`robustlmm`: Robust Estimating Equations and Examples

Manuel Koller

December 18, 2012

This vignette is an excerpt of my dissertation (Koller, 2013). The text has sometimes been slightly altered to avoid references to omitted material. A more detailed development of the method is given in Koller (2013).

The model and some notation is introduced in Section 1. The robust estimating equations are given in Section 2. Then two examples are analyzed in Sections 3 and 4. Both datasets are also analyzed in Bates (2011). For the convenience of the reader, the description of the datasets found there has been included here. Finally, in Section 5 we provide tables of tuning constants and the definition of the *smoothed Huber ψ-function.*

1 The Model

Following Bates (2011), we work with the formulation of mixed effects models in terms of spherical random effects,

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{Z}\boldsymbol{U}_b\left(\boldsymbol{\theta}\right)\boldsymbol{b}^* + \boldsymbol{U}_e\boldsymbol{\varepsilon}^* \, ,$$
$$\boldsymbol{b}^* \sim \mathcal{N}\left(\boldsymbol{0}, \sigma^2\boldsymbol{I}_q\right) \, , \quad \boldsymbol{\varepsilon}^* \sim \mathcal{N}\left(\boldsymbol{0}, \sigma^2\boldsymbol{I}_n\right) \, , \quad \boldsymbol{b}^* \perp \boldsymbol{\varepsilon}^* \, , \tag{1}$$

where q is the number of random effects (the length of $\boldsymbol{b}^*$) and n is the number of observations (length of $\boldsymbol{y}$). The vector $\boldsymbol{\theta}$ parametrizes the lower triangular part $\boldsymbol{U}_b\left(\boldsymbol{\theta}\right)$ of the Cholesky decomposition of the covariance matrix of the random effects $\boldsymbol{V}_b\left(\boldsymbol{\theta}\right)$, i.e., $\boldsymbol{V}_b\left(\boldsymbol{\theta}\right) = \boldsymbol{U}_b\left(\boldsymbol{\theta}\right)\boldsymbol{U}_b\left(\boldsymbol{\theta}\right)^\mathsf{T}$. We assume the random effects to be ordered in a way such that correlated random effects are grouped togheter. The covariance matrix $\boldsymbol{V}_b\left(\boldsymbol{\theta}\right)$ is therefore block diagonal. We use the index i for observations, j for random effects and k for blocks of random effects. For notatinal simplicity, we assume that there is only one type of blocks. Generalizing the estimating equations given below is straightforward and left to the reader.

2 Robust Estimating Equations

2.1 Fixed and Random Effects

Let $k\left(j\right)$ be a function that maps random effect j to the corresponding block k, then the squared Mahalanobis distances of the estimated random effects are

$$\boldsymbol{d} = \left(d\left(b_{k(j)}/\sigma\right)\right)_{j=1,\dots,q} \, , \quad \text{where} \quad d(b_k) = \boldsymbol{b}_k^{*\mathsf{T}}\boldsymbol{b}_k^* \, .$$

Then we may define the robustness weight for the jth random effect as $w_b(d_j)$. We use standard (location and linear regression) robustness weights:

$$w_b(d) = \left\{ \begin{array}{ll} \psi_b\left(\sqrt{d}\right)/\sqrt{d} & \text{if } d \neq 0 \, , \\ \psi_b'(0) & \text{if } d = 0 \, . \end{array} \right.$$

It is convenient to represent the robustness weights as (diagonal) weighting matrix,

$$\mathbf{W}_b\left(\boldsymbol{d}\right) = \mathbf{Diag}\left(w_b\left(d_{k(j)}\right)\right)_{j=1,\dots,q} \, .$$

The robust estimating equations are then

$$X^\mathsf{T} U_e^{-\mathsf{T}} \psi_e(\widehat{\varepsilon}^*/\sigma) = 0 \, ,$$

$$U_b^\mathsf{T} Z^\mathsf{T} U_e^{-\mathsf{T}} \psi_e(\widehat{\varepsilon}^*/\sigma) - \Lambda_b W_b\left(\widehat{d}\right) \widehat{b}^*/\sigma = 0 \, , \tag{2}$$

where $\Lambda_b = \mathbf{Diag}(\lambda_e/\lambda_{b,j})_{j=1,\dots,q}$ is a diagonal matrix with elements depending on the block size $s_{k(j)}$, $\lambda_e = \mathbb{E}_0\left[\psi_e'(\varepsilon^*)\right]$ and

$$\lambda_{b,j} = \mathbb{E}_0\left[\frac{\partial}{\partial b_j} w_b\left(d_{k(j)}\right) b_j/\sigma\right] \, .$$

2.2 Scale

We apply the Design Adaptive Scale approach following Koller and Stahel (2011). We get

$$\sum_{i=1}^n \tau_{e,i}^2 w_e^{(\sigma)}\left(\frac{\widehat{\varepsilon}_i^*}{\tau_{e,i}\widehat{\sigma}}\right)\left[\left(\frac{\widehat{\varepsilon}_i^*}{\tau_{e,i}\widehat{\sigma}}\right)^2 - \kappa_e^{(\sigma)}\right] = 0 \, , \tag{3}$$

where the superscript $\cdot^{(\sigma)}$ is used to distinguish the weighting functions used for the scale and covariance parameters and the ones for the fixed effects. Just as in the linear regression case, we define $\tau_{e,i}$ as the value that zeroes the expectation of the i-th summand in (3), where the distribution of the residuals is approximated using a linear expansion of $\widehat{\beta}$ and $\widehat{b}^*$ around their true values. The expectation is

$$\mathbb{E}\left[w_e^{(\sigma)}\left(\frac{\widehat{\varepsilon}_i^*}{\tau_{e,i}\widehat{\sigma}}\right)\left(\frac{\widehat{\varepsilon}_i^*}{\tau_{e,i}\widehat{\sigma}}\right)^2 - \kappa_e^{(\sigma)} w_e^{(\sigma)}\left(\frac{\widehat{\varepsilon}_i^*}{\tau_{e,i}\widehat{\sigma}}\right)\right] = 0 \, , \tag{4}$$

and $\kappa_e^{(\sigma)}$ as

$$\kappa_e^{(\sigma)} = \mathbb{E}_0\left[w_e^{(\sigma)}(\varepsilon)\,\varepsilon^2\right]\Big/\mathbb{E}_0\left[w_e^{(\sigma)}(\varepsilon)\right] \, .$$

The weighting functions used for the scale estimates are the squared robustness weights used for the estimation of the fixed and random effects, $w_e^{(\sigma)}(x) = (\psi_e^{(\sigma)}(x)/x)^2$, $w_e^{(\sigma)}(0) = \psi_e^{(\sigma)\prime}(0)$, for convex ρ-functions. For redescending ρ-functions, it is not necessary to use the squared robustness weights, using the same weights as for the fixed and random effects still gives robust estimates (assuming $\psi(x)x$ is bounded). When using the squared weights, it is crucial to use a different set of tuning parameters for estimating the scale and covariance parameters.

2.3 Covariance Parameters

For the covariance parameters, we have to treat the diagonal and the non-diagonal U_b case separately.

