Statistical Disclosure Control: Methods and Software Development in R

Matthias Templ

Vienna University of Technology
and
Statistics Austria

Prague

October, 4. 2006

Outline of the talk

Generally speaking:

- Why using R (for statistical disclosure control (SDC))?

Methods, Software implementation, Examples:

(I will show some Methods in more detail - other implemented methods will be not shown)

- Some aspects of Microaggregation
- Very short overview of other Packages dealing with microdata ...
- Linear Programming with R: Just showing an attacker problem on marginal tables.

Existing Software for SDC

Argus twins on

http://neon.vb.cbs.nl/casc/

μ-**Argus** is to software for protecting micro data and it is not really flexible, has poor data import/export facilities, does not provide reproducibility of results, is a stand alone software (not embedded in an (statistical) software), consists of many strange errors, does not provide statistics and diagnostics and has some other disadvantages too. *

τ-**Argus** is for protecting hierarchical tables.

*Of course, if you have a look at it, probably you will find some positive aspects which I cannot see.

Using R for SDC

Some aspects of using R (for Data Disclosure):

R Characterization:

- R is the "lingua franca" for data analysis and statistical computing.
- R is a modern object-oriented high-level programming language and runs under all platforms. Turn your ideas into software easily.

Using R for SDC

Some aspects of using R (for Data Disclosure):

- Characterization:
  - R is the "lingua franca" for data analysis and statistical computing.
  - R is a modern object-oriented high-level programming language and runs under all platforms. Turn your ideas into software easily.

- Useful Features for our purpose:
  - Data Import/Export facilities.
  - Graphical Excellent. Graphics can be useful during the process of data disclosure.
  - Implemented functions and algorithms.

Using R for SDC

Some aspects of using R (for Data Disclosure):

- Characterization:
  - R is the "lingua franca" for data analysis and statistical computing.
  - R is a modern object-oriented high-level programming language and runs under all platforms. Turn your ideas into software easily.
- Useful Features for our purpose:
  - Data Import/Export facilities.
  - Graphical Excellent. Graphics can be useful during the process of data disclosure.
  - Implemented functions and algorithms.
- Literate Programming and Development Tools:
  - `Sweave` & LaTeX for dynamical reports.
  - Lots of development tools for R. Own packages for data disclosure can be created with online-help files.
  - One package, one maintainer, many contributors.

Using R for SDC

Some aspects of using R (for Data Disclosure):

Characterization:

- R is the "lingua franca" for data analysis and statistical computing.
- R is a modern object-oriented high-level programming language and runs under all platforms. Turn your ideas into software easily.

Useful Features for our purpose:

- Data Import/Export facilities.
- Graphical Excellent. Graphics can be useful during the process of data disclosure.
- Implemented functions and algorithms.

Literate Programming and Development Tools:

- `Sweave` & LaTeX for dynamical reports.
- Lots of development tools for R. Own packages for data disclosure can be created with online-help files.
- One package, one maintainer, many contributors.

Free Use:

- R is open source and freely available. Everyone can see the code and can learn from the code, can change the code for himself and can add code.

SDC on (Business) Microdata

We want to give data to researchers and must preserve confidentiality. This can be done via

- Model Based Server:
  - Disadvantages: In general - the user can not look at the data and therefore the quality of the analysis is bad. Limited number of methods implemented. Very inflexible. Restrictive. Expensive.
  - Advantages: Restrictive.
- Remote Access:
  - Disadvantages: Expensive to control outputs. Open confidential questions. Not so restrictive as Model Based Server
  - Advantages: Flexible. Not restrictive.
- **Perturbation of microdata**.
  - Advantages: Lower costs. The level of perturbation can be controlled. Disadvantages: Manipulated data

SDC on Categorical Data

Definition of key variables:

Geographical Info, Sex, job, education, ...

Consider this subset of data with 3 key variables:

observation	Code	Sex	Job	income
334	1030	m	employee	31200
335	8250	m	professor	59100
336	1030	m	employee	29112
...	...	...	...	...
1455	1030	m	employee	20421

SDC on Categorical Data

Definition of key variables:

Geographical Info, Sex, job, education, ...

Consider this subset of data with 3 key variables:

observation	Code	Sex	Job	income
334	1030	m	employee	31200
335	**8250**	**m**	**professor**	**59100**
336	1030	m	employee	29112
...	...	...	...	...
1455	1030	m	employee	20421

$\longrightarrow$ There is only **one** person which life in district **8250**, is a **male** and has a job as **professor** (who is it?) - we all know his (fictive) income (and many things more).

SDC on Categorical Data

We must change something on the data:

- A (categorical or binary) value of the key variables, or
- The (continuous) value of income

Microaggregation (for Cont. Data)

- ↪ Microaggregation is one concept of data manipulation.

- ↪ With Microaggregation k observations of microdata have to be aggregated and replaced.

- ↪ Microaggregation on data with n observations combines k observations and calculates a measure of location (e.g. the mean).

- ↪ Microaggregation will be shown graphically in detail later.

Microaggregation

Note:

- The univariate **and** multivariate structure of our data should not be destroyed:
- Aggregation of **similar** observations.

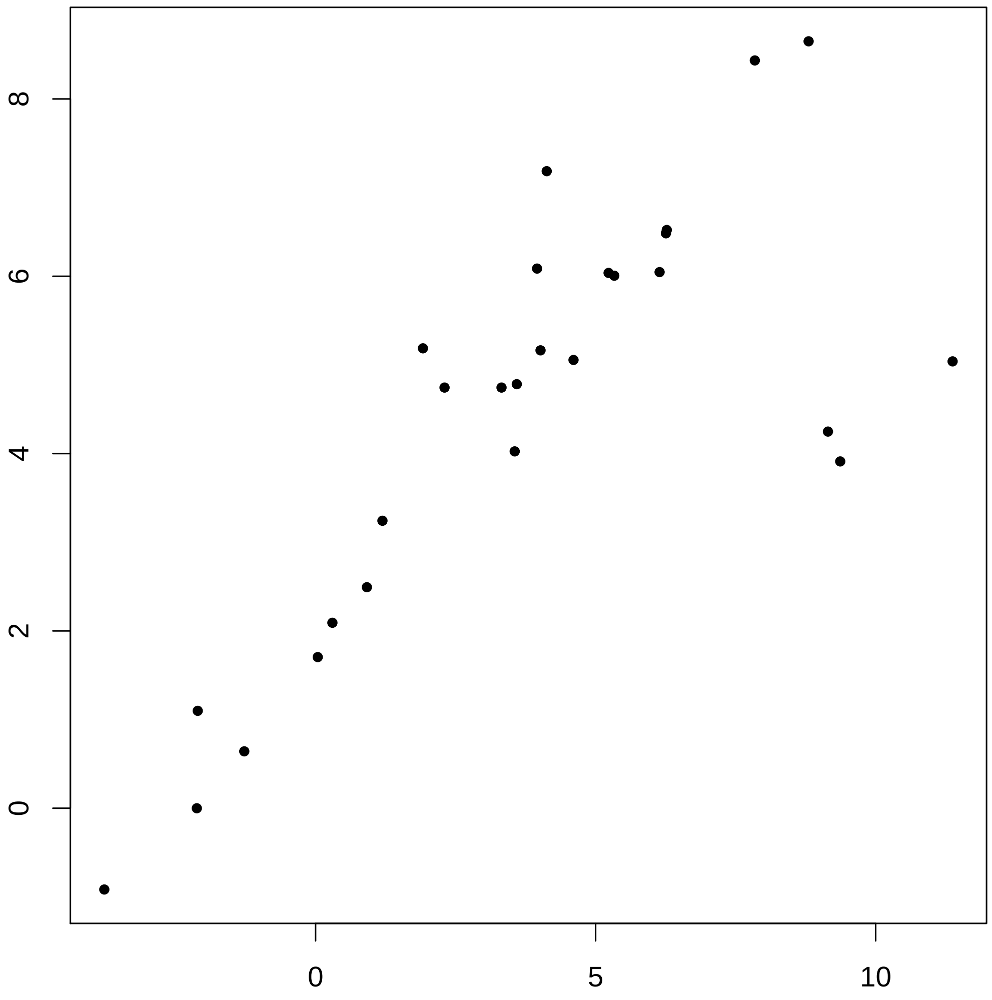
There are some different concepts for microaggregation:

- Sorting on a single variable, sorting on each variable (DeStatis)
- Clustering approach: Sorting on most influential variable in each cluster.

- Nearest neighbor approach: `mdav` (μ-Argus)
- Projection methods: PCA, robust PCA, PCA with Projection Pursuit
- Combinations of Methods

Algorithm 1: simple

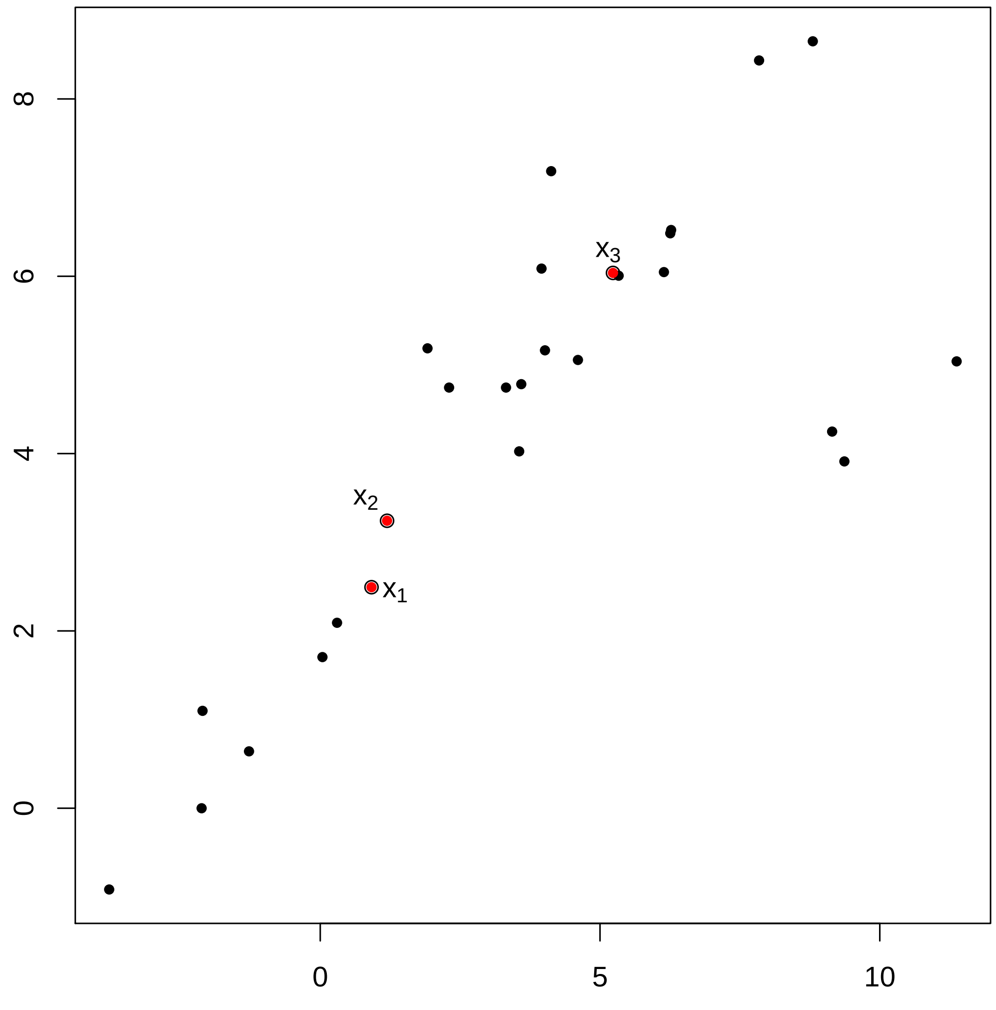
random selection:



Algorithm 1: simple

random selection:

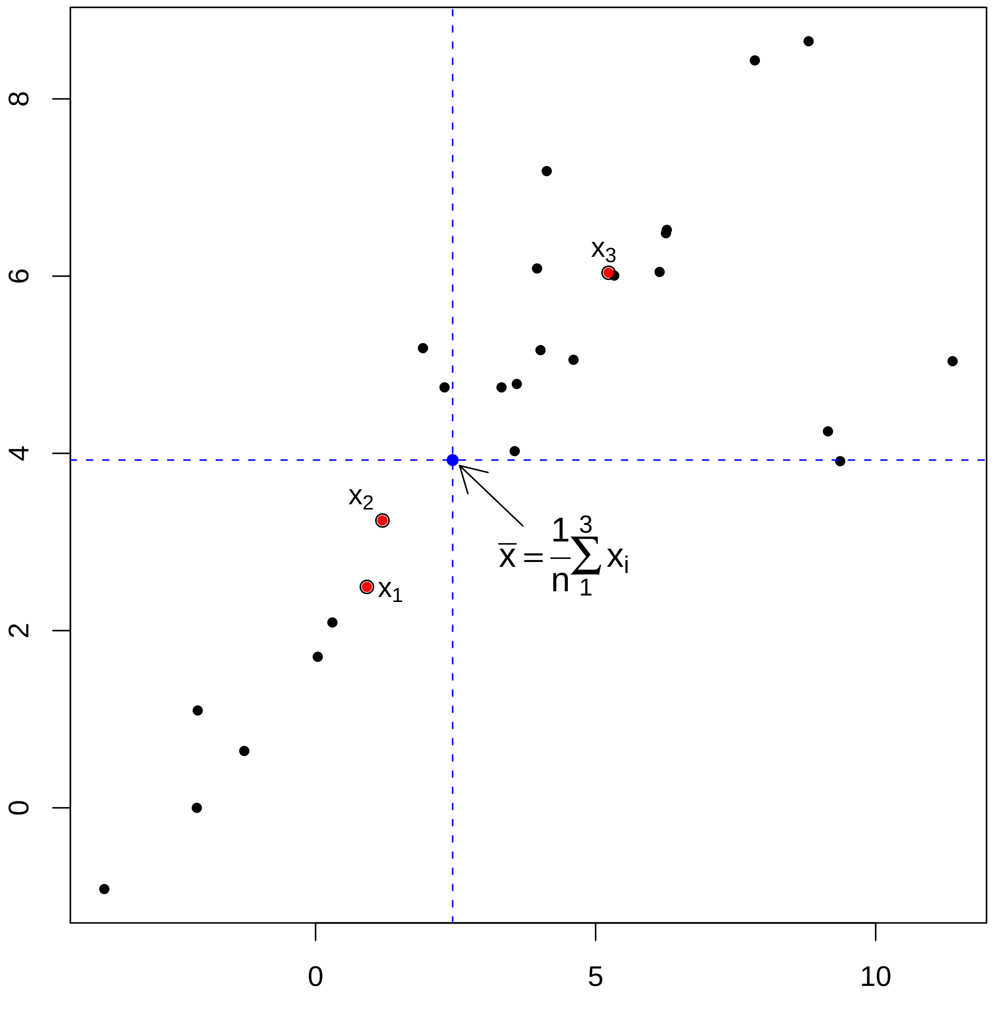
- select e.g. 3 points, randomly



Algorithm 1: simple

random selection:

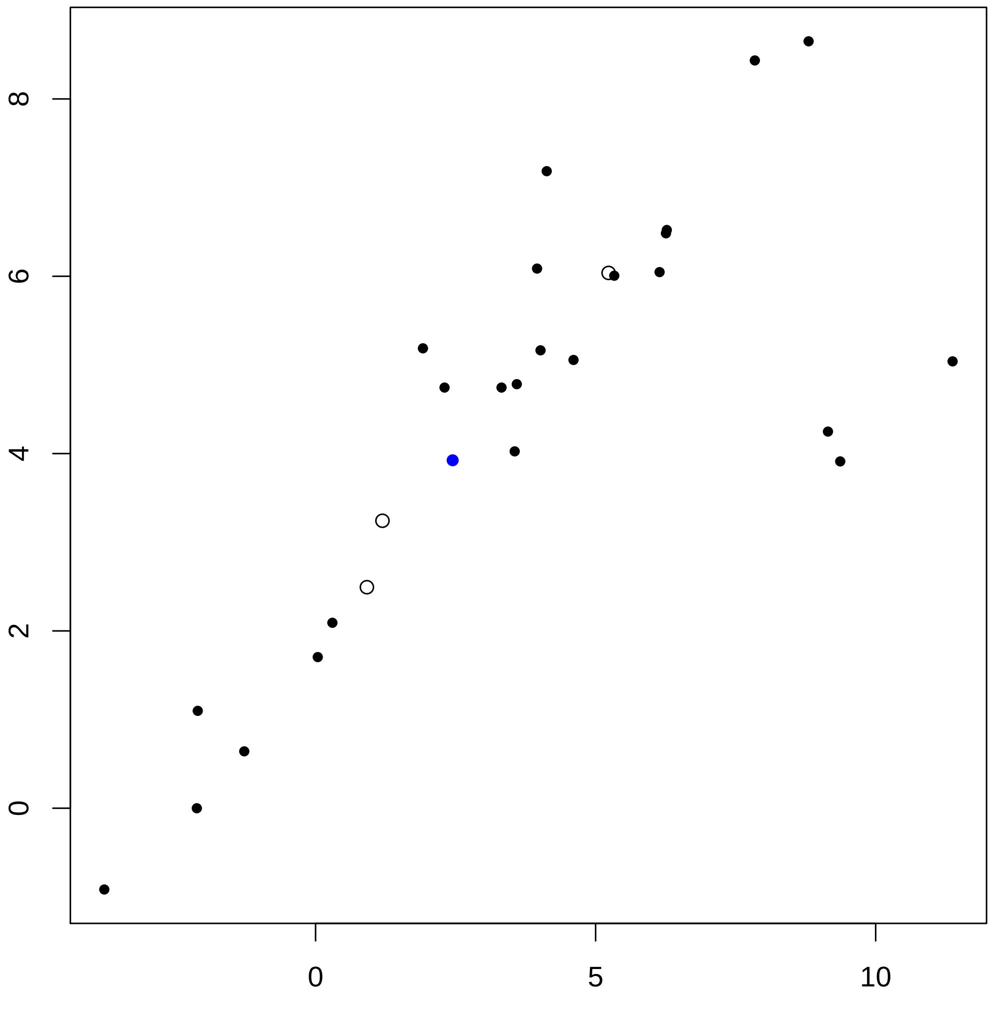
- select e.g. 3 points, randomly
- calculate measure of location (e.g. mean)



Algorithm 1: simple

random selection:

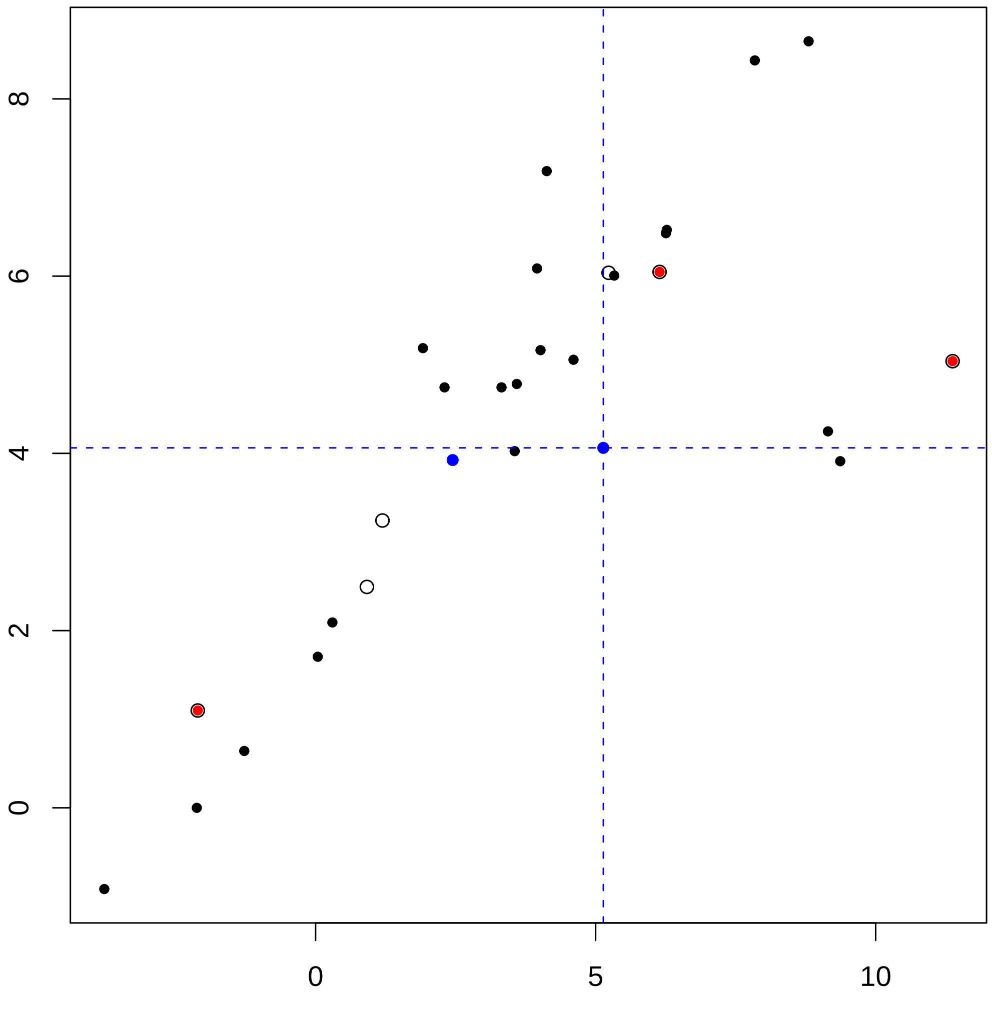
- select e.g. 3 points, randomly
- calculate measure of location (e.g. mean)
- substitute



Algorithm 1: simple

random selection:

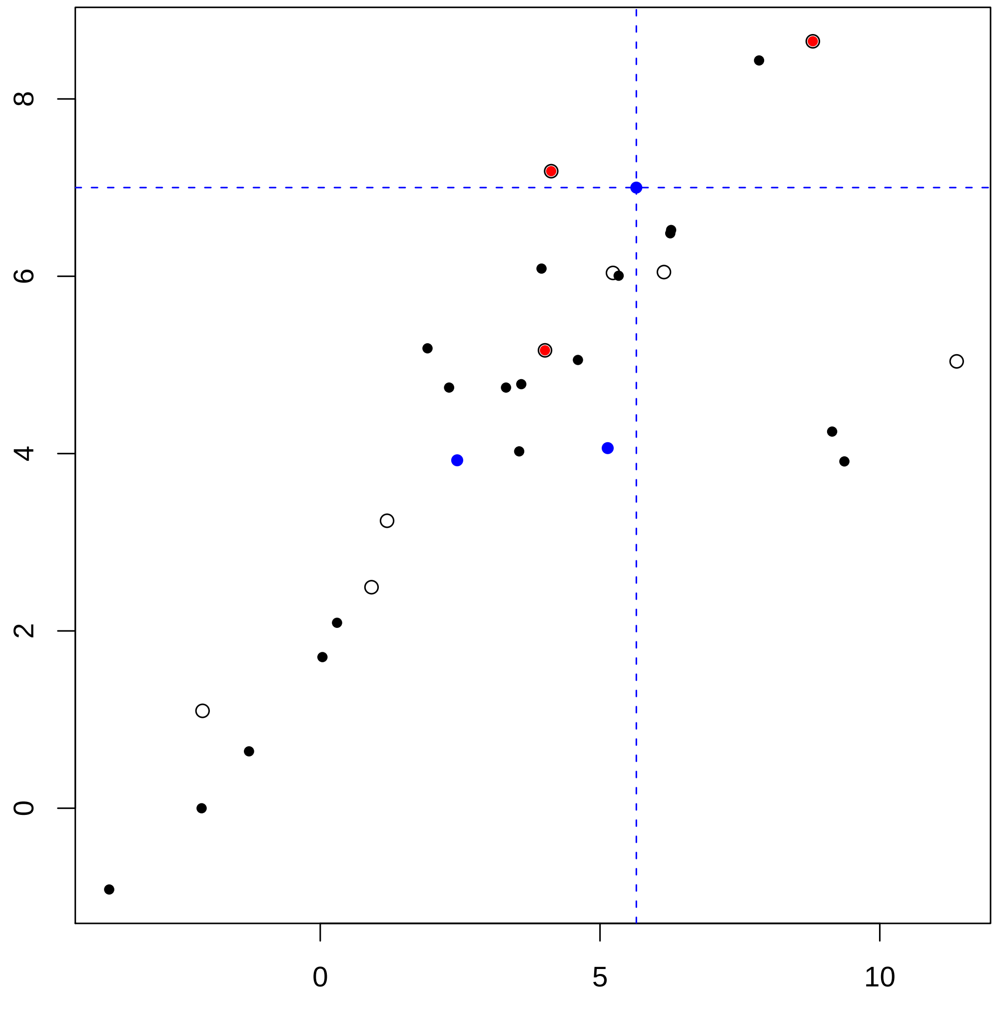
- select e.g. 3 points, randomly
- calculate measure of location (e.g. mean)
- substitute
- untill all points are microaggregated



Algorithm 1: simple

random selection:

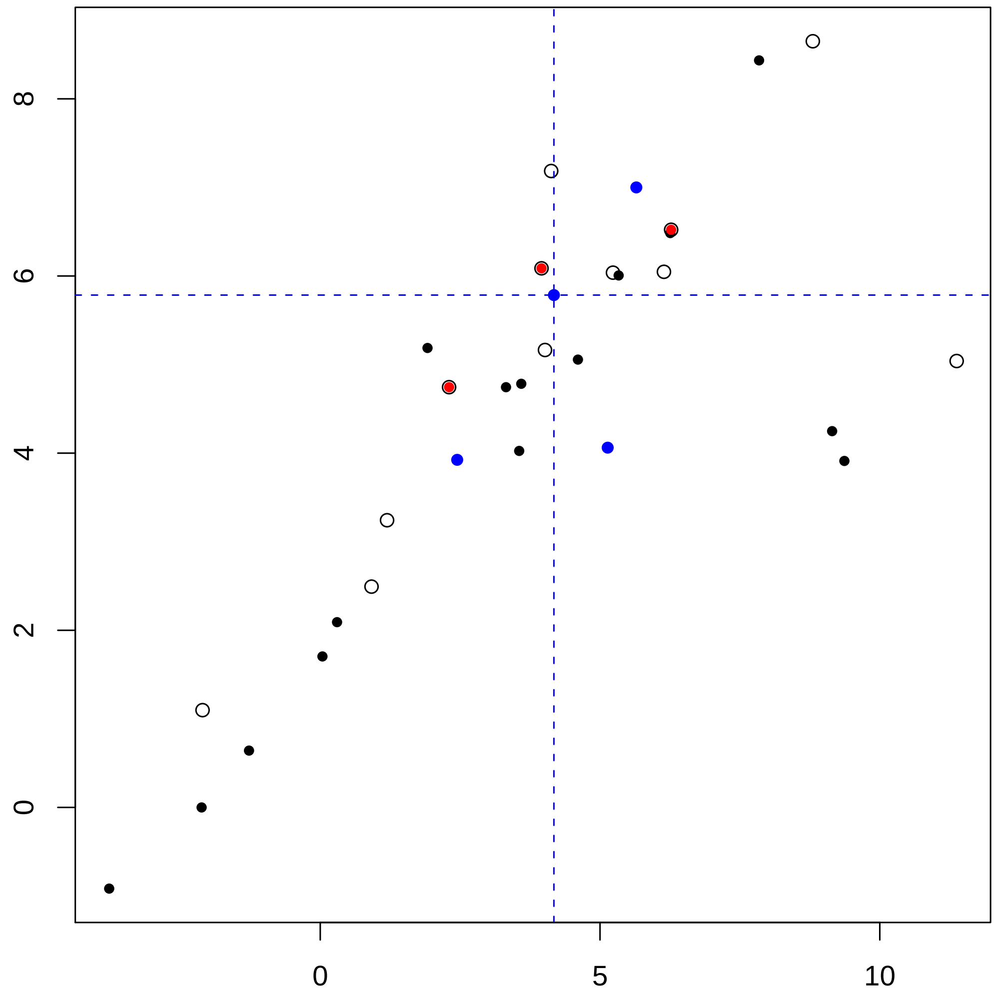
- select e.g. 3 points, randomly
- calculate measure of location (e.g. mean)
- substitute
- untill all points are microaggregated



Algorithm 1: simple

random selection:

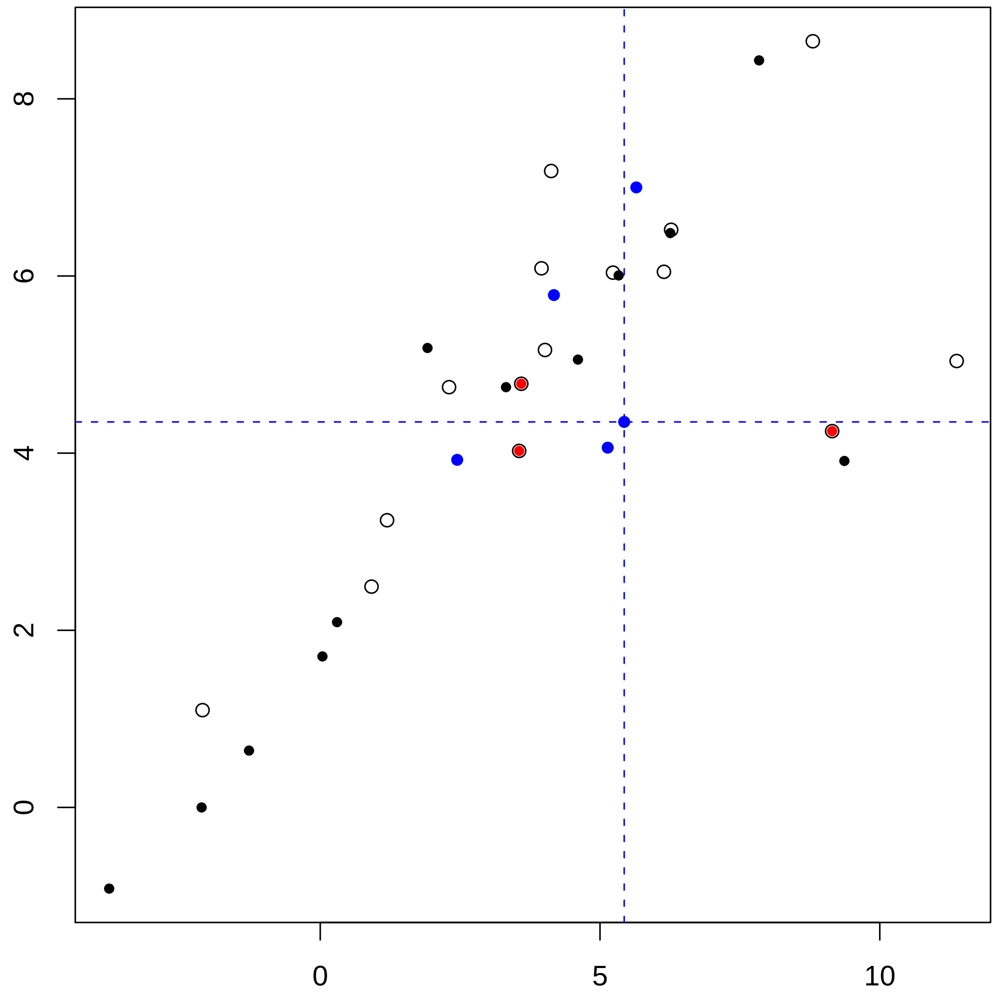
- select e.g. 3 points, randomly
- calculate measure of location (e.g. mean)
- substitute
- untill all points are microaggregated



Algorithm 1: simple

random selection:

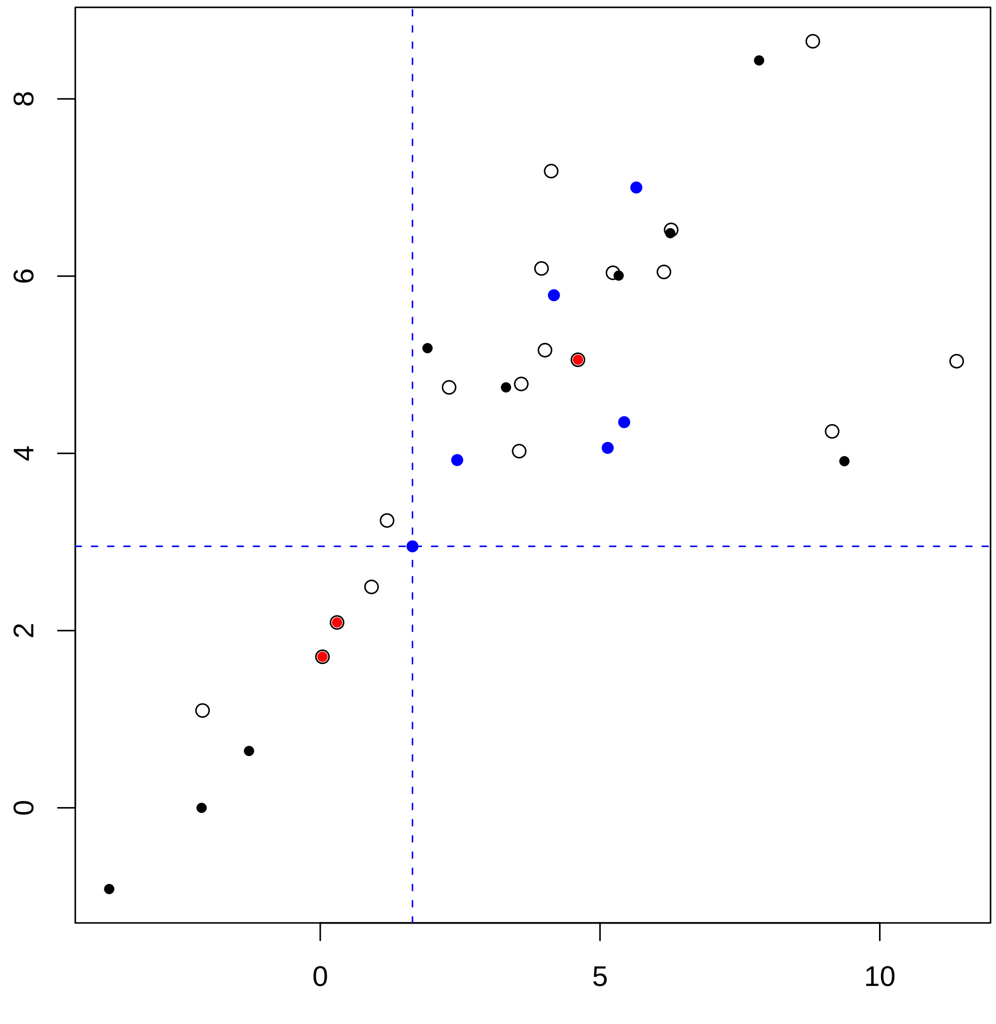
- select e.g. 3 points, randomly
- calculate measure of location (e.g. mean)
- substitute
- untill all points are microaggregated



Algorithm 1: simple

random selection:

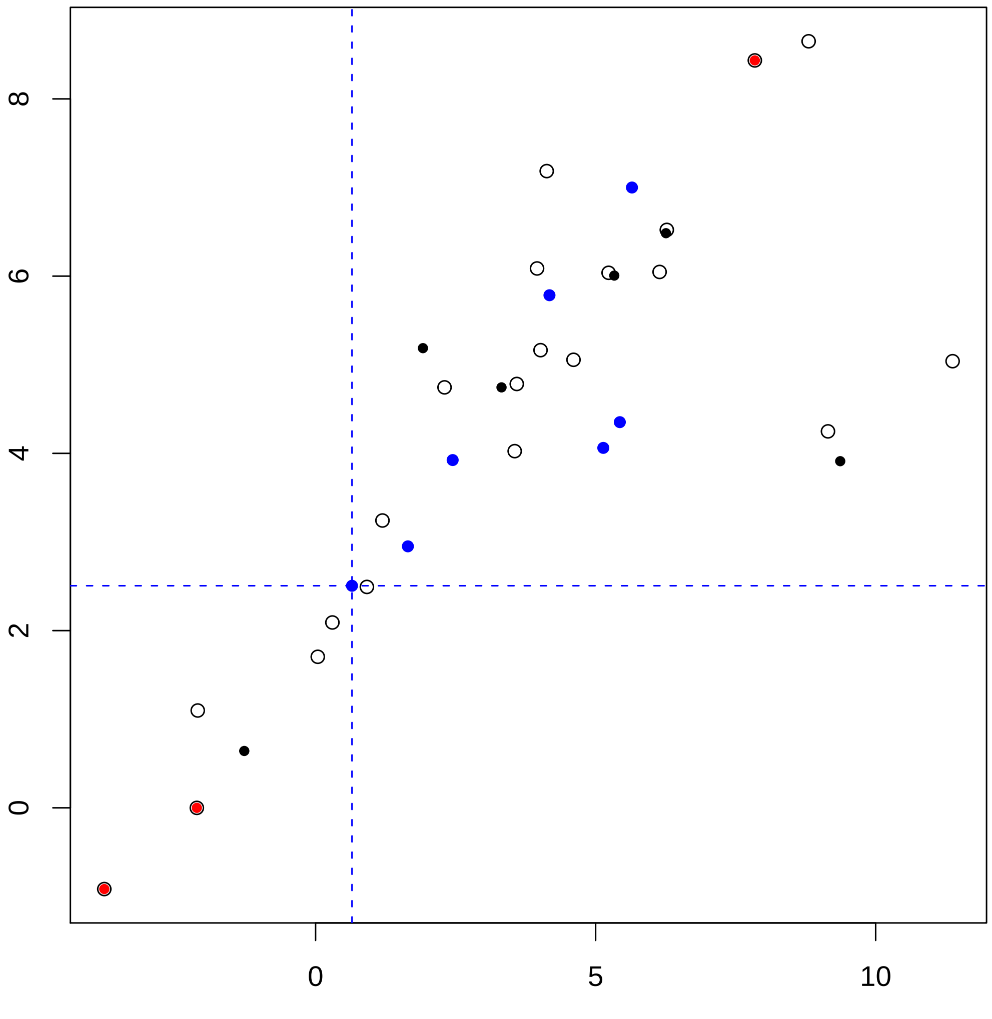
- select e.g. 3 points, randomly
- calculate measure of location (e.g. mean)
- substitute
- untill all points are microaggregated



