Introduction to Robust Statistics

Klaus Nordhausen

Department of Mathematics and Statistics
University of Turku

Autumn 2015

Local vs global minimum

Let ρ be the ρ function for $\boldsymbol{\beta}$ and ρ_0 the ρ function for scale with the properties mentioned previously.

Denote

$$L(\boldsymbol{\beta}) = \sum_{i=1}^{n} \rho\left(\frac{r_i(\boldsymbol{\beta})}{\hat{\sigma}}\right)$$

and any local minima of $L(\boldsymbol{\beta})$ must satisfy $\sum_{i=1}^{n} \psi\left(\frac{r_i}{\hat{\sigma}}\right)\mathbf{x}_i = 0$.

Assume furthermore that

$$\rho_0(u) \geq \rho(u) \quad \forall u \in \mathbb{R}.$$

Local vs global minimum II

Then, if $\hat{\boldsymbol{\beta}}$ is such that

$$L(\hat{\boldsymbol{\beta}}) \leq L(\hat{\boldsymbol{\beta}}_0),$$

then $\hat{\boldsymbol{\beta}}$ is consistent and the BP of $\hat{\boldsymbol{\beta}}$ is not less than that of $\hat{\boldsymbol{\beta}}_0$.

Furthermore for the efficiency of $\hat{\boldsymbol{\beta}}$ it does not matter if it is a local or global minimum!

An estimate fulfilling this is a called an MM-estimate.

MM-estimate

The steps to obtain an MM estimate are therefore.

1. Choose the two ρ functions fulfilling the conditions above.
2. Compute an initial regression estimate $\hat{\boldsymbol{\beta}}_0$
3. Compute an M-estimate of scale $\hat{\sigma}$ based on ρ_0 using the residuals coming from $\hat{\boldsymbol{\beta}}_0$
4. $\hat{\boldsymbol{\beta}}_1$ is any solution of $\sum_{i=1}^n \psi\left(\frac{r_i}{\hat{\sigma}}\right) \mathbf{x}_i = 0$

which therefore must satisfy

$$\sum_{i=1}^n \rho\left(\frac{r_i(\hat{\boldsymbol{\beta}}_1)}{\hat{\sigma}}\right) \leq \sum_{i=1}^n \rho\left(\frac{\hat{\boldsymbol{\beta}}_0)}{\hat{\sigma}}\right)$$

A specific MM-estimate

A popular choice for an MM-estimate is

- Let ρ_1 be the bisquare ρ function with $c = 1$, i.e. $\rho_1 = \min\left\{1, 1 - (1 - t^2)^3\right\}$.
- Denote $\rho_0(r) = \rho_1\left(\frac{r}{c_0}\right)$ and $\rho(r) = \rho_1\left(\frac{r}{c}\right)$
- For consistency at the normal model, c_0 must be 1.56.
- To fulfill the condition $\rho \leq \rho_0$ we need then $c \geq c_1$.
- Choose the value of c then under this constraint and to obtain the desired efficiency.
- For example

Efficiency	0.80	0.85	0.90	0.95
c	3.14	3.44	3.88	4.68

Comments about MM-estimates

- MM-estimates have an unbounded IF.
- MM-estimates have large BP.
- The higher the efficiency is chosen the more suffers the robustness and bias might be introduced. For example in the previous suggestion, 85% efficiency are usually considered reasonable.

Initial value for MM-estimates

The main problem however still remains - we still need a high BP initial regression estimate which does not require a scale estimate.

As candidates we will consider here:

- Least median of squares (LMS)
- S-estimates
- Least trimmed squares LTS

LMS

The problem of LS and LAD is that they average either the squares or absolute value of the residuals and hence each residual has influence.

The idea is then to replace the "averaging" with a more robust alternative, like for example the median.

The Least median of squares (LMS) estimate intuitively finds the "strip" of residuals with minimal width which contain 50% of the observations.

There are algorithms to compute the LMS.

LMS is not very efficient at the normal model, especially not for large sample sizes.

Regression via scale estimation

The above idea can be generalized in the following way.