2.3.1 Diagonal Case

In the case of diagonal $U_b(\theta)$, the estimation of $\widehat{\theta}$ is essentially a scale estimation problem on $\widehat{b}^*$. It can be robustified just like the estimating equation for $\hat{\sigma}$ (3). The robust estimating equations are

$$\sum_{j=1}^q \tau_{b,j}^2 w_b^{(\sigma)}\left(\frac{\widehat{b}_j^*}{\tau_{b,i}\widehat{\sigma}}\right)\left[\left(\frac{\widehat{b}_j^*}{\tau_{b,j}\widehat{\sigma}}\right)^2 - \kappa_b^{(\sigma)}\right] = 0 \, , \tag{5}$$

with $\tau_{b,i}$ such that

$$\mathbb{E}\left[w_b^{(\sigma)}\left(\frac{\widehat{b}_i^*}{\tau_{b,i}\widehat{\sigma}}\right)\left(\frac{\widehat{b}_i^*}{\tau_{b,i}\widehat{\sigma}}\right)^2 - \kappa_b^{(\sigma)} w_b^{(\sigma)}\left(\frac{\widehat{b}_i^*}{\tau_{b,i}\widehat{\sigma}}\right)\right] = 0 \, ,$$

2

and normalizing constant

$$\kappa_b^{(\sigma)} = \mathbb{E}_0 \left[w_b^{(\sigma)}(b^*)\, b^{*2} \right] \Big/ \mathbb{E}_0 \left[w_b^{(\sigma)}(b^*) \right] .$$

2.3.2 Non-diagonal Case

For non-diagonal $\boldsymbol{U}_b(\boldsymbol{\theta})$ we have to take care of the block structure. The normalizing constant $\tau_{b,i}^2$ has to be replaced by a matrix $\boldsymbol{T}_{b,k}$ which is defined for each block k. Analogue to the estimator for the covariance matrix and location problem, we have to use two different weight functions, one for the size of the matrix $w_b^{(\tau)}$ and another one for the shape $w_b^{(\eta)}$. For details, we refer to Stahel (1987) and Hampel et al. (1986, Chapter 5). As is done in the cited references, we introduce a third weight function $w_b^{(\delta)}$ to simplify notation. For block types with dimension $s > 1$, let

$$w_b^{(\delta)}(d) = \left(d w_b^{(\eta)}(d) - \left(d - s\kappa_b^{(\tau)} \right) w_b^{(\tau)} \left(d - s\kappa_b^{(\tau)} \right) \right) \Big/ s ,$$

where $\kappa_b^{(\tau)}$ is defined such that

$$\mathbb{E} \left[\left(u - s\kappa_b^{(\tau)} \right) w_b^{(\tau)} \left(u - s\kappa_b^{(\tau)} \right) \right] = 0 ,$$

for $u \sim \chi_s^2$.

The robust estimating equation in the non-diagonal case can then be defined as follows. For $l = 1, \ldots, r$,

$$\sum_{k=1}^{K} \left[w_b^{(\eta)} \left(d \left(\boldsymbol{T}_{b,k}^{-1/2} \widehat{\boldsymbol{b}}_k^* / \widehat{\sigma} \right) \right) \widehat{\boldsymbol{b}}_k^{*\intercal} \boldsymbol{Q}_{l,k}(\widehat{\boldsymbol{\theta}}) \widehat{\boldsymbol{b}}_k^* / \widehat{\sigma}^2 \right.$$
$$\left. - w_b^{(\delta)} \left(d \left(\boldsymbol{T}_{b,k}^{-1/2} \widehat{\boldsymbol{b}}_k^* / \widehat{\sigma} \right) \right) \operatorname{tr} \left(\boldsymbol{T}_{b,k} \boldsymbol{Q}_{l,k}(\widehat{\boldsymbol{\theta}}) \right) \right] = 0 , \tag{6}$$

where $\boldsymbol{Q}_{l,k}(\widehat{\boldsymbol{\theta}})$ is the $s \times s$ submatrix of $\boldsymbol{Q}_l(\widehat{\boldsymbol{\theta}})$ which acts on block k and $\boldsymbol{T}_{b,k}^{-1/2}$ is the inverse of any square root of the $s \times s$ matrix $\boldsymbol{T}_{b,k}$. As in the diagonal case, we define the matrix $\boldsymbol{T}_{b,k}$ such that each summand has expectation zero. For $l = 1, \ldots, r$,

$$\mathbb{E} \left[w_b^{(\eta)} \left(d \left(\boldsymbol{T}_{b,k}^{-1/2} \widehat{\boldsymbol{b}}_k^* / \sigma \right) \right) \widehat{\boldsymbol{b}}_k^{*\intercal} \boldsymbol{Q}_{l,k}(\widehat{\boldsymbol{\theta}}) \widehat{\boldsymbol{b}}_k^* / \sigma^2 \right.$$
$$\left. - w_b^{(\delta)} \left(d \left(\boldsymbol{T}_{b,k}^{-1/2} \widehat{\boldsymbol{b}}_k^* / \sigma \right) \right) \operatorname{tr} \left(\boldsymbol{T}_{b,k} \boldsymbol{Q}_{l,k}(\widehat{\boldsymbol{\theta}}) \right) \right] = 0 .$$

Remarks. The symmetric matrix $\boldsymbol{T}_{b,k}$ is fully defined for unstructured covariance matrices only, where $r = s(s+1)/2$. For other covariance matrix structures we can replace $\boldsymbol{T}_{b,k}$ by the variance of the linear approximation of $\boldsymbol{b}^*$.

Since in the classical case, the linear approximations for $\widehat{\boldsymbol{b}}^*$ and $\widehat{\boldsymbol{\varepsilon}}^*$ are exact, the estimating equation (6) reduces to the REML estimating equations. A similar argument is valid for the estimating equation for $\hat{\sigma}$ (3).

3 Penicillin Example

The dataset, shown in Figure 1, was originally published by Davies and Goldsmith (1972). They describe it as data coming from an investigation to

> assess the variability between samples of penicillin by the *B. subtilis* method. In this test method a bulk-inoculated nutrient agar medium is poured into a Petri dish of approximately 90 mm. diameter, known as a plate. When the medium has set, six small hollow cylinders or pots (about 4 mm. in diameter) are cemented onto the

surface at equally spaced intervals. A few drops of the penicillin solutions to be compared are placed in the respective cylinders, and the whole plate is placed in an incubator for a given time. Penicillin diffuses from the pots into the agar, and this produces a clear circular zone of inhibition of growth of the organisms, which can be readily measured. The diameter of the zone is related in a known way to the concentration of penicillin in the solution.

The datasets contains the measurements of 6 samples and 24 plates.
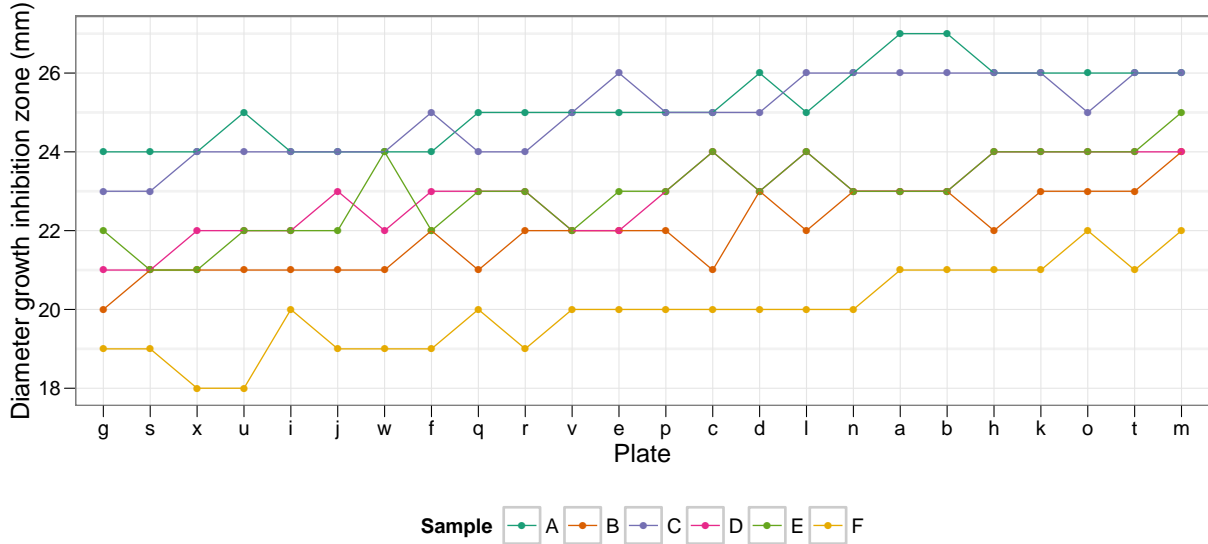

Figure 1: Diameters of growth inhibition zones of 6 samples applied to each of 24 agar plates to assess penicillin concentration in the *B. subtilis* method. The lines join the observations of the same sample. The plates have been reordered by their means.

