# R-packages for infinitesimal robustness

Peter Ruckdeschel[1]    Matthias Kohl[2]

1 UNIVERSITÄT BAYREUTH Mathematisches Institut
Peter.Ruckdeschel@uni-bayreuth.de

www.uni-bayreuth.de/departments/math/org/mathe7/RUCKDESCHEL

2 SIRS Lab SIRS-Lab GmbH, Jena, Germany
Matthias.Kohl@stamats.de

www.stamats.de

Seminar at EPFL
October 20, 2006

# Outline of Section I

Object orientation is useful — let's use it!

- Levels of abstraction in programming
- Some paradigms in OOP
- Object orientation in S/R
- Packages for (Infinitesimal) Robust Statistics

# Outline of Section II

# Outline of Section III

## Infinitesimal Robustness in 10 slides

- $L_2$-differentiable model
- Influence curves and asymptotically linear estimators
  - Influence curves (ICs) and ALEs
- One-step-estimators
- (Shrinking) neighborhood system $\mathcal{U}_*(P_\theta, r)$ to radius $r$
- Optimally robust estimators
  - Risk: Maximal bias and Maximal MSE
  - $G$-optimal IC
- Unknown radius $r$

# Outline of Section IV:

R-Package `ROptEst` for Infinitesimal Robust Statistics

- `S4`-Classes
- Methods
- Special meta-information slots
- Semi-symbolic calculus
- Illustration II: Examples of optimally robust estimation

# Levels of abstraction in programming

(cf. [Stro:92])

- procedural programming
  - one programmer
  - separation of programming problem to *functions/procedures*
- modular programming
  - group of programmers
  - *module* $\hat{=}$ set of procedures + data on which they act
- Data abstraction
  - user defined types: *abstract data types*
  - interfacing functions
- object-orientated programming (OOP)
  - combine user-defined types with corresp. methods to a new structure *class*
  - use inheritance

# Some paradigms in OOP

- Capsulation
- Inheritance
  - methods/slots of mother class available for subclass
  - method overloading
  - extension by new methods / attributes

Lingo

- classes
  - members, attributes — in S: slots
  - methods
- instance, object
- templates

# Object Orientation in S/R

different paradigm:

- particular version of object orientation:
  *F*unction-orientated- *FOOP* as opposed to *COOP*
  - methods *not* part of object but managed by *generic functions*
  - depending on the arguments different methods are dispatched
  - example: **plot**
- for R $\geq$ `1.7.0`: use of S4-class concept, c.f. Chambers[98]

advantages:

- general interfaces (c.f. **lm**, **glm**, rlm, ) possible
- by dispatching mechanism on run-time: general code using particularized methods
- code (may / will) be:
  less redundant, better maintainable, better readable, better extensible

# Packages for (Infinitesimal) Robust Statistics

(Co-)Authors (besides M. Kohl)

- Thomas Stabla: `statho3@web.de`
- Florian Camphausen: `fcampi@gmx.de`

Organization in packages

- distr , distrEx ; [and distrSim , distrTEst ]
- M. Kohl: RandVar , ROptEst ;
  [and RobLox , RobRex , ROptRegTS ]

Availability

- distr , distrEx , distrSim , distrTEst , RandVar :
  published on CRAN; current version 1.8;
  extensive documentation available (see references)
- ROptEst , RobLox , RobRex , ROptRegTS :
  `http://www.stamats.de/RobASt.htm`

# distr: Motivation I

- Situation: algorithm / program shall cope with any distribution
- How to pass a distribution as an argument?
- Construction up to now:
  - a lot of distributions implemented to R
    — Gaussian, Poisson, Exponential, Gamma, etc.
  - for each:
    - cdf [$\hat{=}$p]
    - density / probability function [$\hat{=}$d]
    - quantile function [$\hat{=}$q]
    - function to simulate r.v.'s [$\hat{=}$r]
  - Naming convention: `<prefix><Name>`

