#### Documentation

The README does not state the problems that the software is designed to solve, or its target audience.

🡪 I’ve added this section in the README file :

***# Routliers***

***Routliers is designed to help researchers to detect univariate and multivariate outliers, using robust methods: - The MAD method for detecting univariate outliers (see Leys et al. 2013) - The MMCD method for detecting multivariate outliers (see Leys et al. 2018) – The Mahalanobis distance method for detecting multivariate outliers is also available in order to facilite comparisons with the MMCD method. However, this method is less robust than the MMCD method and should be avoided. Different functions are available in order to (a) detect univariate and multivariate outliers (b) plot univariate and multivariate outliers.***

There are no installation instructions in the README

***🡪 I’ve added an installation section in the README file :***

***## Installation***

***Currently, this package exists in a development version on GitHub. To use the package, you need to install it directly from GitHub using the `install\_github` function from `devtools`. You can use the following code to install the development version of `Routliers`:***

***``` r***

***library(devtools)***

***install\_github("mdelacre/Routliers")***

***```***

***As a note, one of the dependencies in `Routliers` (i.e. MASS) have its own dependencies. You may be prompted to install additional packages to be able to install `Routliers`.***

- [ ] \*\*Vignette(s)\*\* demonstrating major functionality that runs successfully locally

There are vignettes that detail how to use the major functionality of the package

- [ ] \*\*Function Documentation:\*\* for all exported functions in R help

There are three main functions in the package:

- `outliers\_MAD`

- `outliers\_MCD`

- `outliers\_mahalanobis`

***Following afurther suggestion, they are now 6 functions in the package as explained later.***

* ***Outliers\_mad : was splitted into 2 functions : outliers\_mad and plot\_outliers\_mad***
* ***Outliers\_mcd : was splitted into 2 fucntions : outliers\_mcd and plot\_outliers\_mcd***
* ***Outliers\_mahalanobis : was splitted into 2 functions : outliers\_mahalanobis and plot\_outliers\_mahalanobis***

These all have documentation written, although in its current state it will not install as a package because some of the examples need to have the example text commented out

***All examples were reworked and commented out.***

- [ ] \*\*Examples\*\* for all exported functions in R Help that run successfully locally

All but one of the examples can be run locally - the one that does not break is `outliers\_MAD`.

***🡪 All examples can now be run locally.***

- [ ] \*\*Community guidelines\*\* including contribution guidelines in the README or CONTRIBUTING, and DESCRIPTION with `URL`, `BugReports` and `Maintainer` (which may be autogenerated via `Authors@R`).

* **We have added these lines in the description file :**

**Authors@R: c(**

**person("Marie", "Delacre", email = "marie.delacre@ulb.ac.be", role = "cre"),**

**person("Olivier", "Klein", email = "Klein.Olivier@ulb.ac.be", role = "aut"))**

They do not include contribution guidelines, but these are not needed for this review.

#### Functionality

- [ ] \*\*Installation:\*\* Installation succeeds as documented.

Running `devtools::install()` is successful

- [ ] \*\*Functionality:\*\* Any functional claims of the software been confirmed.

The package states:

> Package containing robust ways of dealing with outliers

> the outliers\_MAD funtion can be used in order to detect univariate outliers

> the outliers\_bivar function can be used in order to detect bivariate outliers

***Outliers\_bivar were splitted into outliers\_mcd and outliers\_mahalanobis, but we forgot to adapt the description of the package. We have now corrected it.***

##### Using `outliers\_MAD`

```{r o-mad-rnorm}

library(Routliers)

o\_mad <- outliers\_MAD(data = rnorm(150),

constant = 1.4826,

threshold = 3,

na.rm = TRUE,

plot = TRUE,

verbose = TRUE)

o\_mad

```

This tells me that there are some high values, and what value they are. This is really great! Let's see what happens when we use another dataset.

```{r}

temp\_mad <- outliers\_MAD(data = airquality$Temp,

constant = 1.4826,

threshold = 3,

na.rm = TRUE,

plot = TRUE,

verbose = TRUE)

```

This errored unexpectedly - the values here are integers. I changed the `outliers\_MAD.R` file to use `inherits`, we can then more flexibly suggest that integer and numeric should be acceptable. Unless you specifically do not want to include intgers?

