Robust Principal Component and Discriminant Analysis in R

V. Todorov[1]

[1]United Nations Industrial Development Organization (UNIDO)

ICORS 2020

Vienna, 20-24 September, 2021

Outline

Outline

Outliers and robustness

- Many multivariate data sets contain outliers

Outlier...s

Hawkins, D. (1980) An outlier is an observation that deviates so much from other observations as to arose suspicion that it was generated by different mechanism

- More often we find outliers in large data sets (many observations or variables).
- Often they do not show up by simple visual inspection.

Outliers and Robustness

Outlier detection and Robust estimation are closely related

1. Robust estimation: find an estimate which is not influenced by the presence of outliers in the sample. Robustness is "... insensitivity against small deviations from the assumptions" (Huber, 1987)
2. Outlier detection: find all outliers, which could distort the estimate

- If we have a solution to the first problem we can identify the outliers using robust residuals or distances
- If we know the outliers we can remove or downweight them and use classical estimation methods
- It depends on the particular research, on which problem to set the focus

Multivariate Location and Scatter

- The usual multivariate analysis techniques are based on empirical means and covariance/correlation matrices and least squares fitting.
  - Location: coordinate-wise mean
  - Scatter: covariance matrix
    - Variances of the variables on the diagonal
    - Covariance of two variables as off-diagonal elements
  - Optimally estimated by the sample mean and sample covariance matrix at any multivariate normal model
  - Essential to a number of multivariate data analysis methods
- All these are extremely sensitive to outlying observations.

MCD Estimator

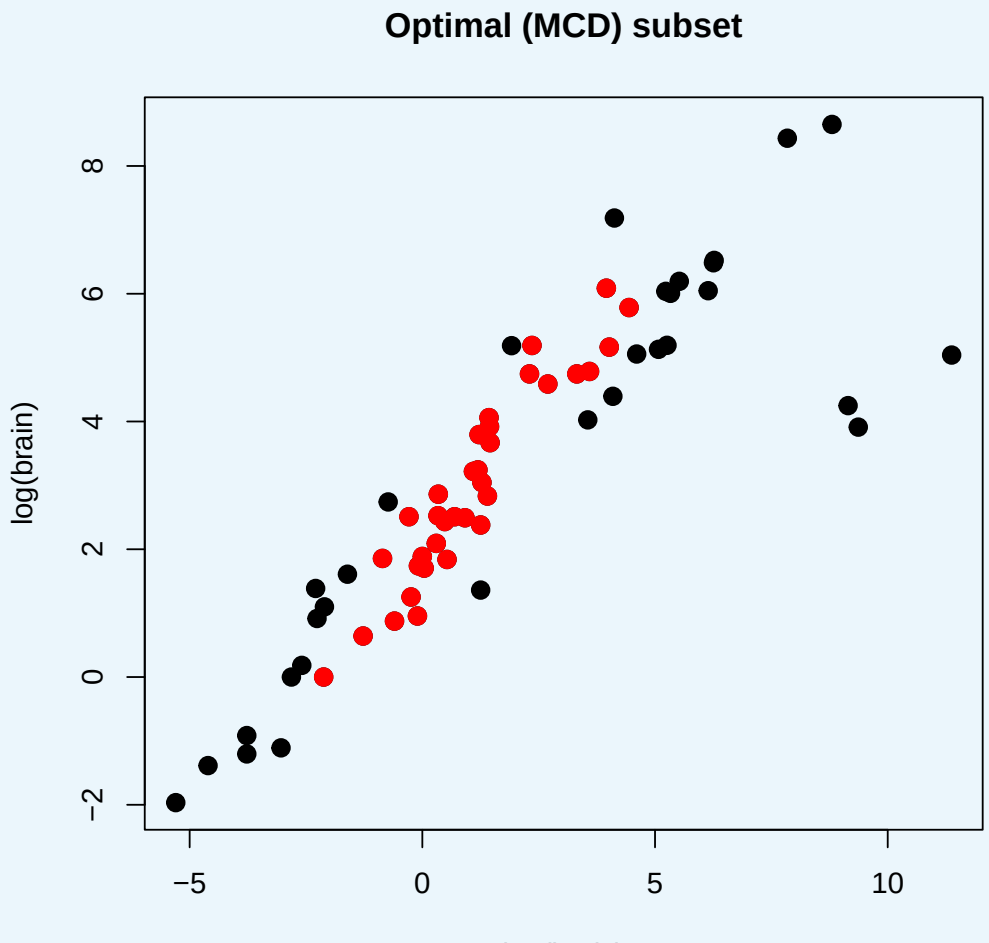
MCD-Estimator - *Minimum Covariance Determinant* (Rousseeuw, 1985)

- Find the subset of h observations out of n whose classical covariance matrix has a smallest determinant
- The MCD location estimator $\mathbf{T}$ is defined by the mean of that subset and the MCD scatter estimator $\mathbf{C}$ is a multiple of its covariance matrix.
- $h = \frac{(n+p+1)}{2}$ yields maximal breakdown point
- Fast algorithm to compute the MCD - (Rousseeuw and Van Driessen, 1999)

MVE-Estimator - *Minimum Volume Ellipsoid* (Rousseeuw, 1985)
- looks for the minimal volume ellipsoid covering at least half of the points

Example

Brain and body weights for 62 species of land animals (data set `Animals2` from package **robustbase**)



About the software

1. **robustbase** 0.93-8: "Essential" Robust Statistics. The goal is to provide tools allowing to analyze data with robust methods. This includes regression methodology including model selections and multivariate statistics where we strive to cover the book "Robust Statistics, Theory and Methods" by Maronna, Martin and Yohai; Wiley 2006.
2. **rrcov** 1.6-0: An object oriented framework for robust multivariate analysis providing a solid base of robust methods for estimation of multivariate location and covariance matrix as well as reference implementation of some basic robust multivariate methods like Principal Component Analysis (PCA), Discriminant analysis (LDA and QDA) and Multivariate tests

Todorov V and Filzmoser P (2009), An Object Oriented Framework for Robust Multivariate Analysis. *Journal of Statistical Software*, **32**(3), 1–47. `http://www.jstatsoft.org/v32/i03/`.

Outline

Principal component analysis as an exploratory tool

- Nowadays in many disciplines large data sets dominate the research. In many cases p is very large and even $p \gg n$. In order to interpret such data sets, methods like PCA are required to reduce their dimensionality without losing too much information.
- Many techniques exists that provide such service, but PCA is one of the oldest (Pearson (1901) and Hotelling (1933)) and most widely used.
- PCA is used in many different disciplines, many times reinvented, known under many different names:
  - Empirical orthogonal functions (EOF) in meteorology and climatology
  - Karhunen-Loeve Transform in digital signal processing
  - Proper orthogonal decomposition (POD) in fluid physics

Principal component analysis as an exploratory tool

- Principal component analysis (PCA), as an exploratory data analysis tool, has the purpose to describe the information contained in a set of correlated variables $X_1, X_2, \ldots, X_p$ in terms of a new set of uncorrelated variables $Z_1, Z_2, \ldots, Z_p$.
- These input variables define p n-dimensional vectors or, equivalently, an $n \times p$ data matrix $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$, whose i-th row is the vector (observation) $\mathbf{x}_i$ with $\mathbf{x}_i \in \mathbb{R}_p$.
- We assume that the input variables are continuous.

Classical PCA

- Find a k-dimensional subspace of $\mathbb{R}_p$ (with $k \leq min(n, p)$) such that the projection of the data on this subspace contains most of the information of the original p-dimensional data.
- "Most of the information" here means that the projection of the data retains as much of the variance as possible.
- We thus search for a center $\hat{\boldsymbol{\mu}}$ and a loading matrix $\mathbf{P}_{p,k}$ of size $p \times k$, such that the k-dimensional scores

$$\mathbf{t}_i = \mathbf{P}^\top(\mathbf{x}_i - \hat{\boldsymbol{\mu}}), i = 1, \ldots, n$$

are most informative (maximize the retained variance).

Classical PCA

- Thus, PCA searches the directions of maximum variability of the data.
- To compute the loading matrix

$$\mathbf{P}_{p,k} = \left[\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_k\right]$$

the first column is chosen as

$$\mathbf{p}_1 = \underset{||p||=1}{\arg\max}\, \mathrm{VAR}(\mathbf{p}^\top(\mathbf{x}_1 - \bar{\mathbf{x}}), \mathbf{p}^\top(\mathbf{x}_2 - \bar{\mathbf{x}}), \ldots \mathbf{p}^\top(\mathbf{x}_n - \bar{\mathbf{x}}))$$

- All following columns are chosen as

$$\mathbf{p}_{j+1} = \underset{||\mathbf{p}||=1, \mathbf{p}\perp\mathbf{p}_1, \ldots, \mathbf{p}\perp\mathbf{p}_j}{\arg\max}\, \mathrm{VAR}(\mathbf{p}^\top(\mathbf{x}_1 - \bar{\mathbf{x}}), \mathbf{p}^\top(\mathbf{x}_2 - \bar{\mathbf{x}}), \ldots \mathbf{p}^\top(\mathbf{x}_n - \bar{\mathbf{x}}))$$

Classical PCA

- This is a maximization problem.
- The solution of this maximization problem yields the loading matrix as the matrix containing the k dominant eigenvectors of the covariance matrix $\mathbf{S}_n$ of the data points.
- In particular, the spectral decomposition of $\mathbf{S}_n$ yields

$$\mathbf{S}_n = \mathbf{PLP}^\top$$

- Here
  - $\mathbf{P}$ is the $p \times p$ orthogonal matrix containing all p eigenvectors of $\mathbf{S}_n$
  - $\mathbf{L}$ is the the diagonal matrix with the p eigenvalues $l_1, \ldots, l_p$ in decreasing order.

Classical PCA

1. The classical PCA loading matrix is the matrix $\mathbf{P}_{p,k}$ which contains the first k columns of $\mathbf{P}$.
2. The eigenvalues l_j equal

$$l_j = \text{VAR}(\mathbf{p}_j^\top(\mathbf{x}_1 - \bar{\mathbf{x}}), \mathbf{p}_j^\top(\mathbf{x}_2 - \bar{\mathbf{x}}), \ldots \mathbf{p}_j^\top(\mathbf{x}_n - \bar{\mathbf{x}}))$$

3. The proportion of variance explained:

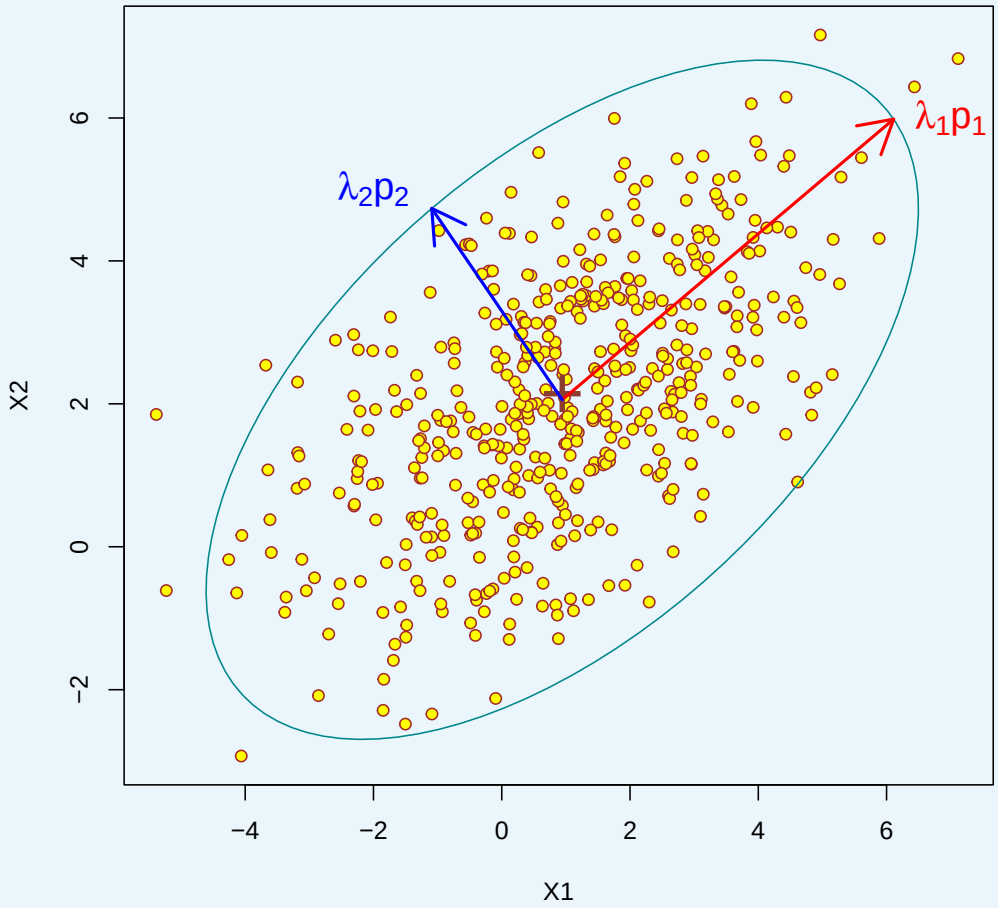
$$\frac{\sum_{i=1}^{k} \lambda_i}{\sum_{i=1}^{p} \lambda_i} = \frac{\sum_{i=1}^{k} \lambda_i}{tr(\mathbf{S})}$$

4. PCA scores $\mathbf{T}_{n,k}$ (the columns of $\mathbf{T}$ are the principal components):

$$\mathbf{T}_{n,k} = (\mathbf{X} - \mathbf{1}\bar{\mathbf{x}}^\top)\mathbf{P}_{p,k}$$

Example

PCA is centering plus rotation (the axes will line up with the directions of highest variance.)



Choosing the number of components

- Retain the first k components which explain a large proportion of the total variation, say, more than 80%.

$$\frac{\sum_{i=1}^{k} l_i}{\sum_{i=1}^{p} l_i} >= 0.8$$

- Examine a scree plot – a plot of the eigenvalues (component variances) versus the component number. The idea is to look for an "elbow" which corresponds to the point after which the eigenvalues decrease more slowly.
- The PREdicted Sum of Squares (PRESS) statistic measures the predictive ability of PCA.
- Many more ... see Jolliffe (1986) for an overview.

Standardization and equivariance

- If the variance of the original variables differs a lot between variables, it is recommended to first standardize the variables – otherwise the first principal components will be dominated by the variables with largest variance.
- When the variables are standardized by dividing them by their standard deviation, classical PCA comes down to decomposing the correlation matrix of the data, instead of the covariance matrix.
- In case of a robust PCA method, we will standardize the variables by dividing them by the MAD or another robust scale estimator.

Standardization and equivariance

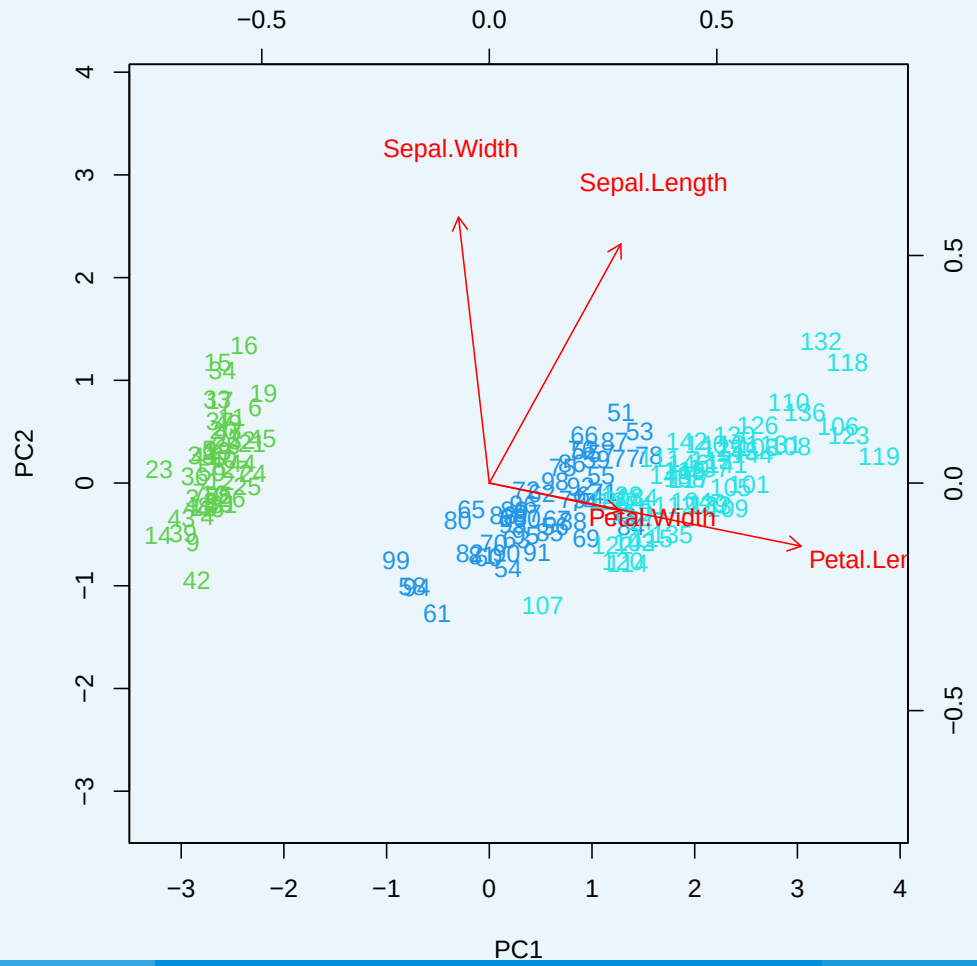
- PCA is sensitive to standardization of the variables and thus it is NOT affine equivariant.
- PCA is however orthogonally equivariant: when the data are rotated or reflected, the center and the principal components are rotated/reflected accordingly.
- This means that any robust PCA method only needs to be orthogonally equivariant which allows, for example, to use the L1-median as robust estimate of the center.

