Robust Statistics Collaborative Package
Development: robustbase

Martin Maechler¹  Andreas Ruckstuhl²

¹ETH Zurich
²ZHW Winterthur
Switzerland
maechler@R-project.org, rks@zhwin.ch

useR! 2006, Wien
June 16, 2006

Outline

Robust Statistics with R : History
  Robust Statistics with S (R) — JWT, etc
  RsR — MASS
  RsR — past: “miscellaneous”

Robust Statistics with R — reloaded
  Workshop in Treviso
  new books on Robust Statistics

The package robustbase
  robustbase: current status overview
  robustbase: lmrob and glmrob examples
  robustbase: plans
  robustbase: relation to other R packages

Package writing collaboration
  The people
  The functions / classes
  The other free software

Conclusions

Robust Statistics with S (R) — JWT

➤ The father of EDA and early robustness: John W. Tukey
➤ @ Bell Labs: heavily influenced development of S. Hence basic robust tools have been part of S forever.
  ➤ median(), mad() (also: mean(*, trim = α))
  ➤ stem(), fivenum() → boxplot() etc
  ➤ medianpolish(), smooth(), line(x,y) (“Tukey line”!)
➤ Robust nonparametric regression: lowess() (but it has been known that lowess() is not really robust.... Because it starts from least-squares instead of robust smooth.
  loess() and locfit from package ‘locfit’ do about the same.
➤ For a better start, I had added runmed() to R 1.7.0, in early 2003 (package modreg, now part of stats).

Robust Statistics with R— the past II

➤ Venables and Ripley had added robust functionality to S and R with their “MASS” book and package
  ➤ huber() and hubers() M-estimator for location
  ➤ cov.rob() (with MVE and MCD) and cov.trob() for “Resistant Estimation of Multivariate Location and Scatter”
  ➤ lqs() incl. LQS, LTS, LMS, and S estimator for high-breakdown point (=: HBP)
  ➤ rlm() for more efficient HBP robust fitting of linear models (MM– or M-estimation).
Additionally, there have been miscellaneous R packages providing robust (or at least “resistant”) methods:

- **quantreg**: “Quantile regression and related methods” by Roger Koenker... of course $L_1$, but has unbounded influence of $x$.
- **sfsmisc** ($SFS =$ Seminar für Statistik, ETH Zurich):
  - `rnls()`: robust nonlinear regression (robust ‘nls’)
  - `f.robftest()`: “Robust F-test, i.e., Wald test for multiple coefficients of rlm() B”; further `rrange()` and `huberM()`.
- **forward**: “Forward search approach to robust analysis in LM and GLM” by Kjell Konis and Marco Riani (for S+)
- **wle**: “Robustness via Weighted Likelihood” by Claudio Agostinelli
- **rrcov**: “Functions for Robust Location and Scatter Estimation and Robust Regression with High Breakdown Point” by Valentin Todorov; originally: new fast MCD and LTS.
- **fields**: robust variograms etc by Doug Nychka
- **covRobust**: `cov.nnve()` by Naisyin Wang and Adrian Raftery
- **amap**: robust PCA `acprob()` and `varrob()`
- **multinomRob**: overdispersed multinominals

Reload of “R s R”:

“Organized” effort to provide more R functionality for robustness...

Several working groups, notably
  ▶ Regression (incl. GLM)
  ▶ “Multivariate”
with the goal to unite efforts in providing more modern, coherent
R functionality for robust statistics.

New books on robust statistics

Several classical books have had re-editions in 2005...  
Robust Statistics, Theory and Methods, Wiley.

Ricardo Maronna and Victor Yohai — very reknown in robust
statistics — took part in Treviso and agreed to support the idea of
taking their book as a target:
→ Collaborate to provide “basic robust statistics” functionality in
R, via one package:

The package robustbase

“The” new package for robustness ...  
How to chose the package name?

