Introduction

An outlier is a value or an observation that is distant from other observations, that is to say, a data point that differs significantly from other data points. An observation must always be compared to other observations made on the same phenomenon before actually calling it an outlier. Indeed, someone who is 200 cm tall (6'7″ in US) will most likely be considered as an outlier compared to the general population, but that same person may not be considered as an outlier if we measured the height of basketball players.

An outlier may be due to the variability inherent in the observed phenomenon. For example, it is often the case that there are outliers when collecting data on salaries, as some people make much more money than the rest. Outliers can also arise due to an experimental, measurement or encoding error. For instance, a human weighting 786 kg (1733 pounds) is clearly an error when encoding the weight of the subject. Her or his weight is most probably 78.6 kg (173 pounds) or 7.86 kg (17 pounds) depending on whether weights of adults or babies have been measured.

In this article, I present several approaches to detect outliers in R, from simple techniques such as descriptive statistics (including minimum, maximum, histogram, boxplot and percentiles) to more formal techniques such as the Hampel filter, the Grubbs, the Dixon and the Rosner tests for outliers.

Although there is no strict or unique rule whether outliers should be removed or not from the dataset before

doing statistical analyses, it is quite common to, at least, remove outliers that are due to an experimental or measurement error (like the weight of 786 kg (1733 pounds) for a human). Some statistical tests require the absence of outliers in order to draw sound conclusions, but removing outliers is not recommended in all cases and must be done with caution.

This article will not tell you whether you should remove outliers or not (nor if you should impute them with the median, mean, mode or any other value), but it will help you to detect them in order to, as a first step, verify them. After their verification, it is then your choice to exclude or include them for your analyses. Removing or keeping outliers mostly depend on three factors:

1. The domain and context of your analyses. In some domains, it is common to remove outliers as they often occur due to a malfunctioning process. In other fields, outliers are kept because they contain valuable information. It also happens that analyses are performed twice, once with and once without outliers to evaluate their impact on the conclusions.
2. Whether the tests you are going to apply are robust to the presence of outliers or not. For instance, the slope of a simple linear regression may significantly vary with just one outlier, whereas non-parametric tests such as the Wilcoxon test are usually robust to outliers.
3. How distant are the outliers from other observations. Some observations considered as outliers (according to the techniques presented below) are actually not really extreme compared to all other observations, while other potential outliers may be really distant from the rest of the observations.

The dataset `mpg` from the `{ggplot2}` package will be used to illustrate the different approaches of outliers detection in R, and in particular we will focus on the variable `hwy` (highway miles per gallon).

Descriptive statistics

Minimum and maximum

The first step to detect outliers in R is to start with some descriptive statistics, and in particular with the minimum and maximum.

In R, this can easily be done with the `summary()` function:

```
dat <- ggplot2::mpg
summary(dat$hwy)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   12.00   18.00   24.00   23.44   27.00   44.00
```

where the minimum and maximum are respectively the first and last values in the output above. Alternatively, they can also be computed with the `min()` and `max()` functions:

```
min(dat$hwy)
```

```
## [1] 12
```

```
max(dat$hwy)
```

```
## [1] 44
```

Some clear encoding mistake like a weight of 786 kg (1733 pounds) for a human will already be easily detected by this very simple technique.

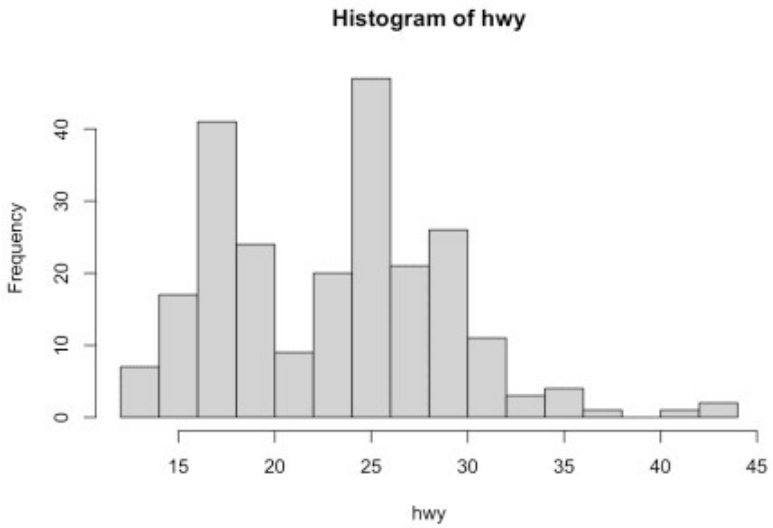
Histogram

Another basic way to detect outliers is to draw a histogram of the data.