Biplots

- The biplot (Gabriel, 1971) is a graphical method for simultaneously displaying the variables and observations in a multivariate data matrix.
- The PCA biplot displays the component scores and the variable loadings obtained by PCA in two (or three) dimensions.

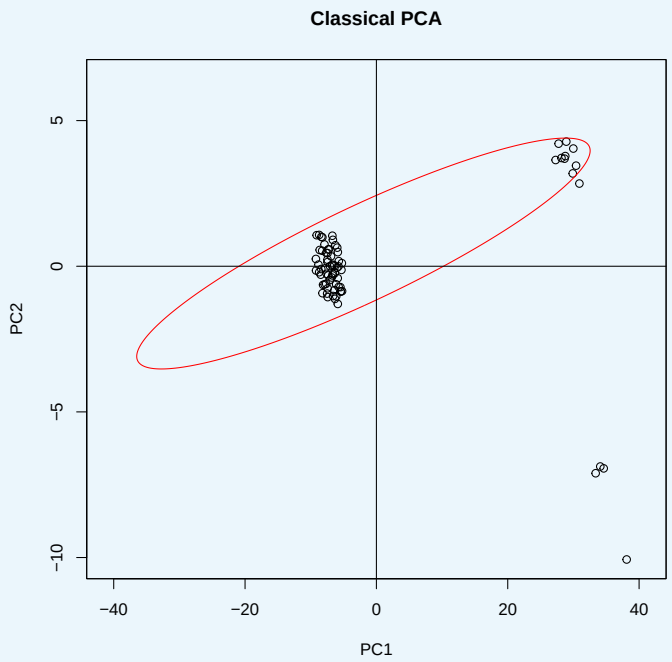
Biplots

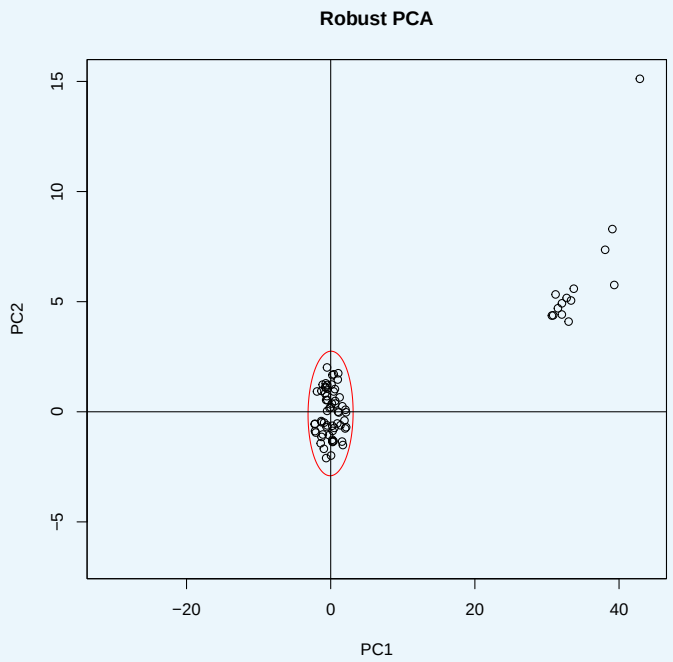
```
> pc <- PcaClassic(iris[, 1:4])
> biplot(pc, scale=0, col=list(as.integer(iris$Species)+2, "red"))
```



Example: Why do we need robust PCA

```
> pca.scoreplot(PcaClassic(hbk))

> pca.scoreplot(PcaCov(hbk))
```





PCA in R

- `prcomp()` (SVD) and `princomp()` (eigendecomposition) in base R
- Many others (some of):
  - `PCA()` in package **FactoMineR**
  - `dudi.pca()` in package **ade4**
  - `acp()` in package **amap**
  - `principal_components()` in package **anomalyDetection**
- (Robust) PCA functions in package **rrcov**
  - `PcaClassic()`
  - `PcaCov()`, `PcaHubert()`, `PcaLocantore()`, `PcaGrid()`, `PcaProj()`

Let us try it now...

Let us try it now...

Skull measurements on Rocky Mountain and Arctic wolves (*Canis Lupus* L.)
Morrison (1990) pp. 283–299

```
library(rrcov); data(wolves)
```

Outline

Outliers in PCA

- In general, outliers are observations which are distant, with respect to some distance measure, from the bulk of the data, we can say, from the center of the data.
- With respect to an estimated PCA model we can consider two aspects, when defining which observations are outlying:
  - Outlying relative to the PCA subspace and
  - Outlying within the PCA subspace.
- Thus, for identifying observations as outlying, we need to define two types of distances:
  - (a) orthogonal distances relative to the PCA subspace and
  - (b) score distances within the PCA subspace.

Orthogonal distances

- For a given data matrix $\mathbf{X}_{n,p}$ any PCA method will produce k (robust) principal components collected in a loadings matrix $\mathbf{P}_{p,k}$, (robust) center $\hat{\boldsymbol{\mu}}$, and a diagonal matrix of eigenvalues $\mathbf{L}_{k,k}$. The robust scores are $\mathbf{t}_i = \mathbf{P}^t(\mathbf{x} - \boldsymbol{\mu}), i = 1, \ldots, n$
- The orthogonal projection of each observation $\mathbf{x}_i$ on the PCA subspace is

$$\hat{\mathbf{x}}_i = \hat{\boldsymbol{\mu}} + \mathbf{P}_{p,k}\mathbf{t}_i = \hat{\boldsymbol{\mu}} + \mathbf{P}_{p,k}(\mathbf{P}^\top)_{k,p}(\mathbf{x}_i - \hat{\boldsymbol{\mu}})$$

- Then the orthogonal distances of each observation to the subspace spanned by the first k principal components are defined by:

$$OD_{i,k} = ||\mathbf{x}_i - \hat{\mathbf{x}}_i|| = ||\mathbf{x}_i - \boldsymbol{\mu}_i - \mathbf{P}_{p,k}\mathbf{t}_i||, i = 1, \ldots, n$$

Score distances

- To define outliers within the PCA subspace we look for observations whose projections are outliers in the subspace. We can measure this by computing a robust distance in the k-dimensional PCA subspace. This distance only uses the scores, hence it is called the score distance.
- Since the scores are centered, and their variability is estimated by the eigenvalues contained in the $\mathbf{L}_{k,k}$ matrix, the score distances are given by:

$$SD_{i,k} = \sqrt{\mathbf{t}_i^\top \mathbf{L}_{k,k}^{-1} \mathbf{t}_i} = \sqrt{\sum_{j=1}^{k} \frac{t_{ij}^2}{l_j}}, i = 1, \ldots, n$$

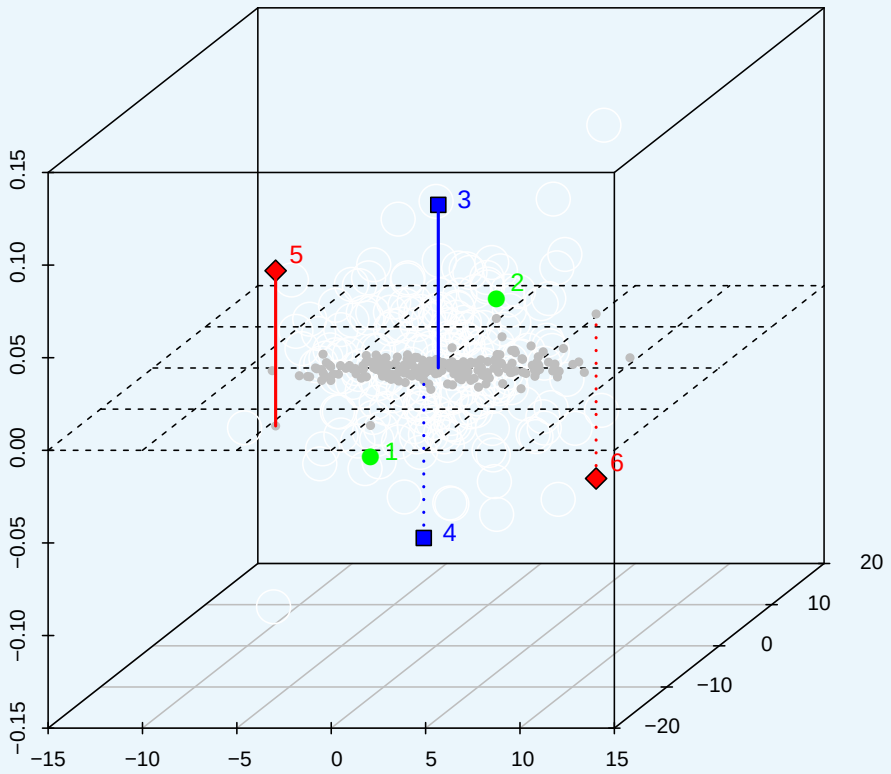
where l_j are the eigenvalues

The Outlier map

- Thus, an outlier in the context of PCA is characterized by the following:
  - Lies far from the subspace spanned by the first k eigenvectors (large orthogonal distance) and/or
  - The projected observation lies far from the bulk of the data within this space (large score distance)
- Using these two distances, four types of observations can be defined:
  - regular observations with small orthogonal and small score distance
  - bad leverage points with large orthogonal and large score distance
  - good leverage points with large score distance and small orthogonal distance and
  - orthogonal outliers with large orthogonal distance and small score distance

The Outlier map

- To illustrate this, let's generate 3-dimensional observations which lie essentially on a plane. Then we replace the first 6 observations by different types of outliers.

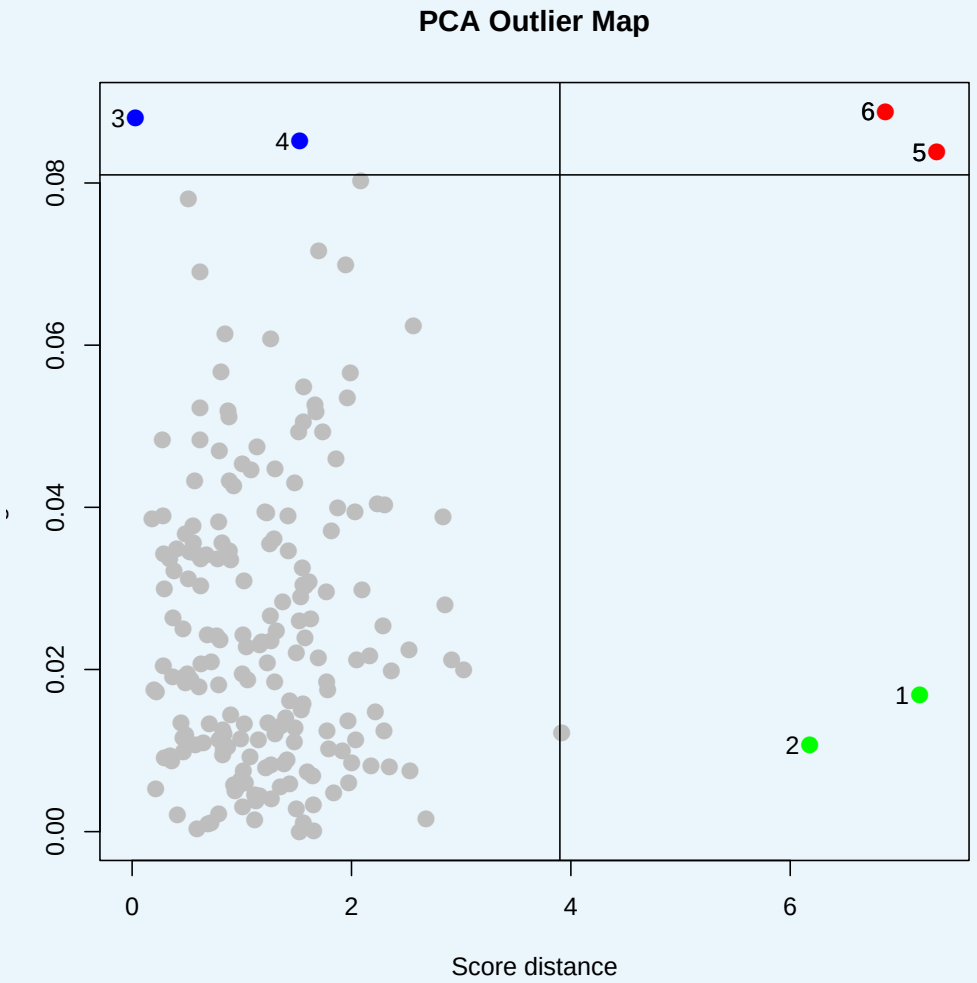


The Outlier map

- The outlier map (Hubert et al, 2005) highlights the outliers on a plot and classifies them in these three types.
- It plots the orthogonal distances versus the score distances.
- For each type of distance, cut-off values are available to flag outliers (Hubert et al, 2005).
- Lines (one horizontal and one vertical) are drawn to distinguish between observations with a small and a large orthogonal distances, and between a small and a large score distances.
- The outlier map is defined only if the number of selected principal components k is smaller than the rank of the matrix - because if $k = p$ the orthogonal distances cannot be computed. In such case we can look only at a plot of the score distances.

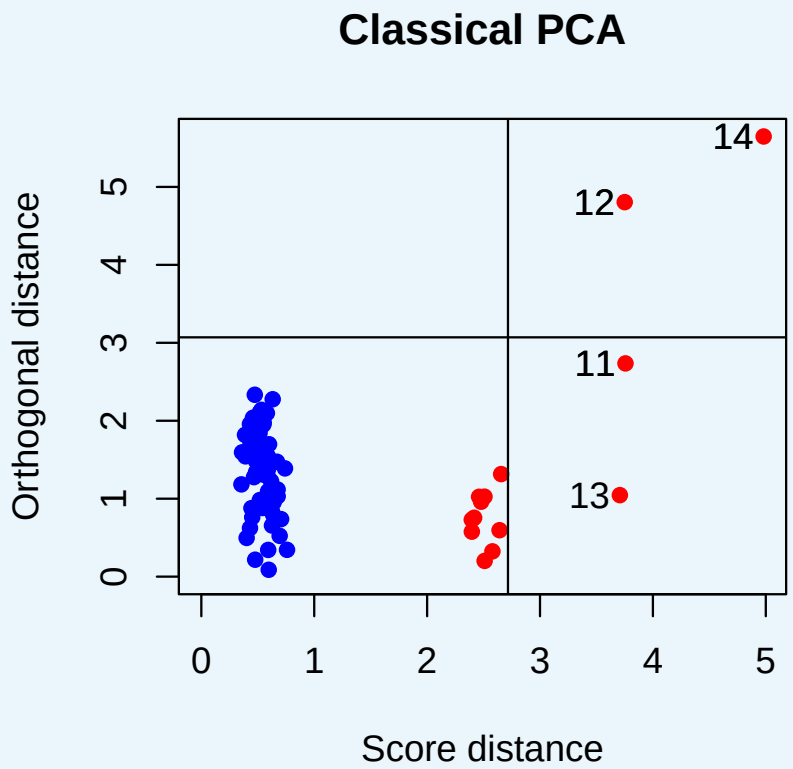
The Outlier map

- The outlier map of the generated data:



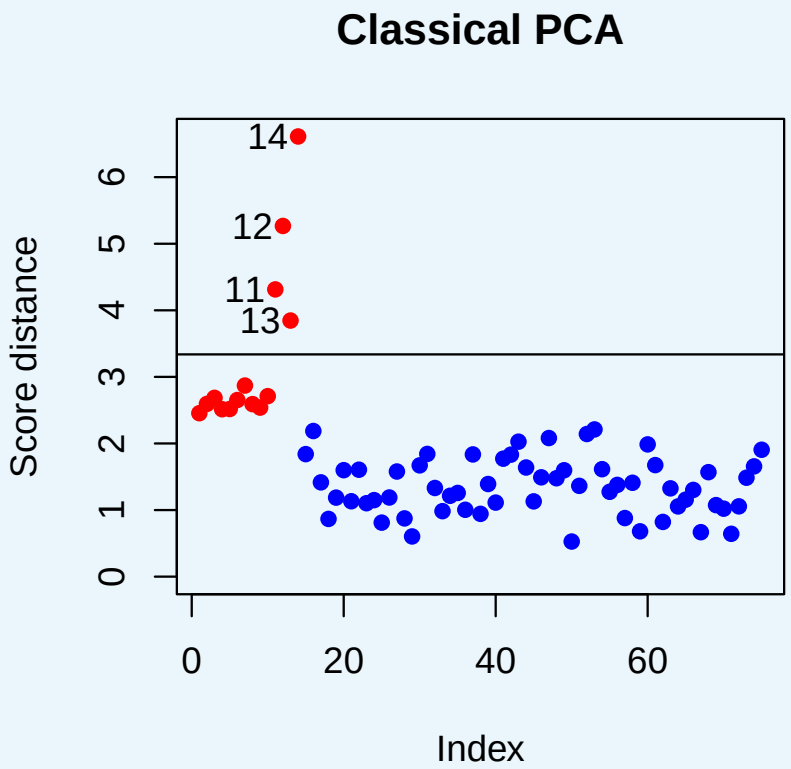
The Outlier map in R

```
> library(rrcov)
> data(hbk)
> pc <- PcaClassic(hbk, k=2)
> plot(pc, id.n.sd=4, off=0.05, pch=16, col=c(rep("red", 14), rep("blue", 76)))
> ## The 14 observations shown in red are known to be outliers, however the classical PCA
> ##  identifies only four of them
```



Score distance plot in R

```
> library(rrcov)
> data(hbk)
> pc <- PcaClassic(hbk)       # k=4 (default)
> plot(pc, id.n.sd=4, off=0.05, pch=16, col=c(rep("red", 14), rep("blue", 76)))
> ## The 14 observations shown in red are known to be outliers, however the classical PCA
> ##  identifies only four of them
```



Cutoff values for the score and orthogonal distances

- In order to distinguish the observations with small and large score distance and small and large orthogonal distance we need cutoff values for these distances.
- Cutoff for the score distance:
  $c_h = \sqrt{\chi^2_{k,0.975}}$
- Cutoff for the orthogonal distances:
  $c_\nu = (\hat{\mu} + \hat{\sigma} z_{0.975})^{3/2}$ where $z_{0.975}$ is the 97.5% quantile of the standard normal distribution. The parameters of the normal distribution can be estimated by the median and MAD of the values $OD_i^{2/3}$ (or by applying the univariate MCD to these values).

