The R Package trafo for Transforming Linear Regression Models

Lily Medina
Humboldt Universität zu Berlin

Piedad Castro
Humboldt Universität zu Berlin

Ann-Kristin Kreutzmann
Freie Universität Berlin

Natalia Rojas-Perilla
Freie Universität Berlin

Abstract

The linear regression model has been widely used for descriptive, predictive, and inferential purposes. This model relies on a set of assumptions, which are not always fulfilled when working with empirical data. In this case, one solution could be the use of more complex regression methods that do not strictly rely in the same assumptions. However, in order to improve the validity of model assumptions, transformations are a simpler approach and enable the user to keep using the well-known linear regression model. But how can a user find a suitable transformation? The R package **trafo** offers a simple user-friendly framework for selecting a suitable transformation depending on the user needs. The collection of selected transformations and estimation methods in the package **trafo** complement and enlarge the methods that are existing in R so far.

Keywords: power transformations, optimal parameter, model assumptions, normality.

1. Introduction

To study the relation between two or more variables, the linear regression model is one of the most employed statistical methods. For an appropriate usage of this model, a set of assumptions needs to be fulfilled. These assumptions are, among others, related to the functional form and to the error terms, such as linearity and homoscedasticity. However, in practical applications, these assumptions are not always satisfied. This leads to the question of how the practitioner can move on with the analysis in such case. One way to proceed is to conduct the analysis ignoring the model assumption violations which is, of course, not recommended as it would likely yield misleading results. Another solution is to use more complex methods such as generalized linear regression models or non-parametric methods, as they might fit the data and problem better. A third method, which also constitutes the focus of the present paper, is the application of suitable transformations. In this work, we mean by transformation that the response variable of the linear model is transformed with a known transformation function. For more flexible transformation functions, we refer e.g., to Hothorn, Möst, and Bühlmann (2018).

Transformations have the potential to correct certain violations and by doing so, enable to continue the analysis with the known (linear) regression model. Due to its convenience,

transformations such as the logarithm or the Box-Cox are commonly applied in many branches of sciences; for example in economics (Hossain 2011) and neuroscience (Morozova, Koschutnig, Klein, and Wood 2016). In order to simplify the choice and the usage of transformations in the linear regression model, the R (R Core Team 2018) package **trafo** (Medina, Castro, Kreutzmann, and Rojas-Perilla 2018) is developed. The present work is inspired by the framework proposed in Rojas-Perilla (2018, pp. 9-45) and extends other existing R packages that provide transformations.

Many packages that contain transformations do not focus especially on the usage of transformations (Venables and Ripley 2002; Fox and Weisberg 2011; Molina and Marhuenda 2015; Ribeiro Jr. and Diggle 2016). Therefore, they often only include popular transformations like the logarithmic or the Box-Cox transformation family. The package **car** (Fox and Weisberg 2011) expands the selection of transformations. It includes the Box-Cox, the basic power, and the Yeo-Johnson transformation families, and uses the maximum likelihood approach for the estimation of the transformation parameter. An exponential transformation proposed by Manly (1976) is provided in the package **caret** (Kuhn 2008) and the multiple parameter Johnson transformation in the packages **Johnson** (Fernandez 2014) and **jtrans** (Wang 2015). While package **MASS** (Venables and Ripley 2002) and package **car** (Fox and Weisberg 2011) only provide the maximum likelihood approach for the estimation of the transformation parameter for the Box-Cox family, the estimation can be conducted by a wide range of methods in the **AID** package (Dag, Asar, and Ilk 2017). Most of the provided methods are based on goodness of fit tests like the Shapiro-Wilk or the Anderson-Darling test. However, the **AID** package only contains the Box-Cox transformation.

It is noticeable that none of the above-mentioned packages helps the user in the process of deciding which transformation is actually suitable according to the users needs. Furthermore, most packages do not provide tools to see at the first sight if the transformation improves the untransformed model with regards to fulfilling the model assumptions. Therefore, package **trafo** combines and extends the features provided by the packages mentioned above. Additionally to transformations that are already provided by existing packages, the **trafo** package includes, among others, the Bickel-Doksum (Bickel and Doksum 1981), modulus (John and Draper 1980), the neglog (Whittaker, Whitehead, and Somers 2005) and glog (Durbin, Hardin, Hawkins, and Rocke 2002) transformations that are modifications of the Box-Cox and the logarithmic transformation, respectively, in order to deal with negative values in the response variable. Furthermore, the selection of estimation methods for the transformation parameter is enlarged by methods based on moments and divergence measures (see e.g., Taylor 1985; Yeo and Johnson 2000; Royston, Lambert *et al.* 2011). The main benefits of the package **trafo** can be summarized as follows:

- An initial check can be conducted that helps to decide if and which transformation is useful for the researchers needs.
- The untransformed model and a model with a transformed dependent variable as well as two transformed models can be run simultaneously, and thus the models can be easily compared with regard to the model assumptions.
- Extensive diagnostics are provided in order to check if the transformation helps to fulfill the model assumptions normality, homoscedasticity, and linearity.

The remainder of this paper is structured as follows. In Section 2, the transformations and estimation methods included in the package are presented. Section 3 demonstrates in form of a case study the functionality of the package. Section 4 summarizes the user-defined function feature of the package. In Section 5, some concluding remarks and potential extensions of the package are discussed. Finally, Appendix A presents the mathematical derivations underlying the package.

2. Transformations and estimation methods

The equation describing and summarizing the relationship between a continuous outcome variable Y and different covariates X (either discrete or continuous) is defined by $y_i = \mathbf{x}_i^T\boldsymbol{\beta}+e_i$, with $i = 1, \ldots, n$. This is also known as the linear regression model and is composed by a deterministic and a random component, which rely on different assumptions. Among others, these assumptions can be summarized as follows:

- Normality (N): The conditional distribution of y given x follows a normal distribution. This is an optional, but often desired assumption.
- Homoscedasticity (H): The conditional variance of y given x is constant.
- Linearity (L): The conditional expectation of the outcome variable y given the continuous covariates x is a linear function in x.

As already mentioned, different approaches have been proposed for achieving these model assumptions. Some of them include using alternative estimation methods of the regression terms or applying more complex regression models (see e.g., Nelder and Wedderburn 1972; Berry 1993). In this paper, we focus on defining a parsimonious re-specification for the model, such as the usage of non-linear transformations of the outcome variable. The transformations implemented in the package **trafo** basically help to achieve normality. However, most of them simultaneously correct other assumptions (see also Table 1 and Table 2).
The transformations can be classified into transformations without a transformation parameter and data-driven transformations with a transformation parameter that needs to be estimated. The first set of transformations presented in Table 1 comprises, among others, the logarithmic transformation, which is considered due to its popularity and straightforward application. The data-driven transformations presented in Table 2 are dominated by the

Table 1: Transformations without transformation parameter.

Transformation	Source	Formula	Support	N	H	L
Log (shift)	Box and Cox (1964)	$\log(y + s)$	$y \in \mathbb{R}$	✗	✗	✗
Glog	Durbin *et al.* (2002)	$\log(y + \sqrt{y^2 + 1})$	$y \in \mathbb{R}$	✗	✗	✗
Neglog	Whittaker *et al.* (2005)	$\text{Sign}(y) \log(\lvert y\rvert + 1)$	$y \in \mathbb{R}$	✗	✗	
Reciprocal	Tukey (1977)	$\frac{1}{y}$	$y \neq 0$	✗	✗	

Box-Cox transformation and its modifications or alternatives, e.g., the modulus or Bickel-Doksum transformation. However, more flexible versions of the logarithmic transformation,

as the log-shift opt, or the Manly transformation, which is an exponential transformation, are also included in the package **trafo**.

Table 2: Data-driven transformations.

