Lineare Regression

Walter Gruber

2025-05-07

Contents

1

Vorwort **1**

Einführung **1**

PART I: GLM-Assumptions **2**

Critical Assumptions **2**

Normality **4**
Contaminated Normal distributions 4
Issues with normality 6

Myth and misconceptions about robustness **8**
Are GLM's robust? 8
Validity of central limit theorem 8

PART II: Robust Methods **9**

Definition of Robust **9**

Solutions to violated assumptions **9**
Alternative estimators 9
Transformation of data 10
Nonparametric tests 10
Robust methods 10
Adjusting the standard errors 11

Selected Robust measures of location **11**
Trimmed mean 11
Winsorized mean 12
M-estimators 13

Measures of Scale **13**

PART III: R-Examples **14**

R-Packages **14**
Other Packages in R . . . 15

Two independent means **15**
Standard t-Test . . . 16
Robust t-Test . . . 17

One-way ANOVA **17**
Standard ANOVA . . . 17
Robust One-way ANOVA . . . 18

Two dependent means **19**
Check Assumptions . . . 20
Standard t-Test . . . 21
Robust t-Test . . . 22

One-way RM-ANOVA **23**
Check Assumptions . . . 23
Standard RM-ANOVA . . . 24
Robust ANOVA . . . 25

Regression Models **26**
Check Data . . . 26
Standard Linear Model . . . 27
Robust Linear Model (RLM) . . . 29

PART IV: Summary **32**

Conclusions and recommendations **32**
Recommendations . . . 33

Assumptions are made
and
most assumptions are wrong

- Albert Einstein -

Vorwort

Dieses Skriptum basiert (großteils) auf Literatur von Andy Field und Rand Wilcox (&. W. Field A. P. 2017), David Erceg-Hurn et.al. (Hurn 2008), Mair (Mair 2020) and Wilcox (Wilcox 2012). Teile der hier verwendeten Inhalte wurden unverändert aus der angegebenen Literatur übernommen.

Einführung

Kapitel 1: Einführung in die Einfache Lineare Regression

In einer Welt, die zunehmend von Daten geprägt ist, sind statistische Methoden unverzichtbare Werkzeuge, um Muster aufzudecken und fundierte Entscheidungen zu treffen. Die einfache lineare Regression ist eine der grundlegendsten, aber zugleich wirkungsvollsten Techniken in der Statistik. Sie ermöglicht es, den Zusammenhang zwischen zwei quantitativen Variablen zu modellieren und vorherzusagen, wie sich Änderungen in einer unabhängigen Variable auf eine abhängige Variable auswirken.

In diesem Kapitel werden wir die Grundlagen der einfachen linearen Regression erkunden:

- Was bedeutet es, einen linearen Zusammenhang zwischen zwei Variablen zu postulieren?
- Wie wird ein lineares Regressionsmodell aufgestellt und interpretiert?

Durch anschauliche Beispiele und Schritt-für-Schritt-Anleitungen (in R) werden wir die Schlüsselkonzepte und mathematischen Grundlagen dieser Methode erläutern.

Ziel ist es, Ihnen ein solides Verständnis für die einfache lineare Regression zu vermitteln, das als Basis für komplexere statistische Analysen dient.

Figure 1: **Figure 1**: use StatParaPlast to solve all your nasty statistical problems. Look inside the package to disover the fantastic world on NNS (No Nonsense Statistic). Available in different sizes and colors.

PART I: GLM-Assumptions

Critical Assumptions

In many fields of psychological research, variants of the general linear model (*GLM*) are used. In this model, an outcome variable Y is predicted from a *linear* and *additive combination* of one or more predictor variables (*predictors*), i.e. $X_1, \cdots, X_n$.

For each predictor, a *parameter* ($\hat{b}_i$, *slope*) is estimated from the data. This parameter represents the relationship between the predictor and outcome variable if the effects of other predictors in the model are held constant[1].

There is also a parameter ($\hat{b}_0$, *constant/intercept*) to estimate the value of the outcome when all predictors are zero.

The errors in prediction are represented by the residuals (ε_i), which are (for each observation, i) the distance between the value of the outcome predicted by the model and the value observed in the data.

$$\hat{Y}_i = \hat{b}_0 + \hat{b}_1 \cdot X_{1i} + \cdots + \hat{b}_n \cdot X_{1n} + \varepsilon_i$$

There are two common ways to estimated the parameters $\hat{b}_k$ with $k \epsilon \{0, n\}$:

- using **O**rdinary **L**east **S**quares (OLS) estimation: which seeks to minimize the squared errors between the predicted
- **M**aximum **L**ikelihood (**ML**) estimation: which seeks to find the parameter values that maximize the likelihood of the observations.

  The OLS estimator will also be the maximum likelihood estimator (ML), when the assumptions of independent-, homoscedastic- and normally-distributed-errors are met! When these assumptions are not met, the ML estimator will yield different results to the OLS.

The **G**eneral **L**inear **M**odel (**GLM**) is a flexible framework through which to predict a continuous outcome variable from predictor variables that are/can be:

- continuous (*regression* or *multiple regression*)
- categorical (*ANOVA*) or
- both (*ANCOVA*), or *Moderation-Analysis*, when interactions are modelled.

Similarly, experimental designs containing repeated measures and longitudinal data are special cases of a **multilevel linear model** in which observations (level 1) are nested within participants (level 2). If we include terms that estimate the variance across contexts (e.g., time) in both the constant (ζ_{0j}) and model parameters (e.g., ζ_{1j}) for each predictor, we get a multilevel model in which observations (i) are nested within contexts (j). These contexts could be individuals (e.g., repeated measures designs) or environments (e.g., classrooms).

$$\hat{Y}_{ij} = \hat{b}_0 + \hat{b}_1 \cdot X_{ij} + (\zeta_{0i} + \zeta_{1j} \cdot X_{ij} + \varepsilon_{ij})$$

What we must not forget is that they are *all variants of the linear model* and, therefore, *have a common set of underlying assumptions!*. The two main assumptions are:

1. additivity and linearity

[1] trivial but worthwhile to note at this point: the parameters are estimated based on the observed data. As a consequence we must conclude, that if the data does not represent the outcome in the population, the parameters will NOT represent the relationship between the predictors and the *real world*.

2. spherical residuals (= independent[2] and homoscedastic[3])

   When applying this models we assume that the outcome variable is linearly related to any predictors and that the best description of the effect of several predictors is that their individual effects can be added together. As such, the **assumption of additivity and linearity is the most important** because it equates to the general linear model being the best description of the process of interest. When these conditions are met (and residuals have a mean of zero) then the linear model derived from OLS estimation will be a best linear unbiased[4] estimator. **If this assumption is not true then you are fitting the wrong model**.

To summarize:

1. When data is heteroscedastic, the formula for the variance of the parameters $\hat{b}_k$ is incorrect.
2. Consequently, the estimation of the Standard Error SE is incorrect.
3. Autocorrelation introduces further bias to SE.
4. Biased[5] standard errors have important consequences for significance tests and confidence intervals (CI's) of model parameters (i.e. CI's can be extremely inaccurate when SE is biased).

Normality

Before looking into different problems with normality we shall have a closer look at distributions which seem to be normal but aren't!

Contaminated Normal distributions

Before we dig into some details let us illustrate the effects of slight depatures form normality. Suppose we sample (an arbitrary variable) from the population of all adults. Within this sample about 10% are of age 70 and over. Let us assume, that the people younger than 70 have a mean $\bar{x}_Y = 0$ and a standard deviation of $sd_Y = 1$. The older people also show a mean of $\bar{x}_O = 0$, but there standard deviation is much higher, e.g. $sd_O = 10$. The following graph shows the standard normal distribution (*black*) vs. the mixed distribution of our example (*blue*). Again, only 10% of the data for this distribution has a different variance (*red*). Do you think these distributions can be considered identical, or at least both of them are normally distributed?

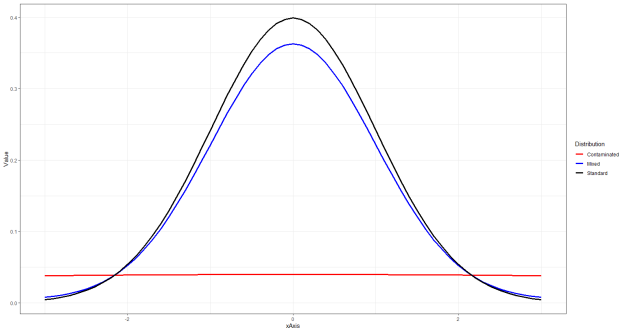


Figure 2: Comparison of different normal distributions

If we observe such results on representative samples, we could assume, that the entire population has a *mixed*, or *contaminated distribution* with $\varepsilon = 0.1$ and $\lambda = 10$. So, in a formal way we would define two independent distributions (X, Y) with different standard deviations and different sample sizes which are mixed in a distribution H with the variance of $Var(H)$, such that:

[2]Independent residuals are ones that are not correlated across observations. Correlation across residuals is known as *autocorrelation*.

[3]*Homoscedastic* residuals are ones that have the same variance for all observations. Residuals without this property are called *heteroscedastic*.

[4]**Unbiased** means that the estimator's expected value for a parameter matches the true value of that parameter.

