ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Cluster analysis, cluster validation, and the German Bundestag elections

Christian Hennig

1. Introduction

2. The clustering methods
   - Gaussian model-based clustering
   - Average Linkage hierarchical clustering

3. Analysis
   - Gaussian model-based clustering
   - Validation - visualisation
   - Validation - stability
   - Average Linkage

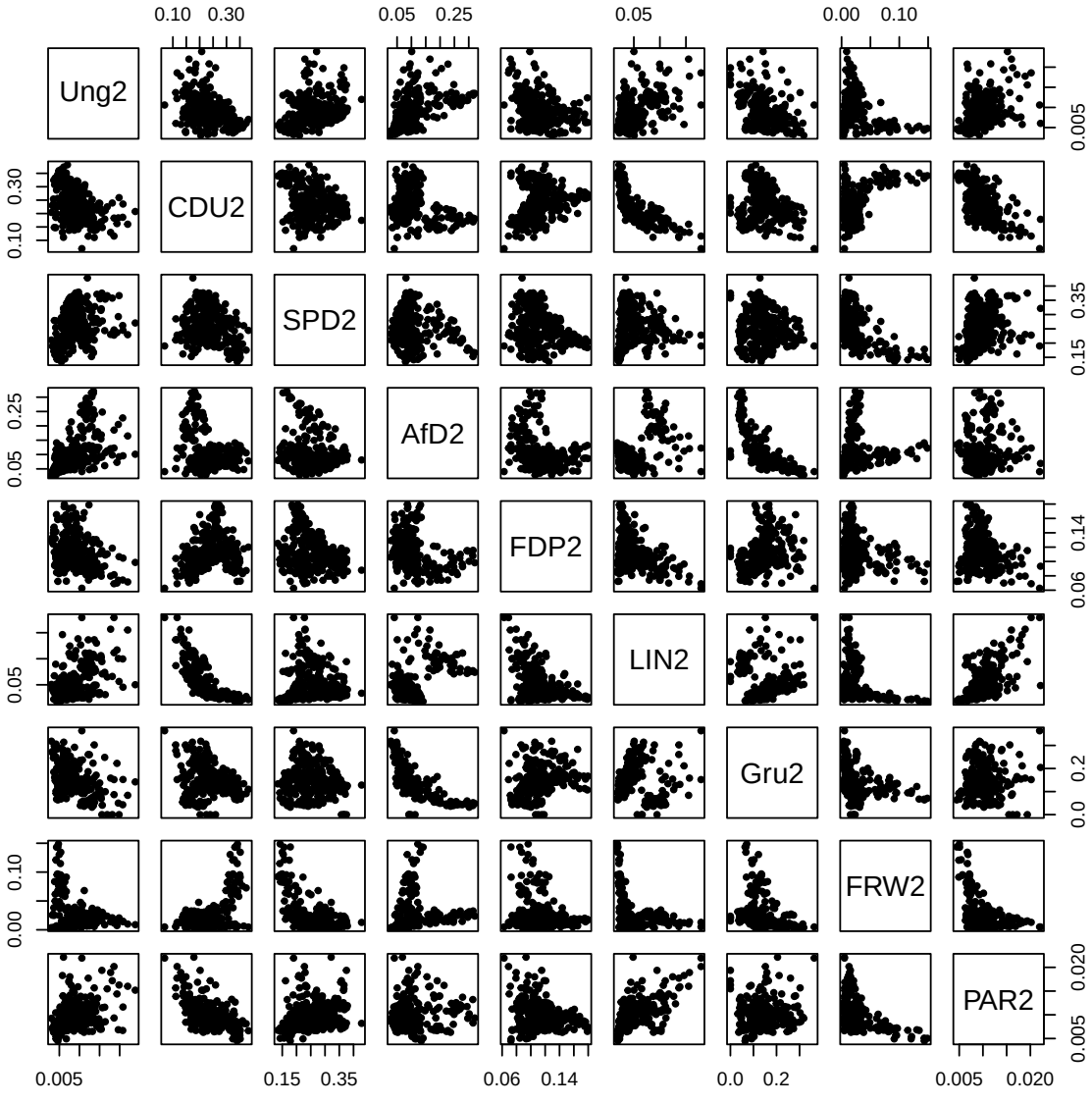
4. Conclusion

1. Introduction

Cluster analysis: Finding groups in data.

Bundestagswahl 2021, data (pre-processed) from
https://www.bundeswahlleiter.de/bundestagswahlen/2021/ergebnisse/opendata.html,
Bundestag21Zweitstimmen.dat

```
> str(wresults2)
'data.frame': 299 obs. of  9 variables:
 $ Ung2: num  0.00826 0.00854 0.00873 0.00682 0.00589 ...
 $ CDU2: num  0.202 0.244 0.241 0.238 0.153 ...
 $ SPD2: num  0.253 0.261 0.275 0.266 0.259 ...
 $ AfD2: num  0.0573 0.0605 0.0829 0.0652 0.0491 ...
 $ FDP2: num  0.107 0.126 0.136 0.119 0.103 ...
 $ LIN2: num  0.0415 0.0305 0.0332 0.0319 0.0605 ...
 $ Gru2: num  0.185 0.155 0.141 0.178 0.283 ...
 $ FRW2: num  0.00952 0.00744 0.01257 0.00724 0.00612 ...
 $ PAR2: num  0.01024 0.00909 0.00988 0.00998 0.01401 ...
```


Ung2
CDU2
SPD2
AfD2
FDP2
LIN2
Gru2
FRW2
PAR2
0.10 0.30
0.05 0.25
0.05
0.00 0.10
0.005
0.10 0.30
0.15 0.35
0.05 0.25
0.06 0.14
0.05
0.0 0.2
0.00 0.10
0.005 0.020
0.005
0.15 0.35
0.06 0.14
0.0 0.2
0.005 0.020

There are many cluster analysis methods,
and on many datasets these may produce
many different clusterings.

Clustering may have different aims.
Different clusterings on the same data
may be appropriate for different aims.

Particularly: Within-cluster homogeneity
vs. between-cluster separation.
Both are often relevant,
but may be in conflict.

Bundestagswahl-data:

May help journalists and politologists to analyse results.

Clusters should not be all too heterogeneous,
but uniform homogeneity is not required.
Separation should be picked up
where existing and meaningful,
but is not necessarily required either.

Cluster validation

Generally concerned with
evaluating the quality of clusterings.

Most clustering methods always deliver a clustering,
but these are not necessarily meaningful.

May also compare different clusterings
(also different numbers of clusters) regarding quality.

Approaches for cluster validation
(and understanding/interpreting the clustering)

- (Use of external information)
- Visual exploration
- Stability assessment
- (Internal validation indexes)
- (Testing for clustering structure)
- (Sensitivity analysis and comparison of different clusterings on same dataset)

2. The clustering methods

2.1 Gaussian model-based clustering (Bouveyron et al. (2019))

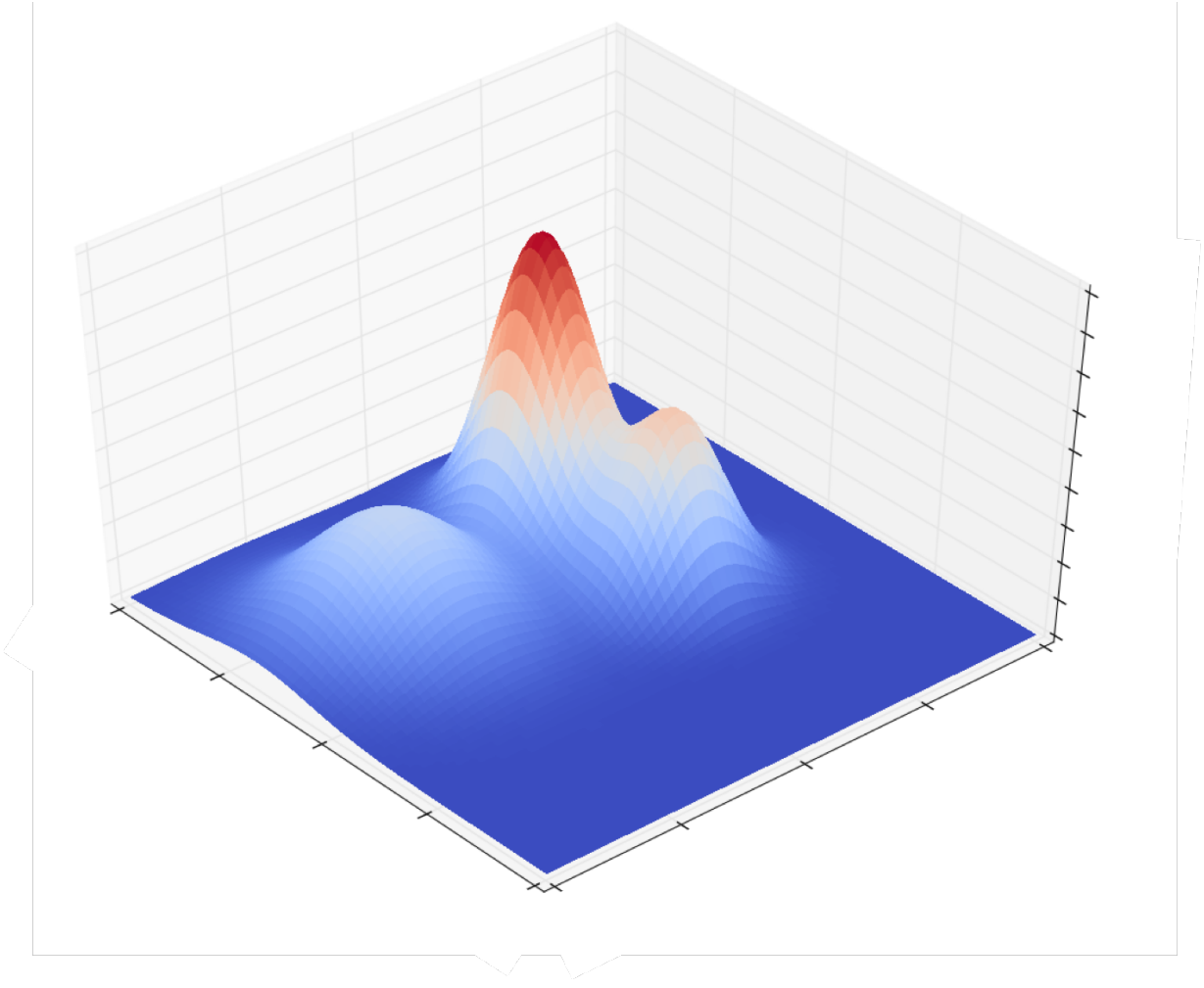
Gaussian mixture model, observations assumed i.i.d. with