Transformation	Source	Formula	Support	N	H	L
Box-Cox (shift)	Box and Cox (1964)	$\begin{cases} \frac{(y+s)^\lambda - 1}{\lambda} & \text{if } \lambda \neq 0; \\ \log(y+s) & \text{if } \lambda = 0. \end{cases}$	$y \in \mathbb{R}$	✗	✗	✗
Log-shift opt	Feng, Hannig, and Marron (2016)	$\log(y + \lambda)$	$y \in \mathbb{R}$	✗	✗	✗
Bickel-Docksum	Bickel and Doksum (1981)	$\frac{\lvert y\rvert^\lambda \text{Sign}(y) - 1}{\lambda}$ if $\lambda > 0$	$y \in \mathbb{R}$	✗	✗	
Yeo-Johnson	Yeo and Johnson (2000)	$\begin{cases} \frac{(y+1)^\lambda - 1}{\lambda} & \text{if } \lambda \neq 0, y \geq 0; \\ \log(y+1) & \text{if } \lambda = 0, y \geq 0; \\ \frac{(1-y)^{2-\lambda} - 1}{\lambda - 2} & \text{if } \lambda \neq 2, y < 0; \\ -\log(1-y) & \text{if } \lambda = 2, y < 0. \end{cases}$	$y \in \mathbb{R}$	✗	✗	
Square Root (shift)	Medina *et al.* (2018)	$\sqrt{y + \lambda}$	$y \in \mathbb{R}$	✗	✗	
Manly	Manly (1976)	$\begin{cases} \frac{e^{\lambda y} - 1}{\lambda} & \text{if } \lambda \neq 0; \\ y & \text{if } \lambda = 0. \end{cases}$	$y \in \mathbb{R}$	✗	✗	
Modulus	John and Draper (1980)	$\begin{cases} \text{Sign}(y) \frac{(\lvert y\rvert + 1)^\lambda - 1}{\lambda} & \text{if } \lambda \neq 0; \\ \text{Sign}(y) \log(\lvert y\rvert + 1) & \text{if } \lambda = 0. \end{cases}$	$y \in \mathbb{R}$	✗		
Dual	Yang (2006)	$\begin{cases} \frac{(y^\lambda - y^{-\lambda})}{2\lambda} & \text{if } \lambda > 0; \\ \log(y) & \text{if } \lambda = 0. \end{cases}$	$y > 0$	✗		
Gpower	Kelmansky, Martínez, and Leiva (2013)	$\begin{cases} \frac{(y + \sqrt{y^2+1})^\lambda - 1}{\lambda} & \text{if } \lambda \neq 0; \\ \log(y + \sqrt{y^2+1}) & \text{if } \lambda = 0. \end{cases}$	$y \in \mathbb{R}$	✗		

Table 1 and 2 provide information about the range of the dependent variable that is supported by the transformation. Some transformations are only suitable for positive values of y. This is generally true for the logarithmic and Box-Cox transformations. However, in case that the dependent variable contains negative values, the values are shifted by a deterministic shift s such that $y + s > 0$ by default in package **trafo**. Furthermore, the tables emphasize which assumptions the transformation helps to achieve. These are general suggestions and the actual success always also depends on the data. For specific properties of each transformation we refer to the original references. The square root shift transformation with a data-driven shift in analogy to the log-shift opt transformation is, to the best of our knowledge, firstly implemented in this work. In contrast, a square root transformation with deterministic shift, for example, is suggested in Bartlett (1947).

Since the transformations in Table 2 contain transformation parameters that need to be estimated, package **trafo** contains different methodologies for this estimation. The benefit of each estimation method depends on the research analysis and the underlying data. They can be summarized as follows:

- Maximum likelihood theory
- Distribution moments optimization: Skewness or kurtosis
- Divergence minimization: Following Kolmogorov-Smirnov (KS), Cramér-von-Mises (KM) or Kullback-Leibler (KL) measurements

Table 3: Diagnostic checks provided in the package **trafo**

Assumption	Diagnostic check	Fast check
Normality	Skewness and kurtosis	✗
	Shapiro-Wilk test	✗
	Quantile-quantile plot	
	Histograms	
Homoscedasticity	Breusch-Pagan test	✗
	Residuals vs. fitted plot	
	Scale-location	
Linearity	Scatter plots between y and x	✗
	Observed vs. fitted plot	

The maximum likelihood estimation method finds the set of values for the transformation parameter that maximizes the likelihood function of the dataset under the selected transformation (Box and Cox 1964). This is a standard approach that is also implemented in several of the mentioned R packages (Venables and Ripley 2002; Fox and Weisberg 2011). However, since the maximum likelihood estimation is rather sensitive to outliers, the skewness or kurtosis optimization might be preferable for the estimation of the transformation parameter in the presence of such outliers (see e.g., Royston *et al.* 2011). These methods are especially favorable when it is important in the analysis to meet these moments. For instance, skewness minimization should be used when it is important to get a symmetric distribution. Additionally, if the focus lies on comparing the whole distribution of the transformed data with a normal distribution, and not only some moments, different divergence measures as the KS, KM or KL can be used (see e.g., Yeo and Johnson 2000). For all estimation methods, a lambda range on which the functions are evaluated needs to be proposed. Therefore, default values are set for the predefined transformations. For more information about different estimation methods we refer to Rojas-Perilla (2018, pp. 9-45).

Since the user can only decide if the transformation is helpful by checking the above mentioned assumptions, the package **trafo** contains a wide range of diagnostic checks (e.g., Shapiro and Wilk 1965; Breusch and Pagan 1979). A smaller selection is used in the fast check that helps to decide if a transformation might be useful. Table 3 summarizes the implemented diagnostic checks that are simultaneously returned for the untransformed and a transformed model or two differently transformed models and indicates which diagnostics are conducted in the fast check. Additionally, plots are provided that help to detect outliers such as the Cook's distance plot and influential observations by the residuals vs leverage plot.

Another feature of the package **trafo** is the possibility of defining a customized transformation. Thus, a user can also use the infrastructure of the package for a transformation that suits the individuals needs better than the predefined transformations. However, in this version of the package **trafo** the user needs to define the transformation and the standardized transformation in order to use this feature. For the derivation of the standardized transformation of all predefined transformations, see Appendix A.

Table 4: Core functions of package **trafo**

Function	Description
`assumptions()`	Enables a fast check which transformation is suitable.
`trafo_lm()`	Compares the untransformed model with a transformed model.
`trafo_compare()`	Compares two differently transformed models.
`diagnostics()`	Returns information about the transformation and different diagnostics checks in form of tests.
`plot()`	Returns graphical diagnostics checks.

3. Case study

In order to show the functionality of the package **trafo**, we present – in form of a case study – the steps a user faces when checking the assumptions of the linear model. For this illustration, we use the data set called `University` from the R package **Ecdat** (Croissant 2016). This data set contains variables about the equipment and costs of university teaching and research and can be obtained as follows:

```
R> library(Ecdat)
R> data(University)
```

A practical question for the head of a university could be how study fees (`stfees`) raise the universities net assets (`nassets`). Both variables are metric. Thus, a linear regression could help to explain the relation between these two variables. A linear regression model can be conducted in R using the `lm` function.

```
R> linMod <- lm(nassets ~ stfees, data = University)
```

The features in the package **trafo** that help to find a suitable transformation for this model and to compare different models are summarized in Table 4 and illustrated in the next sections.

3.1. Finding a suitable transformation

It is well known that the reliability of the linear regression model depends on assumptions. Amongst others, normality, homoscedasticity, and linearity are assumed. In this section, we focus on presenting how the user can decide and assess, if and which, transformations help to fulfill these model assumptions. Thus, a first fast check of these model assumptions can be used in the package **trafo** in order to find out if the untransformed model meets these assumptions or if using a transformation seems suitable. The fast check can be conducted by the function `assumptions`. This function returns the skewness, the kurtosis and the Shapiro-Wilk test for normality, the Breusch-Pagan test for homoscedasticity and scatter plots between the dependent and the explanatory variables for checking the linear relation. All possible arguments of the function `assumptions` are summarized in Table 5. In the following, we only show the returned normality and homoscedasticity tests. The results are ordered by the highest p value of the Shapiro-Wilk and Breusch-Pagan test.

```
R> assumptions(linMod)

The default lambdarange for the log shift opt transformation is calculated
dependent on the data range. The lower value is set to -2035.751 and the upper
value to 404527.249

The default lambdarange for the square root shift transformation is calculated
dependent on the data range. The lower value is set to -2035.751 and the upper
value to 404527.249

Test normality assumption
              Skewness Kurtosis Shapiro_W Shapiro_p
logshiftopt    -0.4201   4.0576    0.9741    0.2132
boxcox         -0.4892   4.2171    0.9621    0.0527
bickeldoksum   -0.4892   4.2171    0.9621    0.0527
gpower         -0.4892   4.2171    0.9621    0.0527
modulus        -0.4892   4.2171    0.9621    0.0527
yeojohnson     -0.4892   4.2171    0.9621    0.0527
dual           -0.4837   4.2180    0.9619    0.0519
sqrtshift       0.6454   5.2752    0.9504    0.0139
log            -1.1653   5.1156    0.9140    0.0004
neglog         -1.1651   5.1150    0.9140    0.0004
glog           -1.1653   5.1156    0.9140    0.0004
untransformed   2.4503  12.7087    0.7922    0.0000
reciprocal     -3.7260  19.0487    0.5676    0.0000

Test homoscedasticity assumption
              BreuschPagan_V BreuschPagan_p
modulus               0.1035         0.7477
yeojohnson            0.1035         0.7477
boxcox                0.1035         0.7476
bickeldoksum          0.1036         0.7476
gpower                0.1035         0.7476
dual                  0.1128         0.7369
logshiftopt           0.1154         0.7341
neglog                0.7155         0.3976
log                   0.7158         0.3975
glog                  0.7158         0.3975
reciprocal            1.6109         0.2044
sqrtshift             5.4624         0.0194
untransformed         9.8244         0.0017
```