The PCA object

```
> library(rrcov)
> data(hbk)
> pc <- PcaClassic(hbk, 2)
> head(pc$od)     # orthogonal distances

        1         2         3         4         5         6
0.5770588 1.0244341 0.5957763 0.7301206 0.2020685 0.9610542

> head(pc$sd)     # score distances

       1        2        3        4        5        6
2.395925 2.460375 2.641565 2.395392 2.509341 2.478453

> head(pc$flag)   # outlier identification

   1    2    3    4    5    6
TRUE TRUE TRUE TRUE TRUE TRUE

> pc$cutoff.od    # OD cutoff

[1] 3.070344

> pc$cutoff.sd    # SD cutoff

[1] 2.716203
```

Outline

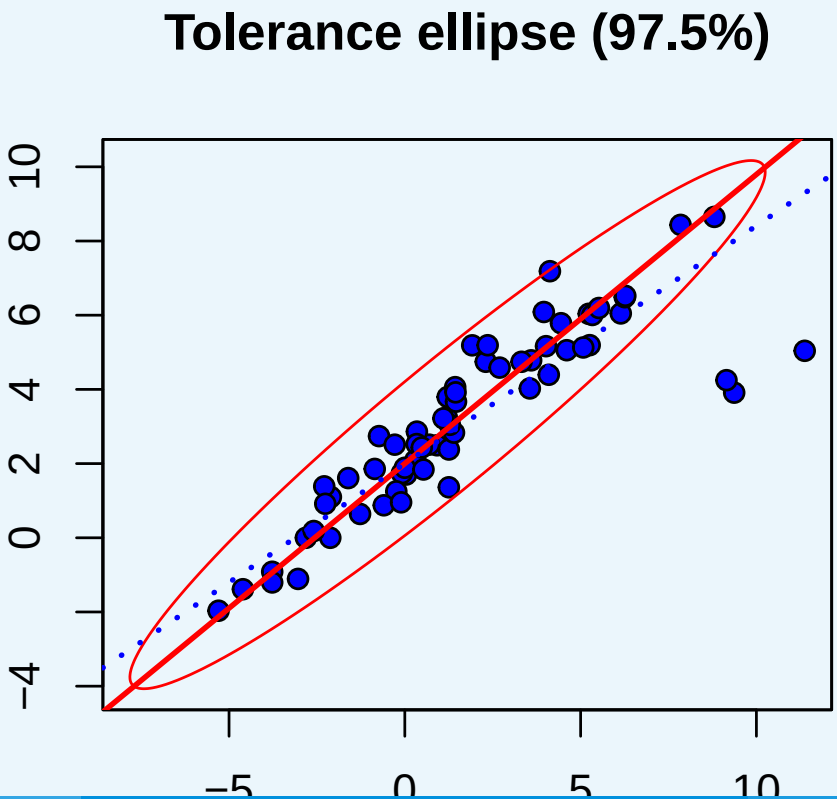
Plug-in PCA

- The easiest way to robustify the PCA is us robust estimates of location and covariance matrix instead of the sample ones.
- Concretely:
  - Replace the sample covariance matrix $\mathbf{S}_n$ by a robust covariance estimate $\hat{\mathbf{\Sigma}}$, e.g. the MCD, multivariate S- or MM-estimator.
  - The robust center corresponds to the robust location estimate associated with $\hat{\mathbf{\Sigma}}$.
  - The k robust eigenvalues then correspond to the k largest eigenvalues of $\hat{\mathbf{\Sigma}}$.
  - Take the corresponding eigenvectors.
- This approach can only be used when $n > 2p$ hence not for high-dimensional data.

Plug-in PCA: Example 1

Brain and body weights for 62 species of land animals (data set `Animals2` from package **robustbase**) – $n = 62, p = 2$.

- The red line is the first eigenvector of the MCD covariance matrix – corresponds to the main axis of the tolerance ellipse.
- The dotted blue line is the first classical eigenvector.



Plug-in PCA: Example 2

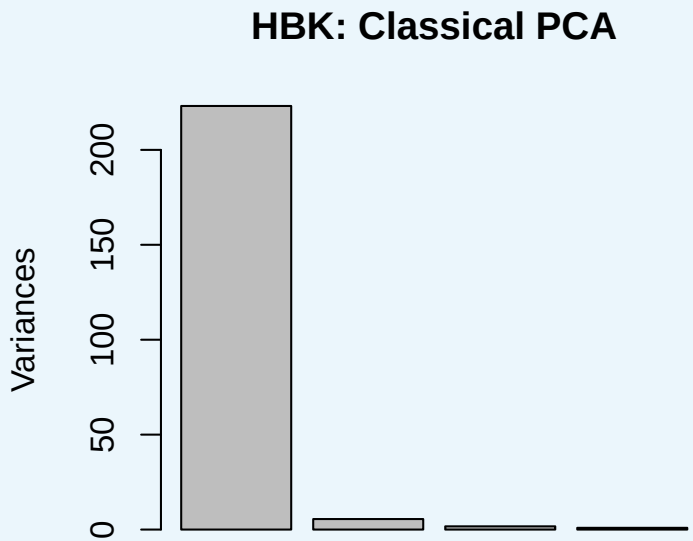
```
> library(rrcov)
> data(hbk)
> ##  ?hbk
```

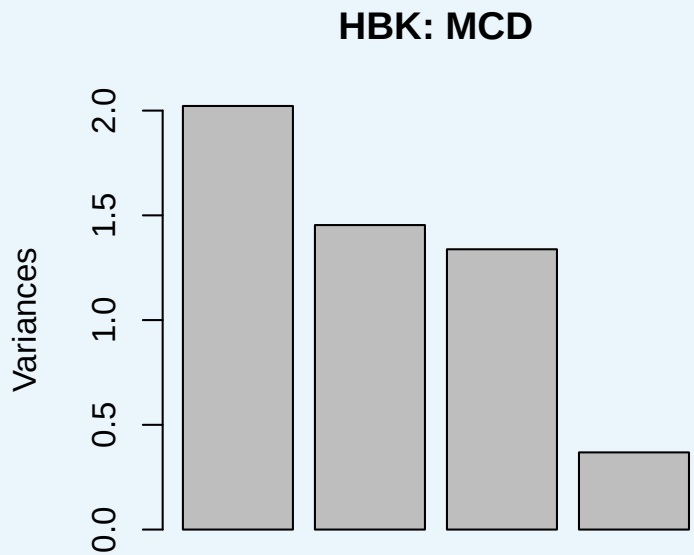
- hbk {robustbase}
- Hawkins, Bradu, Kass's artificial data
- Description: Artificial Data Set generated by Hawkins, Bradu, and Kass (1984). The data set consists of 75 observations in four dimensions (one response and three explanatory variables). It provides a good example of the masking effect. The first 14 observations are outliers, created in two groups: 1-10 and 11-14. Only observations 12, 13 and 14 appear as outliers when using classical methods, but can be easily unmasked using robust distances computed by, e.g., MCD - `CovMcd()`.
- Source: *Hawkins, D.M., Bradu, D., and Kass, G.V. (1984) Location of several outliers in multiple regression data using elemental sets. Technometrics 26, 197–208*

Plug-in PCA: Example 2

Screeplot of the classical and robust (MCD) PCA for the HBK data set.

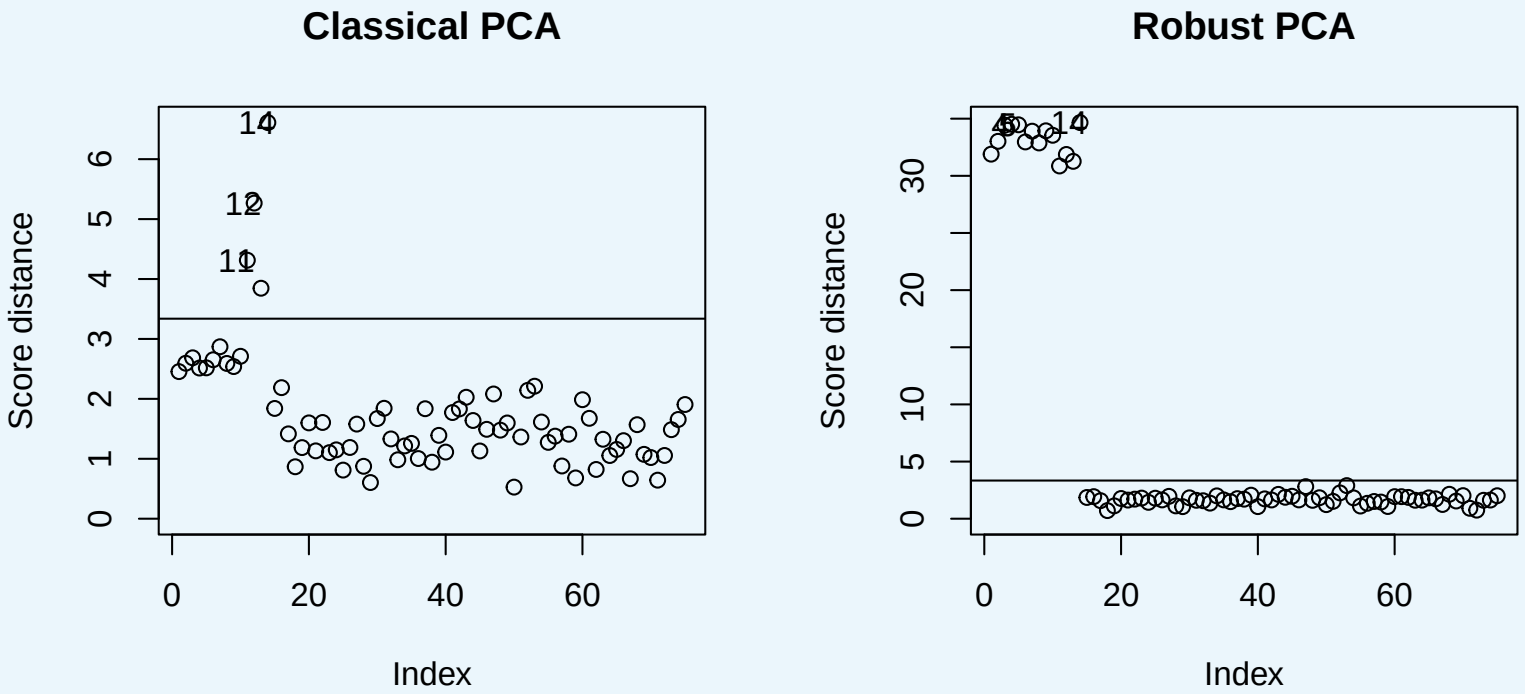
- The first classical eigenvector already explains 96.5% of the total classical variance.
- The robust analysis explains 63% of the total variability when k = 2 and 93% when k = 3.





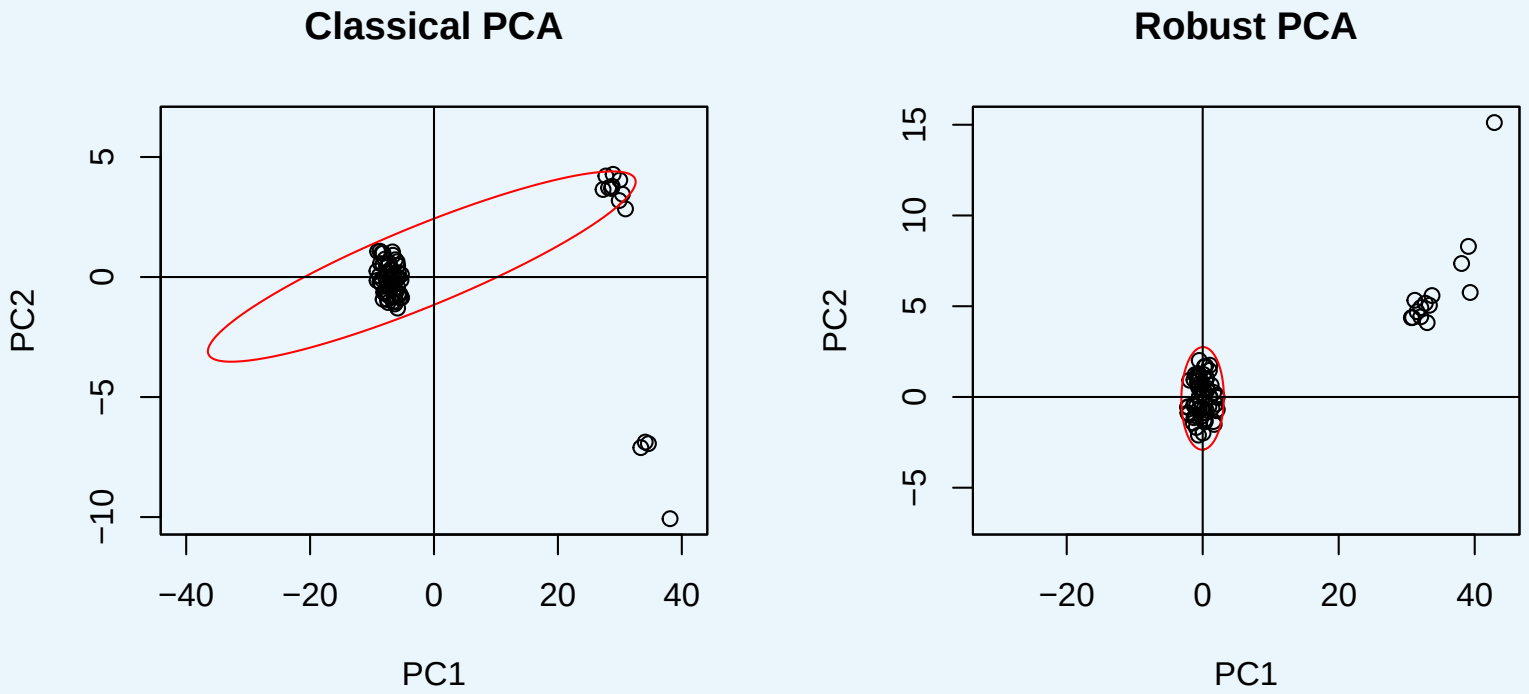
Plug-in PCA: Example 2

If we select all $k = 4$ principal components, we can look at the resulting score distances only:



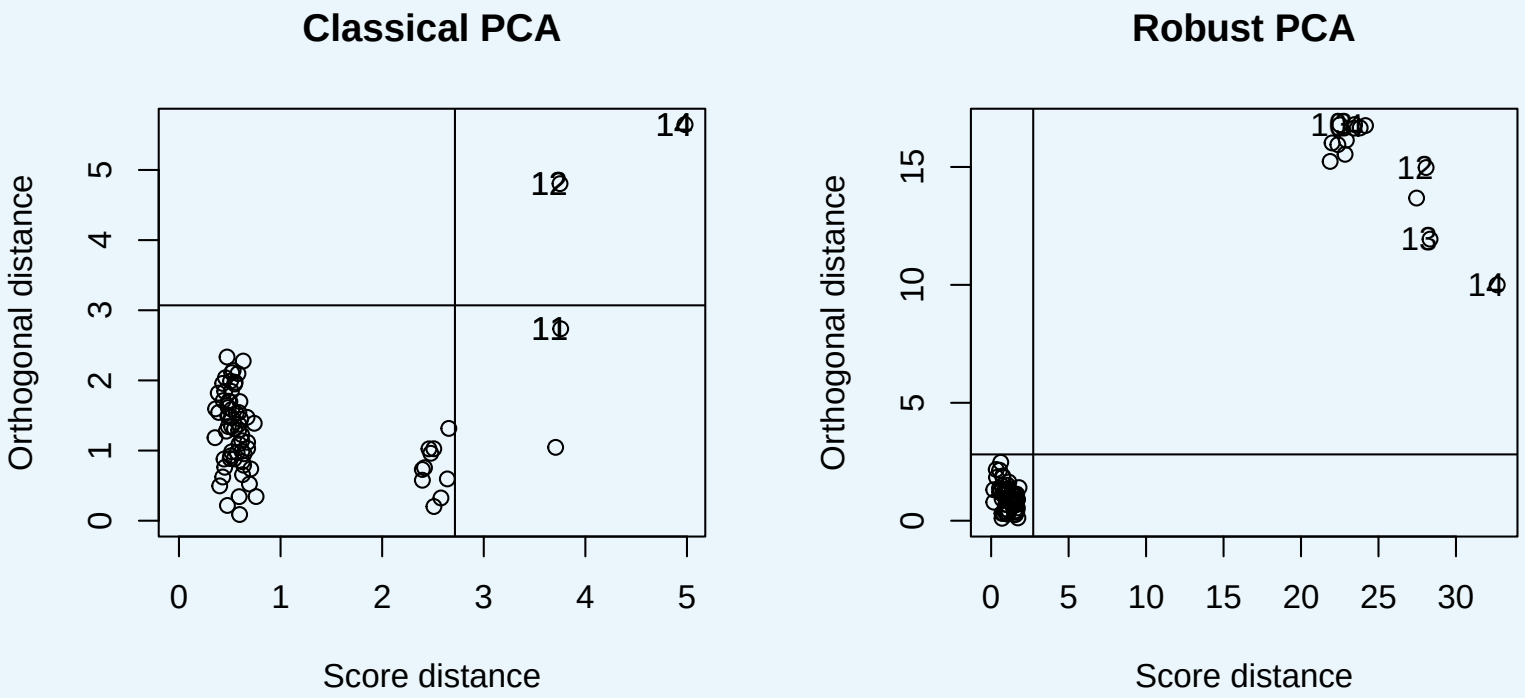
Plug-in PCA: Example 2

Scoreplot: For the classical PCA always $\hat{\mathbf{t}} = 0$, but here $(0, 0)$ is not at the center of the regular observations.