The raw data for the Penicillin example is shown in Figure 1. In R , the data is provided as part of the R package "lme4". It is a data.frame with three columns:

```
> str(Penicillin)

'data.frame':         144 obs. of  3 variables:
$ diameter: num 27 23 26 23 23 21 27 23 ...
$ plate : Factor w/ 24 levels "g","s","x","u",..: 18 18 18 18 18 18 19 19 ...
$ sample : Factor w/ 6 levels "A","B","C","D",..: 1 2 3 4 5 6 1 2 ...
```

The column "diameter" is the response and the two factors "plate" and "sample" indicate where the observation was measured. As Bates (2011), we fit a linear mixed effects model with an intercept and two random effects for the two factors.

We fit the classical linear mixed effects model using the function "lmer" of the R package "lme4". The random effects are specified in brackets. The pipe symbol "|" is used to split the factors and covariables from the grouping variable. In this case, we only have a random intercept "1" that varies by group "plate" and "sample", respectively.

```
> st(classical <- lmer(diameter ~ 1 + (1|plate) + (1|sample),
+                     Penicillin))

   user   system elapsed
  0.309   0.000   0.314
```

The "st" function is just a shortcut to "system.time", a function that measures the time required to evaluate the expression given as argument.

4

The robust mixed effects model is fit using the function "rlmer". The call is quite similar. By default, it uses the smoothed Huber ψ-function with tuning parameter $k = 1.345$ and $s = 10$. Since we are mainly interested in the estimates for the variance components, we adjust the the tuning parameter for the $\psi_\cdot^{(\sigma)}$ functions to $k = 2.28$ and specify that squared weights are used. This can be done with one call to the function "psi2propII". Afterwards, we have a look at the summary of the fitted object.

```
> st(robust <- rlmer(diameter ~ 1 + (1|plate) + (1|sample), Penicillin,
+                    rho.sigma.e = psi2propII(smoothPsi, k = 2.28),
+                    rho.sigma.b = psi2propII(smoothPsi, k = 2.28)))
   user  system elapsed
 13.175   0.024  13.314

> summary(robust)
Robust linear mixed model fit by DAStau
Formula: diameter ~ 1 + (1 | plate) + (1 | sample)
   Data: Penicillin

Random effects:
 Groups   Name        Variance Std.Dev.
 plate    (Intercept) 0.7582   0.8707
 sample   (Intercept) 3.8865   1.9714
 Residual             0.2997   0.5475
Number of obs: 144, groups: plate, 24; sample, 6

Fixed effects:
            Estimate Std. Error t value
(Intercept)  23.0419     0.8464   27.22

Robustness weights for the residuals:
 124 weights are ~= 1. The remaining 20 ones are summarized as
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  0.397   0.670   0.837   0.809   0.955   0.993

Robustness weights for the random effects:
 25 weights are ~= 1. The remaining 5 ones are
     1     2     3    24    30
 0.836 0.836 0.938 0.802 0.858

Rho functions used for fitting:
  Residuals:
    eff: smoothed Huber (k = 1.345, s = 10)
    sig: smoothed Huber, Proposal II (k = 2.28, s = 10)
  Random Effects, variance component 1 (plate):
    eff: smoothed Huber (k = 1.345, s = 10)
    vcp: smoothed Huber, Proposal II (k = 2.28, s = 10)
  Random Effects, variance component 2 (sample):
    eff: smoothed Huber (k = 1.345, s = 10)
    vcp: smoothed Huber, Proposal II (k = 2.28, s = 10)
```

The first half of the summary shows information about the model that was fitted and displays the parameter estimates including standard errors for the fixed effects. After that, a summary of the robustness weights is shown. In this case, we can see that some of the observations have been down weighted, but practically none of the random effects. Finally there is a table that gives details about which ψ-functions were used to fit the model.

Remark. Note that the column "Std.Dev." contains the estimated standard deviances, i.e., just the square roots of the estimated variances. A common mistake is to interpret them as the standard errors of the variance component estimates. The same table is shown for the summary of an lme4 object and to ease change from "lmer" to "rlmer", we use the same convention here.

Alternatively, one might be interested in a model that does not down weight the random effects of "sample" – for example because there might be structural outliers and one is interested

in the variability including these. To enable this, "rlmer" accepts list input for the arguments "rho.b" and "rho.sigma.b". The list entries correspond to the ψ-functions used for the variance components as shown in the summary output. The call to fit a model that does uses the classical estimates for the "sample" variance components is as follows.

```
> st(robust2 <- rlmer(diameter ~ 1 + (1|plate) + (1|sample), Penicillin,
+                 rho.sigma.e = psi2propII(smoothPsi, k = 2.28),
+                 rho.b = list(smoothPsi, cPsi),
+                 rho.sigma.b = list(psi2propII(smoothPsi, k = 2.28),
+                                    cPsi)))
  user  system elapsed
 12.377   0.023  12.532
```

	classical	robust	robust2
Coefficients (Std. Error)			
(Intercept)	23 (0.809)	23 (0.846)	23 (0.806)
Variance components			
(Intercept) | plate	0.847	0.871	0.871
(Intercept) | sample	1.932	1.971	1.921
σ	0.55	0.547	0.547
REML	331		

Table 1: Comparison table of the fitted models for the Penicillin example

The results of the three fits are summarized in Table 1. The differences are minimal. Interestingly, the estimated variance for "sample" is a little smaller for "robust2" than for "robust". The common residual analysis plots, Tukey-Anscombe and qq-normal, are shown in Figure 2 for the "robust" object. The points that got a lower robustness weight are indicated by a darker color. The rest of the observations seem to follow the central model quite nicely. In Figure 3 we again show a plot of the data, this time highlighting the observations that got a low robustness weights.

4 Sleepstudy Example

This dataset is a subset of data gathered by Belenky et al. (2003) for a study of the effects of sleep deprivation time. The 18 subjects were chosen from a population of long distance drivers that were allowed to sleep for only three hours each night. Each subject's reaction time was measured several times on each day of the trial. The measurements were made over a course of 10 days. The data are shown in Figure 4.

The data.frame consists of three columns:

```
> str(sleepstudy)
'data.frame':        180 obs. of  3 variables:
$ Reaction: num 250 259 251 321 ...
$ Days : num 0 1 2 3 4 5 6 7 ...
$ Subject : Factor w/ 18 levels "309","310","370",..: 7 7 7 7 7 7 7 7 ...
```

Again, as in Bates (2011), we will fit a random intercept and slope model. This is an extension of a linear regression model with `Reaction` as response and `Days` as predictor. In this model, the coefficients for each subject are split into a population average (fixed) and a subject specific (random) part.