Following the Shapiro-Wilk test, the best transformation to fulfill the normality assumption is the log-shift opt transformation followed by the Box-Cox, Bickel-Doksum, gpower, modulus and Yeo-Johnson transformation. The similarity or even equality of the test results for different transformations is due to the the same functional form in the case of a positive λ and positive values as e.g., the Box-Cox and Bickel-Doksum transformation, or to the rounding

Table 5: Arguments of function `assumptions`

Argument	Description	Default
`object`	Object of class `lm`.	
`method`	Estimation method for the transformation parameter.	Maximum likelihood
`std`	Normal or standardized transformation.	Normal
`...`	Addtional arguments can be added, especially for changing the lambda range for the estimation of the parameter, e.g. `manly_lr = c(0.000005,0.00005)`	Default values of lambda range of each transformation

at four decimals. For improving the homoscedasticity assumption, all transformations help except the square root (shift) transformation. As mentioned before, default values for the lambda range for all transformations are predefined and these are used in this fast check. Since the default values for the log-shift opt and square root (shift) transformation depend on the range of the response variable, the chosen range is reported in the return. The Manly transformation is not in the list since the default lambda range for the estimation of the transformation parameter is not suitable for this data set. It does not fit since the Manly transformation is an exponential transformation and therefore it rather fits for flat or left-skewed data in contrast to most of the other transformations. In the case that the default lambda range does not work, the user can change the lambda range for the transformations manually. Similarly, the user can change the estimation methods for the transformation parameter. For instance, if symmetry is of special interest for the user the skewness minimization might be a better choice than the default maximum likelihood method. In this case study all assumptions are assumed to be equally important. Thus, we choose the Box-Cox transformation for the further illustrations even though some other transformations would be suitable as well.

3.2. Comparing the untransformed model with a transformed model

For a more detailed comparison of the transformed model with the untransformed model, a function called `trafo_lm` (for the arguments see Table 6) can be used as follows:

```
R> linMod_trafo <- trafo_lm(linMod)
```

The Box-Cox transformation is the default option such that only the `lm` object needs to be given to the function. The object `linMod_trafo` is of class `trafo_lm` and the user can conduct the methods `print`, `summary` and `plot` in the same way as for an object of class `lm`. The difference is that the new methods simultaneously return the results for both models, the untransformed model and the transformed model. Furthermore, a method called `diagnostics` helps to compare results of normality and homoscedasticity tests. In the following, we will show the return of the `diagnostics` method and some selected plots in order to check the normality, homoscedasticity and the linearity assumption of the linear model.

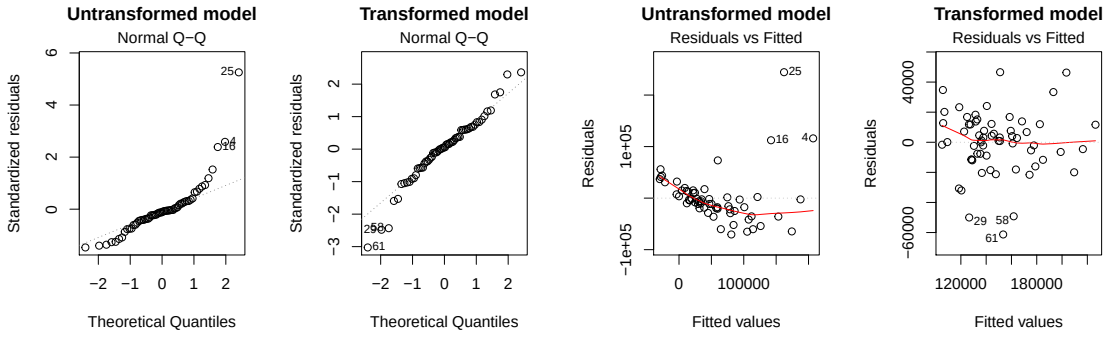
```
R> diagnostics(linMod_trafo)

Diagnostics: Untransformed vs transformed model

Transformation:  boxcox
Estimation method:  ml
Optimal Parameter:  0.1894257

Residual diagnostics:

Normality:
Pearson residuals:
                      Skewness  Kurtosis Shapiro_W    Shapiro_p
Untransformed model  2.4503325 12.708681 0.7921672 6.024297e-08
Transformed model   -0.4892222  4.217105 0.9620688 5.267566e-02

Heteroscedasticity:
                    BreuschPagan_V BreuschPagan_p
Untransformed model      9.8243555     0.00172216
Transformed model        0.1035373     0.74762531
```

The first part of the return shows information of the applied transformation. As chosen, the Box-Cox transformation is used with the optimal transformation parameter around 0.19 which is estimated using the maximum likelihood approach that is also set as default. The optimal transformation parameter differs from 0, which would be equal to the logarithmic transformation, and 1, which means that no transformation is optimal. The Shapiro-Wilk test rejects normality of the residuals of the untransformed model but it does not reject normality for the residuals of the transformed model on a 5% level of significance. Furthermore, the skewness shows that the residuals in the transformed model are more symmetric and the kurtosis is closer to 3, the value of the kurtosis of the normal distribution. The results of the Breusch-Pagan test clearly show that homoscedasticity is rejected in the untransformed model but not in the transformed model. These two findings can be supported by diagnostic plots shown in Figure 1.

```
R> plot(linMod_trafo)
```

In order to evaluate the linearity assumption, scatter plots of the dependent variable against the explanatory variable can help. Figure 2 shows that the assumption of linearity is violated in the untransformed model. In contrast, the relation between the transformed net assets and the study fees seems to be linear. As demonstrated above, the user can receive diagnostics for an untransformed and a transformed model with only a little more effort in comparison to fitting the standard linear regression model without transformation. While we only show the example with the default transformation, the user can also easily change the transformation and the estimation method. For instance, the user could choose the log-shift opt transformation with the skewness minimization as estimation method.

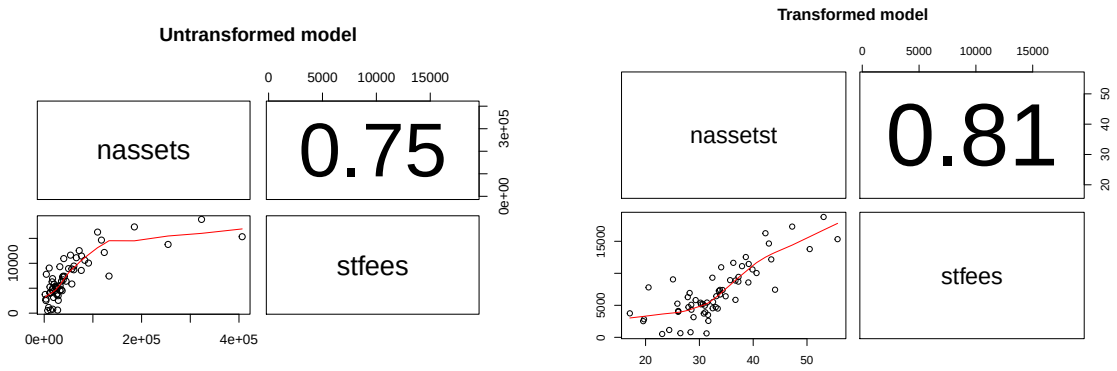
```
R> linMod_trafo2 <- trafo_lm(object = linMod, trafo = "logshiftopt",
+    method = "skew")
```



(a) Q-Q plots of the error terms.

(b) Residuals versus fitted.

Figure 1: Selection of diagnostic plots obtained by using `plot(linMod_trafo)`. (a) shows Q-Q plots error terms of the untransformed and the transformed model. (b) shows the residuals against the fitted values of the untransformed and the transformed model.



(a)

(b)

Figure 2: Selection of obtained diagnostic plots by using `plot(linMod_trafo)`. (a) shows the scatter plot of the untransformed net assets and the study fees (b) shows scatter plot of the transformed net assets and the study fees. The numbers specify the correlation coefficient between the dependent and independent variable.

Table 6: Arguments of function `trafo_lm`.

Argument	Description	Default
`object`	Object of class `lm`.	
`trafo`	Selected transformation.	Box-Cox
`lambda`	Estimation or a self-selected numeric value.	Estimation
`method`	Estimation method for the transformation parameter.	Maximum likelihood
`lambdarange`	Determines lambdarange for the estimation of the transformation parameter.	Default lambdarange for each transformation.
`std`	Normal or standardized transformation.	Normal
`custom_trafo`	Add customized transformation.	None

3.3. Comparing two transformed models

The user can also compare different transformations with regard to meet the model assumptions. In many present-day applications, the logarithm is often used without longer considerations about its usefulness. In order to compare the logarithm, e.g., with the selected Box-Cox transformation, the user needs to specify two objects of class `trafo` as follows:

```
R> boxcox_uni <- boxcox(linMod)
R> log_uni <- logtrafo(linMod)
```

The utility of `trafo` objects is twofold. First, the user can use the functions for each transformation in order to simply receive the transformed vector. The `print` method gives first information about the vector and the method `as.data.frame` returns the whole data frame with the transformed variable in the last column. The variable is named as the dependent variable with an added t.

```
R> head(as.data.frame(boxcox_uni))
```

```
    nassets stfees nassetst
1   3669.71   2821 19.71248
2  12156.00   4037 26.07723
3 185203.00  17296 47.24867
4 323100.00  18800 53.08840
5  32154.00   9314 32.42140
6  41669.00   7388 34.31882
```

Second, the objects can be used to compare linear models with differently transformed dependent variable using function `trafo_compare`. The arguments of this functions are shown in Table 7. The user creates an object of class `trafo_compare` by:

```
R> linMod_comp <- trafo_compare(object = linMod,
+    trafos = list(boxcox_uni, log_uni))
```

For this object, the user can use the same methods as for an object of class `trafo_lm`. In this work, we only want to show the return of method `diagnostics`.

```
R> diagnostics(linMod_comp)
```

```
Diagnostics of two transformed models

Transformations:  Box-Cox and Log
Estimation methods:  ml  and no estimation
Optimal Parameters:  0.1894257  and no parameter
```

Table 7: Arguments of function `trafo_compare`.