Plug-in PCA: Example 2

Outlier map: If we select $k = 2$ we can also look at the orthogonal distances



Plug-in PCA: Example 2

Summary of the Ranalysis with plug-in robust PCA

```
> library(rrcov)
> data(hbk)
> pca <- PcaClassic(hbk)       # Classical PCA
> rpc1 <- PcaCov(hbk)          # Uses MCD with alpha=0.5 (default)
> rpc2 <- PcaCov(hbk, cov.control=CovControlMMest())
> screeplot(rpc1)
> pca.scoreplot(rpc1)
> plot(rpc1)                    # outlier map
> getLoadings(rpc1)
> getEigenvalues(rpc1)
> getScores(rpc1)
```

Outline

Projection pursuit PCA

- Let us again have an $n \times p$ data matrix $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$, whose i-th row is the vector (observation) $\mathbf{x}_i$ with $\mathbf{x}_i \in \mathbb{R}_p$.
- We again assume that the input variables are continuous.
- Further, we assume that the data are centered, e.g.

$$\mathbf{X} = (\mathbf{X} - \mathbf{1}\hat{\mu}^\top)$$

- The L1 median can be used to robustly center the data.
- Furthermore, the input data matrix can be scaled, if necessary.

Projection pursuit PCA

- For the centered (and possibly scaled) data matrix, find linear combinations $\mathbf{t}_j$ that result from a projection of the centered data on a direction $\mathbf{p}_j$:

$$\mathbf{t}_j = \mathbf{X}\mathbf{p}_j \qquad \text{such that}$$

$$\mathbf{p}_j = \arg\max_{\mathbf{p}} \text{Var}(\mathbf{X}\mathbf{p}) \qquad \text{subject to}$$

$\|\mathbf{p}_j\| = 1$ and $\text{Cov}(\mathbf{X}\mathbf{p}_j, \mathbf{X}\mathbf{p}_l) = 0$ for $l < j$ and $j = 1, \ldots, k$ with $k \leq \min(n, p)$.

Projection pursuit PCA

Var is a variance measure:

- Classical case: *Var* is the empirical variance
  $\Rightarrow \mathbf{p}_1, \ldots, \mathbf{p}_k$ correspond to the first k (dominant) eigenvectors of the sample covariance matrix.
  $\Rightarrow l_1, \ldots, l_k$ correspond to the first k eigenvalues of the sample covariance matrix.
- Robust case: *Var* is squared robust scale estimator
  $\Rightarrow \mathbf{p}_1, \ldots, \mathbf{p}_k$ are the directions of the first k robust PC
  $\Rightarrow$ The k robust "eigenvalues" l_j then correspond to the robust variance of the data projected onto $\mathbf{p}_1, \ldots, \mathbf{p}_k$:

$$l_j = \mathrm{VAR}(\mathbf{p}_j^\top \mathbf{x}_1, \mathbf{p}_j^\top \mathbf{x}_2, \ldots \mathbf{p}_j^\top \mathbf{x}_n)$$

Projection pursuit PCA

Robust scale estimators:

Given n univariate observations $y_1, y_2, \ldots, y_n$.

- Median Absolute Deviation:

$$\underset{1 \leq i \leq n}{Median} |y_i - Median(y_1, y_2, \ldots, y_n)|$$

- Q_n-scale:

  First quartile of all distances between pairs of points $|y_i - y_j|$

Advantages of the PP approach

- The projection pursuit (PP) approach was developed by Li and Chen (1985), Hubert et al. (2002), and Croux and Ruiz-Gazen (2005), Croux et al. (2010)
- It can be used when p $>$ n as it projects the data on lines.
- It is performed sequentially and can be stopped whenever sufficiently many components are obtained.
- The solutions are nested: any j-dimensional PCA subspace is a subspace of all higher-dimensional PCA subspaces found later.

Let us try it now...

Let us try it now...

- The bus data set (Hettich and Bay, 1999) corresponds to a study in automatic vehicle recognition (see Maronna et al. 2006, page 213, Example 6.3)). This data set from the Turing Institute, Glasgow, Scotland, contains measures of shape features extracted from vehicle silhouettes. The images were acquired by a camera looking downward at the model vehicle from a fixed angle of elevation.
- Each of the 218 rows corresponds to a view of a bus silhouette, and contains 18 attributes of the image.

Projection pursuit PCA: Example

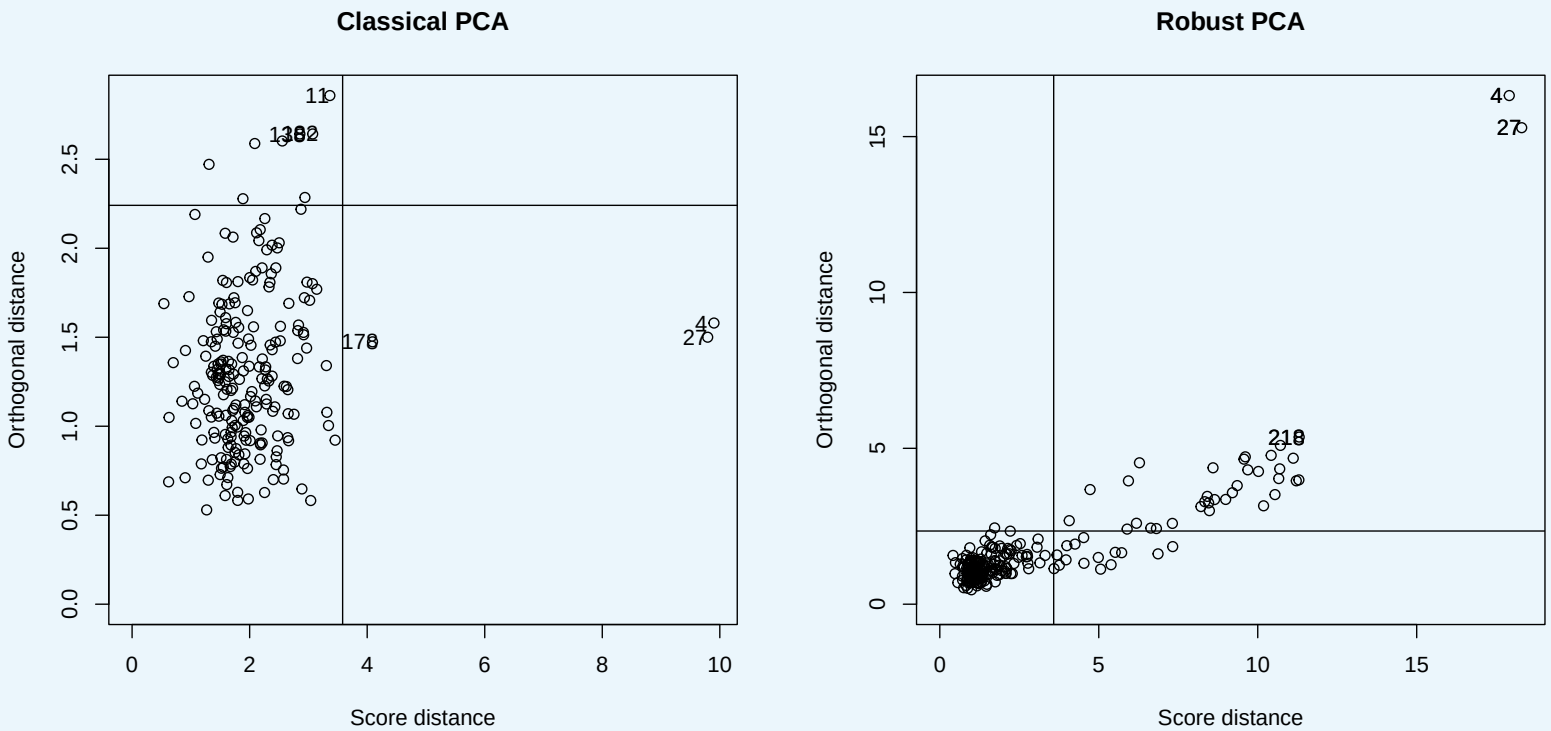
- One variable has zero MAD and is removed, hence p $= 17$.
- We standardize the data, and apply the projection-based PCA with Qn as scale estimator. Then 92% of the variability is explained by $k = 5$ components.
- NOTE: Instead of standardizing by dividing by a robust scale, we can call the Pca function with argument `scale=mad` or `scale=qn`.

Projection pursuit PCA: Example

```
>     data(bus)
>     bus <- as.matrix(bus)
>     ## calculate MADN for each variable
>     xmad <- apply(bus, 2, mad)
>     cat("\nMin, Max of MADN: ", min(xmad), max(xmad), "\n")

Min, Max of MADN:  0 34.8411

>     ## MADN vary between 0 (for variable 9) and 34. Therefore exclude
>     ##  variable 9 and divide the remaining variables by their MADNs.
>     bus1 <- bus[, -9]
>     madbus <- apply(bus1, 2, mad)
>     bus2 <- sweep(bus1, 2, madbus, "/", check.margin = FALSE)
>     rpc  <- PcaGrid(bus2, method="qn")
>     ev <- getEigenvalues(rpc)
>     cumsum(ev)/sum(ev)

 [1] 0.4418105 0.6475868 0.7938430 0.8635211 0.9241256 0.9544322 0.9692368
 [8] 0.9781212 0.9841466 0.9891732 0.9925678 0.9955517 0.9969762 0.9982615
[15] 0.9994451 0.9997683 1.0000000
```

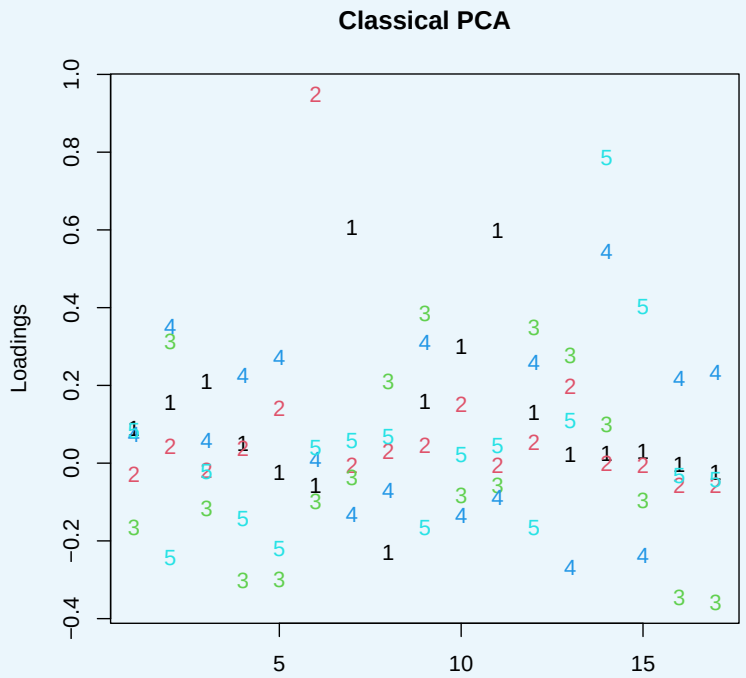
Projection pursuit PCA: Example

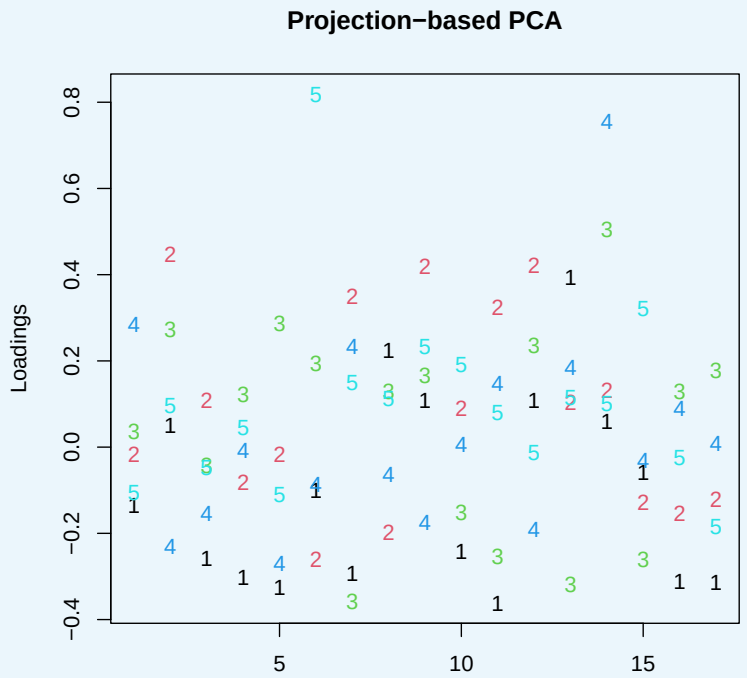
```
> pc <- PcaClassic(bus2, k=5)
> plot(pc)

> rpc <- PcaGrid(bus2, method="qn", k=5)
> plot(rpc)
```



Projection pursuit PCA: Example

- Compare the loadings: The first classical component is highly influenced by the 7-th and 11-th variable. The second classical component is influenced by the 6-th variable. These three variables all have many outliers.





Outline

Spherical PCA

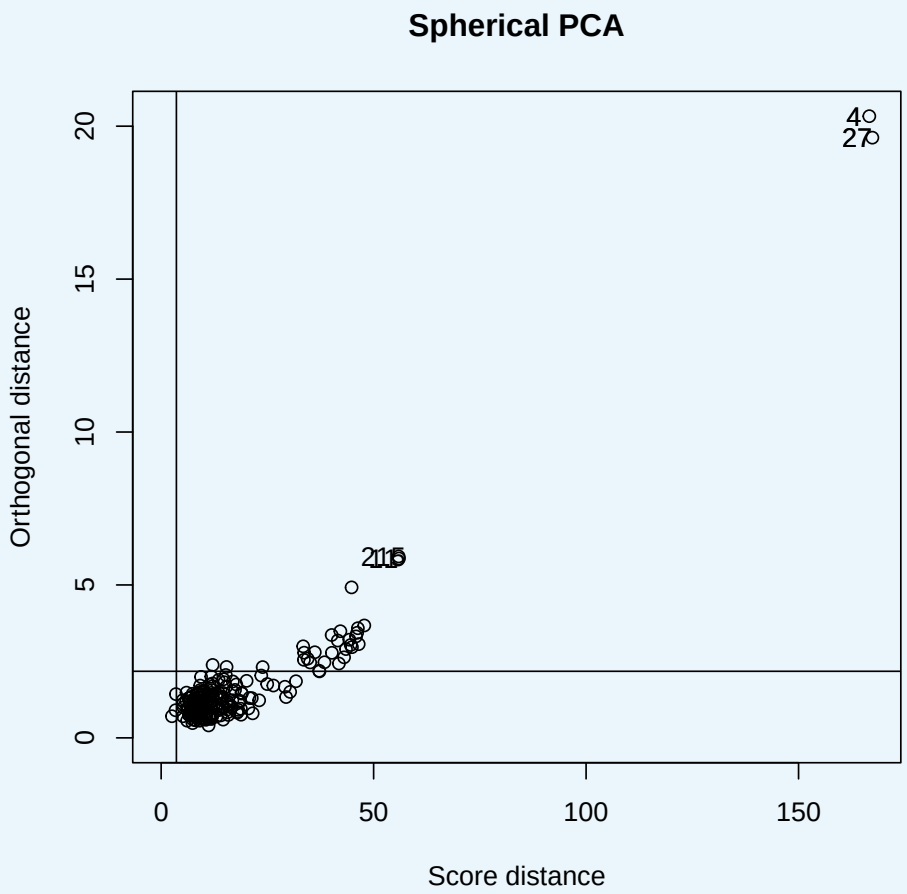
- Introduced by Locantore et al. (1999).
- See Rowsseeuw and Raymaekers for recent extensions
- Center by the L1-median, denoted as $\hat{\boldsymbol{\mu}}$.
- Project the data on the unit sphere with center $\hat{\boldsymbol{\mu}}$.
- The robust eigenvectors are computed as the dominant eigenvectors of the covariance matrix of these projected data points, i.e. the eigenvectors of the sign covariance matrix.

$$\hat{\boldsymbol{\Sigma}} = \frac{1}{n-1} \sum_{i=1}^{n} \frac{(\mathbf{x}_i - \hat{\boldsymbol{\mu}})}{||\mathbf{x}_i - \hat{\boldsymbol{\mu}}||} \frac{(\mathbf{x}_i - \hat{\boldsymbol{\mu}})^\top}{||\mathbf{x}_i - \hat{\boldsymbol{\mu}}||}$$

  with the largest eigenvalues.
- These eigenvalues are not consistent, but they can be replaced by a robust scale (e.g. the squared MAD) of the scores.