Using R base (with the number of bins corresponding to the square root of the number of observations in order to have more bins than the default option):

```
hist(dat$hwy,
  xlab = "hwy",
```

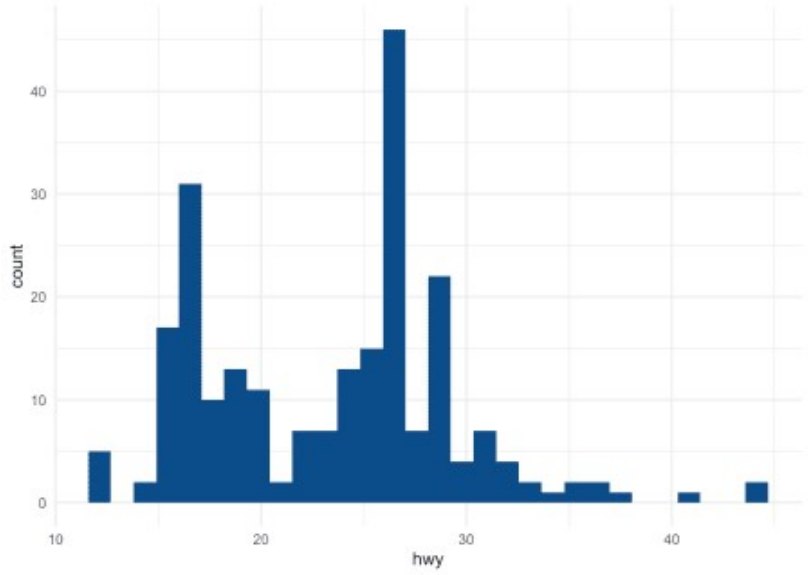
```
  main = "Histogram of hwy",
  breaks = sqrt(nrow(dat))
) # set number of bins
```



or using `ggplot2` (via the `esquisse` addin):

```
library(ggplot2)

ggplot(dat) +
  aes(x = hwy) +
  geom_histogram(bins = 30L, fill = "#0c4c8a") +
  theme_minimal()
```



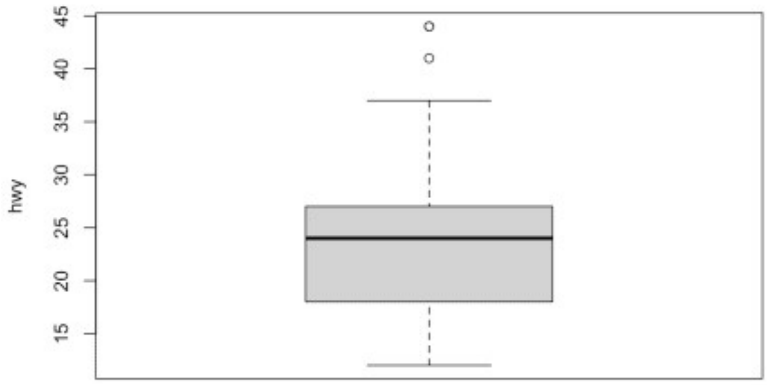
From the histogram, there seems to be a couple of observations higher than all other observations (see the bar on the right side of the plot).

Boxplot

In addition to histograms, boxplots are also useful to detect potential outliers.