Argument	Description	Default
`object`	Object of class `lm`.	
`trafos`	List of objects of class `trafo`.	
`std`	Normal or standardized transformation.	Normal

```
Residual diagnostics:

Normality:
Pearson residuals:
          Skewness Kurtosis Shapiro_W    Shapiro_p
Box-Cox -0.4892222 4.217105 0.9620688 0.0526756632
Log     -1.1653028 5.115615 0.9140135 0.0003534879

Heteroscedasticity:
        BreuschPagan_V BreuschPagan_p
Box-Cox      0.1035373      0.7476253
Log          0.7158162      0.3975197
```

The first part of the return points out that the Box-Cox transformation is a data-driven transformation with a transformation parameter, while the logarithmic transformation does not adapt to the data. Furthermore, we can see that normality is rejected for the model with a logarithmic transformed dependent variable, while it is not rejected when the Box-Cox transformation is used. The violation of the homoscedasticity assumption can be fixed by both transformations.

4. Customized transformation

An additional user-friendly feature in the package **trafo** is the possibility of using the framework also for self-defined transformations. In the following, we show this option for the glog transformation.

In a first step, the transformation and the standardized or scaled transformation need to be defined. The mathematical expression of these two functions is presented in the Appendix A.2.

```
R> glog_trafo <- function(y) {
+    yt <-  log(y + sqrt(y^2 + 1))
+    return(y = yt)}

R> glog_std <- function(y) {
+    zt <-  log(y + sqrt(y^2 + 1)) * sqrt(geometric.mean(1 + y^2))
+    return(zt = zt)}
```

Second, the user inserts the two functions as a list argument to the `trafo_lm` function. Furthermore, the user needs to specify for the `trafo` argument if the transformation is without

a parameter (`"custom_wo"`) or with one parameter (`"custom_one"`). The glog transformation does not rely on a transformation parameter.

```
R> linMod_custom <- trafo_lm(linMod, trafo = "custom_wo",
+    custom_trafo = list(glog_trafo = glog_trafo, glog_std = glog_std))
```

One limitation of this feature is the necessity to insert both the transformation and the scaled transformation since the latter is often not known by the user. Furthermore, the framework is only suitable for transformations without and with one transformation parameter.

5. Conclusions and future developments

Even though the development in computing enables the use of complex methods nowadays, transformations are still a parsimonious way to meet model assumptions in a linear regression model. In Section 3, we demonstrated how the package **trafo** helps the user to decide easily if and which transformation is suitable to fulfill the model assumptions normality, homoscedasticity and linearity. To the best of our knowledge **trafo** is the only R package that supports this decision process. Furthermore, the package **trafo** provides an extensive collection of transformations usable in linear regression models and a wide range of estimation methods for the transformation parameter. In future versions, we plan to enlarge this collection constantly, also for other types of data, e.g, count data. Additionally, more methods that are available for the class `lm` could be developed for objects of class `trafo_lm`. We would also like to expand the infrastructure for linear mixed regression models.

A. Likelihood derivation of the transformations

A.1. Log (shift) transformation

Let $J(y)$ denote the Jacobian of a transformation from y_i to y_i^*. In order to obtain z_i^*, the scaled log (shift) transformation, given by $\frac{y_i^*}{J(y)^{1/n}}$, and for simplicity, we use a modification of the definition of the geometric mean, denoted by $\bar{y}_{\text{LS}}$. Therefore, the Jacobian, the scaled, and the inverse of the log (shift) transformation are given below.

The log (shift) transformation presented in Table 1 is defined as:

$$y_i^* = \log(y_i + s).$$

In case, the fixed shift parameter s would not be necessary, the standard logarithm function (logarithmic transformation with $s = 0$) is applied.

The modification of the definition of the geometric mean for this transformation is:

$$\bar{y}_{LS} = \left[\prod_{i=1}^{n} y_i + s \right]^{\frac{1}{n}}.$$

Therefore, the expression of the Jacobian is defined as:

$$\begin{aligned} J(\boldsymbol{y}) &= \prod_{i=1}^{n} \frac{dy_i^*}{dy} \\ &= \prod_{i=1}^{n} \frac{1}{y_i + s} \\ &= \bar{y}_{LS}^{-n}. \end{aligned}$$

The scaled transformation is given by:

$$z_i^* = \log(y_i + s)\bar{y}_{LS}.$$

The inverse function of the log (shift) transformation is denoted as:

$$\begin{aligned} f(y_i) &= \log(y_i + s) \\ y_i^* &= \log(y_i + s) \\ y_i &= e^{y_i^*} - s \\ \Rightarrow f^{-1}(y_i^*) &= e^{y_i^*} - s. \end{aligned}$$

A.2. Glog transformation

Let $J(y)$ denote the Jacobian of a transformation from y_i to y_i^*. In order to obtain z_i^*, the scaled glog transformation, given by $\frac{y_i^*}{J(y)^{1/n}}$, and for simplicity, we use a modification of the definition of the geometric mean, denoted by $\bar{y}_{\text{GL}}$. Therefore, the Jacobian, the scaled, and the inverse of the glog transformation are given below.

The glog transformation presented in Table 1 is defined as:

$$y_i^* = \log\left(y_i + \sqrt{y_i^2 + 1}\right) \text{ if } \lambda = 0.$$

The modification of the definition of the geometric mean for this transformation is:

$$\bar{y}_{GL} = \left[\prod_{i=1}^{n} 1 + y_i^2\right]^{\frac{1}{n}}.$$

Therefore, the expression of the Jacobian is defined as:

$$\begin{aligned}
J(\boldsymbol{y}) &= \prod_{i=1}^{n} \frac{dy_i^*}{dy} \\
&= \prod_{i=1}^{n} \frac{1}{y_i + \sqrt{y_i^2+1}} \left(1 + \frac{2y_i}{2\sqrt{y_i^2+1}}\right) \\
&= \prod_{i=1}^{n} \frac{1}{y_i + \sqrt{y_i^2+1}} \left(\frac{y_i + \sqrt{y_i^2+1}}{\sqrt{y_i^2+1}}\right) \\
&= \prod_{i=1}^{n} \frac{1}{\sqrt{y_i^2+1}} \\
&= \bar{y}_{GL}^{\frac{-n}{2}}.
\end{aligned}$$

The scaled transformation is given by:

$$z_i^* = \log\left(y_i + \sqrt{y_i^2+1}\right) \bar{y}_{GL}^{\frac{1}{2}}.$$

The inverse function of the glog transformation is denoted as:

$$\begin{aligned}
f(y_i) &= \log\left(y_i + \sqrt{y_i^2+1}\right) \\
y_i^* &= \log\left(y_i + \sqrt{y_i^2+1}\right) \\
e^{y_i^*} - y_i &= \sqrt{y_i^2+1} \\
(e^{y_i^*} - y_i)^2 &= y_i^2 + 1 \\
e^{y_i^{*2}} - 2e^{y_i^*} y_i &= 1 \\
y_i &= -\frac{(1 - e^{y_i^{*2}})}{2e^{y_i^*}} \\
\Rightarrow f^{-1}(y_i^*) &= -\frac{(1 - e^{y_i^{*2}})}{2e^{y_i^*}}.
\end{aligned}$$

A.3. Neglog transformation

Let $J(y)$ denote the Jacobian of a transformation from y_i to y_i^*. In order to obtain z_i^*, the scaled neglog transformation, given by $\frac{y_i^*}{J(y)^{1/n}}$, and for simplicity, we use a modification of the definition of the geometric mean, denoted by $\bar{y}_{\text{NL}}$. Therefore, the Jacobian, the scaled, and the inverse of the neglog transformation are given below.