Spherical PCA: Example

```
> sphpca <- PcaLocantore(bus2, k=5)
> plot(sphpca, main="Spherical PCA")
```



Outline

ROBPCA:Projection pursuit and MCD

- The PCA method ROBPCA proposed by Hubert et al. (2005) tries to combine the advantages of both approaches—the PCA based on a robust covariance matrix and PCA based on projection pursuit.
- The projection pursuit part is used for the initial dimension reduction. The MCD estimator is then applied to this lower-dimensional data space.
- The combined approach should yield more accurate estimates than the raw PP algorithm.

Main steps of ROBPCA: 1

1. The data are preprocessed by reducing their data space to the subspace spanned by the n observations. This is done by singular value decomposition of the input data matrix $\mathbf{X}_{n,p}$. As a result the data are represented in a space whose dimension is $rank(X)$, being at most $n - 1$ without loss of information.

Main steps of ROBPCA: 2

2. A measure of outlyingness is computed for each data point and the $h < n$ "least outlying" points are found ($n/2 < h < n$).
   - The measure used is similar to the Stahel-Donoho outlyingness (SDO) which is orthogonally invariant.

$$SDO(\mathbf{x}_i) = \max_{\mathbf{v} \in B} \frac{|\mathbf{x}_i^\top \mathbf{v} - \hat{\mu}_{mcd}(\mathbf{x}_j^\top \mathbf{v})|}{\hat{s}_{mcd}(\mathbf{x}_j^\top \mathbf{v})}$$

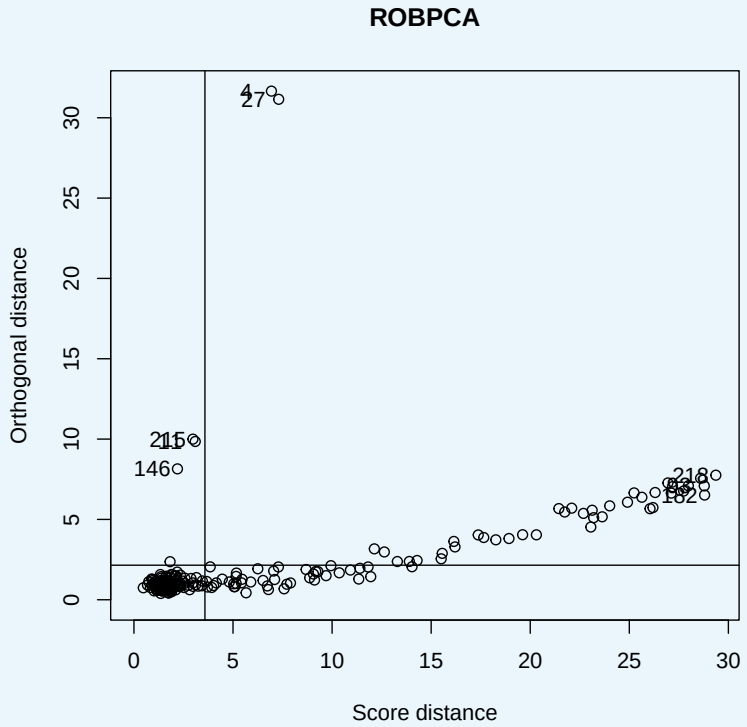
   with $\hat{\mu}_{mcd}$ and $\hat{s}_{mcd}$ the univariate MCD estimators of location and scale.
   - The set B contains 250 directions through two data points, randomly drawn from the data.
   - Let $\hat{\mathbf{\Sigma}}_h$ be the covariance matrix of the h points with smallest outlyingness.

Main steps of ROBPCA: 3

3. The data points are then projected on the k-dimensional subspace spanned by the k eigenvectors corresponding to the largest k eigenvalues of the matrix $\hat{\mathbf{\Sigma}}_h$. The location and scatter of the projected data are computed using the reweighted MCD estimator, and the eigenvectors of this scatter matrix yield the robust principal components.

ROBPCA: Example 1

```
>     pcaHubert <- PcaHubert(bus2, k=5, mcd=FALSE, alpha=0.5)
>     plot(pcaHubert, id.n.od=5, off=0.04, main="ROBPCA")
```



ROBPCA: Glass spectra example

- A data set originating from 180 16th-17th century archeological glass vessels (Janssens et al. 1998).
- There are 750 characteristics for each vessel, coming from an analysis by an electron-probe X-ray micro-analysis. The data set includes four different materials comprising the vessels, with the larger group being of 145 observations.
- It is known from other studies on this data set (Sernels et al. 2005; Filzmoser et al. 2008; Filzmoser and Todorov 2011;2014) that these 145 observations should form two groups, because during the measurement process the detector efficiency has been changed.

ROBPCA: Glass spectra example

- This example is also considered in detail in Hubert et al. (2005) illustrating the newly proposed ROBPCA algorithm.
- The data are available in the package **rospca** as the data set `Glass`. It is a data frame with 180 observations and 750 variables. For this example we will use only the first 500 wavelengths (variables).
- Instead of working with the raw data, we first robustly center the spectra by subtracting the univariate MCD location estimator from each wavelength. Doing so allow us to observe more of the variability that is present in the data.

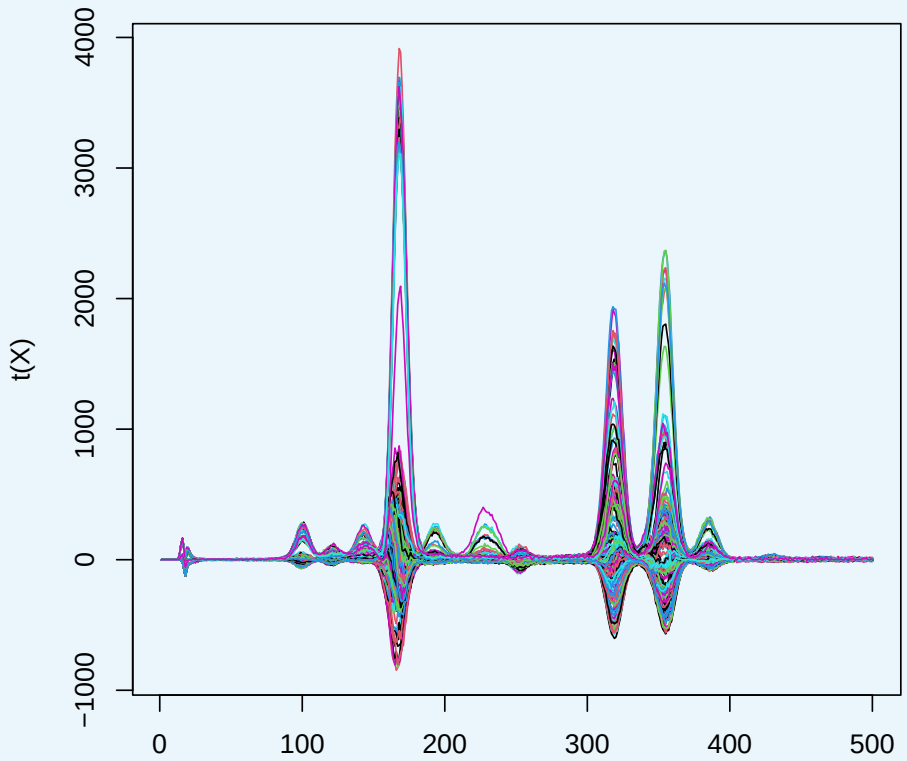
ROBPCA: Glass spectra example

```
> library(rospca)
> data(Glass)
> ##  We define a function to calculate the univariate MCD using the
> ##  internal rrcov function unimcd() and apply it to the data set
> ##  using the robustbase (undocumented) function doScale(). We set
> ##  the scale parameter to NULL - in order to do no scaling note
> ##  that in base::scale() we have scale=FALSE for this purpose.
>
> umcd_center <- function(y)
+     rrcov:::unimcd(y, quan=h.alpha.n(0.5, length(y), 1))$tmcd
> X <- doScale(Glass, center=umcd_center, scale=NULL)$x
> X <- X[, 1:500]
> (n <- nrow(X))

[1] 180

> (p <- ncol(X))

[1] 500
```

ROBPCA: Glass spectra example

- All observations, robustly centered

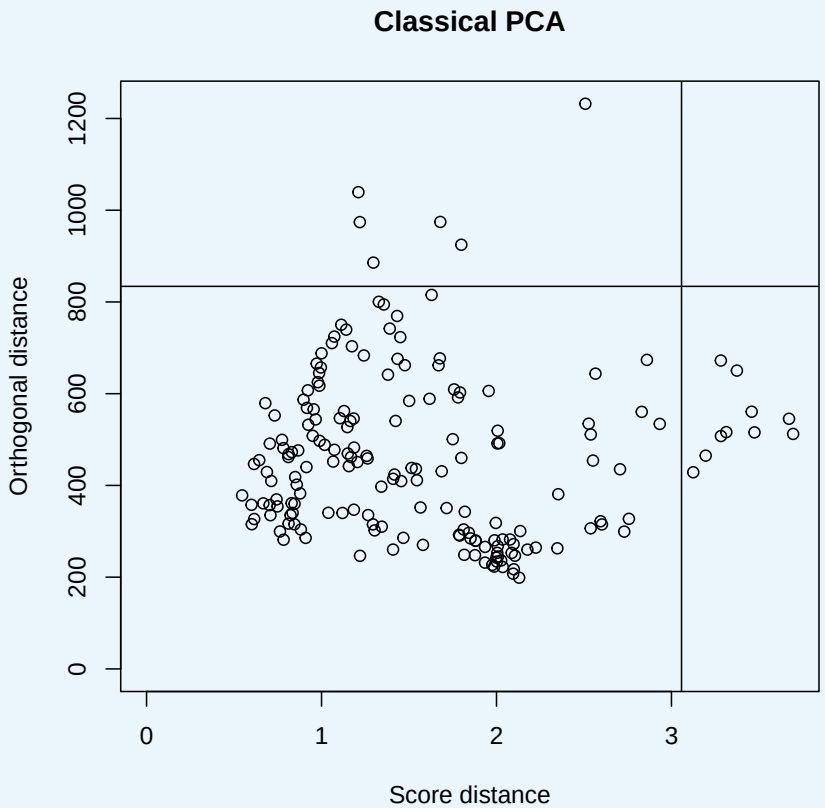
```
> matplot(t(X), type="l", lty=1)
```



ROBPCA: Glass spectra example

- Start by computing the classical PCA. Three components yield 99% explained variance.
- The outlier map shows only mild orthogonal outliers and good leverage points.

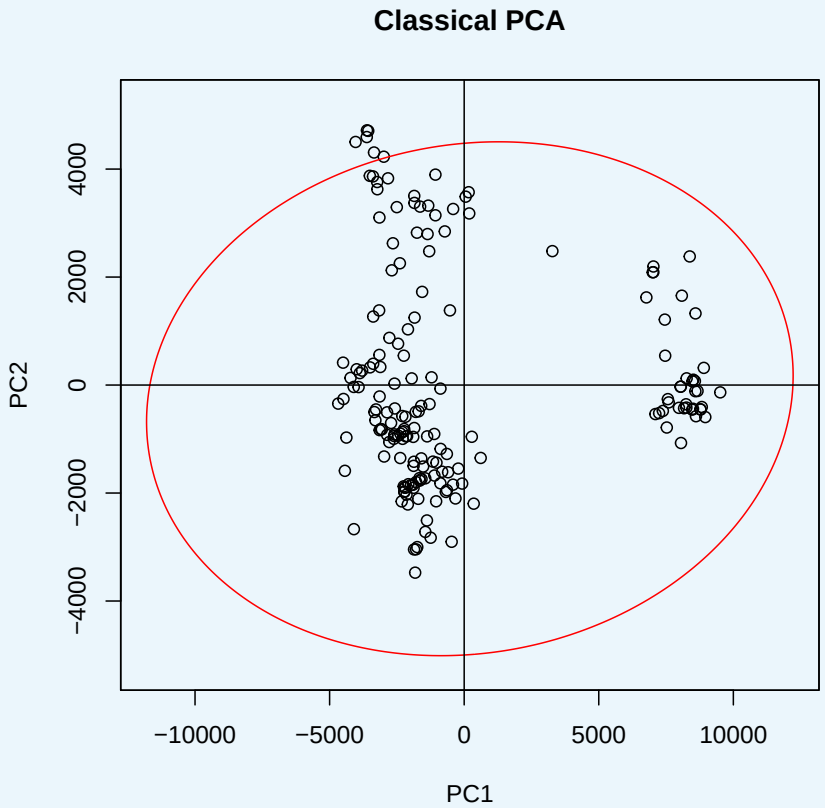
```
> pc <- PcaClassic(X, k=3, scale=FALSE)
> plot(pc, id.n.sd=0, id.n.od=0)
```



ROBPCA: Glass spectra example

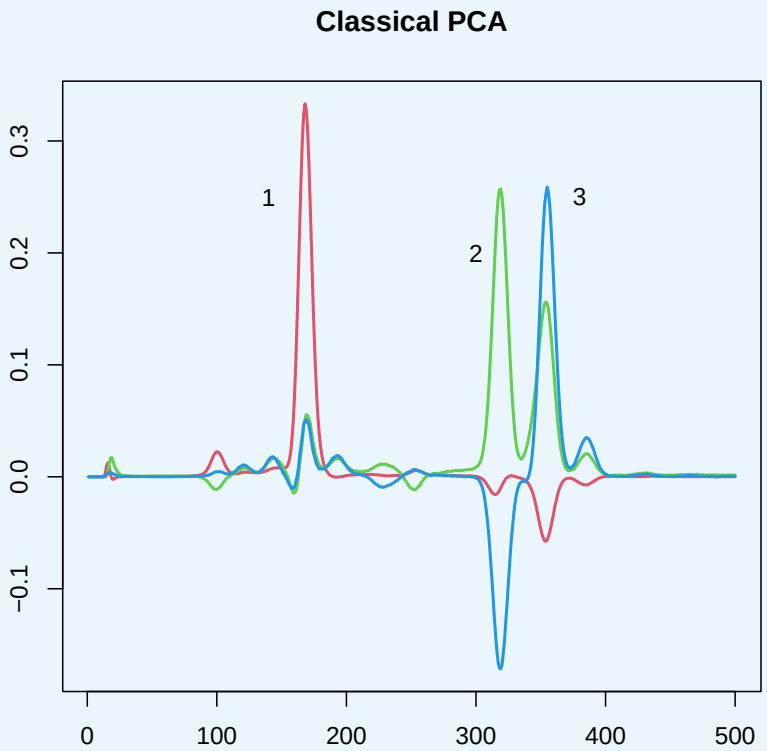
- The score plot shows the observations falling into several groups, but almost all are encompassed by the 0.975 tolerance ellipse.

```
> pca.scoreplot(pc)
```



ROBPCA: Glass spectra example

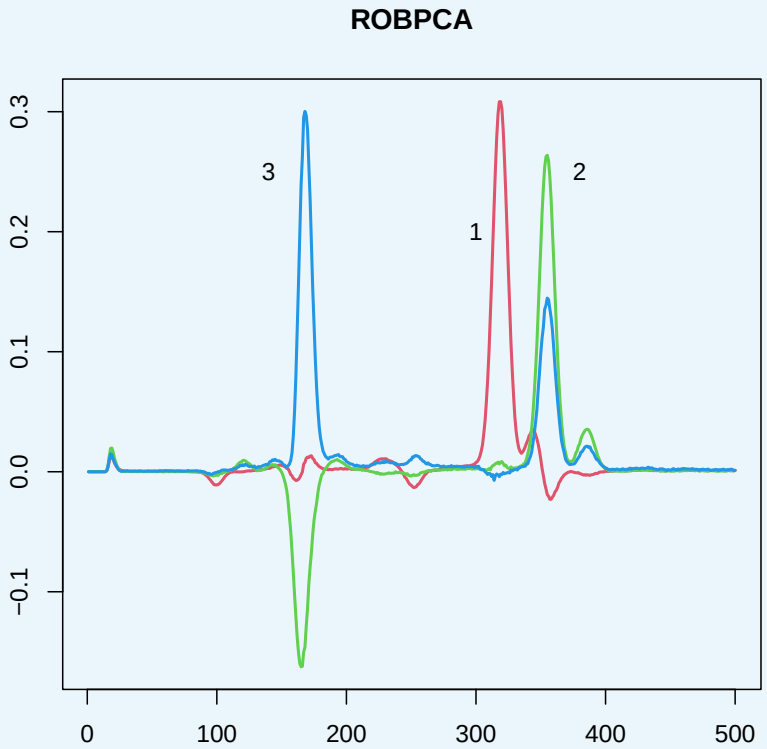
- The first three loadings vectors of classical PCA.
- The second and third peaks are mixed up.