$$f_\theta(\mathsf{x}) = \sum_{j=1}^{k} \pi_j \varphi_{\mathsf{a}_j, \Sigma_j}(\mathsf{x}).$$

Clusters are described by Gaussian distributions.
Elliptical clusters, flexible size and shape.

For fixed k, estimate $\theta = (\pi_j, \mathsf{a}_j, \Sigma_j)_{j=1,\ldots,k}$
by maximum likelihood (EM-algorithm).

Can then estimate probability that x_i belongs to cluster j,
use this to classify x_i.



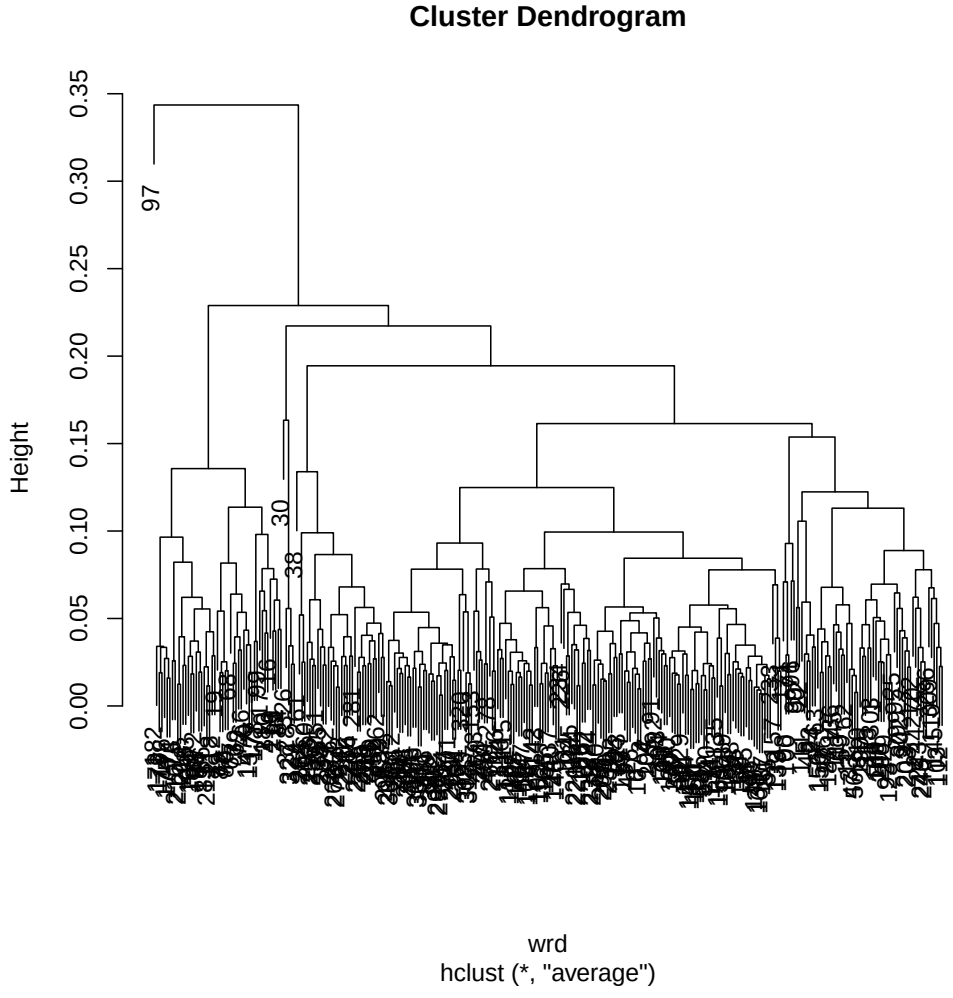
$$f_\theta(\mathsf{x}) = \sum_{j=1}^{k} \pi_j \varphi_{\mathsf{a}_j, \Sigma_j}(\mathsf{x}).$$

Estimate "model complexity" k by maximising
Bayesian Information Criterion ($v(k, \theta)$ degrees of freedom):

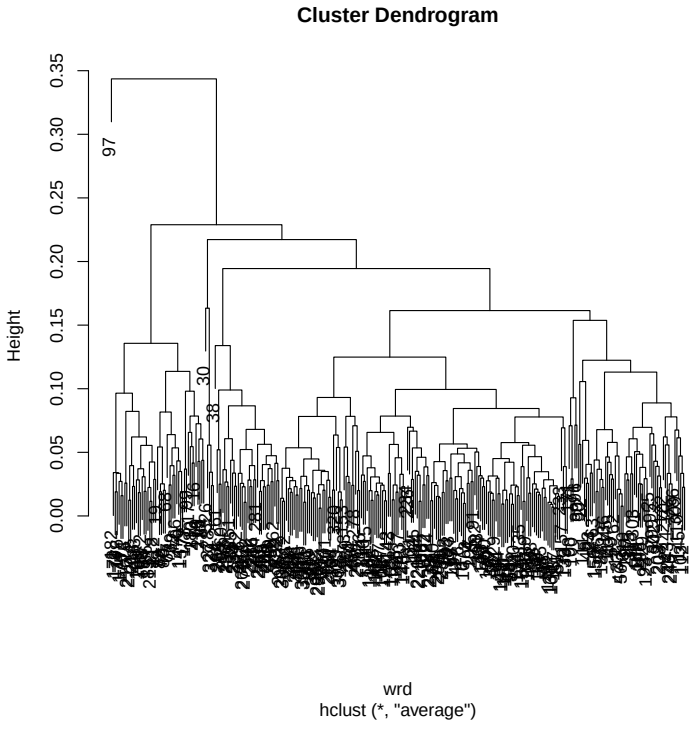
$$\mathrm{BIC}(n, k, \theta) = 2 \log \hat{L}_{n,k,\theta} - v(k, \theta) \log(n).$$

R-mclust package does this, also
covariance matrix modelling, i.e.,
can assume $\Sigma_1 = \ldots = \Sigma_k$ for reducing v, also
Σ_j spherical, diagonal, equal eigenvectors etc.
Use BIC to estimate covariance matrix constraints.

2.2 Average Linkage hierarchical clustering



$$D(B, C) = \frac{1}{|B||C|} \sum_{x \in B, y \in C} d(x, y)$$



Partition by cutting dendrogram at k clusters.
Flexible, can find non-elliptical clusters.
Will isolate separated subsets (even one point),
some homogeneity, but often uneven cluster sizes.

Method for deciding k (internal validation index):

Average silhouette width (ASW)

(Kaufman and Rousseeuw (1990))

$sw(i, \mathcal{C}) = \frac{b(i,\mathcal{C}) - a(i,\mathcal{C})}{\max(a(i,\mathcal{C}), b(i,\mathcal{C}))}$,

$$a(i, \mathcal{C}) = \frac{1}{|C_j| - 1} \sum_{x \in C_j} d(x_i, x), \;\; b(i, \mathcal{C}) = \min_{x_i \notin C_l} \frac{1}{|C_l|} \sum_{x \in C_l} d(x_i, x).$$

Maximum average $sw \Rightarrow$ good $\mathcal{C}$.