Had fun with a vote on choosing the package name. Every voter
was allowed to allocate 3 votes; 20 “contestants” casting votes
within a time limit... the final votes naming a new “basic robust
statistics" R package were

robustbase 45  
robustats 9  
robusta 5  
robustat 1  

where I had voted (0,1,2,0) ...

robustbase: current status

1. Many data sets, particularly from the book of Rousseeuw and
   Leroy, mostly thanks to Valentin Todorov; all with full help
   pages:
   24 datasets, to be used in other packages, by, e.g.,
   data(wood, package = "robustbase"). Data sets from
   Maronna, Martin and Yohai (2006) are also being added to
   the robustbase package.

2. covMcd() and ltsReg() by Valentin Todorov; originally in his
   rrcov package — now using shared code and notably using
   R’s random number generator (and seed).
   There have been cov.mcd() and ltsreg() in MASS. However,
   Valentin’s routines use the fast algorithms of Peter Rousseeuw
   (→ useR! talk by Valentin in Friday’s focus “robustness”)
3. New functionality that hasn’t been available in “public” R packages till now:
   - glmrob() by Andreas Ruckstuhl, based on Eva Cantoni’s work for S-PLUS (and MM’s for R) for robust Binomial GLMs, including model selection based on quasi deviance differences.
     *E. Cantoni and E. Ronchetti (2001)*
     *Robust Inference for Generalized Linear Models;* JASA 96, 1022 ff
   - anova() model selection for both 'lmrob' and 'glmrob'. anova.lmrob() with option to choose between *Wald* and *Deviance* tests.
   - Qn() and Sn() scale estimates by Rousseeuw and Croux [50% breakdown but considerably more efficient than MAD]; based on their S-plus + Fortran code; ported to R by M.

3. (..continued..)
   - covOGK(): The *orthogonalized Gnanadesikan-Kettenring* estimate for “fast” “high-dimensional” cov-estimation, by Maronna and Zamar (2002); based on code from Kjell Konis. This includes their univariate tau-estimate; I’ve called *scaleTau2()* (since there’s a different scaleTau() in other places), however amended with a consistency correction factor.
   - nlrob() for robust non-linear regression; this a slightly enhanced version of what has been available as *rnls()* from package `sfsmisc`. Also based mainly on Andreas Ruckstuhl’s work.
   - huberM() — “a robust” version of MASS::huber()

4. Somewhat experimental code for an S4 class of "psi-function" ($\psi$, $\rho$, $\psi'$, etc) objects.

---

An example of using lmrob():
> data(table.b13, package = "MPV")
> Jet <- table.b13
> Jet.r1 <- lmrob(y ~ ., data = Jet)
> summary(Jet.r1)

Coefficients:

|            | Estimate | Std. Error | t value | Pr(>|t|) |
|------------|----------|------------|---------|----------|
| (Intercept)| -4.023e+03 | 2.820e+03 | -1.426  | 0.163150 |
| x1         | 1.209e+00  | 3.060e-01  | 3.952   | 0.000385 ***|
| x2         | -3.325e-02 | 6.895e-02  | -0.482  | 0.632875 |
| x3         | 2.022e-01  | 1.279e-01  | 1.581   | 0.123449 |
| x4         | 3.525e+00  | 3.748e+00  | 0.941   | 0.353771 |
| x5         | 8.291e-01  | 3.111e-01  | 2.665   | 0.011812 * |
| x6         | -1.629e+01 | 3.461e+00  | -4.706  | 4.38e-05 ***|

---

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Robust residual standard error: 23.77

Convergence in 33 IRWLS iterations
Robustness weights:

<table>
<thead>
<tr>
<th></th>
<th>Min.</th>
<th>1st Qu.</th>
<th>Median</th>
<th>Mean</th>
<th>3rd Qu.</th>
<th>Max.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.5846</td>
<td>0.8970</td>
<td>0.9413</td>
<td>0.9116</td>
<td>0.9932</td>
<td>0.9999</td>
</tr>
</tbody>
</table>

Algorithmic parameters:

<table>
<thead>
<tr>
<th></th>
<th>tuning.chi</th>
<th>bb</th>
<th>tuning.psi</th>
<th>refine.tol</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1.5476400</td>
<td>0.5000000</td>
<td>4.6850610</td>
<td>0.0000001</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>nResample</th>
<th>max.it</th>
<th>groups</th>
<th>n.group</th>
<th>best.r.s</th>
<th>k.fast.s</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>500</td>
<td>50</td>
<td>5</td>
<td>400</td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>k.max</th>
<th>compute.rd</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>200</td>
<td>0</td>
</tr>
</tbody>
</table>

seed : int(0)

Robust model comparison for robustly fit models:

```r
> anova(Jet.r1, y ~ x1 + x5 + x6, test = "Wald")
```

```
Robust Wald Test Table

Model 1: y ~ x1 + x2 + x3 + x4 + x5 + x6
Model 2: y ~ x1 + x5 + x6
Largest model fitted by lmrob(), i.e. MM
```

<table>
<thead>
<tr>
<th></th>
<th>pseudoDf</th>
<th>Test.Stat</th>
<th>Df</th>
<th>Pr(&gt;chisq)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>33</td>
<td></td>
<td>0.2187</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>36</td>
<td>3</td>
<td>0.1360</td>
</tr>
</tbody>
</table>

GLM - "binomial" – robust: glmrob

An example of using `glmrob()` for robust GLM estimation:

```r
> data(carrots)
> Cfit1 <- glm(cbind(success, total - success) ~ logdose + block, data = carrots, family = binomial)
> summary(Cfit1)
```

```
Call: glm(formula = cbind(success, total - success) ~ logdose + block, data = carrots, family = binomial)

Deviance Residuals:

<table>
<thead>
<tr>
<th>Min</th>
<th>1Q</th>
<th>Median</th>
<th>3Q</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1.9200</td>
<td>-1.0215</td>
<td>-0.3239</td>
<td>1.0602</td>
<td>3.4324</td>
</tr>
</tbody>
</table>

AIC: 128.61
Number of Fisher Scoring iterations: 4

Coefficients:

|            | Estimate | Std. Error | z value | Pr(>|z|) |
|------------|----------|------------|---------|---------|
| (Intercept)| 2.0226   | 0.6501     | 3.111   | 0.00186 ** |
| logdose    | -1.8174  | 0.3439     | -5.285  | 1.26e-07 *** |
| blockB2    | 0.3009   | 0.1991     | 1.511   | 0.13073 |
| blockB3    | -0.5424  | 0.2318     | -2.340  | 0.01929 *  |
```
GLM - "binomial" – robust: glmrob – 2 –

```r
> Cfit2 <- glmrob(cbind(success, total - success) ~ logdose +
+ block, family = binomial, data = carrots, method = "Mqle",
+ control = glmrobMqle.control(tcc = 1.2))
> summary(Cfit2)
```

```r
Call: glmrob(formula = cbind(success, total - success) ~ logdose + block, family = binomial, data = carrots, method = "Mqle", control = glmrobMqle.control(tcc = 1.2))
Coefficients:
  Estimate  Std. Error z-value Pr(>|z|)
(Intercept)  2.3883       0.6923   3.450  0.000561 ***
logdose    -2.0491       0.3685  -5.561  2.68e-08 ***
blockB2    -0.2351       0.2122   1.108  0.267828
blockB3    -0.4496       0.2409  -1.866  0.061989 .
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Number of observations: 24
Fitted by method 'Mqle' (in 9 iterations)
(Dispersion parameter for binomial family taken to be 1)
```

robust GLM model comparison

→ `anova.glmrob`

By Andreas Ruckstuhl, based on work by Eva Cantoni (2004) JSS, and E.C.& Ronchetti (2001) JASA:

Continuing the example:

```r
> Efit2 <- glmrob(Ysum ~ Age10 + Base4 * Trt, family = poisson,
+ data = epilepsy, method = "Mqle", control = glmrobMqle.control(tcc = 1.2,
+ maxit = 100))
> summary(Efit2)
```

```r
Call: glmrob(formula = Ysum ~ Age10 + Base4 * Trt, family = poisson, data = epilepsy, method = "Mqle", control = glmrobMqle.control(tcc = 1.2, maxit = 100))
Coefficients:
  Estimate  Std. Error z-value Pr(>|z|)
(Intercept)  2.036768       0.154168  13.211  < 2e-16 ***
Age10       0.158434       0.047444   3.339  0.000840 ***
Base4       0.085132       0.004174  20.395  < 2e-16 ***
Trtprogabide -0.323886      0.087421  -3.705  0.000211 ***
Base4:Trtprogabide  0.011842     0.004967   2.384  0.017124 *
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Number of observations: 59
Fitted by method 'Mqle' (in 14 iterations)
(Dispersion parameter for poisson family taken to be 1)
```

robust GLM model comparison → `anova.glmrob`

```r
> anova(Efit3, Efit2, test = "Wald")
```

Robust Wald Test Table

```r
Model 1: Ysum ~ Age10 + Base4 + Trt
Model 2: Ysum ~ Age10 + Base4 * Trt
Models fitted by method 'Mqle'
pseudoDf Test.Stat Df Pr(>chisq)
1 55
2 54 5.6836 1 0.01712 *
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
> anova(Efit3, Efit2, test = "QD")
```

Robust Quasi-Deviance Table

```r
Model 1: Ysum ~ Age10 + Base4 + Trt
Model 2: Ysum ~ Age10 + Base4 * Trt
Models fitted by method 'Mqle'
pseudoDf Test.Stat Df Pr(>chisq)
1 55
2 54 2.9691 1 0.08487 .
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
> anova(Efit3, Efit2, test = "QDapprox")
```

Robust Quasi-Deviance Table Based on a Quadratic Approximation

```r
Model 1: Ysum ~ Age10 + Base4 + Trt
Model 2: Ysum ~ Age10 + Base4 * Trt
Models fitted by method 'Mqle'
pseudoDf Test.Stat Df Pr(>chisq)
1 55
2 54 5.6836 1 0.01712 *
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```
robustbase: plans for the future

The source package has a file named TODO. It's open to the public at https://svn.r-project.org/R-packages/trunk/robustbase/.

1. Add S4 classes for “Covariance-estimator” objects $\text{Cov}$, i.e., “location and scatter”, based on proposals of the working group in Treviso, then by Peter Filzmoser and Heinrich Fritz, and currently implemented by Valentin Todorov → useR! talk by Valentin in Friday’s focus “robustness”.

2. S4 classes for "psi-function" ($\psi$, $\rho$, $\psi'$, etc) objects, see above. Make use them, and consequently allow others than only Tukey’s biweight.

robustbase: relation to other R packages

▶ robustbase provides basic infrastructure for other R packages:
▶ Basic algorithms: R functions, sometimes also with C API.
▶ Basic classes and methods: Classes "Cov", “psi_function”, see above.
Methods for plotting; possibly in conjunction with modularizing plot.lm into separate functions

Package writing collaboration

Experiences from collaborating with a diverse group of (potential) co-authors …

Package writing collaboration: The people

The DESCRIPTION file has as authors
Author: Original code by many authors, notably Peter Rousseeuw, Christophe Croux, see file 'Copyrights';
Valentin Todorov, Andreas Ruckstuhl, Matias Salibian-Barrera, Martin Maechler
▶ meeting each other some time at first was important
▶ “talking” by e-mail: on a public (archived, searchable) mailing list
▶ talking in person from time to time — necessary (?) ! much better motivation to get things done
Package writing collaboration: The functions / classes

Integration code from four to five different partly unpublished packages needs work, but has been achieved relatively easily:

▶ 'rrcov' (Valentin),
▶ 'sfsmisc' (Andreas, Martin),
▶ 'robGLM' (Eva → Martin → Andreas),
▶ 'RobFit' (Andreas),
▶ 'roblm' (Matias).

Package writing collaboration: other software

▶ There’s the R-SIG-robust mailing list, run via “Mailman”, as R-help and quite a few other lists, → http://stat.ethz.ch/mailman/listinfo
▶ Subversion svn: Version control of files with history, backtracking, branching and merging for collaborative software development
▶ Emacs, gcc, etc.

Conclusions

▶ “robustbase” is there to be used and built upon
▶ It will be extended in several ways
▶ Collaborative package development is exciting!