The calls to "lmer" and "rlmer" are quite similar. This time, we omit the optional "1" for the intercept in both the fixed and the random part. The random effect specification `(Days|Subject)` is interpreted as `(1+Days|Subject)`. Specified in this way, the fitting methods
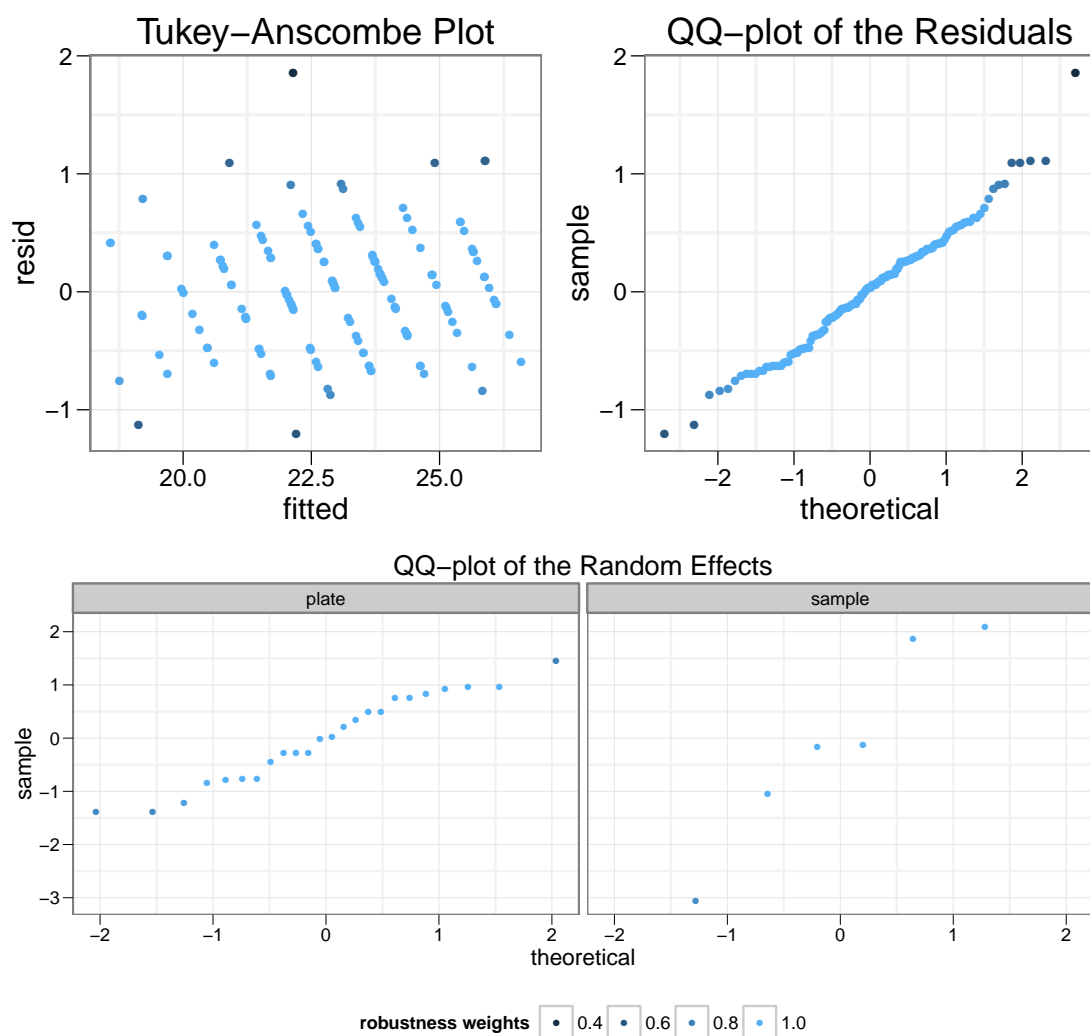
Figure 2: Residual analysis plots for the "robust" object of the Penicillin example.
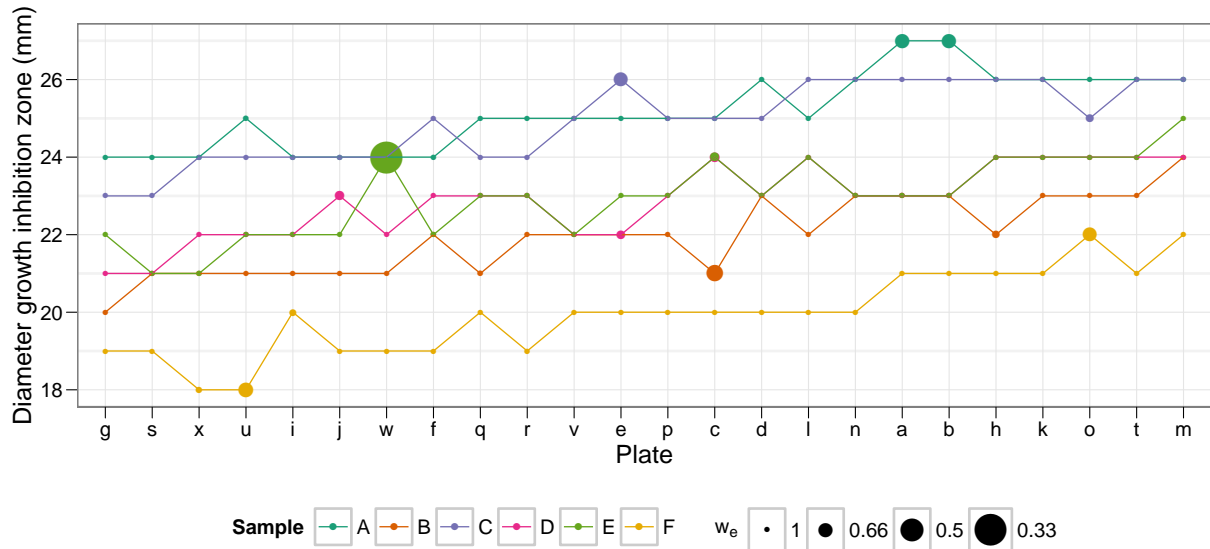
Figure 3: Diameters of growth inhibition zones of 6 samples applied to each of 24 agar plates to assess penicillin concentration in the *B. subtilis* method. The lines join the observations of the same sample. The plates have been reordered by their means. The sizes of the data points show the robustness weights.