Algorithm 1: simple

random selection:

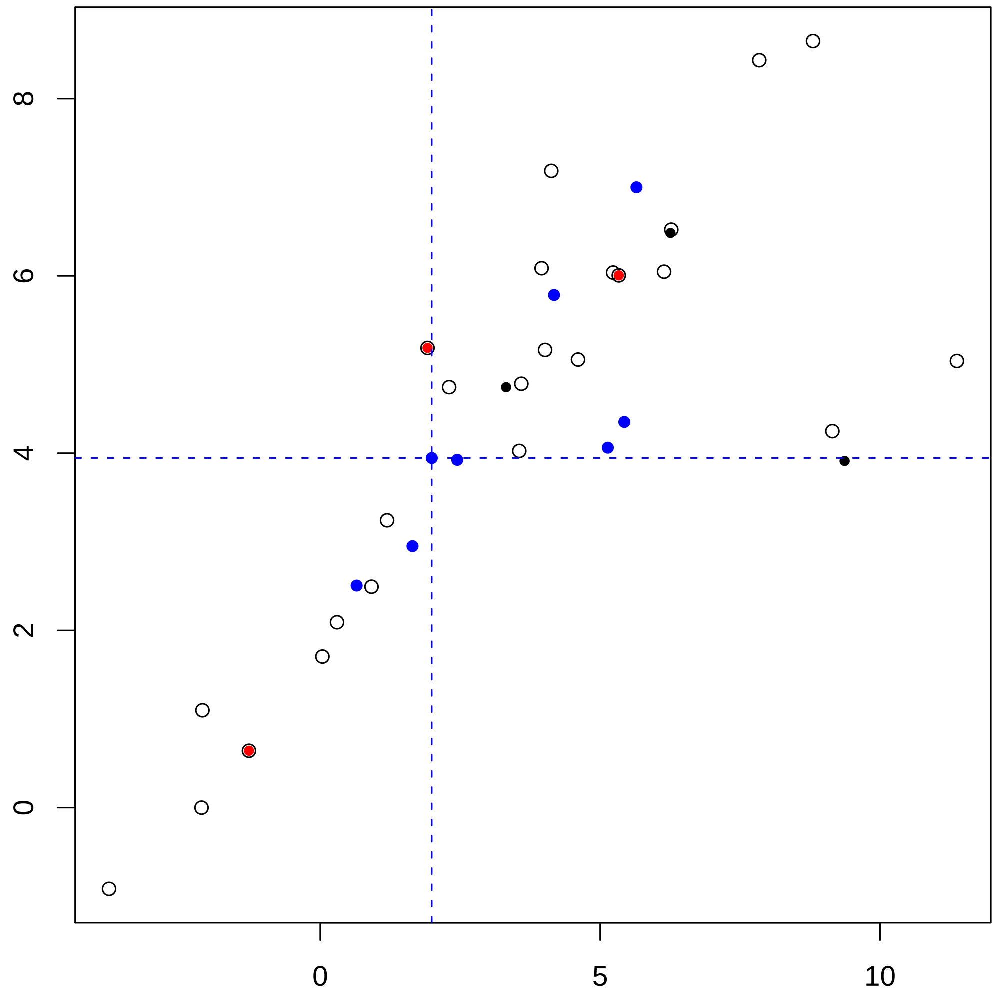
- select e.g. 3 points, randomly
- calculate measure of location (e.g. mean)
- substitute
- untill all points are microaggregated



Algorithm 1: simple

random selection:

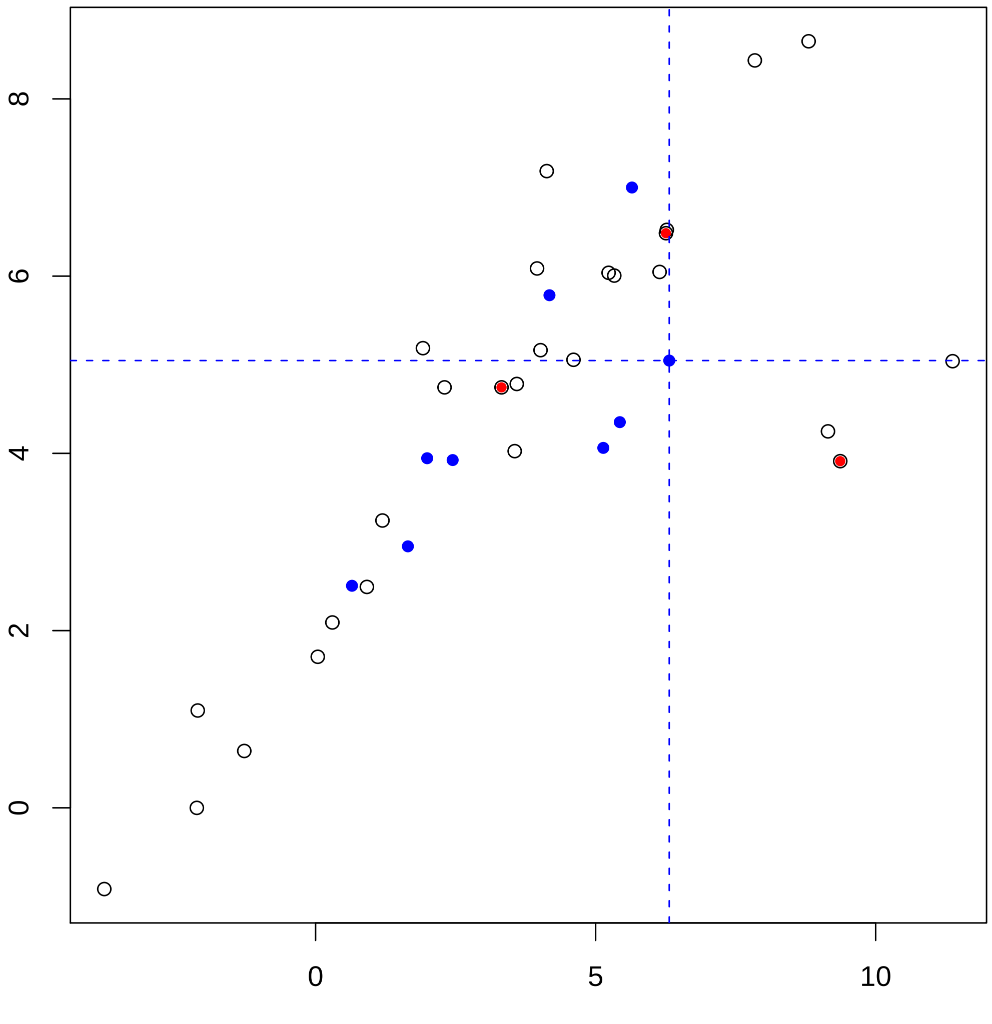
- select e.g. 3 points, randomly
- calculate measure of location (e.g. mean)
- substitute
- untill all points are microaggregated



Algorithm 1: simple

random selection:

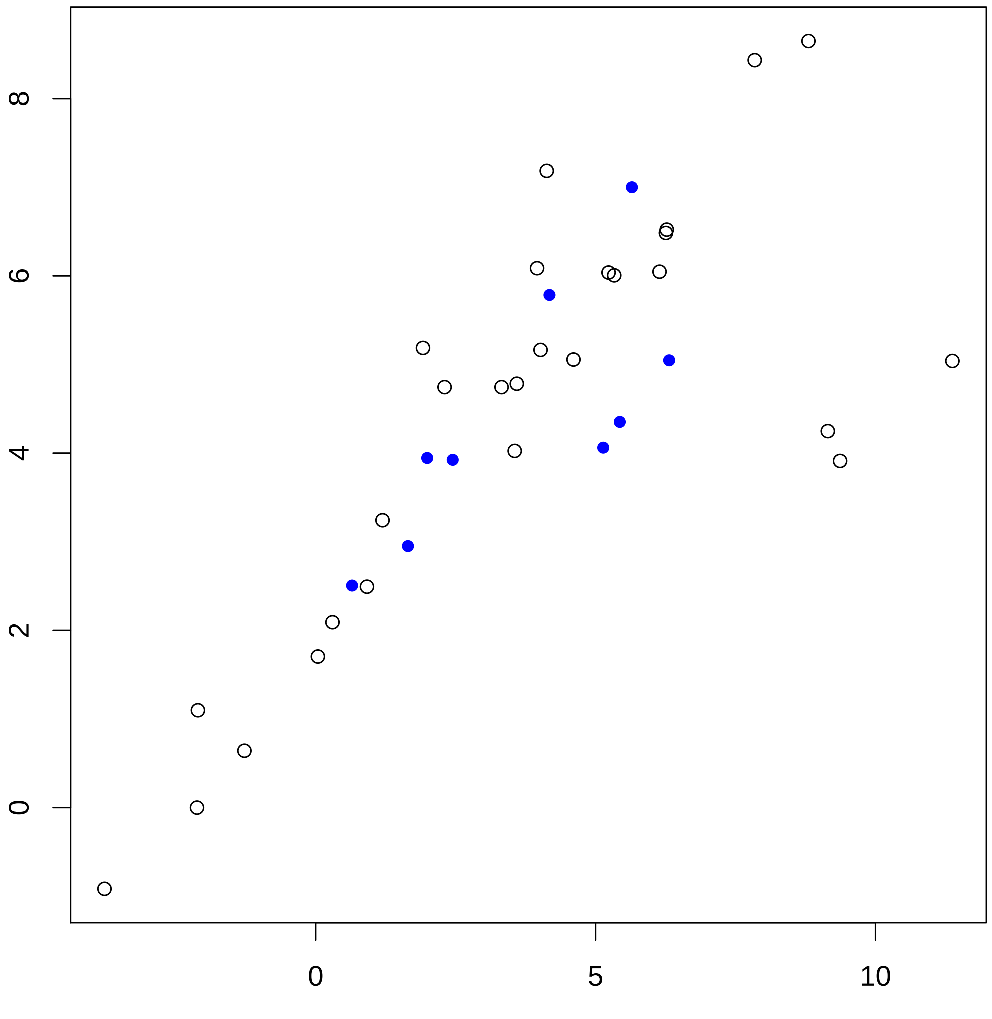
- select e.g. 3 points, randomly
- calculate measure of location (e.g. mean)
- substitute
- untill all points are microaggregated



Algorithm 1: simple

random selection:

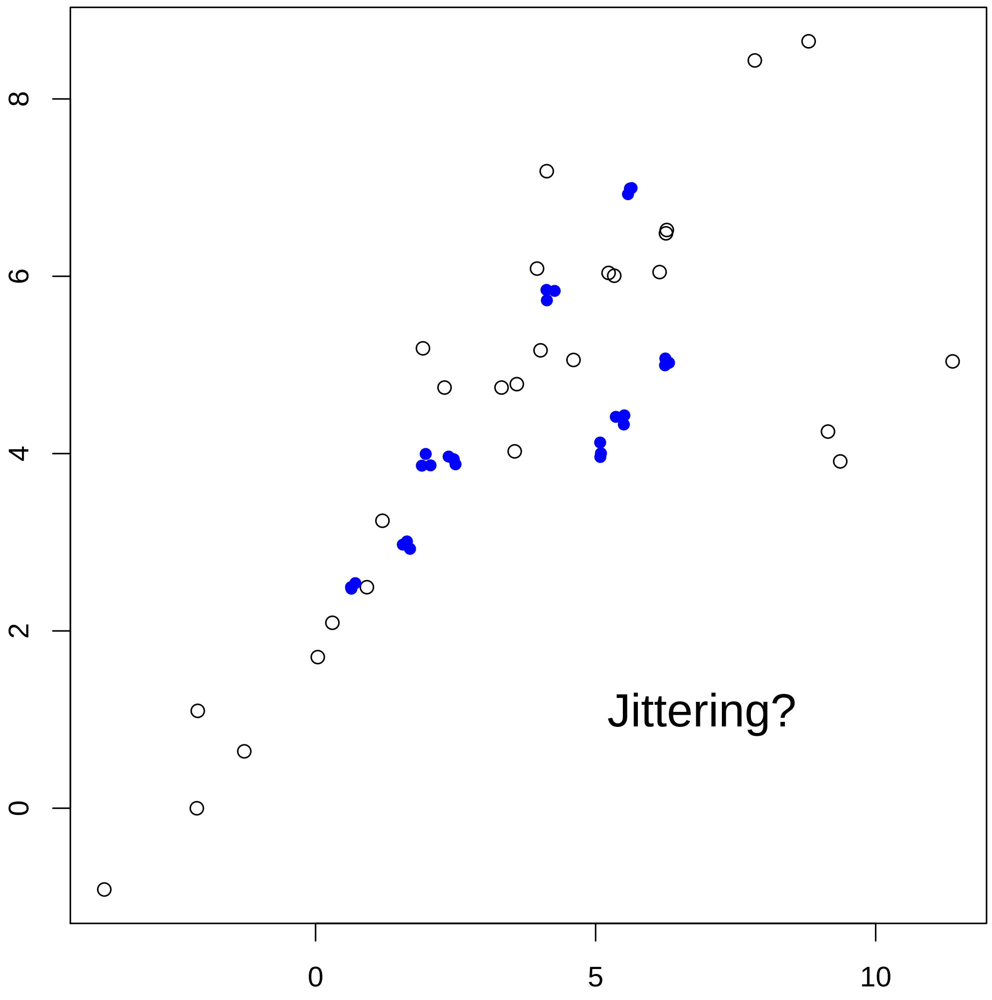
- select e.g. 3 points, randomly
- calculate measure of location (e.g. mean)
- substitute
- untill all points are microaggregated
- $\longrightarrow$ finished



Algorithm 1: simple

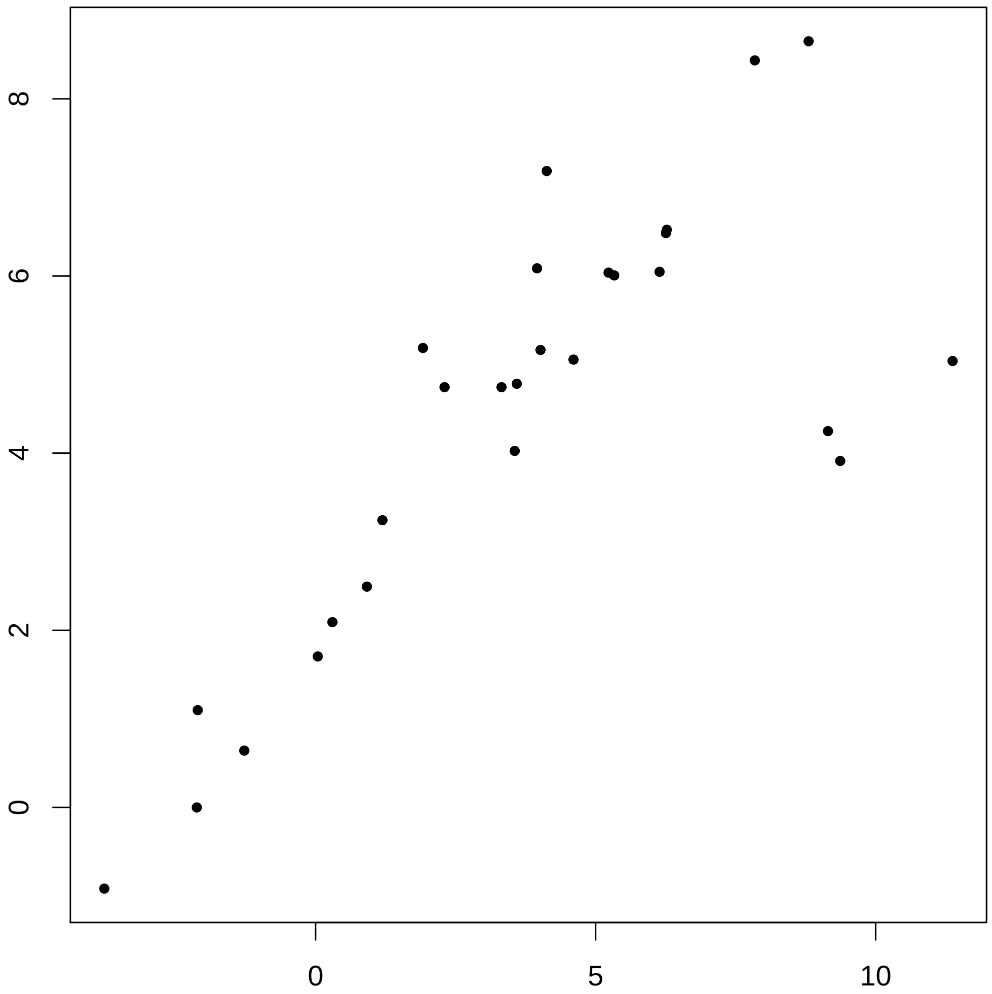
random selection:

- select e.g. 3 points, randomly
- calculate measure of location (e.g. mean)
- substitute
- untill all points are microaggregated
- $\longrightarrow$ finished



Algorithm 2: `mdav`

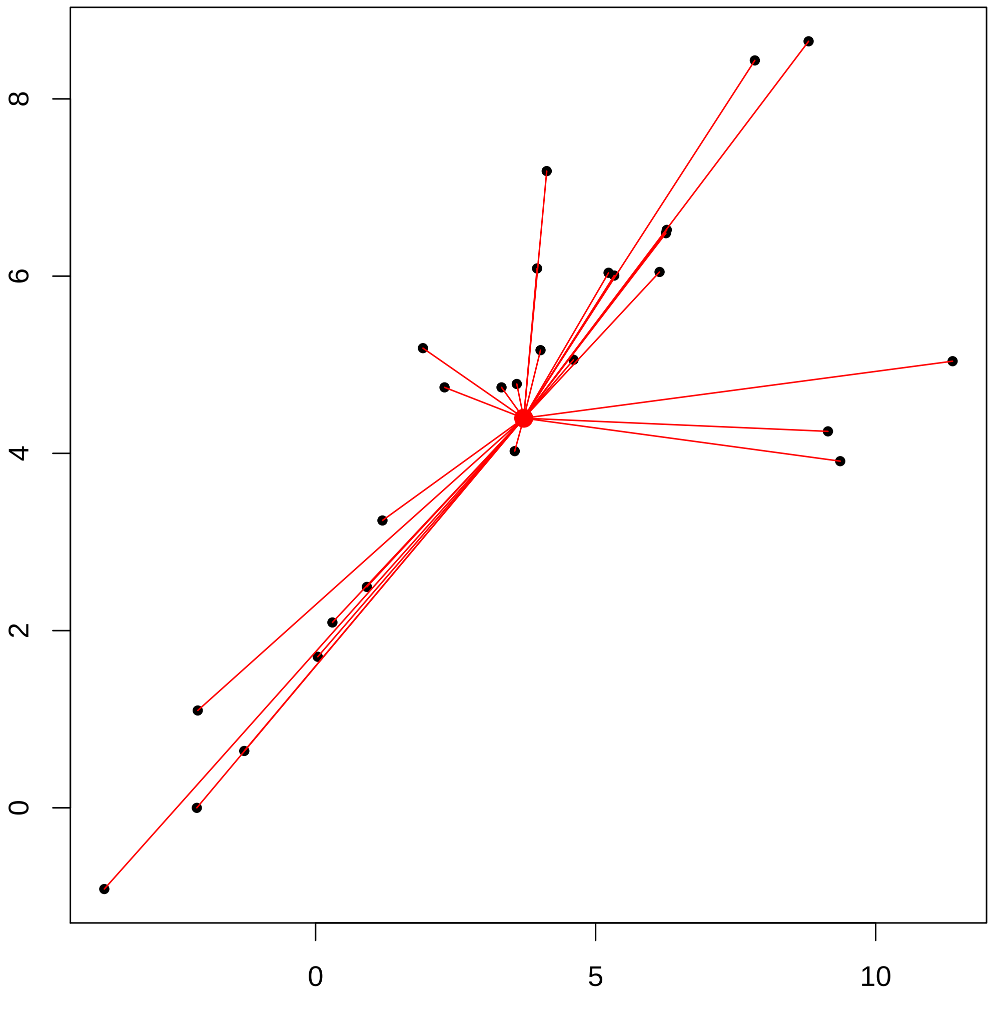
mdav, steps of the algorithm:



Algorithm 2: `mdav`

mdav, steps of the algorithm:

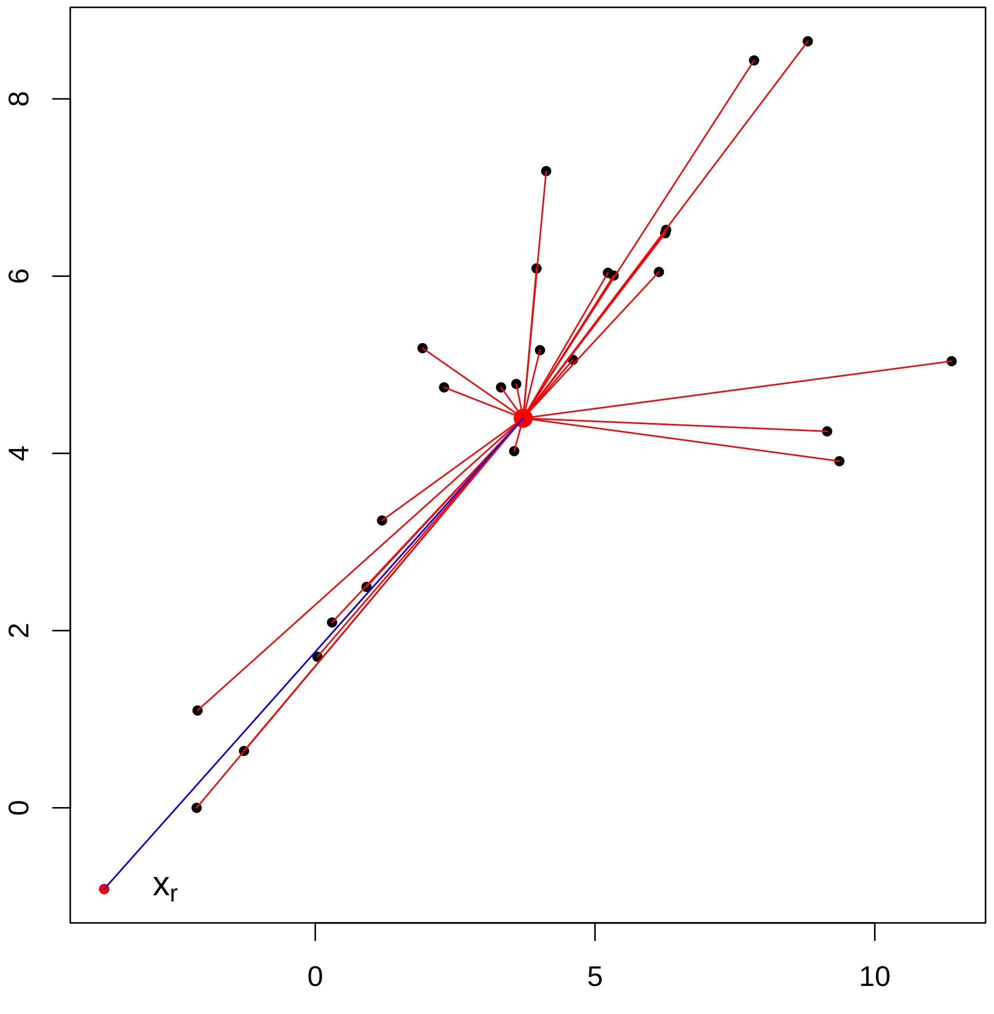
- calculate the mean



Algorithm 2: `mdav`

mdav, steps of the algorithm:

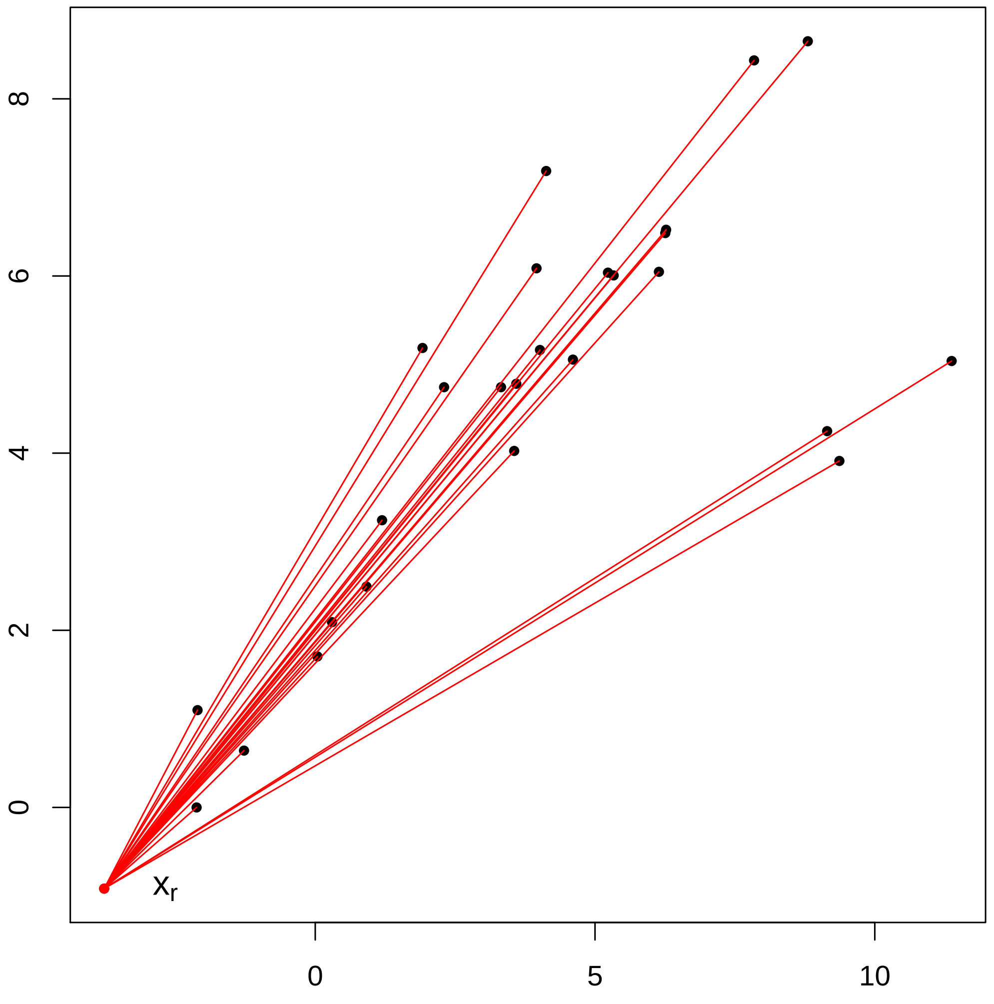
- calculate the mean
- find the most distant observation x_r



Algorithm 2: `mdav`

mdav, steps of the algorithm:

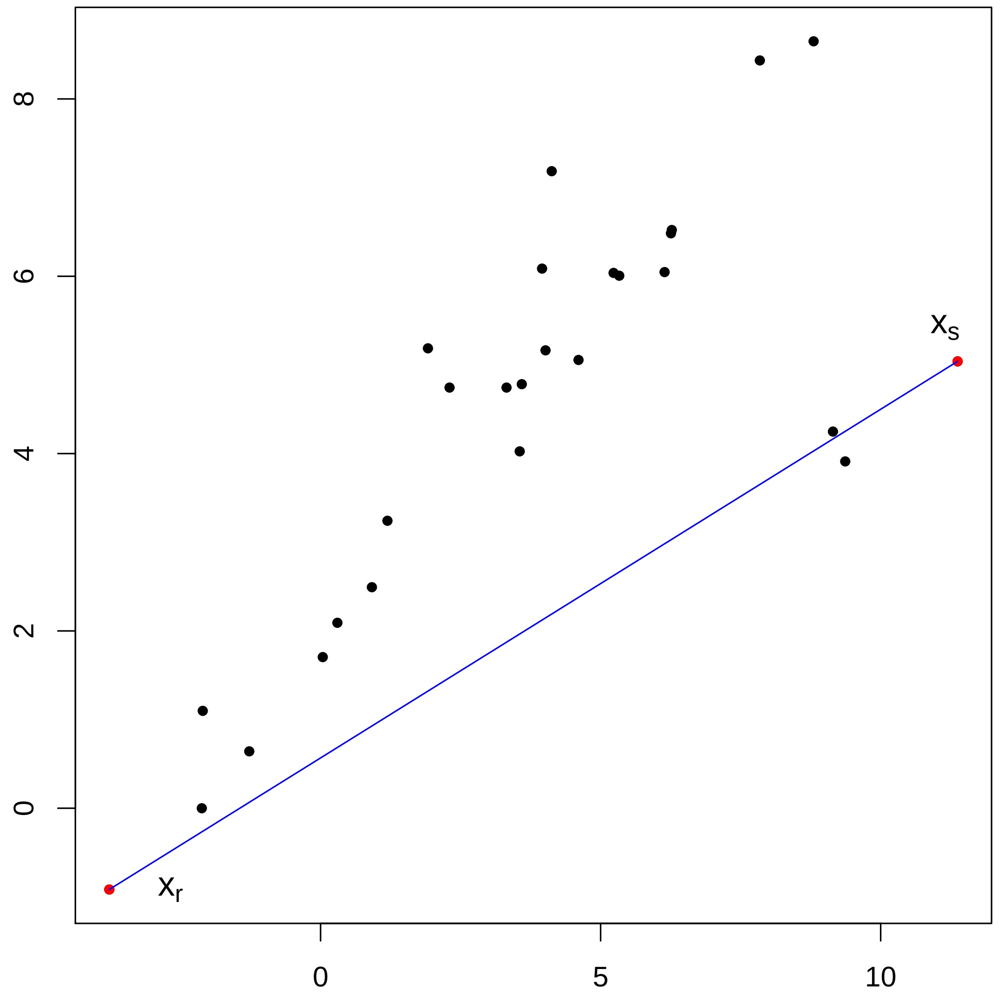
- calculate the mean
- find the most distant observation x_r
- calculate distances from all points to x_r



Algorithm 2: `mdav`

mdav, steps of the algorithm:

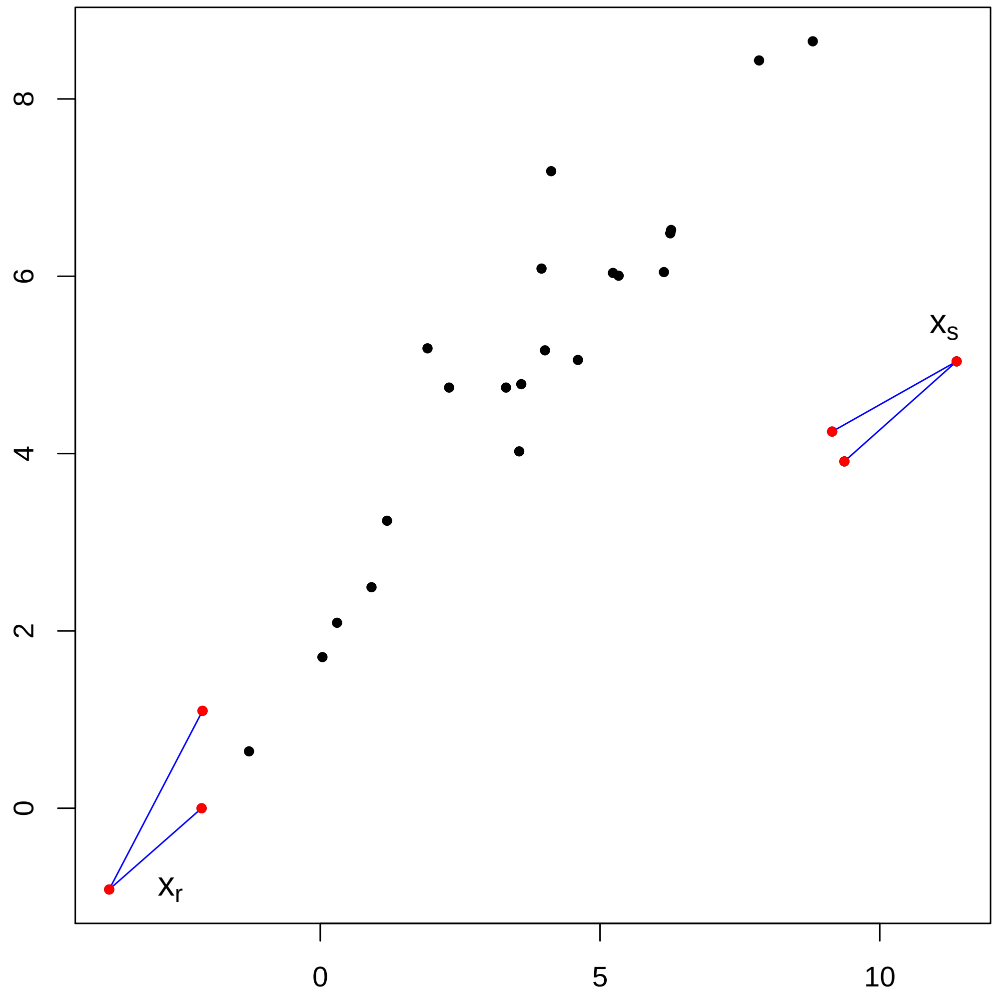
- calculate the mean
- find the most distant observation x_r
- calculate distances from all points to x_r
- choose the most distant observation x_s from x_r



Algorithm 2: `mdav`

mdav, steps of the algorithm:

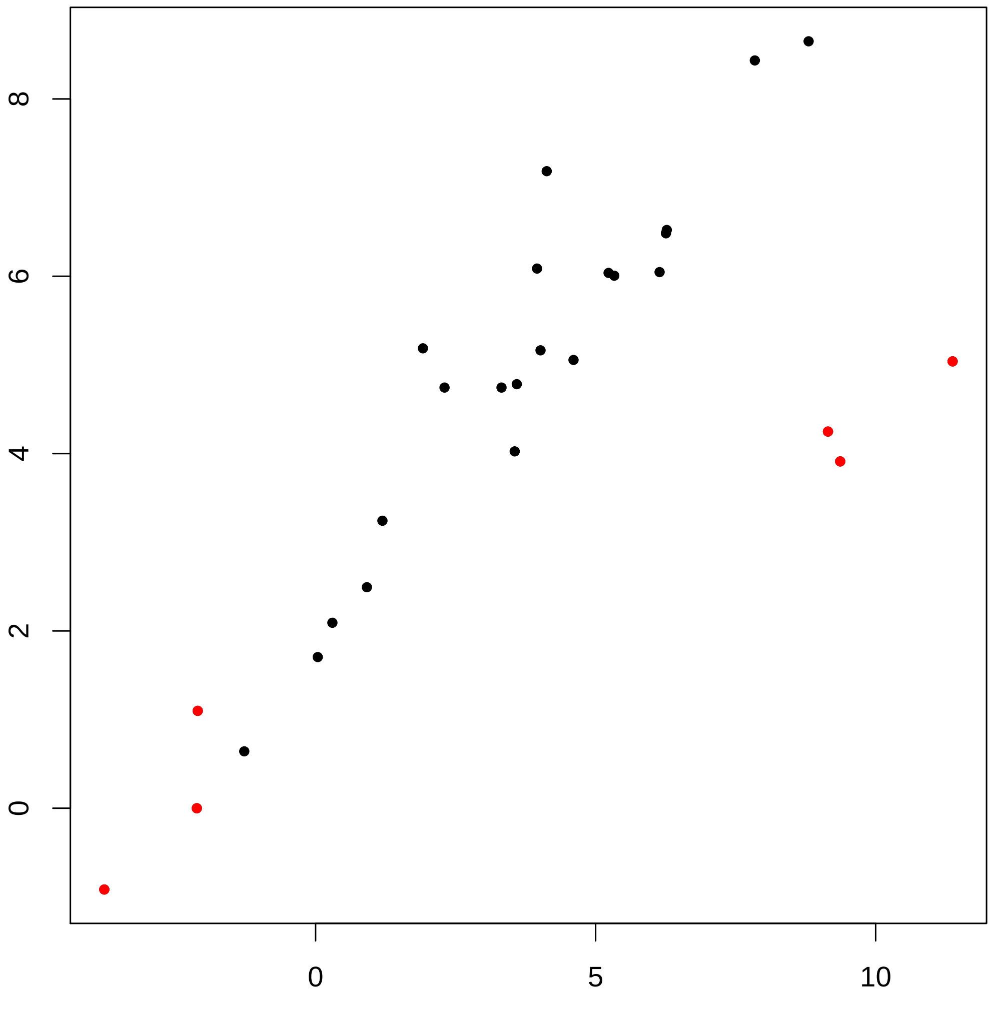
- calculate the mean
- find the most distant observation x_r
- calculate distances from all points to x_r
- choose the most distant observation x_s from x_r
- find the 2 nearest neighbors of x_r and x_s



Algorithm 2: `mdav`

mdav, steps of the algorithm:

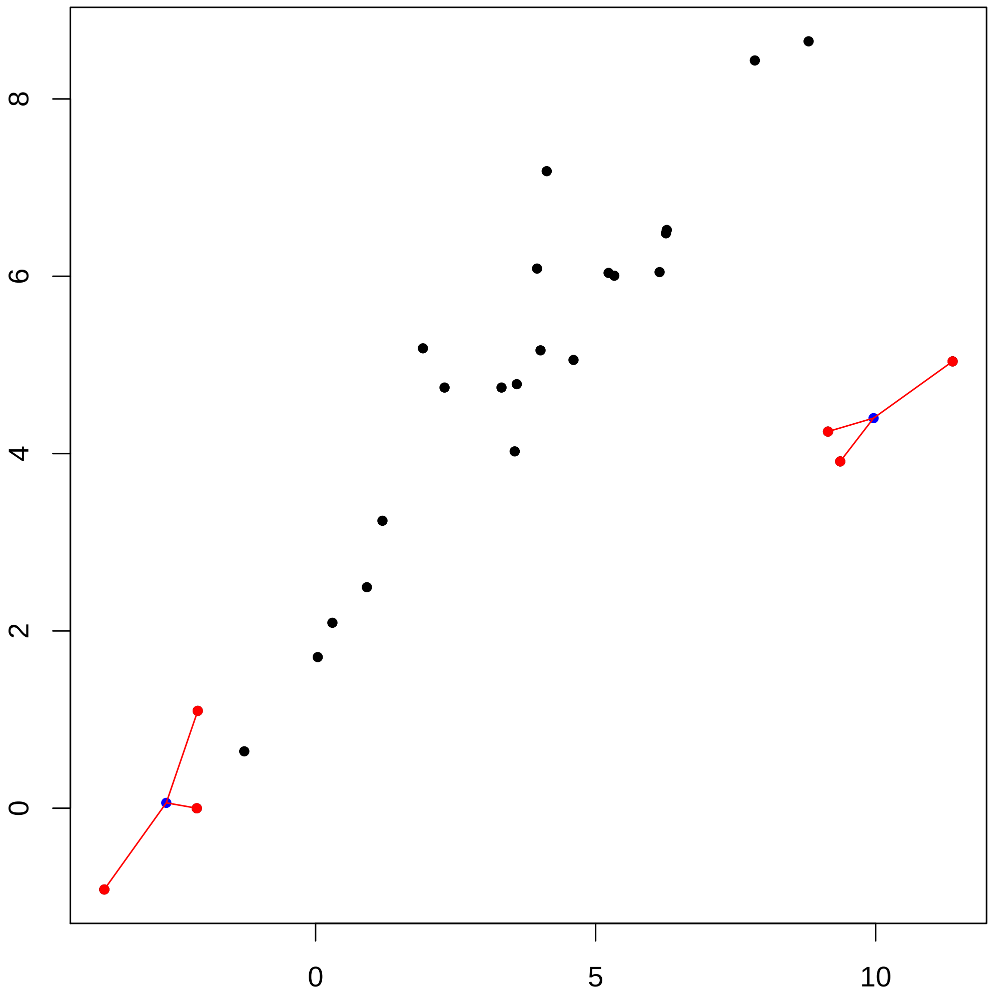
- calculate the mean
- find the most distant observation x_r
- calculate distances from all points to x_r
- choose the most distant observation x_s from x_r
- find the 2 nearest neighbors of x_r and x_s



Algorithm 2: `mdav`

mdav, steps of the algorithm:

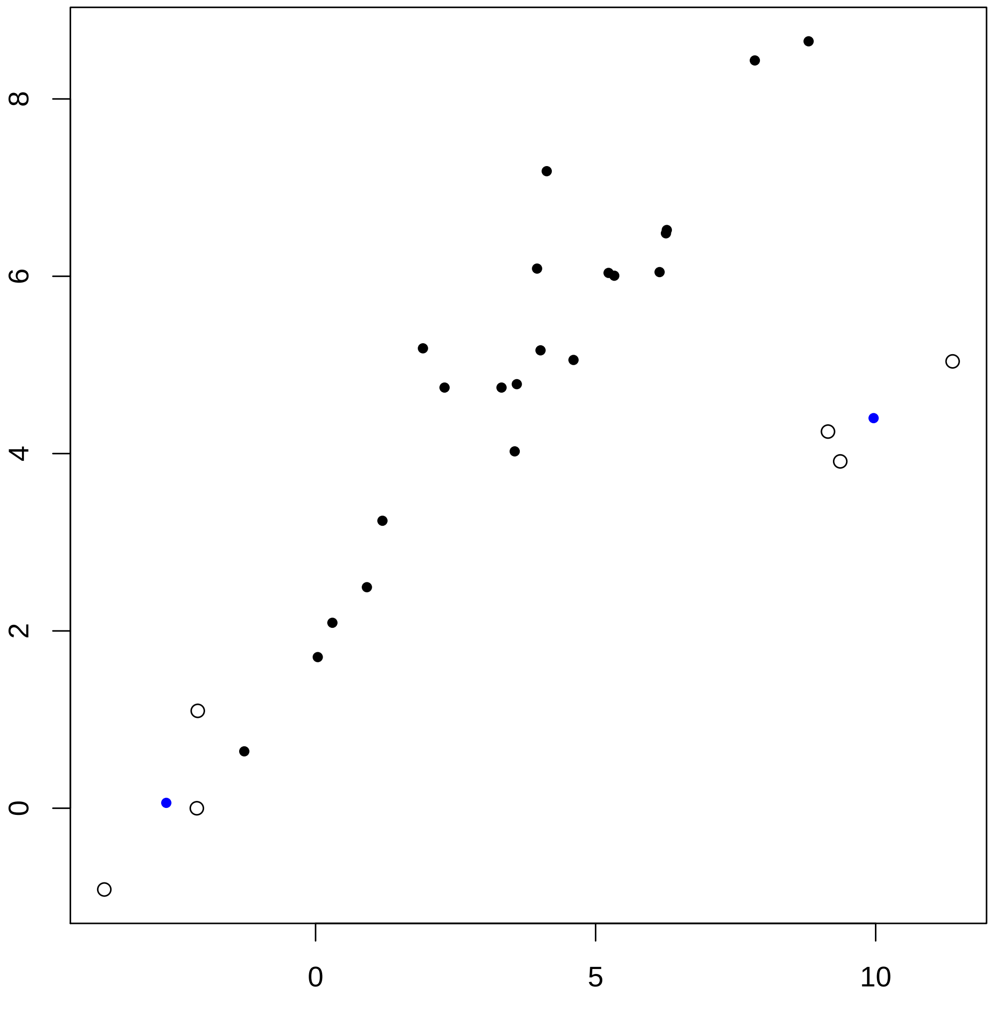
- calculate the mean
- find the most distant observation x_r
- calculate distances from all points to x_r
- choose the most distant observation x_s from x_r
- find the 2 nearest neighbors of x_r and x_s
- aggregate them with the mean



Algorithm 2: `mdav`

mdav, steps of the algorithm:

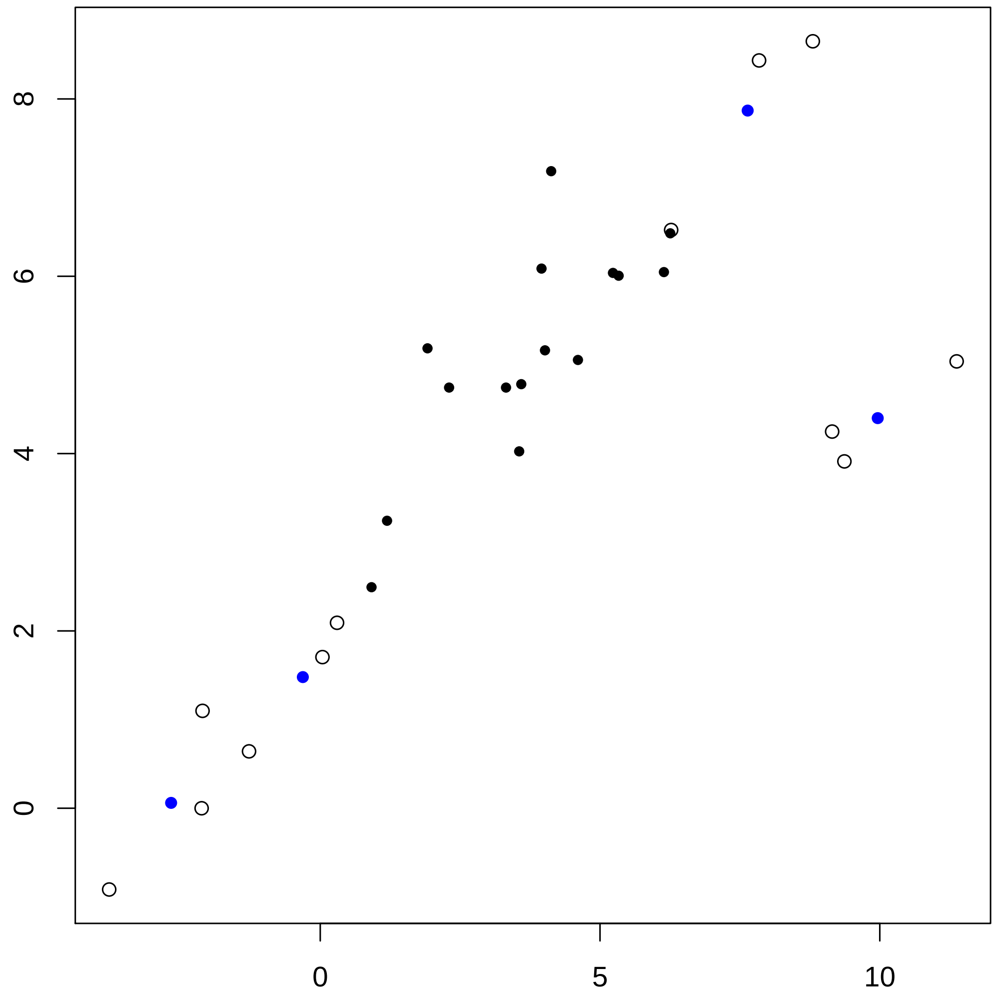
- calculate the mean
- find the most distant observation x_r
- calculate distances from all points to x_r
- choose the most distant observation x_s from x_r
- find the 2 nearest neighbors of x_r and x_s
- aggregate them with the mean



Algorithm 2: `mdav`

mdav, steps of the algorithm:

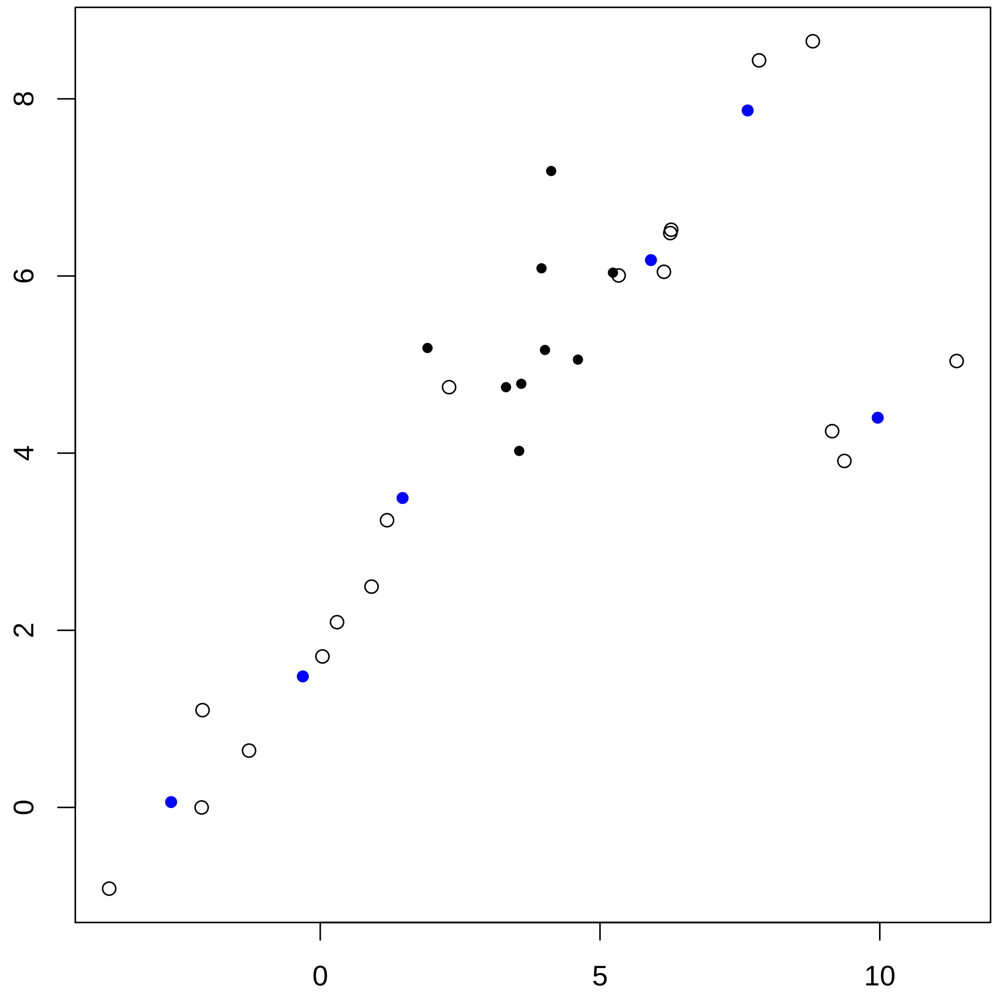
- calculate the mean
- find the most distant observation x_r
- calculate distances from all points to x_r
- choose the most distant observation x_s from x_r
- find the 2 nearest neighbors of x_r and x_s
- aggregate them with the mean
- continue until all observations are aggregated (spezial rules at the end)



Algorithm 2: `mdav`

mdav, steps of the algorithm:

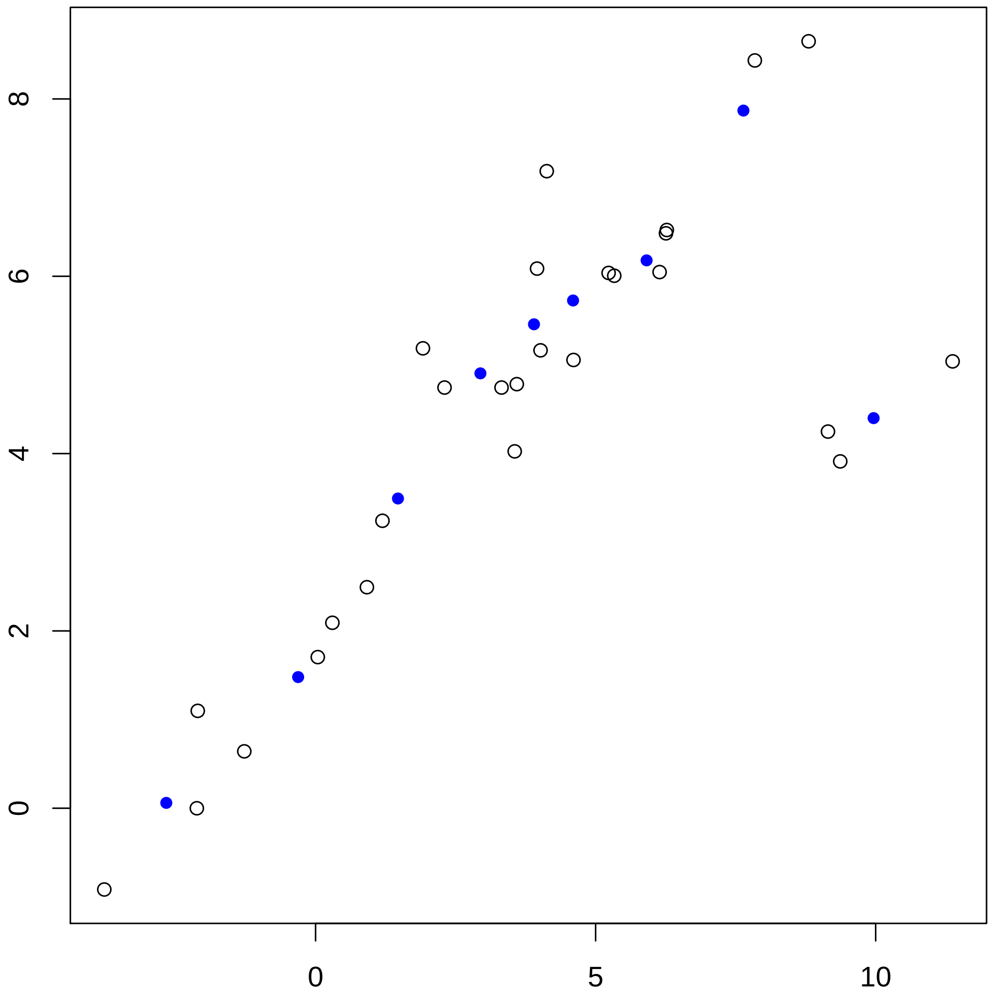
- calculate the mean
- find the most distant observation x_r
- calculate distances from all points to x_r
- choose the most distant observation x_s from x_r
- find the 2 nearest neighbors of x_r and x_s
- aggregate them with the mean
- continue until all observations are aggregated (spezial rules at the end)



Algorithm 2: `mdav`

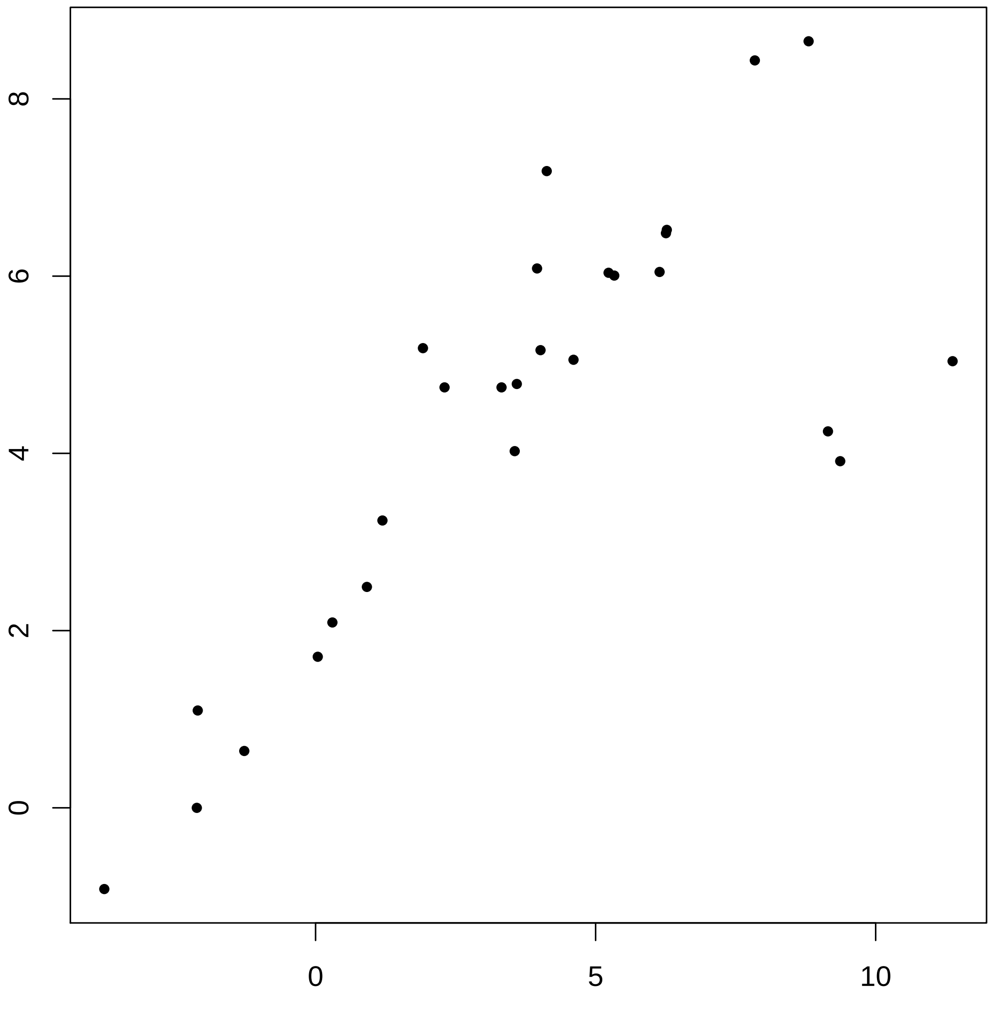
mdav, steps of the algorithm:

- calculate the mean
- find the most distant observation x_r
- calculate distances from all points to x_r
- choose the most distant observation x_s from x_r
- find the 2 nearest neighbors of x_r and x_s
- aggregate them with the mean
- continue until all observations are aggregated (spezial rules at the end)



Projection Methods: PCA

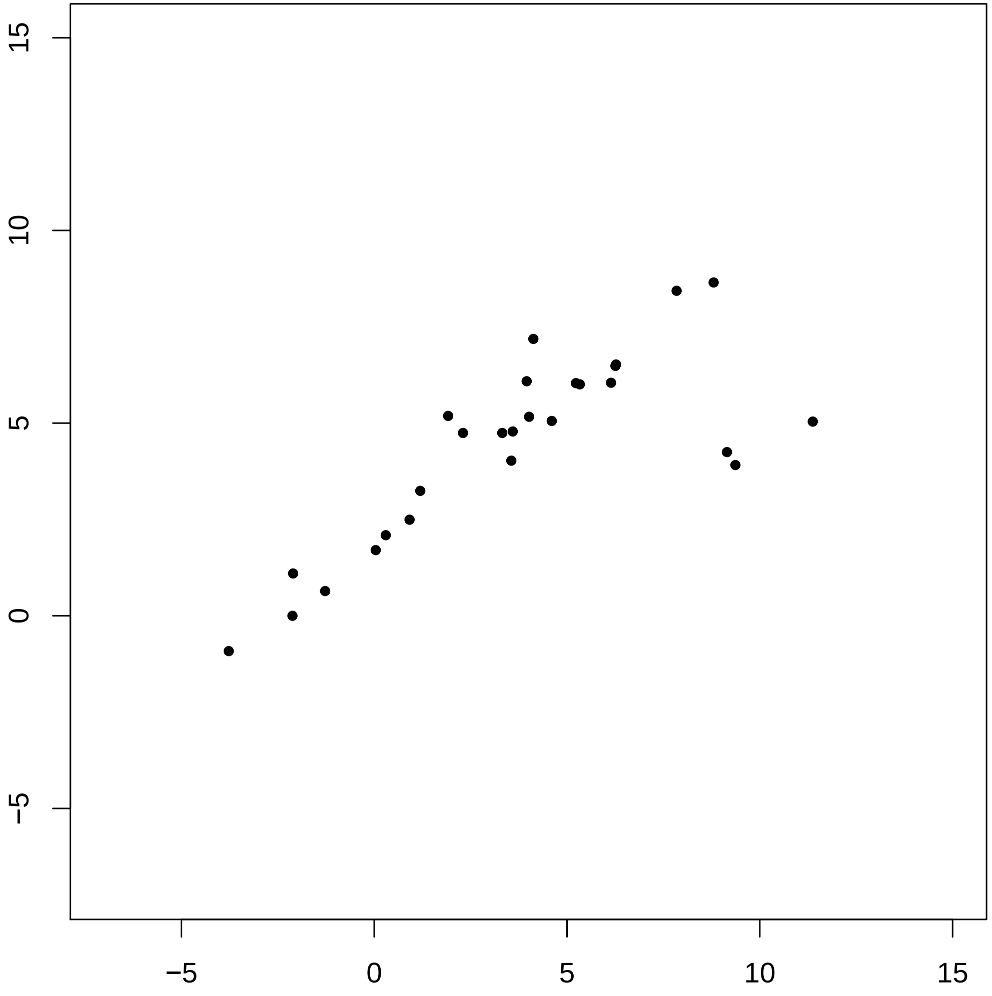
pca method, steps:



Projection Methods: PCA

pca method, steps:

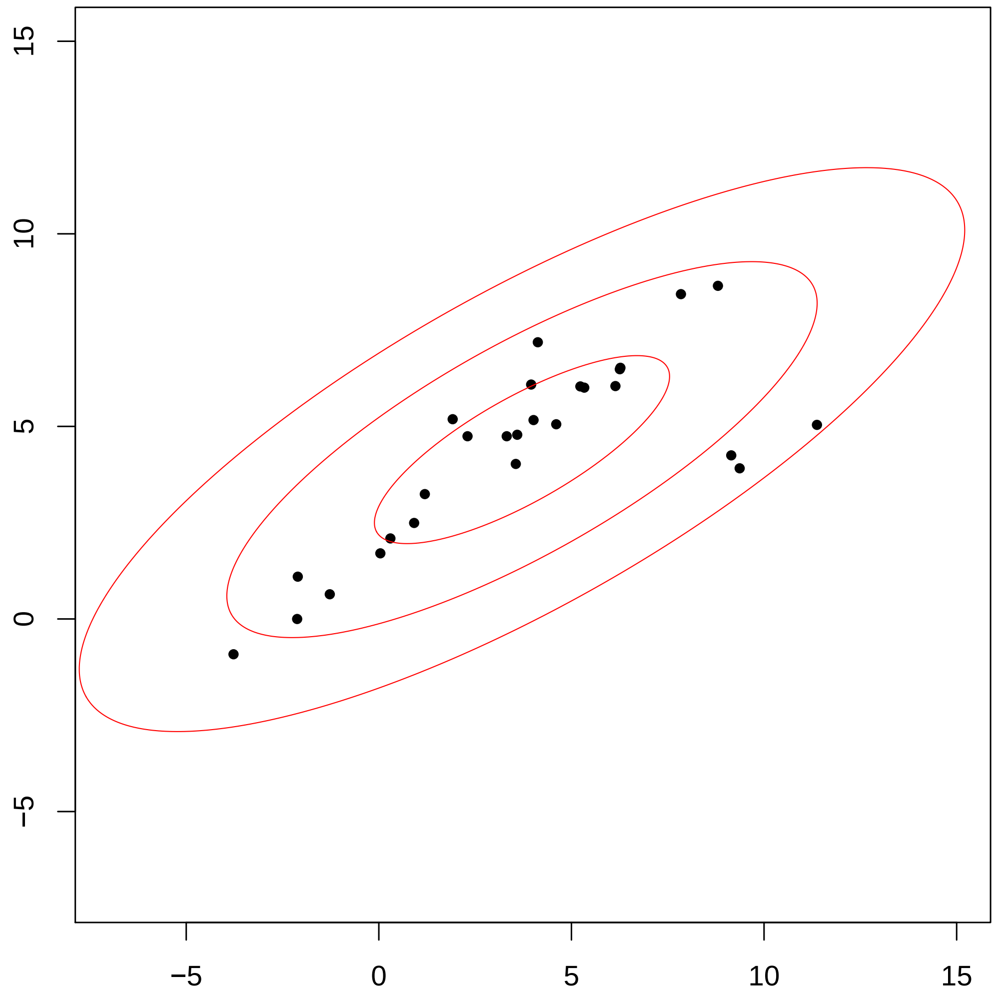
- find first PC



Projection Methods: PCA

pca method, steps:

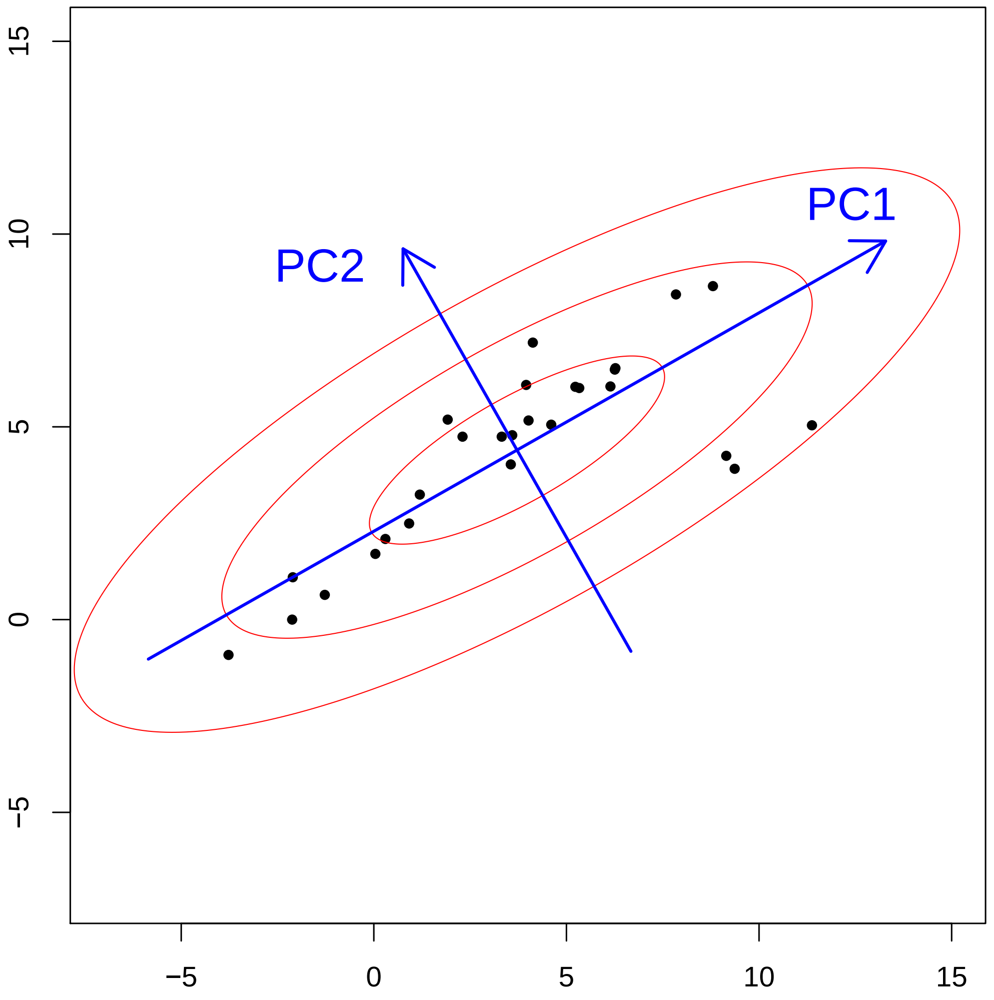
- find first PC



Projection Methods: PCA

pca method, steps:

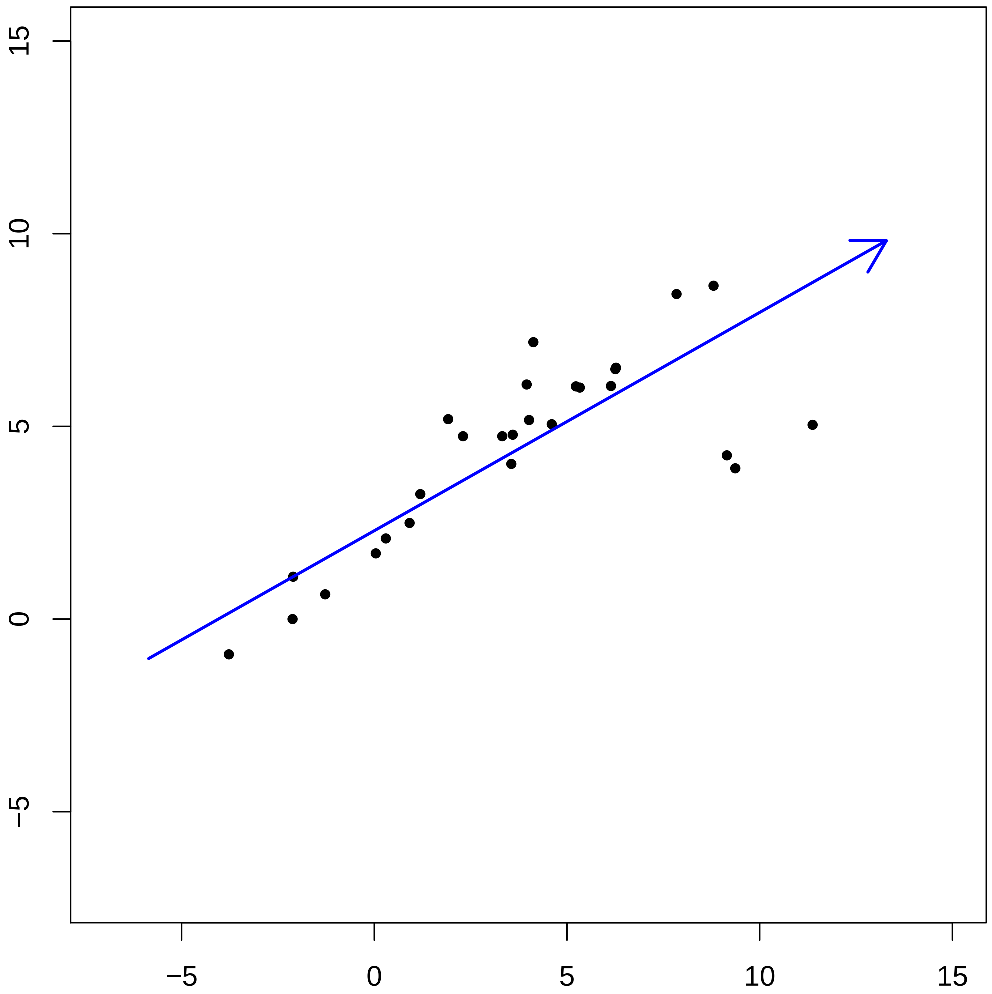
- find first PC



Projection Methods: PCA

pca method, steps:

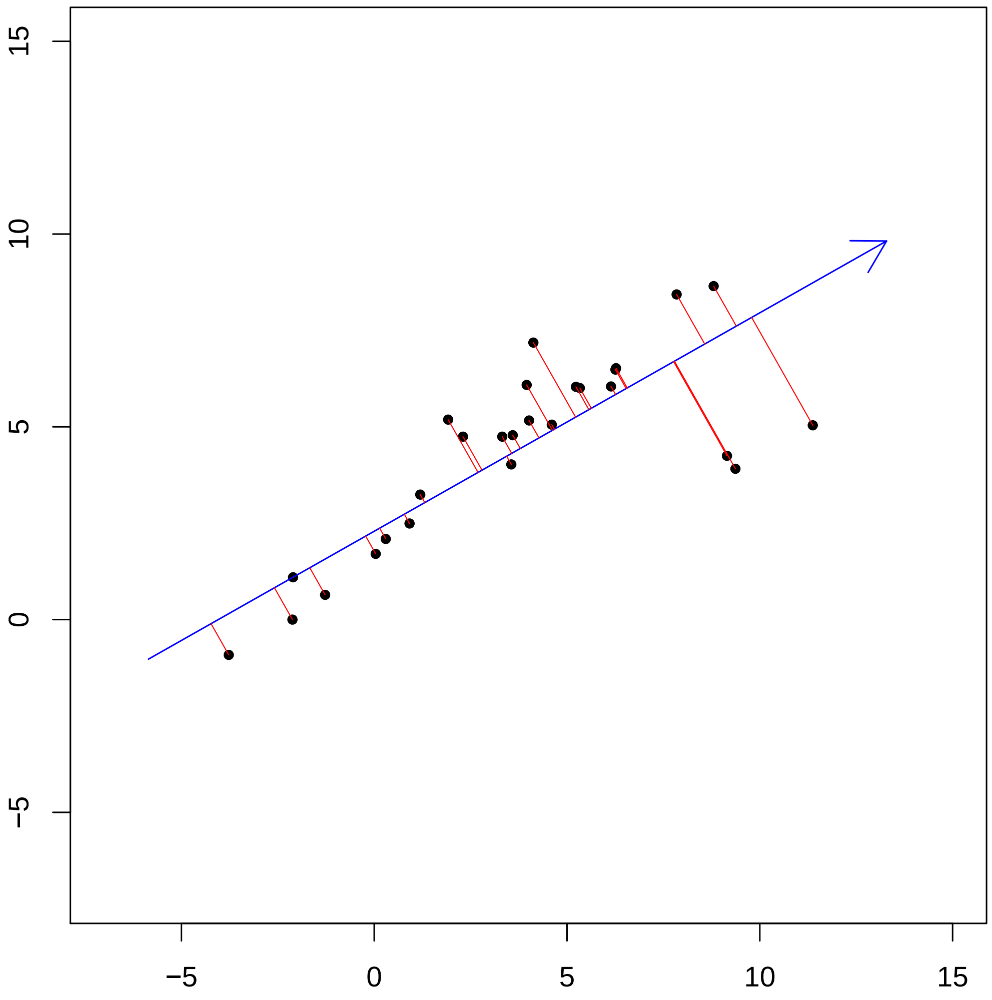
- find first PC



Projection Methods: PCA

pca method, steps:

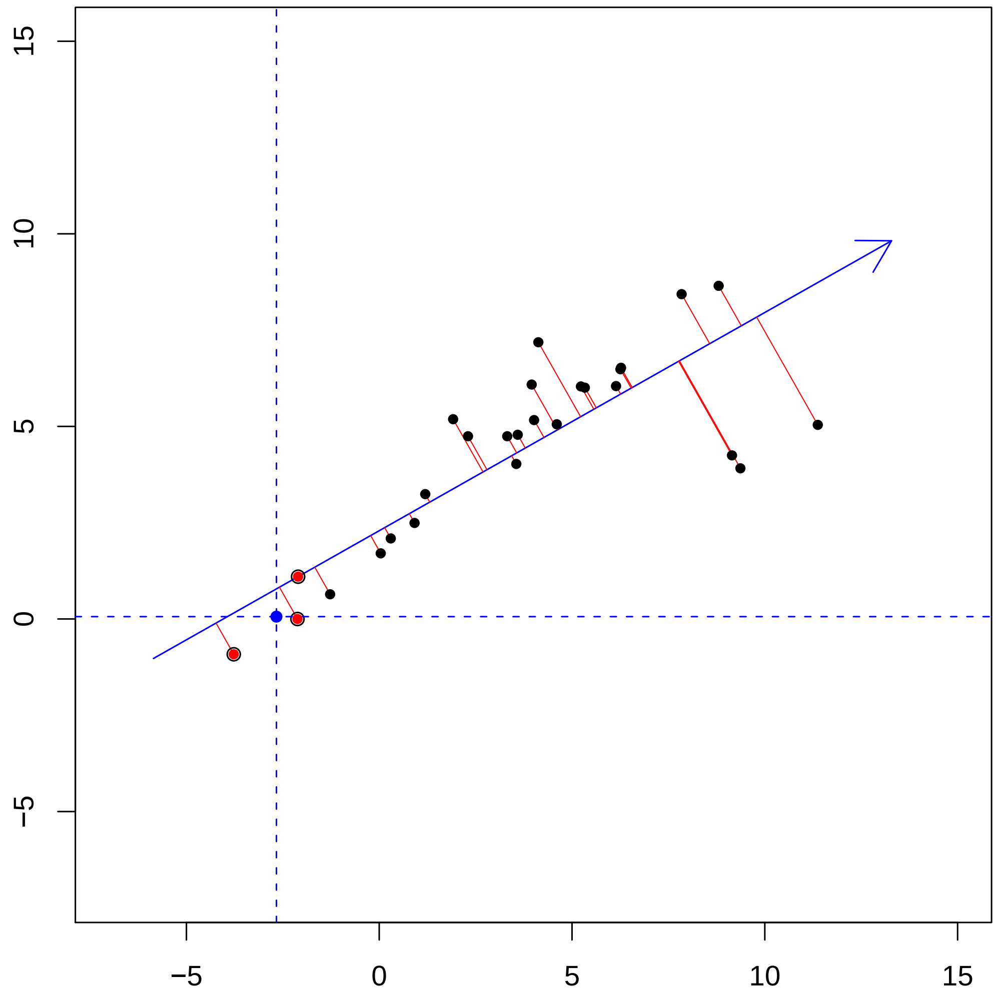
- find first PC
- project data on first PC