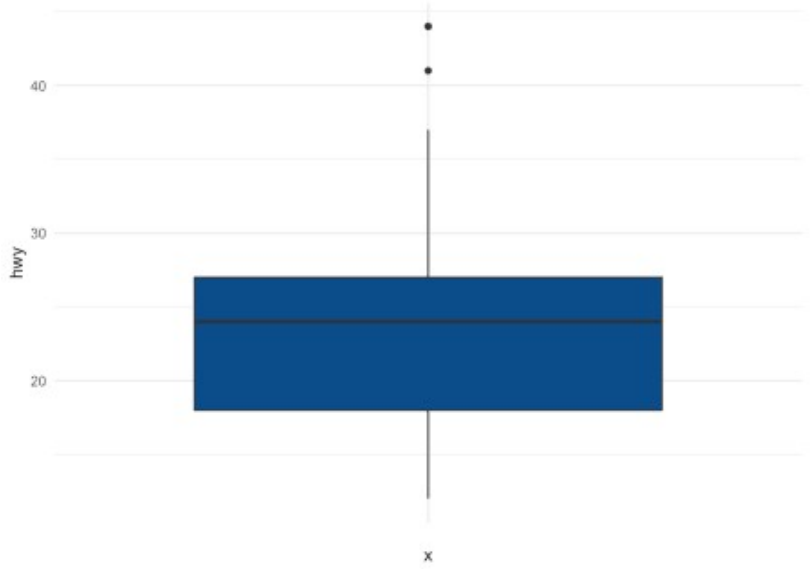
Using R base:

```
boxplot(dat$hwy,
  ylab = "hwy"
)
```



or using `ggplot2`:

```
ggplot(dat) +
  aes(x = "", y = hwy) +
  geom_boxplot(fill = "#0c4c8a") +
  theme_minimal()
```



A boxplot helps to visualize a quantitative variable by displaying five common location summary (minimum, median, first and third quartiles and maximum) and any observation that was classified as a suspected outlier using the interquartile range (IQR) criterion. The IQR criterion means that all observations above $q_{0.75} + 1.5 \cdot IQR$ or below $q_{0.25} – 1.5 \cdot IQR$ (where $q_{0.25}$ and $q_{0.75}$ correspond to first and third quartile respectively, and IQR is the difference between the third and first quartile) are considered as potential outliers by R. In other words, all observations outside of the following interval will be considered as potential outliers:

$$I = [q_{0.25} – 1.5 \cdot IQR; q_{0.75} + 1.5 \cdot IQR]$$

Observations considered as potential outliers by the IQR criterion are displayed as points in the boxplot. Based on this criterion, there are 2 potential outliers (see the 2 points above the vertical line, at the top of the boxplot).

Remember that it is not because an observation is considered as a potential outlier by the IQR criterion that you should remove it. Removing or keeping an outlier depends on (i) the context of your analysis, (ii) whether the tests you are going to perform on the dataset are robust to outliers or not, and (iii) how far is the outlier from other observations.

It is also possible to extract the values of the potential outliers based on the IQR criterion thanks to the `boxplot.stats()$out` function:

```
boxplot.stats(dat$hwy)$out
```

```
## [1] 44 44 41
```

As you can see, there are actually 3 points considered as potential outliers: 2 observations with a value of 44 and 1 observation with a value of 41.

Thanks to the `which()` function it is possible to extract the row number corresponding to these outliers:

```
out <- boxplot.stats(dat$hwy)$out
out_ind <- which(dat$hwy %in% c(out))
out_ind
```

```
## [1] 213 222 223
```

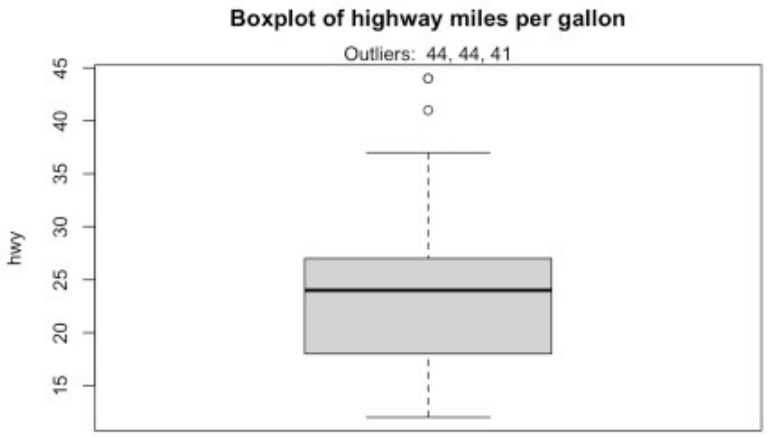
With this information you can now easily go back to the specific rows in the dataset to verify them, or print all variables for these outliers:

```
dat[out_ind, ]
```

```
## # A tibble: 3 x 11
##   manufacturer model   displ  year   cyl trans   drv     cty   hwy fl
class
##
## 1 volkswagen   jetta     1.9  1999     4 manual… f        33    44 d
compact
## 2 volkswagen   new be…   1.9  1999     4 manual… f        35    44 d
subcom…
## 3 volkswagen   new be…   1.9  1999     4 auto(l… f        29    41 d
subcom…
```