[5]a test statistic is **biased** if the probability of rejecting the null is not minimized when the null is true.

$$
\begin{array}{rcl}
X & \approx & N(\mu = 0, sd = 1) \\
Y & \approx & N(\mu = 0, sd = \lambda) = N(\mu = 0, sd = 10) \\
H & = & (1 - \varepsilon) \cdot X + \varepsilon \cdot (x) = 0.9 \cdot X + 0.1 \cdot Y \\
\overline{H} & = & E(H) = 0 \\
Var(H) & = & E(H^2) \\
E(H^2) & = & E(0.9 \cdot X^2 + 0.1 \cdot Y^2 + 2 \cdot X \cdot Y \cdot 0.9 \cdot 0.1) \\
 & = & 0.9 \cdot \underbrace{E(X^2)}_{=1^2} + 0.1 \cdot \underbrace{E(Y^2)}_{=10^2} + 2 \cdot 0.9 \cdot 0.1 \cdot \underbrace{E(X \cdot Y)}_{=\text{Cov(X,Y)}=0} \\
 & = & 0.9 \cdot 1 + 0.1 \cdot 100 = 10.9
\end{array}
$$

If we look at the distribution plots, we notice that both (standard normal and contaminated normal) both look very similar. We could conclude, that the contaminated normal is approximating the standard normal distribution to such extent, that we can assume it is also a normal distribution.

```
xfrom  <- -3
xto    <- +3
Mu     <- 0
SD     <- 1
Epsilon  <- 0.1 # percentage of smaller sample with higher std
Lambda <- 10  # standard deviataion of smaller sample
# x      <- seq(from = xfrom * Lambda, to = xto * Lambda, by = 0.1)
x      <- seq(from = xfrom, to = xto, by = 0.1)
N      <- length(x)
Y1     <- NULL
Y2     <- NULL
# Y3     <- NULL
CN     <- NULL

for (i in 1:N) {
  Y1[i]    <- dnorm(x[i], mean = Mu, sd = SD) # std. normal component
  Y2[i]    <- dnorm(x[i], mean = Mu, sd = SD*Lambda) # contamination
  # Y2[i]    <- dnorm(x[i]/Lambda, mean = Mu, sd = SD) # contamination (same as above)
  CN[i] <- ((1 - Epsilon) * Y1[i]) + (Epsilon * Y2[i])
}
DF_CN  <- data.frame(xAxis = x, Standard = Y1, Contaminated = Y2, Mixed = CN)
  # probability density function (pdf) for one input z-value to
  # show the equivalence to the dnorm() function for the same parameters mean
  # and sd.
  # stddev <- 1
  # zwert  <- -3
  # Phi <- (1/sqrt(2*pi*stddev^2)) * exp(-0.5*(zwert/stddev)^2)
  # Y1[1] == Phi
# ggplot(DF_CN, aes(x = xAxis)) +
#   geom_line(aes(y = Standard), color = "red") +
#   geom_line(aes(y = Contaminated), color = "blue") +
#   geom_line(aes(y = Mixed), color="black", linetype="twodash")
#
# library(tidyverse)
DF_CN_P1 <- DF_CN %>%
            gather(key = "Distribution", value = "Value", -xAxis)
# head(DF_CN_P1)
P1 <- ggplot(DF_CN_P1, aes(x = xAxis, y = Value)) +
      # geom_line(aes(color = Distribution, linetype = Distribution), size = 1.3) +
```

Table 1: Results of the simulation. Note the huge difference in the standard and mixed distribution.

Distribution	Means	Variances
Contaminated	0.03	100.81
Mixed	-0.03	10.87
Standard	-0.01	1.00

```
      geom_line(aes(color = Distribution), size = 1.3) +
      scale_color_manual(values = c("red", "blue", "black")) +
      theme_bw()
DF_CN_P1 <- subset(DF_CN_P1, Distribution == "Standard" | Distribution == "Mixed")
P2 <- ggplot(DF_CN_P1, aes(x = xAxis, y = Value)) +
  geom_line(aes(color = Distribution), size = 1.3) +
  scale_color_manual(values = c("blue", "black")) +
  theme_bw()
```

Even so they look like almost identical, they are NOT! Notice the very slight difference at the left and right side (the tails) of the distribution. It might seem just a slight difference, but has in fact a quite dramatic impact on the variance. The variance for the standard normal (black) is 1, whereas for the combined normal (red) the variance is at 10.9 (c.f. to our results above)! We can simulate this effect quite easy - just copy and paste the following code and compare the results:

```
NSim <- 50000
set.seed(143)
Y3 <- rnorm(NSim, mean = Mu, sd = SD)
set.seed(3143)
Y4 <- rnorm(NSim, mean = Mu, sd = SD * Lambda)
z  <- sample(c(0,1),
             size    = NSim,
             prob    = c(1-Epsilon, Epsilon),
             replace = TRUE)
Hx            <- Y3 * (1-z) + Y4 * z
DF_Sim        <- data.frame(Standard = Y3, Contaminated = Y4, Mixed = Hx)
DF_Sim_L      <- DF_Sim %>% gather(key = "Distribution", value = "Value")
Res           <- summary_by(DF_Sim_L,Value ~ Distribution, FUN = c(mean, var))
colnames(Res) <- c("Distribution", "Means", "Variances")
Res[,2:3]     <- round(Res[,2:3], 2)
knitr::kable(Res, caption = "Results of the simulation. Note the huge difference in the standard and m
```

> In essence, a small proportion of the population of participants can have an inordinately large effect on its value.

A look at the Q-Q-plot reveals the problem of the mixed distribution in a quite impressive way:

Issues with normality

There are three issues related to normality:

1. normality of residuals (ε_i)
2. Normal distribution of test statistics
3. Confidence intervals

Residuals

The well known (simple) regression model in its mathematical form is defined as:

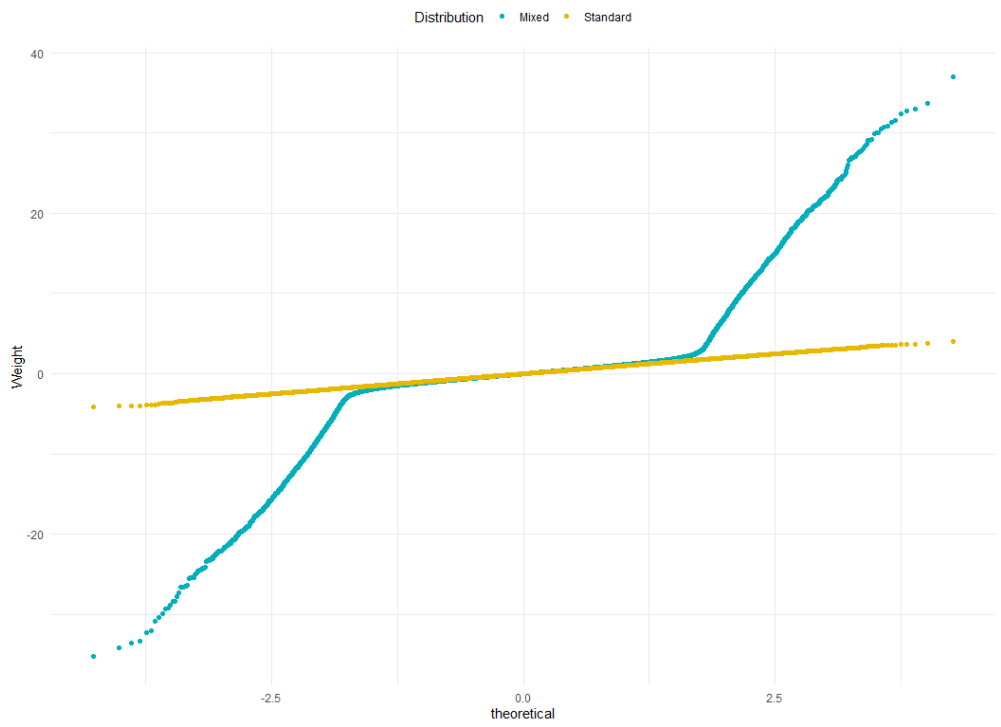


Figure 3: Q-Q-Plot of standard vs. mixed normal (later is also reffered to as contaminated normal)

$$Y_i = b_0 + X_i \cdot b_1 + \varepsilon_i \tag{1}$$

Each case (i) of data (X_i) has a residual ε_i, which is the difference between the predicted and observed values of the outcome. If you inspected a histogram of these residuals for all cases, you would hope to see a normal distribution centered around 0, i.e. $\bar{\varepsilon}_i = 0$.

A mean residual of $\bar{\varepsilon}_i = 0$ means that the model (on average) correctly predicts the outcome value. In other words, if the residual is zero (or close to it) for most cases, then the error in prediction is zero (or close to it) for most cases.

If the model fits well, we might also expect that very extreme over-, or underestimations occur rarely. A well fitting model then would yield residuals that, like a normal distribution, are most frequent around zero and very infrequent at extreme values. This description explains what we mean by normality of residuals.

> It also gives an idea of what it means if the normality of residuals is not given!

Test statistics

The p-values associated with the parameter estimates of the model are based on the assumption that the test statistic associated with them follows a normal distribution (or some variant of it such as the t-distribution).

Essentially, to test the hypothesis that the parameter estimate ($\bar{x}, r, \hat{b}_k$, etc.) is not equal to 0 ($= H_1$) it is necessary to assume a particular shape for the null distribution of the test statistic (i.e., normal).

If the sampling distribution of the test statistic turns out not to be the assumed shape (i.e. normal) then the resulting p-values will be incorrect.

> As a reminder: the p-value tells us about the probability to observe such data as our sample, IF the Null-Hyothesis (H_0) is True, i.e. $p(D|H_0)$! For the H_0 we (often) assume a normal distribution (or a deriviate of a normal, such as χ^2, F, t).

Confidence intervalls

The bounds of confidence intervals for parameter estimates are constructed by adding or subtracting from the estimate the associated standard error multiplied by the quantile of a null distribution associated with the probability level assigned to the interval.

$$\begin{aligned} CI_{Mean} &= \bar{x} \pm SE_{Mean} \cdot t_{crit} \\ CI_{Corr} &= r \pm SE_{Corr} \cdot t_{crit} \\ CI_{Coef} &= \hat{b}_k \pm SE_{Coeff} \cdot t_{crit} \\ CI_{...} &= \cdots \pm SE_{...} \cdot t_{crit} \end{aligned}$$

For tests of parameters in the linear model, the null distribution is assumed to be normal. It is an example of a general strategy in inferential statistics to convert an estimator, such as the mean, into a standardized statistic (z) that is asymptotically standard normal. The general issue is one of determining under what circumstances assuming normality gives a reasonably accurate result.

Myth and misconceptions about robustness

> A common claim, based on the *central limit theorem*, is that with sample sizes greater than 30 the parameter estimate will have a normal sampling distribution. The implication being that if our sample is large we need not worry about checking normality to know that confidence intervals and p-values for a parameter estimate will be accurate. In which case, we can effectively ignore normality in all but quite exceptional cases of fitting a linear model.

However, two things were missed when arriving at this conclusion:

1. the conclusion is based on work using very light-tailed distributions.

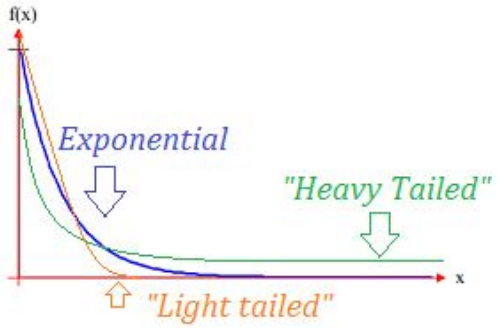


Figure 4: Tails of a distribution

2. the assumption that Student's T performs well if the sample mean has (to a close approximation) a normal distribution turns out to be incorrect under general conditions (Wilcox, 2016, 2017).

Are GLM's robust?