Projection Methods: PCA

pca method, steps:

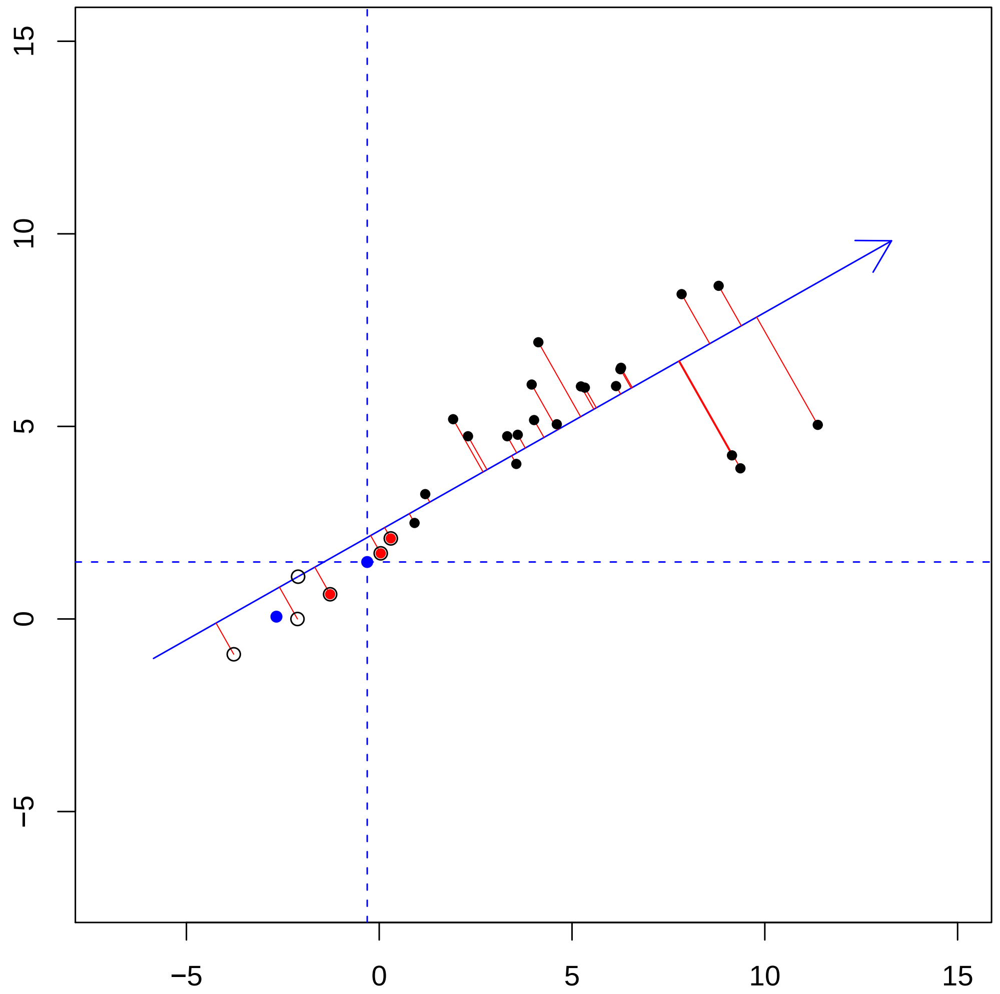
- find first PC
- project data on first PC
- aggregate sorted projected data, e.g with the mean



Projection Methods: PCA

pca method, steps:

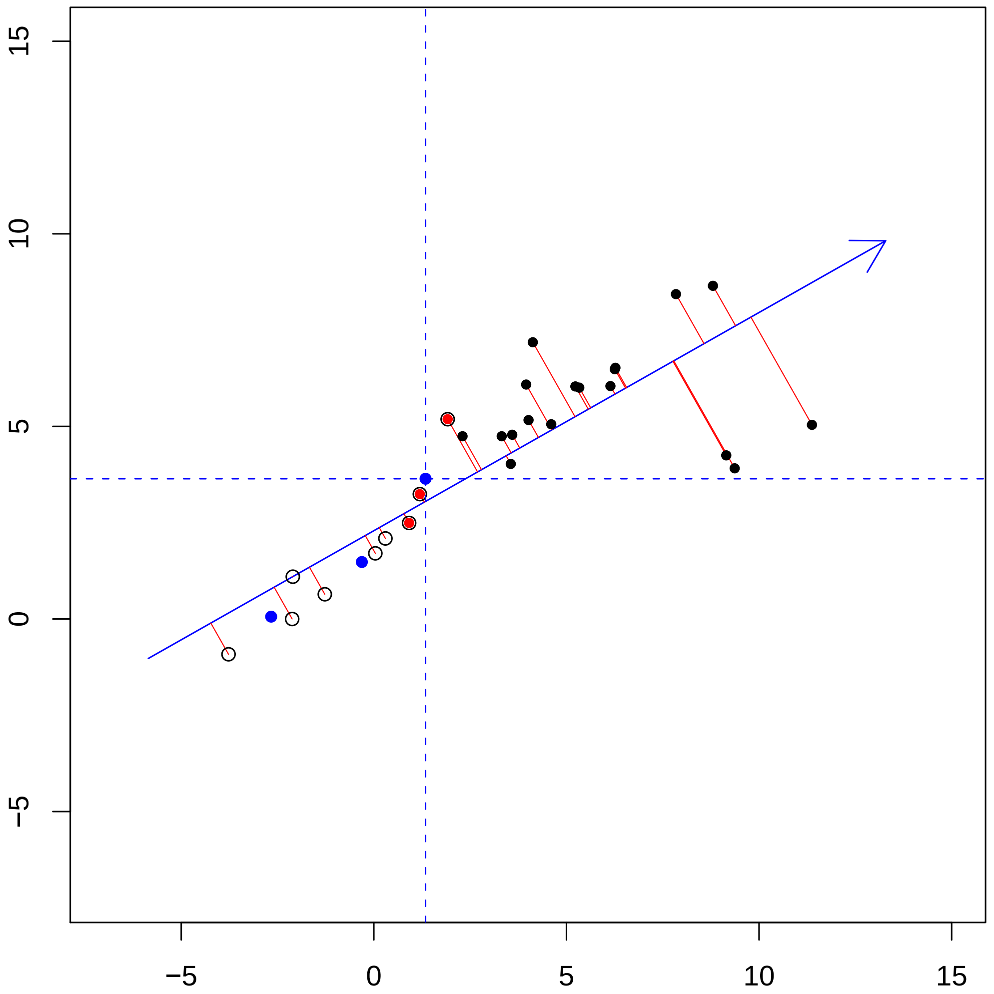
- find first PC
- project data on first PC
- aggregate sorted projected data, e.g with the mean



Projection Methods: PCA

pca method, steps:

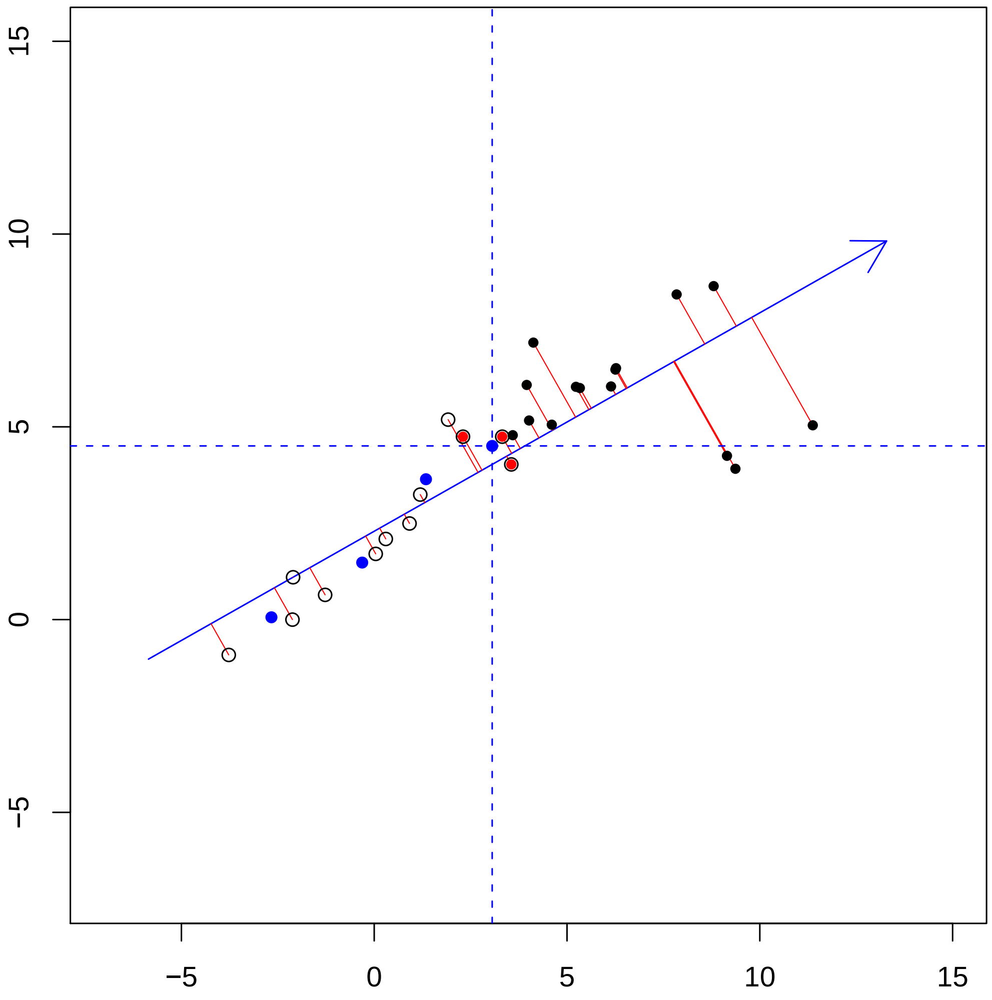
- find first PC
- project data on first PC
- aggregate sorted projected data, e.g with the mean



Projection Methods: PCA

pca method, steps:

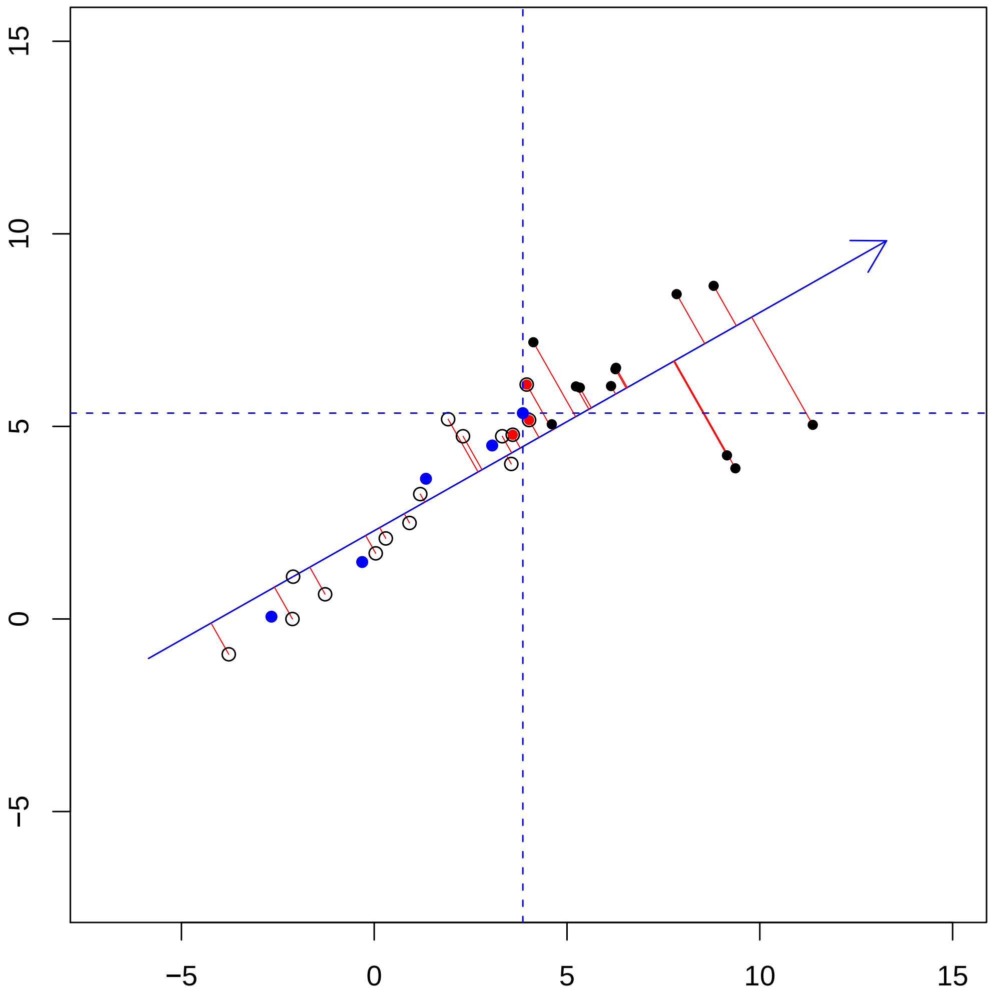
- find first PC
- project data on first PC
- aggregate sorted projected data, e.g with the mean



Projection Methods: PCA

pca method, steps:

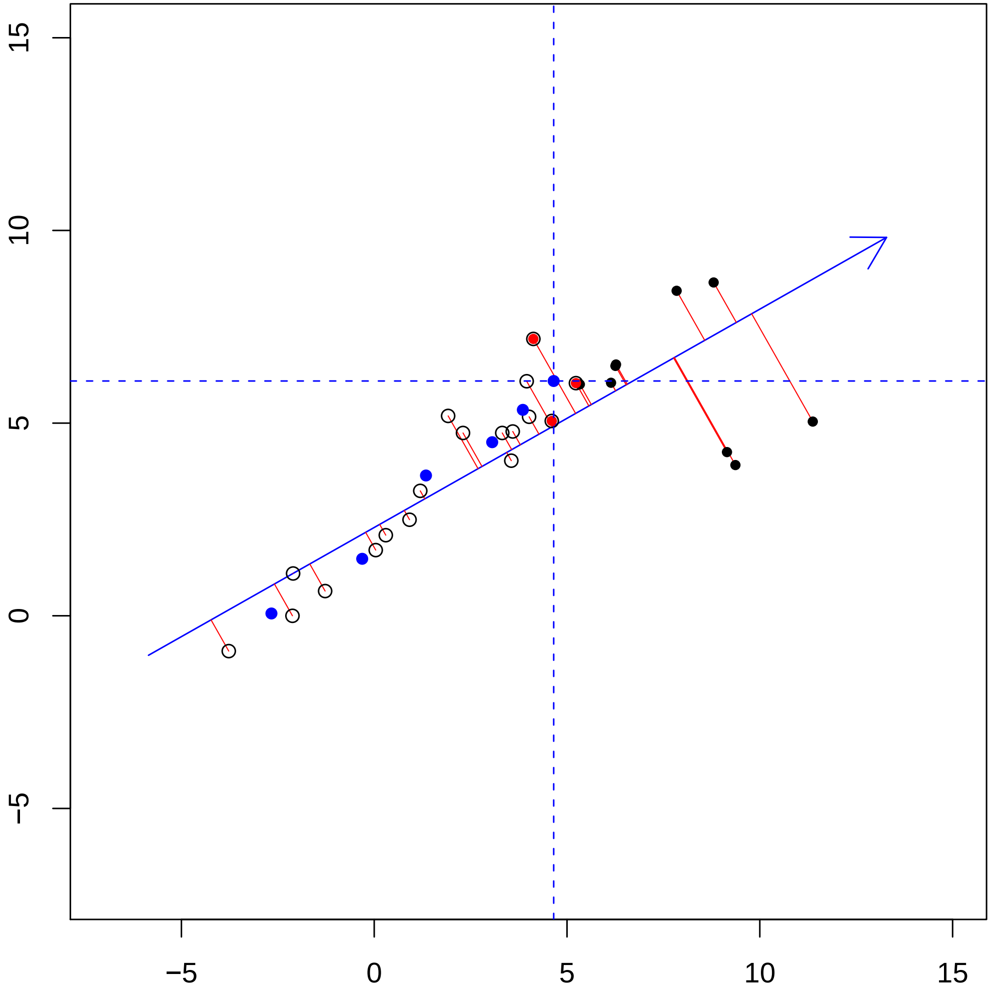
- find first PC
- project data on first PC
- aggregate sorted projected data, e.g with the mean



Projection Methods: PCA

pca method, steps:

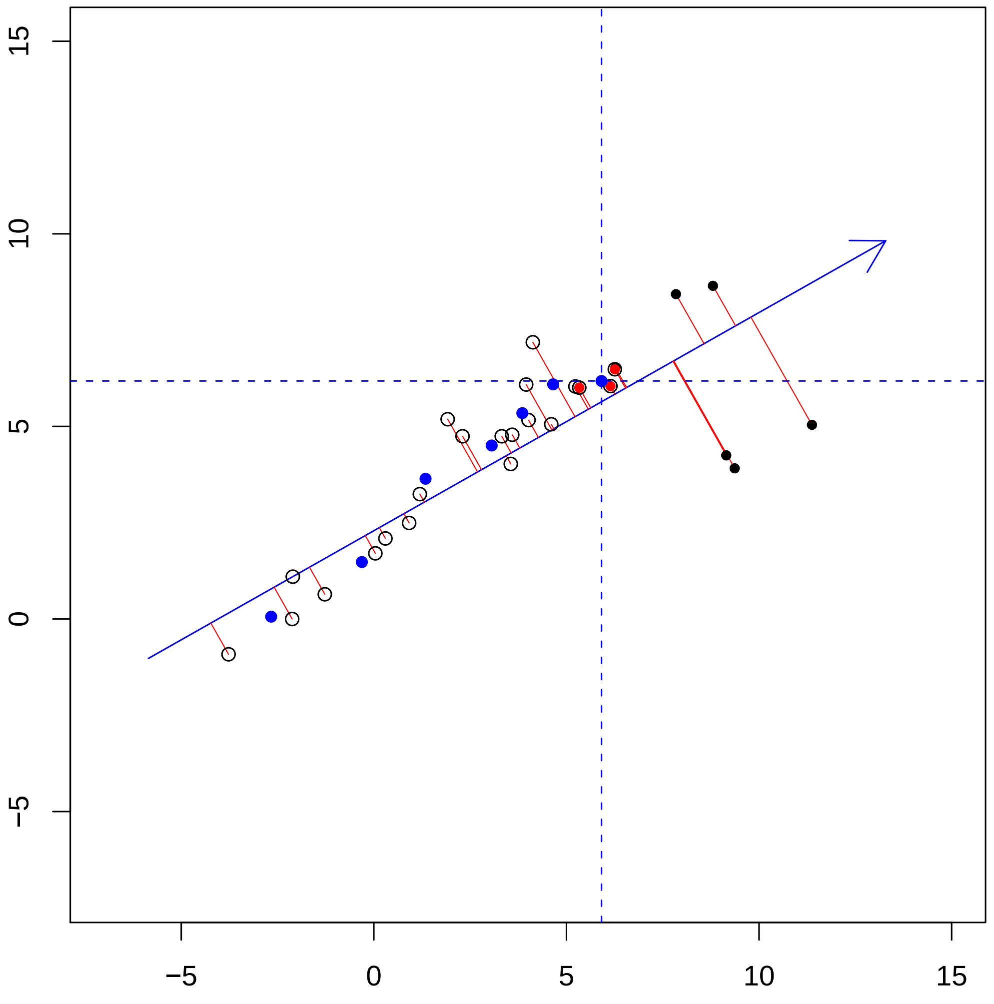
- find first PC
- project data on first PC
- aggregate sorted projected data, e.g with the mean



Projection Methods: PCA

pca method, steps:

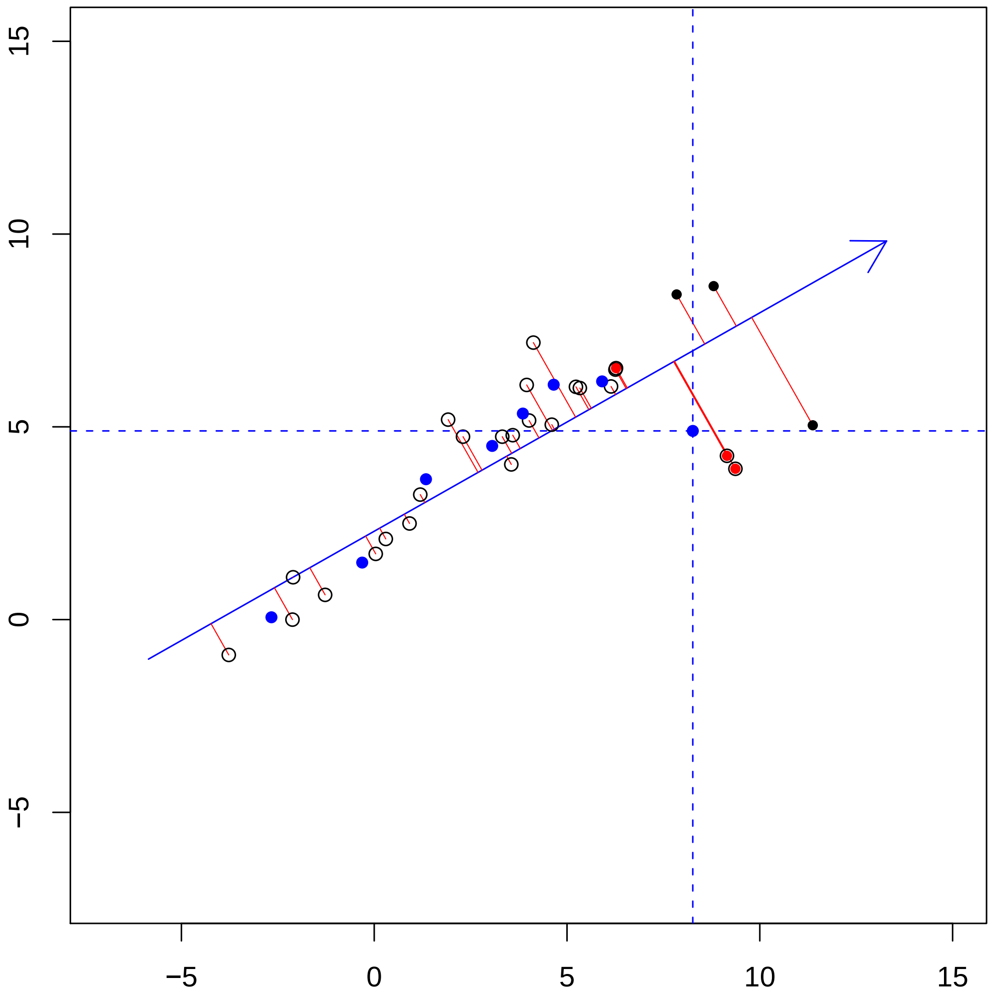
- find first PC
- project data on first PC
- aggregate sorted projected data, e.g with the mean



Projection Methods: PCA

pca method, steps:

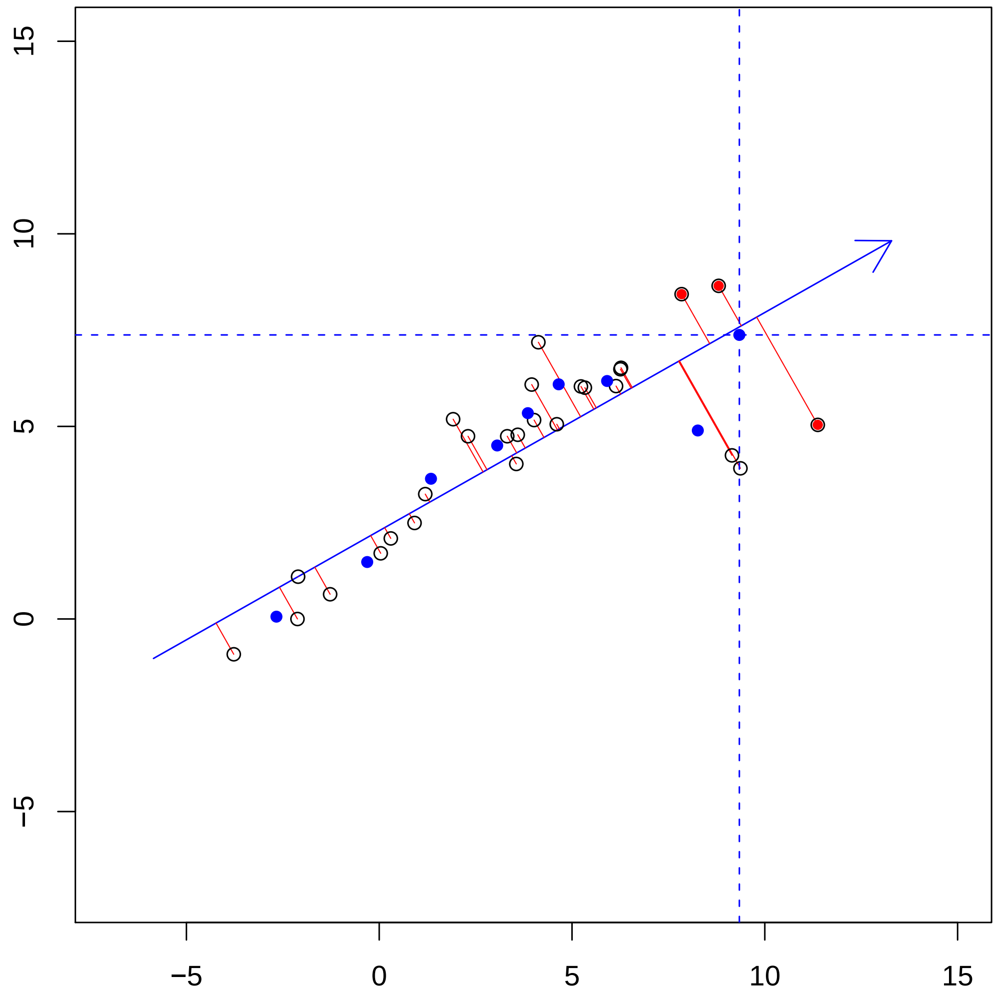
- find first PC
- project data on first PC
- aggregate sorted projected data, e.g with the mean



Projection Methods: PCA

pca method, steps:

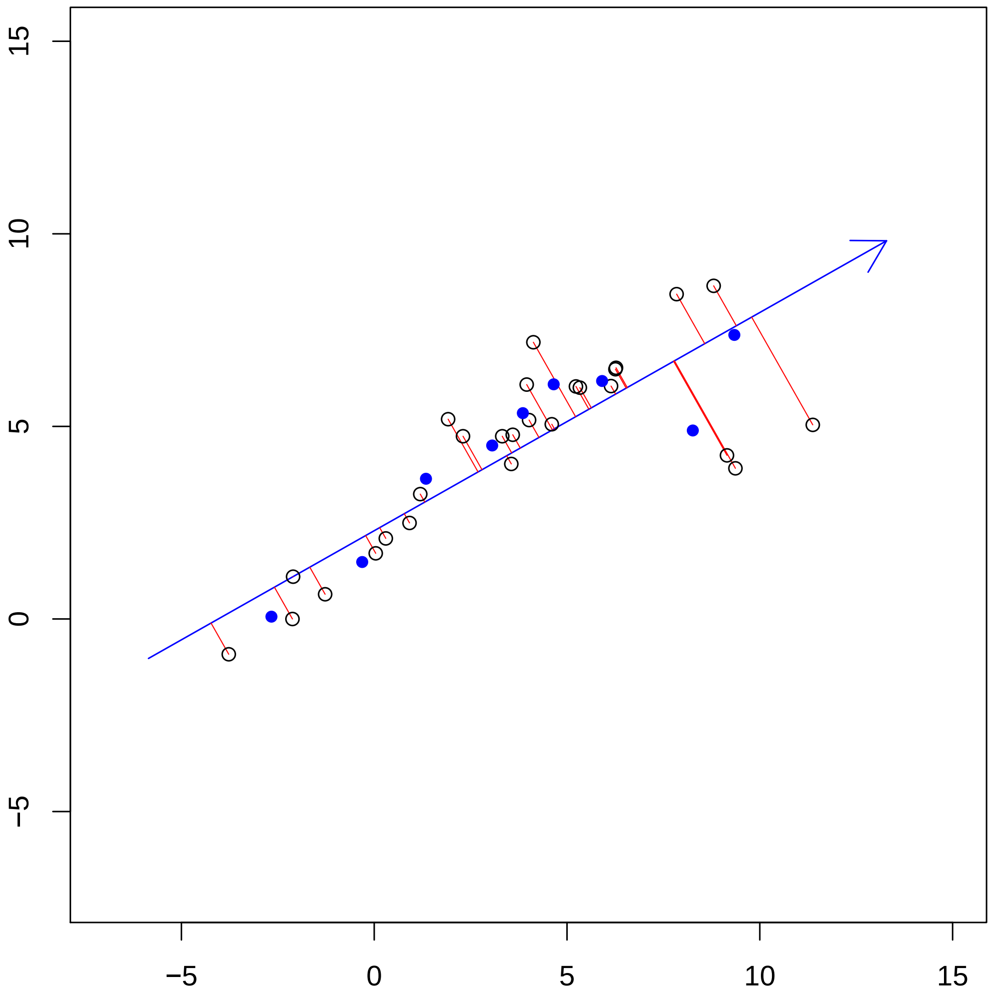
- find first PC
- project data on first PC
- aggregate sorted projected data, e.g with the mean



Projection Methods: PCA

pca method, steps:

- find first PC
- project data on first PC
- aggregate sorted projected data, e.g with the mean



Projection Methods: PCA

Disadvantages of this method (PCA):

- Cannot deal with outliers
- Cannot deal with mixed structures of data

First solution of this problems:

- Robust Principal Component Analyses via MCD-Estimator
- Applied on clustered data
  (for good results, we need a modern clustering method, like *Model based clustering*. Do not use classical partitioning or hierarchical methods!)

Disadvantages of this method (PCA via MCD):

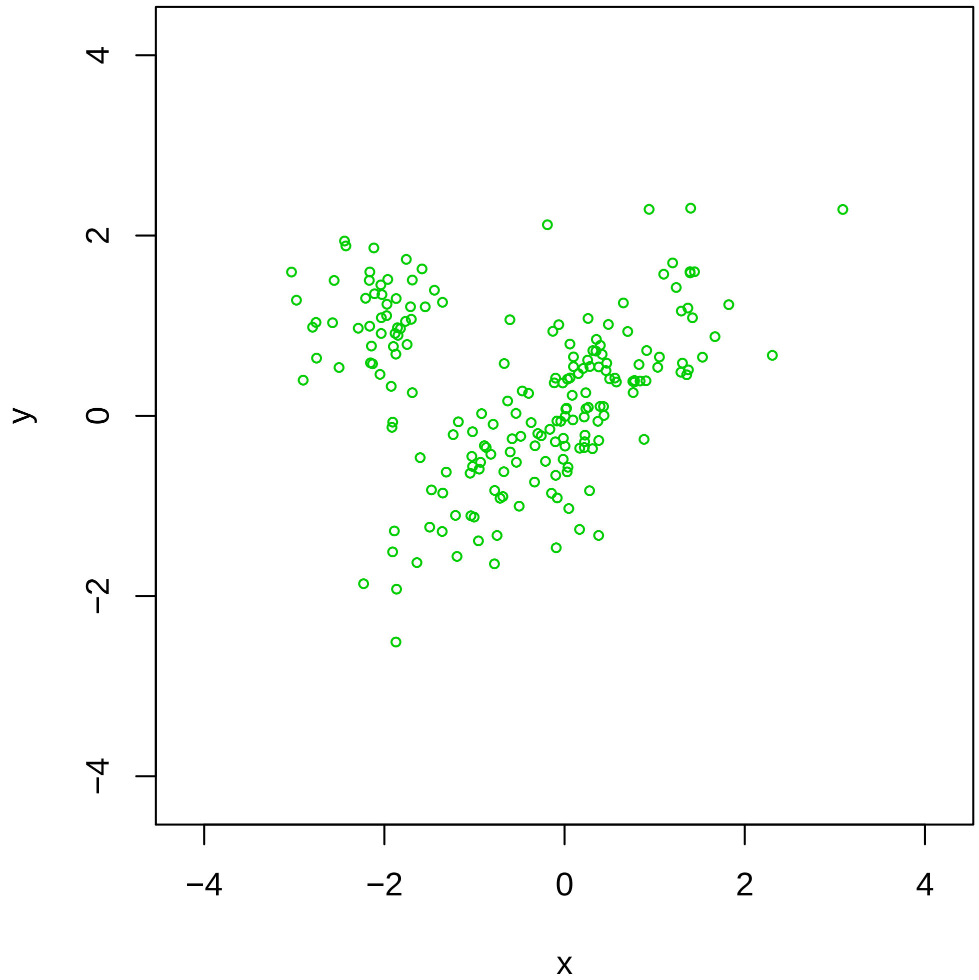
- Computational problems with large data sets and singularity.
- All principal components have to be calculated, but only the first is needed.

