Check your outliers! An accessible introduction to identifying statistical outliers in R with *easystats*

1 Abstract

xyz

2 Introduction

The improper handling of outliers can substantially affect statistical model estimations, thus contributing to false positives (Simmons et al., 2011) but almost certainly to false negatives as well. It is thus essential to address this problem in a thoughtful manner. Fortunately, guidelines exist in this regard. Yet, especially in the field of psychology, many researchers still do not treat outliers in a consistent manner or do so using inappropriate strategies (Leys et al., 2013; Simmons et al., 2011).

One possible reason is that researchers do not know of existing recommendations or currently available software options for their implementation. In this paper, we show how to follow current recommendations for statistical outlier detection (SOD) using R and the {performance} package (Lüdecke et al., 2021) from the *easystats* ecosystem.

3 Identifying Outliers

Although many researchers attempt to identify outliers with measures based on the mean (e.g., z scores), those methods are problematic because the mean and standard deviation themselves are not robust to the influence of outliers and they assume a normal distribution. Therefore, current guidelines recommend using robust methods to identify outliers, such as those relying on the median as opposed to the mean (Leys et al., 2013, 2018, 2019).

Nonetheless, which exact outlier method to use depends on several factors, like the statistical test of interest. When using a regression model, for example, the most relevant information will be regarding observations that do not fit well with the model, known as model-based outliers. When no method is readily available to detect model-based outliers, such as for structural equation modelling (SEM), looking for multivariate outliers may be of relevance. Finally, for simple tests (t tests or correlations) that compare values of the same variable, it can make sense to check for univariate outliers. However, univariate methods can give false positives since t tests and correlations, ultimately, are also models/multivariable statistics [REF?]. They are in this sense more limited, but we show them nonetheless for educational purposes. In the following section, we will go through each of the mentioned methods.

Regardless of the method used, we remind readers that transparency is key (Leys et al., 2019). Researchers should commit (ideally in a preregistration) to an outlier decision tree before collecting the data. They should report in the paper their decisions and details of their methods, as well as any deviation from their original plan. These transparency practices can help reduce false positives due to excessive researchers' degrees of freedom (choice flexibility throughout the analysis).

3.1 Univariate Outliers

For univariate outliers, it is recommended to use the median along the Median Absolute Deviation (MAD), which is more robust than the interquartile range or the mean and its standard deviation (Leys et al., 2013,

2019). The MAD can be calculated as follow:

$$MAD = bM_i(|x_i - M_j(x_j)|)$$

Where b is a scaler, often set to $1/\left(\Phi^{-1}(3/4)\right) \approx 1.4826$.

In {performance}'s `check_outliers()`, one can use this approach with `method = "zscore_robust"`. Although Leys et al. (2013) suggest a default threshold of 2.5 and Leys et al. (2019) a threshold of 3, {performance} uses a less conservative default threshold of ~3.09 by default.[1] That is, data points will be flagged as outliers if they go beyond +/- ~3.09 MAD. Users can adjust this threshold using the `threshold` argument.

Example:

```
library(performance)

# create some fake outliers and an ID column
data <- rbind(mtcars[1:4], 42, 55)
data <- cbind(car = row.names(data), data)

x <- check_outliers(data, method = "zscore_robust", ID = "car")
x
```

```
#> 2 outliers detected: cases 33, 34.
#> - Based on the following method and threshold: zscore_robust (3.09).
#> - For variables: mpg, cyl, disp, hp.
#>
#> ----------------------------------------------------------------------------
#> The following observations were considered outliers for two or more variables
#> by at least one of the selected methods:
#>
#>   Row car n_Zscore_robust
#> 1  33  33               2
#> 2  34  34               2
#>
#> ----------------------------------------------------------------------------
#> Outliers per variable (zscore_robust):
#>
#> $mpg
#>    Row car Distance_Zscore_robust
#> 33  33  33               3.709699
#> 34  34  34               5.848328
#>
#> $cyl
#>    Row car Distance_Zscore_robust
#> 33  33  33               12.14083
#> 34  34  34               16.52502
```

Specific outliers can be obtained by using `which()` on the output object, which can be used for exclusions for example:

```
which(x)
```

```
#> [1] 33 34
```