Recent investigations revealed that differences in:

- *skewness*
- *non-normality* and
- *heteroscedasticity*

interact in complicated ways that impact power (Wilcox, 2017). It was believed that as *kurtosis* increases, the Type I error rate decreases and quickly drops below its nominal $\alpha = .050$ level, and consequently power decreases (Glass et al., 1972).

> This conclusion is correct only if *distributions have the same amount of skewness*, because in this situation the difference between variables will have a *symmetric distribution*.

Unequal variances (violations of *homoscedasticity*), have relatively little influence when:

- group sizes are equal and
- the normality assumption is true

But when group sizes are unequal F varies as a function of whether the largest group has the smallest variance or vice versa (see Field, Miles, & Field, 2012, for a review).

> When normality cannot be assumed equal group sizes do not save F from violations of homoscedasticity (Wilcox, 2010, 2016, 2017).

Validity of central limit theorem

When distributions are symmetric and have light tails the sampling distribution of means is approximately normal using samples of only $N \approx 20$.

When distributions are asymmetric (skewed), even light-tailed distributions can require sample sizes of $N \approx 200$ (e.g. for one-sample t-tests).

Heteroscedasticity, makes matters worse. When distributions have heavy tails samples need to be much larger (up to $N \approx 160$) before the sampling distribution is normal (Wilcox, 2010).

> As such, researchers can be lured into a false sense of security that they can assume normality of the sampling distribution because of the central limit theorem.

PART II: Robust Methods

Definition of Robust

If we look up Wikipedia we find the following description of Robust Statistic:

Robust statistics are statistics with good performance for data drawn from a wide range of probability distributions, especially for distributions that are not normal.

Robust statistical methods have been developed for many common problems, such as estimating location, scale, and regression parameters. One motivation is to produce statistical methods that are not unduly affected by outliers.

Another motivation is to provide methods with good performance when there are small departures from parametric distribution. For example, robust methods work well for mixtures of two normal distributions with different standard-deviations; under this model, non-robust methods like a t-test work poorly. Another definition of robust measures frequently used is:

> Measures that characterize a distribution (such as location and scale) are said to be *robust*, if slight changes in a distribution have a relatively small effect on their value (Wilcox 2012) (page 23).

The mathematical foundation of robust methods (dealing with quantitative, qualitative and infnitesimal robustness of parameters) makes no assumptions regarding the functional form of the probability distribution.

The basic trick is to view parameters as functionals; expressions for the standard error follow from the influence function. Robust inferential methods are available that perform well with relatively small sample sizes, even in situations where classic methods based on means and variances perform poorly with relatively large sample sizes. Modern robust methods have the potential of substantially increasing power even under

slight departures from normality. And perhaps more importantly, they can provide a deeper, more accurate and more nuanced understanding of data compared to classic techniques based on means.

Solutions to violated assumptions

The consequences of normality deviations such as:

- skewed distributions
- data with outliers, or
- heavy-tailed distributions

can influence the results of any classical (i.e. parametric) statistical analysis quite substantially. Seen from a purely descriptive angle, it is trivial that the mean can be heavily affected by outliers or highly skewed distributional shapes.

Alternative estimators

Computing the mean on "ugly" data is just not a good location measure to characterize the sample. In this case one strategy is to use more robust measures such as the *median* or the *trimmed mean*, *winsorized mean*, etc. and perform tests based on the corresponding sampling distribution of such robust measures.

It is quite common in experimental psychopathology research to do manual trims of the data based on outlier detection techniques (e.g., standard deviation based trims[6] or idiosyncratic deletion).

Other popular alternatives are the M-estimators. They determine whether a score is an outlier empirically and if it is, adjustments are made for it. The adjustment could be to completely ignore the observation or to down-weight it. Obvious advantages of M-estimators are that you can:

1. down-weight rather than exclude observations.
2. avoid over- or under-trimming your data.
3. perform non-symmetric trimming (Wilcox 2012)

When assumptions (independent, homoscedastic and normally-distributed errors) are not met, the ML estimator will yield different results to the OLS. The ML estimator is a lot more versatile than OLS and tends to be the default for more complex variants of the linear model (such as multilevel models, models with latent variables etc.).

Transformation of data

Another strategy to deal with such violations (especially with skewed data) is to apply transformations such as the *logarithm* or more sophisticated Box-Cox transformations. However, before working with transformed data, the following points should be considered (also see Wilcox 2012):

1. transformations seldom improve the validity of probability statements.
2. transforming changes the hypothesis being tested (log transformed means compare geometric, rather than arithmetic means).
3. transforming the data also transforms the construct that it measures, so interpretation might become difficult.
4. the consequences of applying the 'wrong' transformation must be less severe than the consequences of analyzing the untransformed scores.
5. heavy tails matter more than skew, so a transformation would need to address (and not make worse) any problems related to tail weight!
6. distributions often remain skewed after transformation.
7. transformations generally do not deal effectively with (real) outliers.

[6] e.g. Ratcliff (1993), with reaction time data to use standard deviation based trims such as excluding scores greater than 2.5 standard deviations from the mean. Be aware that this approach is flawed because both the mean and standard deviation are highly influenced by outliers (whether overt ones, or covert ones such as in a mixed normal distribution!

Nonparametric tests

Another option is to switch into the nonparametric testing world (Brunner E. 2002). Prominent examples for classical nonparametric tests taught in most introductory statistics class are:

- Mann-Whitney U-test (Mann and Whitney, 1947)
- Wilcoxon signed-rank and rank-sum test (Wilcoxon, 1945)
- Kruskal-Wallis ANOVA (Kruskal and Wallis 1952)
- Friedmann ANOVA (Friedmann, 1937)

Robust methods

Developments of robust methods can be traced back to the 1960's with publications by Tukey (1960), Huber (1964), and Hampel (1968).

Modern robust methods have the potential of:

- substantially increasing power even under slight departures from normality.
- can provide a deeper, more accurate and more nuanced understanding of data

compared to classic techniques based on means.

Adjusting the standard errors

There are ways to adjust standard errors to be robust in the presence of heteroscedasticity. One is known as the *Eiker-White-Huber* heteroscedasticity-consistent stranded errors. The resulting robust standard errors can be used to compute confidence intervals, test-statistics (and associated p-values) that are robust to heteroscedasticity.

Another way to deal with bias in standard errors and (confidence intervals) is to estimate them empirically. The *Bootstrap* is a flexible and general empirical method to find standard errors and confidence intervals for any statistic that is usually more accurate than traditional approaches. As with heteroscedasticityconsistent standard errors, bootstrap standard errors (and associated test statistics and p-values) and confidence intervals should be robust to violations of the assumptions.

Selected Robust measures of location

Measures of location are core elements of robust methods. Such measurese are:

- Trimmend mean
- Winsorized mean
- Huber M-estimator

Trimmed mean

The trimmed mean discards a certain percentage at both ends of the distribution. For instance, a 20% trimmed mean cuts off 20% at the low end and 20% the high end. In R, a trimmed mean can be computed via the basic mean function by setting the trim argument accordingly. The following code also shows in which way the trimmed mean is calculated when the argument `trim` of the mean function is set to a value > 0.

```
set.seed(423)
N       <- 20 # length of sample
M       <- 100 # Mean of sample
SD      <- 15  # standard deviation of sample
TF      <- 0.1 # Trim-Faktor
#=== Generate Data ===
IQ      <- rnorm(N, mean = M, sd = SD)
IQ[N]   <- 1240 # introduce outlier
```

Table 2: mean vs. trimmed mean

Mean	T_Mean	Man_T_Mean
159.085	103.143	103.143

Table 3: Standard deviations

SD	T_SD	WRS2_SD
56.988	3.292	4.295

```r
set.seed(2384)
#=== Manual trim ===
# IQ_Red  <- sort(na.omit(IQ)) # remove NA's and sort
IQ_Sort <- sort(IQ) # sort Vector
RedLgth <- N*TF
TInd    <- (RedLgth+1):(N - RedLgth)
IQ_Trim <- IQ_Sort[TInd]
#=== Means ===
M1      <- round(mean(IQ, na.rm = T), 3)
M2      <- round(mean(IQ, na.rm = T, trim = TF), 3)
M3      <- round(mean(IQ_Trim, na.rm = T), 3)
Means   <- data.frame(Mean = M1, T_Mean = M2, Man_T_Mean = M3)
knitr::kable(Means, booktabs = TRUE, caption = 'mean vs. trimmed mean')
```

```r
#=== SDs ===
SD1     <- round(sd(IQ) / sqrt(length(IQ)), 3)       # standard error
SD2     <- round(sd(IQ_Trim)/ sqrt(length(IQ_Trim)), 3) # standard error trimmed
SD3     <- round(trimse(IQ, tr = TF), 3)      # standard error from WRS2
StdDevs <- data.frame(SD = SD1, T_SD = SD2, WRS2_SD = SD3)
knitr::kable(StdDevs, booktabs = TRUE, caption = 'Standard deviations')
```

Note that if the trimming portion is set to $\gamma = 0.5$, the trimmed mean $\bar{x}_t$ results in the *median* $\tilde{x}$ (which by itself *reflects another robust location measure*).

Winsorized mean

A further robust location alternative to the mean is the *Winsorized mean*.

> The process of giving less weight to observations in the tails of the distribution and higher weight to the ones in the center is called *Winsorizing*.

Instead of computing the mean on the original distribution we compute the mean on the Winsorized distribution. Similar to the trimmed mean, the amount of Winsorizing (i.e., the *Winsorizing level*) has to be choosen a priori. The `WRS2` function to compute Windsorized means is called `winmean`.

There is also a function supplied in the `DescTools` Package, the `DescTools::Winsorize()` function. The following code and output shows some examples of these estimators:

```r
#=== Generate Data
timevec <- c(92, 19, 101, 58, 1053, 91, 26, 78, 10, 13, -40, 101,
             86, 85, 15, 89, 89, 28, -5, 41)
# timevec <- c( 77,  87,  88, 114, 151, 210, 219,  246, 253, 262,
#              296, 299, 306, 376, 428, 515, 666, 1310, 2611)
```

```r
timevec <- sort(timevec)
#=== Trimmed Mean and SE
TM      <- mean(timevec, trim = TF) # calculate the trimmed mean of the time vector
TSE     <- trimse(timevec, tr = TF) #calculate the trimmed mean of the time vector
#=== Winsorized Mean an SE, Median and SE of Median
WinS_Mean <- winmean(timevec, tr = TF, na.rm = FALSE) # winsorized mean
WinS_SE   <- winse(timevec, tr = TF) # winsorized mean
Med       <- median(timevec) # winsorized mean
Med_SE    <- msmedse(timevec) # winsorized mean
tv_DT     <- DescTools::Winsorize(timevec,
                                  val = quantile(timevec,
                                                 probs = c(TF, 1-TF),
                                  na.rm = FALSE))
M_DT      <- mean(tv_DT)
#=== manually winsorized
QTV              <- quantile(timevec, probs = c(TF, 1-TF))
QTIndUG          <- timevec <= QTV[1]
QTIndOG          <- timevec >= QTV[2]
timevec[QTIndUG] <- QTV[1]
timevec[QTIndOG] <- QTV[2]
ManMeanWS        <- mean(timevec)
```

Note that winsorizing is not equivalent to simply excluding (trimming) data, but is a method of censoring data. Thus a winsorized mean is not the same as a truncated mean (cf. Winsorizing, Wikipedia).

