D MATH
Department of Mathematics
Seminar für Statistik

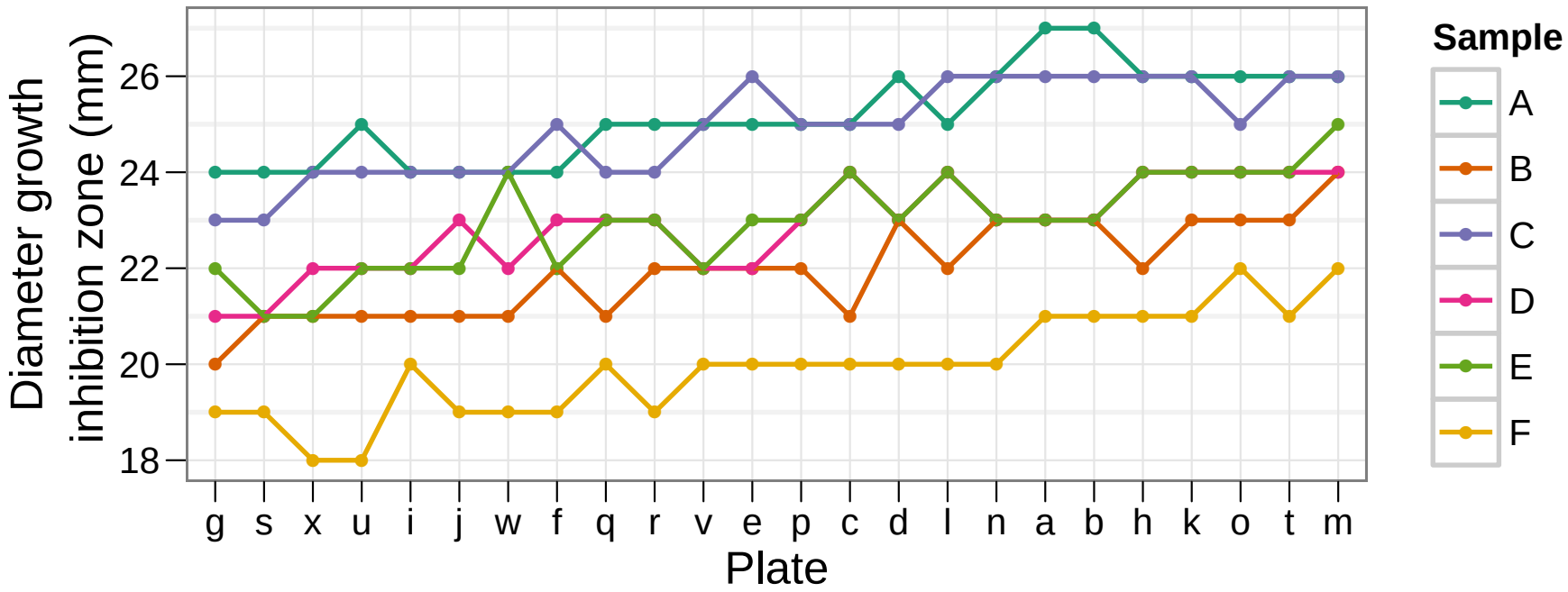
ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Robust Estimation of Linear Mixed Effects Models

Doctoral examination - Manuel Koller

Penicillin example: two-way anova (crossed)

Data from an experiment *to assess the variability between samples of penicillin by the* B. subtilis *method*.

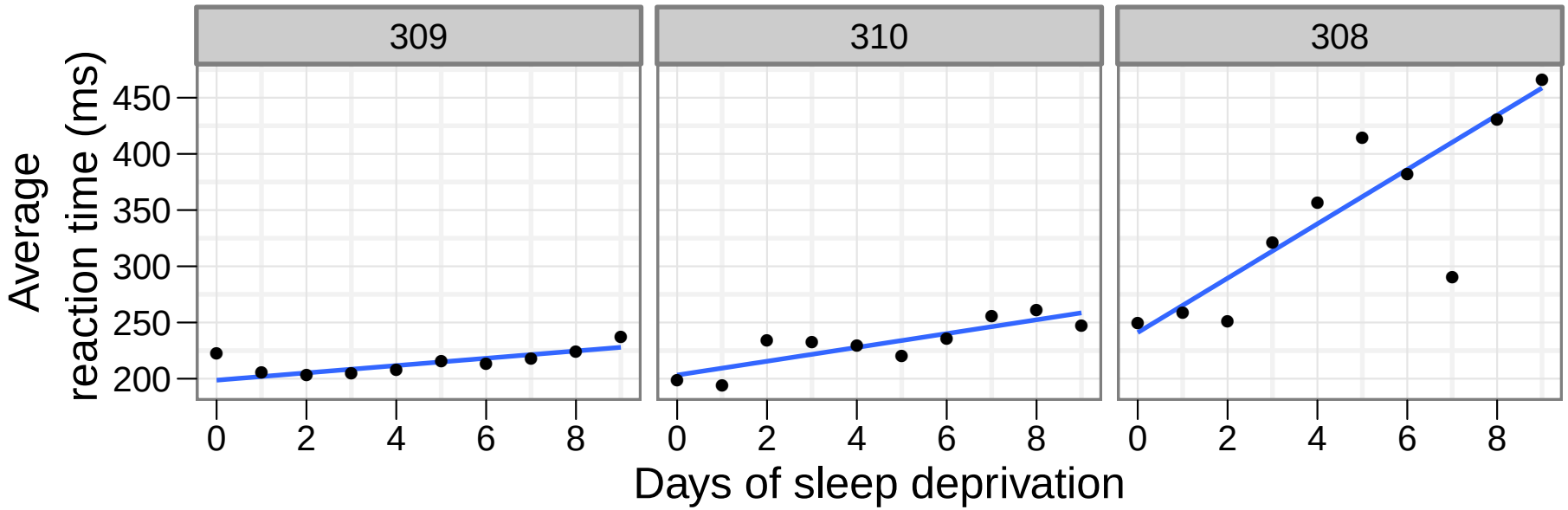


Model: $y_{ij} = \alpha + plate_i + sample_j + \varepsilon_{ij} \quad i = 1, \ldots, 24,\ j = 1, \ldots, 6.$

(Data and plots taken from Bates, 2011.)

Sleepstudy example: random intercept / slope model

Data from a study *of the effects of sleep deprivation on reaction time for a number of subjects chosen from a population of long-distance truck drivers.*