[1] 3.09 is an approximation of the critical value for $p < .001$, obtained through `qnorm(.999)`. We chose this threshold for consistency with the thresholds of all our other methods.

```
data2 <- data[-which(x), ]
```

All `check_outliers()` output objects possess a `plot()` method, meaning it is also possible to visualize the outliers:

```
library(see)
plot(x)
```

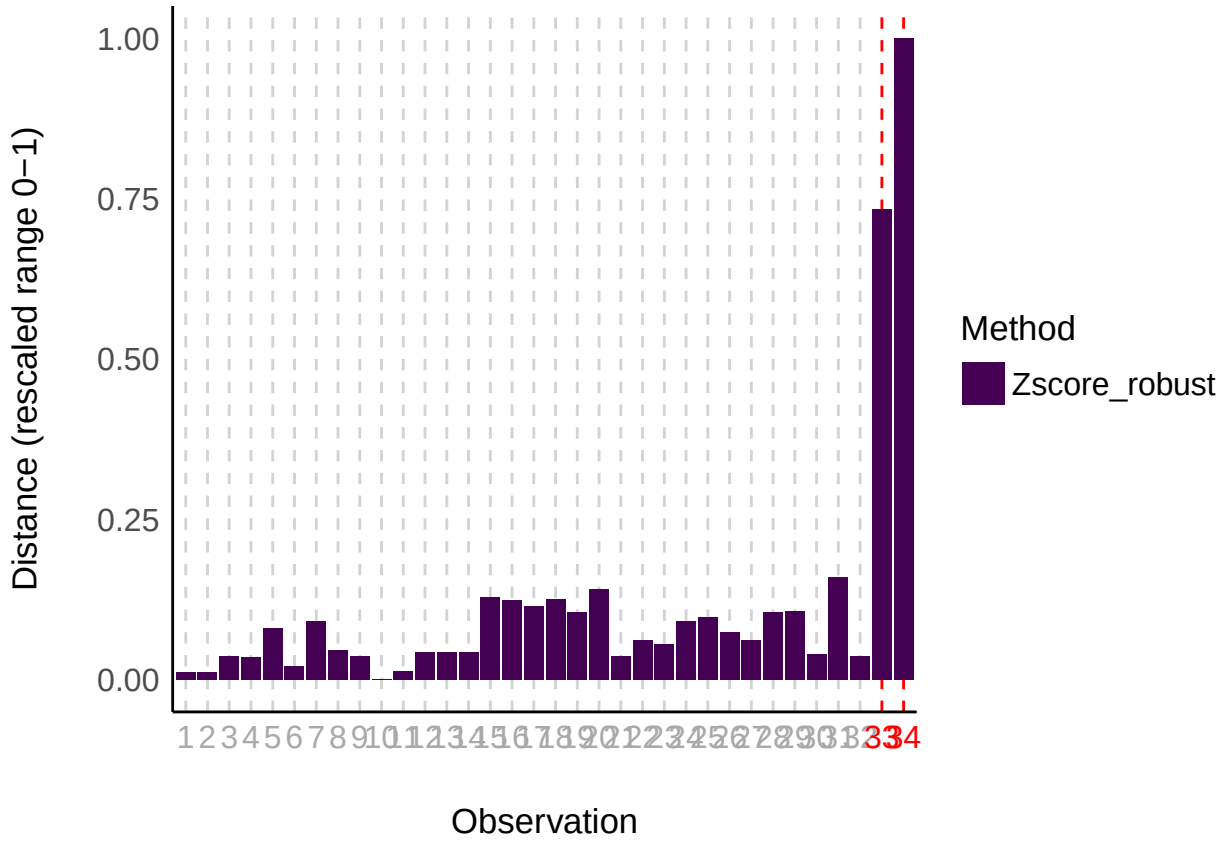

Figure 1: Visual depiction of outliers using the robust z-score method.