- the 10% trimmed mean is the average of the 5th to 95th percentile of the data
- the 90% winsorized mean sets the bottom 5% to the 5th percentile, the top 5% to the 95th percentile, and then averages the data.

Winsorization Round-Up

Following round-up is taken from this SAS-blog:

The good:

- The purpose of Winsorization is to "robustify" classical statistics by reducing the impact of extreme observations.
- If you compare a Winsorized statistic with classical statistic, you can identify variables that might contain contaminated data or are long-tailed and require special handling in models.

The ugly:

Modifying the data is a draconian measure. In his book (Tukey 2009), he says:

When statisticians encounter a few extreme values in data, we are likely to think of them as *strays*, *wild shots* and to focus our attention on how normally distributed the rest of the distribution appears to be. One who does this commits two oversights:

- forgetting Winsor's principle that *all distributions are normal in the middle*, and
- forgetting that the distribution relevant to statistical practice is that of the values actually provided and not of the values which ought to have been provided.

Concluding a bit further on:

> Sets of observations which have been *de-tailed* by *over-vigorous* use of a rule for rejecting outliers are inappropriate, since they are not samples.

M-estimators

A general family of robust location measures are so called M-estimators (the M stands for **M**aximum Likelihood-type"). They are based on a loss function to be minimized. Huber (1981) proposed a function (cf. Wilcox 2012) in which a *bending constant* $\boldsymbol{K}$ increases sensitivity to the tails of the distribution. The estimation of M-estimators is performed iteratively and implemented in the `mest()` function.

```
MWHub <- mest(timevec, bend = 1.28, na.rm = FALSE) # Huber-estimator
SEHub <- mestse(timevec)                           # Huber-estimator for SE
```

Measures of Scale

There are several robust measure of scales, of which we will only name a few, without getting into the mathematical details of their properties. Currently there are two general approaches to measuring scale that are of importance:

1. L-measures: is an estimator which is a linear combination of order statistics[7] of the measurements (which is also called an L-statistic). This can be as little as a single point, as in the median (of an odd number of values), or as many as all points, as in the mean. The main benefits of L-estimators are that they are often extremely simple, and *robust*.
2. M-measures: proposedby Huber (1964) these measures are a generalization of the maximum likelihood estimations. The M stands for *Maximum likelihood type*. M-measures can be constructed for location parameters and scale parameters in univariate and multivariate settings, as well as being used in robust regression.

Different measures of scale used frequently are:

1. Mean Deviation from the Mean
2. Mean Deviation from the Median
3. Median Absolute Deviation
4. q-Quantile Range
5. Winsorized Variance

Some of these measures will be used and discussed in the next chapter. For further details to their definition and properties refer to (page36 Wilcox 2012).

PART III: R-Examples

R-Packages

We will use the WRS2 (Mair, Schoenbrodt, & Wilcox, 2017), robustbase (Rousseeuw et al., 2015), and DescTools packages to access functions for some selected robust tests.

Furthermore, packages such as lme4 (Bates, Maechler, Bolker, & Walker, 2015) and robustlmm (Manuel Koller, 2016) for the multilevel model, and lavaan (Rosseel, 2012) for the latent growth model will be used in the course of this workshop.

To access these packages, paste and copy the following code:

```
rm(list = ls())
graphics.off()
options(digits = 2)
```

[7] together with rank statistics, order statistics are among the most fundamental tools in non-parametric statistics and inference. Examples of order statistics are maximum, minimum, range, quantiles, median, etc.

```r
if (!require("pacman")) install.packages("pacman")
pacman::p_load(corrplot, corrr,
               DescTools, dplyr,
               ggplot2, ggridges, ggpubr,
               knitr,
               lavaan, lme4,
               nlme,
               pander,
               readr, reshape, robustlmm, robustbase, rstatix,
               tidyverse,
               WRS2)

  ##############################################
```

Note to the `DescTools` *package:* the author's intention was to create a toolbox, which facilitates many of the tasks in data analysis, consisting of calculating descriptive statistics, drawing graphical summaries and reporting the results. The package contains furthermore functions to produce documents using MS Word (or PowerPoint) and functions to import data from Excel. Many of the included functions can be found scattered in other packages and other sources written partly by Titans of R. Important for us, robust methods such as robust estimators such as HuberM, TukeyBiweight, Robust data standardization, robust range, Yuen-t-Test, JarqueBeraTest, etc. can be found in this package.

Note to the `tidyverse` *package:* the tidyverse is an opinionated collection of R packages designed for data science. All packages share an underlying design philosophy, grammar, and data structures.

Other Packages in R

R is pretty well endowed with all sorts of robust functions and packages. For an extended overview see the additional packages for robust statistics on CRAN. From this list we name just a few as examples for the manifold statistical application areas:

- robust regression functions:
  - **rlm** in `MASS`
  - **lmrob** and **nlrob** in `robustbase`
- robust mixed-effects models - `robustlmm`
- robust generalized additive models - `robustgam`
- multivariate methods:
  - `rrcov` (robust multivariate variance-covariance estimation and robust principal components analysis (PCA))
  - `FRB` includes bootstrap based approaches for multivariate regression, PCA and Hotelling tests
  - `RSKC` functions for robust k-means clustering
  - `robustDA` performs robust discriminant analysis.

Two independent means

The data *fileKellyetalz.csv* (Kelly 2010) can be downloaded from the Blackboard.The file contains the data from a survey that investigated whether verbal information or modelling were effective in reversing the effect of verbal threat information on children's fears of novel animals.

Children aged 6-8 years old were given threat information or no information about two novel Australian marsupials. Following this information, different groups received one of three *interventions*:

1. positive information about the threat animal

2. a positive modelling experience (an adult placing their hand in a box seemingly containing the threat animal)
3. no further experience.

The children's *fear* of the marsupials was measured using a self-report measure called the **F**ear **B**eliefs **Q**uestionnaire (*FBQ*) or a **B**ehavioral **A**pproach **T**ask (*BAT*) like that described before. In the paper, the authors test the specificity of the interventions by comparing their effects on the subjective (FBQ) and behavioural (BAT) components of the fear emotion. To do so, a single score was computed separately for the *FBQ* and *BAT* that represented the change from pre-to post-intervention for the threat animal relative to the control animal. These scores, therefore, represent the overall effect of the intervention on each measure. The scores were converted to z-scores separately for the *FBQ* and *BAT* so that they could be compared.

Note that because the interventions are expected to reduce fear, greater efficacy is shown up by more negative z-scores (i.e. greater reductions in fear). This part of the study had a *mixed design* with a between group manipulation of intervention:

- positive information
- non-anxious modeling
- no intervention

and a repeated measures manipulation of the type of measure (*FBQ* or *BAT*). The data file contains 4 variables:

1. `id`: indicates the participant number;
2. `Intervention`: is a factor indicating whether the child received positive information, non-anxious modelling or no intervention,
3. `Measure`: indicates whether a score came from the *FBQ* or *BAT*
4. `z`: is the z-score associated with the measure.

Note that *FBQ* and *BAT* scores are in long format (contrasted with the wide format with which SPSS users will be more familiar).

```
# Kelly et al. (2010)
kellyz <- read.csv("Daten/Kellyetalz.csv")
pander(head(kellyz, 6), digits = 3)
```

id	Intervention	Measure	z
egs1	Positive Information	BAT	0.204
egs1	Positive Information	FBQ	-0.614
lrx2	Positive Information	BAT	-0.0234
lrx2	Positive Information	FBQ	-1.49
zej3	Positive Information	BAT	-0.498
zej3	Positive Information	FBQ	-1.66

Standard t-Test

We are going to compare the `FBQ` z-scores in two conditions: *positive information* vs. *No intervention*. To do this, we need to first select this subset of the data by executing these commands:

```
posInfoFBQ <- subset(kellyz, Intervention!= "Non-Anxious Modelling" &
                       Measure == "FBQ")
posInfoFBQ$Intervention <- factor(posInfoFBQ$Intervention)
pander(head(posInfoFBQ[order(posInfoFBQ$id),], 6))
```

	id	Intervention	Measure	z
60	aqa30	Positive Information	FBQ	-1.837
46	asl23	Positive Information	FBQ	0.4333
8	bcv4	Positive Information	FBQ	-1.313
44	bgi22	Positive Information	FBQ	-0.5271
16	czq8	Positive Information	FBQ	1.219
158	dsl79	No Intervention	FBQ	0.5206

The standard function `stats::t.test()` returns the Welch's t-test statistic. It is also referred to as the *unequal variances t-test*. It is an adaptation of Student's t-test and is more reliable when:

- the two samples have unequal variances and/or
- unequal sample sizes.

```
# pander(t.test(z ~ Intervention, data = posInfoFBQ, var.equal = TRUE))
pander(t.test(z ~ Intervention, data = posInfoFBQ))
```

Table 6: Welch Two Sample t-test: `z` by `Intervention` (continued below)

Test statistic	df	P value	Alternative hypothesis
7.119	53.01	2.896e-09 * * *	two.sided

mean in group No Intervention	mean in group Positive Information
0.6553	-0.7867

Robust t-Test

```
pander(YuenTTest(z ~ Intervention, data = posInfoFBQ))
```

Table 8: Yuen Two Sample t-test: `z` by `Intervention` (continued below)

Test statistic	df	trim	P value	Alternative hypothesis
6.244	28.84	0.2	8.371e-07 * * *	two.sided

trimmed mean in group No Intervention	trimmed mean in group Positive Information
0.6245	-0.7701

```
pander(yuenbt(z ~ Intervention, data = posInfoFBQ))
```

- **test**: *6.071*
- **conf.int**: *0.8996* and *1.89*
- **p.value**: *0*
- **df**: *NA*
- **diff**: *1.395*
- **call**: `yuenbt(formula = z ~ Intervention, data = posInfoFBQ)`

```
##############################################
```

We could report the robust test as a significant difference between trimmed mean FBQ z-scores in the positive information intervention compared to no intervention, $M_{diff} = 1.39\ [0.91, 1.88],\ Y_t = 6.07,\ p < 0.001$.