# distr: Motivation II

- e.g. to get the median of a general distribution:

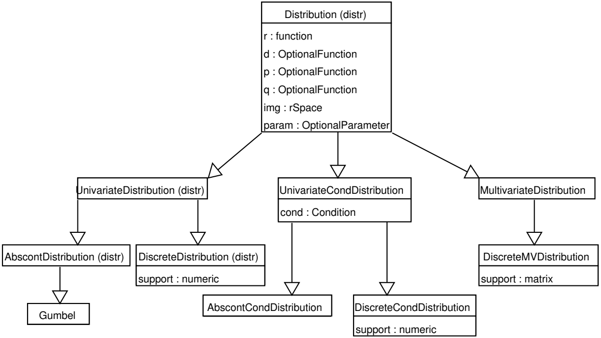
```
mymedian <- function(vtlg, ...)
{ eval(parse(text =
      paste( "x = q", vtlg,
           "(1/2,...)", sep = "")))
  return(x)}
```

- better idea: having a "variable type" *distribution* and functions p, d, q, r defined for this type
- then: **q**(x) returns the quantile function $\rightsquigarrow$

  **median** ←**function**(X){**q**(X)(0.5)}

$\Longrightarrow$ Development of this concept in package distr

# Concept of R-Packages distr




- AbscontDistribtion $\longrightarrow$ Beta , Cauchy , Chisq , Exp , Fd , Gammad , Logis , Lnorm , Norm , Td , Unif , Weibull
- DiscreteDistribtion $\longrightarrow$ Binom , Dirac , Geom , Hyper , Nbinom , Pois (. . . all from stats package)

# Methods

- overloaded: operators `"+"` , `"-"` , `"*"` , `"/"`
  e.g. Y←(3∗X+5)/4 (determined analytically.)
- group `math` of unary mathematical operations is available for objects of class `Distribution` e.g. **exp**(**sin**(3∗X+5)/4)
- `RtoDPQ` : default method for filling slots `d` , `p` , `q` on basis of simulations
- a default convolution method for two independent r.v.'s by means of FFT; c.f. K., R., & Stabla[04]
- particular methods for **plot**, **summary**,. . .
- Caveat: arithmetics operates on underlying random variables, *not* on distributions

# Example: arithmetics for distribution objects

```
> require("distr")
Loading required package: distr
[1] TRUE
> N <- Norm(mean = 2, sd = 1.3)
> P <- Pois(lambda = 1.2)
> Z <- 2 * N + 3 + P # exact transformation
Distribution Object of Class: AbscontDistribution
> plot(Z)
> p(Z)(0.4)
[1] 0.002415384
> q(Z)(0.3)
[1] 6.70507
> r(Z)(10)
 [1] 11.072931  7.519611 10.567212  ....
 [9]  9.358270 10.689527
> Znew <- sin(abs(Z)) # by simulations
> plot(Znew)
> p(Znew)(0.2)
```

# Contents of `distrEx`

Package `distrEx` extends `distr` and includes

- a general expectation operator to a given distribution $F$
- several functionals on distributions like
  median, var, sd, MAD and IQR
- several distances between distributions
  (e.g. Kolmogoroff–, Total-Variation–, Hellinger-distance)
- (factorized) conditional distributions
- (factorized) conditional expectations

# Example: expectation operator

- for a normal variable $D_1$ try to realize $\mathrm{E}\, D_1$, $\mathrm{E}\, D_1^2$, and for some $m_1 \in \mathbb{R}$, $\mathrm{E}(D_1 - m_1)^2$

```r
require("distrEx")
D1 <- Norm(mean=2)
m1 <- E(D1)                    # = 2
E(D1, function(x){ x^2 })  # E(D_1^2)
```

- now —without changing the code— the same for a Poisson variable; this gives the same calls but different dispatched methods

```r
D1 <- Pois(lambda=3)
m1 <- E(D1)                    # = 3
E(D1, function(x){ x^2 })
```