3.2 Multivariate Outliers

For multivariate outliers, it is recommended to use the Minimum Covariance Determinant, a robust version of the Mahalanobis distance (MCD, Leys et al., 2018, 2019). The MCD can be calculated as follow (Hubert et al., 2018):

$$MATTHS_{gohere}^{andB}$$

[mattansb can you add some maths here; perhaps from Hubert et al. (2018)? https://doi.org/10.1002/wics.1421]

In {performance}'s `check_outliers()`, one can use this approach with `method = "mcd"`.[2]

Example:

[2]Our default threshold for the MCD method is defined by `stats::qchisq(p = 1 - 0.001, df = ncol(x))`, which again is an approximation of the critical value for $p < .001$ consistent with the thresholds of our other methods.

```
x <- check_outliers(data, method = "mcd")
x
```

```
#> 9 outliers detected: cases 7, 15, 16, 17, 24, 29, 31, 33, 34.
#> - Based on the following method and threshold: mcd (18.47).
#> - For variables: mpg, cyl, disp, hp.
```

```
plot(x)
```

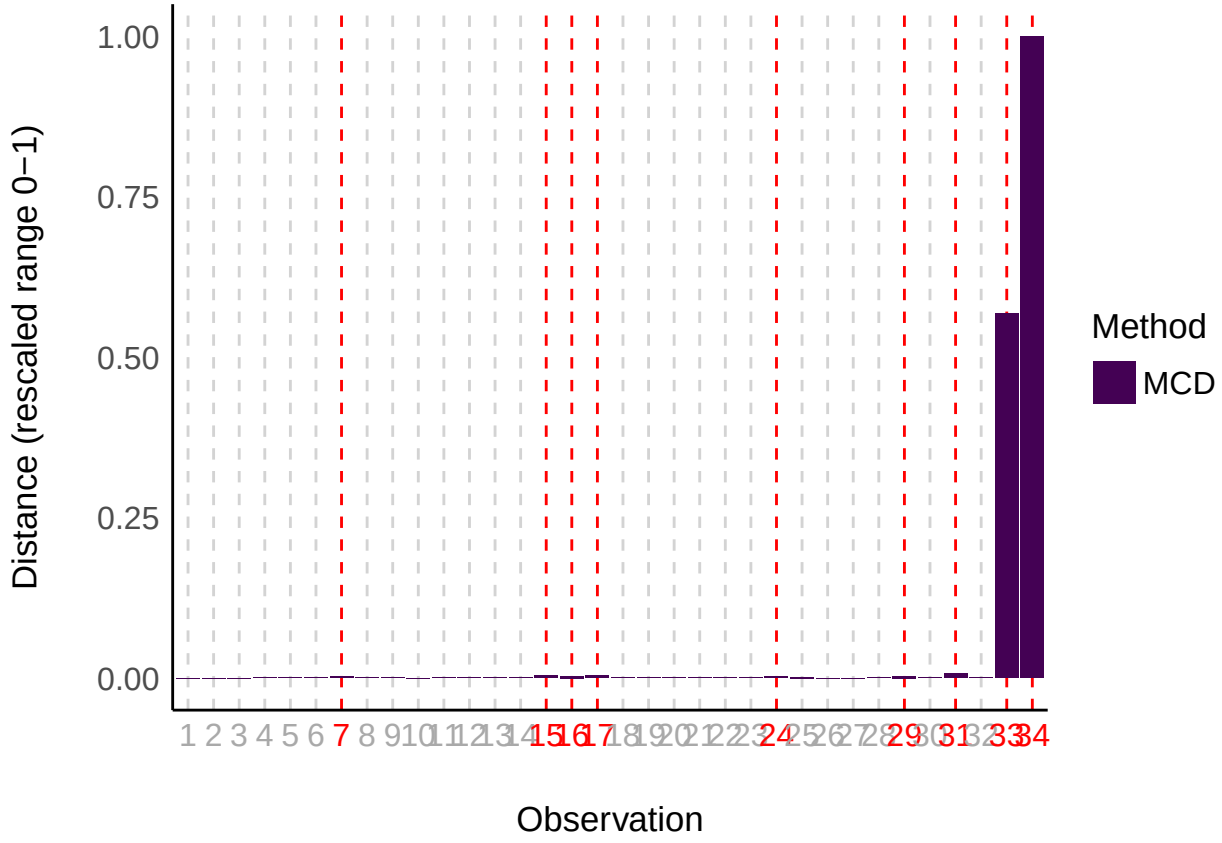

Figure 2: Visual depiction of outliers using the Minimum Covariance Determinant (MCD) method, a robust version of the Mahalanobis distance.