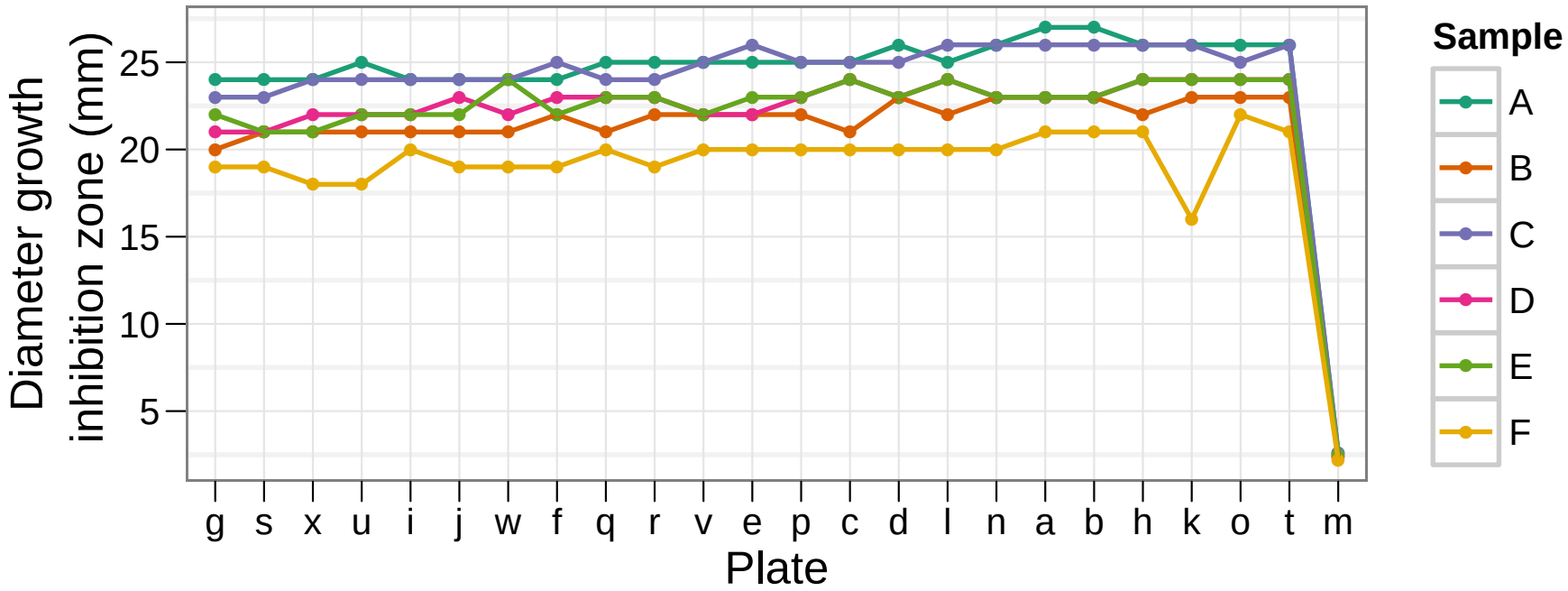


Model: $y_{ij} = (\alpha + a_j) + (\beta + b_j) \cdot days_i + \varepsilon_{ij} \quad i = 1, \ldots, 10,\ j = 1, \ldots, 18.$

(Blue line: robust linear regression fit; subset of data only.)

What is a robust method?

Most of the time we are interested in estimating a model that fits for the bulk of the data. Contaminated parts of the data, for example:



should be automatically detected and dealt with (here: outliers).
Other problems: model misspecification, auto correlated errors, ...

Goals (of the second part) of the dissertation

- Develop a robust method of estimating mixed effects models.
- It should support data such as the Penicillin and Sleepstudy examples.
- It should be able to take care of contamination on different levels.

Linear mixed effects models, matrix formulation

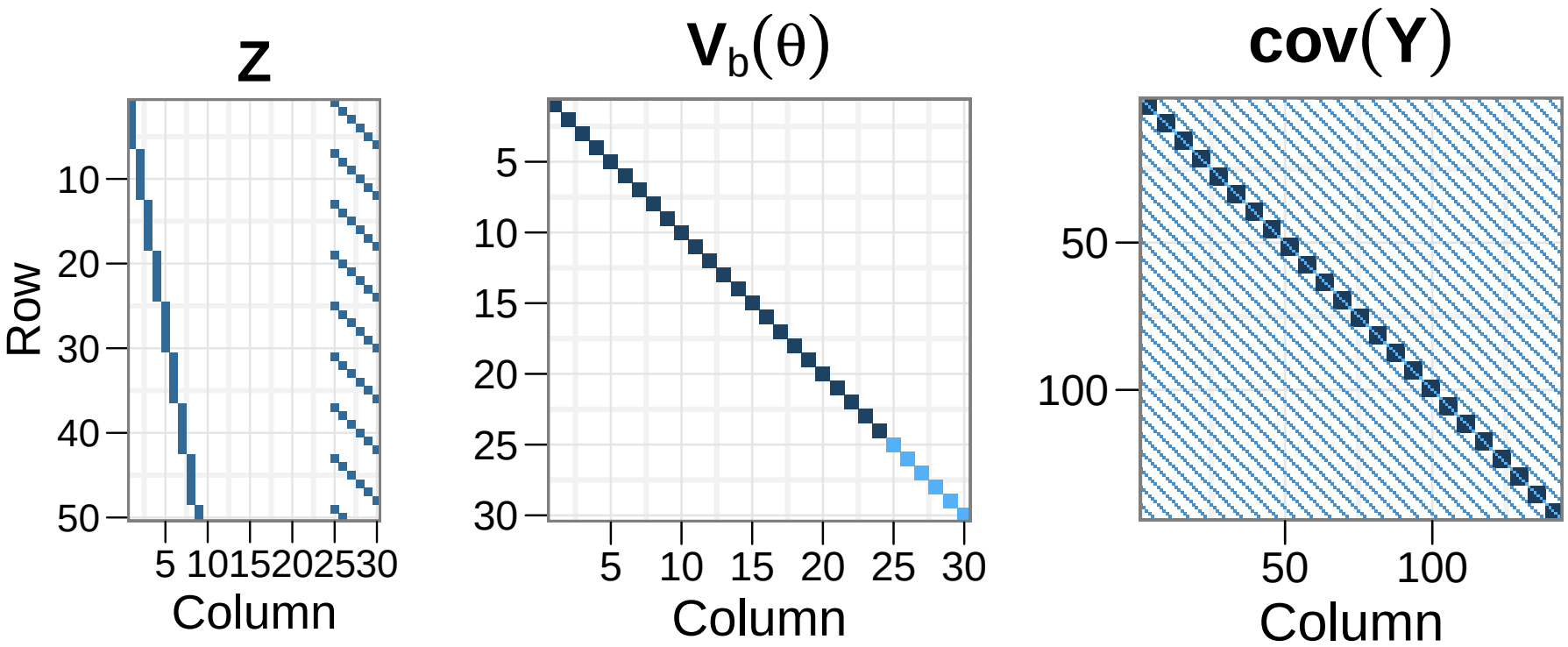
$$\boldsymbol{Y} = \boldsymbol{X}\beta + \boldsymbol{Z}\boldsymbol{B} + \varepsilon \ ,$$

- $\boldsymbol{Y}$ is the vector of the n observations,
- $\boldsymbol{X}$ is the design matrix of the p fixed effects β,
- $\boldsymbol{Z}$ is the design matrix of the q random effects $\boldsymbol{B}$, and
- ε is the vector of the observation level errors.

Assumptions

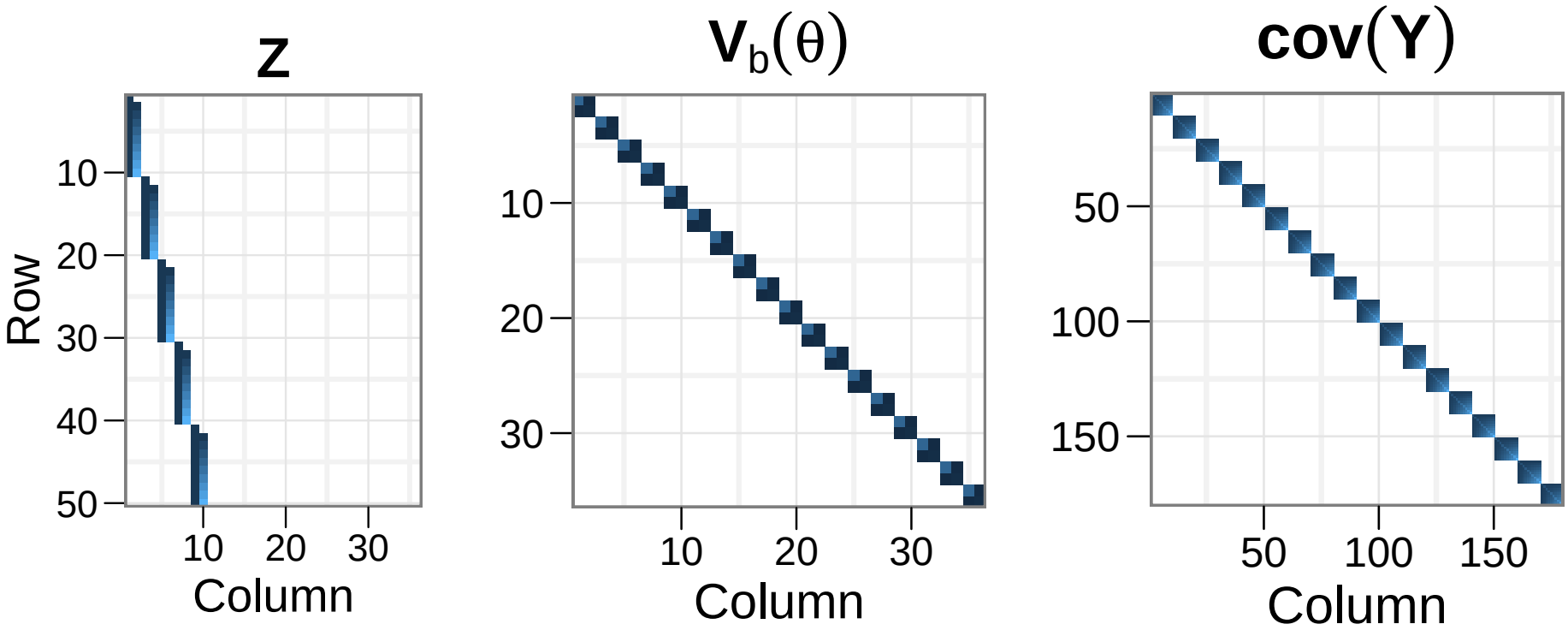
$$\varepsilon \sim \mathcal{N}_n\Big(\mathbf{0}, \sigma^2 \boldsymbol{V}_e\Big) \ , \quad \boldsymbol{B} \sim \mathcal{N}_q\Big(\mathbf{0}, \sigma^2 \boldsymbol{V}_b(\theta)\Big) \ , \quad \varepsilon \perp \boldsymbol{B} \ .$$