The neglog transformation presented in Table 1 is defined as:

$$y_i^* = \text{sign}(y_i) \log\left(|y_i| + 1\right).$$

The modification of the definition of the geometric mean for this transformation is:

$$\bar{y}_{NL} = \left[\prod_{i=1}^{n} (|y_i| + 1)\right]^{\frac{1}{n}}.$$

Therefore, the expression of the Jacobian comes to:

$$\begin{aligned} J(\boldsymbol{y}) &= \prod_{i=1}^{n} \frac{dy_i^*}{dy} \\ &= \prod_{i=1}^{n} \operatorname{sign}(y_i) \frac{1}{|y_i| + 1} \\ &= \operatorname{sign}\left(\prod_{i=1}^{n} y_i \right) \left(\prod_{i=1}^{n} |y_i| + 1 \right)^{-1} \\ &= \operatorname{sign}\left(\prod_{i=1}^{n} y_i \right) \bar{y}_{NL}^{-n}. \end{aligned}$$

The scaled transformation is given by:

$$z_i^* = \operatorname{sign}(y_i) \log\left(|y_i| + 1\right) \operatorname{sign}\left(\prod_{i=1}^{n} y_i \right) \bar{y}_{NL}.$$

The inverse function of the neglog transformation is denoted as:

$$\begin{aligned} f(y_i) &= \operatorname{sign}(y_i) \log\left(|y_i| + 1\right) \\ y_i^* &= \operatorname{sign}(y_i) \log\left(|y_i| + 1\right) \\ |y_i| &= e^{\operatorname{sign}(y_i^*) y_i^*} - 1 \\ \Rightarrow f^{-1}(y_i^*) &= \pm\left[e^{\operatorname{sign}(y_i^*) y_i^*} - 1\right]. \end{aligned}$$

A.4. Reciprocal transformation

Let $J(y)$ denote the Jacobian of a transformation from y_i to y_i^*. In order to obtain z_i^*, the scaled reciprocal transformation, given by $\frac{y_i^*}{J(y)^{1/n}}$, and for simplicity, we use a modification of the definition of the geometric mean, denoted by $\bar{y}_{\mathrm{R}}$. Therefore, the Jacobian, the scaled, and the inverse of the reciprocal transformation are given below.

The reciprocal transformation presented in Table 1 is defined as:

$$y_i^* = \frac{1}{y_i}.$$

The definition of the geometric mean is:

$$\bar{y}_R = \left[\prod_{i=1}^{n} y_i \right]^{\frac{1}{n}}.$$

Therefore, the expression of the Jacobian is defined as:

$$\begin{aligned} J(\boldsymbol{y}) &= \prod_{i=1}^{n} \frac{dy_i^*}{dy} \\ &= \prod_{i=1}^{n} -\frac{1}{y_i^2} \\ &= -\bar{y}_R^{-2n}. \end{aligned}$$

The scaled transformation is given by:

$$z_i^* = -\frac{1}{y_i}\bar{y}_R^2.$$

The inverse function of the reciprocal transformation is denoted as:

$$\begin{aligned} f(y_i) &= \frac{1}{y_i} \\ y_i^* &= \frac{1}{y_i} \\ y_i &= \frac{1}{y_i^*} \\ \Rightarrow f^{-1}(y_i^*) &= \frac{1}{y_i^*}. \end{aligned}$$

A.5. Box-Cox (shift) transformation

$$y_i^*(\lambda) = \begin{cases} \frac{(y_i+s)^\lambda - 1}{\lambda} & \text{if } \lambda \neq 0 \quad (A); \\ \log(y_i + s) & \text{if } \lambda = 0 \quad (B). \end{cases}$$

Box-Cox (shift) transformation case (A)

Let $J(\lambda, y)$ denote the Jacobian of a transformation from y_i to $y_i^*(\lambda)$. In order to obtain $z_i^*(\lambda)$, the scaled Box-Cox (shift)(A) transformation, given by $\frac{y_i^*(\lambda)}{J(\lambda,y)^{1/n}}$, and for simplicity, we use a modification of the definition of the geometric mean, denoted by $\bar{y}_{\text{BC}}$. Therefore, the Jacobian, the scaled, and the inverse of the Box-Cox (shift)(A) transformation are given below.

The Box-Cox (shift)(A) transformation presented in Table 2 is defined as:

$$y_i^*(\lambda) = \frac{(y_i + s)^\lambda - 1}{\lambda} \text{ if } \lambda \neq 0.$$

In case, the fixed shift parameter s is not necessary for making the dataset positive, the standard Box-Cox transformation (with $s = 0$) is applied.

The definition of the geometric mean is:

$$\bar{y}_{BC} = \left[\prod_{i=1}^{n} y_i + s \right]^{\frac{1}{n}}.$$

Therefore, the expression of the Jacobian comes to:

$$\begin{aligned} J(\lambda, \boldsymbol{y}) &= \prod_{i=1}^{n} \frac{dy_i^*(\lambda)}{dy} \\ &= \prod_{i=1}^{n} \frac{\lambda(y_i + s)^{\lambda-1}}{\lambda} \\ &= \prod_{i=1}^{n} (y_i + s)^{\lambda-1} \\ &= \bar{y}_{BC}^{n(\lambda-1)}. \end{aligned}$$

The scaled transformation is given by:

$$z_i^*(\lambda) = \frac{(y_i + s)^\lambda - 1}{\lambda} \frac{1}{\bar{y}_{BC}^{\lambda-1}}.$$

The inverse function of the Box-Cox (shift)(A) transformation is denoted as:

$$\begin{aligned} f(y_i) &= \frac{(y_i + s)^\lambda - 1}{\lambda} \\ y_i^* &= \frac{(y_i + s)^\lambda - 1}{\lambda} \\ y_i &= (\lambda y_i^* + 1)^{\frac{1}{\lambda}} - s \\ \Rightarrow f^{-1}(y_i^*) &= (\lambda y_i^* + 1)^{\frac{1}{\lambda}} - s. \end{aligned}$$

Box-Cox (shift) transformation case (B)

This case is exactly equal to the log (shift) case.

A.6. Log-shift opt transformation

Let $J(\lambda, y)$ denote the Jacobian of a transformation from y_i to $y_i^*(\lambda)$. In order to obtain $z_i^*(\lambda)$, the scaled log-shift opt transformation, given by $\frac{y_i^*(\lambda)}{J(\lambda,y)^{1/n}}$, and for simplicity, we use a modification of the definition of the geometric mean, denoted by $\bar{y}_{\text{LSO}}$. Therefore, the Jacobian, the scaled, and the inverse of the log-shift opt transformation are given below.

The log-shift opt transformation presented in Table 2 is defined as:

$$y_i^*(\lambda) = \log(y_i + \lambda).$$

The modification of the definition of the geometric mean for this transformation is:

$$\bar{y}_{LSO} = \left[\prod_{i=1}^{n} y_i + \lambda \right]^{\frac{1}{n}}.$$

Therefore, the expression of the Jacobian is defined as:

$$\begin{aligned} J(\lambda, \boldsymbol{y}) &= \prod_{i=1}^{n} \frac{dy_i^*(\lambda)}{dy} \\ &= \prod_{i=1}^{n} \frac{1}{y_i + \lambda} \\ &= \bar{y}_{LSO}^{-n}. \end{aligned}$$

The scaled transformation is given by:

$$z_i^*(\lambda) = \log(y_i + \lambda)\bar{y}_{LSO}.$$

The inverse function of the log-shift opt transformation is denoted as:

$$\begin{aligned} f(y_i) &= \log(y_i + \lambda) \\ y_i^* &= \log(y_i + \lambda) \\ y_i &= e^{y_i^*} - \lambda \\ \Rightarrow f^{-1}(y_i^*) &= e^{y_i^*} - \lambda. \end{aligned}$$

A.7. Bickel-Docksum transformation

Let $J(\lambda, y)$ denote the Jacobian of a transformation from y_i to $y_i^*(\lambda)$. In order to obtain $z_i^*(\lambda)$, the scaled Bickel-Docksum transformation, given by $\frac{y_i^*(\lambda)}{J(\lambda,y)^{1/n}}$, and for simplicity, we use a modification of the definition of the geometric mean, denoted by $\bar{y}_{\text{BD}}$. Therefore, the Jacobian, the scaled, and the inverse of the Bickel-Docksum transformation are given below. The Bickel-Docksum transformation presented in Table 2 is defined as:

$$y_i^*(\lambda) = \frac{|y_i|^{\lambda}\text{sign}(y_i) - 1}{\lambda} \text{ if } \lambda > 0.$$

The modification of the definition of the geometric mean for this transformation is:

$$\bar{y}_{BD} = \left[\prod_{i=1}^{n} |y_i|\right]^{\frac{1}{n}}.$$

Therefore, the expression of the jacobian comes to:

$$\begin{aligned} J(\lambda, \boldsymbol{y}) &= \prod_{i=1}^{n} \frac{dy_i^*(\lambda)}{dy} \\ &= \prod_{i=1}^{n} \frac{\text{sign}(y_i)\lambda|y_i|^{\lambda-1}}{\lambda} \\ &= \text{sign}\left(\prod_{i=1}^{n} y_i\right)\left(\prod_{i=1}^{n} |y_i|\right)^{\lambda-1} \\ &= \text{sign}\left(\prod_{i=1}^{n} y_i\right)\bar{y}_{BD}^{n(\lambda-1)}. \end{aligned}$$