3.3 Model-Based Outliers

Whenever possible when working with regression models, we recommend using model-based SOD methods. These methods rely on the concept of leverage, or how much influence a given observation can have on the whole model. If few observations have a relatively strong leverage/influence on our model, then we can be confident that they are probably biasing our estimates. Thus, we can safely flag them as outliers. The Cook's distance, an example of such a model-based SOD, can be calculated as follow:

$$MATTHS_{gohere}^{andB}$$

[mattansb can you add some maths here?]

When working with regression models, model-based outliers can be detected using `check_outliers()` by

specifying `method = "cook"` (or `method = "pareto"` for Bayesian models).[3]

Example:

```
model <- lm(disp ~ mpg * hp, data = data)
x <- check_outliers(model, method = "cook")
x
```

```
#> 2 outliers detected: cases 31, 34.
#> - Based on the following method and threshold: cook (0.86).
#> - For variable: (Whole model).
```

```
plot(x)
```

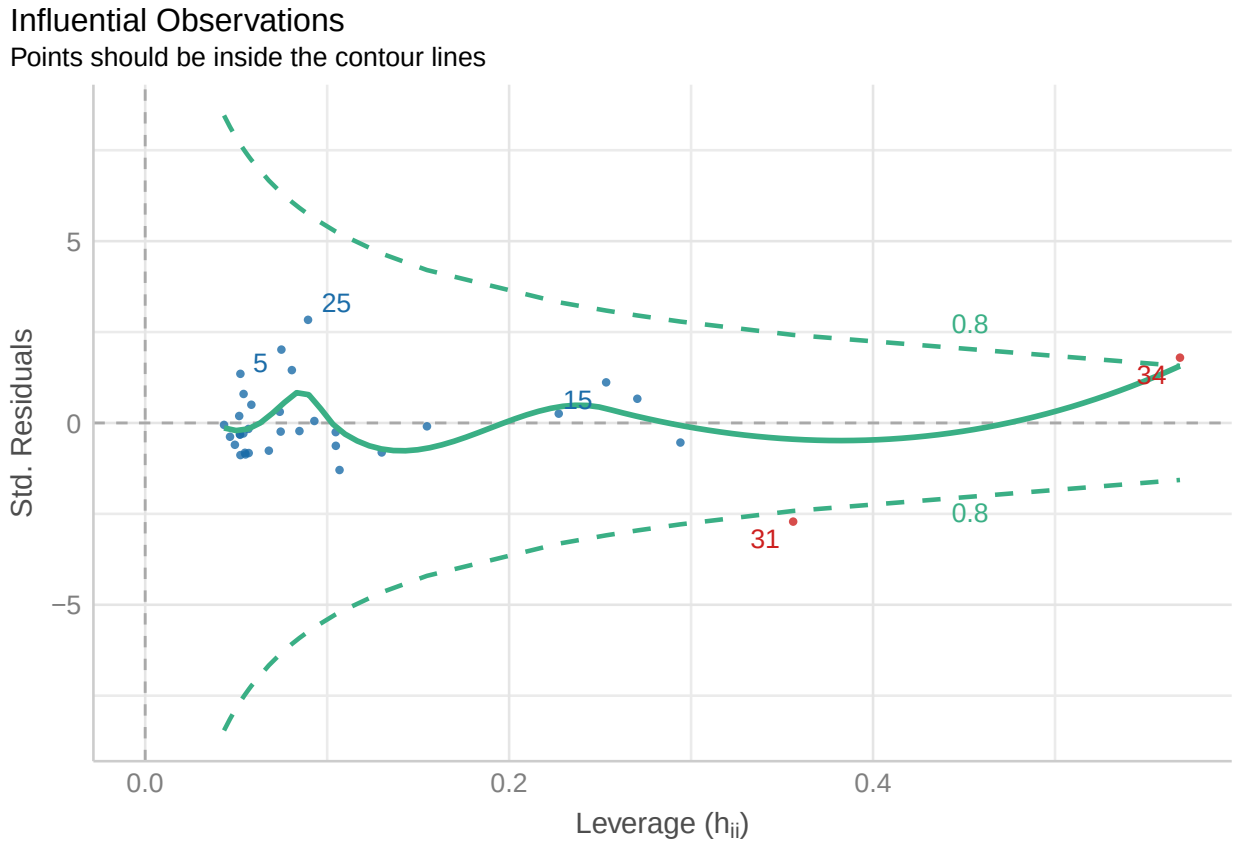

Figure 3: Visual depiction of outliers based on Cook's distance (leverage and standardized residuals).