Matrices of the Penicillin example



Dimensions: $n = 144, \quad p = 1, \quad q = 30.$

Matrices of the Sleepstudy example



Dimensions: $n = 180$, $p = 2$, $q = 36$.

Log likelihood ℓ

To get the likelihood, one usually integrates out the random effects $\boldsymbol{b}$. But one can show that one can just minimize $\widetilde{d}$ for $\boldsymbol{b}$ instead:

$$-2\ell(\boldsymbol{\theta}, \beta, \sigma|\boldsymbol{y}) = \widetilde{d}\Big(\boldsymbol{\theta}, \beta, \widehat{\boldsymbol{b}}(\boldsymbol{\theta}, \beta, \sigma), \sigma|\boldsymbol{y}\Big)$$

$$\begin{aligned}\widetilde{d}(\boldsymbol{\theta}, \beta, \boldsymbol{b}, \sigma|\boldsymbol{y}) = n\log\Big(2\pi\sigma^2\Big) + \log|\boldsymbol{Z}\boldsymbol{V}_b(\boldsymbol{\theta})\boldsymbol{Z}^\top + \boldsymbol{V}_e| + \\ \frac{1}{\sigma^2}(\boldsymbol{y} - \boldsymbol{X}\beta - \boldsymbol{Z}\boldsymbol{b})^\top \boldsymbol{V}_e^{-1}(\boldsymbol{y} - \boldsymbol{X}\beta - \boldsymbol{Z}\boldsymbol{b}) + \\ \frac{1}{\sigma^2}\boldsymbol{b}^\top \boldsymbol{V}_b(\boldsymbol{\theta})^{-1}\boldsymbol{b}\end{aligned}$$

The form has separate terms for the residuals and the random effects, so we can robustify them separately.
The inverse can cause numerical problems ($\theta = 0$): reparametrize.

Reparametrize in terms of spherical random effects $\boldsymbol{b}^*$

$$\boldsymbol{b} = \boldsymbol{U}_b(\theta)\boldsymbol{b}^*, \qquad \boldsymbol{V}_b(\theta) = \boldsymbol{U}_b(\theta)\boldsymbol{U}_b(\theta)^\mathsf{T}$$
$$\varepsilon^*(\beta, \boldsymbol{b}^*) = \boldsymbol{U}_e^{-1}(\boldsymbol{y} - \boldsymbol{X}\beta - \boldsymbol{Z}\boldsymbol{U}_b(\theta)\boldsymbol{b}^*), \qquad \boldsymbol{V}_e = \boldsymbol{U}_e\boldsymbol{U}_e^\mathsf{T}$$

Note: By definition: $\boldsymbol{b}^* \sim \mathcal{N}\left(\boldsymbol{0}, \sigma^2 \boldsymbol{I}_q\right)$. Then

$$\widetilde{d}(\theta, \beta, \boldsymbol{b}^*, \sigma|\boldsymbol{y}) = n\log\left(2\pi\sigma^2\right) + \log|\boldsymbol{Z}\boldsymbol{V}_b(\theta)\boldsymbol{Z}^\mathsf{T} + \boldsymbol{V}_e| + \frac{1}{\sigma^2}\varepsilon^*(\beta, \boldsymbol{b}^*)^\mathsf{T}\varepsilon^*(\beta, \boldsymbol{b}^*) + \frac{1}{\sigma^2}\boldsymbol{b}^{*\mathsf{T}}\boldsymbol{b}^*$$

Simplest approach Replace blue terms by bounded versions. Does not work: $\sigma = 0$ would always be the global minimum (with value $-\infty$).
Robustify estimating equations instead.

ML (Maximum likelihood) estimating equations

$$\boldsymbol{X}^\top \boldsymbol{U}_e^{-\top} \widehat{\varepsilon}^* / \widehat{\sigma} = 0 \,,$$

$$\boldsymbol{U}_b^\top \boldsymbol{Z}^\top \boldsymbol{U}_e^{-\top} \widehat{\varepsilon}^* / \widehat{\sigma} - \widehat{\boldsymbol{b}}^* / \widehat{\sigma} = 0 \,,$$

$$\widehat{\varepsilon}^{*\top} \widehat{\varepsilon}^* / \widehat{\sigma}^2 = \operatorname{tr}\left(\boldsymbol{V}_y(\widehat{\theta})^{-1} \boldsymbol{V}_e \right) \,,$$

$$\widehat{\boldsymbol{b}}^{*\top} \boldsymbol{Q}_l(\widehat{\theta}) \widehat{\boldsymbol{b}}^* / \widehat{\sigma}^2 = \frac{1}{2} \operatorname{tr}\left(\boldsymbol{V}_y(\widehat{\theta})^{-1} \boldsymbol{Z} \frac{\partial \boldsymbol{V}_b(\widehat{\theta})}{\partial \theta_l} \boldsymbol{Z}^\top \right) \quad (l = 1, \ldots, r),$$

where

$$\widehat{\varepsilon}^* = \varepsilon^*(\widehat{\beta}, \widehat{\boldsymbol{b}}^*) = \boldsymbol{U}_e^{-1}(\boldsymbol{y} - \boldsymbol{X}\beta - \boldsymbol{Z}\boldsymbol{U}_b(\theta)\boldsymbol{b}^*) \,,$$

$$\boldsymbol{Q}_l(\theta) = \boldsymbol{U}_b(\theta)^{-1} \frac{\partial \boldsymbol{U}_b(\theta)}{\partial \theta_l} \,.$$

Last part missing: to get restriced maximum likelihood (REML) estimating equations, replace the red terms by the expectations of the left hand sides.