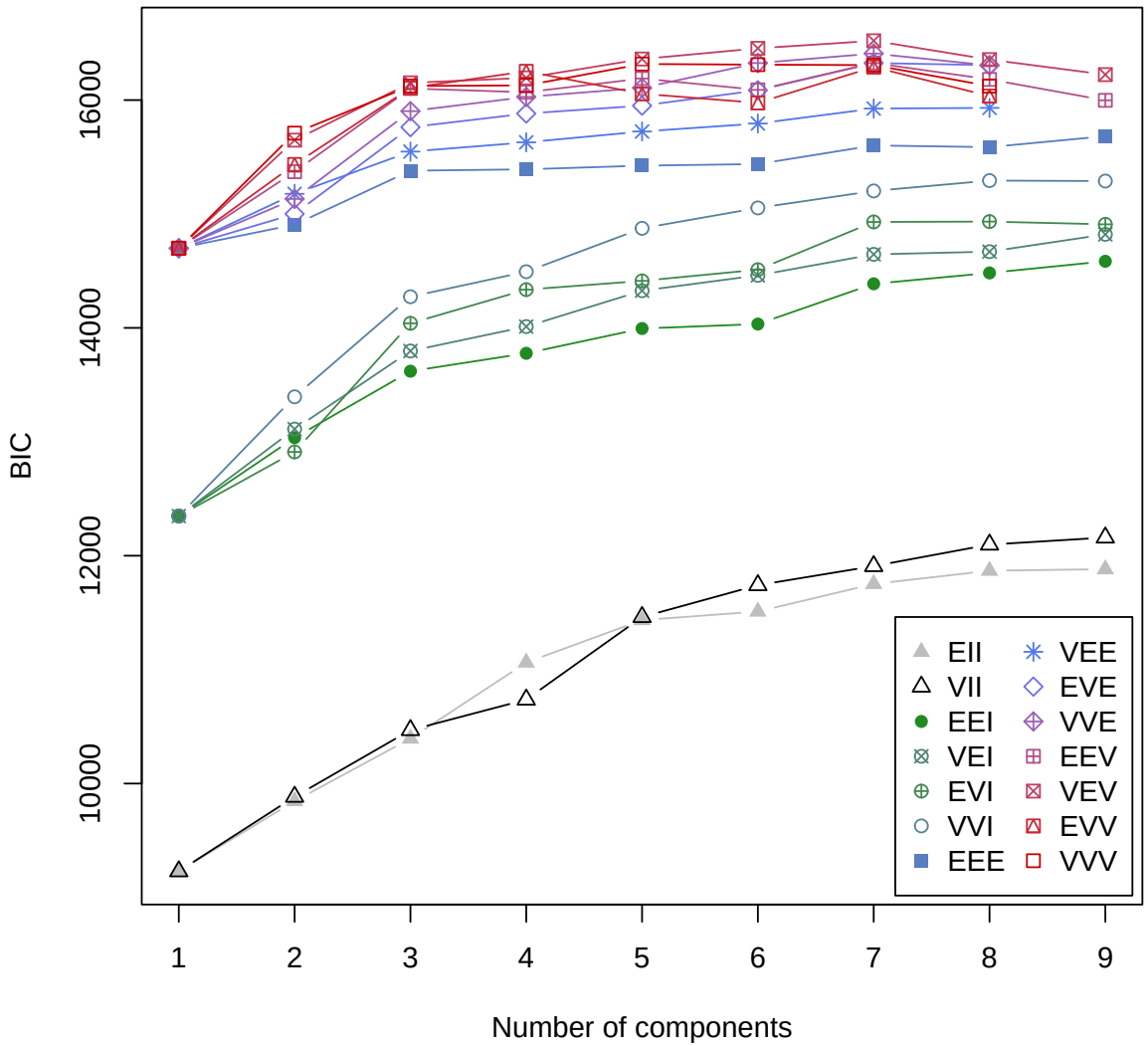
This contrasts within-cluster homogeneity
with separation from neighbouring clusters.

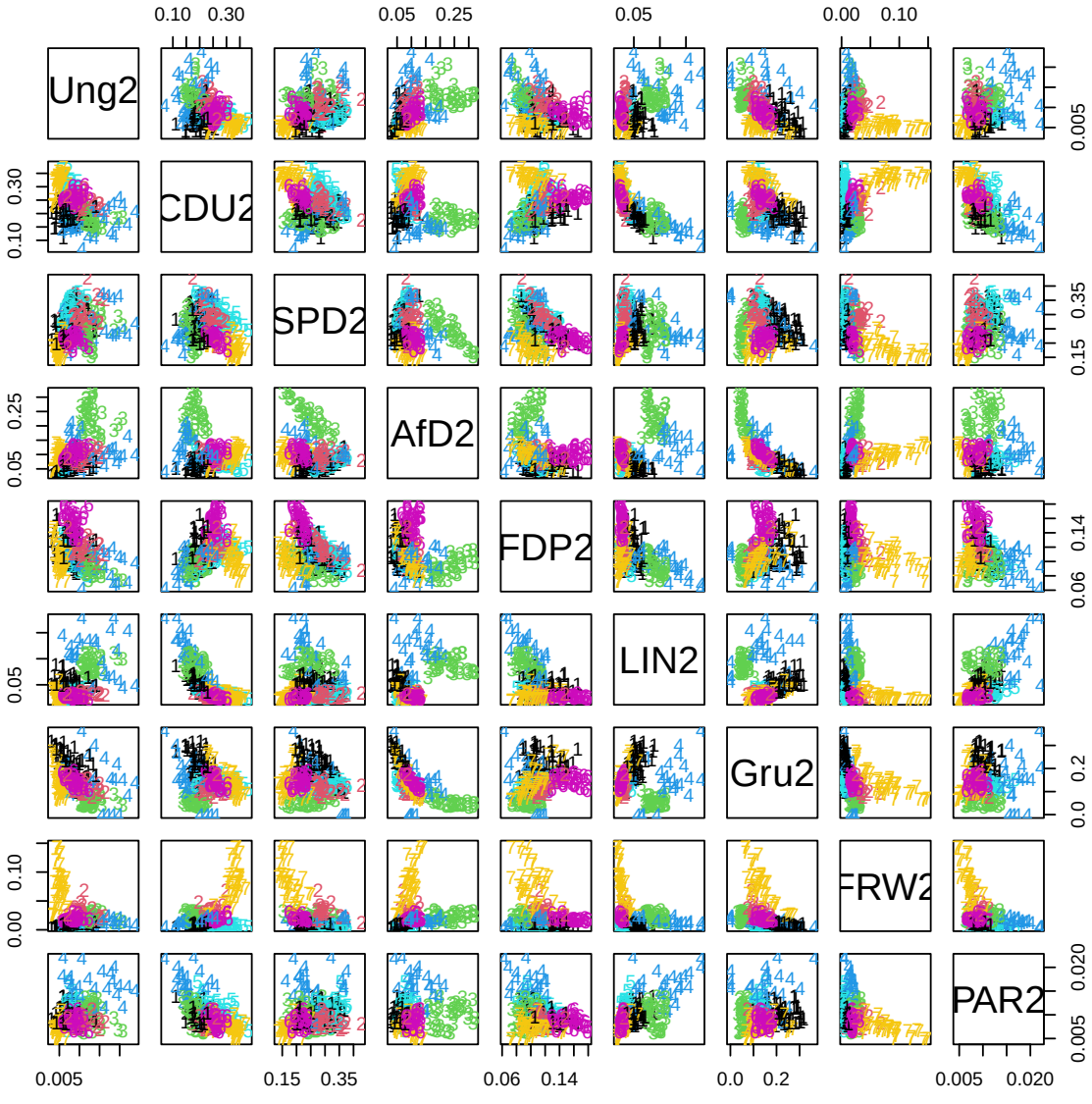
3. Analysis
3.1 Gaussian model-based clustering

```
library(mclust)
library(fpc)

mresults2 <- Mclust(wresults2, G=1:9)
summary(mresults2)

# ----------------------------------------------------
# Gaussian finite mixture model fitted by EM algorithm
# ----------------------------------------------------
#
# Mclust VEV (ellipsoidal, equal shape) model with 7 components:
#
# log-likelihood   n  df      BIC      ICL
#       9217.935 299 336 16520.52 16513.92
#
# Clustering table:
#  1  2  3  4  5  6  7
# 57 34 38 29 63 32 46
```

Visualisation (I)

```
plot(mresults2) # Several plots offered, this is number 1
pairs(wresults2,pch=clusym[mresults2$classification],
  col=mresults2$classification)
```


BIC
16000
14000
12000
10000
1
2
3
4
5
6
7
8
9
Number of components
EII
VII
EEI
VEI
EVI
VVI
EEE
VEE
EVE
VVE
EEV
VEV
EVV
VVV


Ung2
CDU2
SPD2
AfD2
FDP2
LIN2
Gru2
FRW2
PAR2
0.10 0.30
0.05 0.25
0.05
0.00 0.10
0.005
0.10 0.30
0.15 0.35
0.05 0.25
0.06 0.14
0.05
0.0 0.2
0.00 0.10
0.005 0.020
0.005
0.15 0.35
0.06 0.14
0.0 0.2
0.005 0.020