3.3.1 Cook's Distance vs. MCD

Leys et al. (2018) write that they favour the MCD method over Cook's distance. This is because Cook's distance removes one observation at a time and checks its corresponding influence (leverage) on the model each time (Cook, 1977). The method flags as outliers any observation that has a large influence. In the view of these authors, when there are several outliers, the process of removing a single outlier at a time is problematic because the model remains "contaminated" or influenced by other possible outliers in the model. Therefore the method is not robust to the presence of multiple outliers.

[3]Our default threshold for the Cook method is defined by `stats::qf(0.5, ncol(x), nrow(x) - ncol(x))`, which again is an approximation of the critical value for $p < .001$ consistent with the thresholds of our other methods.

Although this is true, we believe that model-based methods are still preferable to the MCD when possible. In our view, our SOD methods should not be agnostic to the results or to our theoretical and statistical model of interest. For example, a very tall person would be expected to also be much heavier than average, but that would still fit with the expected relationship between height and weight. In contrast, a multivariate outlier detection method may flag this person as being an outlier even though the pattern fits perfectly with our predictions, because that person is unusual on two variables, height and weight.

Furthermore, unusual observations happen naturally: there are always extreme observations even in a normal distribution. We believe our models should reflect that, and that multivariate outlier methods are too conservative, in the sense that they will tend to flag too many outliers even if they belong to the proper distribution. If the presence of multiple outliers is a non-trivial concern, we recommend using robust regression methods instead, like t regression or quantile regression.

3.4 Multiple Methods

An alternative approach suggested by *easystats* is to combine several methods. This approach computes a composite outlier score, formed of the average of the binary (0 or 1) results of each method. It represents the probability that each observation is classified as an outlier by at least one method. The default decision rule classifies rows with composite outlier scores superior or equal to 0.5 as outlier observations (i.e., that were classified as outliers by at least half of the methods). In {performance}'s `check_outliers()`, one can use this approach by including all desired methods in the corresponding argument.

Example:

```
x <- check_outliers(
  data,
  method = c("zscore_robust", "iqr", "mcd", "ics"),
  ID = "car"
)
x
```

```
#> 3 outliers detected: cases 31, 33, 34.
#> - Based on the following methods and thresholds: zscore_robust (3.09),
#>   iqr (1.7), mcd (18.47), ics (0).
#> - For variables: mpg, cyl, disp, hp.
#>
#> Note: Outliers were classified as such by at least half of the selected methods.
#>
#> ----------------------------------------------------------------------------
#> The following observations were considered outliers for two or more variables
#> by at least one of the selected methods:
#>
#>   Row                 car n_Zscore_robust n_IQR          n_MCD          n_ICS
#> 1  33                  33               2     2 (Multivariate) (Multivariate)
#> 2  34                  34               2     2 (Multivariate) (Multivariate)
#> 3  31       Maserati Bora               0     1 (Multivariate) (Multivariate)
#> 4   7          Duster 360               0     0 (Multivariate)              0
#> 5  15  Cadillac Fleetwood               0     0 (Multivariate)              0
#> 6  16 Lincoln Continental               0     0 (Multivariate)              0
#> 7  17   Chrysler Imperial               0     0 (Multivariate)              0
#> 8  24          Camaro Z28               0     0 (Multivariate)              0
#> 9  29      Ford Pantera L               0     0 (Multivariate)              0
```

```
plot(x)
```

Outliers (counts or per variables) for individual methods can then be obtained through attributes. For example:

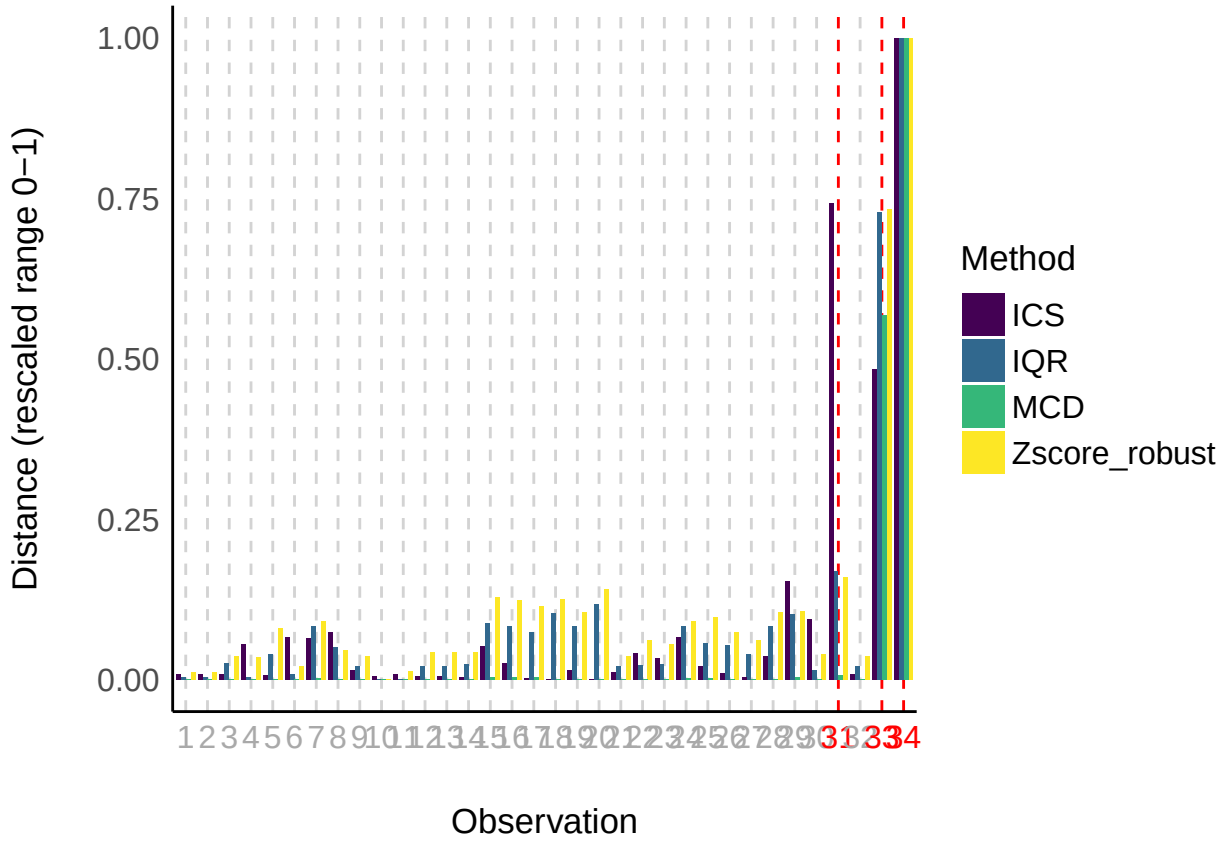

Figure 4: Visual depiction of outliers using several different statistical outlier detection methods.

```
attributes(x)$outlier_var$zscore_robust
```

```
#> $mpg
#>    Row car Distance_Zscore_robust
#> 33  33  33               3.709699
#> 34  34  34               5.848328
#>
#> $cyl
#>    Row car Distance_Zscore_robust
#> 33  33  33               12.14083
#> 34  34  34               16.52502
```

```
attributes(x)$outlier_var$iqr
```

```
#> $mpg
#>    Row car Distance_IQR
#> 33  33  33     1.550881
#> 34  34  34     2.458544
#>
#> $cyl
#>    Row car Distance_IQR
#> 33  33  33     5.294118
#> 34  34  34     7.205882
#>
#> $hp
#>               Row           car Distance_IQR
#> Maserati Bora  31 Maserati Bora     1.348181
```

```
attributes(x)$outlier_count$mcd
```

```
#>                     Row                 car          n_MCD
#> Duster 360            7          Duster 360 (Multivariate)
#> Cadillac Fleetwood   15  Cadillac Fleetwood (Multivariate)
#> Lincoln Continental  16 Lincoln Continental (Multivariate)
#> Chrysler Imperial    17   Chrysler Imperial (Multivariate)
#> Camaro Z28           24          Camaro Z28 (Multivariate)
#> Ford Pantera L       29      Ford Pantera L (Multivariate)
#> Maserati Bora        31       Maserati Bora (Multivariate)
#> 33                   33                  33 (Multivariate)
#> 34                   34                  34 (Multivariate)
```