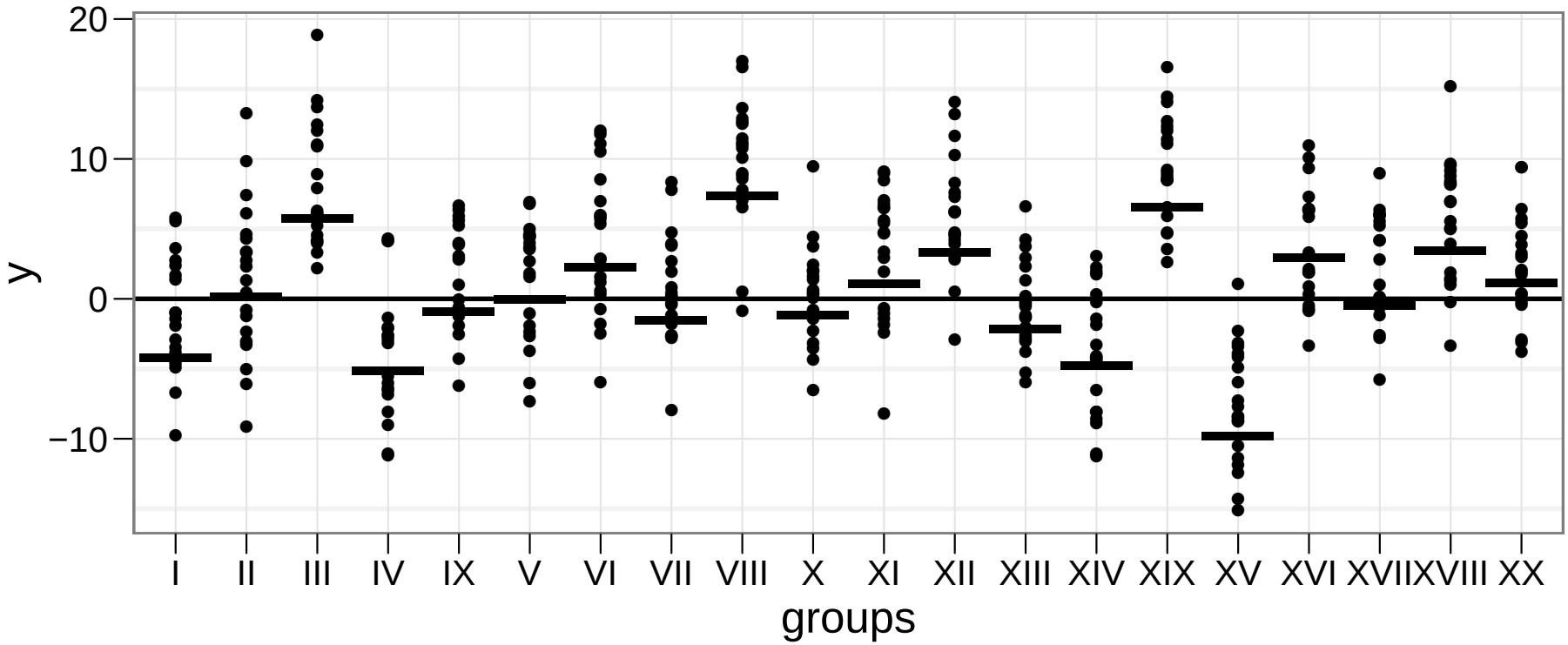
REML (Restricted maximum likelihood) Estimating Equations

$$\begin{aligned}
\boldsymbol{X}^\top \boldsymbol{U}_e^{-\top} \widehat{\varepsilon}^* / \sigma &= 0 \,, \\
\boldsymbol{U}_b^\top \boldsymbol{Z}^\top \boldsymbol{U}_e^{-\top} \widehat{\varepsilon}^* / \sigma - \widehat{\boldsymbol{b}}^* / \sigma &= 0 \,, \\
\widehat{\varepsilon}^{*\top} \widehat{\varepsilon}^* / \widehat{\sigma}^2 &= \mathbb{E}\left[\widehat{\varepsilon}^{*\top} \widehat{\varepsilon}^* / \widehat{\sigma}^2 \right] \,, \\
\widehat{\boldsymbol{b}}^{*\top} \boldsymbol{Q}_l(\widehat{\theta}) \widehat{\boldsymbol{b}}^* / \widehat{\sigma}^2 &= \mathbb{E}\left[\widehat{\boldsymbol{b}}^{*\top} \boldsymbol{Q}_l(\widehat{\theta}) \widehat{\boldsymbol{b}}^* / \widehat{\sigma}^2 \right] \quad (l = 1, \ldots, r),
\end{aligned}$$

where the expectations are computed using the (implied) distribution of the residuals and predicted random effects.

Check: are the estimates robust? Draw sensitivity curves!

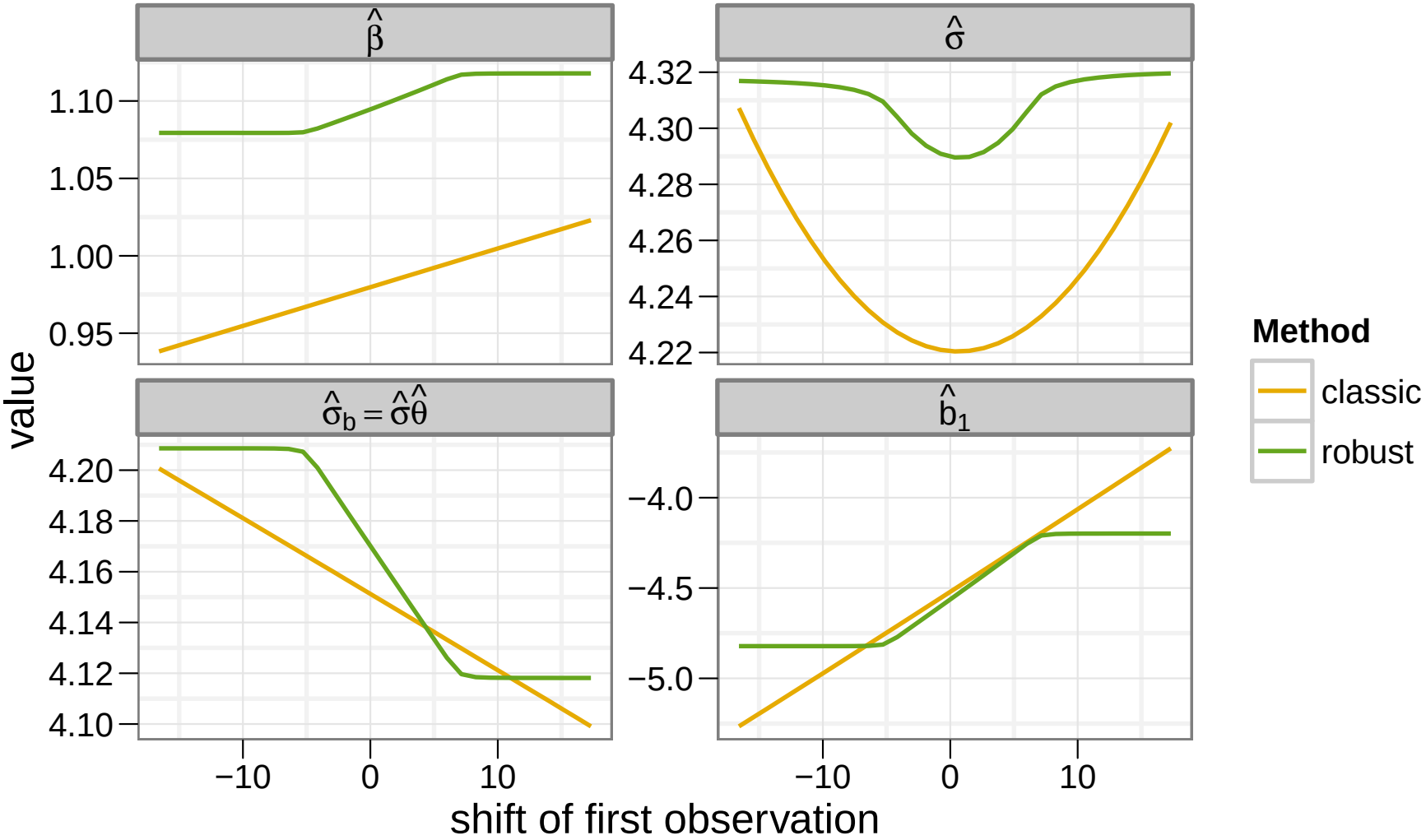
A simple one-way ANOVA with 20 groups, 20 observations per group:



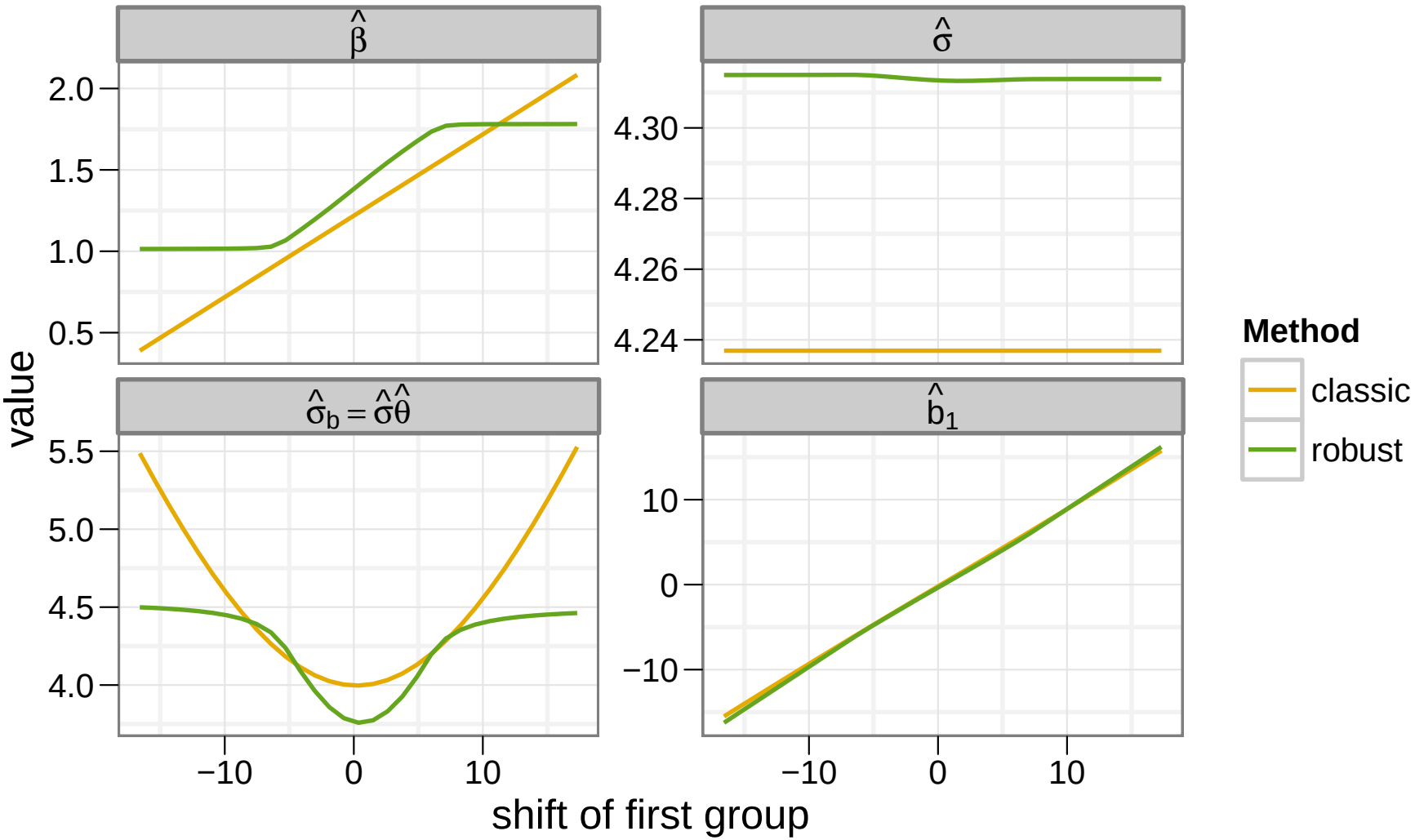
Modify dataset and plot how the estimates change:

Shift an observation, shift / collapse / stretch a group.