One-way ANOVA

As in the previous example, we create a new data frame that includes only the FBQ data:

```
fbqOnly <- subset(kellyz, Measure == "FBQ")
```

Standard ANOVA

The comparison of FBQ means across all three intervention groups can be done with an one-way independent ANOVA. We will compare the robust test to the classic linear model, which can be obtained by executing:

```
pander(summary(aov(z ~ Intervention, data = fbqOnly)))
```

Table 10: Analysis of Variance Model

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Intervention	2	39.02	19.51	30.29	4.314e-11
Residuals	104	66.98	0.6441	NA	NA

For the follow-up tests we will use the pairwise.t.test() function:

```
pander(pairwise.t.test(fbqOnly$z, fbqOnly$Intervention, p.adjust.method = "bonferroni"))
```

- **method**: t tests with pooled SD
- **data.name**: fbqOnly*zandfbqOnly*Intervention
- **p.value**:

	No Intervention	Non-Anxious Modelling
Non-Anxious Modelling	0.04234	NA
Positive Information	3.693e-11	4.819e-06

- **p.adjust.method**: bonferroni

Robust One-way ANOVA

The functions t1waybt() and mcppb20() take a similar form to yuenbt():

```
pander(t1waybt(z ~ Intervention, data = fbqOnly, tr = 0.2, nboot = 599))
```

- **test**: *19.9*
- **p.value**: *0*
- **Var.Explained**: *0.661*
- **Effect.Size**: *0.813*
- **nboot.eff**: *599*
- **call**: t1waybt(formula = z ~ Intervention, data = fbqOnly, tr = 0.2, nboot = 599)

The robust test produces an effect size ($\equiv$ Pearson r). We observe a significant difference between the trimmed mean `FBQ` scores from the intervention groups, $F_t = 19.90,\ p < 0.001$.

For the follow-up tests we will use the `mcppb20()` function.

```
pander(mcppb20(z ~ Intervention,  data = fbqOnly, tr = 0.2, nboot = 599))
```

- **comp**:

Group	Group	psihat	ci.lower	ci.upper	p-value
1	2	0.3825	0.07899	0.7234	0.003339
1	3	1.395	0.9248	1.873	0
2	3	1.012	0.4526	1.482	0

- **fnames**: *Positive Information*, *Non-Anxious Modelling* and *No Intervention*
- **call**: `mcppb20(formula = z ~ Intervention, data = fbqOnly, tr = 0.2, nboot = 599)`

```
################################################
```

The post-hoc tests tell us the difference between trimmed means, the associated bootstrap confidence interval, and the p-value for this difference.

Based on the trimmed mean difference in `FBQ` scores, the intervention was significantly more effective for positive information than modeling, $\hat{\psi} = 0.38$ $[0.07, 0.76]$, and no intervention, $\hat{\psi} = 1.39$ $[0.87, 1.89]$, and for modeling compared to no intervention, $\hat{\psi} = 1.01$ $[0.42, 1.57]$.

Two dependent means

```
# Field & Lawson (2003)
  # fieldWide <- read.csv("Daten/FieldLawson2003.csv")
  # head(fieldWide, 5)
  # fieldLong <- read.csv("Daten/FieldLawson2003Long.csv")
  # head(fieldLong, 5)
# Kelly et al. (2010)
  # kellyz <- read.csv("Daten/Kellyetalz.csv")
  # head(kellyz, 5)
  # kellyz[c(1:6, 91:96, 181:186),]
# Field & Cartwright-Hatton (2008)
  # fieldCH <- read.csv("Daten/FieldCH2008.csv")
  # head(fieldCH, 5)
# RCT data
  rctLong <- read.csv("Daten/RCTLong.csv", header = T)
  rctWide <- read.csv("Daten/RCTWide.csv", header = T)
  # head(rctWide, 5)

  ################################################
```

The data used in this example (*FieldLawson2003.csv*, *FieldLawson2003Long.csv*) can be downloaded from the Blackboard (cf. &. L. Field A. P. 2003).

```
  # Field & Lawson (2003)
  fieldWide <- read.csv("Daten/FieldLawson2003.csv")
  fieldLong <- read.csv("Daten/FieldLawson2003Long.csv")
  pander(head(fieldWide, 5), digits = 2)
```

id	zThreat	zPos	zNone
gup1	-0.38	-1.1	0.016
wdd2	1	2.6	-0.094
epr3	-0.19	-0.0017	-0.11
gna4	0.75	-0.95	0.38
gnn5	-0.06	-1.1	-0.22

```
pander(head(fieldLong[order(fieldLong$id),], 6), digits = 2)
```

	X	id	InfoType	value
23	23	aal26	zThreat	0.016
66	66	aal26	zPos	-0.23
109	109	aal26	zNone	-0.62
36	36	aco50	zThreat	0.21
79	79	aco50	zPos	0.34
122	122	aco50	zNone	-0.54

In this experiment, children aged 6-9 years were given verbal information about two novel Australian marsupials that contained:

- either threat, or
- positive content.

A third marsupial, about which no information was given, acted as a control. (The type of information was counterbalanced across animals for different children). After the information, children were asked to approach three boxes that they were told contained the animals (in fact they did not). *Latency* to approach the boxes acted as a behavior measure of their fear of these animals.

This part of the experiment has a one-way repeated measures design (children approached all three boxes). The *approach times* were *reported as z-scores* where a positive score indicates that children took longer than average to approach, 0 represents the average approach time, and a negative score is indicative of being faster than average to approach.

The data frame `fieldWide` contains 4 variables:

1. `id`: indicates the participant code
2. `zThreat`: threat information
3. `zPos`: positive information
4. `zNone`: no information

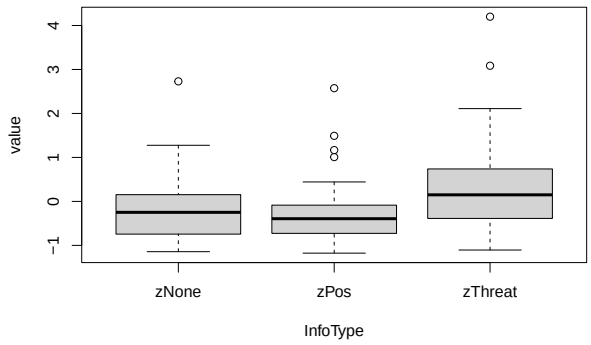
A version of this data file in 'long' format (*FieldLawson2003Long.csv*) contains these data (`fieldLong`) restructured into four variables:

1. `x`: consecutive number (not used for the analysis)
2. `id`: as above
3. `InfoType`: codes whether a score relates to an animal about which threat, positive or no information was given, and
4. `value`: contains the z-score for the time for a given child to approach a given box.

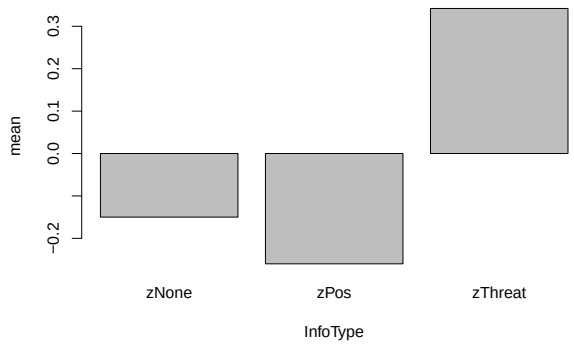
Check Assumptions

Explore the data by drawing boxplot, histograms and if desired some descriptive statistics. Discuss the results of your analysis.

```
boxplot(value ~ InfoType, fieldLong) # use Long Format Data
```



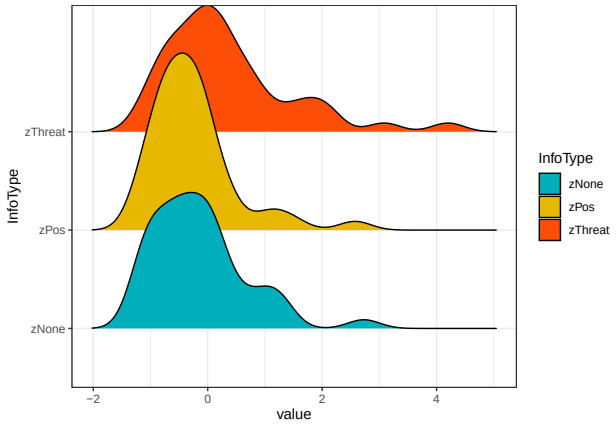

```
  DF <- fieldWide %>% gather(InfoType, value, 2:4) # or change to Long
  DF$InfoType <- as.factor(DF$InfoType)
  boxplot(value ~ InfoType, DF)
DF_Means <- DF %>% group_by(InfoType) %>%
            summarise_at(.vars = names(.)[3],.funs = c(mean="mean"))
barplot(mean ~ InfoType, DF_Means)
```




```
# hist(fieldWide$zThreat)
# hist(fieldWide$zPos)
# hist(fieldWide$zNone)
```

A valuable plot to see the distributional properties of the data is the histogram or the density plot, e.g.:

```
  ggplot(fieldLong, aes(x = value, y = InfoType)) +
  geom_density_ridges(aes(fill = InfoType)) +
  scale_fill_manual(values = c("#00AFBB", "#E7B800", "#FC4E07")) +
    theme_bw()
```



Standard t-Test

Calculate the standard parametric t-Test for repeatet measure designs and discuss the results.

```
pander(t.test(fieldWide$zThreat, fieldWide$zNone, paired = T), digits = 3)
```

Table 15: Paired t-test: `fieldWide$zThreat` and `fieldWide$zNone`

Test statistic	df	P value	Alternative hypothesis	mean difference
2.87	42	0.006405 * *	two.sided	0.492

Robust t-Test

Next we will look at the Yuen's modified t-Test for trimmed means (Yuen 1974). For an extented version with bootstrapping refer to (Keselman 2004)

```
pander(yuend(fieldWide$zThreat, fieldWide$zNone, tr = 0.2), digits = 3)
```

- **test**: *2.528*
- **conf.int**: *0.0756* and *0.7335*
- **se**: *0.16*
- **p.value**: *0.01789*
- **df**: *26*
- **diff**: *0.4046*
- **effsize**: *0.3321*
- **call**: `yuend(x = fieldWide$zThreat, y = fieldWide$zNone, tr = 0.2)`

```
set.seed(123)
pander(Dqcomhd(fieldWide$zThreat, fieldWide$zNone, nboot = 200, q = c(0.25, 0.5, 0.75)),  digits = 3)
```

- **partable**:

q	n1	n2	est1	est2	est1-est.2	ci.low	ci.up	p.crit	p.value
0.25	43	43	-0.4243	-0.7578	0.3335	0.01017	0.6655	0.05	0.03
0.5	43	43	0.1021	-0.2668	0.3688	0.07493	0.685	0.01667	0
0.75	43	43	0.8274	0.2319	0.5956	0.1083	1.198	0.025	0.03

- **call**: `Dqcomhd(x = fieldWide$zThreat, y = fieldWide$zNone, q = c(0.25, 0.5, 0.75), nboot = 200)`

```r
set.seed(123)
pander(dep.effect(fieldWide$zThreat, fieldWide$zNone))
```

	NULL	Est	S	M	L	ci.low	ci.up
AKP	0	0.3869	0.1	0.3	0.5	0.1349	0.7173
QS (median)	0.5	0.7209	0.54	0.62	0.69	0.5116	0.814
QStr	0.5	0.6977	0.54	0.62	0.69	0.5349	0.7907
SIGN	0.5	0.3256	0.46	0.38	0.31	0.192	0.476

```r
##############################################
```

Both tests yield significant differences. Note that trimming reduces:

1. the mean difference from 0.49 to 0.40, and
2. the test statistic is smaller in the robust version.