It is also possible to print the values of the outliers directly on the boxplot with the `mtext()` function:

```
boxplot(dat$hwy,
  ylab = "hwy",
  main = "Boxplot of highway miles per gallon"
)
mtext(paste("Outliers: ", paste(out, collapse = ", ")))
```



Percentiles

This method of outliers detection is based on the percentiles. With the percentiles method, all observations that lie outside the interval formed by the 2.5 and 97.5 percentiles will be considered as potential outliers. Other percentiles such as the 1 and 99, or the 5 and 95 percentiles can also be considered to construct the interval.

The values of the lower and upper percentiles (and thus the lower and upper limits of the interval) can be computed with the `quantile()` function:

```
lower_bound <- quantile(dat$hwy, 0.025)
lower_bound
```

```
## 2.5%
##   14
```

```
upper_bound <- quantile(dat$hwy, 0.975)
upper_bound
```

```
##  97.5%
## 35.175
```

According to this method, all observations below 14 and above 35.175 will be considered as potential outliers. The row numbers of the observations outside of the interval can then be extracted with the `which()` function:

```
outlier_ind <- which(dat$hwy < lower_bound | dat$hwy > upper_bound)
outlier_ind
```

```
##  [1]  55  60  66  70 106 107 127 197 213 222 223
```

Then their values of highway miles per gallon can be printed:

```
dat[outlier_ind, "hwy"]
```

```
## # A tibble: 11 x 1
##      hwy
##
##  1    12
##  2    12
##  3    12
##  4    12
##  5    36
##  6    36
##  7    12
##  8    37
##  9    44
## 10    44
## 11    41
```

Alternatively, all variables for these outliers can be printed:

```
dat[outlier_ind, ]
```

```
## # A tibble: 11 x 11
##    manufacturer model    displ  year   cyl trans  drv     cty   hwy fl
class
##
##  1 dodge        dakota …   4.7  2008     8 auto(… 4         9    12 e
pickup
##  2 dodge        durango…   4.7  2008     8 auto(… 4         9    12 e     suv
##  3 dodge        ram 150…   4.7  2008     8 auto(… 4         9    12 e
pickup
```

```
##  4 dodge        ram 150…   4.7  2008     8 manua… 4         9    12 e
pickup
##  5 honda        civic      1.8  2008     4 auto(… f        25    36 r
subco…
##  6 honda        civic      1.8  2008     4 auto(… f        24    36 c
subco…
##  7 jeep         grand c…   4.7  2008     8 auto(… 4         9    12 e     suv
##  8 toyota       corolla    1.8  2008     4 manua… f        28    37 r
compa…
##  9 volkswagen   jetta      1.9  1999     4 manua… f        33    44 d
compa…
## 10 volkswagen   new bee…   1.9  1999     4 manua… f        35    44 d
subco…
## 11 volkswagen   new bee…   1.9  1999     4 auto(… f        29    41 d
subco…
```

There are 11 potential outliers according to the percentiles method. To reduce this number, you can set the percentiles to 1 and 99:

```
lower_bound <- quantile(dat$hwy, 0.01)
upper_bound <- quantile(dat$hwy, 0.99)

outlier_ind <- which(dat$hwy < lower_bound | dat$hwy > upper_bound)

dat[outlier_ind, ]
```

```
## # A tibble: 3 x 11
##   manufacturer model   displ  year   cyl trans   drv     cty   hwy fl
class
##
## 1 volkswagen   jetta     1.9  1999     4 manual… f        33    44 d
compact
## 2 volkswagen   new be…   1.9  1999     4 manual… f        35    44 d
subcom…
## 3 volkswagen   new be…   1.9  1999     4 auto(l… f        29    41 d
subcom…
```

Setting the percentiles to 1 and 99 gives the same potential outliers as with the IQR criterion.