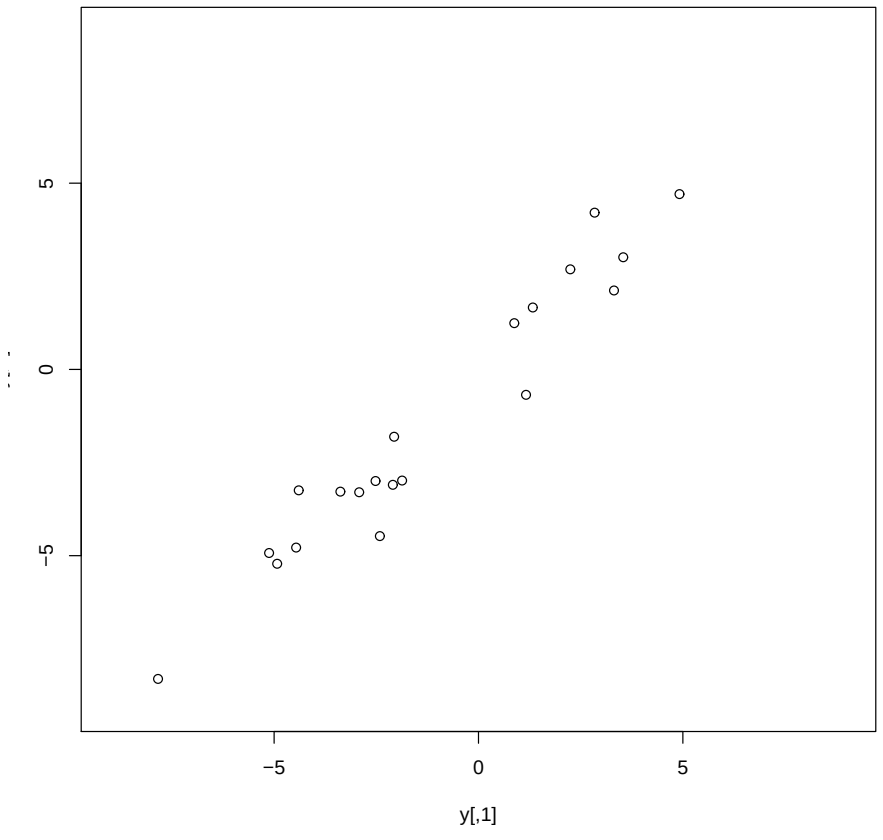
Use of external information (example)

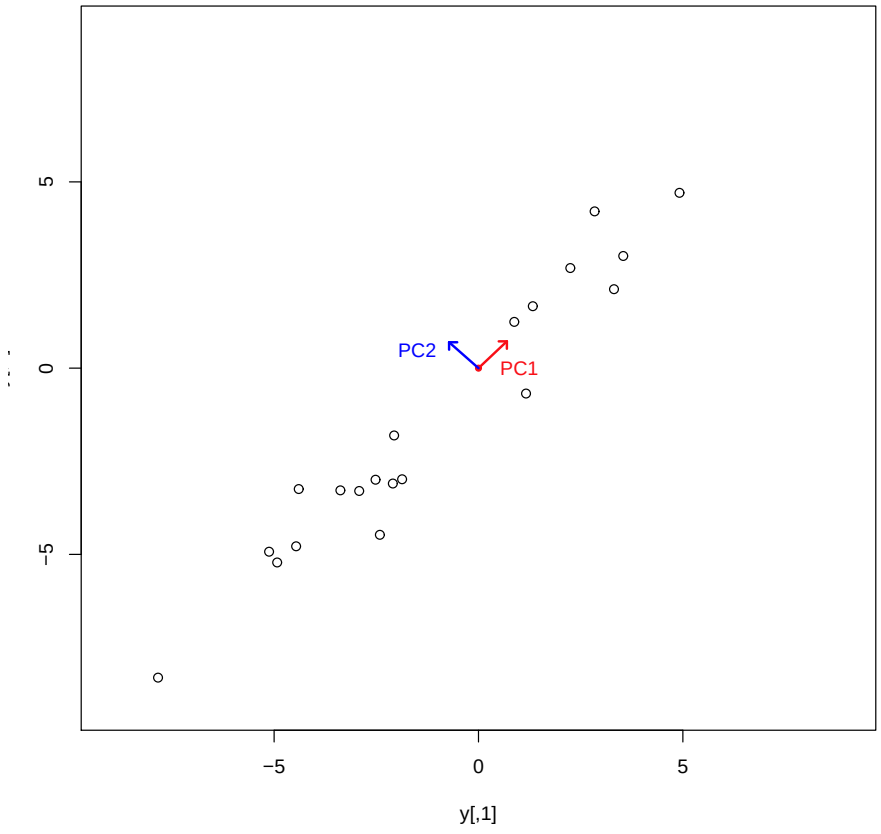
```
> table(mresults2$classification,land)
   land
    Baden-Württemberg Bayern Berlin Brandenburg Bremen Hamburg Hessen
  1                 8      0      2           0      2       6      7
  2                 1      0      0           0      0       0     10
  3                 0      0      0           9      0       0      0
  4                 0      0     10           1      0       0      2
  5                 0      0      0           0      0       0      0
  6                29      0      0           0      0       0      3
  7                 0     46      0           0      0       0      0
   land
    Mecklenburg-Vorpommern Niedersachsen Nordrhein-Westfalen Rheinland-Pfalz
  1                      0             9                  17               0
  2                      0             9                   0              14
  3                      5             0                   0               0
  4                      1             0                   1               1
  5                      0            12                  46               0
  6                      0             0                   0               0
  7                      0             0                   0               0
   land
    Saarland Sachsen Sachsen-Anhalt Schleswig-Holstein Thüringen
  1        0       0              0                  6         0
  2        0       0              0                  0         0
  3        0      11              7                  0         6
  4        4       5              2                  0         2
  5        0       0              0                  5         0
  6        0       0              0                  0         0
  7        0       0              0                  0         0
```

Visualisation (II)

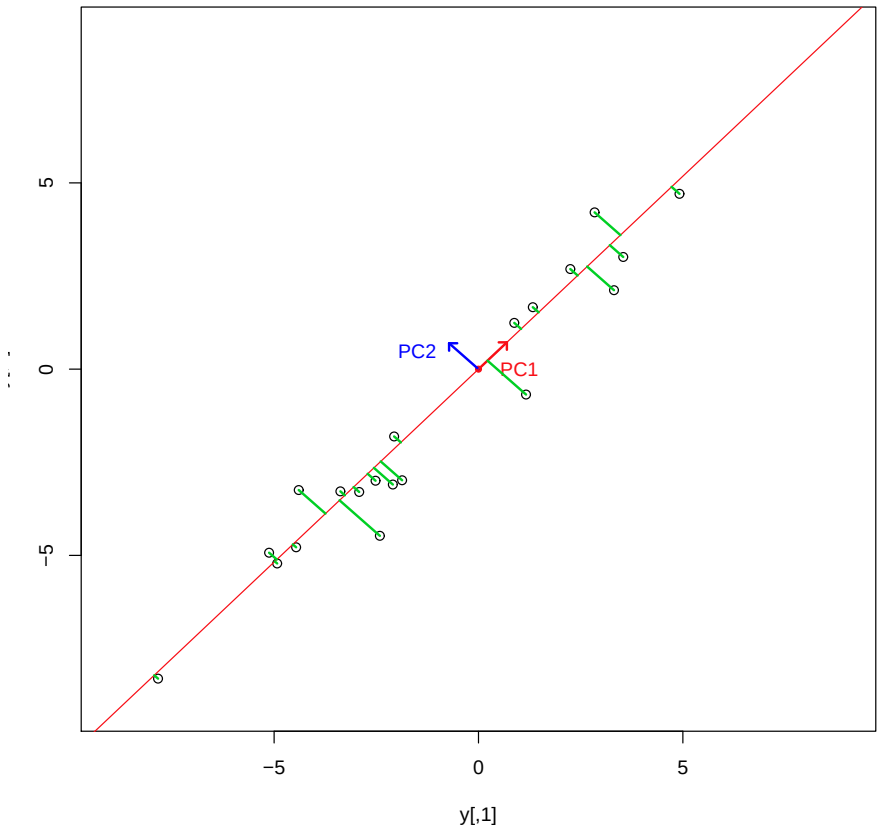
For high-dimensional data may want low-d (2-d) visualisation.
Principal Components Analysis: *Projection method*,
projects higher dimensional data onto low-d.

Toy example: Find PCs in 2-d data.
(Really we want to find PCs in high-d data.)

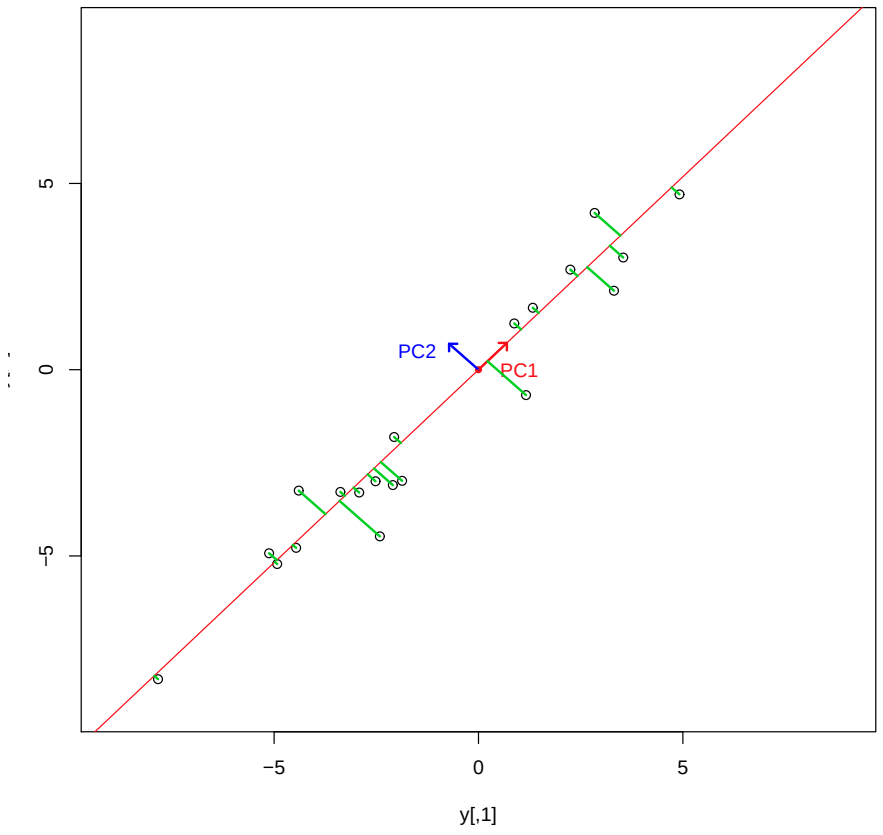




Amounts to *changing coordinate system*.