# Illustration 1: CLT —under arbitrary distribution

- we want to illustrate the Lindeberg-Lévy theorem
- input should be any univariate distribution `Distr`
- notation: $X_i \stackrel{\text{i.i.d.}}{\sim} F$, $S_n = \sum_{i=1}^n X_i$, $T_n = (S_n - \mathrm{E}\, S_n)/\sqrt{\mathrm{Var}\, S_n}$
- output: sequence of length `len` of plots of $\mathcal{L}(T_n)$
- realized in `illustrateCLT(Distr, len)`
- essential code
  - a function for standardizing and centering

    ```
    make01 <- function(x)(x-E(x))/sd(x)
    ```

  - update in a loop starting with `Sn <- 0`

    ```
    Sn <- Sn + Distr
    Tn <- make01(Sn)
    ## here: Distr is absolutely continuous
    dTn <- d(Tn)(x)
    ```

# Illustration 2: Minimum-distance- and ML-functionals

- we want to estimate the parameter $\theta$ in a parametric family
- methods: minimum-distance and ML
- in both cases in an optimization a member in the class is distinguished as "closest" to the data

- input: data and parametric model
- output: estimate

- implementation: parametric model as class with slots
  - name, distribution ,
  - \+ additionally: a slot modifparameter, a function realizing theta$\mapsto$ $P_{\text{theta}}$
- generic functions MDE(**model**,**data**,**distance**), MLE(**model**,**data**)

# Illustration 2: Minimum-distance- and ML-functionals II

essential code

- to fit a distribution distr to **data** according to criterium ( distr ,**data**) we use

```
fitParam <- function(model, data0, criterium ....)
  { #define a function in theta to be optimized:
    ftoOptimize <- function(theta)
       {Ptheta  <- modifparameter(model)(theta)
        criterium(Ptheta,data0)        }
    #use "optimize" or "optim" dep. on dim; here:
    theta <- optimize(f = ftoOptimize,
               interval = searchinterval0, ...)$minimum
    return(theta)}
```

- criterium: e.g. negative log-likelihood or distance (e.g. Kolmogoroff-.) theoretical : empirical distribution

# Illustration 3: Deconvolution I

- Situation: $X \sim K$, $\varepsilon \sim F$, stoch. independent; $Y = X + \varepsilon$
- goal: reconstruction $X$ by means of $Y$
- methods: $\mathrm{E}[X|Y]$, $\mathrm{postmode}(X|Y)$
- input: any univariate distributions $K =$ Regr, $F =$ Error
- output: mappings $y \mapsto \mathrm{E}[X|Y = y],\ \mathrm{postmode}(X|Y = y)$
- realized by means of PrognCondDistribution(Regr, Error)

$\rightsquigarrow$ generates $\mathcal{L}(X|Y = y)$ where $y$ is coded as parameter cond

# Illustration 3: Deconvolution II

essential code

- filling of the slots r, d, p, q for some machine-eps

```
rf←function(n,cond) cond - r(Error)(n)
df←function(x,cond) d(Regr)(x)*d(Error)(cond-x)
qf←function(x,cond) cond-q(Error)(1-x)
pf←function(x,cond) integrate(df, low=q(Error)(eps),
                        up=x, cond=cond)$value
```

- conditional expectation $\mathrm{E}[X|Y=y]$

```
PXy ← PrognCondDistribution(Regr, Error)
E(PXy, cond=y)
```

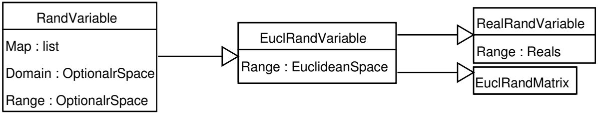
- posterior mode $\mathrm{postmode}(X|Y=y)$

```
post.mod ← function(cond, e1) {
    optimize(f = d(PXy), c(q(PXy)(eps, cond),
        q(PXy)(1-eps, cond)), cond = cond)$maximum}
```

# R-Package RandVar

Random variable as a class concept

- Definition




Mathematical operations

- there are **many**. . .
- essentially: usual vector arithmetic available for conformal `"RealRandVector"` , `"EuclRandVector"` and `"EuclRandMatrix"`
- also: group `math` , e.g. sin, cos, exp, (log), ($\sqrt{\ }$, . . .)