The scaled transformation is given by:

$$z_i^*(\lambda) = \frac{|y_i|^\lambda \text{sign}(y_i) - 1}{\lambda} \frac{1}{\text{sign}\left(\prod_{i=1}^n y_i\right) \bar{y}_{BD}^{(\lambda-1)}}.$$

The inverse function of the Bickel-Docksum transformation is denoted as:

$$\begin{aligned} f(y_i) &= \frac{|y_i|^\lambda \text{sign}(y_i) - 1}{\lambda} \\ y_i^* &= \frac{|y_i|^\lambda \text{sign}(y_i) - 1}{\lambda} \\ |y_i| &= \left[\text{sign}(y_i^*)(y_i^*\lambda + 1)\right]^{\frac{1}{\lambda}} \\ \Rightarrow f^{-1}(y_i^*) &= \pm\left[\text{sign}(y_i^*)(y_i^*\lambda + 1)\right]^{\frac{1}{\lambda}}. \end{aligned}$$

A.8. Yeo-Johnson transformation

$$y_{ij}^*(\lambda) = \begin{cases} \frac{(y_i+1)^\lambda - 1}{\lambda} & \text{if } \lambda \neq 0, y_i \geq 0 \quad (A); \\ \log(y_i + 1) & \text{if } \lambda = 0, y_i \geq 0 \quad (B); \\ -\frac{(1-y_i)^{2-\lambda} - 1}{2-\lambda} & \text{if } \lambda \neq 2, y_i < 0 \quad (C); \\ -\log(1 - y_i) & \text{if } \lambda = 0, y_i < 0 \quad (D). \end{cases}$$

Yeo-Johnson transformation case (A)

This case is exactly equal to the Box-Cox (shift) case (A), with $s = 1$.

Yeo-Johnson transformation case (B)

This case is exactly equal to the log (shift) case, with $s = 1$.

Yeo-Johnson transformation case (C)

Let $J(\lambda, y)$ denote the Jacobian of a transformation from y_i to $y_i^*(\lambda)$. In order to obtain $z_i^*(\lambda)$, the scaled Yeo-Johnson(C) transformation, given by $\frac{y_i^*(\lambda)}{J(\lambda,y)^{1/n}}$, and for simplicity, we use a modification of the definition of the geometric mean, denoted by $\bar{y}_{\text{YC}}$. Therefore, the Jacobian, the scaled, and the inverse of the Yeo-Johnson(C) transformation are given below. The Yeo-Johnson(C) transformation presented in Table 2 is defined as:

$$y_i^*(\lambda) = -\frac{(1 - y_i)^{2-\lambda} - 1}{2 - \lambda} \text{ if } \lambda \neq 2 \text{ and } y_i < 0.$$

The modification of the definition of the geometric mean for this transformation is:

$$\bar{y}_{YC} = \left[\prod_{i=1}^n 1 - y_i\right]^{\frac{1}{n}}.$$

Therefore, the expression of the Jacobian comes to:

$$\begin{aligned} J(\lambda, \boldsymbol{y}) &= \prod_{i=1}^{n} \frac{dy_i^*(\lambda)}{dy} \\ &= \prod_{i=1}^{n} \frac{(2-\lambda)(1-y_i)^{1-\lambda}}{2-\lambda} \\ &= \prod_{i=1}^{n} (1-y_i)^{1-\lambda} \\ &= \bar{y}_{YC}^{n(1-\lambda)}. \end{aligned}$$

The scaled transformation is given by:

$$z_i^*(\lambda) = -\frac{(1-y_{ij})^{2-\lambda}-1}{2-\lambda} \bar{y}_{YC}^{n(1-\lambda)}.$$

The inverse function of the Yeo-Johnson(C) transformation is denoted as:

$$\begin{aligned} f(y_i) &= -\frac{(1-y_i)^{2-\lambda}-1}{2-\lambda} \\ y_i^* &= -\frac{(1-y_i)^{2-\lambda}-1}{2-\lambda} \\ -y_i^*(2-\lambda) &= (1-y_i)^{2-\lambda} - 1 \\ y_i &= 1 - \big[-y_i^*(2-\lambda) + 1 \big]^{\frac{1}{2-\lambda}} \\ \Rightarrow f^{-1}(y_i^*) &= 1 - \big[-y_i^*(2-\lambda) + 1 \big]^{\frac{1}{2-\lambda}}. \end{aligned}$$

Yeo-Johnson transformation case (D)

Let $J(y)$ denote the Jacobian of a transformation from y_i to y_i^*. In order to obtain z_i^*, the scaled Yeo-Johnson(D) transformation, given by $\frac{y_i^*}{J(y)^{1/n}}$, and for simplicity, we use a modification of the definition of the geometric mean, denoted by $\bar{y}_{\text{YD}}$. Therefore, the Jacobian, the scaled, and the inverse of the Yeo-Johnson(D) transformation are given below.

The Yeo-Johnson(D) transformation presented in Table 2 is defined as:

$$y_i^* = -\log(1-y_i).$$

The modification of the definition of the geometric mean for this transformation is:

$$\bar{y}_{YD} = \left[\prod_{i=1}^{n} 1 - y_i \right]^{\frac{1}{n}}.$$

Therefore, the expression of the Jacobian is defined as:

$$\begin{aligned} J(\lambda, \boldsymbol{y}) &= \prod_{i=1}^{n} \frac{dy_i^*}{dy} \\ &= \prod_{i=1}^{n} \frac{1}{1-y_i} \\ &= \bar{y}_{YD}^{-n}. \end{aligned}$$

The scaled transformation is given by:

$$z_i^* = -\log(1 - y_i)\bar{y}_{YD}.$$

The inverse function of the Yeo-Johnson(D) transformation is denoted as:

$$\begin{aligned} f(y_i) &= -\log(1 - y_i) \\ y_i^* &= -\log(1 - y_i) \\ y_i &= -e^{-y_i^*} + 1 \\ \Rightarrow f^{-1}(y_i^*) &= -e^{-y_i^*} + 1. \end{aligned}$$

A.9. Square root-shift opt transformation

Let $J(\lambda, y)$ denote the Jacobian of a transformation from y_i to $y_i^*(\lambda)$. In order to obtain z_i^*, the scaled square root-shift opt transformation, given by $\frac{y_i^*(\lambda)}{J(\lambda,y)^{1/n}}$, and for simplicity, we use a modification of the definition of the geometric mean, denoted by $\bar{y}_{\text{SR}}$. Therefore, the Jacobian, the scaled, and the inverse of the square root-shift opt transformation are given below.

The square root-shift opt transformation presented in Table 2 is defined as:

$$y_i^*(\lambda) = \sqrt{y_i + \lambda}.$$

The definition of the geometric mean is:

$$\bar{y}_{SR} = \left[\prod_{i=1}^{n} y_i + \lambda\right]^{\frac{1}{n}}.$$

Therefore, the expression of the Jacobian is defined as:

$$\begin{aligned} J(\lambda, \boldsymbol{y}) &= \prod_{i=1}^{n} \frac{dy_i^*}{dy} \\ &= \prod_{i=1}^{n} -\frac{1}{2\sqrt{y_i + \lambda}} \\ &= \frac{1}{2}\bar{y}_{SR}^{\frac{-n}{2}}. \end{aligned}$$

The scaled transformation is given by:

$$z_i^* = -\frac{1}{y_i}\bar{y}_{SR}^2.$$

The inverse function of the square root-shift opt transformation is denoted as:

$$\begin{aligned} f(y_i) &= \sqrt{y_i + \lambda} \\ y_i^* &= \sqrt{y_i + \lambda} \\ y_i &= y_i^{*2} - \lambda \\ \Rightarrow f^{-1}(y_i^*) &= y_i^{*2} - \lambda. \end{aligned}$$

A.10. Manly transformation

$$y_i^*(\lambda) = \begin{cases} \frac{e^{\lambda y_i}-1}{\lambda} & \text{if } \lambda \neq 0 \quad (A); \\ y_i & \text{if } \lambda = 0 \quad (B). \end{cases}$$

Manly transformation case (A)

Let $J(\lambda, y)$ denote the Jacobian of a transformation from y_i to $y_i^*(\lambda)$. In order to obtain $z_i^*(\lambda)$, the scaled Manly(A) transformation, given by $\frac{y_i^*(\lambda)}{J(\lambda,y)^{1/n}}$, and for simplicity, we use a modification of the definition of the geometric mean, denoted by $\bar{y}_{\mathrm{M}}$. Therefore, the Jacobian, the scaled, and the inverse of the Manly(A) transformation are given below.