also include a correlation term. To get uncorrelated random effects, one would have to use two terms, namely `(1|Subject) + (0+Days|Subject)`. The `0` tells the method not to include an intercept term. Since the random effects now have a non-diagonal covariance matrix $U_b(\theta)$, we have to use another tuning constant for "rho.sigma.b". It corresponds roughly to the square of the one used in the diagonal case.

```
> st(classical <- lmer(Reaction ~ Days + (Days|Subject), sleepstudy))

   user  system elapsed
  0.115   0.000   0.116

> st(robust <- rlmer(Reaction ~ Days + (Days|Subject), sleepstudy,
+                    rho.sigma.e = psi2propII(smoothPsi, k = 2.28),
+                    rho.sigma.b = chgDefaults(smoothPsi, k = 5.11, s=10)))

    user   system  elapsed
1078.412    0.133 1085.129

> summary(robust)

Robust linear mixed model fit by DAStau
Formula: Reaction ~ Days + (Days | Subject)
  Data: sleepstudy

Random effects:
 Groups   Name        Variance Std.Dev. Corr
 Subject  (Intercept) 784.05   28.001
          Days         41.68    6.456   -0.037
 Residual             404.16   20.104
Number of obs: 180, groups: Subject, 18

Fixed effects:
            Estimate Std. Error t value
(Intercept)  252.090      7.295   34.55
Days          10.827      1.646    6.58

Correlation of Fixed Effects:
     (Intr)
```
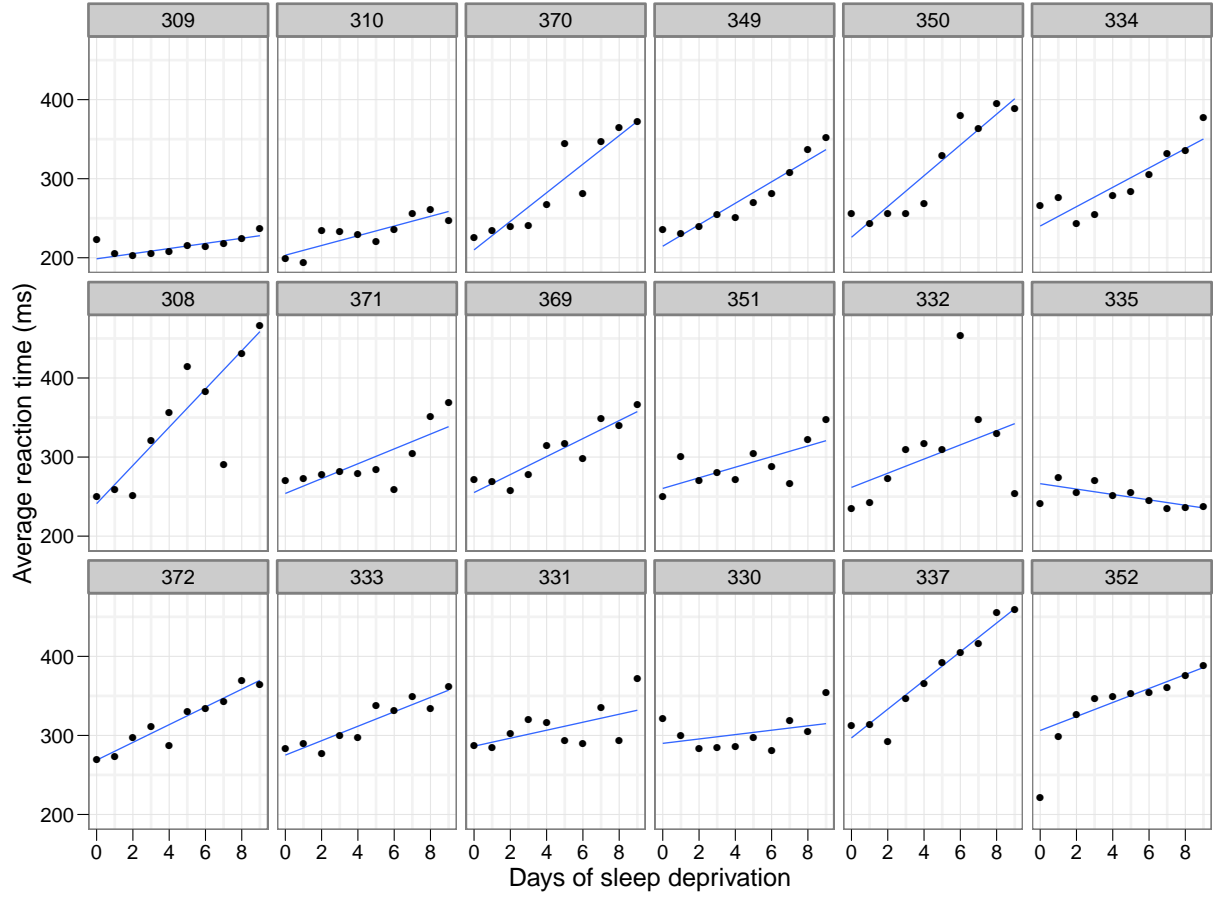
8

Figure 4: The average reaction time of subjects versus days of sleep deprivation. Each subject is shown in a separate facet. The lines show the robust linear regression fit to the subject's data. The subjects have been ordered by increasing intercept. The robust fits were computed using the method *lmrob* of the R package *robustbase* (Rousseeuw et al., 2012) using *setting="KS2011"*.

Days -0.121

Robustness weights for the residuals:
 155 weights are ˜= 1. The remaining 25 ones are summarized as
   Min. 1st Qu. Median   Mean 3rd Qu.   Max.
  0.204   0.597   0.725   0.687   0.872   0.977

Robustness weights for the random effects:
 24 weights are ˜= 1. The remaining 12 ones are summarized as
   Min. 1st Qu. Median   Mean 3rd Qu.   Max.
  0.631   0.692   0.723   0.735   0.742   0.901

Rho functions used for fitting:
  Residuals:
    eff: smoothed Huber (k = 1.345, s = 10)
    sig: smoothed Huber, Proposal II (k = 2.28, s = 10)
  Random Effects, variance component 1 (Subject):
    eff: smoothed Huber (k = 1.345, s = 10)
    vcp: smoothed Huber (k = 5.11, s = 10)

The residual analysis plots are shown in Figure 5. There are some points that are for outside the bulk of the residuals and accordingly get quite a low robustness weight. The qq-plot of the random effects shows some structure, but admittedly, the sample size of 18 is quite low.
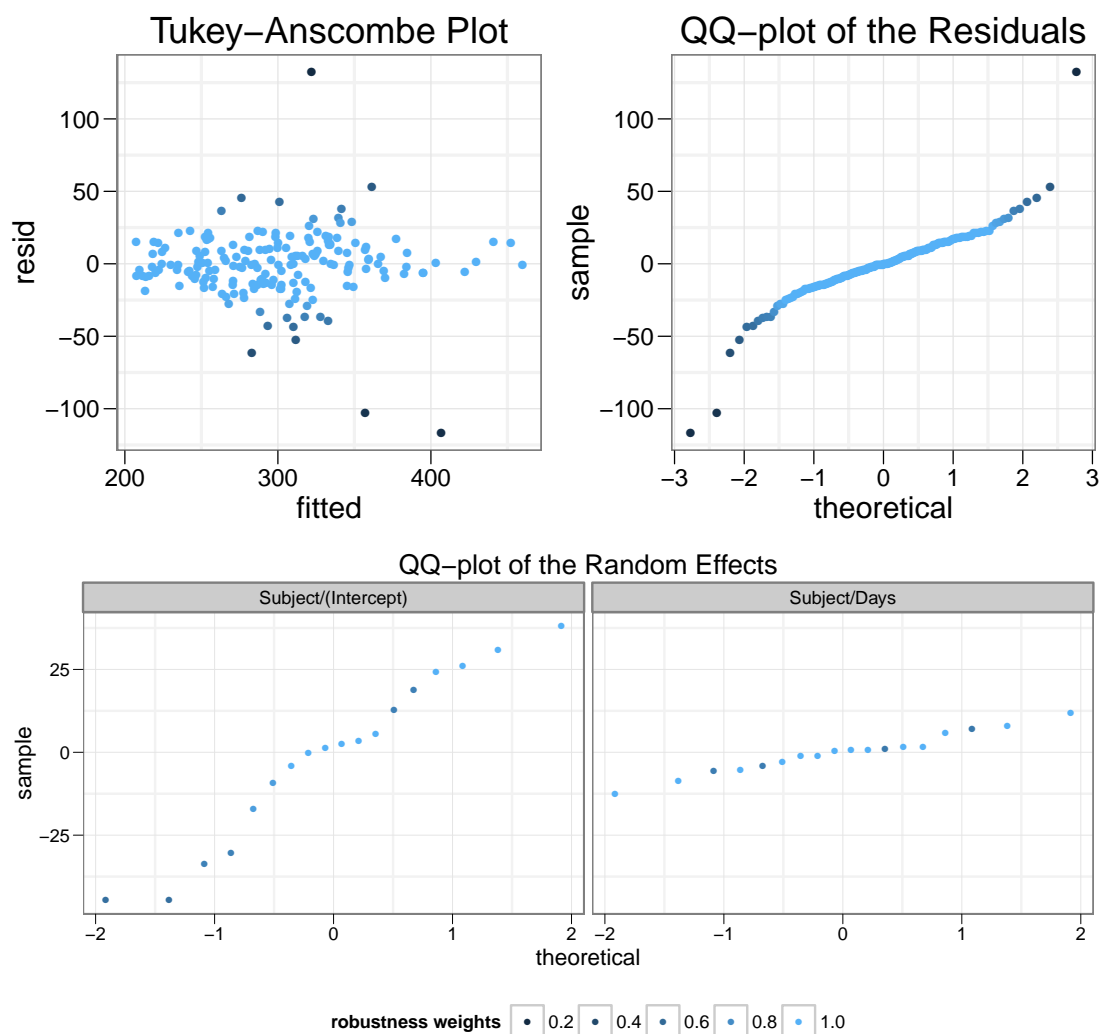