* ***We don’t have reason to not include integers, so as suggested, we used inherits in the R code (see lines 50-51 in the outliers\_mad script) :***

***# If data are numeric or integer, applying the function. Otherwise, stopping it.***

***if(inherits(x,c("numeric","integer")) == FALSE) stop("Data are neither numeric nor integer")***

A few things that would improve the usability of this function,

- use all lowercase - `outliers\_mad` ***🡪 done***

- Create a separate plotting function `plot\_outlier\_mad`. In general it is best practice for functions to have only one purpose.

***As previously mentioned, the original outliers\_mad function was splitted into outliers\_mad (printing main results) and plot\_outliers\_mad (plotting outliers).***

- Improve the print method. Creating good print methods is hard! But you should instead remove the option `verbose = TRUE`, and make an outliers\_MAD.print method that contains the `cat` code. Generally speaking `verbose = TRUE` arguments are intended for use with functions that provide output during calculation - such as long running statistical models.

***We made an outliers\_mad.print method, following an excellent tutorial of Friendrich Leisch (2009).***

- The `constant` argument. This seems to be a very very specific number - from the documentation I am unclear what this number does, and why I would want control of it. I would suggest removing it as an option, or providing a clearer description of what the constant is.

***The constant argument (now called b), is a constant depending on the assumed distribution underlying the data, that equals 1/Q(0.75). When the normal distribution is assumed, the constant 1.4826 is used (and it makes the MAD and SD of normal distributions comparable).***

***We have added this information in the documentation of the outliers\_mad function.***

- I would suggest using some real data for an example use with this function, while a simulation works, but it would be much better to see it used with a real dataset.

***We totally agree that it is much better. We have added three datasets in the package :***

* ***Attacks : data collected the day after the terrorist attacks in Brussels (on the morning of 22 March 2016) assessing the Sense of Coherence (SOC-13 self report questionnaire; Antonovsky, 1987), anxiety and depression symptoms (HSCL-25; Derogatis et al., 1974) of 2077 subjects. These data were collected by Christophe Leys.***
* ***Morality : replication of expriments evaluating impact of psychological distance on moral judgment (Eyal, Liberman & Trope, 2008 ; Gong & Medin, 2012), found here :*** <https://osf.io/8wqvc/>
* ***Intention* *: study five of Rogers, T. & Milkman, K.L. (2016). Reminders through association. Psychological Science, 27, 973-986.***

##### Using `outliers\_MCD`

Let's use the example from the package.

```{r}

Sigma = matrix(c(1,.5,.5,1),2,2)

data = MASS::mvrnorm(100,

mu = rep(0,2),

Sigma = Sigma)

o\_mcd <-

outliers\_MCD(data = data,

h = .3,

na.rm = TRUE,

plot = TRUE,

verbose = TRUE)

```

- The example did not run because the MASS package needs to be loaded to use `mvrnorm`.

***🡪 We replaced this example by examples based on the datasets we implemented in the package***

- Similar note to earlier that the plot function should be converted to a `plot\_outlier\_mcd` function.

***🡪 As previously mentioned, we splitted the outliers\_mcd function into outliers\_mcd and plot\_outliers\_mcd functions.***

- Another note on the plot - what does this plot show me? I get a scatterplot with a regression line, but I am not sure how to interpret this. It this good?

***🡪 When there are no outliers, the regression line is the one using all data. On the other side, when there are outliers, two regression lines are plotted : one using all data and one excluding detected outliers.***

- I would also suggest finding some real data to use with the example ***🡪 done***

- A note on the options of the function

- `h` is not particularly descriptive for it's purpose, I would suggest something like `prop\_data` or describe what the `h` stands for in the documentation in terms of the MCD method.

* ***We now better explain what is the « h » in the documentation : if is the proportion of dataset to use in order to compute sample means and covariances***

- `alpha`. This seems to be a crucial important argument to help define outliers, but the documentation does not state this. `alpha` can mean many things - for example it is often used to define transparency in a plot. I suggest making this clearer in the documentation.

***🡪 You are totally right and we apology for the ambiguity. We define alpha as the nominal type I error probability (by default .01).***

A minor note on style.

- The package consistently uses `=` for assignment, which is fine, but there needs to be consistent spacing around the `=` sign.

***🡺 We now use <- for assignments (because it is the standard for R users and developers) with consistent spacing.***

###### outliers\_mahalanobis

```{r outlier-mahalanobis}

Sigma=matrix(c(1,.5,.5,1),2,2)

data=MASS::mvrnorm(100,mu=rep(0,2),Sigma=Sigma)

o\_ma <- outliers\_mahalanobis(data=data, h=.4,na.rm=TRUE,plot=TRUE, verbose=TRUE)

o\_ma$nbrow

```

- It seems that this uses the same documentation as `outliers\_MCD`. I suggest updating the documentation to describe how this is different.