# References:


- P.J. Bickel (1981): Quelques aspects de la statistique robuste. In *Ecole d'été der probabilités de Saint Flour IX-1979*, Lect. Notes Math. 876, p.2–72.
- H. Rieder (1994): *Robust asymptotic statistics*. Springer.
- ——, M.K., and P.R. (2001): The Costs of not Knowing the Radius. Submitted. `http://www.uni-bayreuth.de/departments/math/org/ /mathe7/RIEDER/pubs/RR.pdf`.
- P.R. and H. Rieder (2004): Optimal IC's for general loss functions. *Statistics and Decisions* **22**, p.201–223
- M.K.(2005): *Numerical contributions to the asymptotic theory of robustness*. Dissertation, Universität Bayreuth. Available under `http://stamats.de/ThesisMKohl.pdf`

# $L_2$-differentiable model

$\mathcal{P} = \{P_\theta \mid \theta \in \Theta\}$, $\Theta \subset \mathbb{R}^k$ open

- Examples:
  - Gaussian location:
    $\mathcal{P}_1 = \{\mathcal{N}(\theta, 1) \mid \theta \in \Theta\}$, $\Theta = \mathbb{R}$
  - Gaussian scale:
    $\mathcal{P}_2 = \{\mathcal{N}(1, \theta(= \sigma^2)) \mid \theta \in \Theta\}$, $\Theta = (0, \infty)$
  - Gaussian location and scale:
    $\mathcal{P}_3 = \{\mathcal{N}(\theta_1, \theta_2) \mid \theta \in \Theta\}$, $\Theta = \mathbb{R} \times (0, \infty)$

$L_2$-differentiability

:: $\sqrt{dP_{\theta+h}} = \sqrt{dP_\theta}\,(1 + \frac{1}{2}\Lambda_\theta^\tau h) + \mathrm{o}(|h|)$

- also: Fisher-information $\mathcal{I}_\theta := \int \Lambda_\theta \Lambda_\theta^\tau \, dP_\theta$ finite and regular

# $L_2$-differentiable model II

- Consequence:
  - $P^n_{\theta+h/\sqrt{n}}$ and $P^n_\theta$ are contiguous
  - Loglikelihood-expansion:

    $$\log dP^n_{\theta+h/\sqrt{n}}/P^n_\theta = \frac{1}{\sqrt{n}}\sum_i h^\tau \Lambda_\theta(x_i) - \frac{1}{2}h^\tau \mathcal{I}_\theta h + \mathrm{o}_{P^n_\theta}(1)$$

  $\Longrightarrow$ model is LAN (locally asymptotically normal)
- differentiable parameter transformation

  $$\tau\colon \mathbb{R}^k \to \mathbb{R}^p, \qquad \tau'(\theta) = D = D(\theta)$$

  Examples:
  - estimation of sd in scale model $\mathcal{P}_2$: $\tau(x) = \sqrt{x}$
  - nuisance parameter:
    estimation of location $\theta_1$ without knowing scale $\theta_2$ in $\mathcal{P}_3$