Projection Pursuit

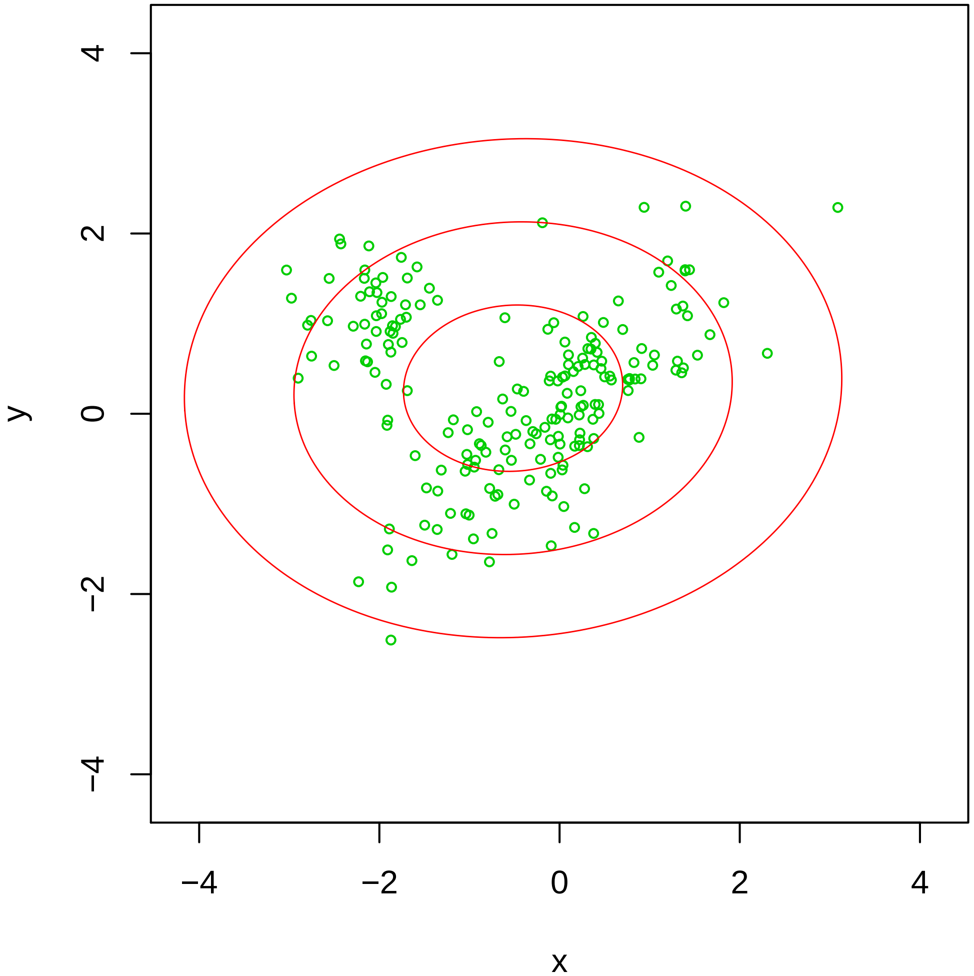
Final solution:

- Robustification of PCA with **Projection Pursuit**:
- Fast computation of the first PC.
- Algorithm of Peter Filzmoser from Vienna University of Technology

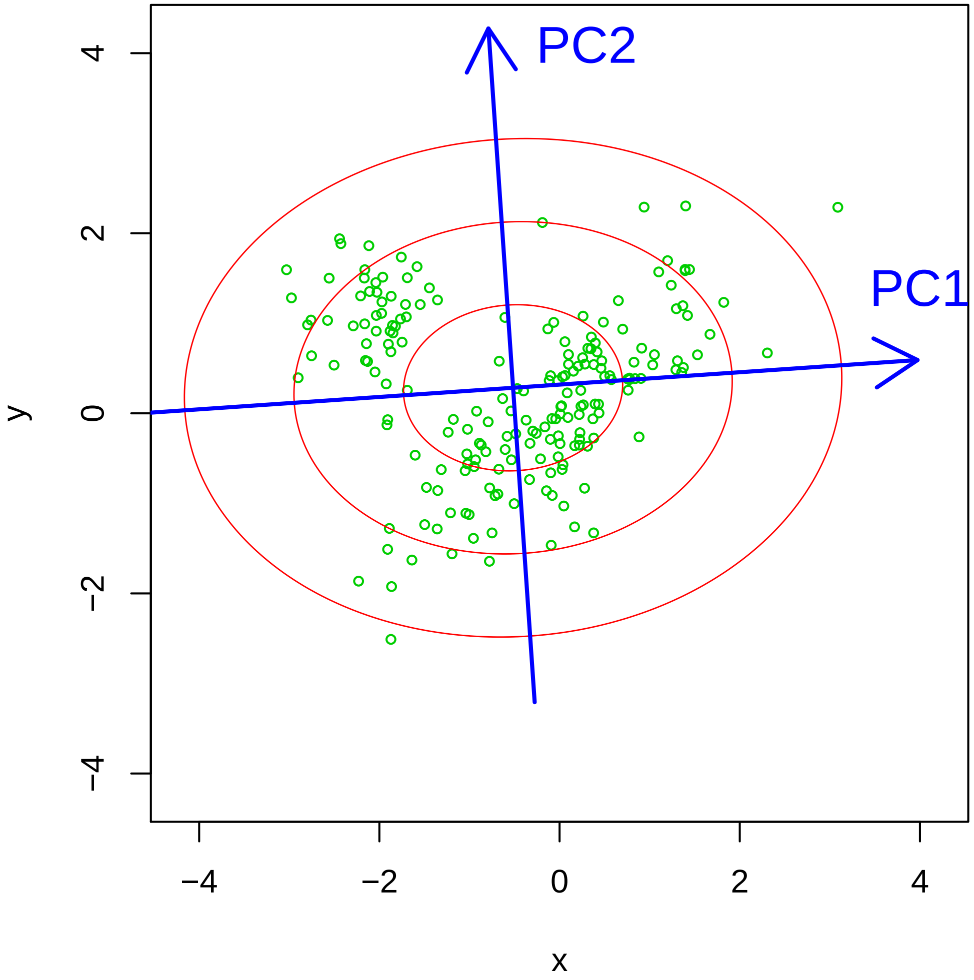
Outliers



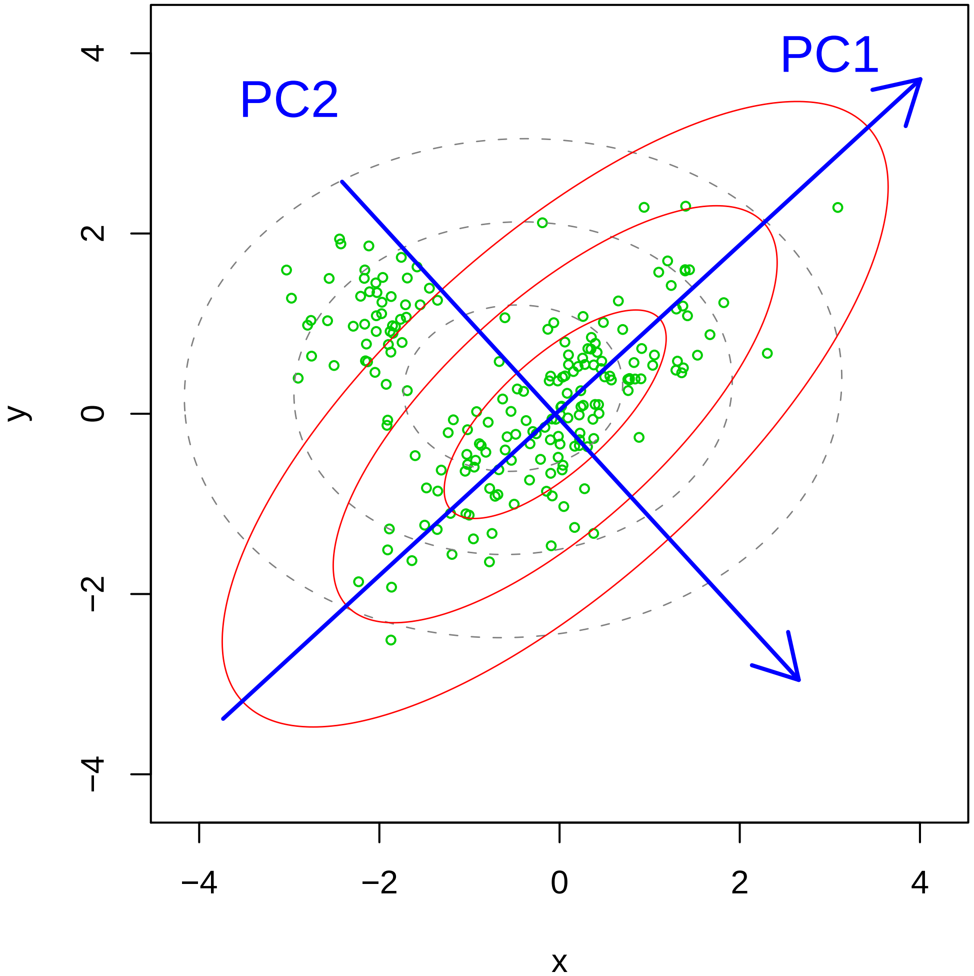
PCA with Outliers



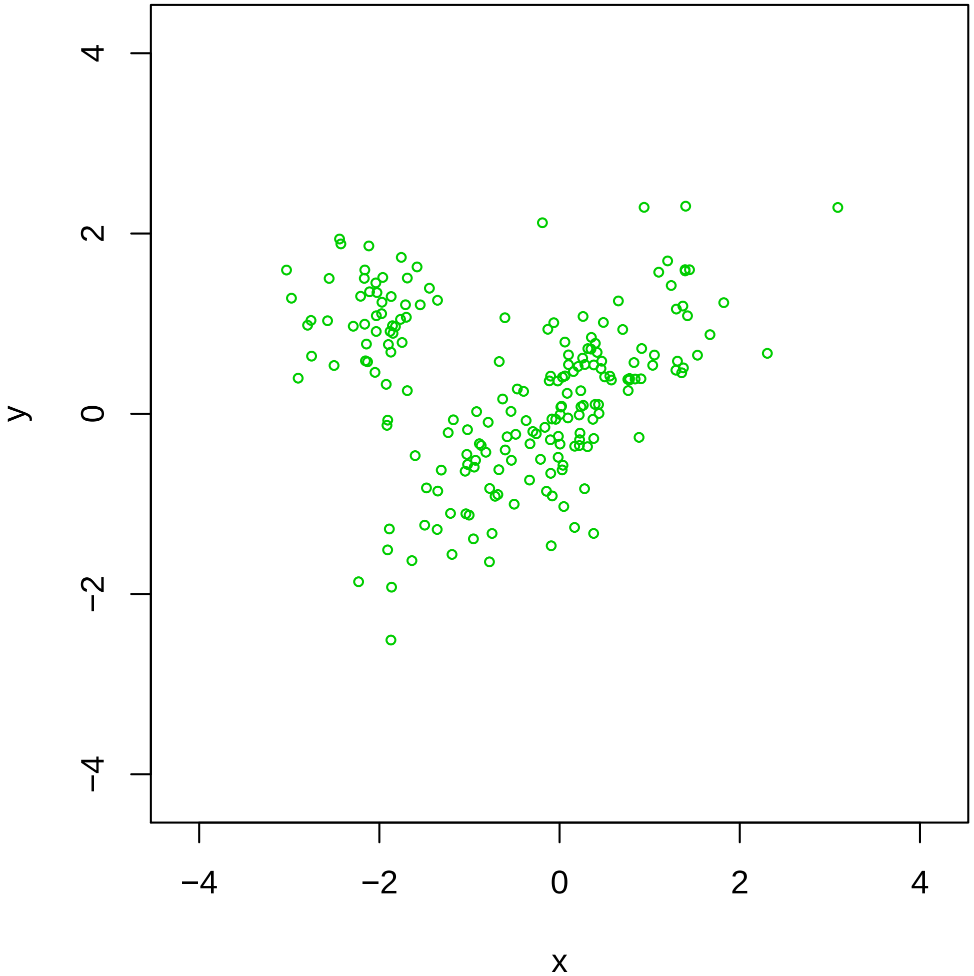
PCA with Outliers



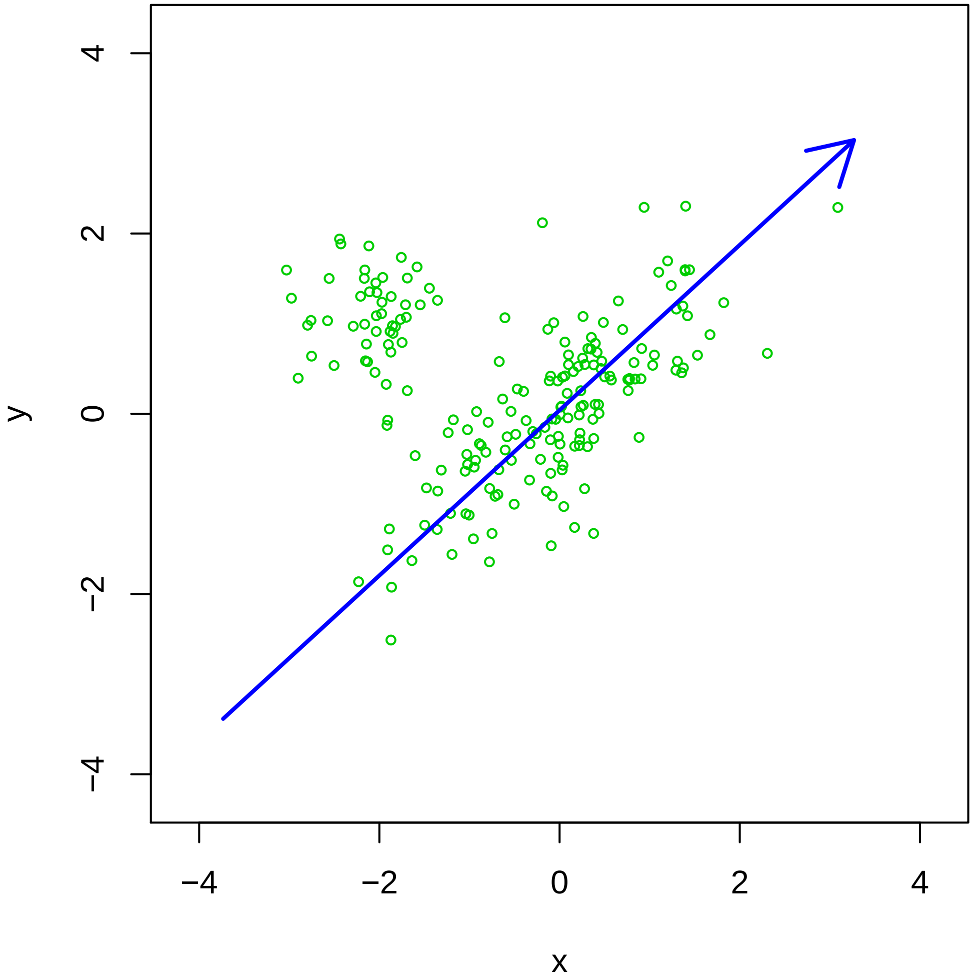
PCA with Outliers



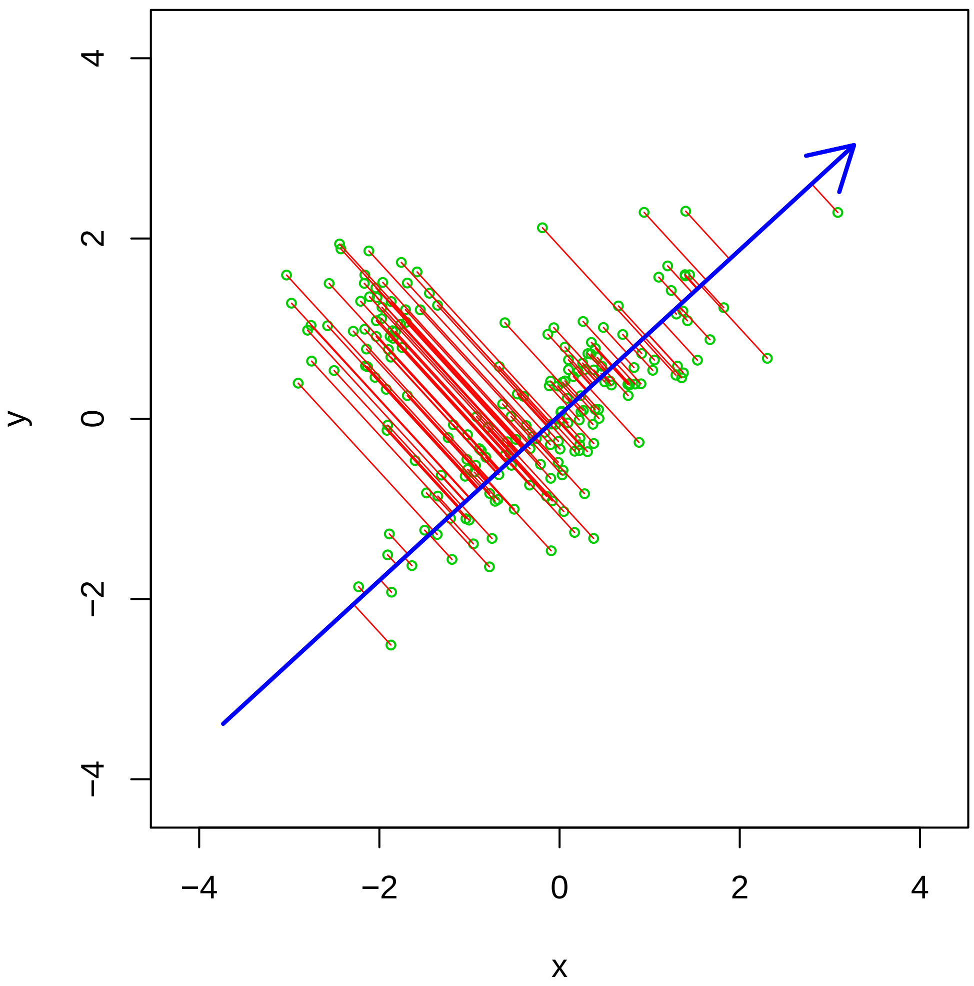
PCA by Projection Pursuit



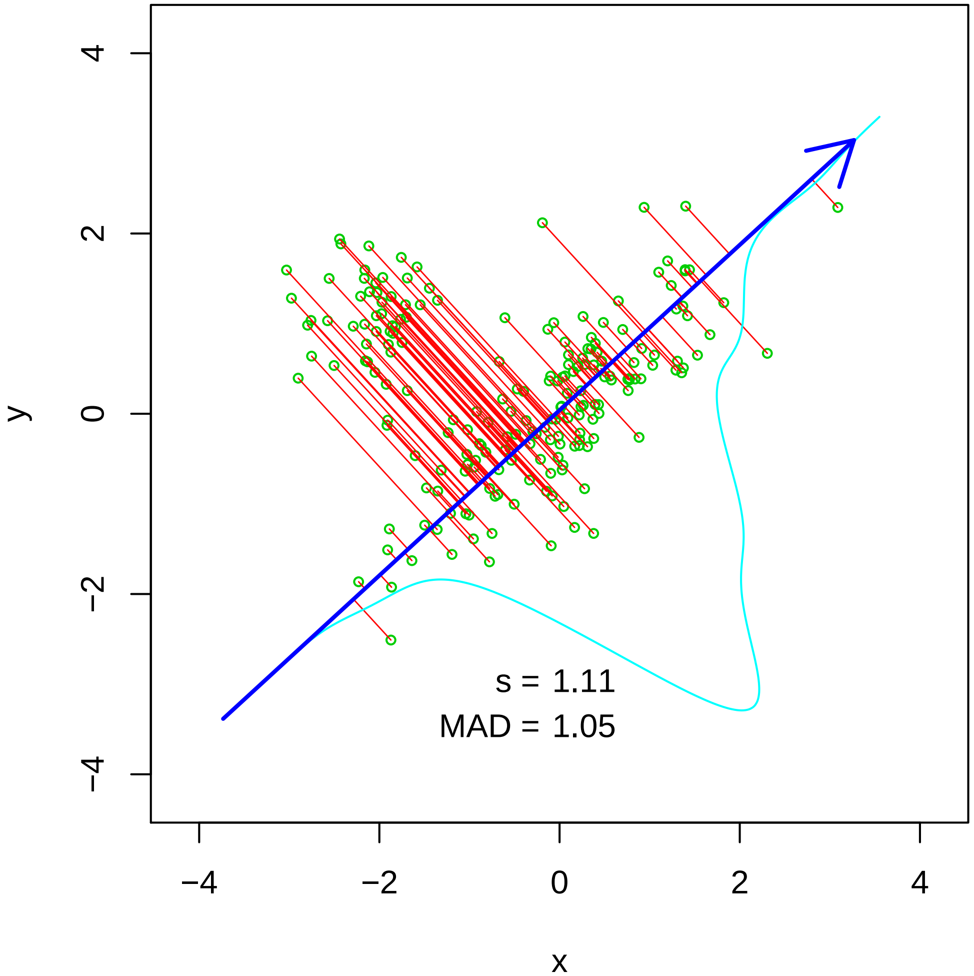
PCA by Projection Pursuit



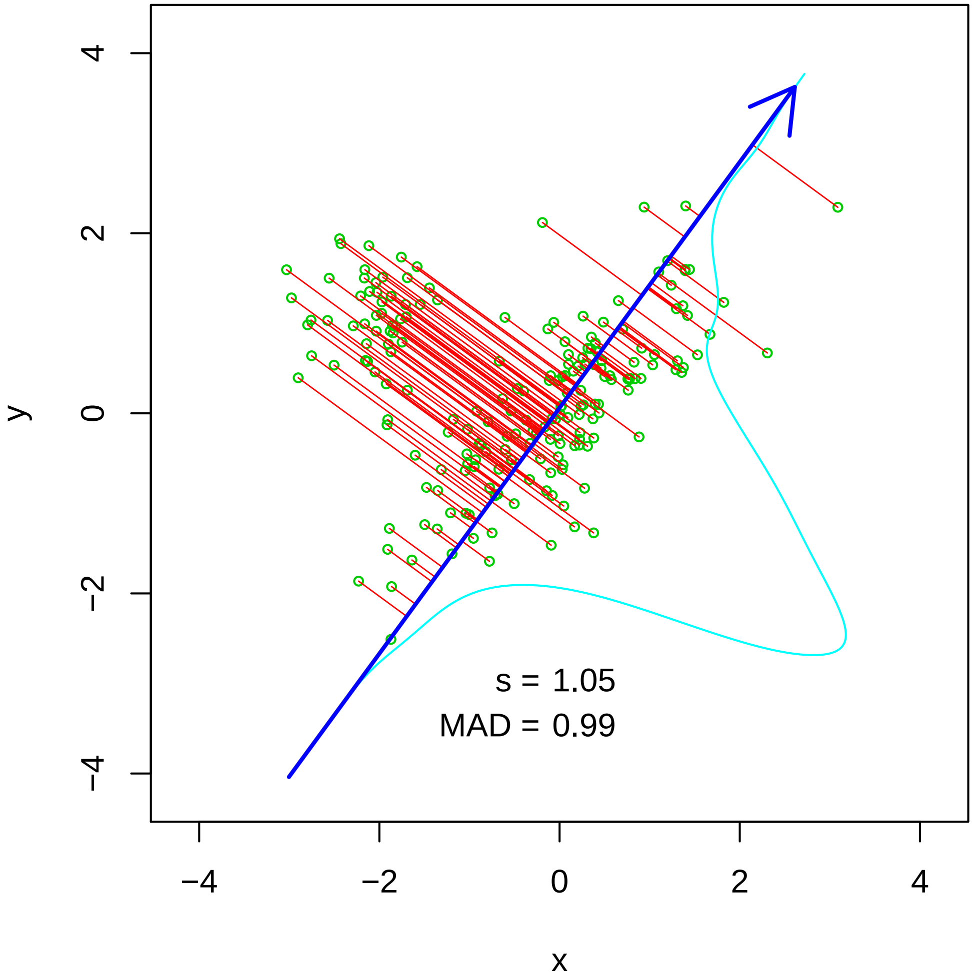
PCA by Projection Pursuit



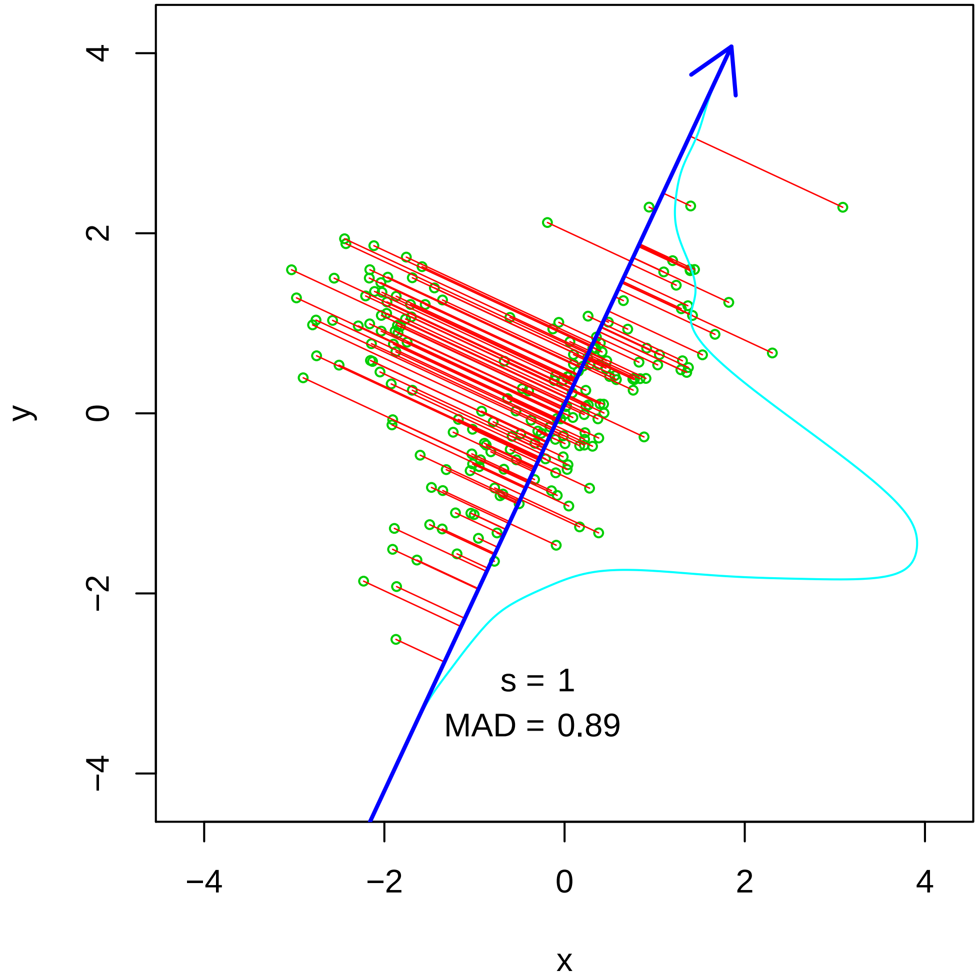
PCA by Projection Pursuit



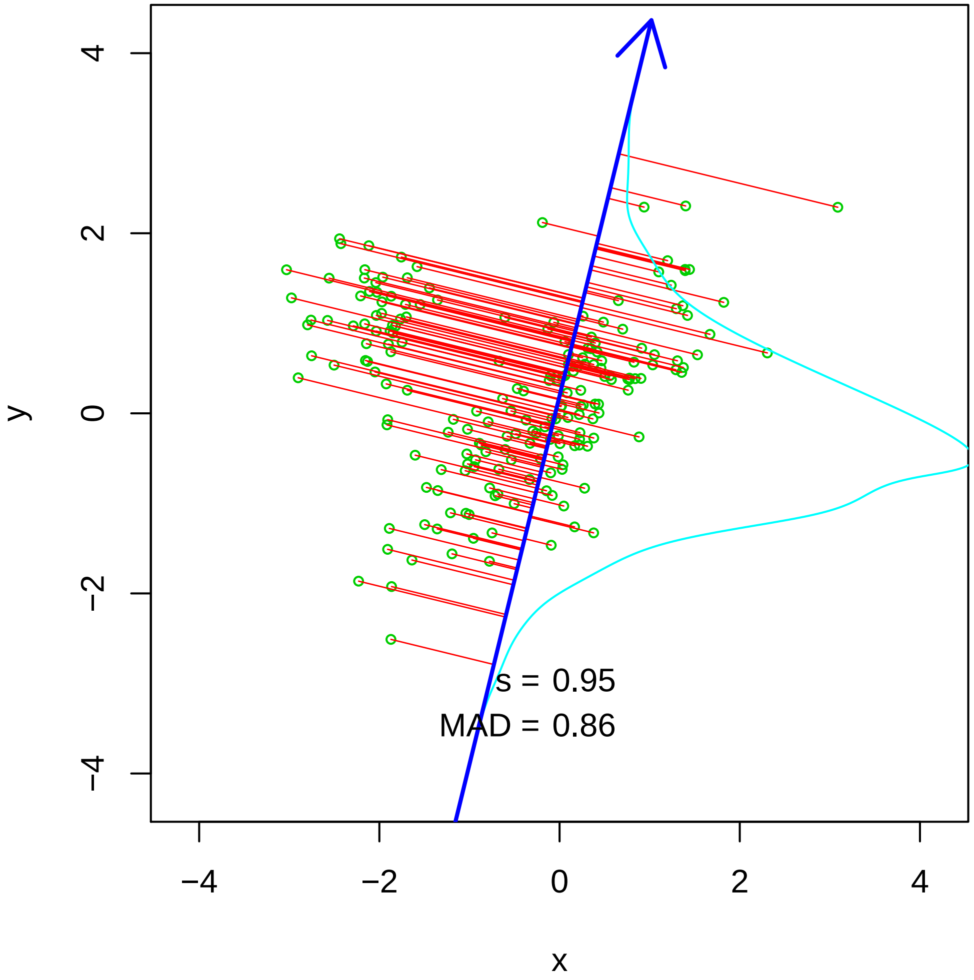
PCA by Projection Pursuit



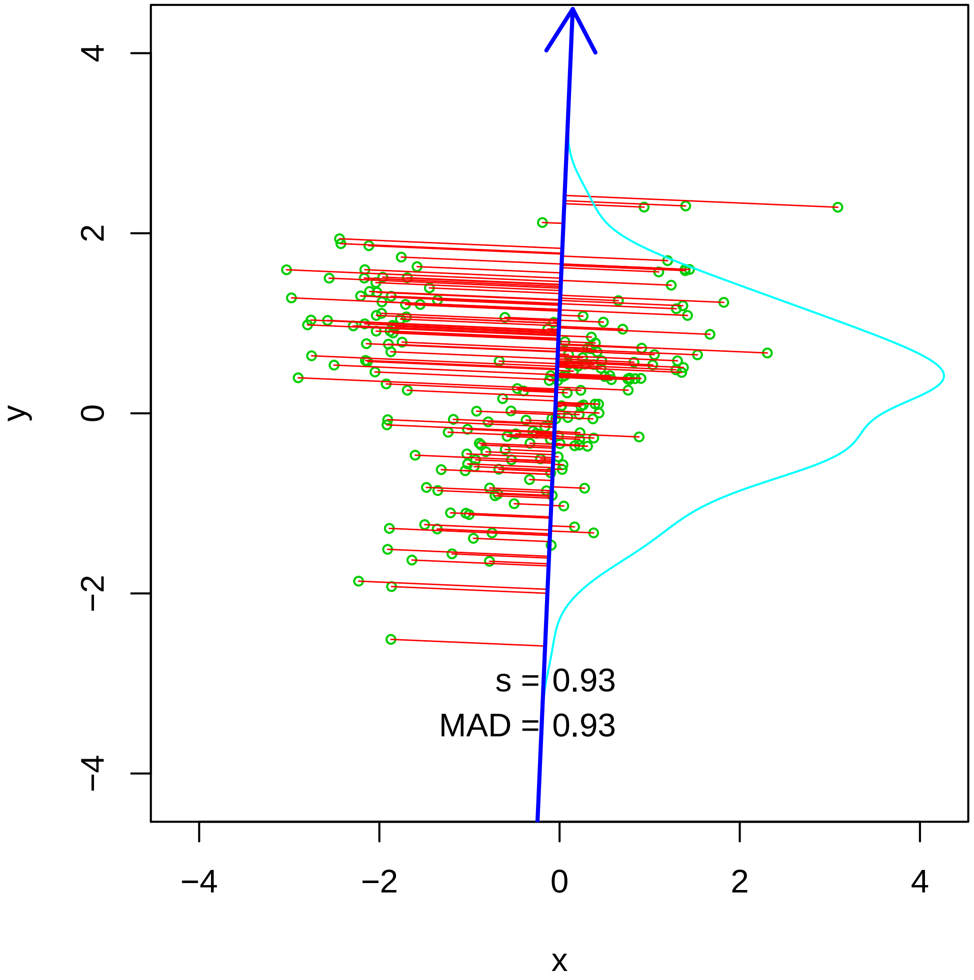
PCA by Projection Pursuit



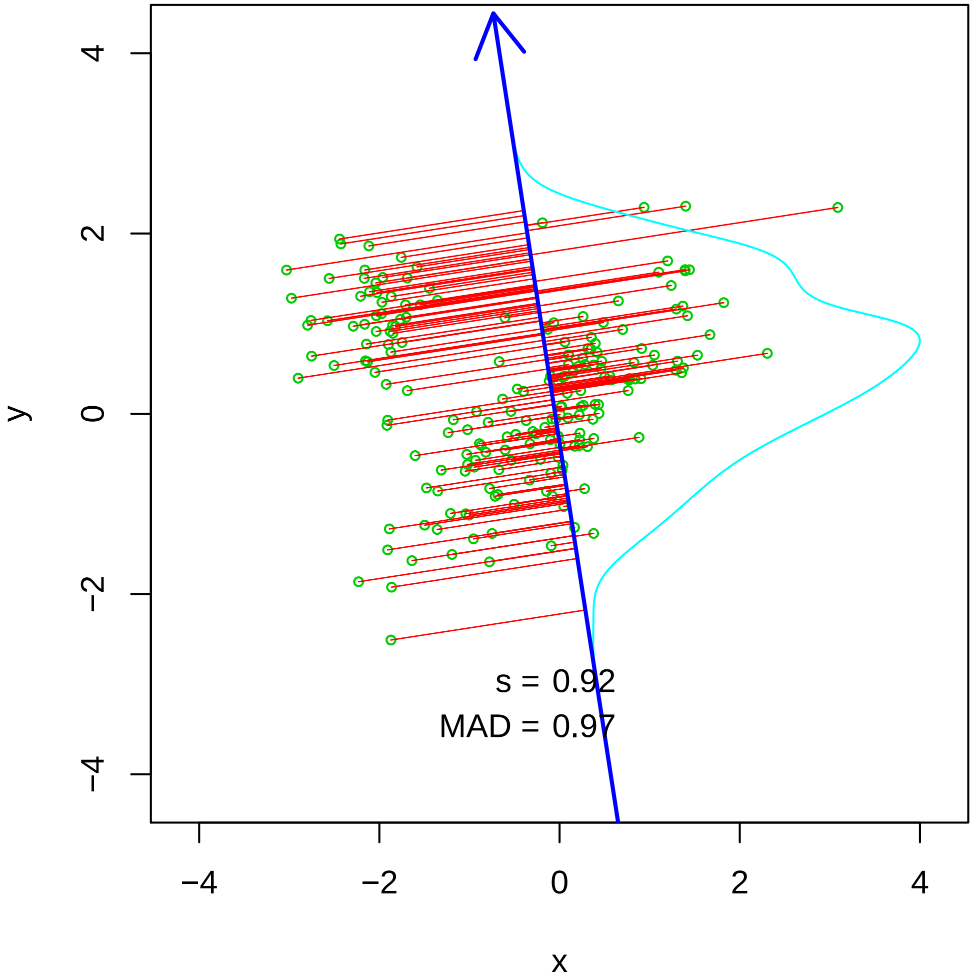
PCA by Projection Pursuit



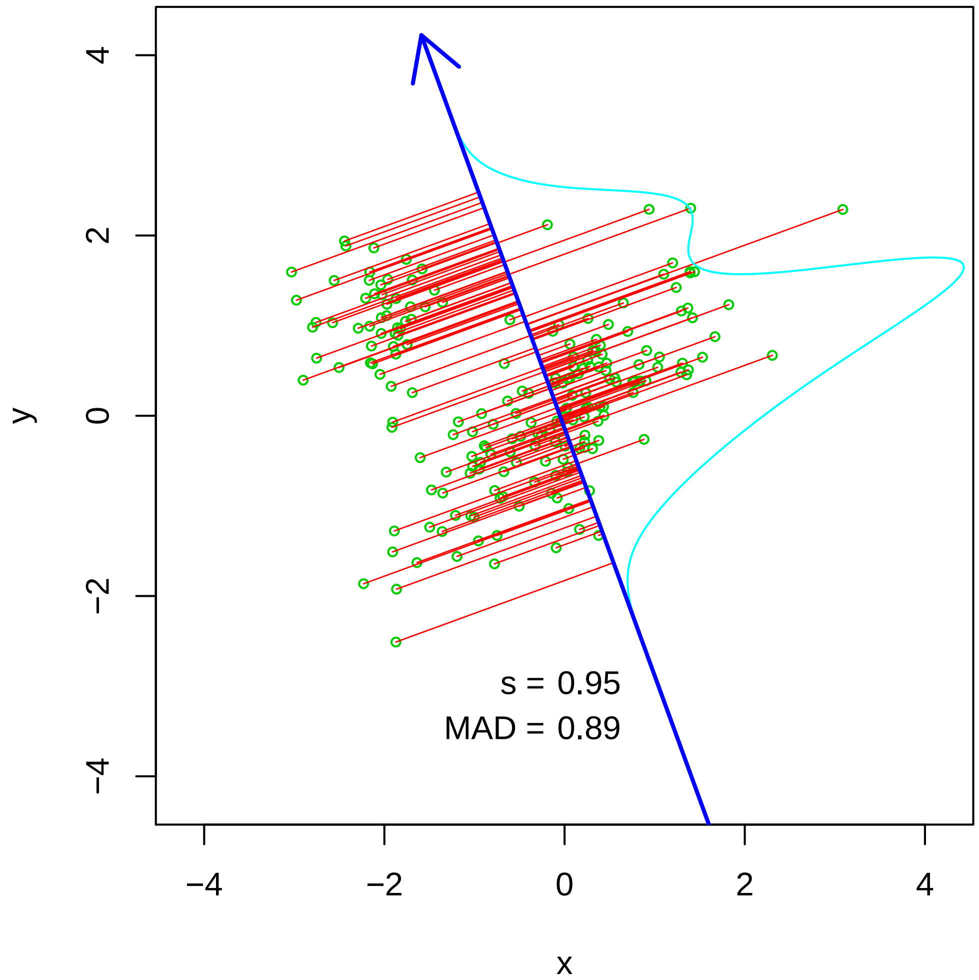
PCA by Projection Pursuit



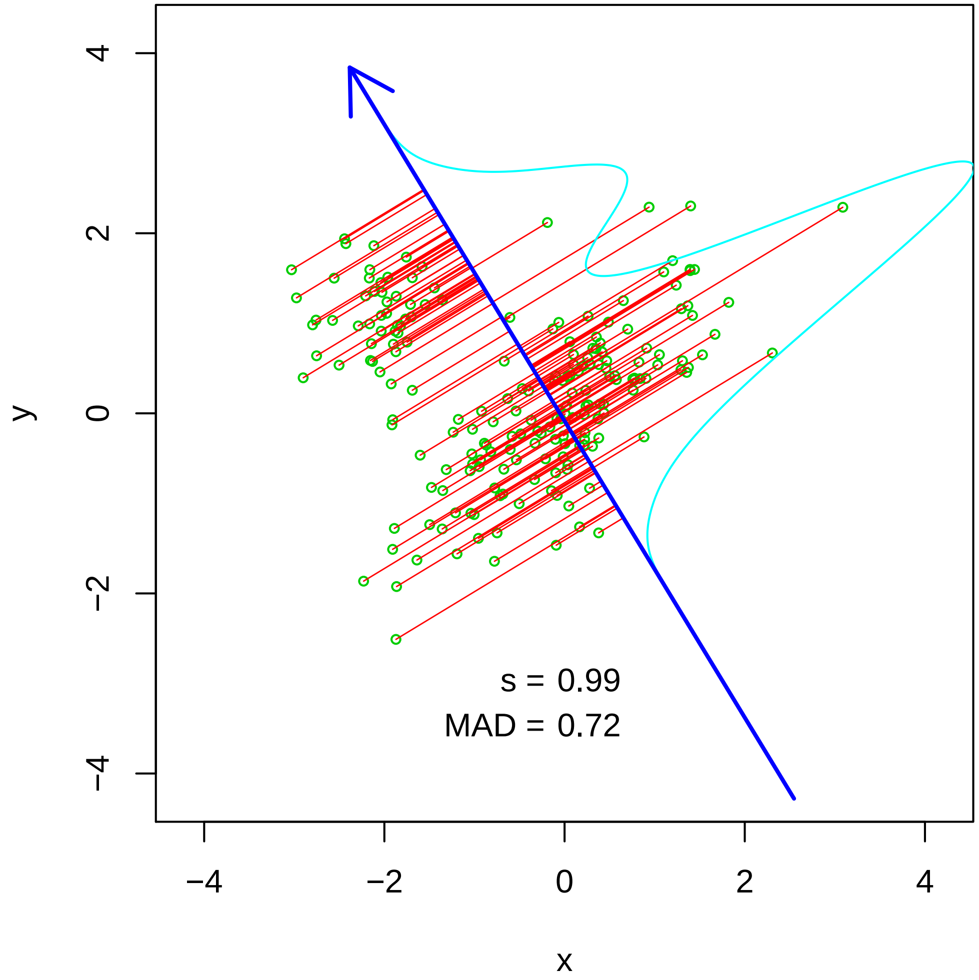
PCA by Projection Pursuit



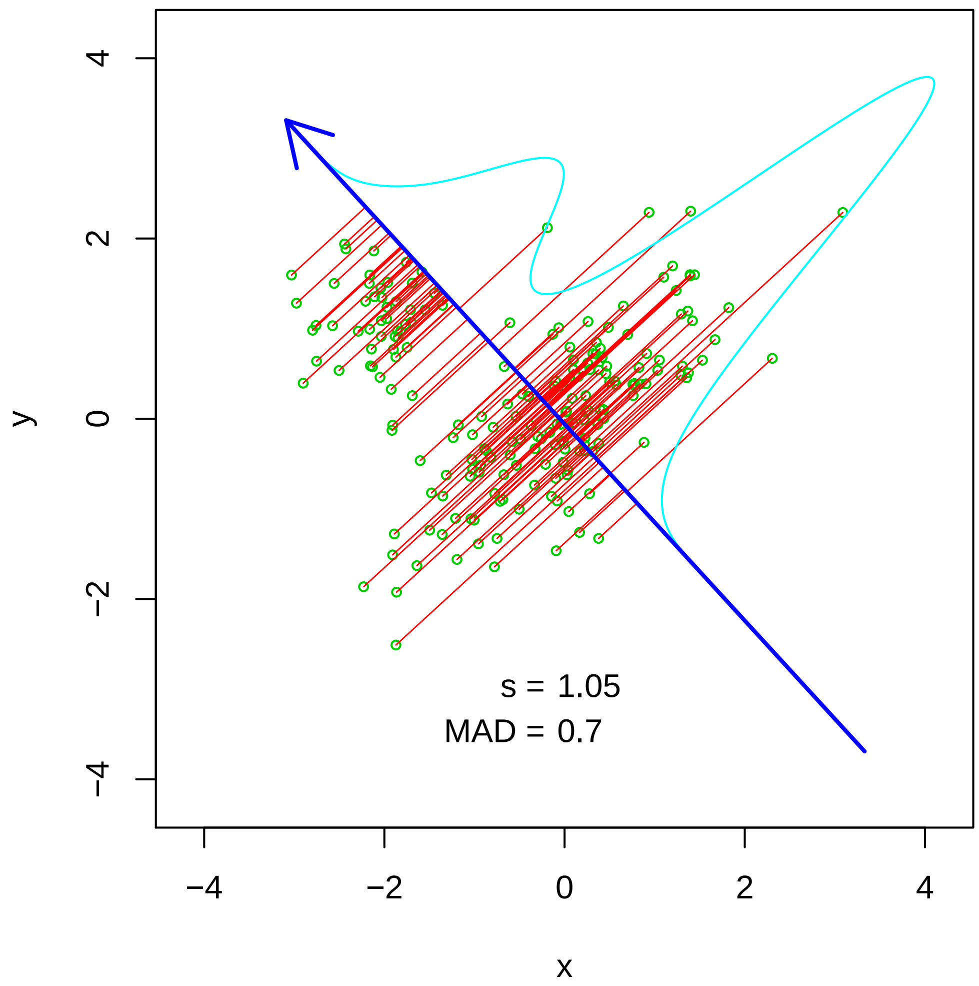
PCA by Projection Pursuit



PCA by Projection Pursuit



PCA by Projection Pursuit



Choice of the Projection Directions

Problem:

search for the direction in $\mathbb{R}^p$ that maximizes the variance of the projected data.