Figure 5: Residual analysis plots for robust fit "robust" of the Sleepstudy example.
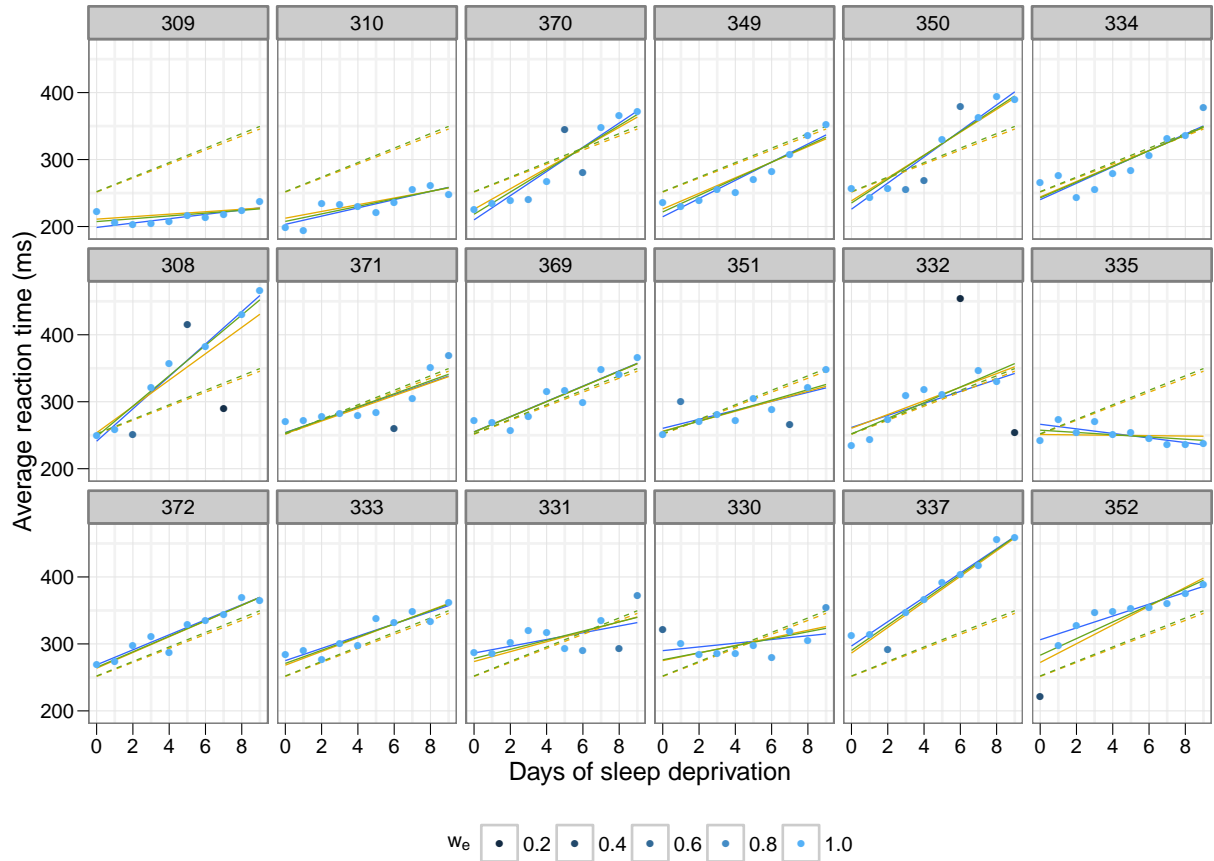
10

Figure 6: The average reaction time of subjects versus days of sleep deprivation. Each subject is shown in a separate facet. The blue lines show the robust linear regression fit to the subject's data. The yellow lines show the fitted values of the classical mixed effects model including random effects. The green lines show the corresponding robust linear mixed effects fit. The dashed lines show the population wide fit, robust and classical methods almost coincide. The subjects have been ordered by increasing intercept. The robust linear regression fits were computed using the method *lmrob* of the R package *robustbase* (Rousseeuw et al., 2012) using *setting="KS2011"*.

The fitted values for classical and the robust fits as well as the robust per-subject fit are show in Figure 6. While most of the subjects follow the general population fit quite closely, others, such as subject 335, show even a negative trend. Nevertheless, the robustness weights for the random effects do not show any clear outliers. The lowest robustness weight is assigned to subject 309 while subject 335 is given a weight of 0.69. The three subjects with the most notable difference between classical and robust fit are shown again in Figure 7. The differences in Subject 308 are most pronounced. The predicted slope of the classical fit is pulled downwards, causing the observations to lie outside or just at the border of the estimated confidence intervals. The confidence intervals for the population level estimates of the robust and classical fit are very similar. Compared to the total number of observations, there are only a very little observations with quite low robustness weight. They are most probably not able to increase the variance components estimates. We may check this statement by fitting the same model with redescending ψ-functions. As initial estimate, we may just take the values of the above robust fit. With "rlmer", one can do this conveniently by using the "update" function and specifying the new ψ-functions:

```
> st(redesc <-
+     update(robust, rho.e = chgDefaults(lqqPsi, cc=c(1.47, 0.98, 1.5)),
+             rho.sigma.e = chgDefaults(lqqPsi, cc=c(2.19, 1.46, 1.5)),
```
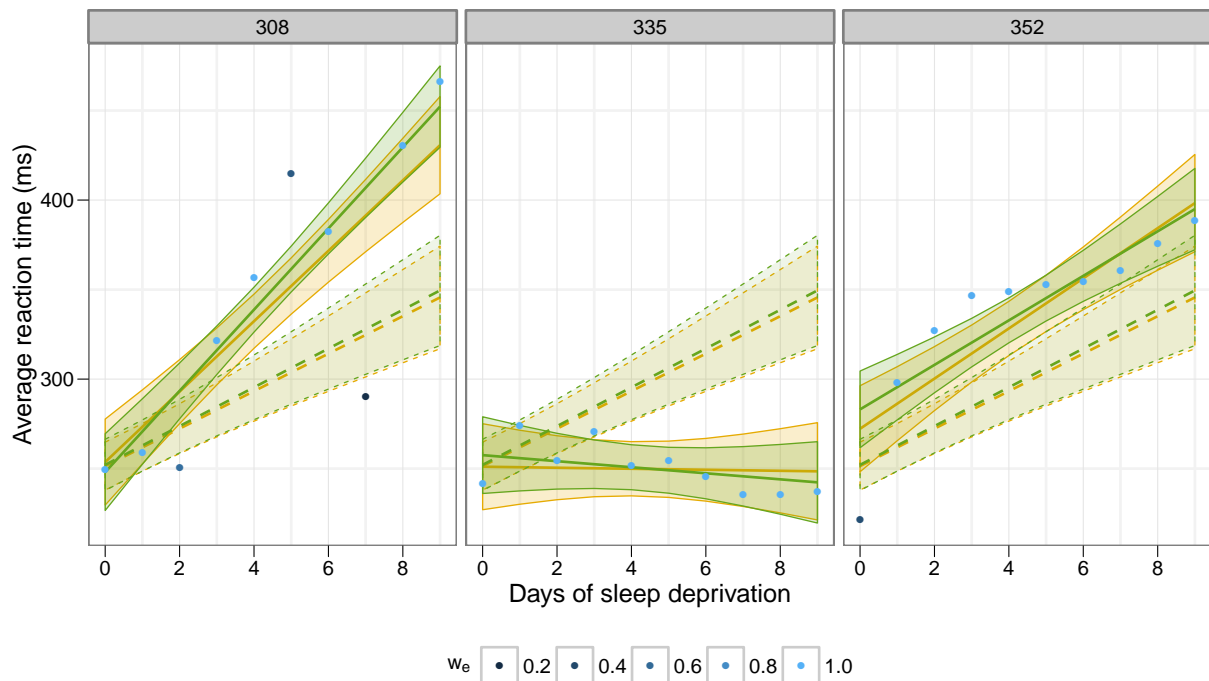
11