# Influence curves (ICs) and ALEs

[partial] Influence curve ([p]IC)

$$\eta_\theta \in L_2^p(P_\theta) \quad \text{s.t.} \qquad \mathrm{E}_\theta\, \eta_\theta = 0, \quad \mathrm{E}_\theta\, \eta_\theta \Lambda_\theta^\tau = \mathbb{I}\,[D] \qquad\qquad (\mathrm{E}_\theta = \mathrm{E}_{P_\theta})$$

here: pIC as a possible linearization of an estimator

Asymptotically linear estimator (ALE): estimators with expansion

$$\sqrt{n}\,(S_n - \theta) = \frac{1}{\sqrt{n}} \sum_{i=1}^{n} \eta_\theta(X_i) + \mathrm{o}_{P_\theta^n}(n^0)$$

for some pIC $\eta_\theta$

- conditions for pIC $\iff$ local uniform as. normality of ALE

Examples

- in $\mathcal{P}_1$: $S_n = \bar{X}_n$ — $\eta_\theta(x) = x - \theta$,
- in $\mathcal{P}_1$: $S_n = \mathrm{Median}_n$ — $\eta_\theta(x) = \sqrt{\pi/2}\,\mathrm{sign}(x - \theta)$

# One-step-estimators

defined to starting estimate $\theta_0$ and IC $\eta$ as

$$S_n^{(1)} := \tau(\theta_0) + \frac{1}{n}\sum_{i=1}^{n} \eta_{\theta_0}(X_i)$$

## Theorem ("One step is enough"[Ri:94])

*Assumptions:*

- $\sqrt{n}(\theta_0 - \theta) = \mathrm{O}_{Q_n^{(n)}}(1)$ *uniformly for all* $Q_n^{(n)}$ *in the neighborhood*
- *IC* $\eta_\theta$ *is bounded and* $\lim_{h\to 0}\sup_x |\eta_{\theta+h}(x) - \eta_\theta(x)| = 0$

THEN $S_n^{(1)}$ *is an ALE to pIC* $\eta_\theta$*:*

$$\boldsymbol{S_n^{(1)} - \tau(\theta) = \frac{1}{n}\sum_{i=1}^{n}\eta_\theta(X_i) + \mathrm{o}_{P_\theta^n}(n^{-1/2})}$$

# (Shrinking) neighborhood system $\mathcal{U}_*(P_\theta, r)$ to radius $r$

- $\boldsymbol{\mathcal{U}_*(P_\theta, r)}$: all $Q_n^{(n)} = \bigotimes_{i=1}^{n} Q_{n,i}$ with $\boldsymbol{d_*(Q_{n,i}, P_\theta) \leq r/\sqrt{n}}$ for

  $* =$c *convex contaminations:* $d_c(P, Q)$:
  smallest $r \geq 0$ s.t. $\exists$ p.m. $H$ with $Q = (1 - r)P + rH$

  $* =$v *total variation:* $2d_v(P, Q) = \int |dP - dQ|$

  $* =$h *Hellinger:* $2d_h(P, Q)^2 = \int (\sqrt{dP} - \sqrt{dQ})^2$

  THEN for all such $Q_n^{(n)} \in \mathcal{U}_*(P_\theta, r)$

  $$\sqrt{n}\,\Big(S_n^{(1)} - \tau(\theta) - \tfrac{1}{n}\sum_{i=1}^{n} \int \eta_\theta\, dQ_{n,i}\Big) \circ Q_n^{(n)} \xrightarrow{\mathrm{w}} \mathcal{N}_p(0, \mathrm{E}_\theta\, \eta_\theta \eta_\theta^\tau)$$

- shrinking necessary to control bias and variance simultaneously (for fixed radius, bias is of order $\sqrt{n}$ )

# Risk: Maximal bias and Maximal MSE

Fact (Maximal asymptotic bias on $\mathcal{U}_*(P_\theta, r)$: [Ri:94])

— *explicit terms:*

- $r\omega_*(\eta_\theta) := \sup_{Q_n^{(n)} \in \mathcal{U}_*(P_\theta, r)} \frac{1}{n}\sum_{i=1}^n \int \eta_\theta \, dQ_{n,i}$