Hampel filter

Another method, known as Hampel filter, consists of considering as outliers the values outside the interval (I) formed by the median, plus or minus 3 median absolute deviations (MAD):[1]

$$I = [median – 3 \cdot MAD; median + 3 \cdot MAD]$$

where MAD is the median absolute deviation and is defined as the median of the absolute deviations from the data's median $\tilde{X} = median(X)$:

$$MAD = median(|X_i – \tilde{X}|)$$

For this method we first set the interval limits thanks to the `median()` and `mad()` functions:

```
lower_bound <- median(dat$hwy) - 3 * mad(dat$hwy)
lower_bound
```

```
## [1] 1.761
```

```
upper_bound <- median(dat$hwy) + 3 * mad(dat$hwy)
upper_bound
```

```
## [1] 46.239
```

According to this method, all observations below 1.761 and above 46.239 will be considered as potential outliers. The row numbers of the observations outside of the interval can then be extracted with the `which()` function:

```
outlier_ind <- which(dat$hwy < lower_bound | dat$hwy > upper_bound)
outlier_ind
```

```
## integer(0)
```

According to the Hampel filter, there is no potential outlier for the `hwy` variable.

Statistical tests

In this section, we present 3 more formal techniques to detect outliers:

1. Grubbs's test
2. Dixon's test
3. Rosner's test

These 3 statistical tests are part of more formal techniques of outliers detection as they all involve the computation of a test statistic that is compared to tabulated critical values (that are based on the sample size and the desired confidence level).

Note that the 3 tests are appropriate only when the data (without any outliers) are **approximately normally distributed**. The normality assumption must thus be verified before applying these tests for outliers (see how to test the normality assumption in R).

Grubbs's test

The Grubbs test allows to detect whether the highest or lowest value in a dataset is an outlier.

The Grubbs test detects one outlier at a time (highest or lowest value), so the null and alternative hypotheses are as follows:

- H_0: The *highest* value is **not** an outlier
- H_1: The *highest* value is an outlier

if we want to test the highest value, or:

- H_0: The *lowest* value is **not** an outlier
- H_1: The *lowest* value is an outlier

if we want to test the lowest value.

As for any statistical test, if the ***p*-value is less** than the chosen **significance threshold** (generally $\alpha = 0.05$) then the null hypothesis is rejected and we will conclude that the **lowest/highest value is an outlier**. On the contrary, if the ***p*-value is greater or equal** than the significance level, the null hypothesis is not rejected, and we will conclude that, based on the data, we do not reject the hypothesis that the **lowest/highest value is not an outlier**.

Note that the Grubbs test is not appropriate for sample size of 6 or less ($n \le 6$).

To perform the Grubbs test in R, we use the `grubbs.test()` function from the `{outliers}` package:

```
# install.packages("outliers")
library(outliers)
test <- grubbs.test(dat$hwy)
test
```

```
##
```

```
##  Grubbs test for one outlier
##
## data:  dat$hwy
## G = 3.45274, U = 0.94862, p-value = 0.05555
## alternative hypothesis: highest value 44 is an outlier
```

The *p*-value is 0.056. At the 5% significance level, we do not reject the hypothesis that the *highest* value 44 is **not** an outlier.

By default, the test is performed on the highest value (as shown in the R output: `alternative hypothesis: highest value 44 is an outlier`). If you want to do the test for the lowest value, simply add the argument `opposite = TRUE` in the `grubbs.test()` function:

```
test <- grubbs.test(dat$hwy, opposite = TRUE)
test
```

```
##
##  Grubbs test for one outlier
##
## data:  dat$hwy
## G = 1.92122, U = 0.98409, p-value = 1
## alternative hypothesis: lowest value 12 is an outlier
```

The R output indicates that the test is now performed on the lowest value (see `alternative hypothesis: lowest value 12 is an outlier`).

The *p*-value is 1. At the 5% significance level, we do not reject the hypothesis that the *lowest* value 12 is **not** an outlier.

For the sake of illustration, we will now replace an observation with a more extreme value and perform the Grubbs test on this new dataset. Let's replace the \(34^{th}\) row with a value of 212:

```
dat[34, "hwy"] <- 212
```

And we now apply the Grubbs test to test whether the highest value is an outlier:

```
test <- grubbs.test(dat$hwy)
test
```

```
##
##  Grubbs test for one outlier
##
## data:  dat$hwy
## G = 13.72240, U = 0.18836, p-value < 2.2e-16
## alternative hypothesis: highest value 212 is an outlier
```

The *p*-value is < 0.001. At the 5% significance level, we conclude that the *highest* value 212 is an outlier.