ROBPCA: Glass spectra example

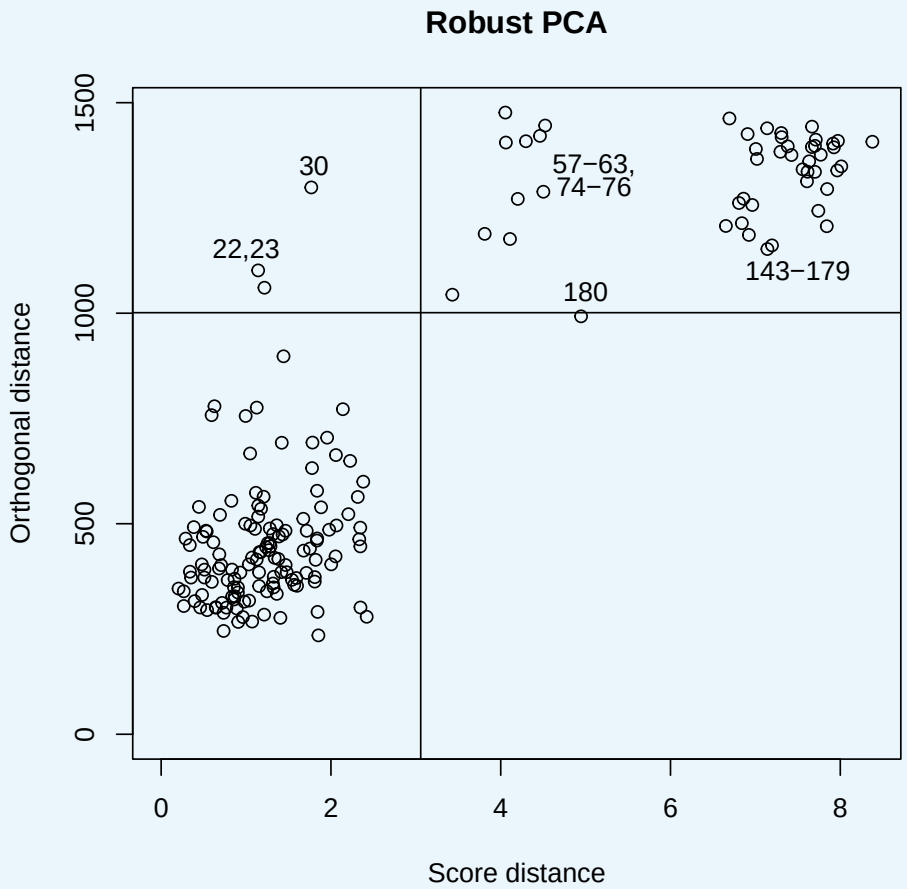
- The first three loadings vectors of the robust PCA.
- ROBPCA keeps the peaks more separate.

```
> rpc <- PcaHubert(X, k=3, scale=FALSE, alpha=0.7)
```



ROBPCA: Glass spectra example

- Outlier map from ROBPCA.
- The bad leverage points are easily seen.

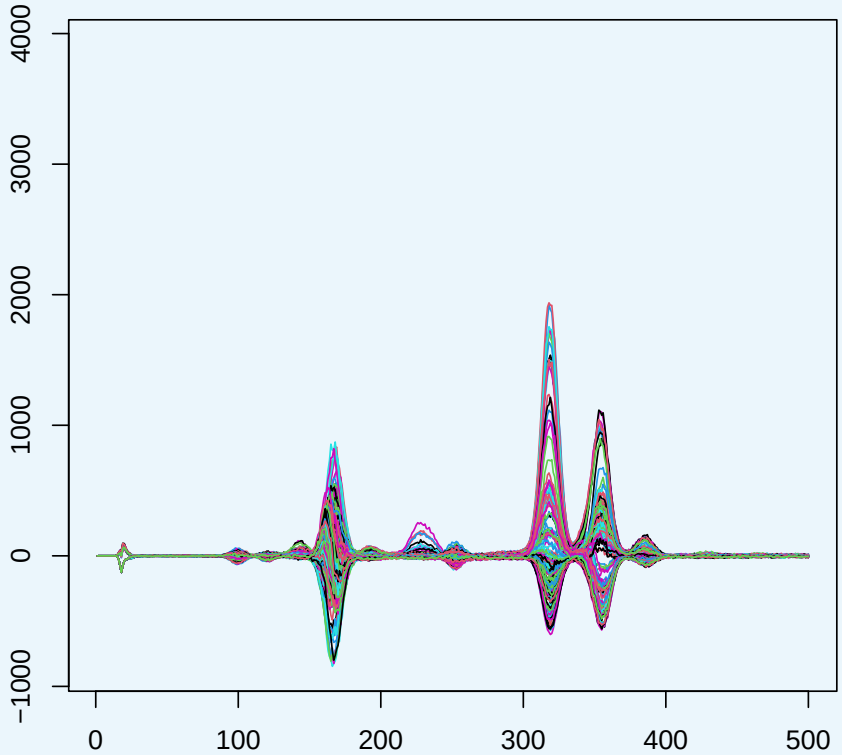


ROBPCA: Glass spectra example

- What has caused the outliers in the glass data?
  - The window of the detector system was cleaned before the last 38 spectra were measured and as a result less radiation was absorbed, hence more was detected.
  - Observations 57–63 and 74–76 are samples with a large concentration of calcium.
  - Observations 22, 23 and 30 are borderline cases (with a larger concentration of phosphor).
  - Observation 180 is also a borderline case.

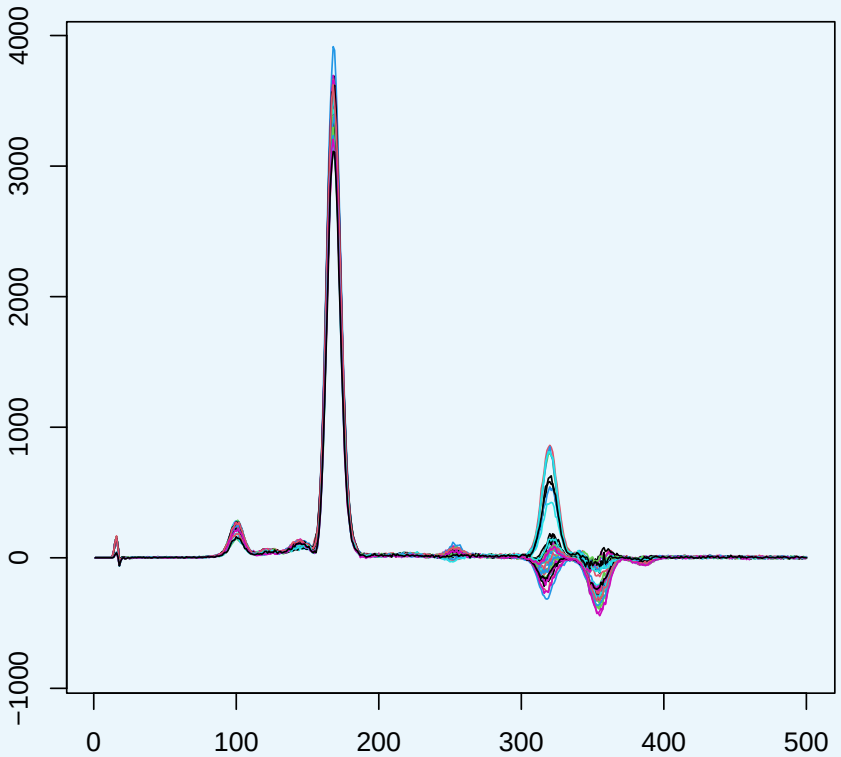
ROBPCA: Glass spectra example

- Regular observations.
- These clearly have lower measurements at channels 160–175 than did samples 143–179 (next slide).



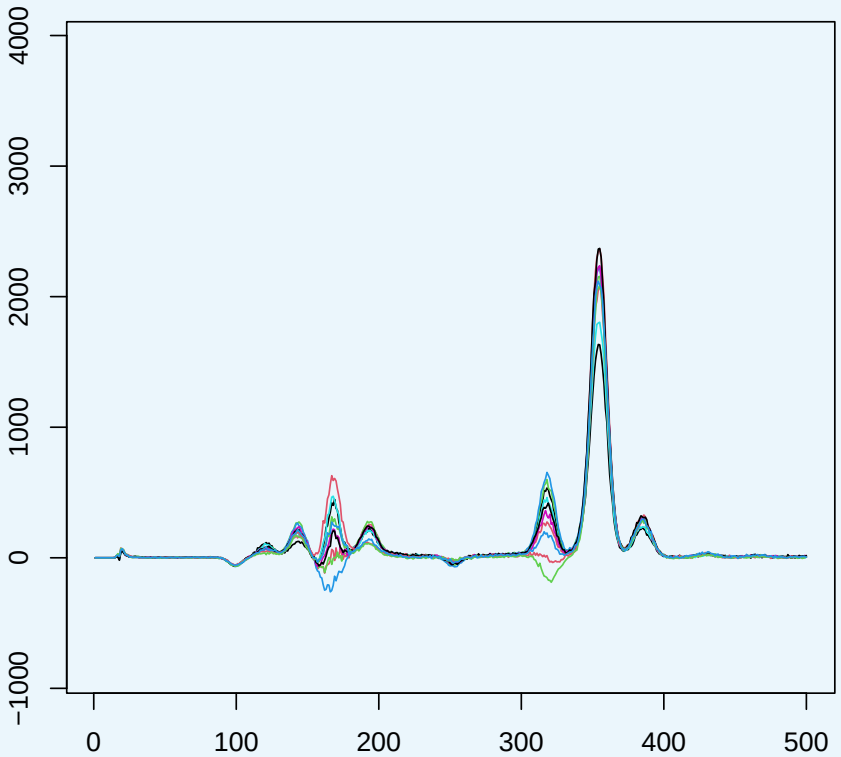
ROBPCA: Glass spectra example

- Bad leverage points 143–179.



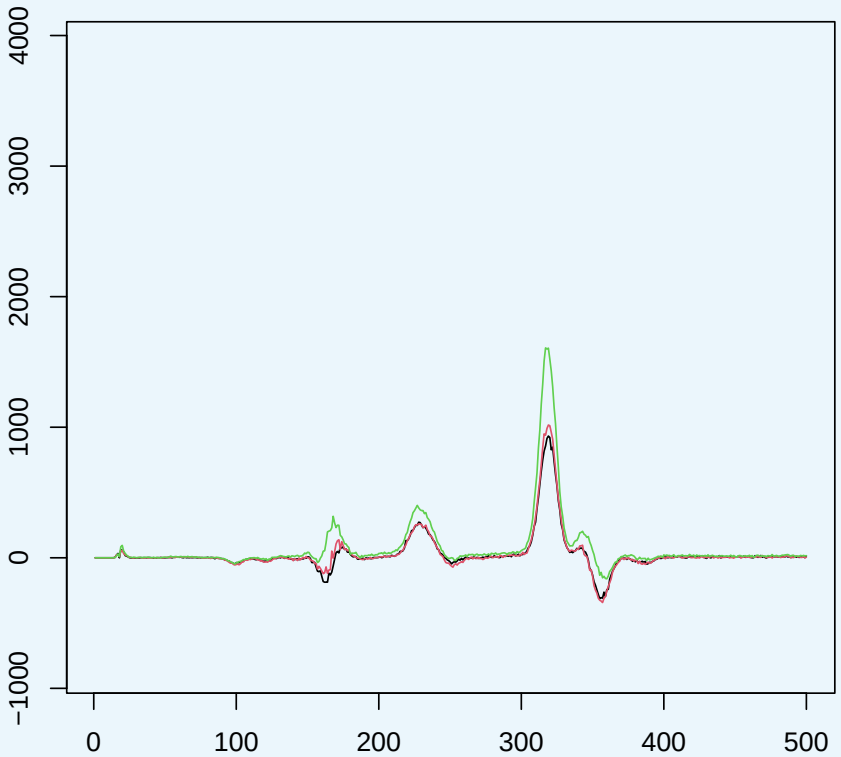
ROBPCA: Glass spectra example

- Bad leverage points 57–63 and 74–76.



ROBPCA: Glass spectra example

- Orthogonal outliers.



Exercise: Fruit data

Let us try it... your turn now ...

- The fruit data set (originally collected by Colin Greensill, Central Queensland University, Rockhampton, Australia) contains the spectra of three different cultivars of the same fruit (cantaloupe – *Cucumismelo L. Cantaloupensis*). Parts of it were analyzed in Hubert and Van Driessen (2004), Hubert et al. (2008), Vranckx et al. (2021)
- The cultivars (named D, M and HA) have sizes 490, 106 and 500, and all spectra were measured in 256 wavelengths.
- The dataset thus contains 1096 observations and 256 variables.

Exercise: Fruit data

- The data set is available in the development version of the package **rrcov** available at GitHub: `https://github.com/valentint/rrcov`

```
> ## install.packages("remotes")
> ##  remotes::install_github("valentint/rrcov")
> data(fruit)
> dim(fruit)
```

```
[1] 1096  257
```

```
> table(fruit$cultivar)
```

```
  D  HA   M
490 500 106
```

- Apply classical PCA to the whole data set and to each of the cultivars. Compare the outlier maps.
- Apply robust PCA to the whole data set and to each of the cultivars (which robust PCA methods can you use?). Compare the outlier maps.
- How many components will you retain?

Outline

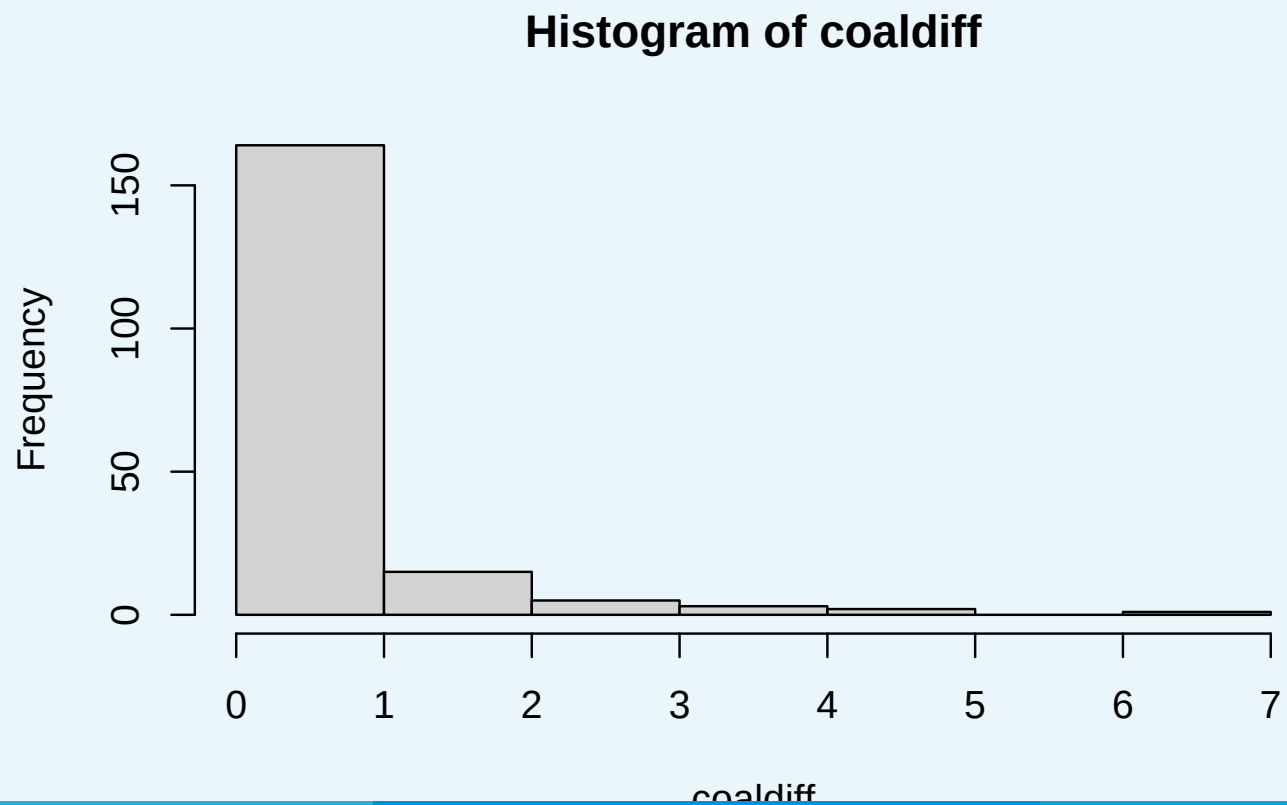
Adjusted boxplot

- The boxplot is a very popular graphical tool to visualize the distribution of continuous unimodal data
- When the data are skewed, usually many points exceed the whiskers and are often erroneously declared as outliers.
- An adjustment of the boxplot that includes a robust measure of skewness in the determination of the whiskers (Hubert and Vandervieren, 2006)
- This results in a more accurate representation of the data and of possible outliers.
- Function `adjbox()` in package **robustbase**.