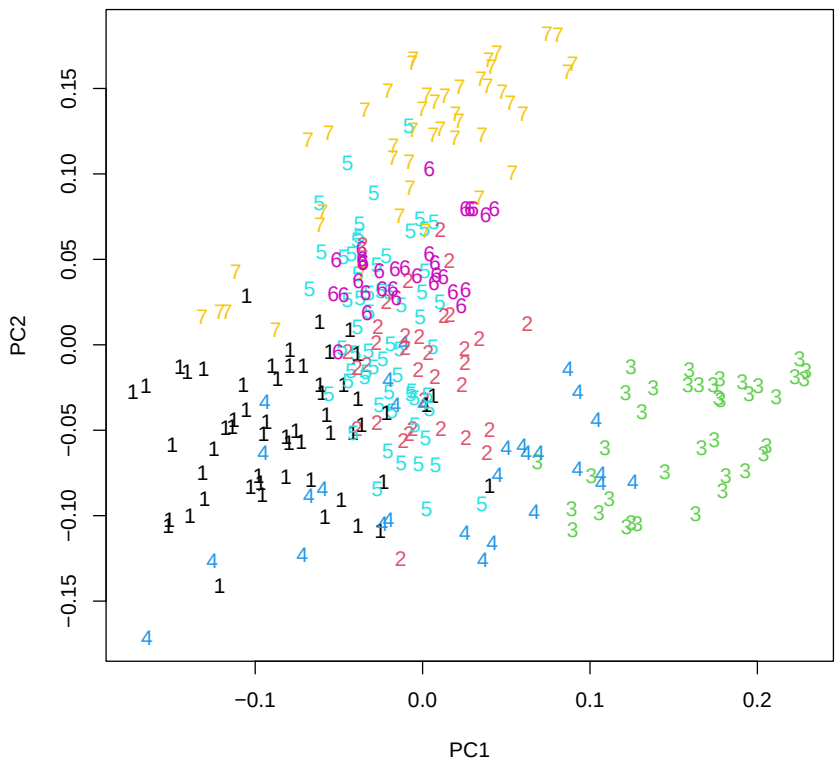


PC-Values of points are *projections* on PCs.



Objective function: Maximise variance along PC1 (and then along PC2 assuming orthogonality).
Here variance of PC1 is 98.5% of sum of all variances.

```
pcwresults <- prcomp(wresults2)
plot(pcwresults$x,pch=clusym[mresults2$classification],
   col=mresults2$classification)
```



Problem with PCA:

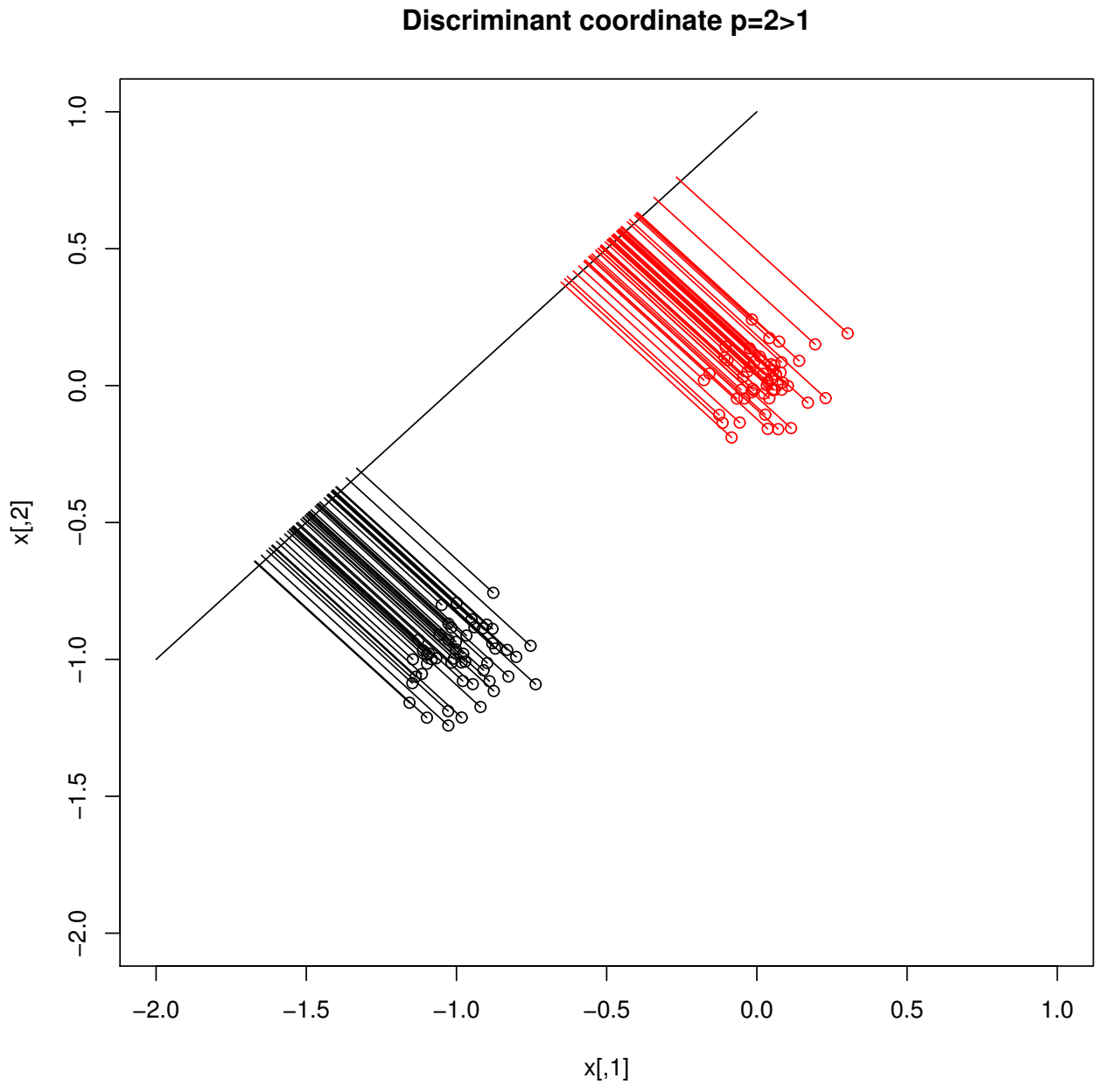
Optimises variance but doesn't take
cluster information into account.