An example sentence for reporting the usage of the composite method could be:

> Based on a composite outlier score (see the 'check_outliers' function in the 'performance' R package, Lüdecke et al., 2021) obtained via the joint application of multiple outliers detection algorithms ((a) median absolute deviation (MAD)-based robust z-scores, Leys et al., 2013; (b) interquartile range (IQR), (c) Mahalanobis minimum covariance determinant (MCD), Leys et al., 2019; and (d) invariant coordinate selection (ICS), Archimbaud et al., 2018), we excluded three participants that were classified as outliers by at least half of the methods used.

4 Handling Outliers

We have at this point demonstrated how to identify outliers. But what should we do with these outliers once identified? Although it is common to automatically discard any observation that has been marked as "an outlier" as if it might infect the rest of the data with its statistical ailment, we believe that the use of SOD methods is but one step in the get-to-know-your-data pipeline; a researcher or analyst's *domain knowledge*

must be involved in the decision of how to deal with observations marked as outliers by means of SOD. Indeed, automatic tools can help detect outliers, but they are not perfect. Although they can be useful to flag suspect data, they can have misses and false alarms, and they cannot replace human eyes and proper vigilance from the researcher.

4.1 Error, Interesting, and Random Outliers

Leys et al. (2019) distinguish between error outliers, interesting outliers, and random outliers.

Error outliers are likely due to human error and should be corrected before data analysis or outright removed since they are invalid observations. *Interesting outliers* are not due to technical error and may be of theoretical interest; it might thus be relevant to investigate them further even though they should be removed from the current analysis of interest. *Random outliers* are assumed to be due to chance alone and to belong to the correct distribution and, therefore, should be retained.

It is recommended to *keep* observations which are expected to be part of the distribution of interest, even if they are outliers (Leys et al., 2019). However, if it is suspected that the outliers belong to an alternative distribution, then those observations could have a large impact on the results and call into question their robustness, especially if significance is conditional on their inclusion.

On the other hand, there are also outliers that cannot be detected by statistical tools, but should be found and removed. For example, if we're studying the effects of X on Y among teenagers and we have one observation from a 20-year-old, this observation might not be a *statistical outlier*, but it is an outlier in the *context* of our research, and should be discarded to allow for valid inferences of interest.

Removing outliers can in this case be a valid strategy, and ideally one would report results with and without outliers to see the extent of their impact on results. This approach however can reduce statistical power. Therefore, some propose a *recoding* approach, namely, winsorization: bringing outliers back within acceptable limits (e.g., 3 MAD. Tukey & McLaughlin, 1963). However, if possible, it is recommended to collect enough data so that even after removing outliers, there is still sufficient statistical power without having to resort to winsorization (Leys et al., 2019).

4.2 Winsorization

Above, we mentioned a recoding approach to handling outliers: winsorization (Tukey & McLaughlin, 1963). The *easystats* ecosystem makes it easy to incorporate this step into your workflow through the `winsorize()` function of the {datawizard} package (Patil et al., 2022). This procedure will bring back univariate outliers within the limits of 'acceptable' values, based either on the percentile, the z score, or, ideally, the robust z score (based on the MAD).