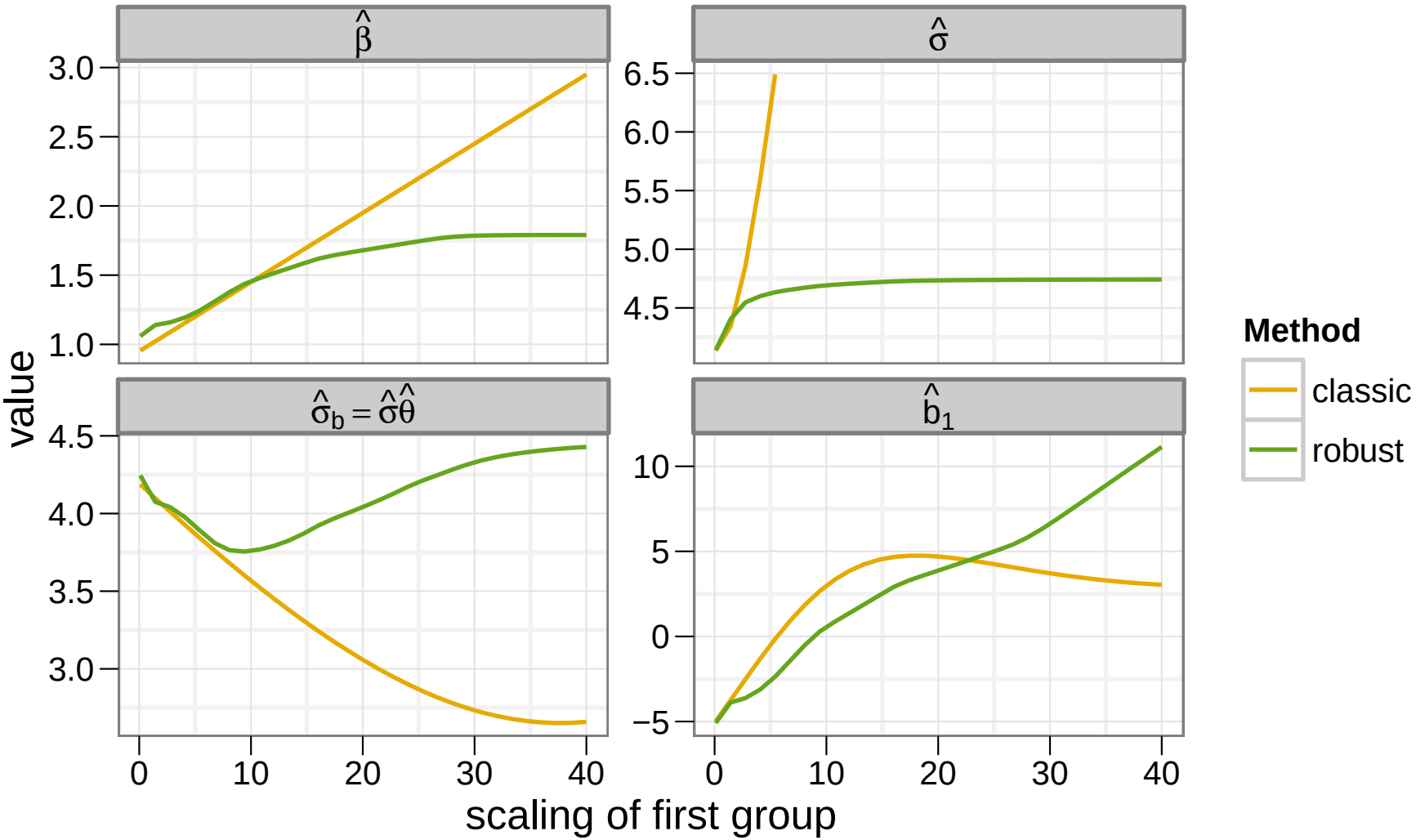
Shift an observation



Shift a group

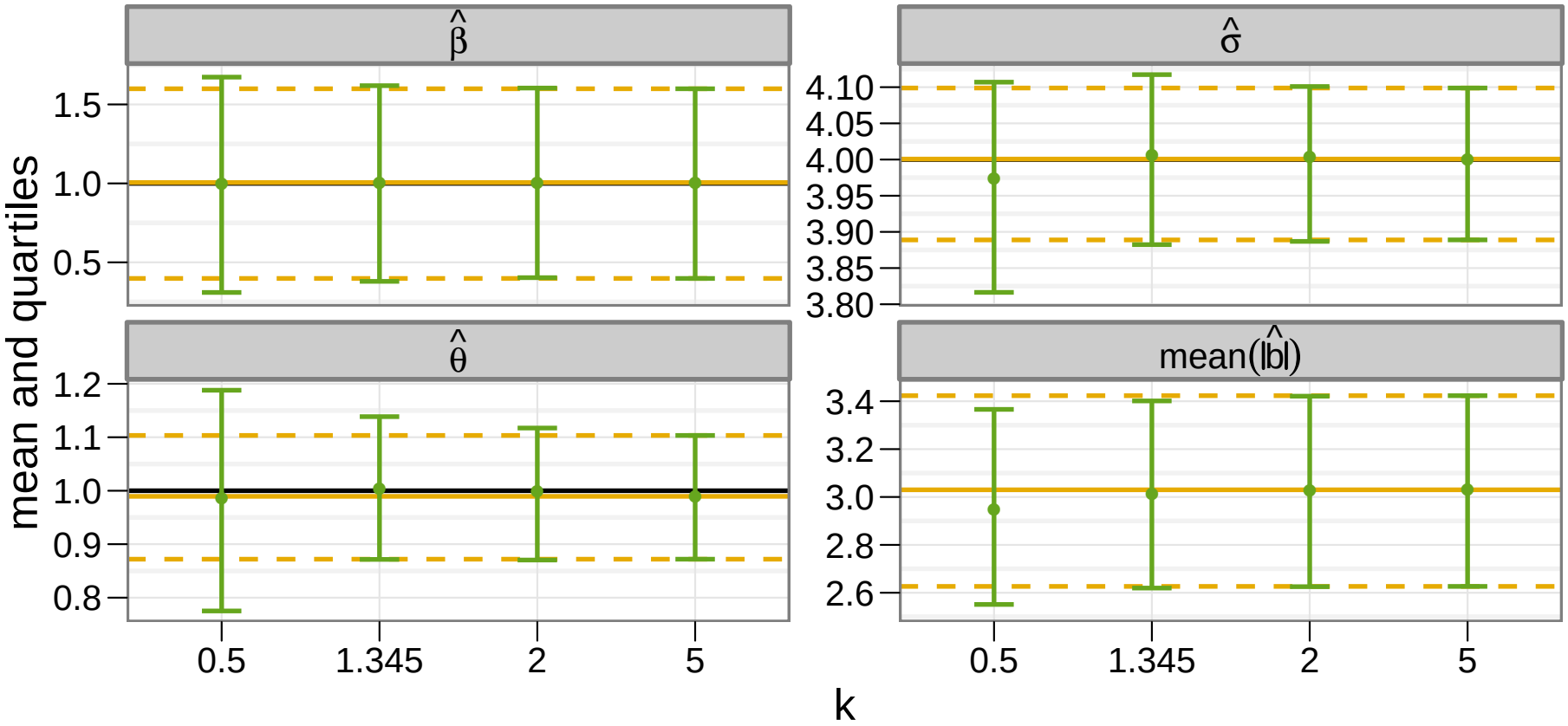


Collapse / stretch a group



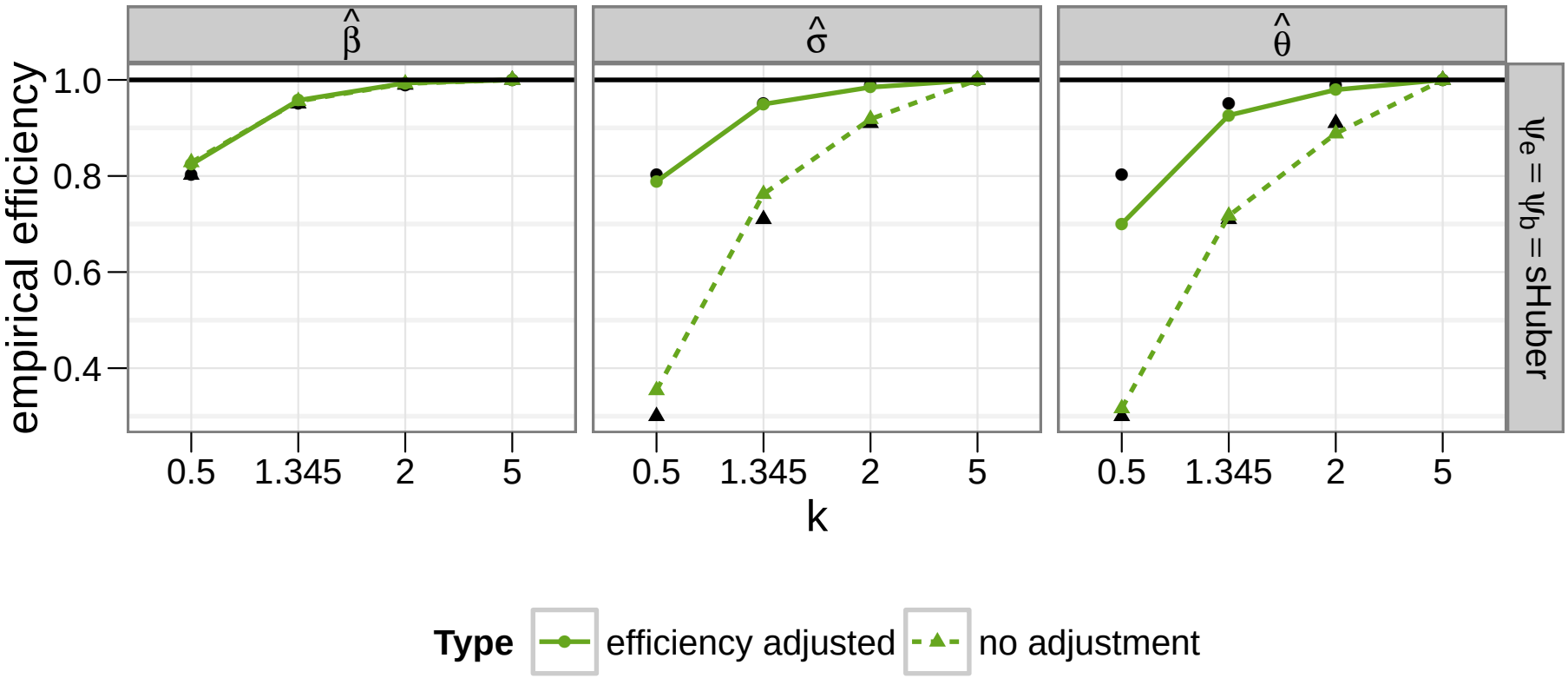
Check for bias

Generate the data a 1000 times and compute the robust fits for various tuning parameters k, plot mean and quartiles of estimates.