Discriminant Coordinates (Rao (1952)):
Define projection that optimises

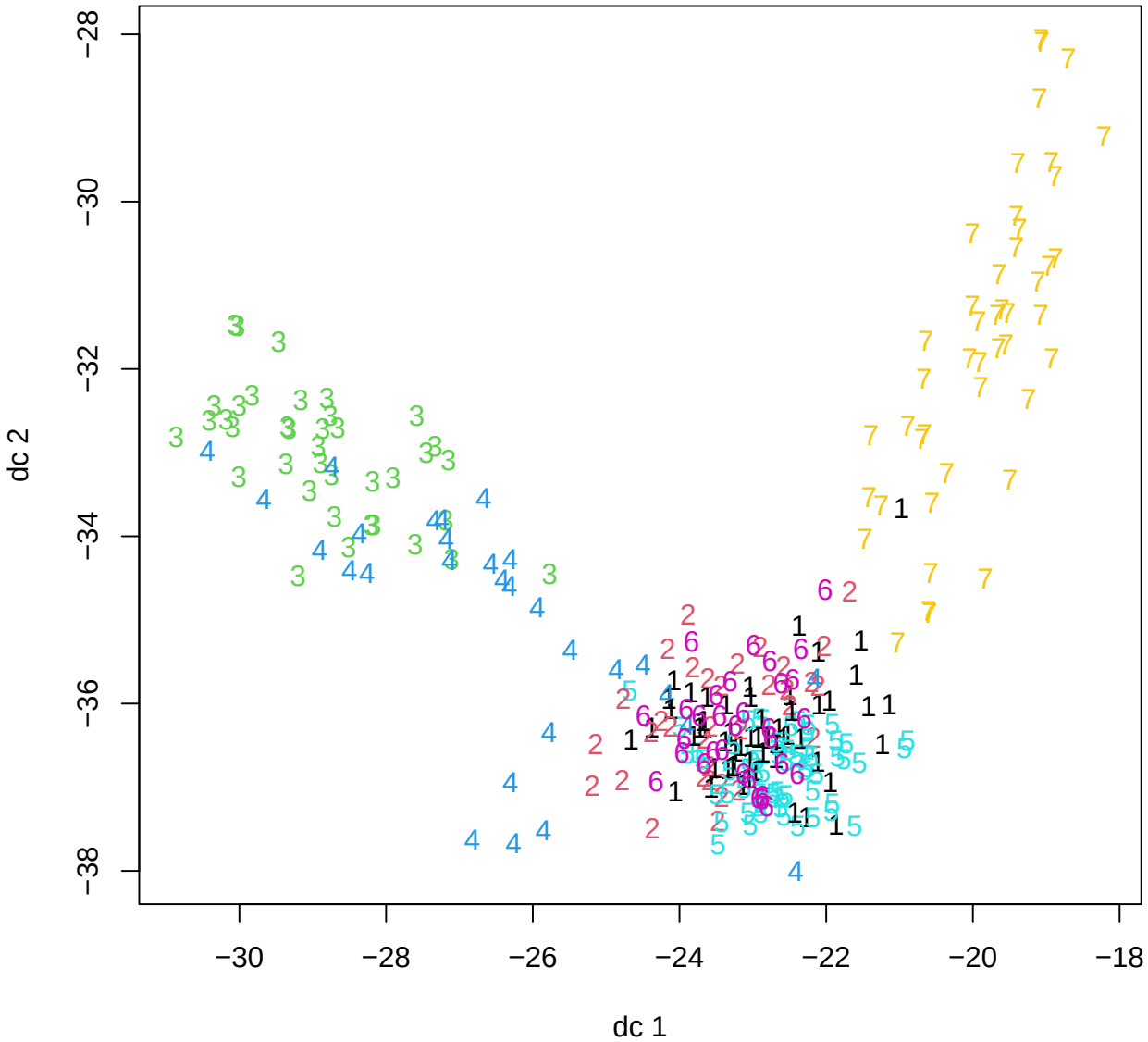
$$\frac{\textit{variation between cluster means}}{\textit{variation within clusters}}.$$

Show clusters as homogeneous and separated as possible!

Function plotcluster in fpc.


Discriminant coordinate p=2>1
x[,2]
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
x[,1]

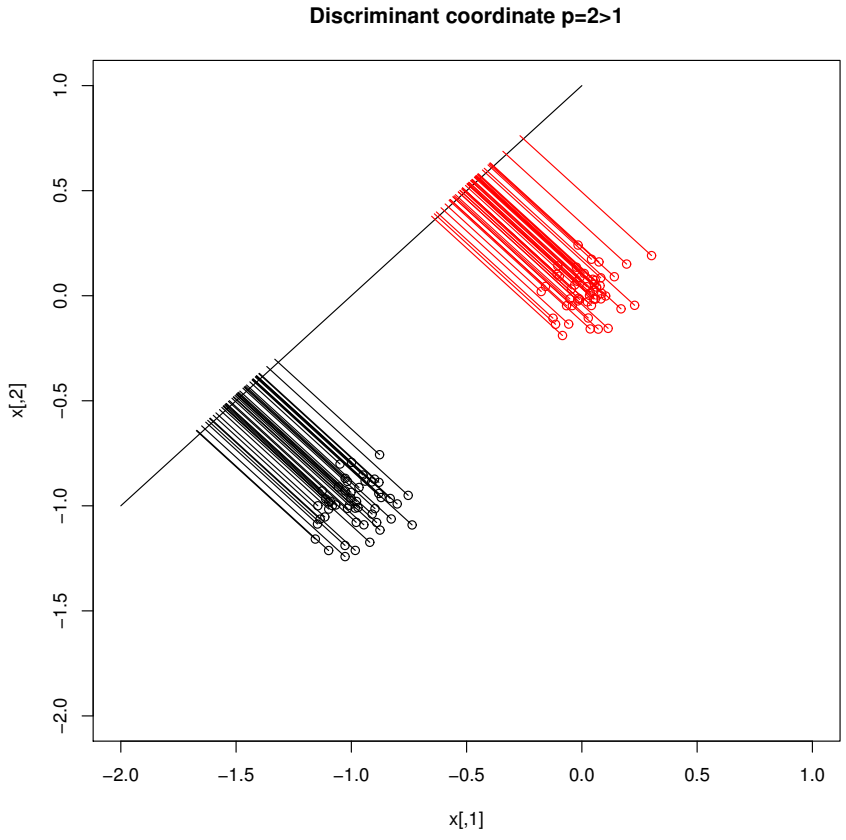
```
plotcluster(wresults2,mresults2$classification)
```

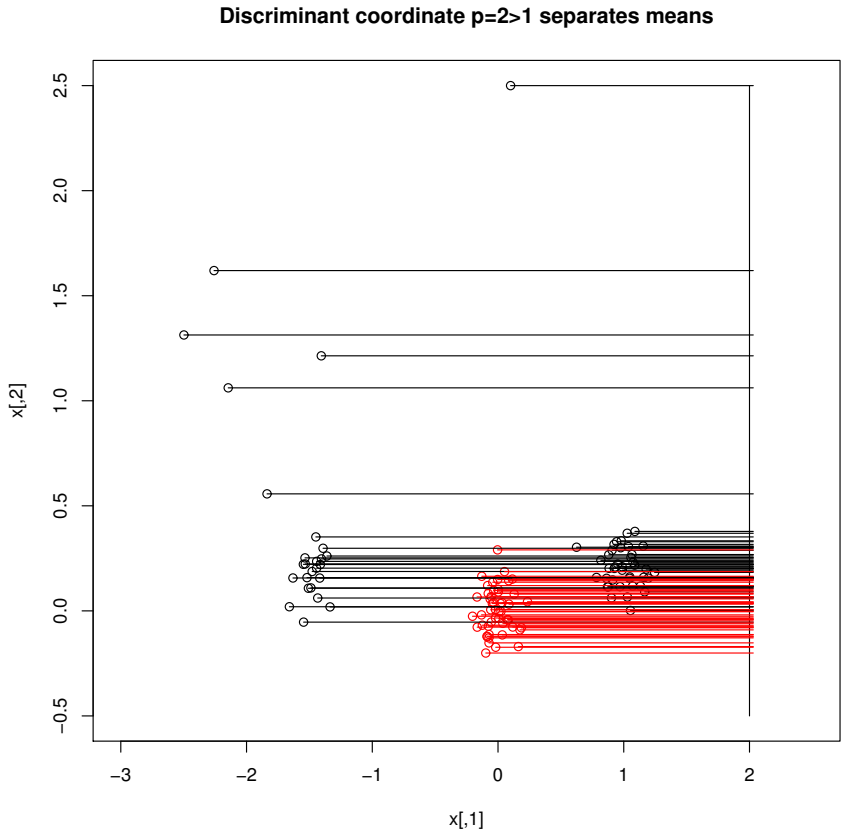


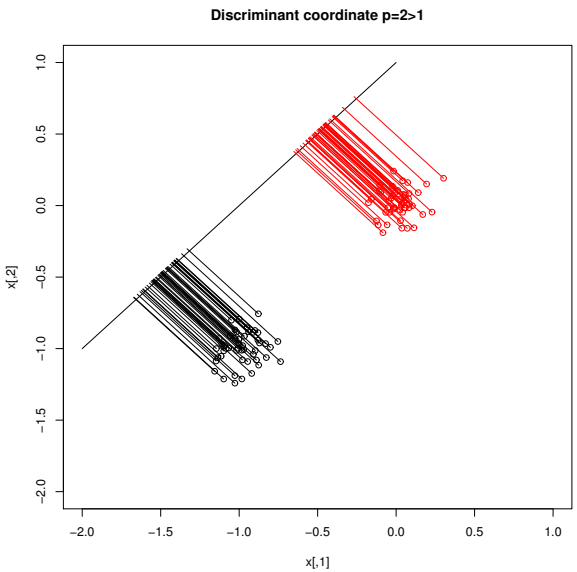
Difficulties with DC:

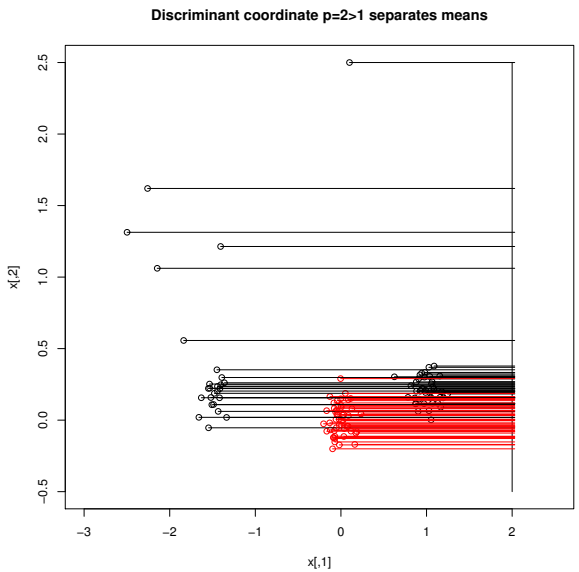
- Separation between cluster means is shown.
- All within-cluster cov-matrices equal implicitly assumed.
- More than 3 clusters: cannot see everything in 2-d.
- DCs may be dominated by outliers.