Denote $\mathbf{r}(\boldsymbol{\beta}) = (r_1(\boldsymbol{\beta}), \ldots, r_n(\boldsymbol{\beta}))$, then a regression estimate can be defined as

$$\hat{\boldsymbol{\beta}} = \operatorname*{argmin}_{\boldsymbol{\beta}} \sigma(\mathbf{r}(\boldsymbol{\beta})),$$

where σ is any scale equivariant scale estimate.

The resulting regression estimate is then regression, scale and affine equivariant.

S-estimate

A special case of the regression via scale approach is when the scale estimate is a M-estimate of scale

$$\frac{1}{n}\sum_{i=1}^{n}\rho_s\left(\frac{r_i}{\hat{\sigma}}\right)=\delta$$

has a bounded ρ function. Such regression estimates are called S estimates.

The asymptotic BP is then $\min(\delta, 1-\delta)$.

Usually the bisquare function is used for ρ.

If $\mathbf{x}$ has a density and $\delta = 0.5$ then an estimate has the maximum BP. However S-estimate cannot have a high BP and high efficiency. If the BP is 0.5, then the efficiency can be at most 33%. (The bisquare has efficiency 0.29.)

L-estimates of scale

Let us return again to L-statistics.
Consider for a centered sample $x_1 \dots, x_n$ the order statistics of the absolute values

$$|x|_{(1)}, \dots, |x|_{(n)}$$

then scale estimates can be linear combinations of the form

$$\hat{\sigma} = \sum_{i=1}^{n} a_i |x|_{(i)} \quad \text{or} \quad \hat{\sigma} = \left(\sum_{i=1}^{n} a_i |x|_{(i)}^2 \right)^{1/2},$$

where the $a_i \geq 0$ are constants.

LTS

In the regression case the least trimmed squares estimate (LTS) uses for this purpose

$$\hat{\sigma} = \left(\sum_{i=1}^{h} a_i |r|^2_{(i)} \right)^{1/2}$$

and hence trims the $n - h$ largest residuals.

If $\mathbf{X}$ is in general position and $h = 0.5(n + p + 1)$ the maximal BP is obtained. Then LTS has however the unbelievable low efficiency of 7% at the normal model.

Efficiency and high BP

So LMS, S-estimates and LTS have a high BP - but all have disappointing low efficiency at the normal model.

Hence in practise they cannot be used as such - but as initial estimates they are very valuable. Just that all of them are computational expensive to compute.

There exist one-step reweighting strategies to improve efficiency of these estimates - we do however not go into detail here as the state of art seems to be to use an S-estimate as initial value for MM-regression.

Exact fit property

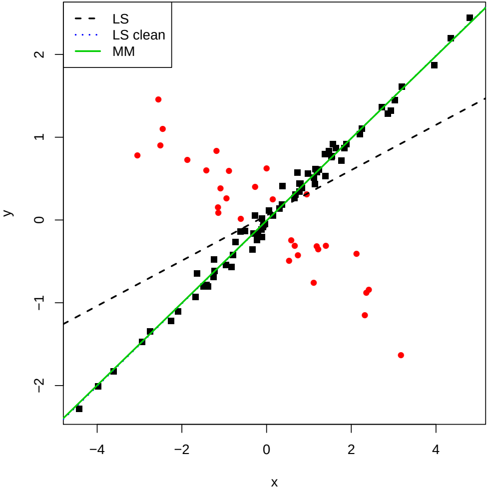
Assume that a proportion α of your data lies exactly in a subspace and that $1 - \alpha$ is lower than the BP of an regression and scale equivariant estimate. The exact fit property states then that the estimate will correspond to this subspace.