***🡪 One again you are totally right. There were mistakes due to copy paste from the outliers\_mcd function. For example, the « h » argument was in the outliers\_mahalanobis function while it is not used at all inside the function. The documentation and the function were updated accordingly.***

- [ ] \*\*Performance:\*\* Any performance claims of the software been confirmed.

- The package helps identify outliers as suggested, but there could be more work done to improve the usability of the package.

- [ ] \*\*Automated tests:\*\* Unit tests cover essential functions of the package

and a reasonable range of inputs and conditions. All tests pass on the local machine.

There are no unit tests for this package.

***🡪 We have added unit tests, in order to be sure that our functions return objects from expected class (« outliers\_mad », « outliers\_mcd » and « outliers\_mahalanobis) because it is required for the print methods.***

---

### Review Comments

General points on the package

- This package provides methods to identify outliers with three methods:

- `outliers\_MAD`

- `outliers\_MCD`

- `outliers\_mahalanobis`

These functions all work, and appear to identify outliers.

As it stands, the journal does not have package guidelines, and so I used the rOpenSci package onboarding template to assist with the review, and also drew from my experience as a package developer.

At the moment, there are some general house keeping changes that need to be made to the package in order for it to be publication ready.

For example, in the DESCRIPTION file, the line:

```

COPYRIGHT HOLDER:

Marie Delacre,

Olivier Klein

```

Should be removed, and the information put into the LICENSE file.

This has been added in my version, using `devtools::use\_mit\_license()`.

* ***We didn’t know this function and used it in order to create a license file.***

Additionally, the package has the following warnings and notes (after I made some small changes to the package).

```

N checking DESCRIPTION meta-information ...

Malformed Description field: should contain one or more complete sentences.

WARNING

Dependence on R version ‘3.4.2’ not with patchlevel 0

...

── R CMD check results ─────────────────────────────── Routliers 0.0.0.9000 ────

Duration: 21.2s

❯ checking DESCRIPTION meta-information ... NOTE

Malformed Description field: should contain one or more complete sentences.

WARNING

Dependence on R version ‘3.4.2’ not with patchlevel 0

❯ checking DESCRIPTION meta-information ... NOTE

Malformed Description field: should contain one or more complete sentences.

WARNING

Dependence on R version ‘3.4.2’ not with patchlevel 0

0 errors ✔ | 1 warning ✖ | 1 note ✖

Error: R CMD check found WARNINGs

Execution halted

Exited with status 1.

```

This was generated with code `devtools::check()`

* ***In the current version of the package, there are neither warnings nor notes as we took all notes and warnings generated with the devtools ::check() into account.***

Additionally, it would be a good idea to try and address as many of the good practice checks found by running `goodpractice::gp()`

* ***We adressed as many of the good practice checks as found by this function. Considerint the unit tests, as previously mentioned, we have checked that our functions return objects from expected class (« outliers\_mad », « outliers\_mcd » and « outliers\_mahalanobis) because it is required for the print methods.***

```

────────────────────────────────────────────────────────────────────────

It is good practice to

✖ write unit tests for all functions, and all package code in

general. 0% of code lines are covered by test cases.

R/outliers\_MAD.R:25:NA

R/outliers\_MAD.R:26:NA

R/outliers\_MAD.R:29:NA

R/outliers\_MAD.R:30:NA

R/outliers\_MAD.R:31:NA

... and 202 more lines

✖ omit "Date" in DESCRIPTION. It is not required and it gets

invalid quite often. A build date will be added to the package when you

perform `R CMD build` on it.

* ***We removed it***

✖ add a "URL" field to DESCRIPTION. It helps users find

information about your package online. If your package does not have a

homepage, add an URL to GitHub, or the CRAN package package page.

* ***We have added an URL to Github, as the package in not on the CRAN yet.***

✖ use '<-' for assignment instead of '='. '<-' is the standard,

and R users and developers are used it and it is easier to read your code

for them if you use '<-'. 🡪 ***Changed***.

R/outliers\_MAD.R:18:13

R/outliers\_MAD.R:30:8

R/outliers\_MAD.R:31:14

R/outliers\_MAD.R:34:9

R/outliers\_MAD.R:35:6

... and 17 more lines

✖ avoid long code lines, it is bad for readability. Also, many

people prefer editor windows that are about 80 characters wide. Try make

your lines shorter than 80 characters

* ***There are no long code lines anymore.***

R/outliers\_MAD.R:47:1

R/outliers\_MAD.R:48:1

R/outliers\_MAD.R:57:1

R/outliers\_MAD.R:58:1

R/outliers\_MAD.R:60:1

... and 25 more lines

✖ avoid 1:length(...), 1:nrow(...), 1:ncol(...), 1:NROW(...) and

1:NCOL(...) expressions. They are error prone and result 1:0 if the

expression on the right hand side is zero. Use seq\_len() or seq\_along()

instead.

R/outliers\_mahalanobis.R:33:13

R/outliers\_MCD.R:31:13

* ***We replaced 1 :length() with seq\_len()***

✖ fix this R CMD check NOTE: Malformed Description field: should

contain one or more complete sentences. 🡪 ***Corrected***