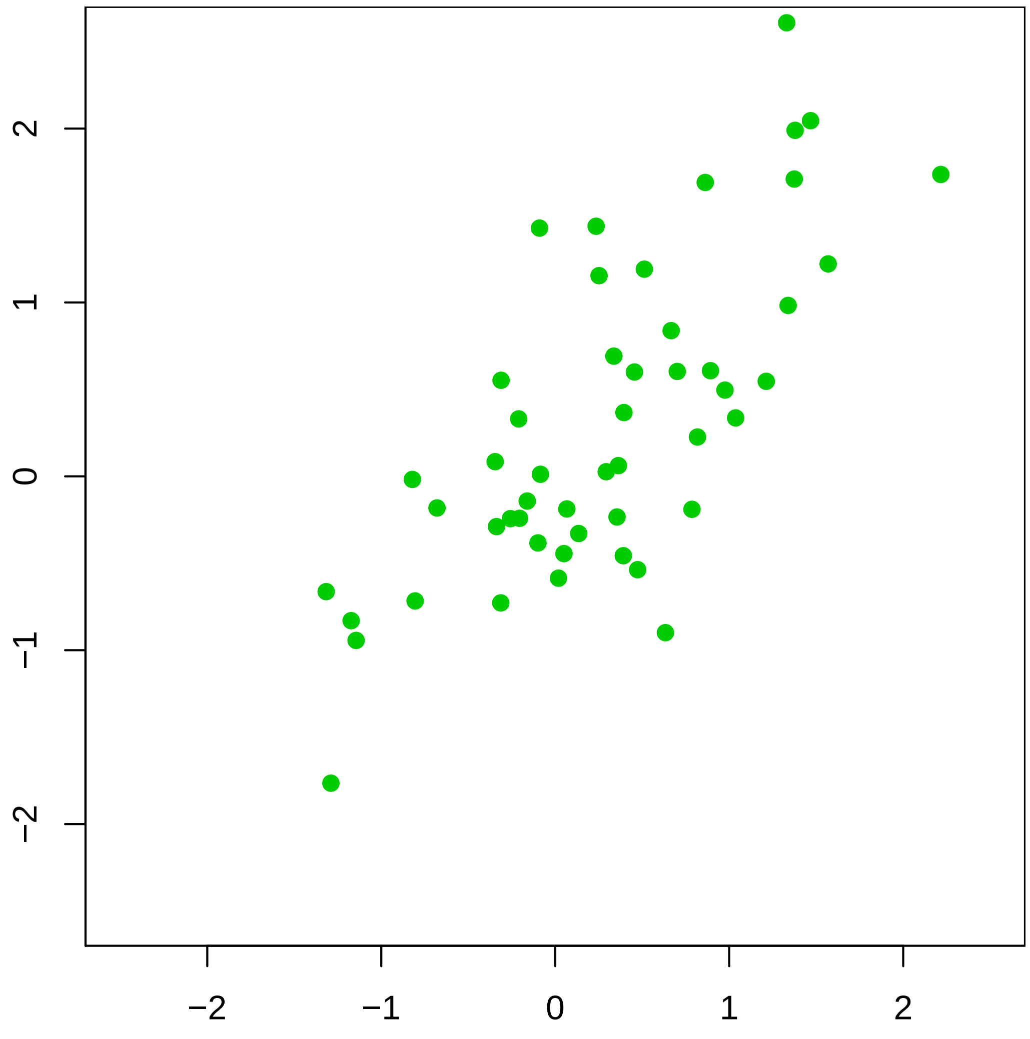
$\Longrightarrow$ Number of possible directions is ∞

Approach:

reduction to a feasible number of **candidate directions**.

Projection Directions: `PCAproj`

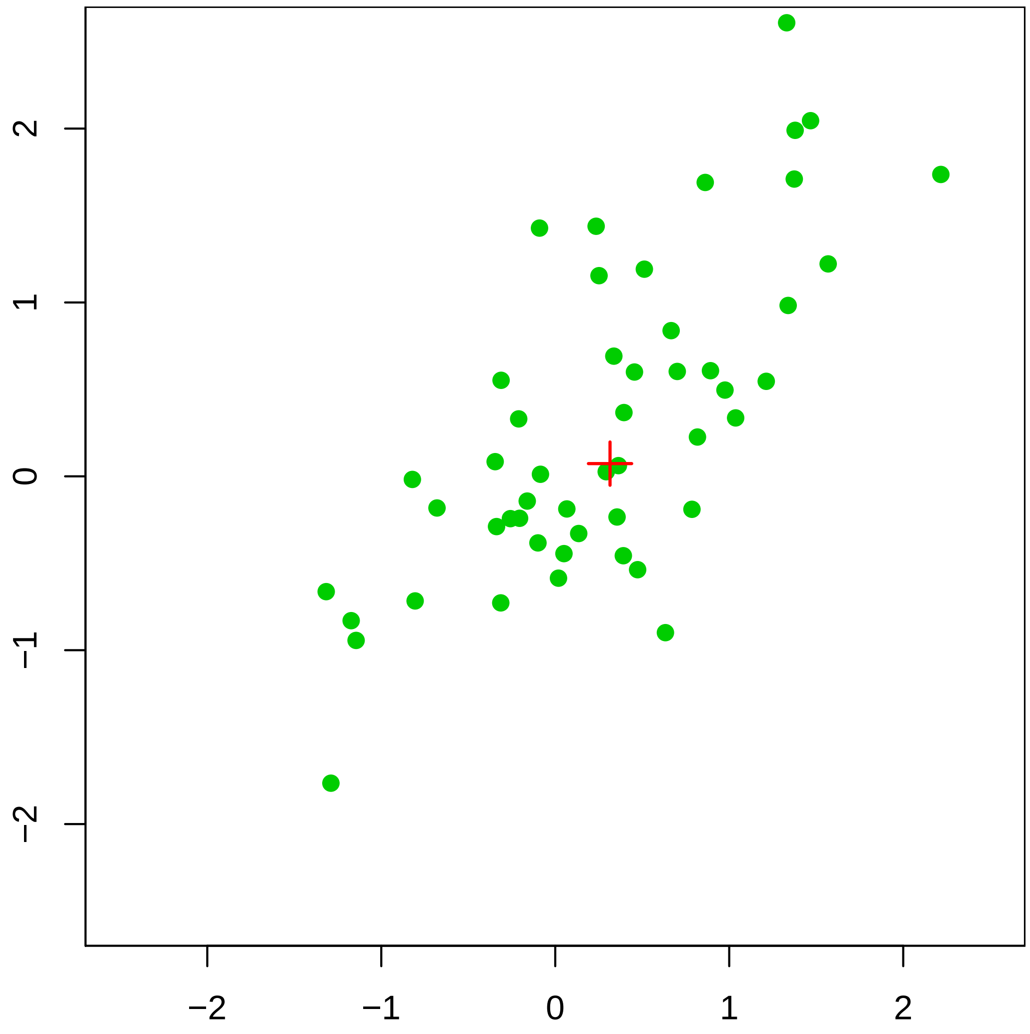
Candidate Directions:



Projection Directions: `PCAproj`

Candidate Directions:

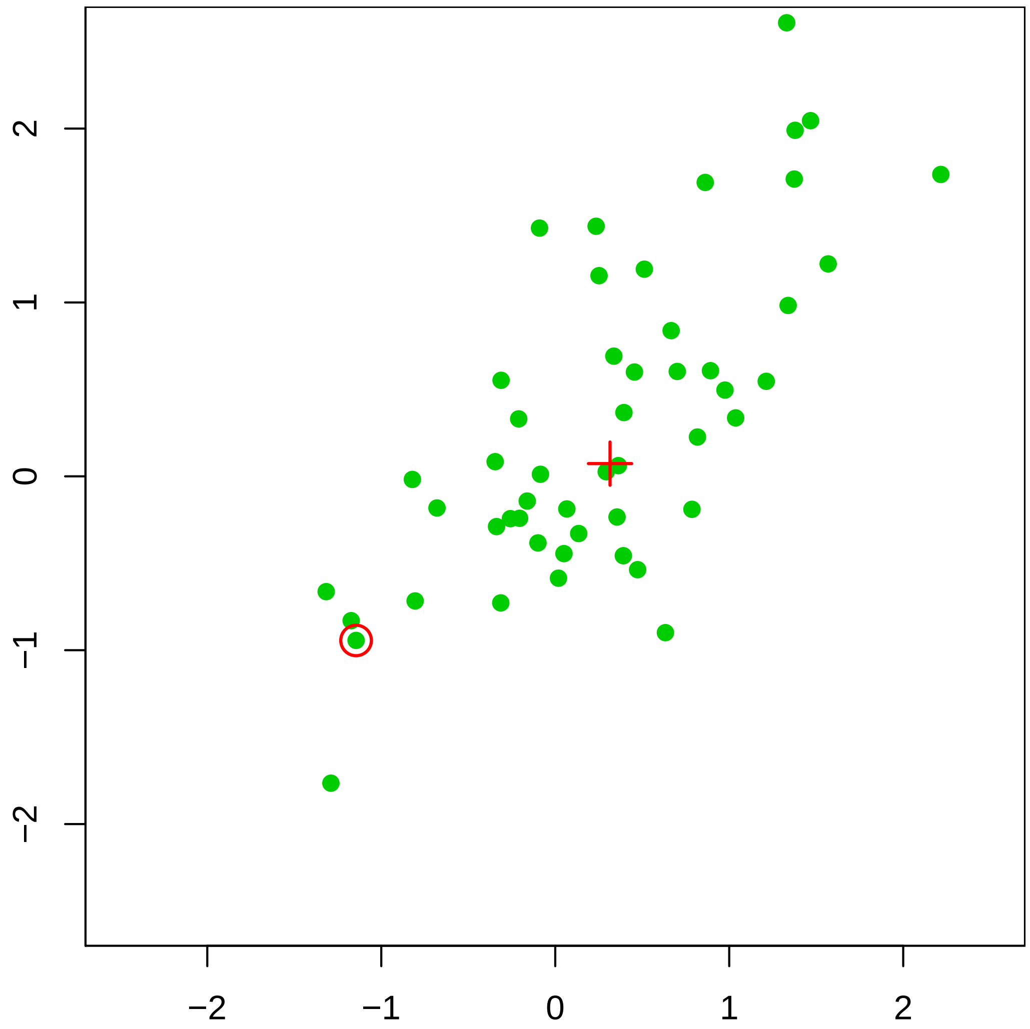
– robust center (L_1-median, coordinatewise median)



Projection Directions: `PCAproj`

Candidate Directions:

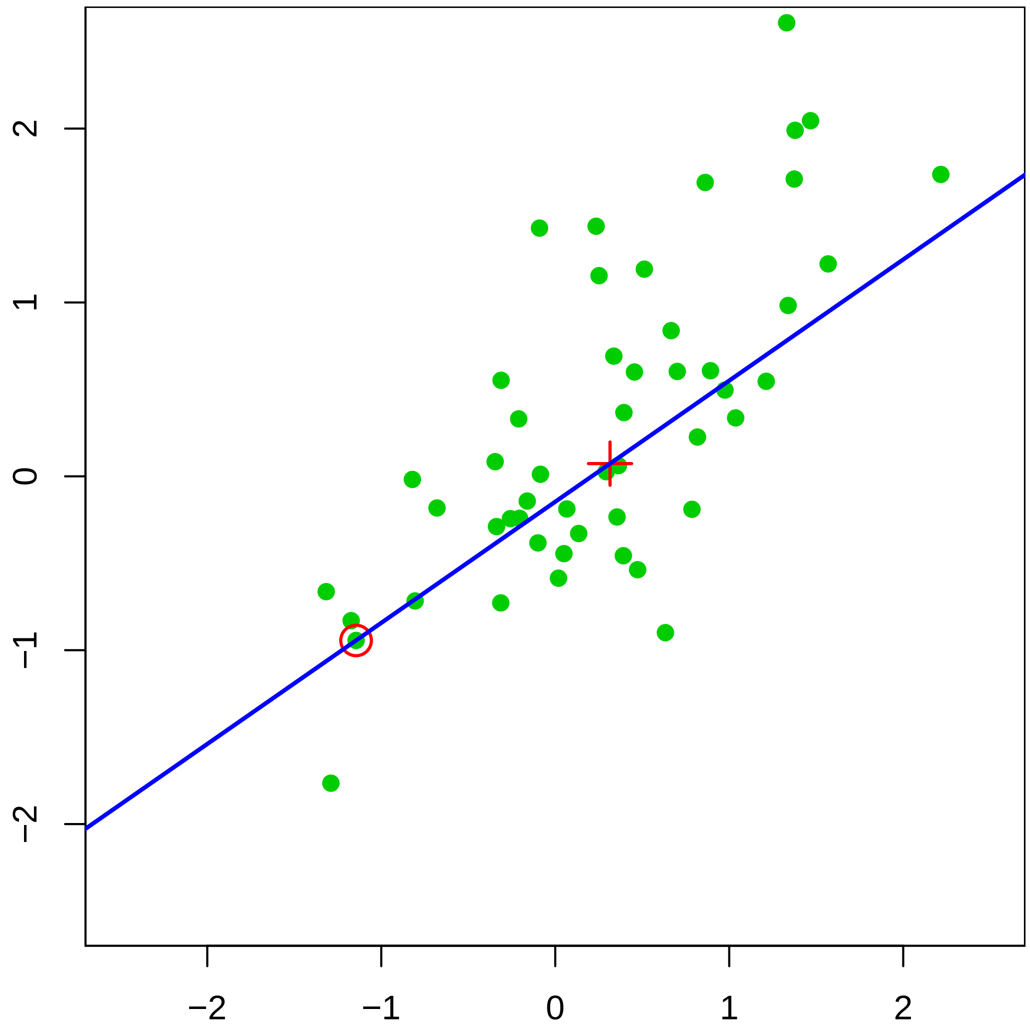
- robust center (L_1-median, coordinatewise median)
- each data point



Projection Directions: `PCAproj`

Candidate Directions:

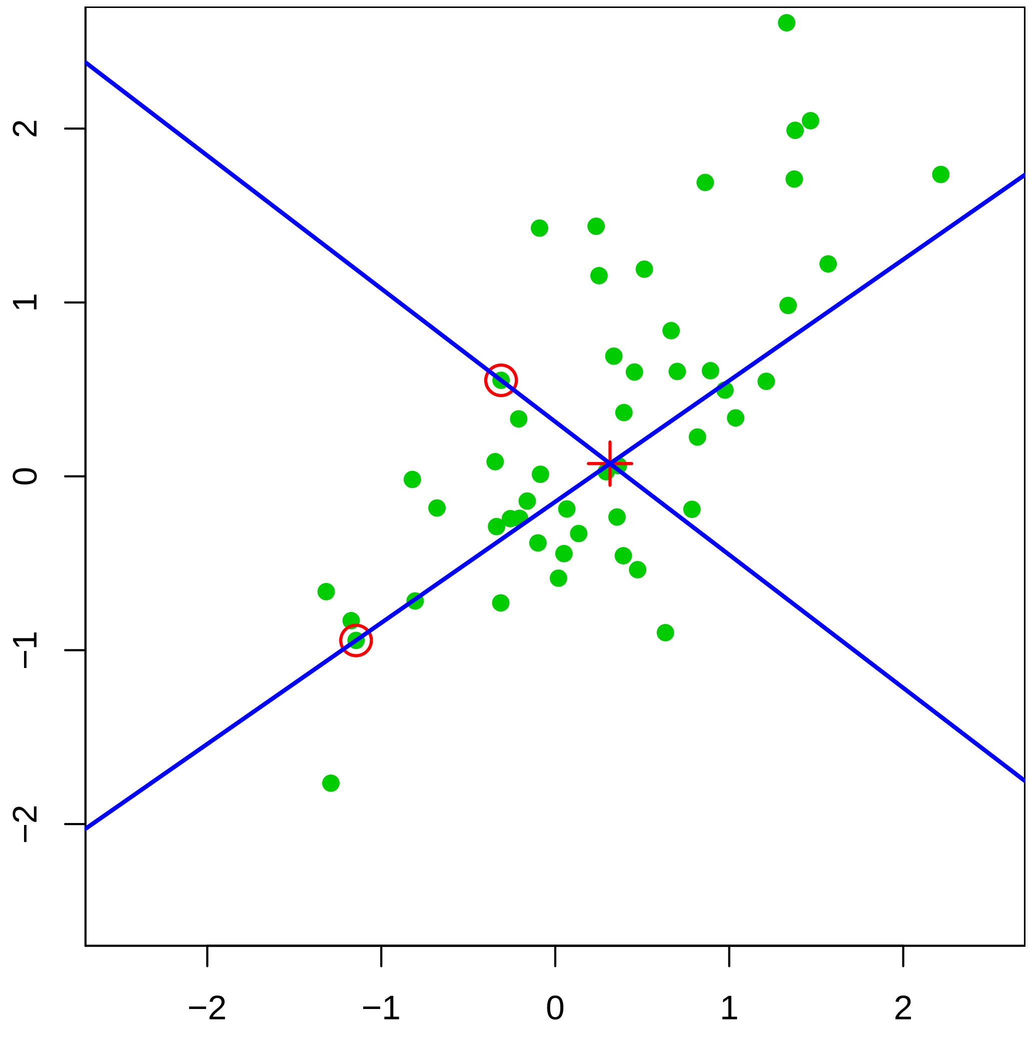
- robust center (L_1-median, coordinatewise median)
- each data point



Projection Directions: `PCAproj`

Candidate Directions:

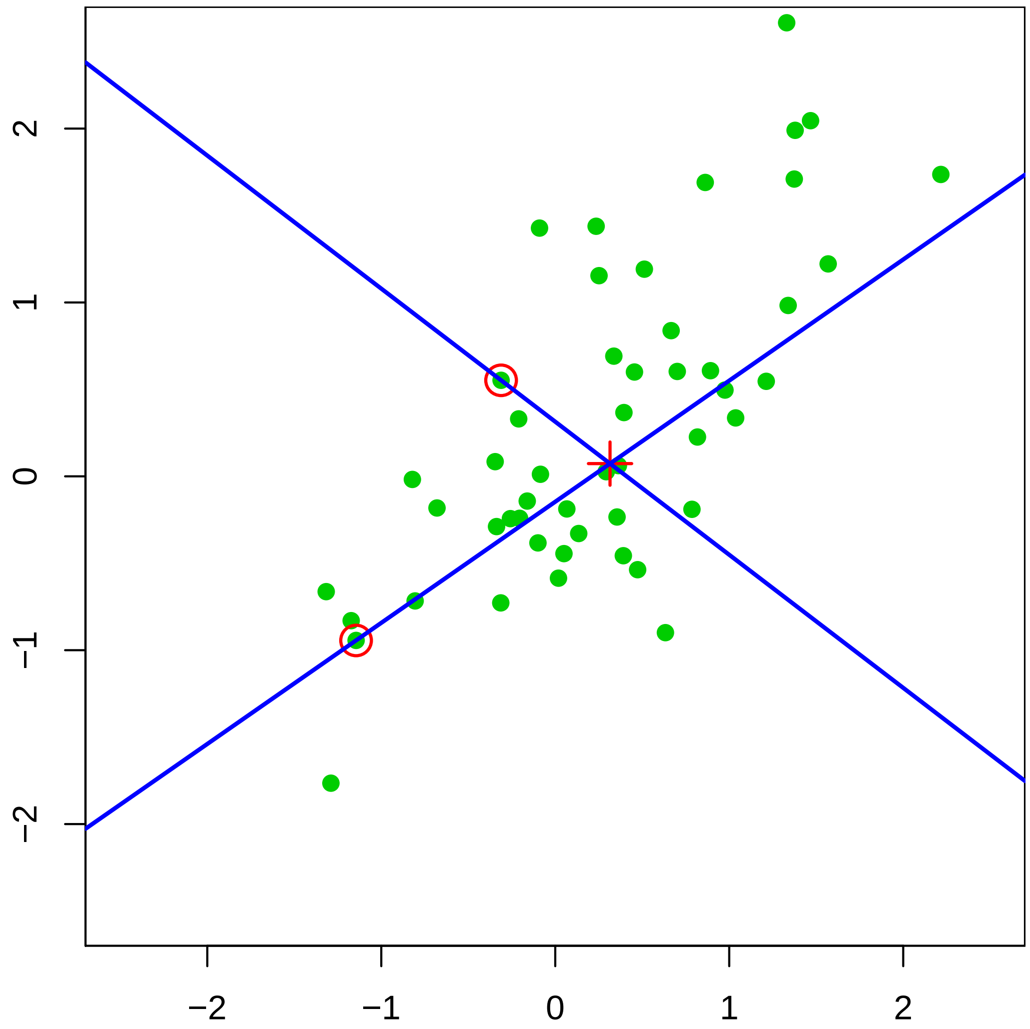
- robust center (L_1-median, coordinatewise median)
- each data point



Projection Directions: `PCAproj`

Candidate Directions:

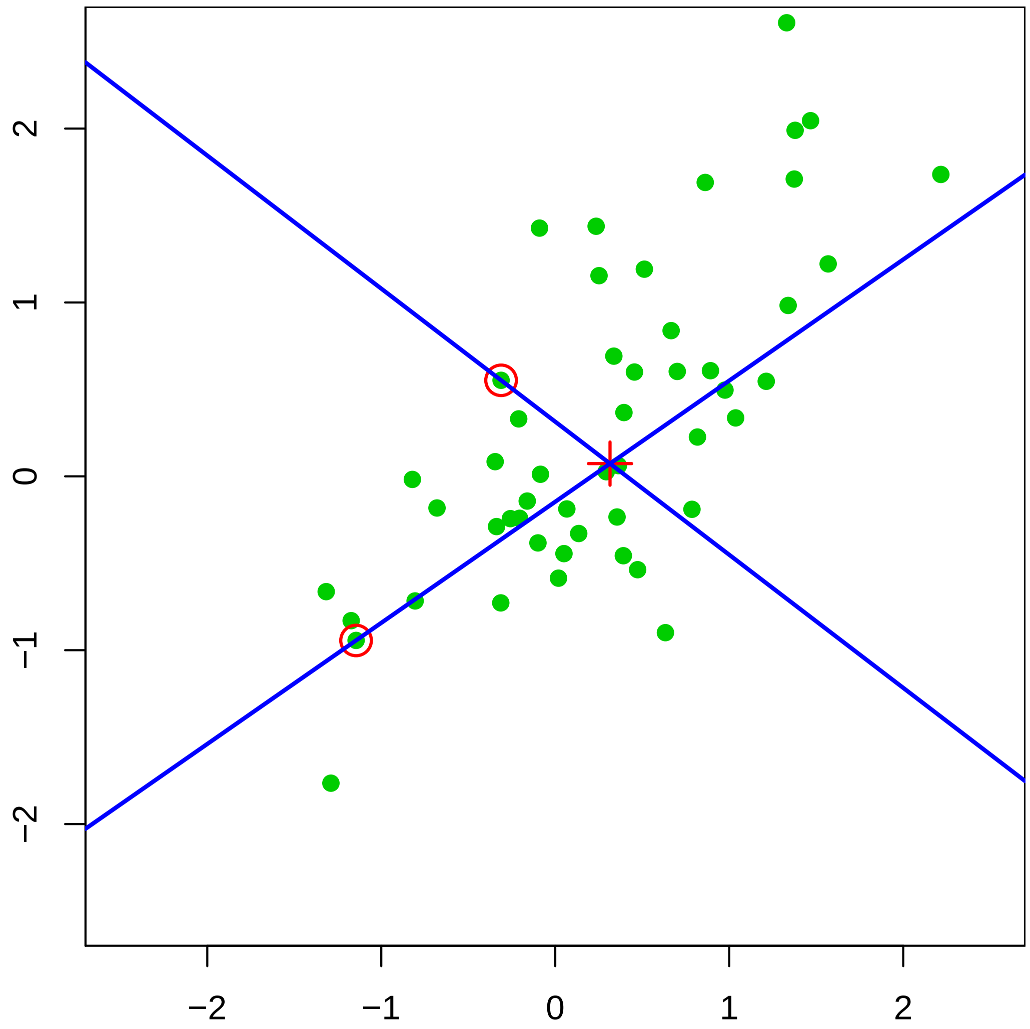
- robust center (L_1-median, coordinatewise median)
- each data point
- additionally random directions through center



Projection Directions: `PCAproj`

Candidate Directions:

- robust center (L_1-median, coordinatewise median)
- each data point
- additionally random directions through center
- additional directions by linear combinations of data points



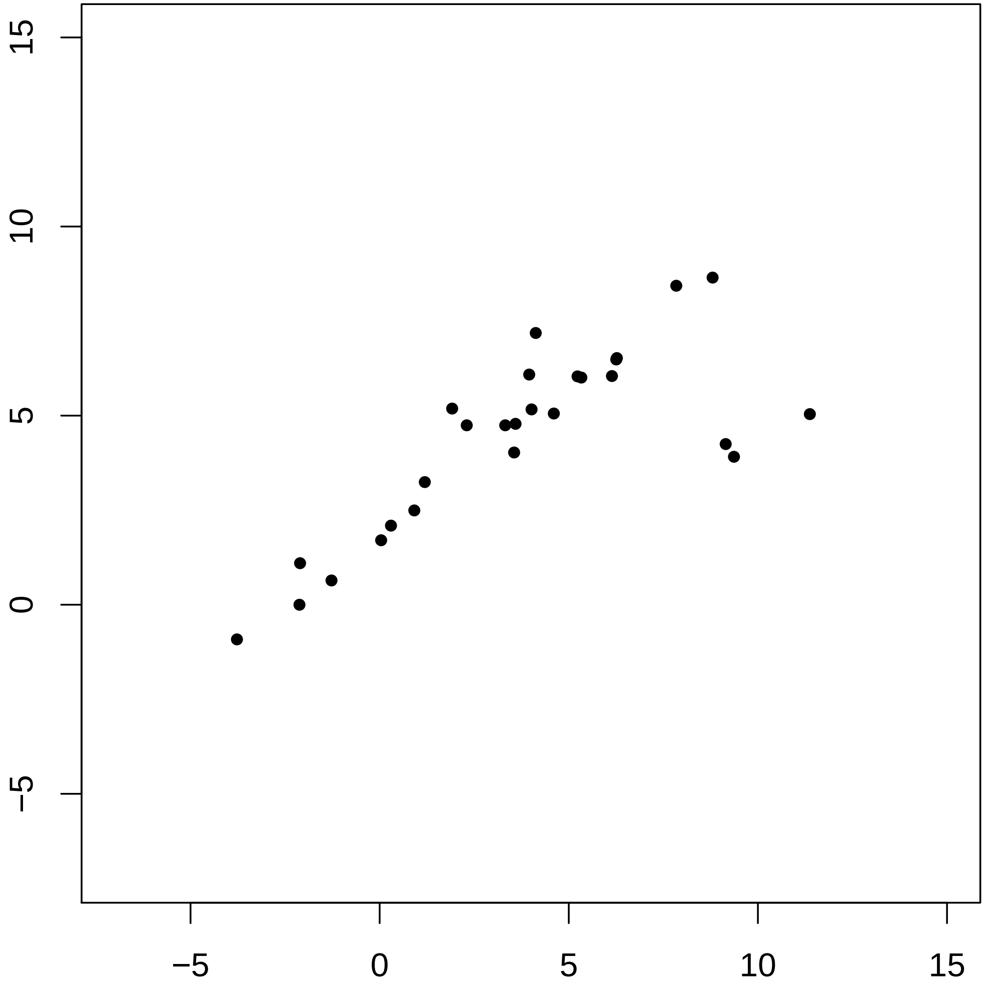
Projection Directions: `PCAgrad`, `PCAgrid`

There are other solutions for this problem ...

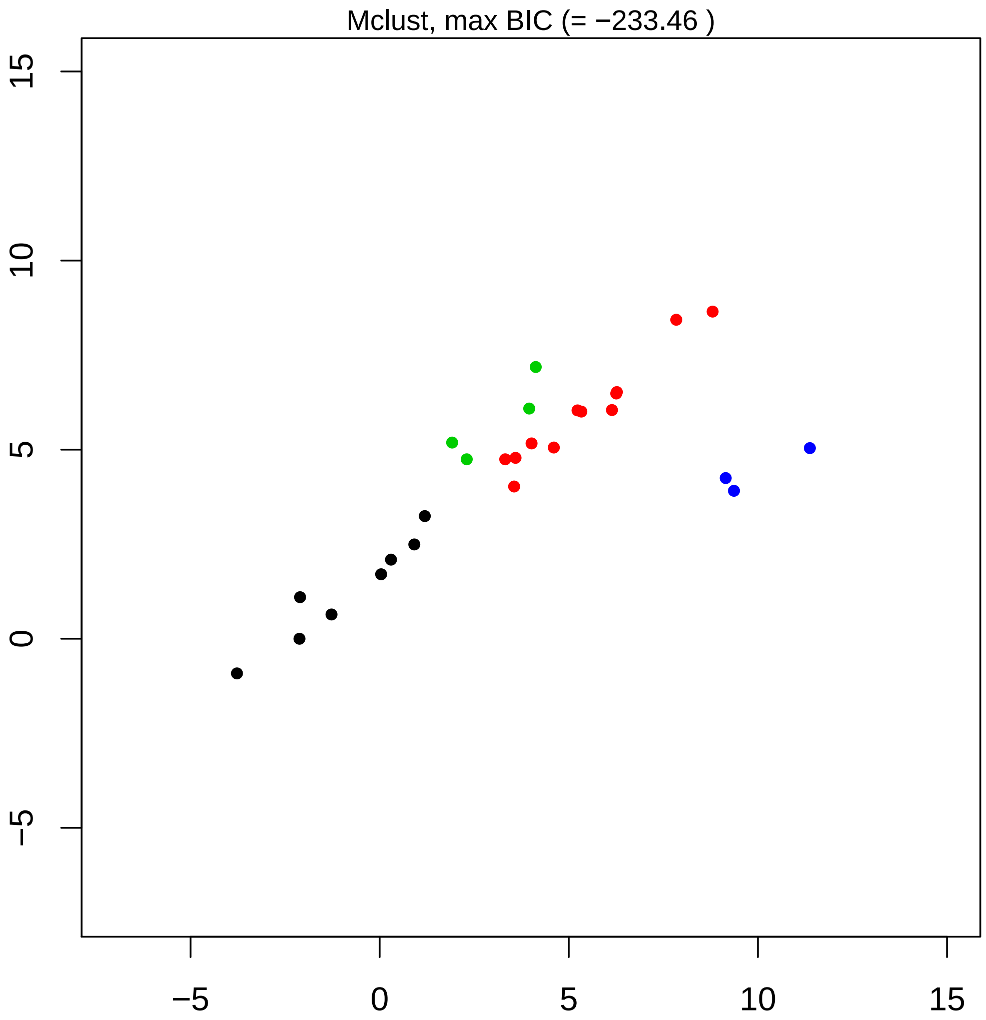
All these methods on PCA by projection pursuit are implemented in an R-package from Fritz Heinrich and Peter Filzmoser.

It will be available in near future on CRAN.