Hence

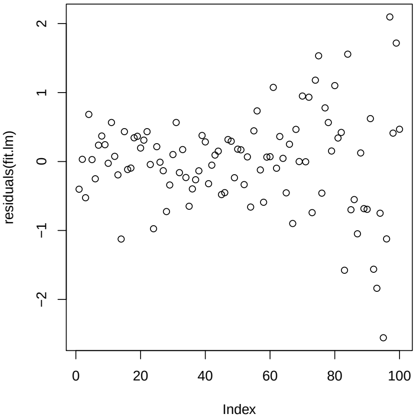
- If a large number of observations are well approximated by a linear fit $y_i \approx \mathbf{x}_i' \gamma$, the regression estimate will be $\hat{\boldsymbol{\beta}} \approx \gamma$.
- If a data set will consist of two linear subspaces the robust estimate will choose to fit one of them. The other can be discovered when looking at the residuals. (LS will try to "average" them to get a compromise fit.)

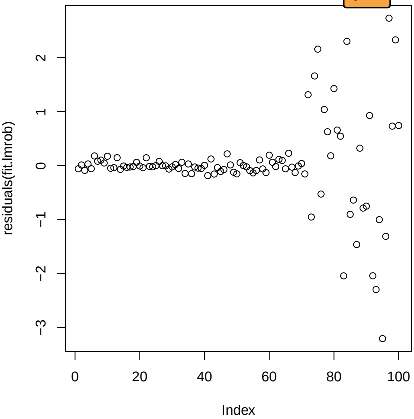
Exact fit property example

Consider the following example with n=100. Observations 1:70 follow one regression model and observations 71:100 another model.



Exact fit property example II





Modifications of MM-estimation

We introduced now the general theory behind MM-estimation - which is considered the state of the art robust regression approach.
However when using software many tiny complicated improvements have been added which will not be discussed here. Examples are:

- For confidence intervals usually not the asymptotic covariances as introduced here are used as these are not robust - but more sophisticated versions.
- To accommodate inference in small sample sizes also improvements in these directions are available.
- The theory outlined here was for iid data - but CI's and tests are also available for heteroscedastic data.

Note that for example also bootstrap methods might be available (note that standard bootstrap is not advisable in the context of robust regression because due to the standard resampling with replacement outliers might be upweighted and also it is computationally usually not feasible.)

Model selection in robust regression

While the literature on model selection for LS is huge, there seems to be not that much for the robust case.

Assume the goal is to compare different subsets of predictors $\mathbf{x}_{C_i}$ and of interest is the prediction error. In LS the idea would be to use the MSE. To evaluate this however robustly not only the fit must be robust but also the criterion. Otherwise it might be that the selected choice just fits some outlier best.

Crucial in the robust context is that for all subsets the same scale $\hat{\sigma}$ estimate is used. The robust final prediction error (RFPE) is then defined for a subset C_i as

$$RFPE(C_i) = E_\rho \left(\frac{y - \mathbf{x}'_{C_i} \hat{\boldsymbol{\beta}}_{C_i}}{\sigma} \right)$$

Model selection in robust regression II

Hence usually for the different subsets $\hat{\boldsymbol{\beta}}_{C_i}$ is computed separately but $\hat{\sigma}$ comes from the model containing all predictors from the subsets combined. To estimate RFPE(C_i) one uses

$$RFPE(C_i) = \frac{1}{n}\sum_{i=1}^{n} \rho\left(\frac{r_{i,C_i}}{\hat{\sigma}}\right) + \frac{q_{C_i}}{n}\frac{\hat{A}}{\hat{B}},$$

q_{C_i} is the number of predictors in C_i and

$$\hat{A} = \sum_{i=1}^{n} \psi\left(\frac{r_{i,C_i}}{\hat{\sigma}}\right)^2 \quad \text{and} \quad \hat{B} = \sum_{i=1}^{n} \psi'\left(\frac{r_{i,C_i}}{\hat{\sigma}}\right).$$

This is however a costly procedure if the number of subsets is large.

R for robust regression

Many `R` packages over robust regression methods. To name a few

- `quantreg` offers quantile regression including the LAD (function `rq`).
- `MASS` has the functions `rlm`, `lqs`.
- `robust` has the function `lmRob`.
- `robustbase` has the functions `lmrob` and `ltsReg` among others.

Robust regression using MASS package

The function `rlm` in the `MASS` package is one of the first implementations in R for robust regression.