THEN

$$
\begin{aligned}
*=\text{c} \quad & \omega_c(\eta_\theta) = \sup |\eta_\theta| \\
*=\text{v}(p=1) \quad & \omega_v(\eta_\theta) = \sup \eta_\theta - \inf \eta_\theta \\
*=\text{h} \quad & \omega_h(\eta_\theta) \doteq \sqrt{8}\ \mathrm{maxev}(\mathrm{E}_\theta\, \eta_\theta \eta_\theta^\tau)
\end{aligned}
$$

Maximal asymptotic MSE on $\mathcal{U}_*(P_\theta, r)$:

$$\mathrm{asMSE}(\eta, r) = \mathrm{E}_\theta\, |\eta_\theta|^2 + r^2 \omega_*^2(\eta_\theta)$$

MSE problem: to given $r \geq 0$, find pIC $\hat{\eta}_r$ minimizing $\mathrm{asMSE}$

# MSE-optimal IC

## Theorem (Solution to MSE problem: [Ri:94])

*to given $\theta$ (suppressed in notation)*

$* = \mathrm{c}$ $\boldsymbol{\hat\eta_r = Y \min\{1, b/|Y|\}}$ *for* $Y = A\Lambda - a$ *(Hampel-form)*

*where* $b > 0$ *s.t.* $\boldsymbol{r^2 b = \mathrm{E}(|Y| - b)_+ =: \gamma_c}$

$* = \mathrm{v}(p = 1)$ $\boldsymbol{\hat\eta_r = c \wedge A\Lambda \vee (c + b)}$

*where* $b > 0$ *s.t.* $\boldsymbol{r^2 b = \mathrm{E}(c - A\Lambda)_+ =: \gamma_v}$

$* = \mathrm{h}$ $\boldsymbol{\hat\eta_r = D\mathcal{I}^{-1}\Lambda}$

*for* $A \in \mathbb{R}^{p \times k}$, $a \in \mathbb{R}^p$, $c \in (-b, 0)$ *Lagrange multipliers s.t.* $\hat\eta_r$ *is an IC*

# $G$-optimal IC

## Theorem (More general risk: [R.:Ri:04])

- *fix $\theta$; assume that maximal asymptotic risk on $\mathcal{U}_*(P, r)$ representable as*

  $$\tilde{G}(\eta, r) = G(r\omega_*(\eta), \sigma_\eta) \quad \textit{for}$$

  - $\sigma_\eta^2 = \mathrm{E}_P\,|\eta|^2$
  - *$G = G(w, s)$ convex, isotone in both arguments*

THEN *for $* =$c or $* =$v($p = 1$):*
*again asMSE-type of solutions, but $b$ determined as*

$$r\sigma_\eta G_w(rb, \sigma_\eta) = \gamma_* G_s(rb, \sigma_\eta)$$

- examples:
  $G = \int |x|^q \, d\mathcal{N}(w, s)$ ($L_q$-risk),
  $G = \int \mathrm{I}(|x| > \tau) \, d\mathcal{N}(w, s)$ (Maximin covering probability)

# Unknown radius $r$

- situation: $r$ not known, only available information $r \in [r_l, r_u]$
- relative inefficiency of $\eta_r$ when used at radius $s$:

$$\rho(r,s) := \max_{\mathcal{U}} \mathrm{asRisk}(\eta_r, s) / \max_{\mathcal{U}} \mathrm{asRisk}(\eta_s, s)$$

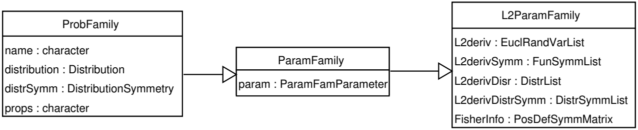
- minimax radius/inefficiency:
  $r = r_0$ such that $\hat{\rho}(r)$ is minimal for $\hat{\rho}(r) := \sup_{s \in [r_l, r_u]} \rho(r,s)$