Figure 7: The average reaction time of subjects versus days of sleep deprivation. Additional to the features shown also in Figure 6, we show also the pointwise 95% confidence intervals for the fitted values. For better visibility, we omit the linear regression fit and show only the three subjects with the most pronounced differences between classical and robust fits.

```
+               rho.b = chgDefaults(lqqPsi, cc=c(1.47, 0.98, 1.5)),
+               rho.sigma.b = chgDefaults(lqqPsi, cc=c(5.95, 3.97, 1.5)))))
     user   system  elapsed
2811.761    0.594 2830.209

> summary(redesc)

Robust linear mixed model fit by DAStau
Formula: Reaction ~ Days + (Days | Subject)
   Data: sleepstudy

Random effects:
 Groups   Name        Variance Std.Dev. Corr
 Subject  (Intercept) 809.17   28.446
          Days          43.66    6.607   -0.069
 Residual             399.39   19.985
Number of obs: 180, groups: Subject, 18

Fixed effects:
            Estimate Std. Error t value
(Intercept)  251.775      7.435   33.86
Days          10.822      1.690    6.40

Correlation of Fixed Effects:
     (Intr)
Days -0.146

Robustness weights for the residuals:
 143 weights are ~= 1. The remaining 37 ones are summarized as
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.0118  0.6730  0.8890  0.7820  0.9800  0.9990
```

```
Robustness weights for the random effects:
 14 weights are ~= 1. The remaining 22 ones are summarized as
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  0.690   0.778   0.929   0.879   0.977   0.995

Rho functions used for fitting:
  Residuals:
    eff: lqq (cc1 = 1.47, cc2 = 0.98, cc3 = 1.5)
    sig: lqq (cc1 = 2.19, cc2 = 1.46, cc3 = 1.5)
  Random Effects, variance component 1 (Subject):
    eff: lqq (cc1 = 1.47, cc2 = 0.98, cc3 = 1.5)
    vcp: lqq (cc1 = 5.95, cc2 = 3.97, cc3 = 1.5)
```

	classical	robust	redesc
Coefficients (Std. Error)			
(Intercept)	251.4 (6.82)	252.1 (7.30)	251.8 (7.44)
Days	10.5 (1.55)	10.8 (1.65)	10.8 (1.69)
Variance components			
(Intercept) | Subject	24.74	28.00	28.45
Days | Subject	5.92	6.46	6.61
Correlations			
(Intercept) × Days | Subject	0.0656	-0.0369	-0.0695
σ	25.6	20.1	20
REML	1744		

Table 2: Comparison table of the fitted models for the Sleepstudy example

A comparison table of the three fits is shown in Table 2. The two robust fits are quire similar. The variance attributed to the between subjects differences is a little lower for the fit using redescending ψ-function, while the two estimates of the residual standard errors are almost identical. We may therefore conclude that the few observations with a small robustness weight were not able to unduly increase the estimates residual standard error. While the estimated residual standard errors are smaller for both robust fits are smaller than the classical fit, the estimated standard errors of the fixed effects estimates are a little increased (the estimated standard error for the classical fit is 6.82 for the intercept and 1.55 for "Days"). The robust fits return a negative estimate of the correlation between the random intercept and slopes. When choosing smaller tuning parameters for the functions "rho.b" and "rho.sigma.b", the correlation is estimated even lower. A scatterplot of the estimated random effects is shown in Figure 8. With help from the coloring of the points, one can see a hint of a negative correlation between the two random effects (suppressing the points below and above the falling diagonal). This is picked up by the estimator. For smaller tuning parameters, the weight function decreases more quickly and the off-diagonal points get a lower weight, finally this leads to a negative estimate of the correlation.
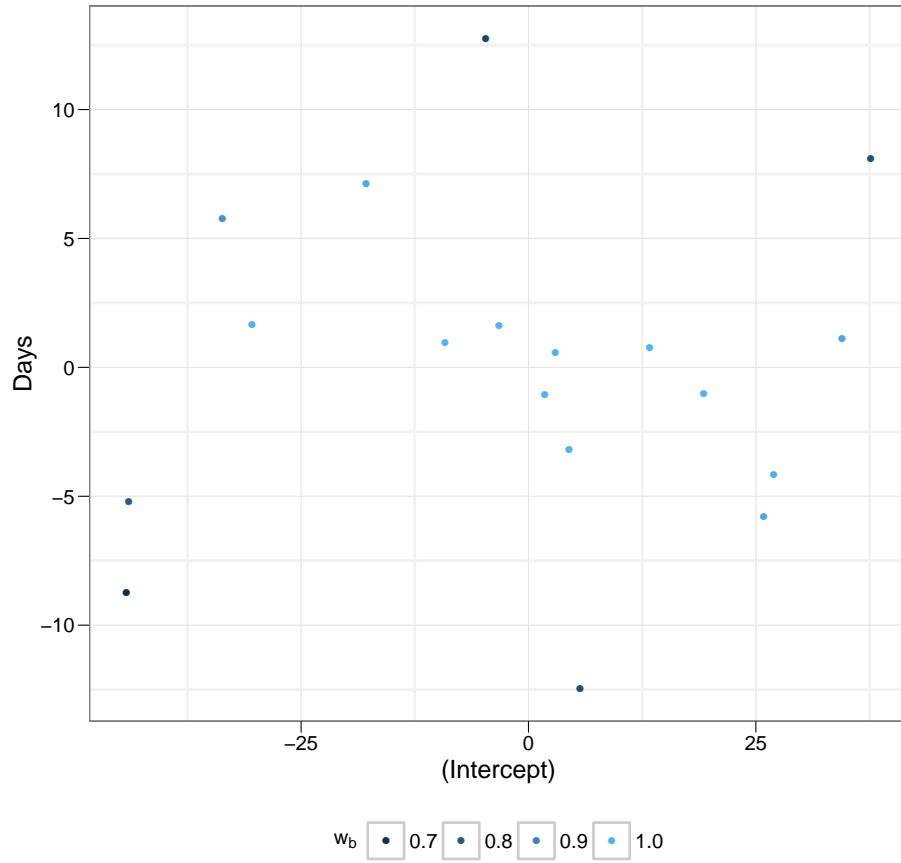
Figure 8: Scatterplot of the estimated random effects for the "redesc" fit for the Sleepstudy example using redescending lqq ψ-functions.