It can do either M or MM regression, where later is usually based on an initial S-estimate. M estimate is the default with Huber's ρ function and constant c=1.345. The default scale is the iterated MAD.

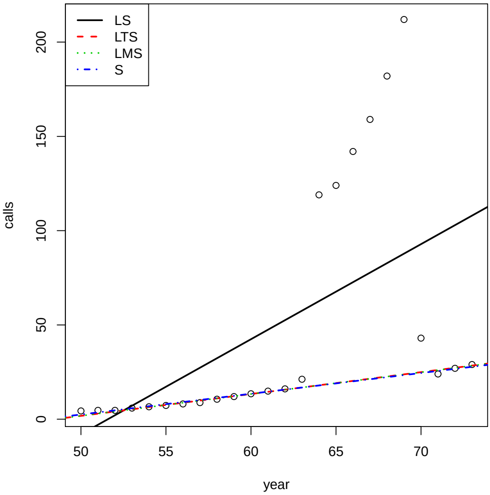
There are summary methods and so on available.

The function `lqs` offers lts, lms, and S-estimates. lts is the default. For LMS and LTS two scale estimates are returned. The first is based on the fit criterion and the second is the variance of the residuals of magnitude no more than 2.5 times the first scale estimate.

Demonstration of lqs using the Belgium phone call data

```
> library(MASS)
> data(phones)
> fit.lm <- lm(calls~year, data=phones)
> fit.lts <- lqs(calls~year, data=phones)
> fit.lms <- lqs(calls~year, data=phones, method="lms")
> fit.s <- lqs(calls~year, data=phones, method="S")
>
> with(phones, plot(year, calls))
> abline(fit.lm, lty=1, col=1, lwd=2)
> abline(fit.lts, lty=2, col=2, lwd=2)
> abline(fit.lms, lty=3, col=3, lwd=2)
> abline(fit.s, lty=4, col=4, lwd=2)
> legend("topleft", c("LS","LTS", "LMS", "S"),
+ lty=1:4, col=1:4, lwd=2)
```

Demonstration of lqs using the Belgium phone call data II



Demonstration of lqs using the Belgium phone call data III

```
> fit.lts
Call:
lqs.formula(formula = calls ~ year, data = phones)

Coefficients:
(Intercept)         year
    -56.162        1.159

Scale estimates 1.249 1.131
```

Demonstration of lqs using the Belgium phone call data IV

```
> fit.lms
Call:
lqs.formula(formula = calls ~ year, data = phones, method = "lms")

Coefficients:
(Intercept)         year
    -55.947        1.155

Scale estimates 0.9377 0.9095
```

Demonstration of lqs using the Belgium phone call data V

```
> fit.s
Call:
lqs.formula(formula = calls ~ year, data = phones, method = "S")

Coefficients:
(Intercept)         year
      -52.5          1.1

Scale estimates 2.129
```

Demonstration of rlm using the Belgium phone call data I

```
> fit.M <- rlm(calls~year, data=phones, init="lts", maxit=40)
> summary(fit.M)

Call: rlm(formula = calls ~ year, data = phones, init = "lts", maxi
Residuals:
    Min      1Q  Median      3Q     Max
-18.254  -5.932  -1.676  26.462 173.822

Coefficients:
            Value    Std. Error t value
(Intercept) -102.4491  26.4922   -3.8671
year           2.0381   0.4281    4.7611

Residual standard error: 8.989 on 22 degrees of freedom
```

Demonstration of rlm using the Belgium phone call data II

```
> fit.MM <- rlm(calls~year, data=phones, method="MM")
> summary(fit.MM)

Call: rlm(formula = calls ~ year, data = phones, method = "MM")
Residuals:
     Min       1Q   Median       3Q      Max
 -1.7442  -0.4543   0.2148  39.0082 188.4577

Coefficients:
            Value    Std. Error t value
(Intercept) -52.4230   2.9159   -17.9783
year          1.1009   0.0471    23.3669

Residual standard error: 2.129 on 22 degrees of freedom
```

Scottish hill data

The Scottish Hill Data gives the record times for 35 hill races and has the variables

- time: record time in minutes.
- dist: distance in miles on the map.
- climb: total height gained in feet during the route.