Dixon's test

Similar to the Grubbs test, Dixon test is used to test whether a single low or high value is an outlier. So if more than one outliers is suspected, the test has to be performed on these suspected outliers individually.

Note that Dixon test is most useful for small sample size (usually \(n \le 25\)).

To perform the Dixon's test in R, we use the `dixon.test()` function from the `{outliers}` package. However, we restrict our dataset to the 20 first observations as the Dixon test can only be done on small sample size (R will throw an error and accepts only dataset of 3 to 30 observations):

```
subdat <- dat[1:20, ]
test <- dixon.test(subdat$hwy)
test
```

```
##
##  Dixon test for outliers
##
## data:  subdat$hwy
## Q = 0.57143, p-value = 0.006508
## alternative hypothesis: lowest value 15 is an outlier
```

The results show that the lowest value 15 is an outlier (*p*-value = 0.007).

To test for the highest value, simply add the `opposite = TRUE` argument to the `dixon.test()` function:

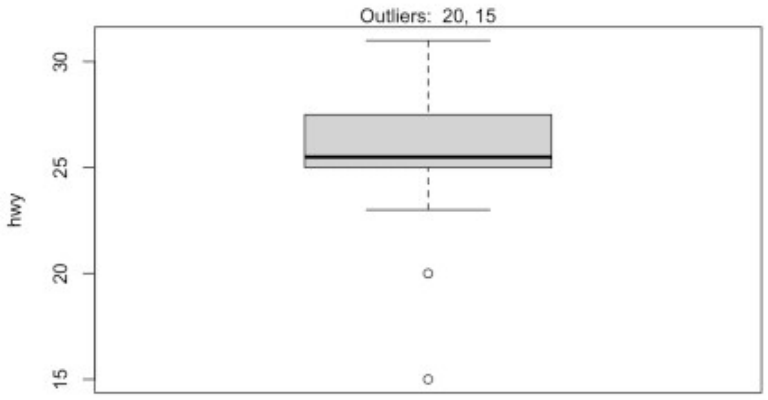
```
test <- dixon.test(subdat$hwy,
  opposite = TRUE
)
test
```

```
##
##  Dixon test for outliers
##
## data:  subdat$hwy
## Q = 0.25, p-value = 0.8582
## alternative hypothesis: highest value 31 is an outlier
```

The results show that the highest value 31 is **not** an outlier (*p*-value = 0.858).

It is a good practice to always check the results of the statistical test for outliers against the boxplot to make sure we tested **all** potential outliers:

```
out <- boxplot.stats(subdat$hwy)$out
boxplot(subdat$hwy,
  ylab = "hwy"
)
mtext(paste("Outliers: ", paste(out, collapse = ", ")))
```



From the boxplot, we see that we could also apply the Dixon test on the value 20 in addition to the value 15 done previously. This can be done by finding the row number of the minimum value, excluding this row number from the dataset and then finally apply the Dixon test on this new dataset:

```
# find and exclude lowest value
remove_ind <- which.min(subdat$hwy)
subsubdat <- subdat[-remove_ind, ]

# Dixon test on dataset without the minimum
```

```
test <- dixon.test(subsubdat$hwy)
test
```

```
##
##  Dixon test for outliers
##
## data:  subsubdat$hwy
## Q = 0.44444, p-value = 0.1297
## alternative hypothesis: lowest value 20 is an outlier
```

The results show that the second lowest value 20 is **not** an outlier (*p*-value = 0.13).

Rosner's test

Rosner's test for outliers has the advantages that:

1. it is used to **detect several outliers at once** (unlike Grubbs and Dixon test which must be performed iteratively to screen for multiple outliers), and
2. it is designed to avoid the problem of masking, where an outlier that is close in value to another outlier can go undetected.

Unlike Dixon test, note that Rosner test is most appropriate when the sample size is large ($n \ge 20$). We therefore use again the initial dataset `dat`, which includes 234 observations.

To perform the Rosner test we use the `rosnerTest()` function from the `{EnvStats}` package. This function requires at least 2 arguments: the data and the number of suspected outliers `k` (with `k = 3` as the default number of suspected outliers).