We could report the robust test as a significant difference between trimmed mean approach times to the threat and control animals, $M_{diff} = 0.40\ [0.08, 0.74],\ Y_t(26) = 2.53, p = 0.018$.

One-way RM-ANOVA

The data used in this example is the same as for the dependent t-Test (i.e.: *FieldLawson2003.csv*)

Check Assumptions

To compare the latencies for all three boxes, we could use a repeated measure ANOVA function such as the `aov()`, or as an alternative the `rstatix::anova_test()` function. But before we do the analysis, we should check for possible violations of the assumptions:

```r
# Assumption check
fieldLong           <- fieldLong[,2:4]
fieldLong$id        <- factor(fieldLong$id)
fieldLong$InfoType <- factor(fieldLong$InfoType)
pander(fieldLong %>% group_by(InfoType) %>% identify_outliers(value))
```

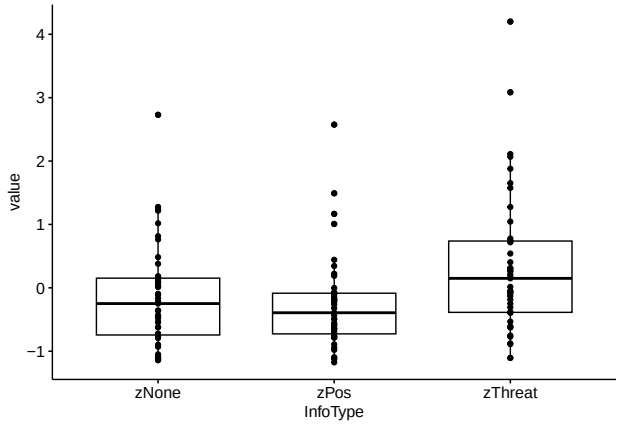
InfoType	id	value	is.outlier	is.extreme
zNone	bku59	2.729	TRUE	FALSE
zPos	wdd2	2.574	TRUE	TRUE
zPos	inl37	1.491	TRUE	FALSE
zPos	vxp18	1.008	TRUE	FALSE
zPos	pap44	1.166	TRUE	FALSE
zThreat	tat36	4.2	TRUE	TRUE
zThreat	gtu30	3.085	TRUE	FALSE

```r
pander(rbind(shapiro_test(fieldWide$zThreat),
             shapiro_test(fieldWide$zPos),
             shapiro_test(fieldWide$zNone)))
```

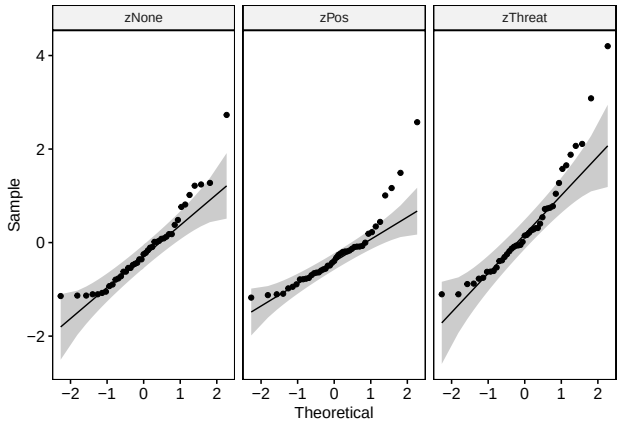
variable	statistic	p.value
fieldWide$zThreat	0.8894	0.0006053
fieldWide$zPos	0.8463	4.207e-05

variable	statistic	p.value
fieldWide$zNone	0.9047	0.001741

```
bxp         <- ggboxplot(fieldLong,
                         x = "InfoType",
                         y = "value",
                         add = "point")
print(bxp)
```




```
qqp         <- ggqqplot(fieldLong,
                        "value",
                        facet.by = "InfoType")
print(qqp)
```



Standard RM-ANOVA

Since ANOVA's are special forms of linear models, we could also use the `lm()` function. The difference between these functions is the output. The `aov()` returns the table of F-statistics, whereas `lm()` returns the specific parameter estimates, significance tests and overall fit statistics.

The formula (model) of the `aov()` is specified as:

Value ~ InfoType + Error(id/InfoType)

In words: we predict the *Value* from the variable *InfoType* plus an *error term* for that variable that is nested within the variable *id*. It is the error term that tells the function that it is a repeated measures design

(because the error term for the predictor variable is nested within cases).

```
#Non-robust
pander(summary(aov(value ~ InfoType + Error(id/InfoType), data = fieldLong)))
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Residuals	42	51.33	1.222	NA	NA
InfoType	2	8.839	4.42	7.103	0.001412
Residuals1	84	52.26	0.6222	NA	NA

```
# Alternativ to aov(): rstatix::anova_test()
res.anova  <- pander(anova_test(data = fieldLong,
                                dv = value,
                                wid = id,
                                within = InfoType))
```

The results show means which are significantly different with latencies after threat information being significantly longer than for positive or no information. The results of the follow-up-test are shown below:

```
pander(pairwise.t.test(fieldLong$value, fieldLong$InfoType, p.adjust.method = "bonferroni", paired = 
```

- **method**: paired t tests
- **data.name**: fieldLong*valueandfieldLong*InfoType
- **p.value**:

	zNone	zPos
zPos	1	NA
zThreat	0.01922	0.005809

- **p.adjust.method**: bonferroni

Robust ANOVA

For the robust test we will use the `rmanovab()` function and get the post hoc tests with `pairdepb()`. The option `tr` controls the amount of trim (and the default of 20% is advised). For the bootstrap-option the default is set to `nboot = 599` (sufficient for now, but it is common to use `nboot = 1000` or `nboot = 2000`).

```
#Robust
pander(rmanovab(fieldLong$value, fieldLong$InfoType, fieldLong$id, tr = 0.2, nboot = 599))
```

- **test**: *6.751*
- **crit**: *3.439*
- **call**: `rmanovab(y = fieldLong$value, groups = fieldLong$InfoType, blocks = fieldLong$id, tr = 0.2, nboot = 599)`

The robust test results also show a significant difference between trimmed mean approach times to the three animals, $F_t = 6.75$, $p < .050$. The results of the robust follow-up test show the difference between trimmed means ($\hat{\psi}$), the associated bootstrap confidence interval, the test of this difference, the critical value of the test and whether the trimmed means are significantly different (at $\alpha = 0.05$):

```
pander(pairdepb(fieldLong$value, fieldLong$InfoType, fieldLong$id, tr = 0.2, nboot = 599))
```

- **comp**:

Group	Group	psihat	ci.lower	ci.upper		crit
1	2	0.5232	0.1599	0.8866	3.389	2.353
1	3	0.4046	0.02792	0.7812	2.528	2.353
2	3	-0.1187	-0.4294	0.192	-0.8989	2.353

- **fnames**: *zThreat*, *zPos* and *zNone*
- **call**: `pairdepb(y = fieldLong$value, groups = fieldLong$InfoType, blocks = fieldLong$id, tr = 0.2, nboot = 599)`

```
############################################
```

We would report that the trimmed mean difference in latency between the threat box and the positive, $\hat{\psi} = 0.52$ $[0.14, 0.90]$, and no information, $\hat{\psi} = 0.40$ $[0.01, 0.80]$[8] boxes were significant. The trimmed mean difference between the positive and the no information box was not, $\hat{\psi} = -0.12$ $[-0.44, 0.90]$.

Regression Models

Download the data for this example *FieldCH2008.csv* from the Blackboard. The study examines the extent to which *social anxiety* can be predicted from measures of *worry*, *shame*, *visual imagery* and *obsessive beliefs*.

The example demonstrates a standard and a robust linear model with multiple predictors (i.e., multiple regression). A subset of the data is shown in the following table:

```
fieldCH <- read_csv("Daten/FieldCH2008.csv")
head(fieldCH)
```

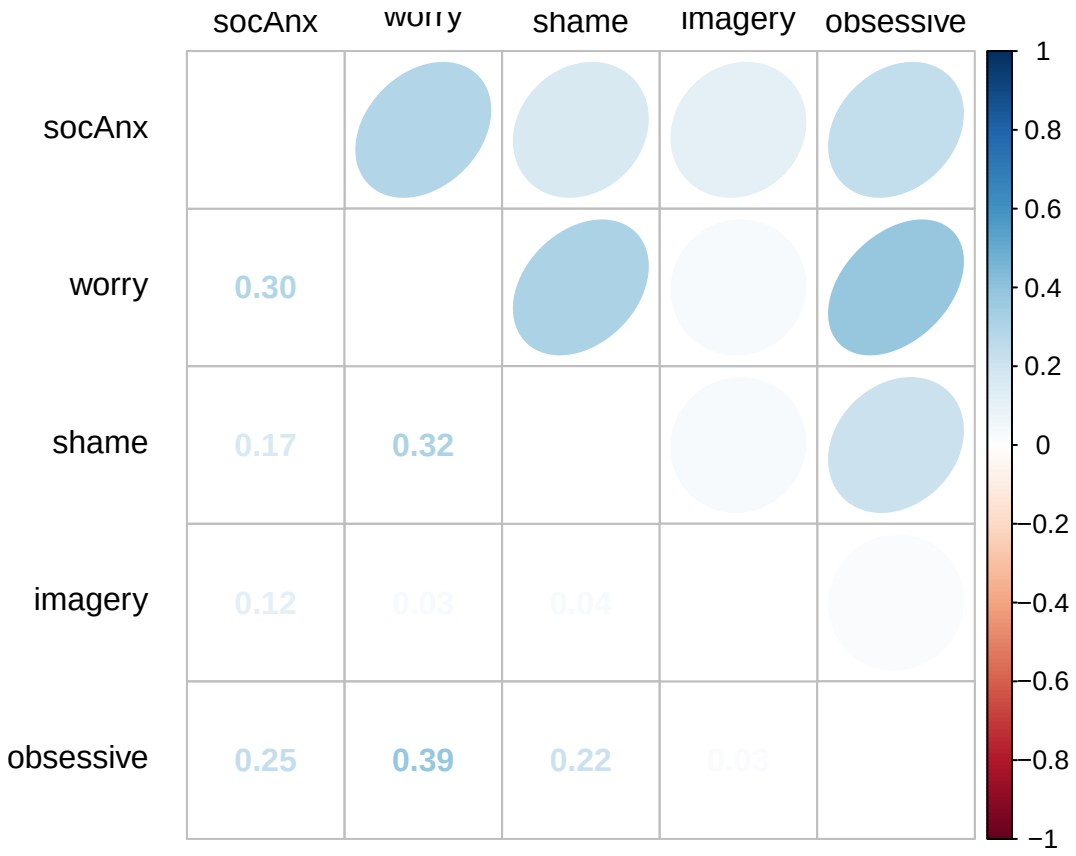
```
## # A tibble: 6 x 6
##      id socAnx worry shame imagery obsessive
##   <dbl>  <dbl> <dbl> <dbl>   <dbl>     <dbl>
## 1     1     36    64  151       68       319
## 2     2     24    41  139.      73       158
## 3     3    106    78  179      104       421
## 4     4     42    37  147.      58       221
## 5     5     55    49  158       88       332
## 6     6     77    62  165      118       255
```

Check Data

Before running a regression model we should have a closer look at the correlations between all the variables of the model. One very convinient way to do this is to use the `corrplot()` function:

```
A <- na.omit(fieldCH[,2:6])
CorrMat <- cor(A)
corrplot.mixed(CorrMat, upper = "ellipse", lower = "number",
               tl.pos = "lt", tl.col = "black", tl.offset=1, tl.srt = 0)
```

[8] be cautios with the interpretation of an effect if the CI has an upper, or lower limit so close to Null!



Scatter-Plots are also a good way to get a better insight in the beahaviour of your data:

Again, many different routines and functions are available to check for the assumptions of a linear model. For now we will leave it with that and turn to the linear model itself.

Standard Linear Model

We use the `lm()` at its most basic form:

```
socAnx.normal <- lm(socAnx ~ worry + shame + imagery + obsessive, data = fieldCH)
A <- summary(socAnx.normal)
pander(summary(socAnx.normal))
```

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	5.289	7.369	0.7177	0.4733
worry	0.4379	0.09484	4.617	4.946e-06
shame	0.05957	0.04145	1.437	0.1513
imagery	0.1216	0.04847	2.508	0.01245
obsessive	0.04917	0.01525	3.224	0.001346

Table 24: Fitting linear model: socAnx ~ worry + shame + imagery + obsessive

Observations	Residual Std. Error	R^2	Adjusted R^2
512	23.26	0.1229	0.116

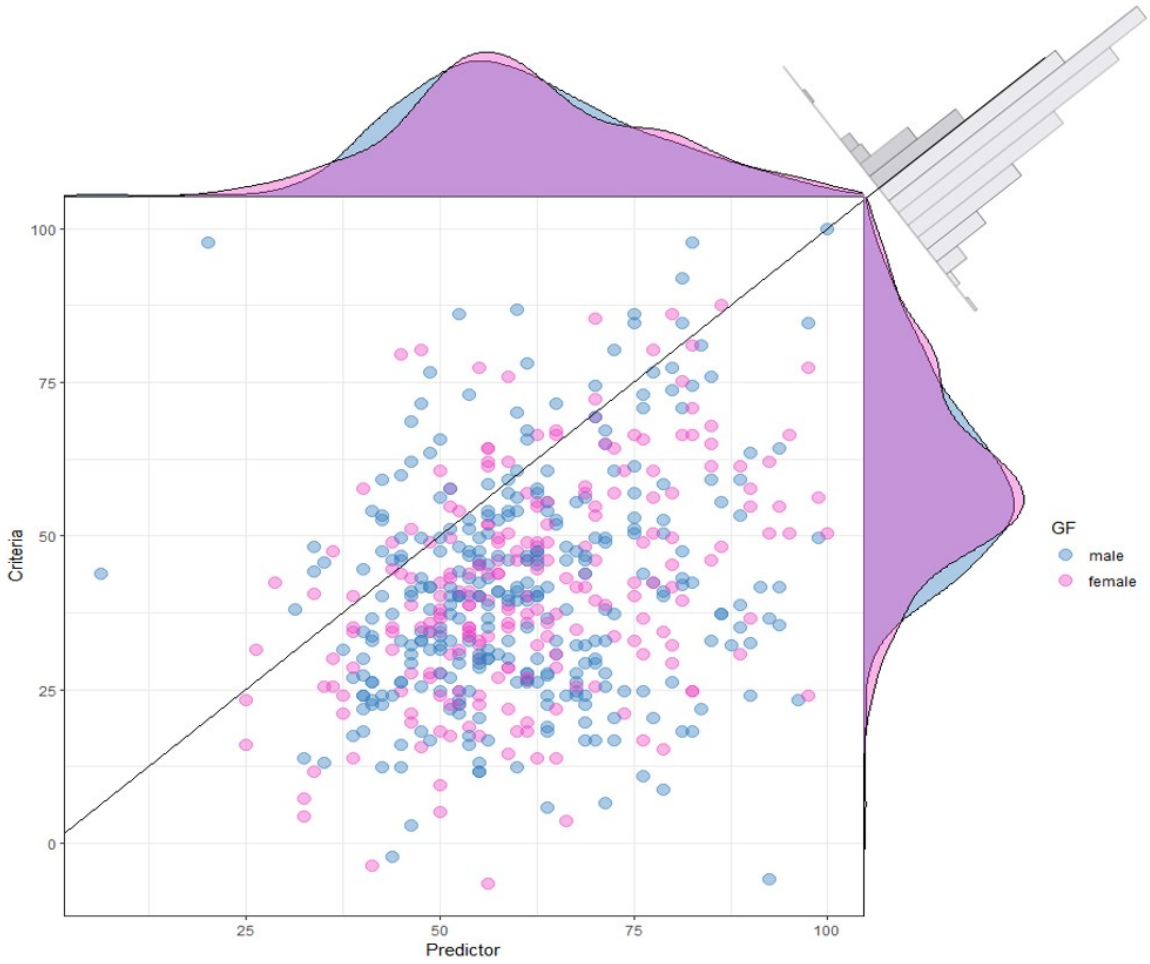


Figure 5: Figure: Scatterplot for data-check

Robust Linear Model (RLM)

There are several options for fitting robust regression in R. We will demonstrate the `lmrob()` function, which fits a robust variant of the social anxiety model based on an M-estimator (Koller 2011) using iteratively reweighted least squares (IRWLS) estimation.

This function, at its most basic, takes the same form as `lm()`, which means that we can simply replace `lm` with `lmrob` and proceed as before.

```
socAnx.robust <- lmrob(socAnx ~ worry + shame + imagery + obsessive, data = fieldCH)
summary(socAnx.robust)
```

```
##
## Call:
## lmrob(formula = socAnx ~ worry + shame + imagery + obsessive, data = fieldCH)
##  \--> method = "MM"
## Residuals:
##      Min       1Q   Median       3Q      Max
## -77.8329 -15.1522   0.0181  13.8852 101.7230
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   5.3514     7.8857    0.68  0.49769
## worry         0.4844     0.1006    4.81    2e-06 ***
## shame         0.0259     0.0496    0.52  0.60162
## imagery       0.1294     0.0475    2.72  0.00666 **
## obsessive     0.0552     0.0158    3.48  0.00054 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Robust residual standard error: 22
##   (47 Beobachtungen als fehlend gelöscht)
## Multiple R-squared:  0.141,  Adjusted R-squared:  0.135
## Convergence in 15 IRWLS iterations
##
## Robustness weights:
##  observation 222 is an outlier with |weight| <= 8.9e-06 ( < 0.0002);
##  45 weights are ~= 1. The remaining 466 ones are summarized as
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.17    0.87    0.95    0.90    0.98    1.00
## Algorithmic parameters:
##        tuning.chi                bb        tuning.psi        refine.tol
##          1.55e+00          5.00e-01          4.69e+00          1.00e-07
##           rel.tol         scale.tol         solve.tol          zero.tol
##          1.00e-07          1.00e-10          1.00e-07          1.00e-10
##       eps.outlier             eps.x warn.limit.reject warn.limit.meanrw
##          1.95e-04          9.53e-10          5.00e-01          5.00e-01
##      nResample         max.it       best.r.s       k.fast.s          k.max
##            500             50              2              1            200
##    maxit.scale      trace.lev            mts     compute.rd fast.s.large.n
##            200              0           1000              0           2000
##                   psi           subsampling                   cov
##            "bisquare"         "nonsingular"         ".vcov.avar1"
## compute.outlier.stats
##                  "SM"
## seed : int(0)
```

Note that the b-values (now labelled *Estimates*), standard errors, t-values and p-values are slightly different.

The interpretation of the model does not change substantially (worry, visual imagery and obsessive beliefs significantly predict social anxiety, shame does not) but:

> the parameter estimates and associated standard error, test statistic and p-value from the robust model will have been relatively unaffected by the shape of the model residuals and outliers etc.

RLM-Extended

Given that our earlier examples are also variants of the linear model, we could also use the `lmrob()` function if we wanted to use an M-estimator instead of *trimmed means*.

For example, the classical models that compared two independent meansand several independent means were obtained using the `aov()` function, but this function is a *wrapper* for the `lm()` function that expresses the model in terms of F-statistics (as in ANOVA) rather than model parameters.

If we use the `lm()` function directly to fit these models we obtain the model parameters:

```
    summary(lm(z ~ Intervention, data = posInfoFBQ))
```

```
##
## Call:
## lm(formula = z ~ Intervention, data = posInfoFBQ)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.7089 -0.4072  0.0399  0.4222  2.0057
##
## Coefficients:
##                                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)                        0.655      0.148    4.44  3.3e-05 ***
## InterventionPositive Information  -1.442      0.206   -7.00  1.3e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.87 on 70 degrees of freedom
## Multiple R-squared:  0.412,  Adjusted R-squared:  0.403
## F-statistic:   49 on 1 and 70 DF,  p-value: 1.26e-09
```

```
    summary(lm(z ~ Intervention, data = fbqOnly))
```

```
##
## Call:
## lm(formula = z ~ Intervention, data = fbqOnly)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.7089 -0.4178  0.0399  0.4752  2.0057
##
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                         0.655      0.136    4.83  4.7e-06 ***
## InterventionNon-Anxious Modelling  -0.479      0.192   -2.50    0.014 *
## InterventionPositive Information   -1.442      0.189   -7.62  1.2e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 0.8 on 104 degrees of freedom
## Multiple R-squared:  0.368,  Adjusted R-squared:  0.356
## F-statistic: 30.3 on 2 and 104 DF,  p-value: 4.31e-11
```

It is a simple matter to estimate these parameters with an M-estimator by replacing `lm` with `lmrob`:

```
    summary(lmrob(z ~ Intervention, data = posInfoFBQ))
```

```
##
## Call:
## lmrob(formula = z ~ Intervention, data = posInfoFBQ)
##  \--> method = "MM"
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.7796 -0.4382  0.0142  0.4031  1.9350
##
## Coefficients:
##                                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)                       0.6572     0.0824    7.97  2.0e-11 ***
## InterventionPositive Information  -1.3732     0.2239   -6.13  4.6e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Robust residual standard error: 0.66
## Multiple R-squared:  0.465,  Adjusted R-squared:  0.458
## Convergence in 16 IRWLS iterations
##
## Robustness weights:
##  13 weights are ~= 1. The remaining 59 ones are summarized as
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.04    0.75    0.94    0.84    0.97    1.00
## Algorithmic parameters:
##        tuning.chi                bb        tuning.psi        refine.tol
##          1.55e+00          5.00e-01          4.69e+00          1.00e-07
##           rel.tol         scale.tol         solve.tol          zero.tol
##          1.00e-07          1.00e-10          1.00e-07          1.00e-10
##       eps.outlier             eps.x warn.limit.reject warn.limit.meanrw
##          1.39e-03          1.82e-12          5.00e-01          5.00e-01
##      nResample         max.it       best.r.s       k.fast.s          k.max
##            500             50              2              1            200
##    maxit.scale      trace.lev            mts     compute.rd fast.s.large.n
##            200              0           1000              0           2000
##                  psi           subsampling                   cov
##            "bisquare"         "nonsingular"         ".vcov.avar1"
## compute.outlier.stats
##                  "SM"
## seed : int(0)
```

```
    summary(lmrob(z ~ Intervention, data = fbqOnly))
```

```
##
## Call:
## lmrob(formula = z ~ Intervention, data = fbqOnly)
##  \--> method = "MM"
## Residuals:
##     Min      1Q  Median      3Q     Max
```

```
## -2.7795 -0.4527  0.0143  0.4204  1.9351
##
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                        0.6572     0.0823    7.98  2.0e-12 ***
## InterventionNon-Anxious Modelling -0.4391     0.1360   -3.23   0.0017 **
## InterventionPositive Information  -1.3733     0.2246   -6.11  1.7e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Robust residual standard error: 0.65
## Multiple R-squared:  0.399,  Adjusted R-squared:  0.388
## Convergence in 16 IRWLS iterations
##
## Robustness weights:
##  15 weights are ~= 1. The remaining 92 ones are summarized as
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.03    0.86    0.94    0.87    0.98    1.00
## Algorithmic parameters:
##        tuning.chi                bb        tuning.psi        refine.tol
##          1.55e+00          5.00e-01          4.69e+00          1.00e-07
##           rel.tol         scale.tol         solve.tol          zero.tol
##          1.00e-07          1.00e-10          1.00e-07          1.00e-10
##       eps.outlier             eps.x warn.limit.reject warn.limit.meanrw
##          9.35e-04          1.82e-12          5.00e-01          5.00e-01
##      nResample         max.it       best.r.s       k.fast.s          k.max
##            500             50              2              1            200
##    maxit.scale      trace.lev            mts     compute.rd fast.s.large.n
##            200              0           1000              0           2000
##                   psi           subsampling                   cov
##            "bisquare"         "nonsingular"         ".vcov.avar1"
## compute.outlier.stats
##                  "SM"
## seed : int(0)
```

PART IV: Summary

Conclusions and recommendations

Overall we can conclude that:

1. the assumptions of the statistical models frequently used are (highly) unlikely to be met for psychological data
2. violating these assumptions has unpleasant and undesirable effects on the:
   - model parameters
   - their associated standard errors, confidence intervals and
   - p-values
3. traditional methods for dealing with violations of model assumptions are (often) ineffective
4. there are numerous robust alternatives to the models
5. they are straightforward to implement

Recommendations

The possible **practical consequences** of violating assumptions include:

- relatively low power
- inaccurate confidence intervals
- inaccurate measures of effect size (that miss important differences)

Given Micceri's findings (Micceri 1989) and according to the paper of Field and Wilcox (&. W. Field A. P. 2017) it seems highly improbable that every paper not explicitly demonstrating that model assumptions have been met have, in reality, met the model assumptions. Hence, following recommondations are given:

- Scientists, reviewers and editors should assume that assumptions have not been met unless there is an explicit and compelling statement.
- It must be backed up by evidence, that the assumptions of the models fit have been met, but it is not recommend that these statements are based upon significance tests of assumptions[9]
- **Currently the best way** of investigating the impact of violations of model assumptions is to **use a modern robust method and compare the results to the standard model**. If the assumptions are met, the expectation is that they will give consistent results. Otherwise, the conventional method is in doubt.
- Do not rely on large sample size!
  - Heavy tail problems (Micceri 1989)[10]
  - Skew problems[11]
- insist on sensitivity analysis for all frequentist analyses, i.e. non-robust estimators (such as OLS and ML) are compared to a robust variant. Where the two models yield ostensibly the same results then either model may be reported, where the models deviate substantially then the robust model should be reported unless a compelling evidence-based case can be made that model assumptions have been met.
- when for a certain problem a specific test is unavailable, it should be technically possible to bootstrap standard errors and confidence intervals from pretty much any model.
- statements along the lines of 'ANOVA/regression/the t-test is robust' should be banned!

Brunner E., Langer F., Domhof S. 2002. *Nonparametric Analysis of Longitudinal Data in Factorial Experiments.* John Wiley & Sons, New York.

Field, & Lawson, A. P. 2003. "Fear Information and the Development of Fears During Childhood: Effects on Implicit Fear Responses and Behavioural Avoidance." *Behaviour Research and Therapy, 41(11), 1277-1293.* https://doi.org/10.1016/s0005-7967(03)00034-2.

Field, & Wilcox, A. P. 2017. "Robust Statistical Methods: A Primer for Clinical Psychology and Experimental Psychopathology Researchers." *Behaviour Research and Therapy, 98, 19-38.* https://doi.org/https://doi.org/10.1016/j.brat.2017.05.013.

Hurn, M. David. 2008. "Modern Robust Statistical Methods: An Easy Way to Maximize the Accuracy and Power of Your Research." *American Psychologist, Vol. 63, No. 7, 591–601.* https://doi.org/10.1037/0003-066X.63.7.591.

Kelly, Barker, V. L. 2010. "Can Rachman's Indirect Pathways Be Used to Un-Learn Fear? A Prospective Paradigm to Test Whether Children's Fears Can Be Reduced Using Positive Information and Modelling a Non-Anxious Response." *Behaviour Research and Therapy, 48(2), 164-170.* https://doi.org/10.1016/j.brat.2009.10.002.

[9] because, under general conditions, such tests do not have enough power to detect violations of assumptions that have practical consequences (Kesleman 2016).

[10] he studied the distributional characteristics of 440 large-sample psychology-relevant measures. Remarkably, when looking at tail weight only 15.2% approximated a normal distribution and nearly 67% had at least one tail that was moderately to extremely heavy. In terms of symmetry, only 28.4% approximated a normal distribution with the remainder moderately to extremely skewed. Looking at both symmetry and tail weight together only 6.8% of the 440 distributions approximated normality. These data show that tail weight and symmetry consistent with a normal distribution is extremely rare in psychological data.

[11] poor control over the Type I error probability can occur even with large sample sizes. Imagine a two-sample Student's T, $N_1 = 400, sd = 1$, **sampling distribution = lognormal** distribution. $N_2 = 1000, sd = 1$, **sampling distribution = normal**. The Type I error probability is approximately **0.14** rather than the nominal 0.05 → regardless of how large the sample size might be, results can be misleading.

Keselman, H. J. 2004. "The New and Improved Two-Sample t Test." *American Psychological Society, 15(1), 47-51.* https://doi.org/10.1111/j.0963-7214.2004.01501008.x.

Kesleman, Othman, H. J. 2016. "Generalized Linear Model Analyses for Treatment Group Equality When Data Are Non-Normal." *Journal of Modern Applied Statistical Methods, 15(1), 32-61.* https://doi.org/http://dx.doi.org/10.1037//0033-2909.105.1.156.

Koller, & Stahel, M. 2011. "Sharpening Wald-Type Inference in Robust Regression for Small Samples." *Computational Statistics & Data Analysis, 58(8), 2504-2515.* https://doi.org/http://dx.doi.org/10.1016/j.csda.2011.02.014.

Mair, Wilcox R., P. 2020. "Robust Statistical Methods in r Using the WRS2 Package." *Behavioural Research Methods, 52(2), 464-488.* https://doi.org/10.3758/s13428-019-01246-w.

Micceri, T. 1989. "The Unicorn, the Normal Curve, and Other Improbable Creatures." *Psychological Bulletin, 105(1), 156-166.* https://doi.org/http://dx.doi.org/10.1037//0033-2909.105.1.156.

Tukey, John W. 2009. *A Survey of Sampling from Contaminated Distributions.* Princeton, New Jersey: Princeton University.

Wilcox, R. 2012. *Introduction to Robust Estimation & Hypothesis Testing.* 3rd ed. Amsterdam, The Netherlands: Elsevier.

Yuen, K. K. 1974. "2-Sample Trimmed t for Unequal Population Variances." *Biometrika, 61(1), 165-170.* https://doi.org/http://dx.doi.org/10.1093/biomet/61.1.165.