Note that because of inaccuracy of measurements, the observations are often weighted regarding the distance as shorter races might be more accurate then longer ones.

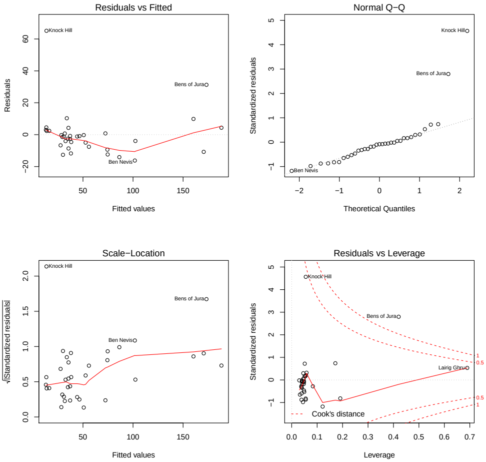
Scottish hill data II

```
> library(MASS)
> data(hills)
> fit.lm <- lm(time ~ dist + climb, data = hills)
```

Scottish hill data III

```
> summary(fit.lm)

Call:
lm(formula = time ~ dist + climb, data = hills)

Residuals:
    Min      1Q  Median      3Q     Max
-16.215  -7.129  -1.186   2.371  65.121

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -8.992039   4.302734  -2.090   0.0447
dist         6.217956   0.601148  10.343 9.86e-12
climb        0.011048   0.002051   5.387 6.45e-06
---

Residual standard error: 14.68 on 32 degrees of freedom
Multiple R-squared:  0.9191,    Adjusted R-squared:  0.914
F-statistic: 181.7 on 2 and 32 DF,  p-value: < 2.2e-16
```

Scottish hill data IV



Scottish hill data V

```
> # removing Knock Hill
> fit.lm.wKH <- update(fit.lm, subset = -18)
> coef(fit.lm.wKH)
(Intercept)        dist       climb
-13.5303520   6.3645623   0.0118549
>
> # Removing also Jura
> fit.lm.wKH.J <- update(fit.lm, subset = -c(7,18))
> coef(fit.lm.wKH.J)
  (Intercept)          dist          climb
-10.361645718   6.692113548   0.008046826
```

Scottish hill data VI

```
> # comment, still not realistic model
> # no intercept? or weights 1/dist^2
> fit.lm2 <- lm(time ~ dist + climb, weight=1/dist^2, data = hills)
> summary(fit.lm2)

Call:
lm(formula = time ~ dist + climb, data = hills, weights = 1/dist^2)

Weighted Residuals:
    Min      1Q  Median      3Q     Max
-3.7279 -1.5210 -0.5135  0.3242 18.6203

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 3.627150   6.267656   0.579  0.56684
dist        5.939599   1.714958   3.463  0.00154
climb       0.003837   0.004823   0.796  0.43214
---

Residual standard error: 3.698 on 32 degrees of freedom
Multiple R-squared:  0.4584,    Adjusted R-squared:  0.4245
F-statistic: 13.54 on 2 and 32 DF,  p-value: 5.483e-05
```

Scottish hill data VII

```
> fit.MM <- rlm(time ~ dist + climb, weight=1/dist^2, data = hills, method="MM")
> summary(fit.MM)

Call: rlm(formula = time ~ dist + climb, data = hills, weights = 1/dist^2,
    method = "MM")
Residuals:
    Min      1Q  Median      3Q     Max
-2.7246 -0.4181  0.0154  0.5629 20.8691

Coefficients:
            Value   Std. Error t value
(Intercept) -4.0621  1.6507    -2.4608
dist         5.8217  0.4517    12.8893
climb        0.0075  0.0013     5.9372

Residual standard error: 0.8193 on 32 degrees of freedom
```