For this example, we set the number of suspected outliers to be equal to 3, as suggested by the number of potential outliers outlined in the boxplot.[2]

```
library(EnvStats)
test <- rosnerTest(dat$hwy,
  k = 3
)
test
```

```
## $distribution
## [1] "Normal"
##
## $statistic
##       R.1       R.2       R.3
## 13.722399  3.459098  3.559936
##
## $sample.size
## [1] 234
##
## $parameters
## k
## 3
##
## $alpha
## [1] 0.05
##
## $crit.value
## lambda.1 lambda.2 lambda.3
## 3.652091 3.650836 3.649575
##
## $n.outliers
## [1] 1
```

```
##
## $alternative
## [1] "Up to 3 observations are not\n                                        from the
same Distribution."
##
## $method
## [1] "Rosner's Test for Outliers"
##
## $data
##   [1]  29  29  31  30  26  26  27  26  25  28  27  25  25  25  25  24  25  23
##  [19]  20  15  20  17  17  26  23  26  25  24  19  14  15  17  27 212  26  29
##  [37]  26  24  24  22  22  24  24  17  22  21  23  23  19  18  17  17  19  19
##  [55]  12  17  15  17  17  12  17  16  18  15  16  12  17  17  16  12  15  16
##  [73]  17  15  17  17  18  17  19  17  19  19  17  17  17  16  16  17  15  17
##  [91]  26  25  26  24  21  22  23  22  20  33  32  32  29  32  34  36  36  29
## [109]  26  27  30  31  26  26  28  26  29  28  27  24  24  24  22  19  20  17
## [127]  12  19  18  14  15  18  18  15  17  16  18  17  19  19  17  29  27  31
## [145]  32  27  26  26  25  25  17  17  20  18  26  26  27  28  25  25  24  27
## [163]  25  26  23  26  26  26  26  25  27  25  27  20  20  19  17  20  17  29
## [181]  27  31  31  26  26  28  27  29  31  31  26  26  27  30  33  35  37  35
## [199]  15  18  20  20  22  17  19  18  20  29  26  29  29  24  44  29  26  29
## [217]  29  29  29  23  24  44  41  29  26  28  29  29  29  28  29  26  26  26
##
## $data.name
## [1] "dat$hwy"
##
## $bad.obs
## [1] 0
##
## $all.stats
##   i   Mean.i      SD.i Value Obs.Num     R.i+1 lambda.i+1 Outlier
## 1 0 24.21795 13.684345   212      34 13.722399   3.652091    TRUE
## 2 1 23.41202  5.951835    44     213  3.459098   3.650836   FALSE
## 3 2 23.32328  5.808172    44     222  3.559936   3.649575   FALSE
##
## attr(,"class")
## [1] "gofOutlier"
```

The interesting results are provided in the `$all.stats` table:

```
test$all.stats
```

```
##   i   Mean.i      SD.i Value Obs.Num     R.i+1 lambda.i+1 Outlier
## 1 0 24.21795 13.684345   212      34 13.722399   3.652091    TRUE
## 2 1 23.41202  5.951835    44     213  3.459098   3.650836   FALSE
## 3 2 23.32328  5.808172    44     222  3.559936   3.649575   FALSE
```

Based on the Rosner test, we see that there is only one outlier (see the `Outlier` column), and that it is the observation 34 (see `Obs.Num`) with a value of 212 (see `Value`).

Additional remarks

You will find many other methods to detect outliers:

1. in the `{outliers}` packages,
2. via the `lofactor()` function from the `{DMwR}` package: Local Outlier Factor (LOF) is an algorithm used to identify outliers by comparing the local density of a point with that of its neighbors,
3. the `outlierTest()` from the `{car}` package gives the most extreme observation based on the given model and allows to test whether it is an outlier, and

4. in the `{OutlierDetection}` package.

Note also that some transformations may “naturally” eliminate outliers. The natural log or square root of a value reduces the variation caused by extreme values, so in some cases applying these transformations will eliminate the outliers.

Thanks for reading. I hope this article helped you to detect outliers in R via several descriptive statistics (including minimum, maximum, histogram, boxplot and percentiles) or thanks to more formal techniques of outliers detection (including Hampel filter, Grubbs, Dixon and Rosner test). It is now your turn to verify them, and if they are correct, decide how to treat them (i.e., keeping, removing or imputing them) before conducting your analyses.