5 ψ-functions and Tables of Tuning Constants

The *smoothed Huber ψ-function* is defined as

$$\psi(x, k, s) = \begin{cases} x & |x| \leq c \\ \text{sign}(x) \left(k - \frac{1}{(|x|-d)^s} \right) & \text{otherwise} \end{cases} , \tag{7}$$

where $c = k - s^{\frac{-s}{s+1}}$ and $d = c - s^{\frac{1}{s+1}}$. We have always used $s = 10$ for our simulations. With this value, the asymptotic properties of the regular Huber function and the smoothed Huber function are almost identical. We can therefore safely use the same tuning parameter k for both ψ-functions. A comparison of the two ψ-functions is shown in Figure 9. Useful tables of tuning constants for this and the lqq ψ-function are given in Tables 3 to 6.
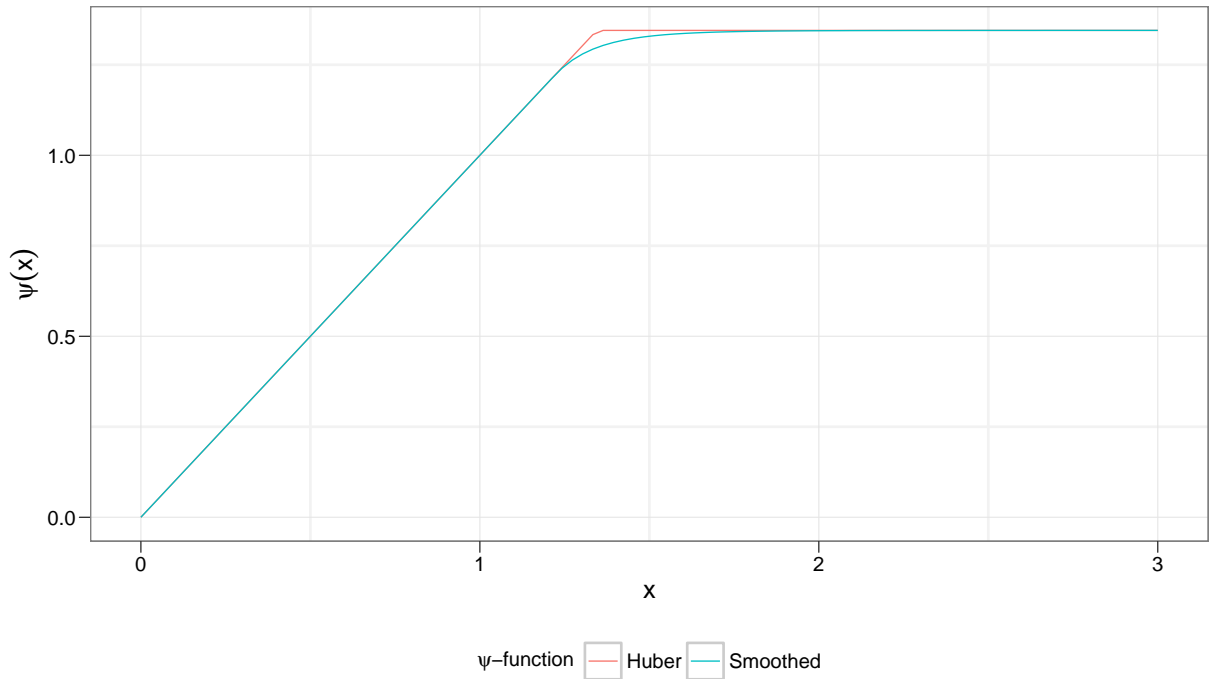

Figure 9: Comparison of the Huber and the smoothed Huber ψ-function for $k = 1.345$ and $s = 10$.

efficiency	k for $\hat{\mu}$	k for $\hat{\sigma}_{(3)}$	k for $\hat{\sigma}_{(3)}$, Prop. II
0.80	0.53	0.50	1.49
0.85	0.73	0.71	1.69
0.90	0.98	1.08	1.94
0.95	1.345	1.66	2.28

Table 3: Tuning parameters k for scale estimates such that they reach the same asymptotic efficiency as the location estimate. For the Huber ψ-function.

s	2	3	4	5	6	7
b_η	5.66	6.41	7.14	7.87	8.58	9.28
b_τ	5.15	5.55	5.91	6.25	6.55	6.84
b_μ	1.5	1.63	1.73	1.81	1.87	1.9

Table 4: Tuning parameters for the optimal B-estimator to yield 95% efficiency. For the Huber ψ-function.

efficiency	cc for $\hat{\mu}$	cc for $\hat{\sigma}_{(3)}$
0.80	(0.946,0.631)	(1.414,0.942)
0.85	(1.058,0.705)	(1.57,1.05)
0.90	(1.214,0.809)	(1.79,1.19)
0.95	(1.474,0.982)	(2.19,1.46)

Table 5: Tuning parameters for lqq ψ-function for the location and scale estimates such that they reach the given asymptotic efficiency. The third parameter is always taken to be 1.5.

s	2	3	4	5	6
cc_η	(6.44,4.29)	(7.23,4.82)	(8.01,5.34)	(8.77,5.85)	(9.52,6.35)
cc_τ	(5.95,3.97)	(6.41,4.27)	(6.82,4.55)	(7.2,4.8)	(7.55,5.03)
cc_μ	(1.63,1.09)	(1.77,1.18)	(1.88,1.26)	(1.99,1.32)	(2.08,1.39)

Table 6: Tuning parameters for the lqq weight function to yield 95% efficiency. The third parameter is always taken to be 1.5.

References

Bates, D. M. (2011). lme4: Mixed-effects modeling with R. `http://lme4.r-forge.r-project.org/book/`.

Belenky, G., Wesensten, N. J., Thorne, D. R., Thomas, M. L., Sing, H. C., Redmond, D. P., Russo, M. B., and Balkin, T. J. (2003). Patterns of performance degradation and restoration during sleep restriction and subsequent recovery: a sleep dose-response study. *Journal of Sleep Research*, 12:1–12.

Davies, O. L. and Goldsmith, P. L., editors (1972). *Statistical Methods in Research and Production*. Hafner, 4th edition.

Hampel, F., Ronchetti, E., Rousseeuw, P., and Stahel, W. (1986). *Robust Statistics: The Approach Based on Influence Functions*. John Wiley & Sons.

Koller, M. (2013). *Robust Estimation of Linear Mixed Models*. PhD thesis, ETH Zürich.

Koller, M. and Stahel, W. A. (2011). Sharpening wald-type inference in robust regression for small samples. *Computational Statistics & Data Analysis*, 55(8):2504–2515.

Rousseeuw, P., Croux, C., Todorov, V., Ruckstuhl, A., Salibian-Barrera, M., Verbeke, T., Koller, M., and Maechler, M. (2012). *robustbase: Basic Robust Statistics*. R package version 0.9-5.

Stahel, W. (1987). Estimation of a covariance matrix with location: Asymptotic formulas and optimal b-robust estimators. *Journal of multivariate analysis*, 22(2):296–312.