Scottish hill data VIII

```
> fit.lm2 <- lm(time ~ dist + climb, weight=1/dist^2, data = hills, subset = - 18)
> summary(fit.lm2)
Call:
lm(formula = time ~ dist + climb, data = hills, subset = -18,
    weights = 1/dist^2)

Weighted Residuals:
     Min       1Q   Median       3Q      Max
-2.66603 -0.58817  0.00677  0.53522  3.09838

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -5.808538   2.034248  -2.855   0.0076
dist         5.820760   0.536084  10.858 4.33e-12
climb        0.009029   0.001537   5.873 1.76e-06
---

Residual standard error: 1.156 on 31 degrees of freedom
Multiple R-squared:  0.9249,    Adjusted R-squared:  0.9201
F-statistic: 190.9 on 2 and 31 DF,  p-value: < 2.2e-16
```

Robust regression using lmRob

The `lmRob` function is more sophisticated than `rlm`. It chooses different initial estimates depending on if factors are present or not.

It either gives an MM estimate or an reweighted estimate (not discussed in this course) to increase efficiency.

The function works basically the same as `lm` and has the methods:

`add1`, `anova`, `coef`, `deviance`, `drop1`, `fitted`, `formula`, `labels`, `plot`, `print`, `residuals`, `summary`, `update`.

Compare robust and LS in robust package

The package `robust` makes it easy to compare robust and nonrobust fits.

```
> library(robust)
> library(MASS)
> data(hills)
> fits <- fit.models(list(LS = "lm", MM = "lmRob"),
+                    time ~ dist + climb, weights=1/hills$dist^2,
>
> summary(fits)
```

Compare robust and LS in robust package II

```
Calls:
LS: lm(formula = time ~ dist + climb, data = hills, weights = 1/hills$dist^2)
MM: lmRob(formula = time ~ dist + climb, data = hills, weights = 1/hills$dist^2)

Residual Statistics:
       Min      1Q   Median     3Q   Max
LS: -3.728 -1.5210 -0.51350 0.3242 18.62
MM: -2.821 -0.4674 -0.07759 0.4432 20.81

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept): LS:  3.627150   6.267656   0.579 0.566837
             MM: -3.571786   2.932275  -1.218 0.232094

       dist: LS:  5.939599   1.714958   3.463 0.001537
             MM:  5.607241   0.748696   7.489 1.59e-08

      climb: LS:  0.003837   0.004823   0.796 0.432144
             MM:  0.008451   0.002278   3.709 0.000787
---

Residual Scale Estimates:
LS: 3.698 on 32 degrees of freedom
MM: 0.7921 on 32 degrees of freedom

Multiple R-squared:
LS: 0.4584
MM: 0.3763
```

Robust regression using lmrob

The `lmrob` function in `robustbase` is the state of the art. It is under constant development and has the most options.

The current recommendation is to use the default plus `setting = "KS2014")`, especially if factors are present.

The function has the methods:

`alias`, `anova`, `case.names`, `confint`, `dummy.coef`, `family`, `hatvalues`, `kappa`, `labels`, `model.matrix`, `nobs`, `plot`, `predict`, `print`, `qr`, `residuals`, `sigma`, `summary`, `variable.names`, `vcov`, `weights`

lmrob and Scottish Hills I

```
> library(MASS)
> library(robustbase)
> data(hills)
> fit.rb <- lmrob(time ~ dist + climb, weight=1/dist^2, data = hills, setting = "KS2014")
> summary(fit.rb)

Call:
lmrob(formula = time ~ dist + climb, data = hills, weights = 1/dist^2, setting = "KS2014")
 --> method = "SMDM"
Residuals:
      Min        1Q    Median        3Q       Max
-12.19616  -3.45206   0.07527   3.74303  63.01646

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -4.659897   1.822204  -2.557   0.0155
dist         5.823062   0.468810  12.421 8.79e-14
climb        0.008069   0.001392   5.796 1.96e-06
---
Robust residual standard error: 1.017
Multiple R-squared:  0.9571,    Adjusted R-squared:  0.9544
Convergence in 12 IRWLS iterations
```