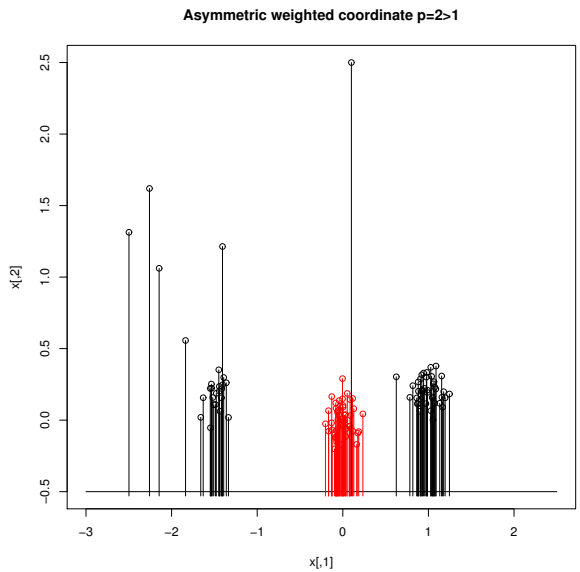
Idea: Projection method that separates single cluster from rest.


Discriminant coordinate p=2>1
x[,2]
x[,1]
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0


Discriminant coordinate p=2>1 separates means
x[,2]
x[,1]
−0.5
0.0
0.5
1.0
1.5
2.0
2.5
−3
−2
−1
0
1
2


Discriminant coordinate p=2>1
x[,2]
x[,1]
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0


Discriminant coordinate p=2>1 separates means
x[,2]
x[,1]
−3
−2
−1
0
1
2
−0.5
0.0
0.5
1.0
1.5
2.0
2.5


Asymmetric weighted coordinate p=2>1
x[,2]
x[,1]
−3
−2
−1
0
1
2
−0.5
0.0
0.5
1.0
1.5
2.0
2.5

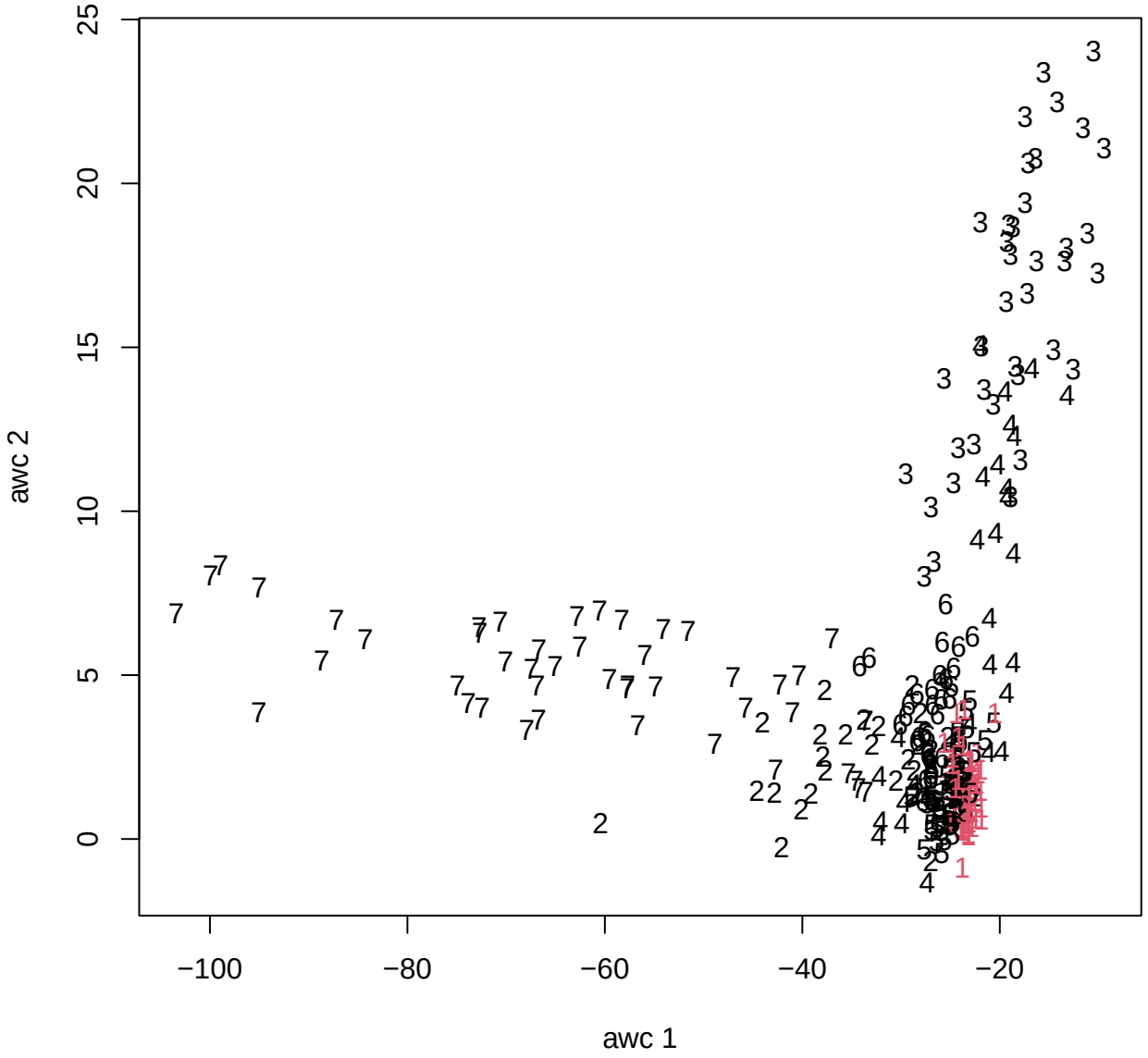
"Asymmetric weighted coordinates" (Hennig (2004)):
Projection that optimises, for "focus cluster",

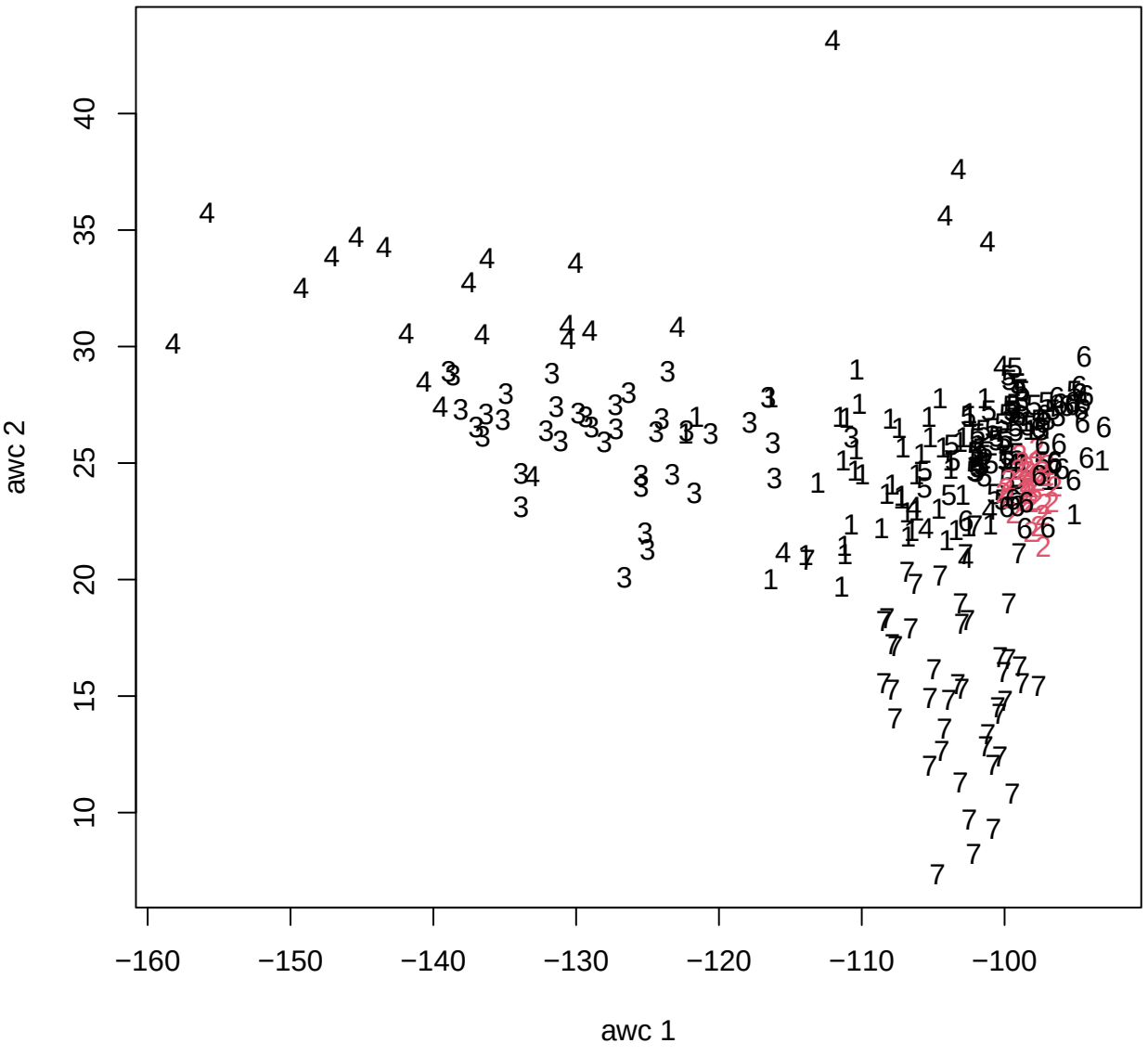
$$\frac{\textit{weighted variation between focus cluster and other clusters}}{\textit{variation within focus cluster}}.$$