```

There are eight main things that would greatly improve the usability of the package.

#### 1. function names

IT is great that the functions all have the same preface, `outliers\_`. However, the function names should use all lowercases. I suggest that they are renamed to:

- `outliers\_mcd`

- `ouliers\_mahanalobis`

- `outliers\_mad`

***Functions were renamed accordingly.***

#### 2. Create a `plot` function to plot the output of each of the `outliers\_\*` functions

Separating the plotting process from running diagnostics is important for clarity in analysis. Here, one function does one task. This means that you have a function that performs the analysis. And then another function that takes the output from the analysis and the data, and then generates a plot.

E.g.,

```r

o\_mcd <- outliers\_mcd(data)

o\_mcd

```

```

Results:

Median: 79

MAD: 8.896

Acceptable range of values:

lower MAD limit: 52.3132

upper MAD limit: 105.6868

Outliers: <none detected>

```

Note that I have truncated the output of `Outliers` as there were no outliers detected, and also indented some of the data.

```r

plot\_outliers\_mcd(o\_mcd, data)

```

<plot generated>

* ***As previously mentioned, the plotting is now separate from the running diagnostics (thanks to the creation of new plot functions).***

#### 3. Use print methods instead of verbose = TRUE

I suggest creating an `outliers\_mcd.print` method, so that when you run the function `o\_mad <- outliers\_mad(data)`, it does not print the output, unless you run `o\_mad` as is.

For an example, see this [stack overflow thread](<https://stackoverflow.com/questions/10938427/example-needed-change-the-default-print-method-of-an-object>)

***🡪 As previously mentioned, we made an outliers\_mad.print method, following an excellent tutorial of Friendrich Leisch (2009).***

#### 4. Use real data in the examples

Using real data as examples will help users use their own data with your functions.

***🡪 As previously mentioned, we included real datasets in our package.***

#### 5. Provide a way to identify the position of outliers

The functions identify that there are outliers, but I am not clear how I am supposed to use this information from the function to then identify which values in my data are outliers. I would suggest a new function like `detect\_outliers\_mcd` could be used like so:

```r

library(dplyr)

data %>%

mutate(outlier\_mcd = detect\_outliers\_mcd(column))

```

```

# A tibble: 5 x 2

x outlier\_mcd

<dbl> <lgl>

1 -1.76 TRUE

2 -1.41 TRUE

3 -1.68 TRUE

4 1.10 FALSE

5 0.404 FALSE

```

* ***We are not sure how to proceed, but the coordinates of outliers can be extracted in our functions. It is now better illustrated in the documentation.***

#### 6. Write a vignette to demonstrate how to use the functions in an analysis

Provide a demonstration of the workflow that you expect for each of your functions.

#### 7. Functions should work with dataframes

Some of the functions require data as a matrix with two columns (I believe), I suggest allowing them to work with two columns in a dataframe as well, to help them work better in a data analysis.

* ***Current functions can work with a dataframe and a matrix.***

#### 8. improve documentation and describe other literature

This package could be really useful in analysis, writing good documentation will help others use it!

The README file should help answer the following questions

You should also perhaps try and answer the question: "how is this different from package XX?". In particular, there are two packages that work with outlier detection that I would be interested in knowing how they differ:

1. HDOutliers (https://cran.r-project.org/web/packages/HDoutliers/index.html)

2. Outliers (https://cran.r-project.org/web/packages/Outliers)

3. anomalize (https://cran.r-project.org/web/packages/anomalize/)