Example:

```
data[33:34, ]
```

```
#>    car mpg cyl disp hp
#> 33  33  42  42   42 42
#> 34  34  55  55   55 55
```

```
# winsorizing using the MAD
library(datawizard)
winsorized.data <- winsorize(data, method = "zscore", robust = TRUE, threshold = 3)

winsorized.data[33:34, ]
```

```
#>    car      mpg     cyl disp hp
#> 33  33 37.68598 14.8956   42 42
#> 34  34 37.68598 14.8956   55 55
```

4.3 The Importance of Transparency

We note that no matter which outlier method you use, the handling of outliers should be specified *a priori* with as much detail as possible, and ideally preregistered, to limit researchers' degrees of freedom and therefore risks of false positives (Leys et al., 2019). This is especially true given that interesting outliers and random outliers are oftentimes hard to distinguish in practice. Thus, researchers should always prioritize transparency and report all of the following information: (a) how many outliers were identified; (b) according to which method and criteria, (c) using which function which R package (if applicable), and (d) how they were handled (excluded or winsorized, if the latter, using what threshold). If at all possible, (e) the corresponding code script along with the data should be shared on a public repository like the Open Science Framework, so that the exclusion criteria can be reproduced precisely.

5 Conclusion

In this paper, we have showed how to investigate outliers using the `check_outliers()` function of the {performance} package while following current good practices. We note that in addition to using the current functions and respecting existing recommendations, it is also important to pre-specify your plans to manage outliers, such as with preregistration, or at the very least justify your choices and stay consistent. Ideally, one would additionally also report the package, function, and threshold used (ideally linking to the full code). We hope that this paper will help more researchers engage in good research practices while providing a smooth outlier detection experience.

6 Acknowledgments

{performance} is part of the collaborative *easystats* ecosystem. Thus, we thank all members of easystats, contributors, and users alike.

References

Archimbaud, A., Nordhausen, K., & Ruiz-Gazen, A. (2018). ICS for multivariate outlier detection with application to quality control. *Computational Statistics and Data Analysis*, *128*, 184–199. https://doi.org/10.1016/j.csda.2018.06.011

Cook, R. D. (1977). Detection of influential observation in linear regression. *Technometrics*, *19*(1), 15–18. https://doi.org/10.1080/00401706.1977.10489493

Hubert, M., Debruyne, M., & Rousseeuw, P. J. (2018). Minimum covariance determinant and extensions. *Wiley Interdisciplinary Reviews: Computational Statistics*, *10*(3), e1421. https://doi.org/10.1002/wics.1421

Leys, C., Delacre, M., Mora, Y. L., Lakens, D., & Ley, C. (2019). How to classify, detect, and manage univariate and multivariate outliers, with emphasis on pre-registration. *International Review of Social Psychology*. https://doi.org/10.5334/irsp.289

Leys, C., Klein, O., Dominicy, Y., & Ley, C. (2018). Detecting multivariate outliers: Use a robust variant of the mahalanobis distance. *Journal of Experimental Social Psychology*, *74*, 150–156. https://doi.org/https://doi.org/10.1016/j.jesp.2017.09.011

Leys, C., Ley, C., Klein, O., Bernard, P., & Licata, L. (2013). Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median. *Journal of Experimental Social Psychology*, *49*(4), 764–766. https://doi.org/10.1016/j.jesp.2013.03.013

Lüdecke, D., Ben-Shachar, M. S., Patil, I., Waggoner, P., & Makowski, D. (2021). performance: An r package for assessment, comparison and testing of statistical models. *Journal of Open Source Software*, *6*(60), 3139. https://doi.org/10.21105/joss.03139

Patil, I., Makowski, D., Ben-Shachar, M. S., Wiernik, B. M., Bacher, E., & Lüdecke, D. (2022). datawizard: An R package for easy data preparation and statistical transformations. *Journal of Open Source Software*, *7*(78), 4684. https://doi.org/10.21105/joss.04684

Simmons, J. P., Nelson, L. D., & Simonsohn, U. (2011). False-positive psychology: Undisclosed flexibility

in data collection and analysis allows presenting anything as significant. *Psychological Science*, *22*(11), 1359–1366. https://doi.org/10.1177/0956797611417632

Tukey, J. W., & McLaughlin, D. H. (1963). Less vulnerable confidence and significance procedures for location based on a single sample: Trimming/winsorization 1. *Sankhyā: The Indian Journal of Statistics, Series A*, 331–352.