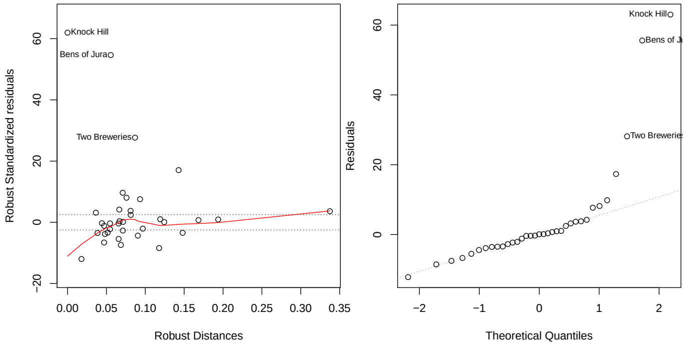
lmrob and Scottish Hills II

```
Robustness weights:
 observation 18 is an outlier with |weight| = 0 ( < 0.0029);
 26 weights are ~= 1. The remaining 8 ones are
 Bens of Jura   Cairn Table    Eildon Two    Black Hill      Cow Hill    Creag Dubh        Acmony
       0.2674        0.8757        0.9185        0.4684        0.8069        0.6962        0.9141
Two Breweries
       0.8982
Algorithmic parameters:
      tuning.chi1       tuning.chi2       tuning.chi3       tuning.chi4                bb
       -5.000e-01         1.500e+00                NA         5.000e-01         5.000e-01
      tuning.psi1       tuning.psi2       tuning.psi3       tuning.psi4        refine.tol
       -5.000e-01         1.500e+00         9.500e-01                NA         1.000e-07
          rel.tol         solve.tol       eps.outlier             eps.x warn.limit.reject
        1.000e-07         1.000e-07         2.857e-03         9.095e-10         5.000e-01
warn.limit.meanrw
        5.000e-01
     nResample         max.it       best.r.s       k.fast.s          k.max    maxit.scale      trace.lev
          1000            500             20              2           2000            200              0
           mts     compute.rd      numpoints fast.s.large.n
          1000              0             10           2000
              setting                   psi           subsampling                   cov
             "KS2014"                 "lqq"         "nonsingular"            ".vcov.w"
compute.outlier.stats
               "SMDM"
seed : int(0)
```

lmrob and Scottish Hills III

```
> round(weights(fit.rb, type="r"),3)
     Greenmantle           Carnethy      Craig Dunain           Ben Rha
           1.000              1.000             1.000             1.000
      Ben Lomond           Goatfell      Bens of Jura       Cairnpapple
           1.000              1.000             0.267             1.000
          Scolty           Traprain       Lairig Ghru            Dollar
           1.000              1.000             1.000             1.000
         Lomonds        Cairn Table        Eildon Two         Cairngorm
           1.000              0.876             0.918             1.000
     Seven Hills         Knock Hill        Black Hill        Creag Beag
           1.000              0.000             0.468             1.000
    Kildcon Hill Meall Ant-Suidhe     Half Ben Nevis          Cow Hill
           1.000              1.000             1.000             0.807
   N Berwick Law         Creag Dubh         Burnswark         Largo Law
           1.000              0.696             1.000             1.000
         Criffel             Acmony         Ben Nevis       Knockfarrel
           1.000              0.914             1.000             1.000
   Two Breweries          Cockleroi      Moffat Chase
           0.898              1.000             1.000
> which(weights(fit.rb, type="r") < 0.3) #(often compared to 0.2)
Bens of Jura   Knock Hill
           7           18
> plot(fit.rb, which=1:4)
# some warning appears...
```

lmrob and Scottish Hills IV



Key references

The course material is mainly based on the following books:

- Maronna, R.A., Martin, D.R. & Yohai, V.J. (2006): Robust Statistics: Theory and Methods, Wiley.
- Jureckova, J. & Picek, J. (2005): Robust Statistical Methods with R, Chapman and Hall/CRC
- Huber, P.J. (1981): Robust Statistics, Wiley.

Conclusion

It is perfectly proper to use both classical and robust/resistant methods routinely, and only worry when they differ enough to matter. But when they differ, you should think hard.

John Tukey