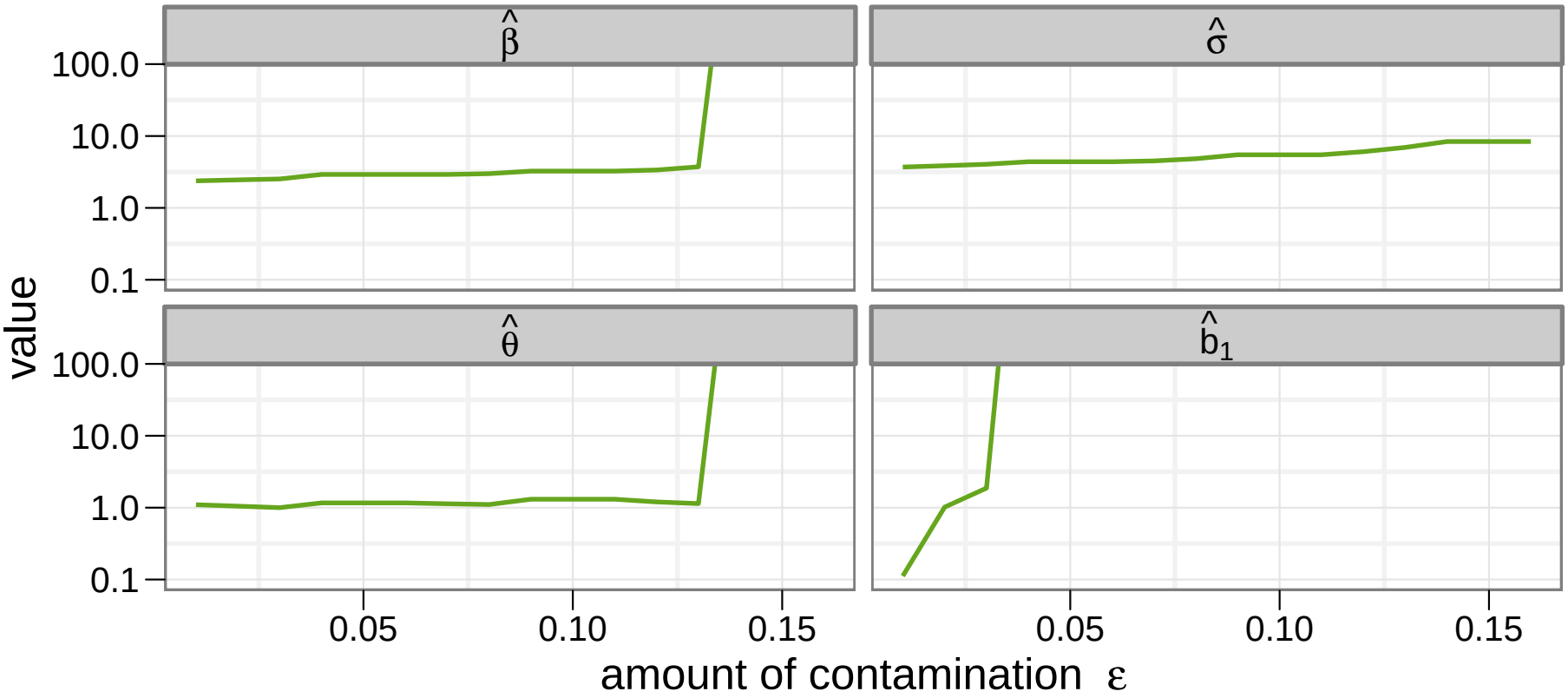
Efficiency (empirical)

Comparing robust and classical estimates for the same replicates used in the bias simulation. Black: (simplified) asymptotic efficiency.



Breakdown

Take a balanced one-way ANOVA dataset (20×5),
contaminate observation after observation, group after group.



```
> require(robustlmm)
```

R implementation demo: Penicillin example

```
> require(robustlmm)
> ## load Penicillin data and create contaminated data
> data(Penicillin, package="lme4")
> Penicillin <- within(Penicillin, plate <- reorder(plate, diameter))
> PenicillinC <- within(Penicillin, {
+   diameter[plate == "m"] <- diameter[plate == "m"] / 10
+   diameter[plate == "k" & sample == "F"] <- 16
+ })
> attr(PenicillinC$plate, "scores") <- NULL
> str(PenicillinC)
'data.frame':        144 obs. of  3 variables:
$ diameter: num 27 23 26 23 23 ...
$ plate : Factor w/ 24 levels "g","s","x","u",..: 18 18 18 18 18 ...
$ sample : Factor w/ 6 levels "A","B","C","D",..: 1 2 3 4 5 ...
```

Fit classic linear mixed effects model:

```
> st(classicalC <- lmer(diameter ~ 1 + (1|plate) + (1|sample), PenicillinC))
   user  system elapsed
  0.088   0.004   0.095
```

Fit robust linear mixed effects model:

```
> st(robustC <- rlmer(diameter ~ 1 + (1|plate) + (1|sample), PenicillinC,
+                     rho.e = smoothPsi, rho.b = smoothPsi,
+                     rho.sigma.e = psi2propII(smoothPsi, k = 2.28),
+                     rho.sigma.b = psi2propII(smoothPsi, k = 2.28)))

   user  system elapsed
 16.116   0.012  16.259
```

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

Random effects:
 Groups   Name        Variance Std.Dev.
 plate    (Intercept) 0.8622   0.9286
 sample   (Intercept) 3.9187   1.9796
 Residual             0.3580   0.5984
Number of obs: 144, groups: plate, 24; sample, 6

Fixed effects:
            Estimate Std. Error t value
(Intercept)  22.9063     0.8526   26.87

Robustness weights for the residuals:
 126 weights are ~= 1. The remaining 18 ones are summarized as
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  0.170   0.488   0.732   0.691   0.895   0.960

Robustness weights for the random effects:
```

```
 26 weights are ~= 1. The remaining 4 ones are
    1     2    24    30
0.965 0.965 0.062 0.854

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)
```

It is also possible to tune the ψ-functions for the two variance components separately. Here: fit *plate* variance component robustly, but use most efficient (classical) method for *sample*.