The Manly(A) transformation presented in Table 2 is defined as:

$$y_i^*(\lambda) = \frac{e^{\lambda y_i} - 1}{\lambda} \text{ if } \lambda \neq 0.$$

The modification of the definition of the geometric mean for this transformation is:

$$\begin{aligned} \bar{y}_M &= \left[\prod_{i=1}^{n} e^{y_i}\right]^{\frac{1}{n}} \\ &= \left[e^{\sum_{i=1}^{n} y_i}\right]^{\frac{1}{n}} \\ &= e^{\bar{y}}. \end{aligned}$$

Therefore, the expression of the Jacobian comes to:

$$\begin{aligned} J(\lambda, \boldsymbol{y}) &= \prod_{i=1}^{n} \frac{dy_i^*(\lambda)}{dy} \\ &= \prod_{i=1}^{n} \frac{\lambda e^{\lambda y_i}}{\lambda} \\ &= \left(\prod_{i=1}^{n} e^{y_i}\right)^{\lambda} \\ &= \bar{y}_M^{\lambda n} \\ &= e^{\lambda n \bar{y}}. \end{aligned}$$

The scaled transformation is given by:

$$\begin{aligned} z_i^*(\lambda) &= \frac{e^{\lambda y_i} - 1}{\lambda} \frac{1}{\bar{y}_M^{\lambda}} \\ &= \frac{e^{\lambda y_i} - 1}{\lambda} \frac{1}{e^{\lambda \bar{y}}}. \end{aligned}$$

The inverse function of the Manly(A) transformation is denoted as:

$$\begin{aligned} f(y_i) &= \frac{e^{\lambda y_i} - 1}{\lambda} \\ y_i^* &= \frac{e^{\lambda y_i} - 1}{\lambda} \\ \lambda y_i^* + 1 &= e^{\lambda y_i} \\ y_i &= \frac{\log(\lambda y_i^* + 1)}{\lambda} \\ \Rightarrow f^{-1}(y_i^*) &= \frac{\log(\lambda y_i^* + 1)}{\lambda}. \end{aligned}$$

Manly transformation case (B)

The variable remains equal, $y_i^* = y_i$.

A.11. Modulus transformation

$$y_i^*(\lambda) = \begin{cases} \text{sign}(y_i) \frac{(|y_i|+1)^\lambda - 1}{\lambda} & \text{if } \lambda \neq 0 \quad (A); \\ \text{sign}(y_i) \log(|y_i| + 1) & \text{if } \lambda = 0 \quad (B). \end{cases}$$

Modulus transformation case (A)

Let $J(\lambda, y)$ denote the Jacobian of a transformation from y_i to $y_i^*(\lambda)$. In order to obtain $z_i^*(\lambda)$, the scaled modulos(A) transformation, given by $\frac{y_i^*(\lambda)}{J(\lambda,y)^{1/n}}$, and for simplicity, we use a modification of the definition of the geometric mean, denoted by $\bar{y}_{\text{MA}}$. Therefore, the Jacobian, the scaled, and the inverse of the modulus(A) transformation are given below.

The modulus(A) transformation presented in Table 2 is defined as:

$$y_i^*(\lambda) = \text{sign}(y_i) \frac{(|y_i| + 1)^\lambda - 1}{\lambda} \text{ if } \lambda \neq 0.$$

The modification of the definition of the geometric mean for this transformation is:

$$\bar{y}_{MA} = \left[\prod_{i=1}^{n} |y_i| + 1 \right]^{\frac{1}{n}}.$$

Therefore, the expression of the Jacobian comes to:

$$\begin{aligned} J(\lambda, \boldsymbol{y}) &= \prod_{i=1}^{n} \frac{dy_i^*(\lambda)}{dy} \\ &= \prod_{i=1}^{n} \frac{\text{sign}(y_i)\lambda(|y_i|+1)^{\lambda-1}}{\lambda} \\ &= \text{sign}\Big(\prod_{i=1}^{n} y_i\Big)\Big(\prod_{i=1}^{n} |y_i|+1\Big)^{\lambda-1} \\ &= \text{sign}\Big(\prod_{i=1}^{n} y_i\Big)\bar{y}_{MA}^{n(\lambda-1)}. \end{aligned}$$

The scaled transformation is given by:

$$z_i^*(\lambda) = \text{sign}(y_i)\frac{(|y_i|+1)^{\lambda}-1}{\lambda}\frac{1}{\text{sign}\Big(\prod_{i=1}^{n} y_i\Big)\bar{y}_{MA}^{(\lambda-1)}}.$$

The inverse function of the modulus(A) transformation is denoted as:

$$\begin{aligned} f(y_i) &= \text{sign}(y_i)\frac{(|y_i|+1)^{\lambda}-1}{\lambda} \\ y_i^* &= \text{sign}(y_i)\frac{(|y_i|+1)^{\lambda}-1}{\lambda} \\ |y_i| &= \big[\text{sign}(y_i^*)\lambda+1\big]^{\frac{1}{\lambda}}-1 \\ \Rightarrow f^{-1}(y_i^*) &= \pm\big[\big(\text{sign}(y_i^*)\lambda+1\big)^{\frac{1}{\lambda}}-1\big]. \end{aligned}$$

Modulus transformation case (B)

This case is exactly equal to the neglog transformation case.

A.12. Dual power transformation

$$y_i^*(\lambda) = \begin{cases} \frac{y_i^{\lambda}-y_i^{-\lambda}}{2\lambda} & \text{if } \lambda > 0 \quad (A); \\ \log(y_i) & \text{if } \lambda = 0 \quad (B). \end{cases}$$

Dual power transformation case (A)

Let $J(\lambda, y)$ denote the Jacobian of a transformation from y_i to $y_i^*(\lambda)$. In order to obtain $z_i^*(\lambda)$, the scaled dual power(A) transformation, given by $\frac{y_i^*(\lambda)}{J(\lambda,y)^{1/n}}$, and for simplicity, we use a modification of the definition of the geometric mean, denoted by $\bar{y}_{\text{DA}}$. Therefore, the

Jacobian, the scaled, and the inverse of the dual power(A) transformation are given below. The dual power(A) transformation presented in Table 2 is defined as:

$$y_i^*(\lambda) = \frac{y_i^\lambda - y_i^{-\lambda}}{2\lambda} \text{ if } \lambda > 0.$$

The modification of the definition of the geometric mean for this transformation is:

$$\bar{y}_{DA} = \left[\prod_{i=1}^{n}\left(y_i^{\lambda-1} + y_i^{-\lambda-1}\right)\right]^{\frac{1}{n}}.$$

Therefore, the expression of the Jacobian comes to:

$$\begin{aligned} J(\lambda, \boldsymbol{y}) &= \prod_{i=1}^{n} \frac{dy_i^*(\lambda)}{dy} \\ &= \prod_{i=1}^{n} \frac{\lambda y_i^{\lambda-1} + \lambda y_i^{-\lambda-1}}{2\lambda} \\ &= \frac{1}{2}\bar{y}_{DA}^n. \end{aligned}$$

The scaled transformation is given by:

$$z_i^*(\lambda) = \frac{y_i^\lambda - y_i^{-\lambda}}{2\lambda}\frac{2}{\bar{y}_{DA}}.$$

The inverse function of the dual power(A) transformation is found by solving the quadratic by completing the square as:

$$\begin{aligned} f(y_i) &= \frac{y_i^\lambda - y_i^{-\lambda}}{2\lambda} \\ y_i^* &= \frac{y_i^\lambda - y_i^{-\lambda}}{2\lambda} \\ 2\lambda y_i^* &= y_i^\lambda - y_i^{-\lambda} \\ 2\lambda y_i^* &= y_i^\lambda - \frac{1}{y_i^\lambda} \\ 2\lambda y_i^* &= \frac{y_i^{2\lambda} - 1}{y_i^\lambda} \\ 2\lambda y_i^* y_i^\lambda &= y_i^{2\lambda} - 1 \\ 1 + \lambda^2 y_i^{*2} &= y_i^{2\lambda} - 2\lambda y_i^* y_i^\lambda + \lambda^2 y_i^{*2} \\ 1 + \lambda^2 y_i^{*2} &= (y_i^\lambda - \lambda y_i^*)^2 \\ \sqrt{1 + \lambda^2 y_i^{*2}} + \lambda y_i^* &= y_i^\lambda \\ y_i &= \left[\sqrt{1 + \lambda^2 y_i^{*2}} + \lambda y_i^*\right]^{\frac{1}{\lambda}} \\ \Rightarrow f^{-1}(y_i^*) &= \left[\sqrt{1 + \lambda^2 y_i^{*2}} + \lambda y_i^*\right]^{\frac{1}{\lambda}}. \end{aligned}$$

Dual power transformation case (B)

This case is exactly equal to the Box-Cox (shift) transformation, case (B).

A.13. Gpower transformation

$$y_i^*(\lambda) = \begin{cases} \frac{\left(y_i+\sqrt{y_i^2+1}\right)^{\lambda}-1}{\lambda} & \text{if } \lambda \neq 0 \quad (A); \\ \log\left(y_i + \sqrt{y_i^2+1}\right) & \text{if } \lambda = 0 \quad (B). \end{cases}$$

Gpower transformation case (A)

Let $J(\lambda, y)$ denote the Jacobian of a transformation from y_i to $y_i^*(\lambda)$. In order to obtain $z_i^*(\lambda)$, the scaled gpower(A) transformation, given by $\frac{y_i^*(\lambda)}{J(\lambda,y)^{1/n}}$, and for simplicity, we use a modification of the definition of the geometric mean, denoted by $\bar{y}_{\text{GA}}$. Therefore, the Jacobian, the scaled, and the inverse of the gpower(A) transformation are given below.