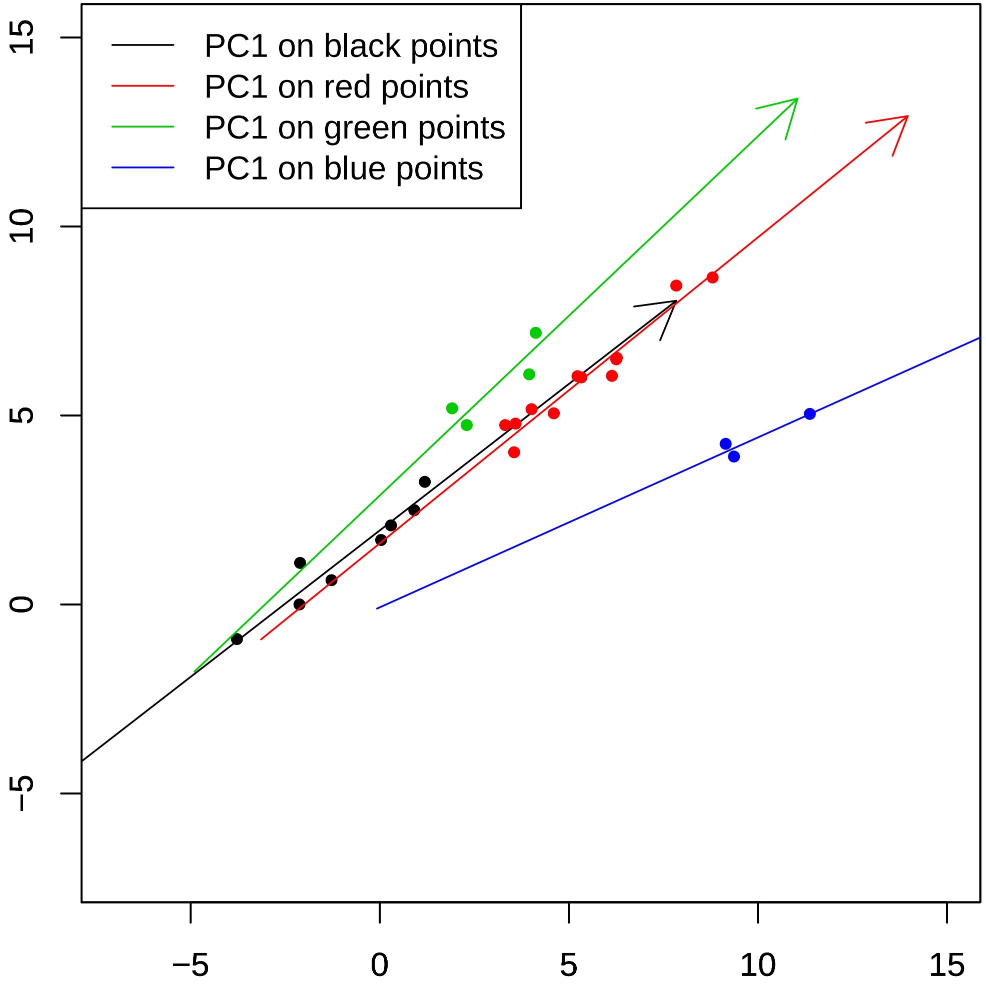
Algorithm: `clustPPPCA`



Algorithm: `clustPPPCA`



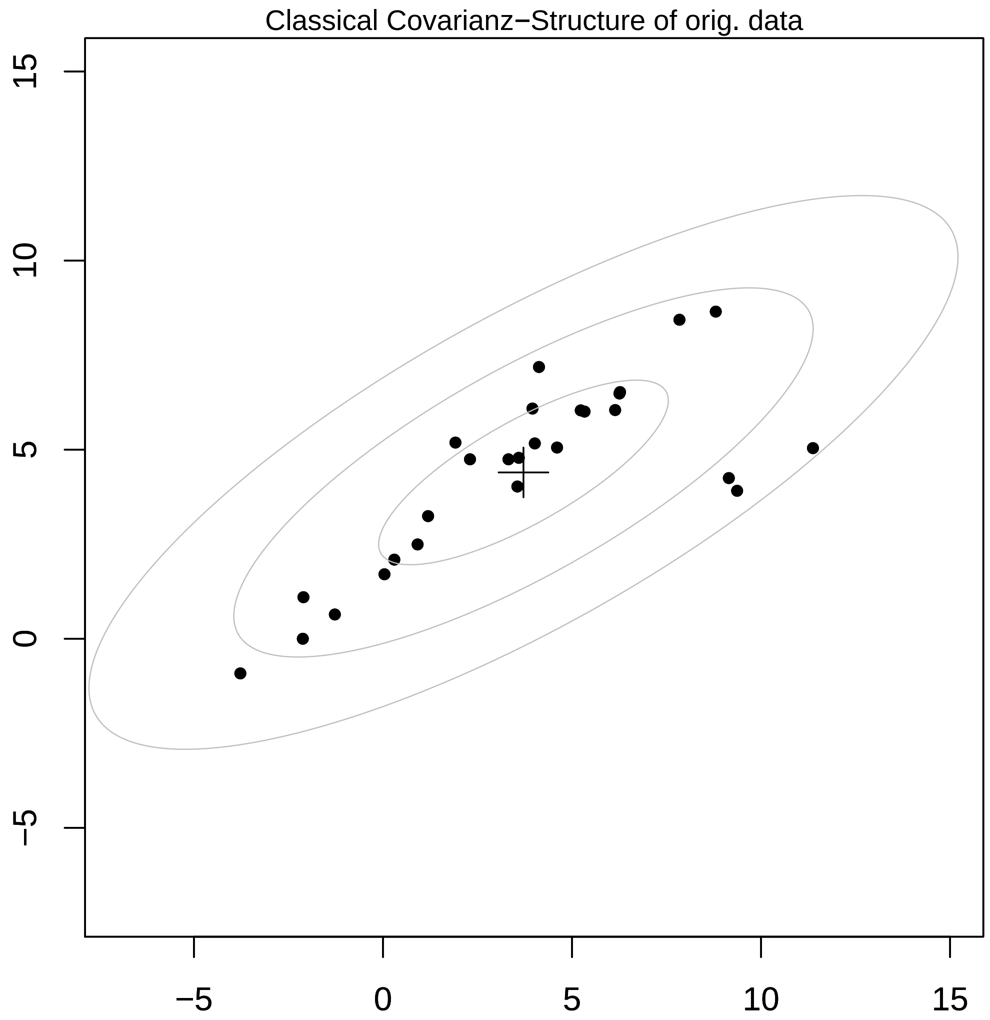
Algorithm: `clustPPPCA`



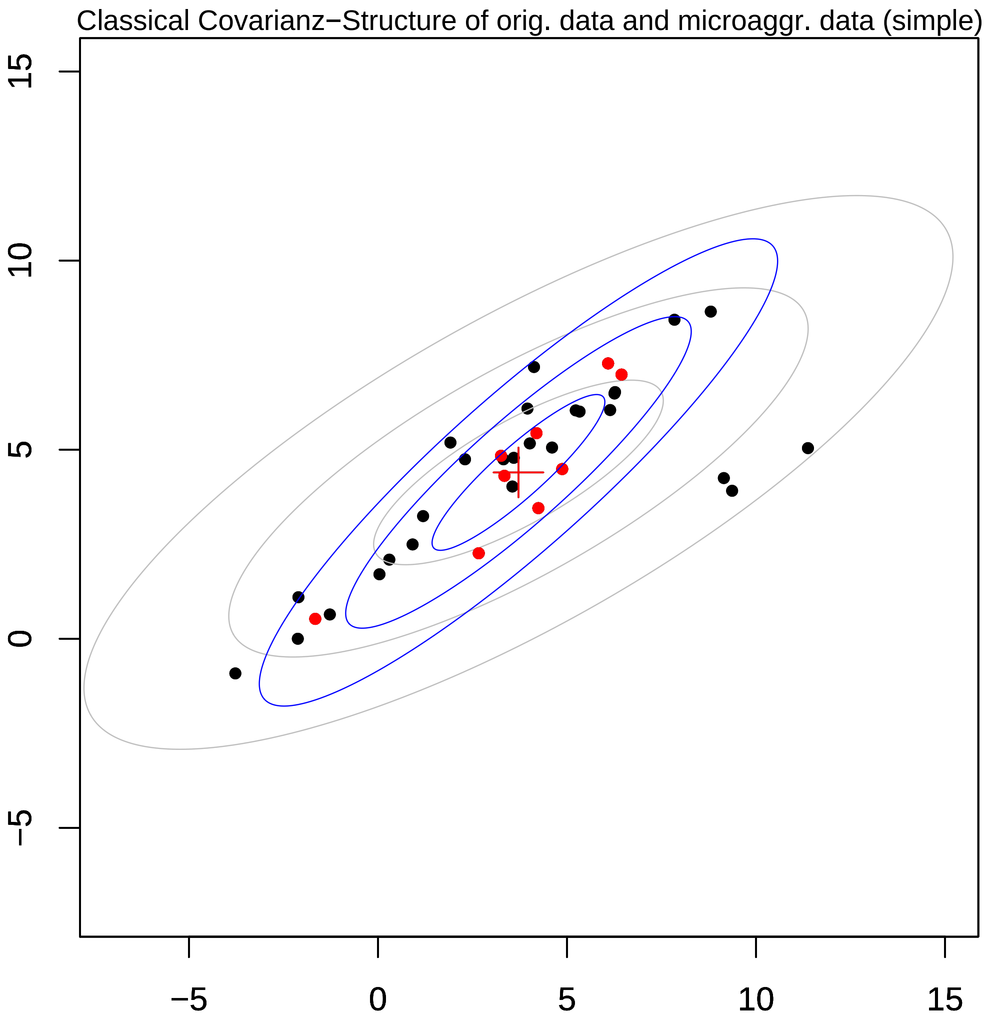
Algorithm: `clustPPPCA`

- To appear to software R in the near future.

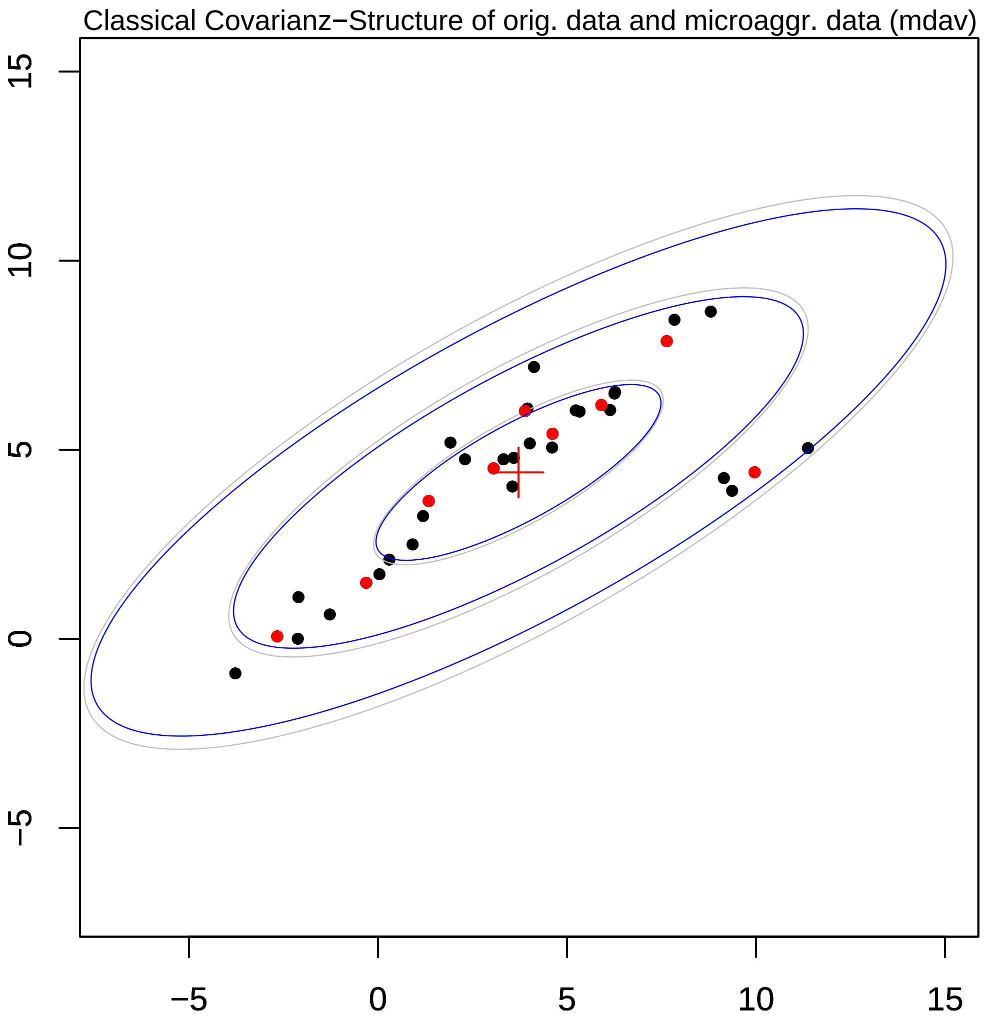
Multivariate Structures, Results



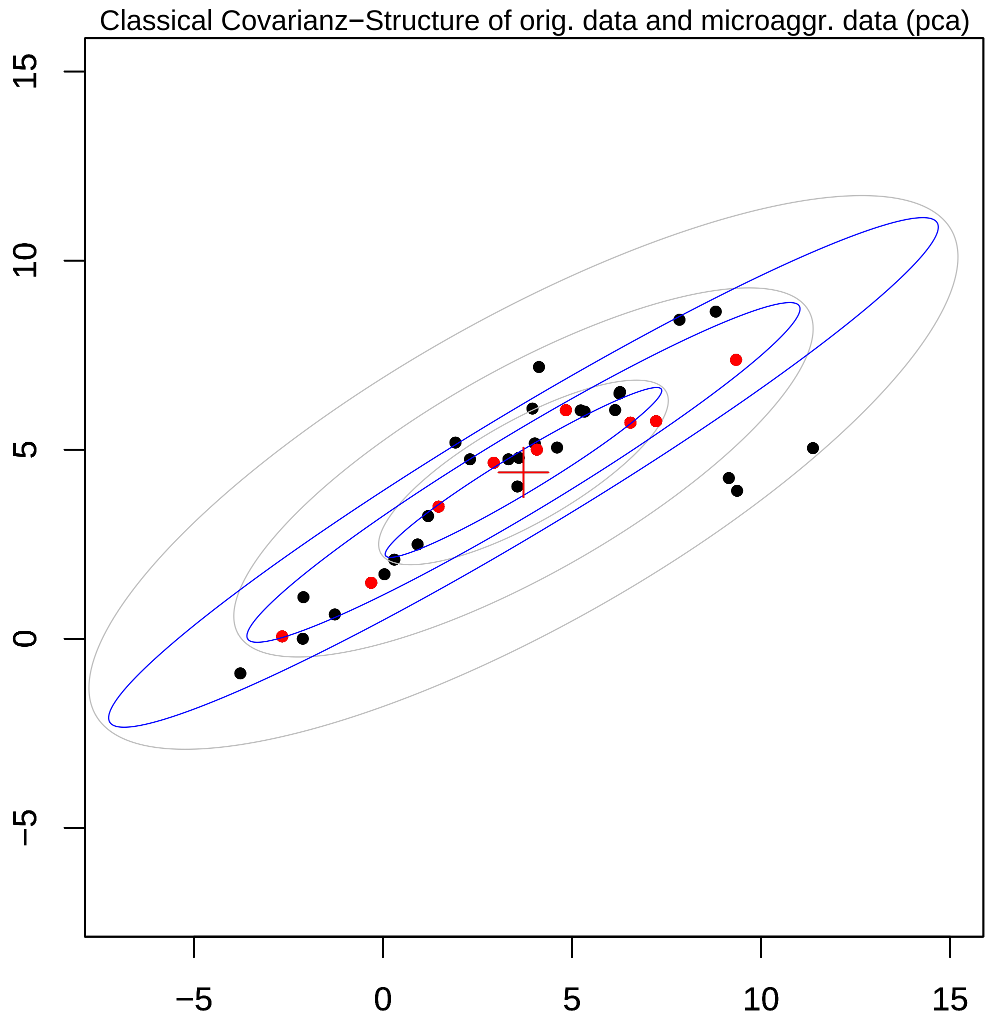
Multivariate Structures, Results



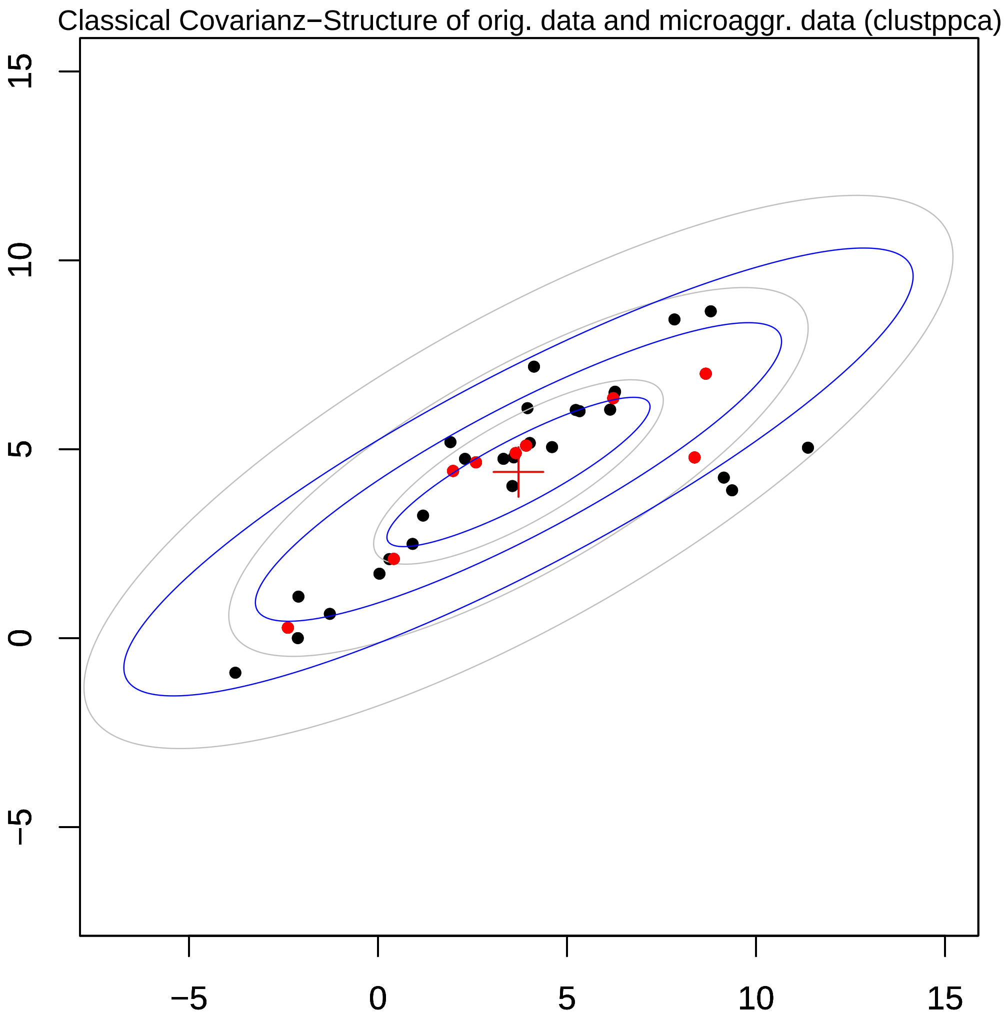
Multivariate Structures, Results



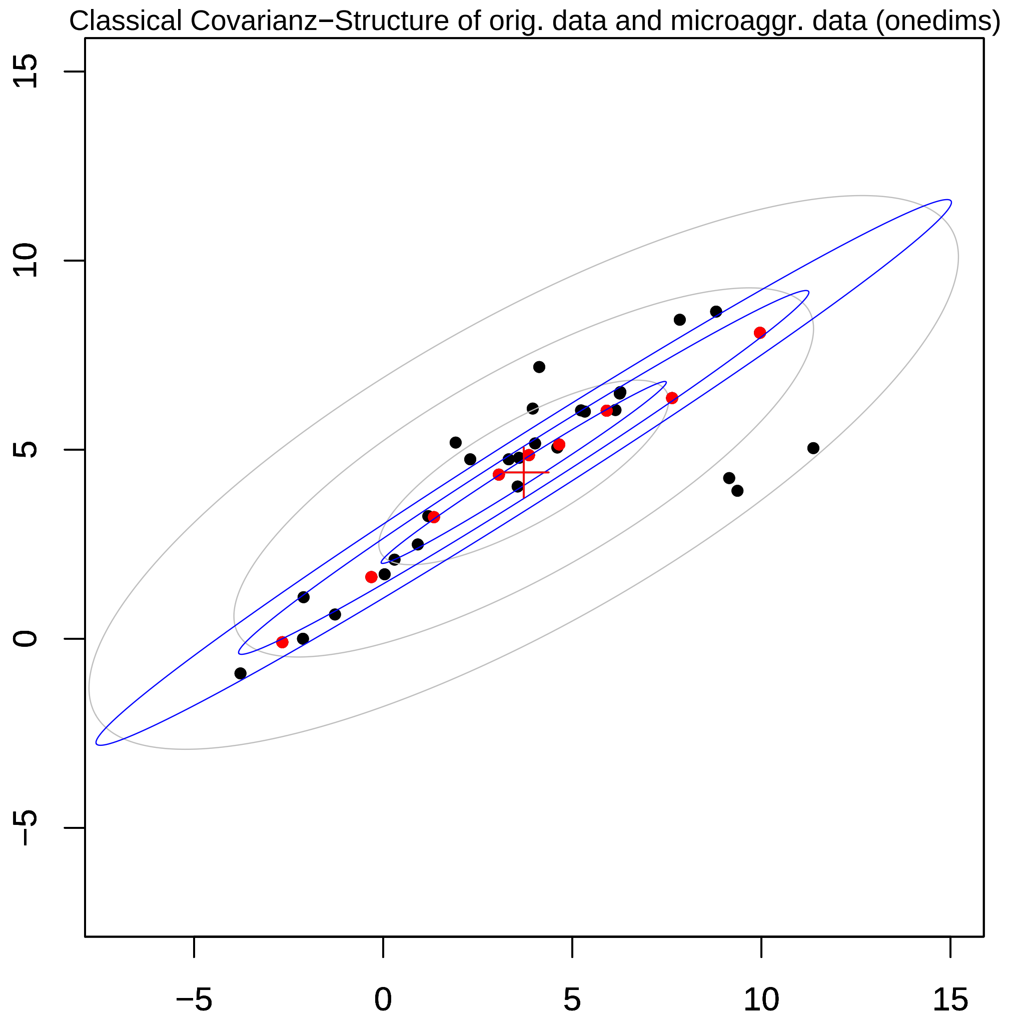
Multivariate Structures, Results



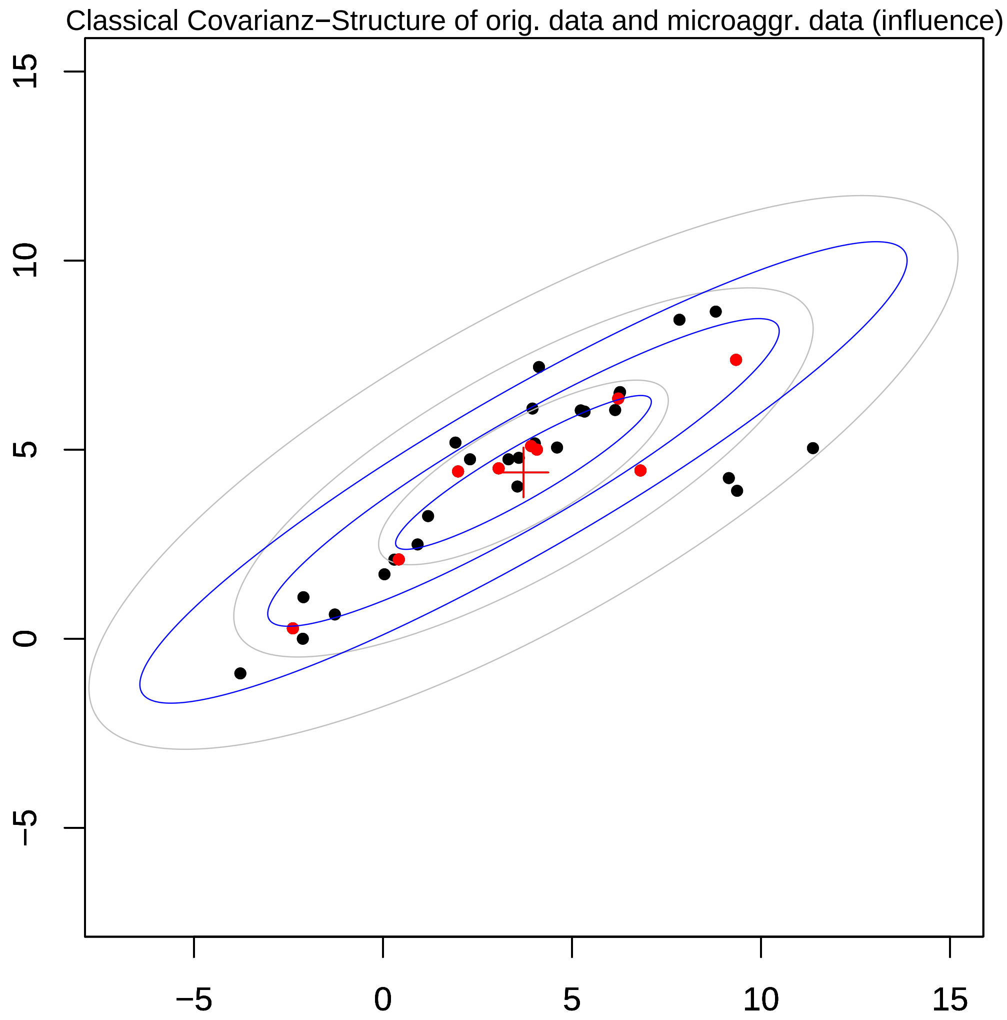
Multivariate Structures, Results



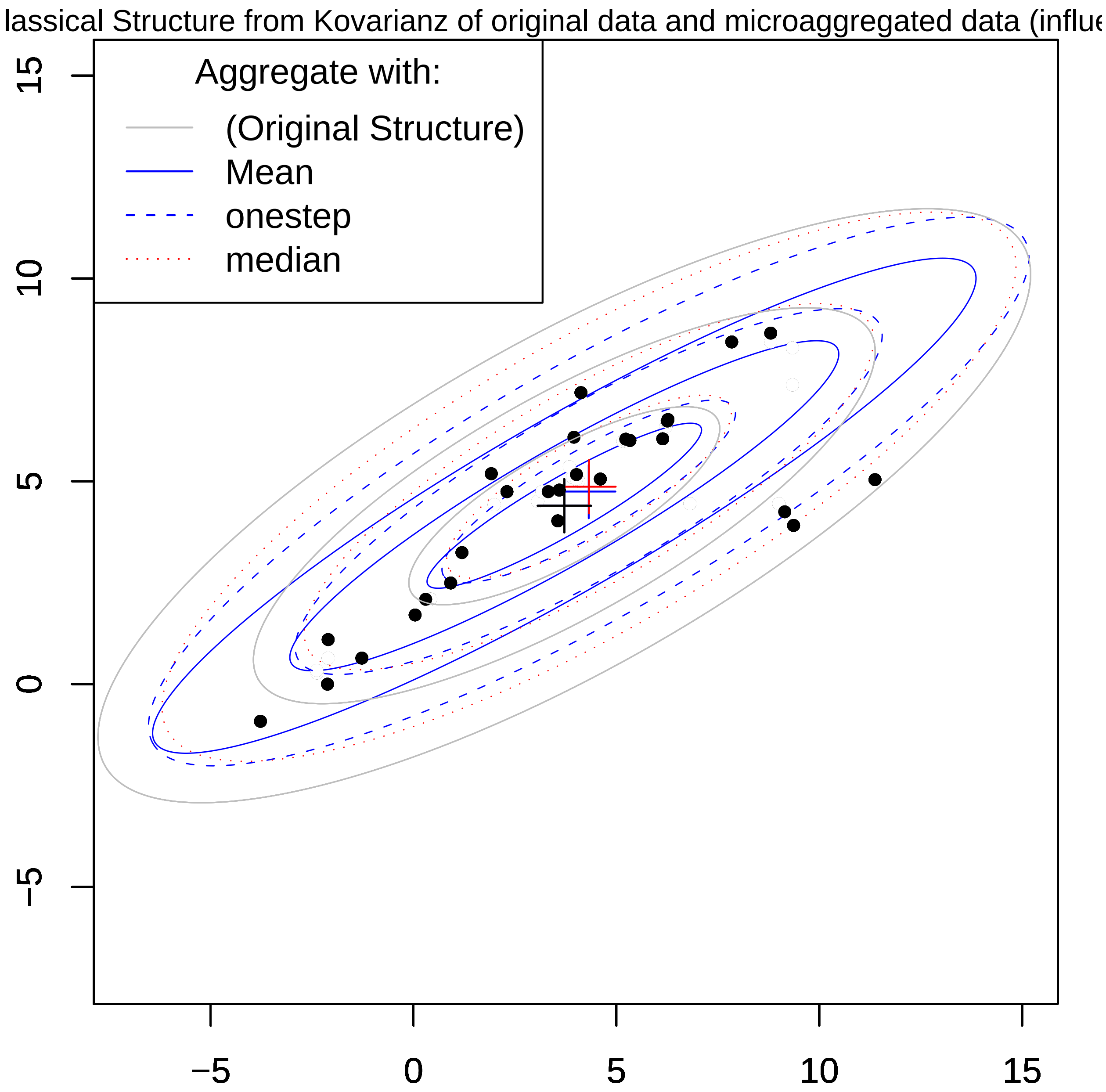
Multivariate Structures, Results



Multivariate Structures, Results



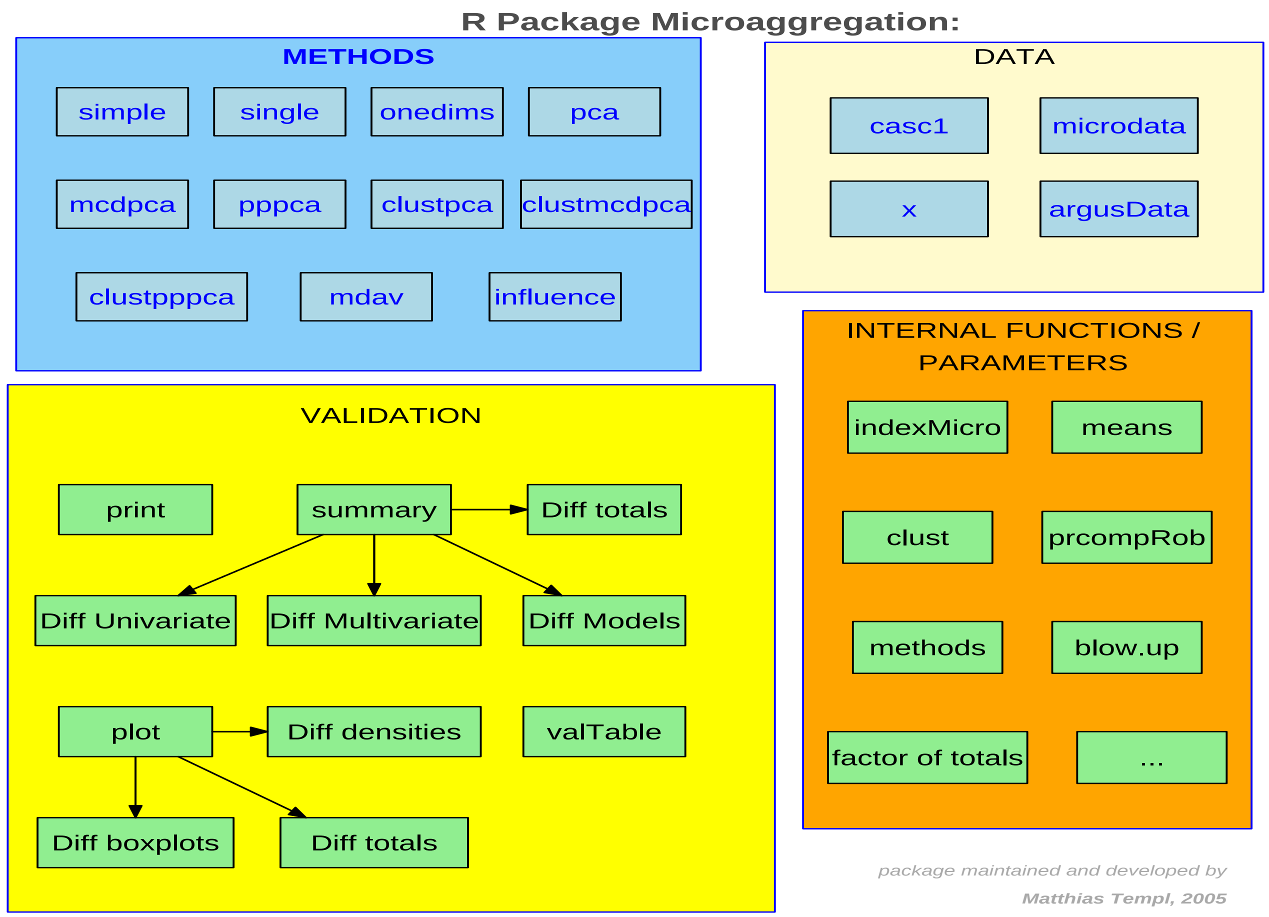
Multivariate Structures, Results



R-Package Microaggregation

10 Algorithms are implemented, the most important are:

- `mdav` (like μ-Argus)
- `clustppca`: Robustified PCA with Projection Pursuit on clustered data.


R Package Microaggregation:
METHODS
simple
single
onedims
pca
mcdpca
pppca
clustpca
clustmcdpca
clustpppca
mdav
influence
DATA
casc1
microdata
x
argusData
VALIDATION
print
summary
Diff totals
Diff Univariate
Diff Multivariate
Diff Models
plot
Diff densities
valTable
Diff boxplots
Diff totals
INTERNAL FUNCTIONS / PARAMETERS
indexMicro
means
clust
prcompRob
methods
blow.up
factor of totals
...
package maintained and developed by
Matthias Templ, 2005

Basic Routines

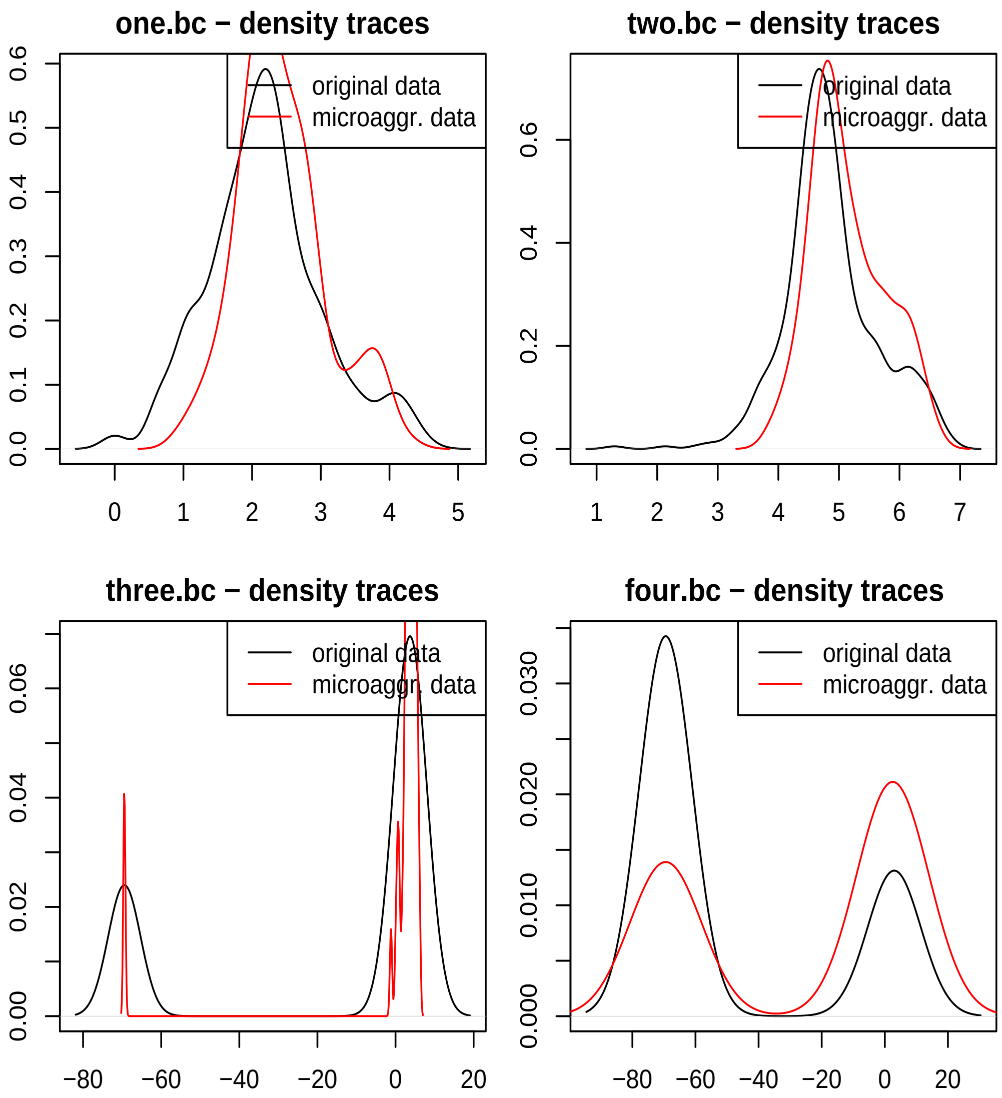
```
> args(microaggregation)

function (x, method = "pca", aggr = 3, nc = 8, clustermethod = "Mclust",
    opt = FALSE, measure = "mean", trim = 0, varsort = 1, transf = "log",
    blow = FALSE, blowxm = 0)
NULL

> args(summaryMicro)

function (x, robCov = TRUE, robReg = TRUE)
NULL

> args(plotMicro)

function (x, p, which.plot = 1:3)
NULL

> args(valTable)

function (x, method = c("simple", "single", "onedims", "pca",
    "pppca", "clustpca", "clustpppca", "mdav"), measure = "mean",
    clustermethod = "Mclust", aggr = 3, nc = 8, transf = "log")
NULL
```

Small Example

```
> load("/usr/lib64/R/library/Microaggregation/data/x.rda")
> w <- which(floor(x[, "nace"]/1000) == 151)
> x <- x[w, c(3, 5, 6, 7)]
> colnames(x) <- c("one", "two", "three", "four")
> m1 <- microaggregation(x, method = "simple")
> m2 <- microaggregation(x, method = "onedims")
> m3 <- microaggregation(x, method = "clustpppca", clustermethod = "Mclust")
> load("/home/templ/STAT/Templ/Matthias/Q2006/vortrag/m4.RData")
> library(disclosure)
```

```
>␣plotMicro2(m1,␣which.plot=1)
```