```
> st(robustC2 <- rlmer(diameter ~ 1 + (1|plate) + (1|sample), PenicillinC,
+                      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
 14.836   0.000  14.895
```

Comparison of the classical and robust estimates

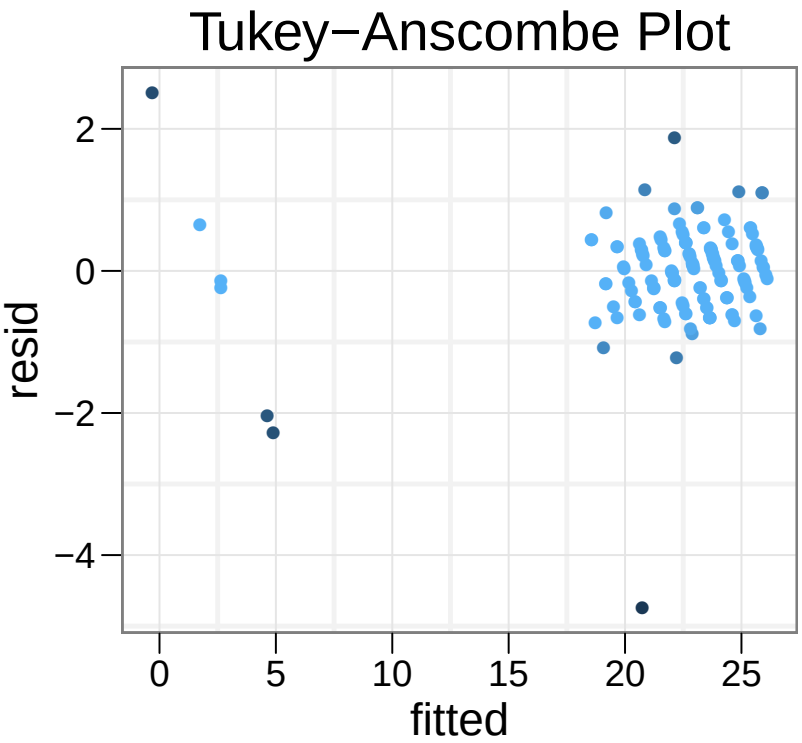
```
> print(xtable(compare(classicalC, robustC, classical, show.rho.functions=FALSE),
+              #caption="Comparison table of the fitted models for the Penicillin
+              label="tab:cmpPenicillin"), floating=TRUE, size="\\normalsize")
```

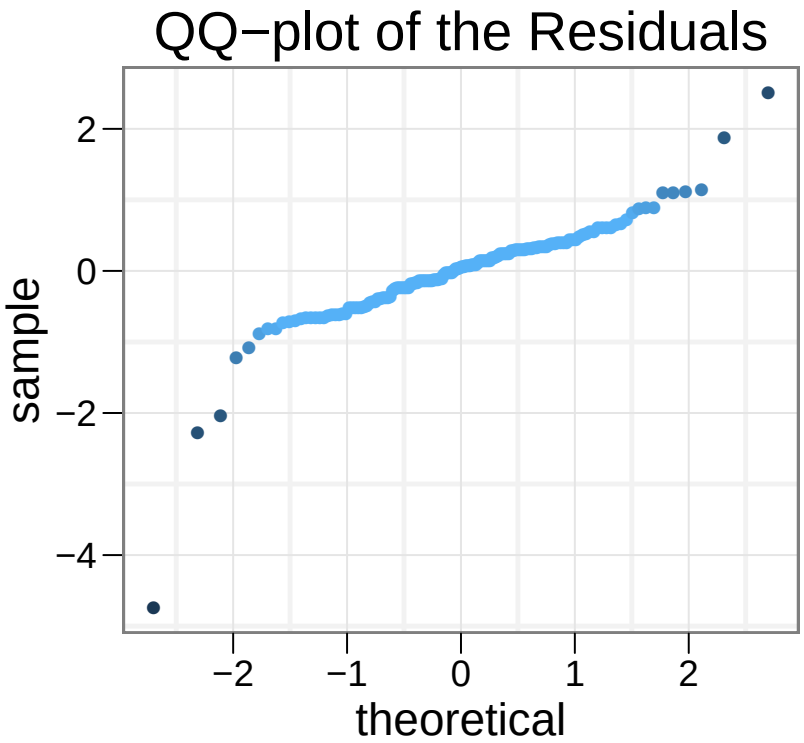
	classicalC	robustC	classical
Coefficients (Std. Error)			
(Intercept)	22 (1.17)	22.9 (0.853)	23 (0.809)
Variance components			
(Intercept) | plate	4.229	0.929	0.847
(Intercept) | sample	1.939	1.980	1.932
σ	0.777	0.598	0.55
REML	483		331

Classical fit on contaminated data is clearly off.

Only minor differences between robust and classical fit on clean data.

Residual analysis

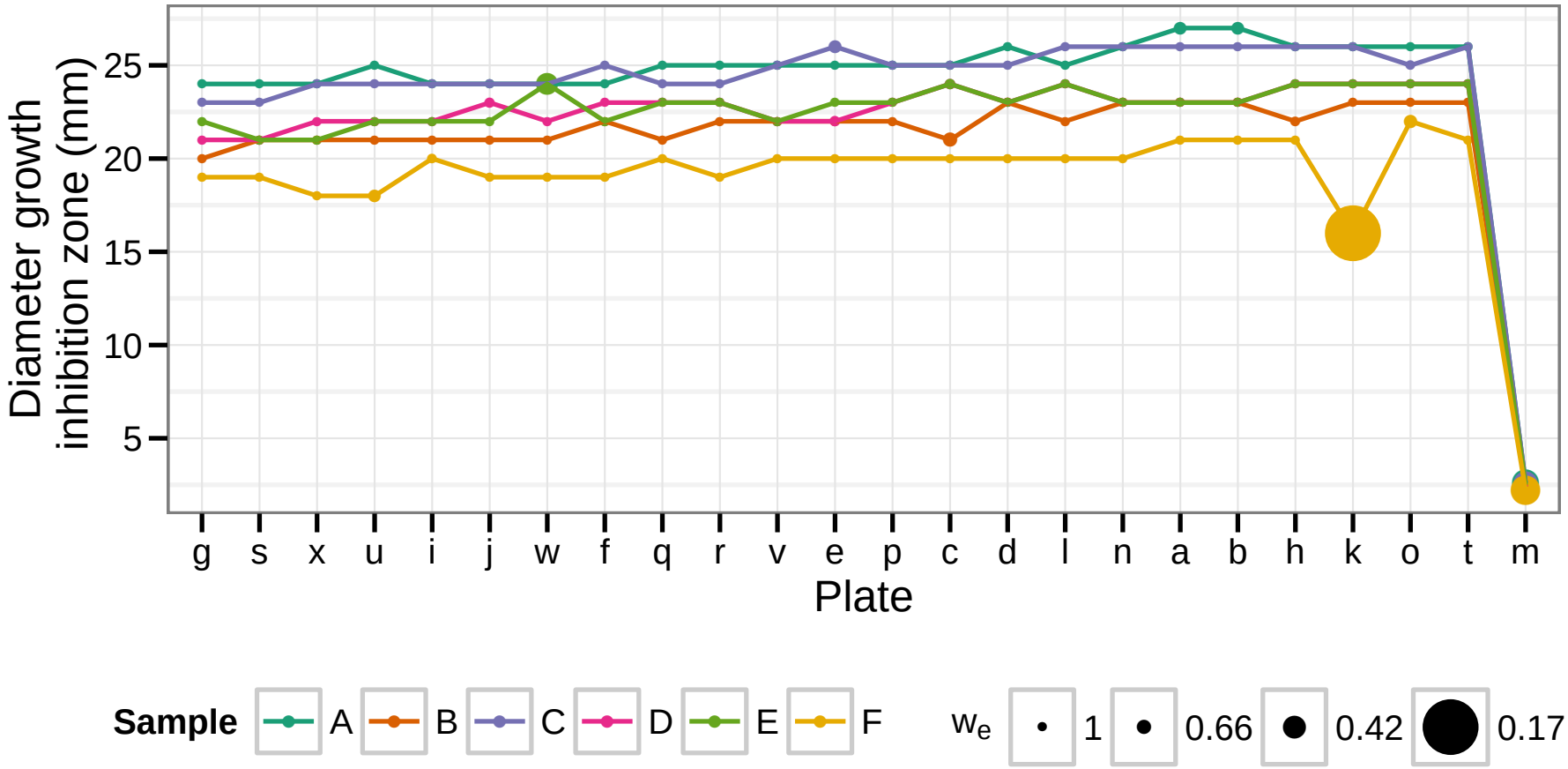





QQ−plot of the Random Effects
plate
sample
sample
theoretical
0
−5
−10
−15
−20
−2
−1
0
1
2
robustness weights
0.2
0.4
0.6
0.8
1.0

Which observations were downweighted?

```
> tmp <- cbind(PenicillinC, wgt.e = wgt.e(robustC))
> print(ggplot(tmp, aes(plate, diameter, color = sample)) +
+       geom_point(aes(size=1/wgt.e)) + geom_line(aes(as.numeric(plate))) +
+       scale_colour_brewer("Sample", palette="Dark2") +
+       scale_y_continuous(breaks=c(0,5,10,15,20,25)) +
+       scale_size_continuous(expression(w[e]),breaks=c(1,1/0.66,1/0.412,1/0.17),
+                             labels=c(1,0.66,0.42,"0.17     "), range=c(1,6)) +
+       xlab("Plate") + ylab("Diameter growth\n inhibition zone (mm)") +
+       opts(legend.position = "bottom", legend.box = "horizontal"))
```


Diameter growth inhibition zone (mm)
25
20
15
10
5
g s x u i j w f q r v e p c d l n a b h k o t m
Plate
Sample
A
B
C
D
E
F
w_e
1
0.66
0.42
0.17

Conclusions

- Developed a new robust method for estimating mixed effects models.
- The method supports crossed data structures and non-diagonal covariance matrices of the random effects.
- It can take care of contamination of different levels individually.

References

M. Koller (2013). *Robust estimation of linear mixed effects models*, Dissertation, ETH Zürich.

M. Koller (2013). *robustlmm: Robust estimation of linear mixed effects models*, R package version 0.7. *(https://github.com/kollerma/robustlmm)*.

D. M. Bates (2012). *lme4: Mixed-effects modeling with R*, *(http://lme4.r-forge.r-project.org/lMMwR/)*.

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

J. C. Pinheiro and D. M. Bates (2000). *Mixed-Effects Models in S and S-PLUS*, Springer.