Theorem (Radius-minimax procedure [R.:Ri:04])

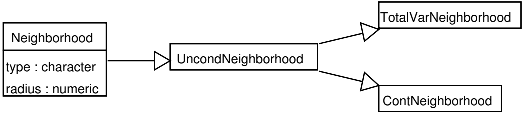
*For all homogeneous $G$ (i.e.; $G(\nu w, \nu s) = \nu^{\alpha} G(w,s)$), the radius-minimax pIC does* **not** *depend on $G$!*

# Classes

- $L_2$-differentiable model:

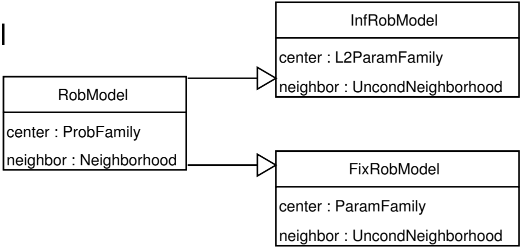



- neighborhood system to some given radius $r$

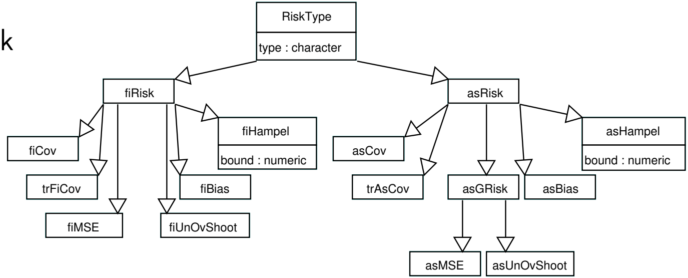


# Classes II

- robust model

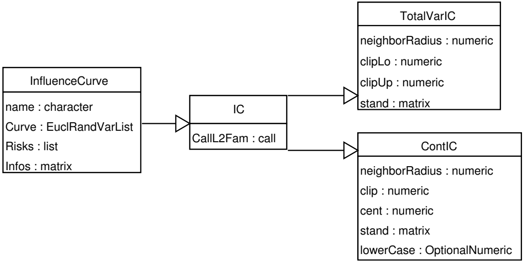



- risk



# Classes III

- IC



# Methods I

- accessor and replacement functions, **show**, **plot**
- `addInfo` , `addProp` , `addRisk`
- `checkL2deriv` , `checkIC` , `evalIC` , `getRiskIC` , `infoPlot` , `ksEstimator` , `leastFavorableRadius` , `locMEstimator` , `oneStepEstimator` , `optIC` , `optRisk` , `radiusMinimaxIC`
- easy generating functions for implemented $L_2$-families like `NormLocationScaleFamily` , `BinomFamily`

# Special meta-information slots

- information gathered during generation of objects is stored in information slots, e.g.

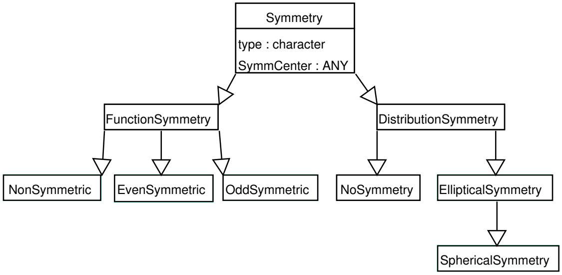
```
    ### props:
[1] "The normal location and scale family is invariant under"
[2] "the group of transformations 'g(x) = sd*x + mean'"
[3] "with location parameter 'mean' and scale parameter 'sd'"
```

# Semi-symbolic calculus: Situation

- Situation:
  - we have a certain abstract property for our model (e.g. symmetry)
  - whether this property holds or not cannot be decided (exactly) on basis of numeric evaluations (e.g. convergence?)
  - as a logical statement we can "calculate" with this property and even deduce further properties
  - important for evaluation of high dimensional integrals