The gpower(A) transformation presented in Table 2 is defined as:

$$y_i^*(\lambda) = \frac{\left[y_i + \sqrt{y_i^2+1}\right]^{\lambda}-1}{\lambda} \text{ if } \lambda \neq 0.$$

The modification of the definition of the geometric mean for this transformation is:

$$\bar{y}_{GA} = \left[\prod_{i=1}^{n}\left(y_i + \sqrt{y_i^2+1}\right)^{\lambda-1}\left(1 + \frac{y_i}{\sqrt{y_i^2+1}}\right)\right]^{\frac{1}{n}}.$$

Therefore, the expression of the Jacobian comes to:

$$\begin{aligned} J(\lambda, \boldsymbol{y}) &= \prod_{i=1}^{n} \frac{dy_i^*(\lambda)}{dy} \\ &= \prod_{i=1}^{n} \frac{\lambda\left(y_i + \sqrt{y_i^2+1}\right)^{\lambda-1}\left(1 + \frac{2y_i}{2\sqrt{y_i^2+1}}\right)}{\lambda} \\ &= \bar{y}_{GA}^{n}. \end{aligned}$$

The scaled transformation is given by:

$$z_i^*(\lambda) = \frac{\left[y_i + \sqrt{y_i^2+1}\right]^{\lambda}-1}{\lambda}\frac{1}{\bar{y}_{GA}}.$$

The inverse function of the gpower(A) transformation is denoted as:

$$
\begin{aligned}
f(y_i) &= \frac{\left[y_i + \sqrt{y_i^2 + 1}\right]^{\lambda} - 1}{\lambda} \\
y_i^* &= \frac{\left[y_i + \sqrt{y_i^2 + 1}\right]^{\lambda} - 1}{\lambda} \\
\lambda y_i^* + 1 &= \left[y_i + \sqrt{y_i^2 + 1}\right]^{\lambda} \\
(\lambda y_i^* + 1)^{\frac{1}{\lambda}} &= y_i + \sqrt{y_i^2 + 1} \\
\left[(\lambda y_i^* + 1)^{\frac{1}{\lambda}} - y_i\right]^2 &= \left[\sqrt{y_i^2 + 1}\right]^2 \\
(\lambda y_i^* + 1)^{\frac{2}{\lambda}} - 2y_i(\lambda y_i^* + 1)^{\frac{1}{\lambda}} + y_i^2 &= y_i^2 + 1 \\
-y_i(\lambda y_i^* + 1)^{\frac{1}{\lambda}} &= \frac{1 - (\lambda y_i^* + 1)^{\frac{2}{\lambda}}}{2} \\
y_i &= -\left[\frac{1 - (\lambda y_i^* + 1)^{\frac{2}{\lambda}}}{2(\lambda y_i^* + 1)^{\frac{1}{\lambda}}}\right] \\
\Rightarrow f^{-1}(y_i) &= -\left[\frac{1 - (\lambda y_i + 1)^{\frac{2}{\lambda}}}{2(\lambda y_i + 1)^{\frac{1}{\lambda}}}\right].
\end{aligned}
$$

Gpower transformation case (B)

This case is exactly equal to the glog transformation case.

References

Bartlett MS (1947). "The use of transformations." *Biometrics*, **3**(1), 39–52.

Berry WD (1993). *Understanding Regression Assumptions*. SAGE Publications, Thousand Oaks.

Bickel PJ, Doksum KA (1981). "An analysis of transformations revisited." *Journal of the American Statistical Association*, **76**(374), 296–311.

Box GEP, Cox DR (1964). "An analysis of transformations." *Journal of the Royal Statistical Society Series B*, **26**(2), 211–252.

Breusch TS, Pagan AR (1979). "A simple test for heteroscedasticity and random coefficient variation." *Econometrica*, **47**(5), 1287–1294.

Croissant Y (2016). ***Ecdat**: Data Sets for Econometrics*. R package version 0.3-1, URL https://CRAN.R-project.org/package=Ecdat.

Dag O, Asar O, Ilk O (2017). ***AID**: Box-Cox Power Transformation*. R package version 2.3, URL https://CRAN.R-project.org/package=AID.

Durbin BP, Hardin JS, Hawkins DM, Rocke DM (2002). "A variance-stabilizing transformation for gene-expression microarray data." *Bioinformatics*, **18**(1), 105–110.

Feng Q, Hannig J, Marron JS (2016). "A Note on Automatic Data Transformation." *Stat*, **5**(1), 82–87.

Fernandez ES (2014). ***Johnson**: Johnson Transformation*. R package version 1.4, URL https://CRAN.R-project.org/package=Johnson.

Fox J, Weisberg S (2011). *An R Companion to Applied Regression*. SAGE Publications, Thousand Oaks.

Hossain MZ (2011). "The use of Box-Cox transformation technique in economic and statistical analyses." *Journal of Emerging Trends in Economics and Management Sciences*, **2**(1), 32–39.

Hothorn T, Möst L, Bühlmann P (2018). "Most likely transformations." *Scandinavian Journal of Statistics*, **45**(1), 110–134.

John JA, Draper NR (1980). "An alternative family of transformations." *Journal of the Royal Statistical Society Series C*, **29**(2), 190–197.

Kelmansky DM, Martínez EJ, Leiva V (2013). "A new variance stabilizing transformation for gene expression data analysis." *Statistical Applications in Genetics and Molecular Biology*, **12**(6), 653–666.

Kuhn M (2008). "Building Predictive Models in R Using the **caret** Package." *Journal of Statistical Software*, **28**(5), 1–26.

Manly BFJ (1976). "Exponential data transformations." *Journal of the Royal Statistical Society Series D*, **25**(1), 37–42.

Medina L, Castro P, Kreutzmann AK, Rojas-Perilla N (2018). ***trafo**: Estimation, Comparison and Selection of Transformations*. R package version 1.0.0, URL https://CRAN.R-project.org/package=trafo.

Molina I, Marhuenda Y (2015). "**sae**: An R Package for Small Area Estimation." *The R Journal*, **7**(1), 81–98. URL https://journal.r-project.org/archive/2015/RJ-2015-007/index.html.

Morozova M, Koschutnig K, Klein E, Wood G (2016). "Monotonic Non-linear Transformations as a Tool to Investigate Age-related Effects on Brain White Matter Integrity: A Box-Cox Investigation." *NeuroImage*, **125**, 1119–1130.

Nelder JA, Wedderburn RWM (1972). "Generalized linear models." *Journal of the Royal Statistical Society Series A*, **135**(3), 370–384.

R Core Team (2018). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna. URL https://www.R-project.org/.

Ribeiro Jr PJ, Diggle PJ (2016). "**geoR**: Analysis of Geostatistical Data." *R News*, **1**(2), 15–18. URL http://cran.R-project.org/doc/Rnews.

Rojas-Perilla N (2018). *The Use of Data-Driven Transformations and their Application in Small Area Estimation*. Ph.D. thesis, Freie Universität Berlin. Unpublished.

Royston P, Lambert PC, *et al.* (2011). *Flexible Parametric Survival Analysis using Stata: Beyond the Cox Model*. Stata Press, College Station.

Shapiro SS, Wilk MB (1965). "An Analysis of Variance Test for Normality." *Biometrika*, **52**(3/4), 591–611.

Taylor JMG (1985). "Power transformations to symmetry." *Biometrika*, **72**(1), 145–152.

Tukey JW (1977). *Exploratory Data Analysis*. Pearson, London.

Venables WN, Ripley BD (2002). *Modern Applied Statistics with S*. Springer-Verlag, New York.

Wang Y (2015). ***jtrans**: Johnson Transformation for Normality*. R package version 1.1.0, URL https://CRAN.R-project.org/package=jtrans.

Whittaker J, Whitehead C, Somers M (2005). "The neglog transformation and quantile regression for the analysis of a large credit scoring database." *Journal of the Royal Statistical Society Series C*, **54**(4), 863–878.

Yang Z (2006). "A modified family of power transformations." *Economics Letters*, **92**(1), 14–19.

Yeo IK, Johnson RA (2000). "A new family of power transformations to improve normality or symmetry." *Biometrika*, **87**(4), 954–959.

Affiliation:

Ann-Kristin Kreutzmann
Institute for Statistics and Econometrics
Faculty of Economics
Freie Universität Berlin
14195 Berlin, Germany
E-mail: ann-kristin.kreutzmann@fu-berlin.de
URL: http://www.wiwiss.fu-berlin.de/fachbereich/vwl/Schmid/Team/Kreutzmann/index.html