after_removal
```

```
##        fivenum Initial data Data w/o 2 outliers
## 1:     minimum        342.0                 342
## 2: lower-hinge        658.5                 653
## 3:      median        831.0                 826
## 4: upper-hinge       1057.5                1030
## 5:     maximum       1993.0                1674
```

Median is still more lower-bound and has not changed much, but the hinges have - and now the maximum points do not seem to be too far from the median, as it was in the beginning.

Overall, removing the two outliers has helped us bring the distribution of the data to normal and balance out the tails of the data (though the right tail is still bigger - that's just how the data is - perhaps there are less situations where the number of crimes is very high). Now with this data we would be able to start building a model.