# Semi-symbolic calculus: Approach and Realization

- Approach
  - in classical (linear) hierarchical inheritance relations of objects: not clear in which order we should inherit abstract properties...
  - introduce symbolic/logical flags as members(slots) of objects and interfere into dispatching mechanism...
- Realization



# Setup of the Examples I

1. Estimation of location and scale
   - $\texttt{X}$ a contaminated sample from $\mathcal{N}(\texttt{mean}, \texttt{sd}^2)$
   - goal: optimally robust estimation of `mean` and `sd`
   - $\hat{=}$ example for an existing implemented model
2. Generation of a new $L_2$-differentiable family:
   - censored Poisson distribution with parameter $\lambda > 0$, i.e. we only observe realizations $> 0$
   - goal: optimally robust estimation of $\lambda$
   - $\hat{=}$ example for the new implementation of a model and then use of existing methods (without new programming!)

# Setup of the Examples II

3. Estimation of regression and scale
   - X a contaminated sample from regression model $\mathtt{Y} = \mathtt{X}^\tau \mathtt{theta} + \varepsilon$, $\varepsilon \sim \mathcal{N}(0, \mathtt{sd}^2)$
   - goal: estimation of theta (and sd) at (artificial) data set exAM by Antille and May (c.f. robustbase )
   - optimally robust: (depending on neighborhood type)
     - Huber- and Hampel-Krasker-type ICs (without scale)
     - with scale: weight $w = \min\{1, b/\sqrt{|A_1 X|^2 u^2 + a_2(u^2 - a_3)^2}\}$ for $u$ residual and $b$, $A_1$ $a_2$, $a_3$ constants determined in the algo's depending on the radius (independent of $Y$ but dependent on $X$)

# Summary

covered so far:

- computation of optimal ICs for all(!) $L_2$-diff'ble models based on univariate distributions
- Kolmogorov minimum distance estimator as starting estimator
- provide optimally robust estimators by means of one-step constructions

# Open Issues

1. use of S-classes for model formula $\rightsquigarrow$ `rlm` extending `lm` also available for infinitesimal robustness
2. better and standardized user-interfaces
3. (more) standardized output
4. use of other robust diagnostic plots...
5. reporting: use of XML for the storage of meta-information about generated objects
6. use of package `Matrix`
7. one generic method for `ksEstimator`
8. extension of class `RiskType` : getRiskIC
9. mStepEstimator `m = Inf` : $\widehat{=}$ iteration until "convergence"
10. better use of symmetry and group invariances
11. special group generic for invertible operators for the exact determination of image distributions
12. `liesInSupport` : allow for logical operations for slot `'img'` of distributions
13. Lower case for Dimension $> 1$

... many more

# Bibliography


J. M. Chambers.
*Programming with Data. A guide to the S language.*
Springer, 1998.
URL http://cm.bell-labs.com/cm/ms/departments/sia/Sbook/.

R Development Core Team.
R*: A language and environment for statistical computing*.
R Foundation for Statistical Computing, Vienna, Austria, 2005
URL http://www.R-project.org.

M. Kohl, P. Ruckdeschel, and T. Stabla.
General Purpose Convolution Algorithm for Distributions in S4-Classes by means of FFT. Technical Report. Feb. 2005.
URL http://www.uni-bayreuth.de/departments/math/org/mathe7/RUCKDESCHEL/pubs/comp.pdf

P. Ruckdeschel, M. Kohl, T. Stabla, and F. Camphausen.
S4 Classes for Distributions.
*R-News*, **6**(2): 10–13.
http://CRAN.R-project.org/doc/Rnews/Rnews_2006-2.pdf.
Also available as manual for packages distr, distrSim, distrTEst version 1.8, Oct. 2006.
URL http://www.uni-bayreuth.de/departments/math/org/mathe7/DISTR/distr.pdf.