Adjusted boxplot

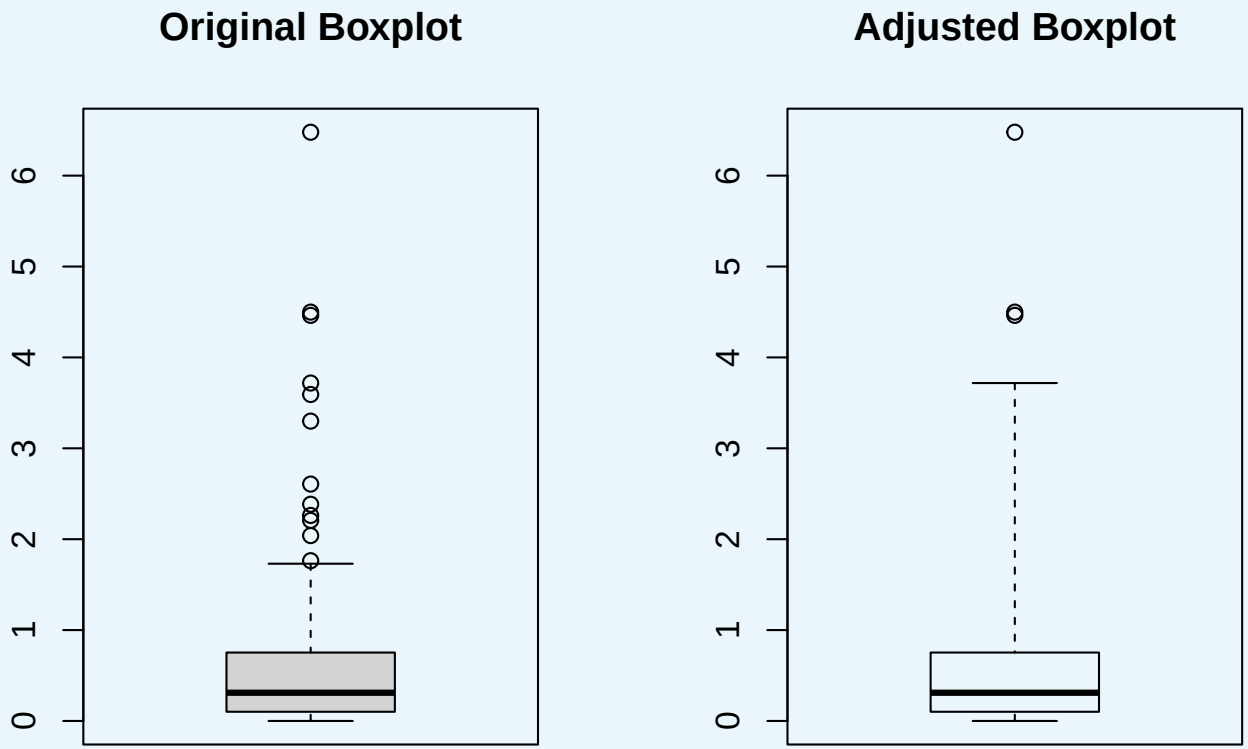
Example: Dates of Coal Mining Disasters, available in package **boot**

```
> ### Hubert and Vandervieren (2006), p. 10, Fig. 4.
> data(coal, package = "boot")
> coaldiff <- diff(coal$date)
> hist(coaldiff)
```



Adjusted boxplot

```
> op <- par(mfrow = c(1,2))
> boxplot(coaldiff, main = "Original Boxplot")
> adjbox(coaldiff, main  = "Adjusted Boxplot")
> par(op)
```



Adjusted outlyingness (AO)

- The Stahel-Donoho outlyingness (SDO) used in ROBPCA assumes symmetry!
- Too many observations might be marked as outliers
- Adjusted outlyingness: For univariate data with median M, the adjusted outlyingness is defined as:

$$AO_i = \frac{|x_i - M|}{(w_2 - M)I[x_i > M] + (M - w1)I[x_i < M]}$$

- where with w_1 and w_2 are the lower and upper whiskers of the adjusted boxplot.
- Skewness is thus used to estimate the scale differently on both sides of the median.

Adjusted outlyingness (AO)

- For multivariate data, the projection pursuit idea can again be used (Brys et al. 2005; Hubert and Van der Veeken 2008):

$$AO_i = \sup_{a \in R^p} AO(\mathbf{a}^\top \mathbf{x}_i, \mathbf{X}_n \mathbf{a})$$

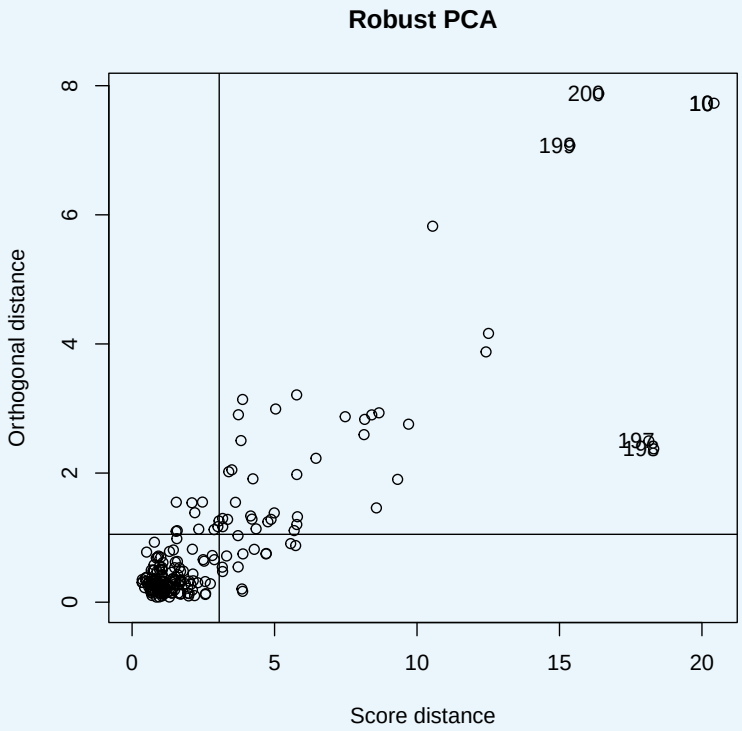
- In practice: consider $250p$ directions, generated as the direction perpendicular to the subspace spanned by p observations, randomly drawn from the data set.

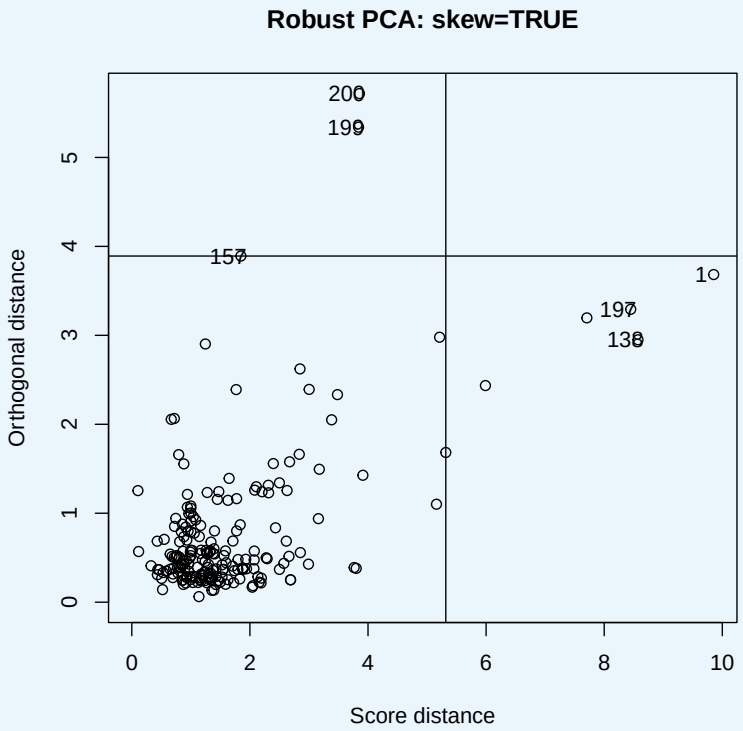
PCA for skewed data: Example

```
> library(rrcov)
> data(machines)
> rownames(machines) <- NULL      # names are too long
> ##  ?machines
```

- machines {rrcov}
- Computer Hardware data
- Description: A data set containing relative CPU performance data of 209 machines on 8 variables.
- Source: UCI Archive

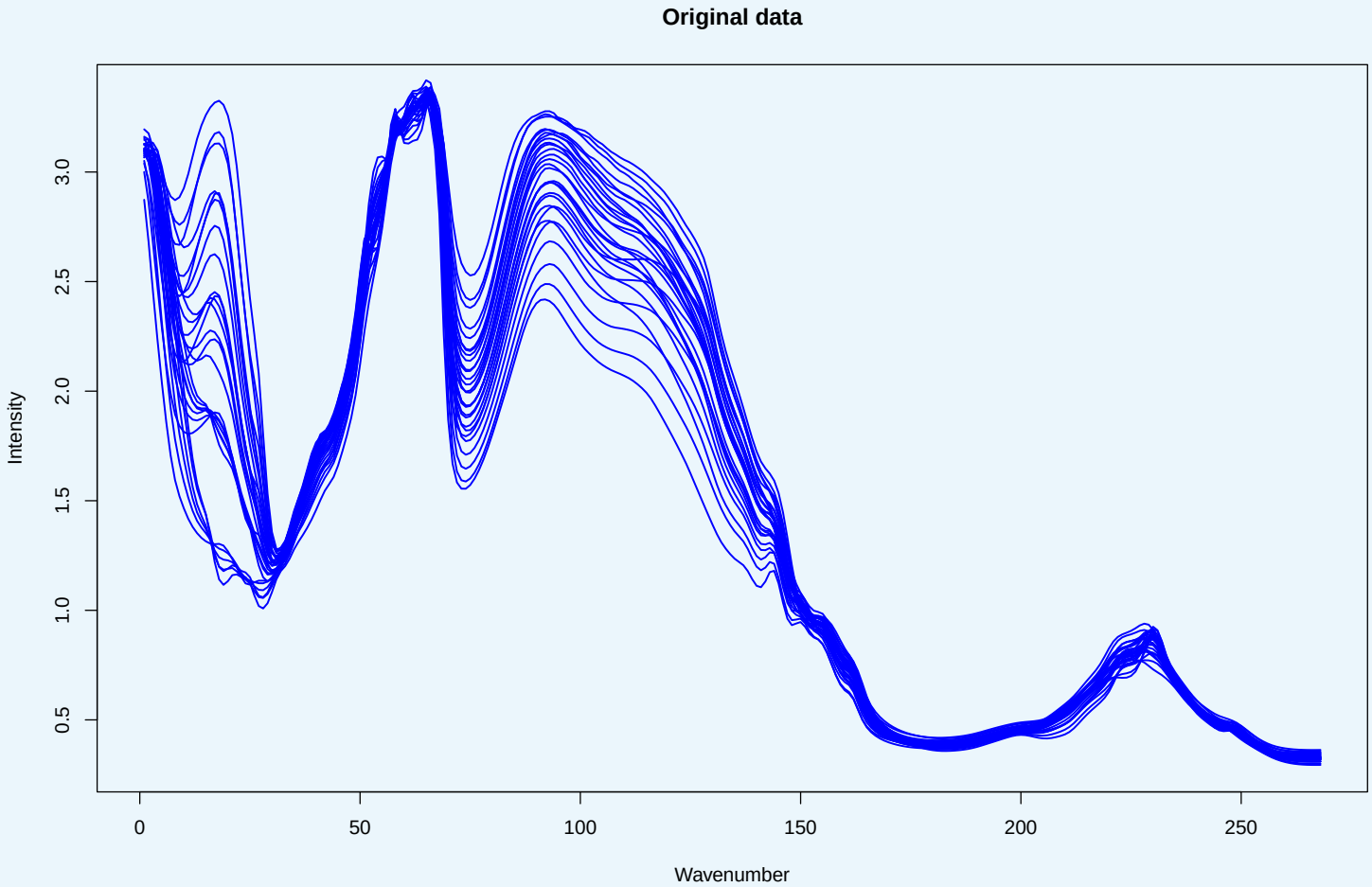
PCA for skewed data: Example





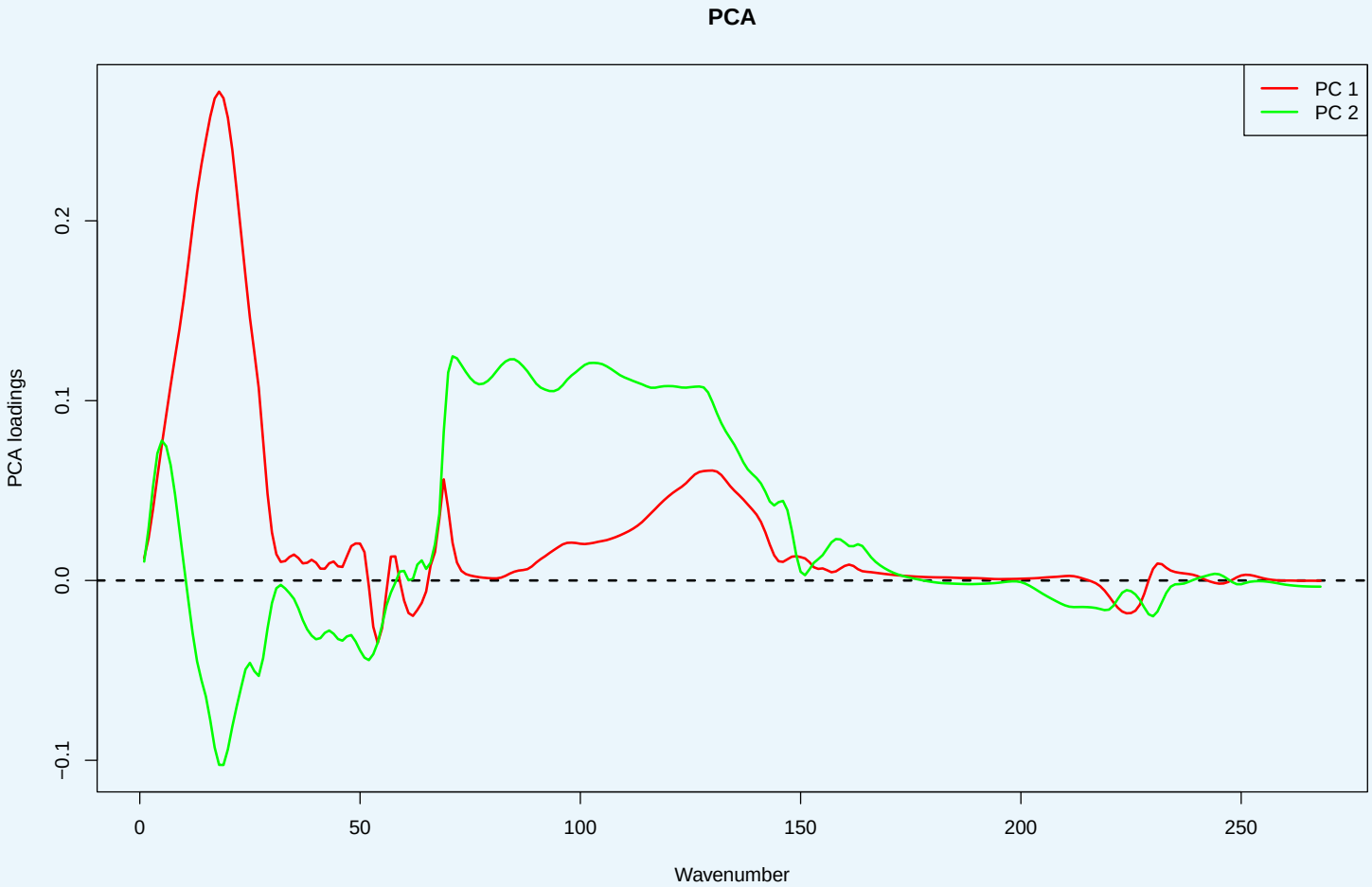
Sparse PCA: Example

Data set yarn: 28 near-infrared spectra (NIR) of PET yarns, measured at 268 wavelengths as predictors, and density as response (density)—Swierenga *et al.* (1999), package **pls**.