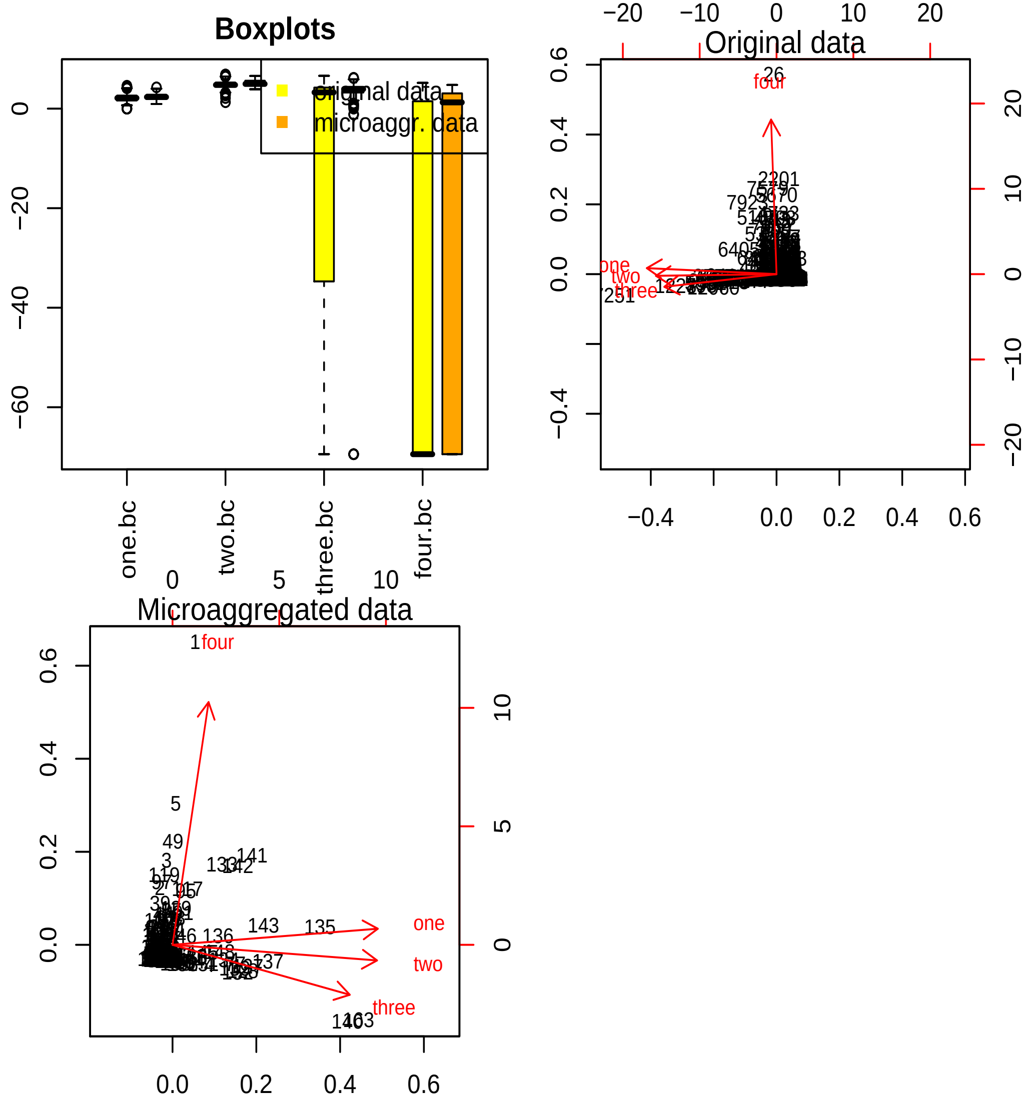
Small Example: Algorithm simple



Small Example: Algorithm simple

```
> plotMicro2(m1,which.plot=2)
```

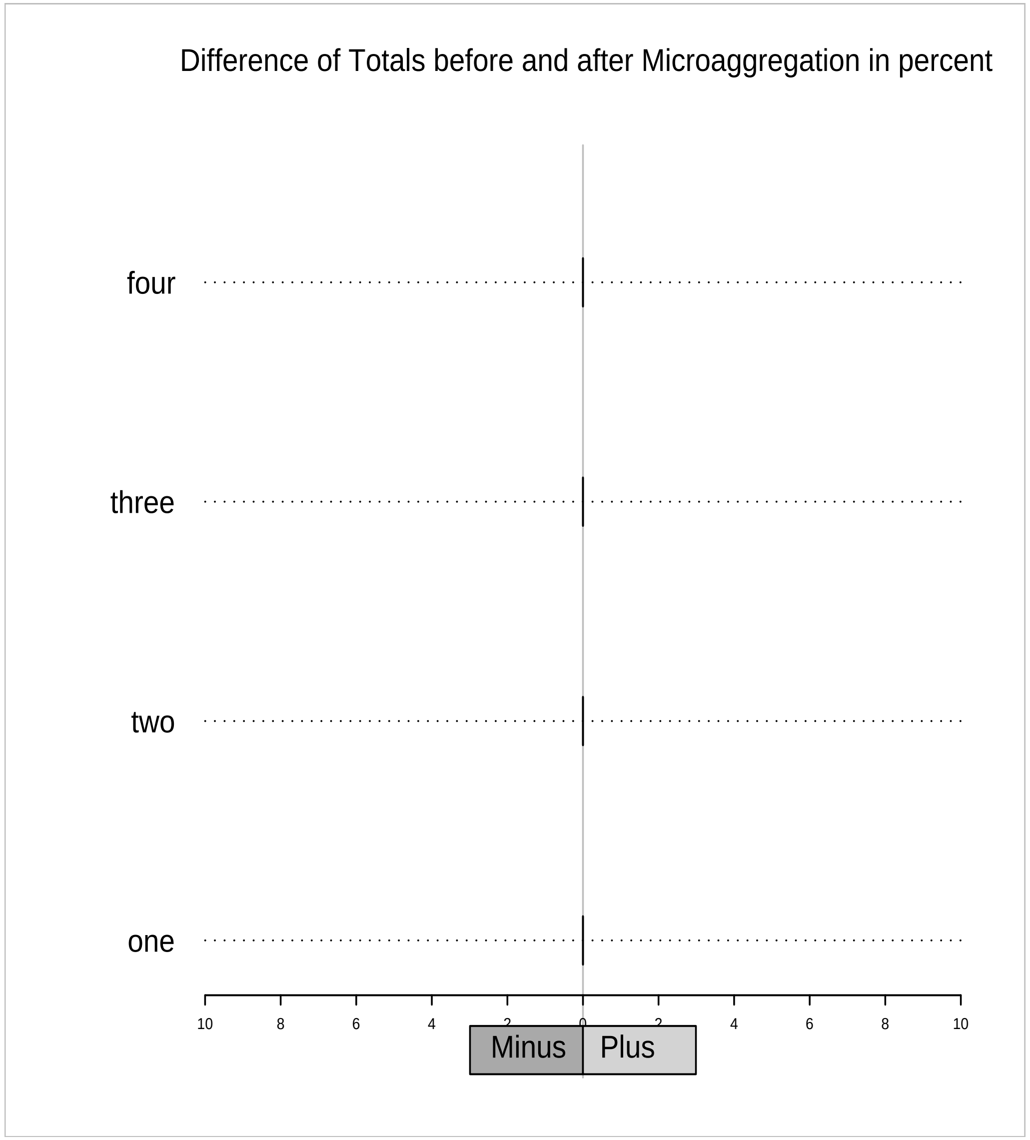
Small Example: Algorithm simple



Small Example: Algorithm simple

```
> plotMicro2(m1,which.plot=3)
```

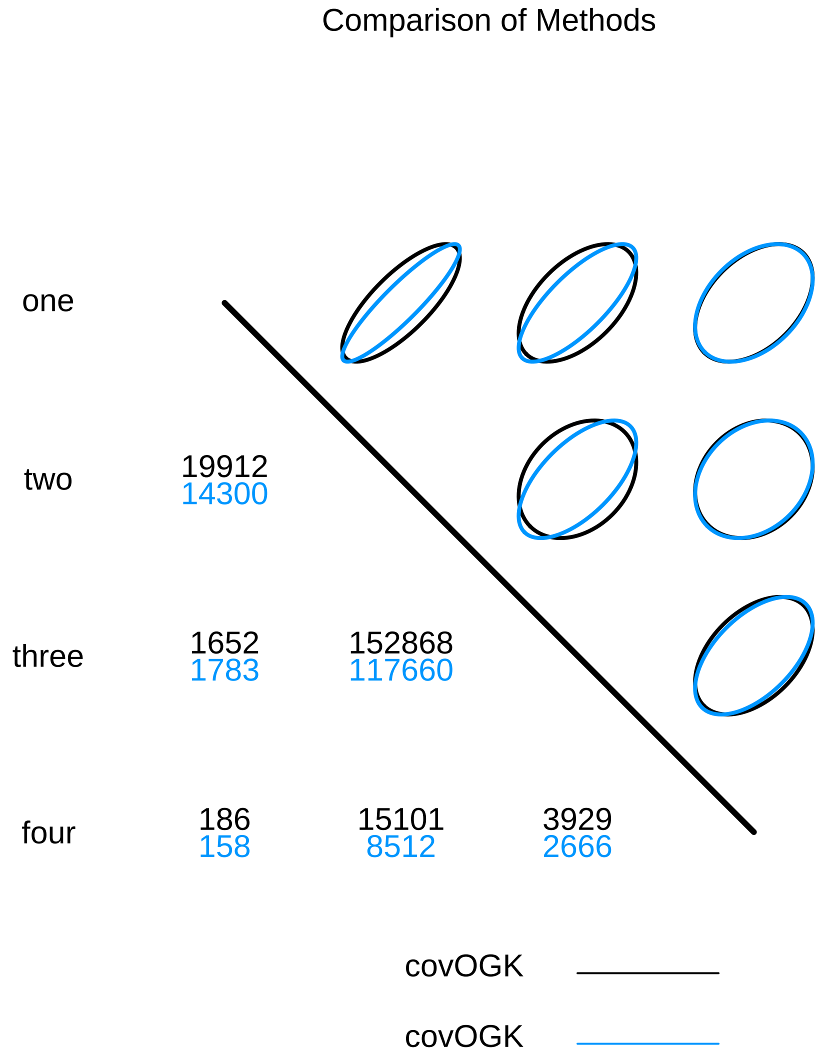
Small Example: Algorithm simple



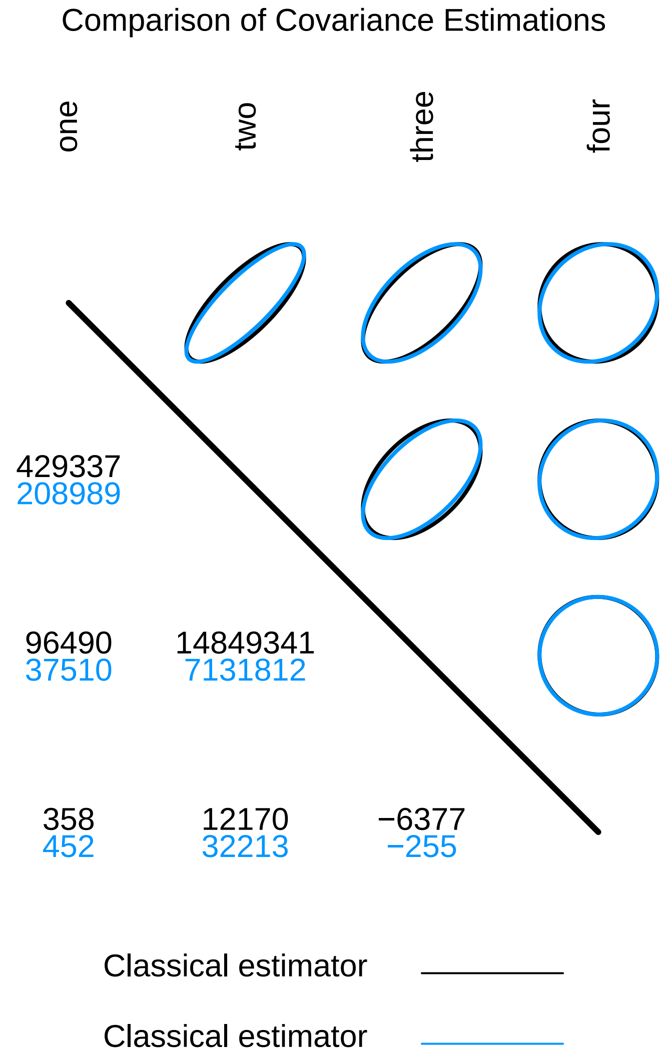
Small Example: Algorithm simple

```
Attaching package: 'robustbase'


        The following object(s) are masked from package:rrcov :

         covMcd covPlot ltsPlot ltsReg rrcov.control
```

Small Example: Algorithm simple



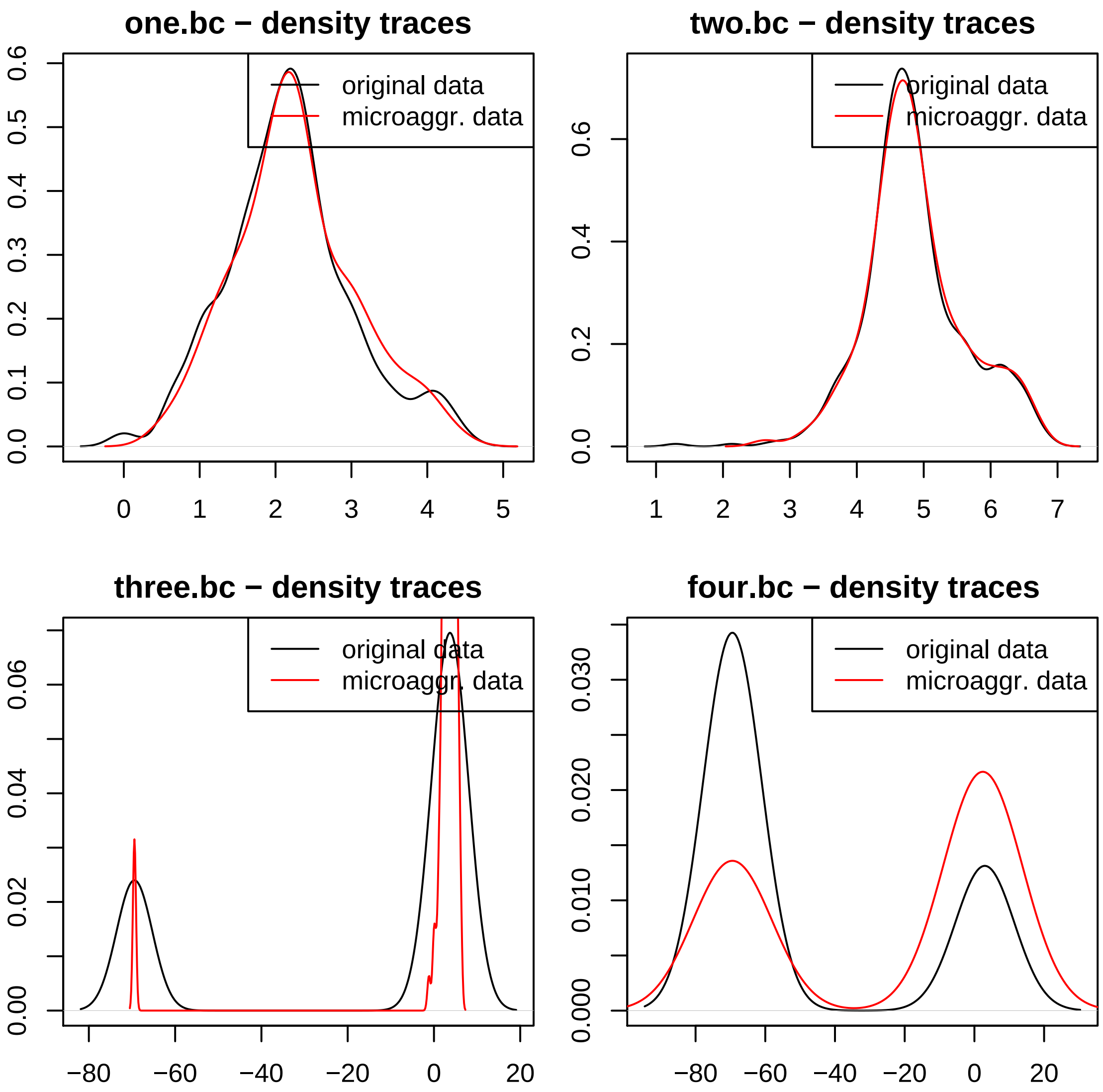
Small Example: Algorithm simple



Small Example: Algorithm clustpppca

```
> plotMicro2(m3,which.plot=1)
```

Small Example: Algorithm clustpppca



Measures of Information Loss

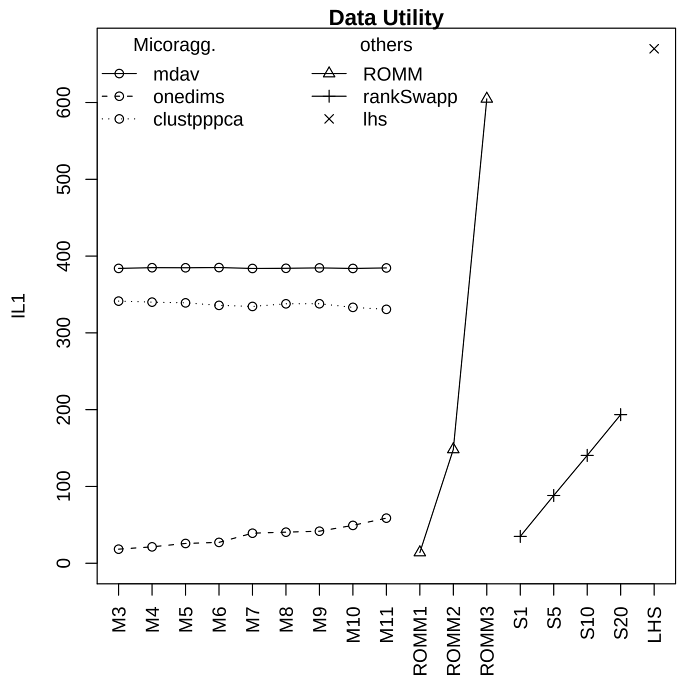
```
> data(x)
> w <- which(floor(x[, "nace"]/1000) == 151)
> x <- x[w, c(3, 5, 6, 7)]
> method <- c("simple", "single", "onedims", "pca", "pppca", "influence",
+     "clustpppca", "mdav")
> load("valTableOutput.RData")
```

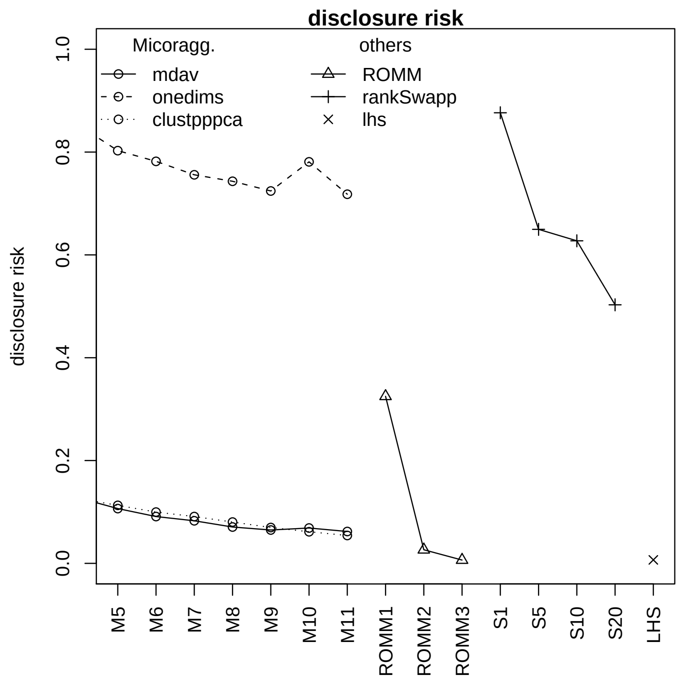
Measures of Information Loss

```
> print(g)

      method amean amedian aonestep devvar  amad   acov   acor acors  adlm
1     simple 1.809   0.538    0.186  2.859 0.582  1.430  0.606 0.424 0.012
2     single 0.995   0.322    0.301  2.933 0.318  1.466  2.573 0.698 0.017
3    onedims 0.100   0.007    0.004 66.549 0.007 33.274 39.750 7.259 0.194
4        pca 0.782   0.289    0.225  1.676 0.239  0.838  0.285 0.809 0.035
5      pppca 1.064   0.363    0.293  1.919 0.322  0.959  0.894 0.510 0.089
6  influence 0.943   0.207    0.231  2.169 0.289  1.084  1.107 0.572 0.041
7 clustpppca 0.996   0.226    0.161  2.025 0.229  1.013  1.577 0.400 0.057
8       mdav 1.225   0.267    0.226  3.131 0.364  1.566  7.022 0.430 0.016
  apcaload apppcaload atotals pmtotals
1    0.099      2.395   1.809   -1.809
2    0.051      1.283   0.995   -0.995
3    1.416      2.585   0.100   -0.100
4    0.101      4.063   0.782   -0.764
5    0.236      4.640   1.064   -1.058
6    0.222      0.523   0.943   -0.943
7    0.369      3.071   0.996   -0.932
8    0.288      2.736   1.225   -1.223
```

Disclosure Risk and Data Utility

Disclosure Risk and Data Utility





- **Data utility**: $IL1 = \frac{1}{d}\sum_{i=1}^{p} \frac{|x_{ij} - x'_{ij}|}{\sqrt{2}S_j}$
  where S_j is the standard deviation of the j-th variable in the original data
  Disadvantages: It does not evaluate how well univariate or multivariate statistics are preserved.

- **Disclosure Risk**: Given the value of a masked variable, check whether the corresponding original value falls within an interval centered on the masked value.
  Disadvantages: Assumes that an intruder has additional information (disclosure scenarios) so that one can link the masked record of an individual to its original version.

In my point of diagnostic methods shown in the last slides would be better.

S4-class R-Package AddNoise

```
Description:

Package:       AddNoise
Type:          Package
Title:         Adding Noise to data
Version:       1.0
Date:          2005-11-16
Author:        Matthias Templ
Maintainer:    Matthias Templ <Matthias.Templ@statistik.gv.at>
Depends:       R (>= 2.2.0), methods, Microaggregation, bootstrap, car,
               MASS, rrcov, far
Collate:       outdect.R addNoise.R addNoise2.R plot.outdect.R
               print.outdect.R print.addNoise.R print.addNoise2.R
               summary.outdect.R summary.addNoise.R summary.addNoise2.R
               comparePlot.R
SaveImage:     yes
LazyLoad:      yes
Description:   later
License:       GPL 2 or newer?
Built:         R 2.2.0; i386-pc-mingw32; 2005-11-21 12:23:11; windows
```

S4-class R-Package AddNoise

Index:

S4-class R-Package `AddNoise`

Package `AddNoise` includes very different methods.

Some Properties of the package are:

- Some Methods are implemented, amongst others ROMM (Ting and Fienberg, 2005) (Note: ROMM fulfills not all the functions of confidentiality).
- S4-Class style (define own classes!)
- System of error messages
- Flexible Package - User can include own code easily.
- print, summary and plot methods for all classes.

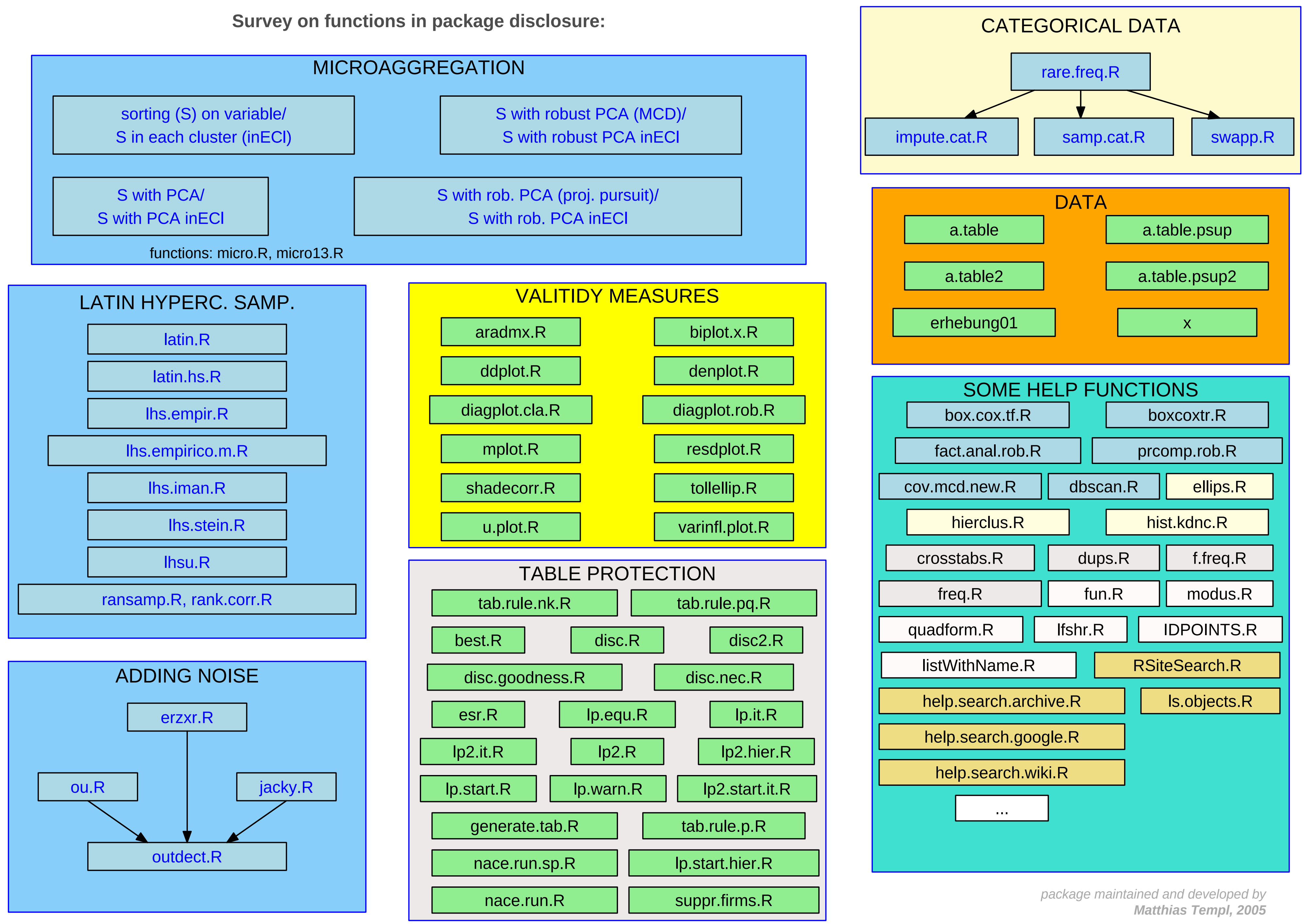
R-Package disclosure

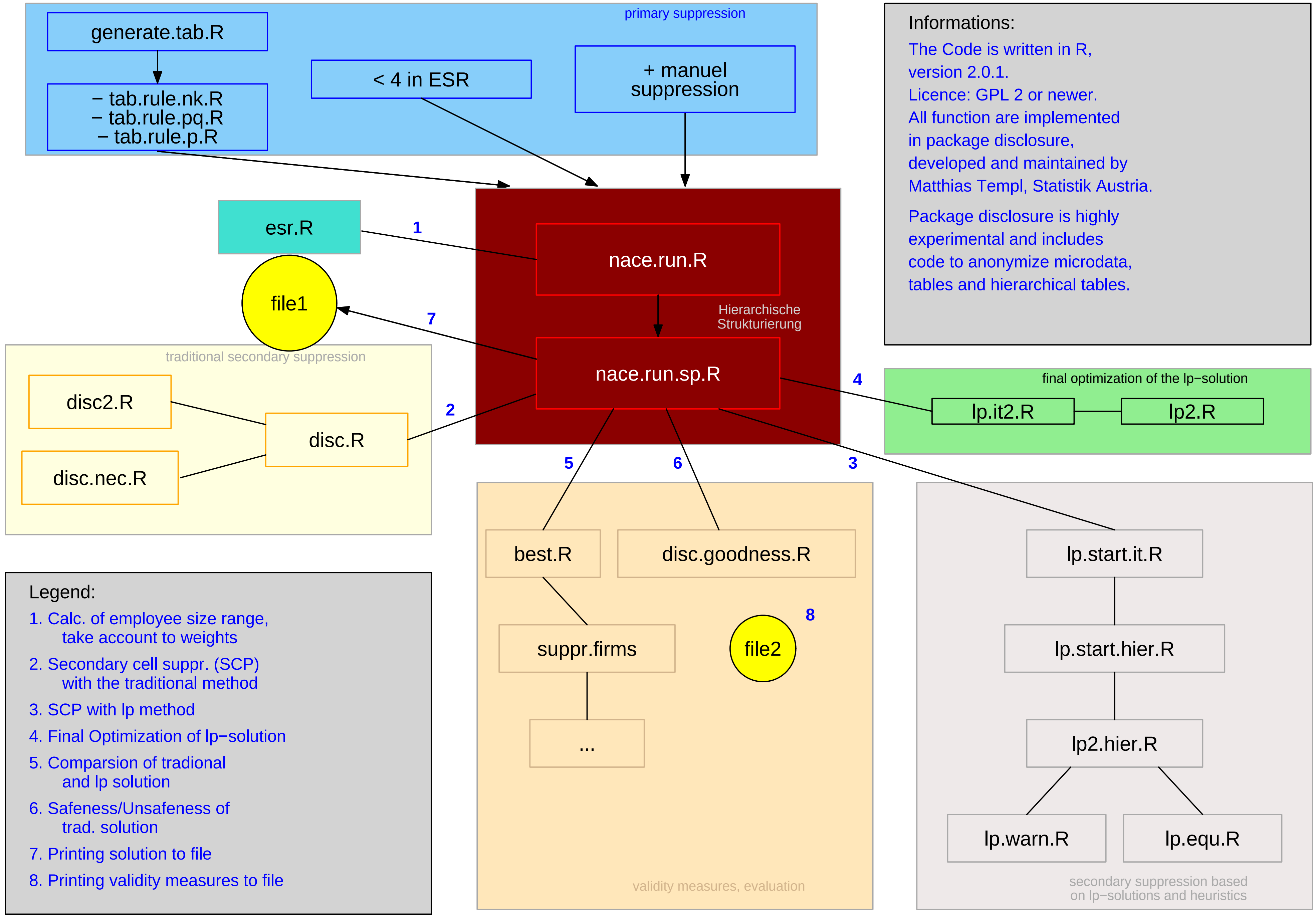
```
Description:

Package:       disclosure
Title:         data protection
Version:       2.0
Date:          2005-02-10
Author:        Matthias Templ <matthias.templ@statistik.gv.at>
Maintainer:    Matthias Templ <Matthias.templ@statistik.gv.at>
Depends:       R (>= 2.0.0), cluster, lqs, car, boot, bootstrap,
               lpSolve
Description:   Protection of sensible data
License:       GPL version 2 or newer
Packaged:      Mon Feb 14 17:08:49 2005; TEMPL$
Built:         R 2.1.0; i386-pc-mingw32; 2005-02-14 17:08:51; windows
```

More Packages

There are more packages:

- RankSwapp (rank swapping)
  reproducible (when specified), multivariate structure is destroyed.
- Latin Hypercube Sampling
  results are not satisfactory, even for iterations.
- Drisk
  (on an very early stage)
- disclosure
  Protecting (hierarchical) tables


Survey on functions in package disclosure:
MICROAGGREGATION
sorting (S) on variable/
S in each cluster (inECl)
S with robust PCA (MCD)/
S with robust PCA inECl
S with PCA/
S with PCA inECl
S with rob. PCA (proj. pursuit)/
S with rob. PCA inECl
functions: micro.R, micro13.R
CATEGORICAL DATA
rare.freq.R
impute.cat.R
samp.cat.R
swapp.R
DATA
a.table
a.table.psup
a.table2
a.table.psup2
erhebung01
x
LATIN HYPERC. SAMP.
latin.R
latin.hs.R
lhs.empir.R
lhs.empirico.m.R
lhs.iman.R
lhs.stein.R
lhsu.R
ransamp.R, rank.corr.R
VALITIDY MEASURES
aradmx.R
biplot.x.R
ddplot.R
denplot.R
diagplot.cla.R
diagplot.rob.R
mplot.R
resdplot.R
shadecorr.R
tollellip.R
u.plot.R
varinfl.plot.R
SOME HELP FUNCTIONS
box.cox.tf.R
boxcoxtr.R
fact.anal.rob.R
prcomp.rob.R
cov.mcd.new.R
dbscan.R
ellips.R
hierclus.R
hist.kdnc.R
crosstabs.R
dups.R
f.freq.R
freq.R
fun.R
modus.R
quadform.R
lfshr.R
IDPOINTS.R
listWithName.R
RSiteSearch.R
help.search.archive.R
ls.objects.R
help.search.google.R
help.search.wiki.R
...
TABLE PROTECTION
tab.rule.nk.R
tab.rule.pq.R
best.R
disc.R
disc2.R
disc.goodness.R
disc.nec.R
esr.R
lp.equ.R
lp.it.R
lp2.it.R
lp2.R
lp2.hier.R
lp.start.R
lp.warn.R
lp2.start.it.R
generate.tab.R
tab.rule.p.R
nace.run.sp.R
lp.start.hier.R
nace.run.R
suppr.firms.R
ADDING NOISE
erzxr.R
ou.R
jacky.R
outdect.R
package maintained and developed by
Matthias Templ, 2005


primary suppression
generate.tab.R
− tab.rule.nk.R
− tab.rule.pq.R
− tab.rule.p.R
< 4 in ESR
+ manuel suppression
Informations:
The Code is written in R,
version 2.0.1.
Licence: GPL 2 or newer.
All function are implemented
in package disclosure,
developed and maintained by
Matthias Templ, Statistik Austria.
Package disclosure is highly
experimental and includes
code to anonymize microdata,
tables and hierarchical tables.
esr.R
1
nace.run.R
file1
7
Hierarchische Strukturierung
nace.run.sp.R
traditional secondary suppression
disc2.R
disc.R
disc.nec.R
2
4
final optimization of the lp−solution
lp.it2.R
lp2.R
5
6
3
best.R
disc.goodness.R
suppr.firms
file2
8
...
validity measures, evaluation
lp.start.it.R
lp.start.hier.R
lp2.hier.R
lp.warn.R
lp.equ.R
secondary suppression based on lp−solutions and heuristics
Legend:
1. Calc. of employee size range, take account to weights
2. Secondary cell suppr. (SCP) with the traditional method
3. SCP with lp method
4. Final Optimization of lp−solution
5. Comparsion of tradional and lp solution
6. Safeness/Unsafeness of trad. solution
7. Printing solution to file
8. Printing validity measures to file

Anonymizing Tables

- Starting point: microdata
- Calculating marginal tables from microdata:

	1	2	3	sum
1	20	50	10	80
2	8	19	22	49
3	17	32	12	61
sum	45	101	44	190

Anonymizing Tables

- Starting point: microdata
- Calculating marginal tables from microdata:
- We know e.g. the turnover of a enterprise, if too less enterprises contributes to a cell

	1	2	3	sum
1	20	50	10	80
2	8	19	22	49
3	17	32	12	61
sum	45	101	44	190

Anonymizing Tables

- Starting point: microdata
- Calculating marginal tables from microdata:
- We know e.g. the turnover of a enterprise, if too less enterprises contributes to a cell
  → primary suppression (or perturbation) of these cells

	1	2	3	sum
1	20	50	10	80
2	NA	19	22	49
3	17	32	12	61
sum	45	101	44	190

Anonymizing Tables

- Starting point: microdata
- Calculating marginal tables from microdata:
- We know e.g. the turnover of a enterprise, if too less enterprises contributes to a cell
  $\longrightarrow$ primary suppression (or perturbation) of these cells
- Cell value could be recalculated (49 - 19 - 22 = 8)

	1	2	3	sum
1	20	50	10	80
2	NA	19	22	49
3	17	32	12	61
sum	45	101	44	190

Anonymizing Tables

- Starting point: microdata
- Calculating marginal tables from microdata:
- We know e.g. the turnover of a enterprise, if too less enterprises contributes to a cell
  $\longrightarrow$ primary suppression (or perturbation) of these cells
- Cell value could be recalculated (49 - 19 - 22 = 8)
  $\longrightarrow$ Secondary cell suppression.

	1	2	3	sum
1	20	50	10	80
2	NA	19	NA	49
3	NA	32	NA	61
sum	45	101	44	190

Anonymizing Tables

- Becomes much more complicated and NP-hard in case of hierarchical tables (and also harder for linked tables):

	1	2	3	sum
21	20	50	10	80
22	NA	19	NA	49
23	NA	32	NA	61
sum	45	101	44	190

	1	2	3	sum
221	6	5	10	21
222	NA	NA	10	16
223	NA	NA	11	12
22	NA	19	NA	49

AnonymizingTables

There are many different methods for anonymizing tables (rounding of cell values, . . .), but for legal reasons only cell suppression is possible in most of the countries.

There are different primary cell suppression rules (easy) and different secondary cell suppression methods (hard).

The most powerful (and most complex) methods based on **linear programming**.

Methods based on LP

Methods and Software:

- Primary cell suppression methods
- Secondary cell suppression methods based on linear Programming.

  Aim is to minimise the amount of suppressed cells in (hierarchical) tables under the constraint that each primary suppressed cell can be computed only on a ppredefinediinterval(can not be computed exactly enough).
- `lpSolve`

  is a freely available (under LGPL 2) software for solving linear, integer and mixed integer programs. `R` supply a *wrapper* function in `C` and some `R` functions that solve general linear/integer problems, assignment problems, and transportation problems.

Unsafe Cells for non-LP methods

```
> x <- xp <- as.matrix(data.frame(esr1 = c(5168, 442, 3092, 457,
+     9159), esr2 = c(750, 60, 200, 67, 1077), esr3 = c(415, 32,
+     59, 78, 584), esr4 = c(134, 10, 10, 54, 208), esr = c(9,
+     0, 1, 7, 17), sum = c(6476, 544, 3362, 663, 11045)))
> xp[which(xp < 4 & xp > 0, arr.ind = TRUE)] <- NA
> xp[5, 5] <- NA
> xp
```

```
  esr1 esr2 esr3 esr4 esr   sum
1 5168  750  415  134   9  6476
2  442   60   32   10   0   544
3 3092  200   59   10  NA  3362
4  457   67   78   54   7   663
5 9159 1077  584  208  NA 11045
```

Unsafe Cells

Hypercube solution:

```
> xsHC <- disc(xp)
> xsHC

  esr1 esr2 esr3 esr4 esr   sum
1 5168  750  415  134   9  6476
2  442   60   32   10   0   544
3 3092  200   59   NA  NA  3362
4  457   67   78   54   7   663
5 9159 1077  584   NA  NA 11045

> suppressWarnings(lp2.hier(x, xsHC)$lp.out2)

     min max true nrow ncol ind
[1,]   0  11   10    3    4   0
[2,]   0  11    1    3    5   0
[3,] 198 209 -208    5    4   1
[4,]  16  27  -17    5    5   0
```

Too close! We need suppressions based on linear programming.

LP

	1	2	3	sum
1	20	50	10	80
2	8	19	22	49
3	17	32	12	61
sum	45	101	44	190

	1	2	3	sum
1	20	50	10	80
2	NA	19	NA	49
3	17	32	NA	61
sum	45	101	44	190

	1	2	3	4	5	6	7	8	9
1	0	0	0	0	0	0	0	0	0
2	0	0	0	1	0	1	0	0	0
3	0	0	0	0	0	0	0	0	1
4	0	0	0	1	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	1	0	0	1

```
> suppressWarnings(lp2(a.table, a.table.psup)["l.r

1 2 3 1 2 3

0 30 12 8 0 34

> suppressWarnings(lp2(a.table, a.table.psup)["lp.

min max true nrow ncol ind

[1,] 8 8 8 2 1 1

[2,] 22 22 22 2 3 1

[3,] 12 12 12 3 3 1
```

LP

With cell suppression by LP we must fulfill

- Minimize, e.g. the amount of suppressed cells in hierarchicalables.
- Each primary suppressed cell must be protected enough, i.e. one should not be able to calculate the cell value too accurately in an pre-defined interval).
- fast computation (less than 3 days?), i.e. finding a good local optimum.

LP

Remarks:

- Simulations based on Structural Business Data have shown that nearly 2.5 Percent of primary suppressed cells are not safe enough (can be estimated too accurately) with methods without LP.

- All Attackers have a powerful, easy manageable tool to disclose tables.

Conclusion

Using R for statistical disclosure control has many advantages:

- Flexibility in Data Import/Export facilities

  (note: fixed format data files resulting from punching cards)

- A powerful system for analysing results (graphically) and implementing methods.
- Making business microdata confidential by minimal modification of the data structure as a kind of explorative data analysis.

  Several methods have to be evaluated on basis of your data and diagnostic tools have to be used at the same time.

- The packages are highly extendable and can be well documented by use of online-help pages, vignettes and integrated examples.
- Everybody can evaluate errors, because all is open source.

Conclusion

- All Attackers have a powerful, easy manageable tool to disclose tables.

  That is the disadvantage.

- Mixed linear integer programming on hierarchical tables is in general a difficult problem

  but this problem is solvable. (general solution: maybe in some years)

- When calculation time is important, all the code can be written in C or Fortran and can be included in an R package easily.

- All people are invited to contribute to this packages or to make own R packages for statistical disclosure control.

Thank you

for your attention

matthias.templ(at)statistik.gv.at