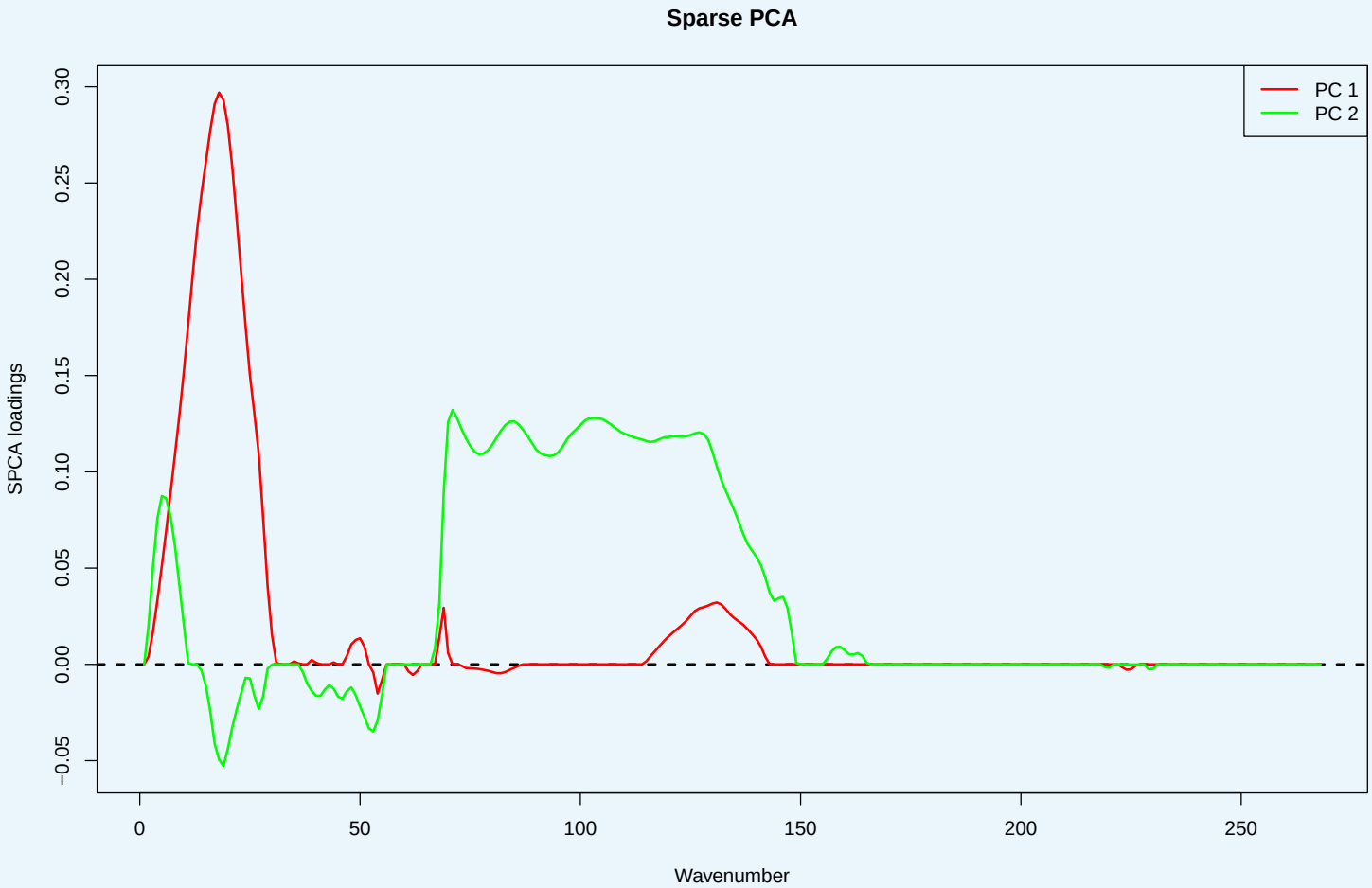
Sparse PCA: Example

First two PCA loadings:



Sparse PCA: Example

First two sparse PCA loadings:



Sparse and robust PCA - Croux at al. (2012)

... add an L1 penalty in the objective function (see Tibshirani, 1996)

$$\mathbf{p}_j = \arg\max_{\mathbf{p}} \mathrm{Var}(\mathbf{Xp}) - \lambda_j \|\mathbf{p}\|_1$$

- with the tuning parameter λ_j:
  - Setting $\lambda_j = 0$: unconstrained j-th PCA direction $\mathbf{p}_j$;
  - increasing λ_j: sparsity gains importance compared to (robust) variance maximization.
- To achieve robustness: replace *Var* by a robust scale estimator: Median Absolute Deviation (MAD) or Q_n-scale, see Croux et al. (2005, 2007).
- To solve this optimization problem - use the grid algorithm Croux et al (2007) available as function `PcaGrid()` in the R package **rrcov**.

Sparse PCA: Example

```
>     library(mvtnorm)         # multivariate data with outliers
>     set.seed(6543)
>     x <- rbind(rmvnorm(200, rep(0, 6), diag(c(5, rep(1,5)))),
+                rmvnorm( 15, c(0, rep(20, 5)), diag(rep(1, 6))))
>     lambda <- c (0.23, 0.34, 0.005) #  applying sparse PCA
>     SPcaGrid (x, k = 3, lambda = lambda)

Call:
SPcaGrid(x = x, k = 3, lambda = lambda)

Standard deviations:
[1] 2.416348 1.389316 1.348614
```

Outline

The discriminant analysis

- A data set $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_N)$ of dimension p is split into K groups, each with n_k objects, $k = 1, \ldots, K$ and $N = \sum n_k$ (training data set).
- The task: Assign new observations (test data set) to one of the groups
- π_k are prior probabilities
- Conditional distributions $N(\boldsymbol{\mu}_\mathbf{k}, \boldsymbol{\Sigma}_\mathbf{k})$

The discriminant analysis

- The (Bayesian) discriminant rule is (quadratic discriminant analysis):

$$d_k(\mathbf{x}) = -\frac{1}{2} log(|\Sigma_k|) - \frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1}(\mathbf{x} - \boldsymbol{\mu}_k) + log(\pi_k)$$

- In case of homoscedasticity $\boldsymbol{\Sigma} = \boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\Sigma}_k$ we have (linear discriminant analysis):

$$d_k(\mathbf{x}) = \boldsymbol{\mu}_k^T \boldsymbol{\Sigma}_k^{-1} \mathbf{x} - -\frac{1}{2} \boldsymbol{\mu}_k^T \boldsymbol{\Sigma}_k^{-1} \boldsymbol{\mu}_k + log(\pi_k)$$

The discriminant analysis: Parameter estimation

- The parameters $\boldsymbol{\mu}_k$, $\boldsymbol{\Sigma}_k$ and $\boldsymbol{\Sigma}$ are estimated from the sample:
  - the sample means $\hat{\mathbf{x}}_k$
  - the inverse of the sample covariance matrices $\hat{\boldsymbol{\Sigma}}_k$
  - the inverse of the pooled sample covariance matrix $\hat{\boldsymbol{\Sigma}} = \frac{1}{N-K} \sum_{k=1}^{K} n_k \boldsymbol{\Sigma}_k$
- Instead of pooling the covariance matrices one can center the data by subtracting $\hat{\mathbf{x}}_k$ from the data and then computing the sample covariance matrix of the centered data
- Robust discriminant analysis is obtained by replacing $\boldsymbol{\mu}_k$, $\boldsymbol{\Sigma}_k$ and $\boldsymbol{\Sigma}$ by their robust estimates $\mathbf{T}_k$, $\mathbf{C}_k$ and $\mathbf{C}$

The discriminant analysis: Parameter estimation

- The parameters $\boldsymbol{\mu}_k$, $\boldsymbol{\Sigma}_k$ and $\boldsymbol{\Sigma}$ are estimated from the sample:
  - the sample means $\hat{\mathbf{x}}_k$
  - the inverse of the sample covariance matrices $\hat{\boldsymbol{\Sigma}}_k$
  - the inverse of the pooled sample covariance matrix $\hat{\boldsymbol{\Sigma}} = \frac{1}{N-K} \sum_{k=1}^{K} n_k \boldsymbol{\Sigma}_k$
- Instead of pooling the covariance matrices one can center the data by subtracting $\hat{\mathbf{x}}_k$ from the data and then computing the sample covariance matrix of the centered data
- Robust discriminant analysis is obtained by replacing $\boldsymbol{\mu}_k$, $\boldsymbol{\Sigma}_k$ and $\boldsymbol{\Sigma}$ by their robust estimates $\mathbf{T}_k$, $\mathbf{C}_k$ and $\mathbf{C}$

Robust DA Example: Reaven and Miller diabetes data

```
> library(rrcov)
> data(diabetes)
> ##  ?diabetes
```

- diabetes {rrcov}
- Reaven and Miller diabetes data
- Description: The data set contains five measurements made on 145 non-obese adult patients classified into three groups.
- Reaven and Miller, following Friedman and Rubin (1967), applied cluster analysis to the three primary variables and identified three clusters: "normal", "chemical diabetic", and "overt diabetic" subjects.
- The column group contains the classifications of the subjects into these three groups, obtained by current medical criteria.

Robust DA Example: diabetes

```
> dim(diabetes)

[1] 145   6

> head(diabetes)

    rw fpg glucose insulin sspg  group
1 0.81  80     356     124   55 normal
2 0.95  97     289     117   76 normal
3 0.94 105     319     143  105 normal
4 1.04  90     356     199  108 normal
5 1.00  90     323     240  143 normal
6 0.76  86     381     157  165 normal

> table(diabetes$group)

  normal chemical    overt
      76       36       33
```

Robust DA Example: diabetes

```
> ### Classical LDA
> lda <- LdaClassic(group~insulin+glucose+sspg, data=diabetes)
> (pr <- predict(lda))

Apparent error rate 0.1724

Classification table
          Predicted
Actual     normal chemical overt
  normal       73        3     0
  chemical     15       21     0
  overt         2        5    26

Confusion matrix
          Predicted
Actual     normal chemical overt
  normal    0.961    0.039 0.000
  chemical  0.417    0.583 0.000
  overt     0.061    0.152 0.788
```

Robust DA Example: diabetes

```
> ## Robust LDA (using MCD group means and covariance matrix)
> rlda <- Linda(group~insulin+glucose+sspg, data=diabetes)
> (rpr <- predict(rlda))

Apparent error rate 0.1103

Classification table
          Predicted
Actual     normal chemical overt
  normal       73        3     0
  chemical      6       30     0
  overt         0        7    26

Confusion matrix
          Predicted
Actual     normal chemical overt
  normal    0.961    0.039 0.000
  chemical  0.167    0.833 0.000
  overt     0.000    0.212 0.788
```

Robust DA Example: diabetes

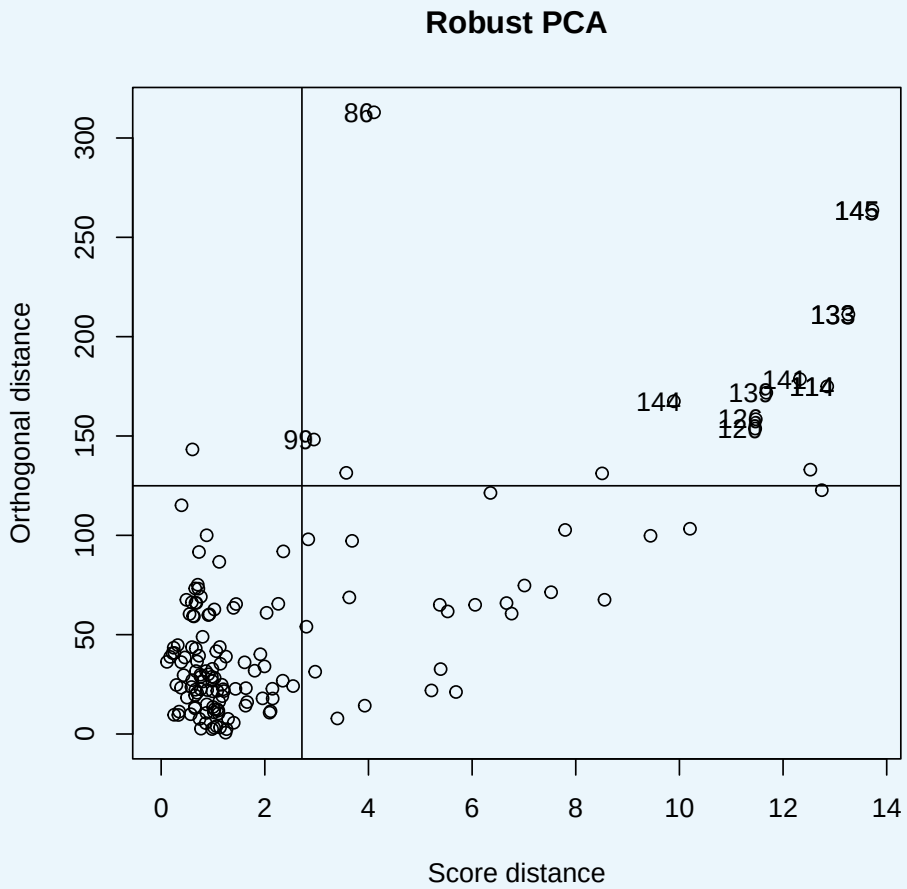
```
> ## Robust LDA (using OGK group means and covariance matrix)
> rlda.OGK <- Linda(group~insulin+glucose+sspg, data=diabetes,
+          cov.control=CovControlOgk())
> (rpr <- predict(rlda.OGK))

Apparent error rate 0.0966

Classification table
          Predicted
Actual     normal chemical overt
  normal       71        5     0
  chemical      2       34     0
  overt         0        7    26

Confusion matrix
          Predicted
Actual     normal chemical overt
  normal    0.934    0.066 0.000
  chemical  0.056    0.944 0.000
  overt     0.000    0.212 0.788
```

Robust DA Example: diabetes

```
> ## Robust QDA with MCD (default)
> rqda <- QdaCov(group~insulin+glucose+sspg, data=diabetes)
> (qpr <- predict(rqda))

Apparent error rate 0.1034

Classification table
          Predicted
Actual     normal chemical overt
  normal       69        7     0
  chemical      2       30     4
  overt         0        2    31

Confusion matrix
          Predicted
Actual     normal chemical overt
  normal    0.908    0.092 0.000
  chemical  0.056    0.833 0.111
  overt     0.000    0.061 0.939
```

Robust DA Example: diabetes

```
> ## Outlier map with ROBPCA
> plot(PcaHubert(diabetes[, -ncol(diabetes)]), id.n.od=10)
```



Outline

Summary and conclusions

- Robust PCA functions in R
  - PcaHubert - ROBPCA, (Hubert, Rousseeuw and Vanden Branden, 2005)
  - PcaLocantore - Spherical PCA (Locantore et *al.*, 1999)
  - PcaCov - PCA based on a robust covariance matrix
  - PcaGrid - (Croux, Filzmoser, Oliveira, 2007)
  - PcaProj - (Croux and Ruiz-Gazen, 2005)
  - PCA based on robust scales (Maronna, 2005)
  - PCA based on generalized sign covariance matrix (Raymaekers and Rousseeuw, 2019)

References I

Brys G, Hubert M, Struyf A (2004).
A Robust Measure of Skewness.
Journal of Computational and Graphical Statistics, **13**(4), 996–1017.

Croux C, Filzmoser P, Oliveira M (2007).
Algorithms for Projection-pursuit Robust Principal Component Analysis.
Chemometrics and Intelligent Laboratory Systems, **87**(218), 218–225.

Croux C, Haesbroeck G (2000).
Principal Components Analysis Based on Robust Estimators of the Covariance or Correlation Matrix: Influence Functions and Efficiencies.
Biometrika, **87**, 603–618.

Croux C, Ruiz-Gazen A (2005).
High Breakdown Estimators for Principal Components: The Projection-pursuit Approach Revisited.
Journal of Multivariate Analysis, **95**, 206–226.

Filzmoser P, Todorov V (2011).
Review of robust multivariate statistical methods in high dimension.
Analytica Chimica Acta, **705**, 2–14.

References II


Hubert M, Rousseeuw P, Vanden Branden K (2005).
ROBPCA: A New Approach to Robust Principal Component Analysis.
Technometrics, **47**, 64–79.


Hubert M, Rousseeuw PJ, van Aelst S (2008).
High-Breakdown Robust Multivariate Methods.
Statistical Science, **23**, 92–119.


Jolliffe IT (2002).
Principal Component Analysis.
Springer-Verlag, New York, NY, USA.


Locantore N, Marron J, Simpson D, Tripoli N, Zhang J, Cohen K (1999).
Robust Principal Components for Functional Data.
Test, **8**, 1–28.


Maronna RA, Martin D, Yohai V, Salibian-Barrera M (2019).
Robust Statistics: Theory and Methods (with R): Second edition.
John Wiley & Sons, New York.

References III

Raymaekers J, Rousseeuw P (2019).
A generalized spatial sign covariance matrix.
Journal of Multivariate Analysis, **171**, 94–111.

Todorov V, Filzmoser P (2014).
Software Tools for Robust Analysis of High-Dimensional Data.
Austrian Journal of Statistics, **43**, 255–266.

Todorov V, Filzmoser P (2009).
An Object Oriented Framework for Robust Multivariate Analysis.
Journal of Statistical Software, **32**(3), 1–47.