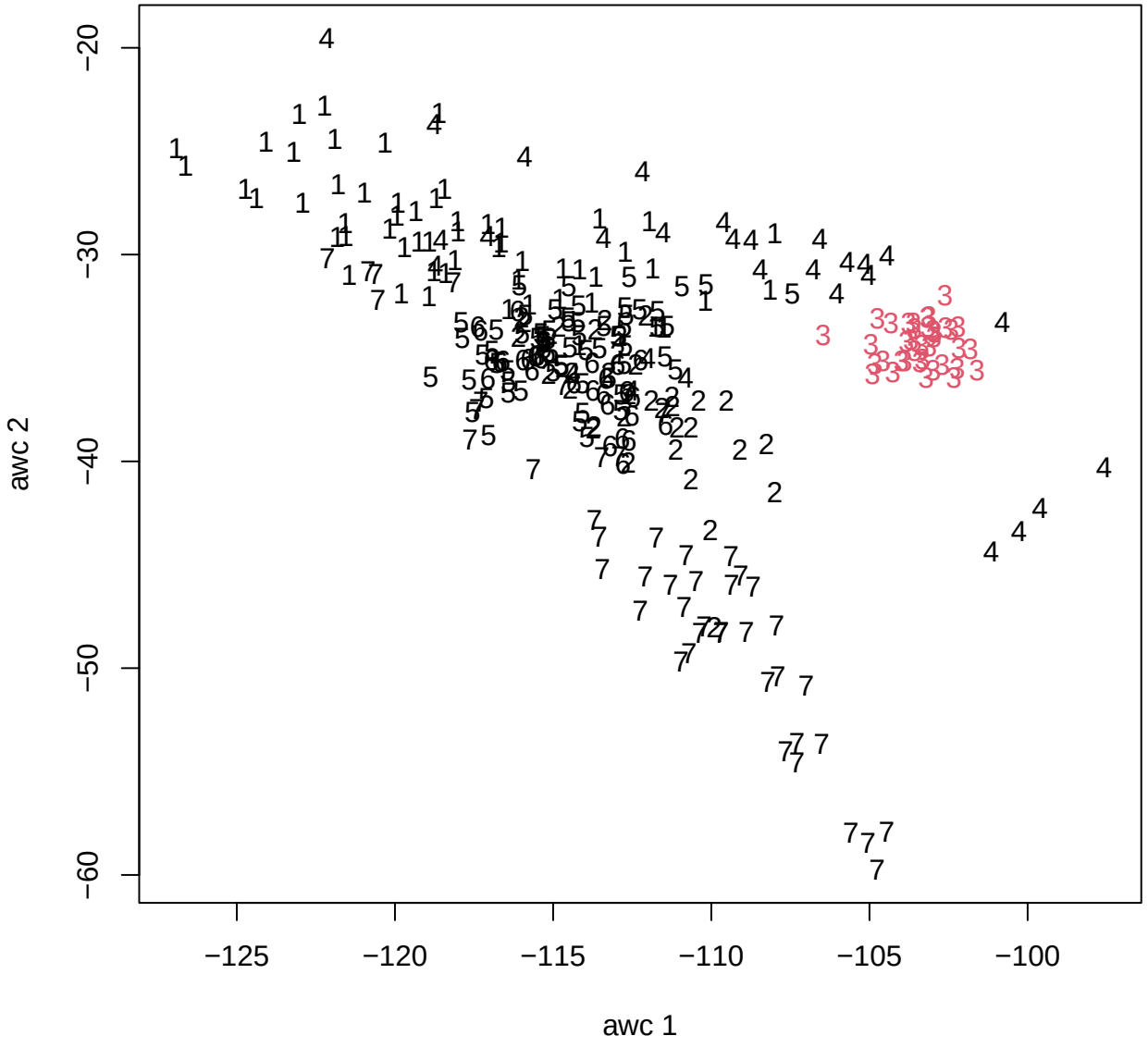
Higher weight on separation between focus cluster
and *closest points from other clusters*.

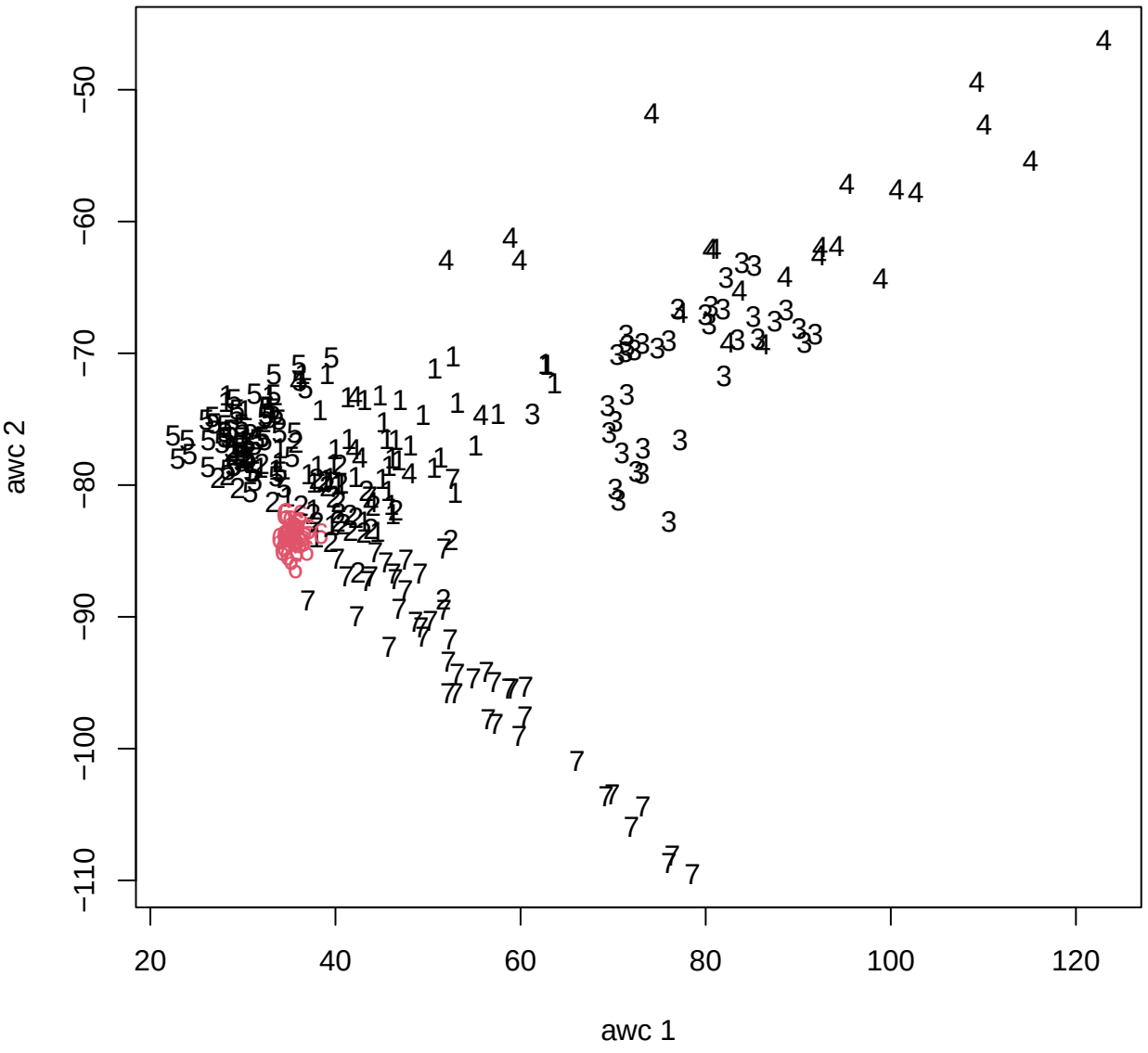
plotcluster does this when specifying `clnum`.

```
plotcluster(wresults2,mresults2$classification,clnum=1,
  col=1+(mresults2$classification==1))
plotcluster(wresults2,mresults2$classification,clnum=2,
  col=1+(mresults2$classification==2))
plotcluster(wresults2,mresults2$classification,clnum=3,
  col=1+(mresults2$classification==3))
plotcluster(wresults2,mresults2$classification,clnum=6,
  col=1+(mresults2$classification==6))
```


awc 2
awc 1
25
20
15
10
5
0
−100
−80
−60
−40
−20


awc 2
awc 1
10
15
20
25
30
35
40
−160
−150
−140
−130
−120
−110
−100


awc 2
−20
−30
−40
−50
−60
−125
−120
−115
−110
−105
−100
awc 1


awc 2
−50
−60
−70
−80
−90
−100
−110
20
40
60
80
100
120
awc 1

"Rotating" through larger than 2-dimensional space:
tourr package.

```
library(tourr)
animate(wresults2, tour_path = grand_tour(),
  display = display_xy(col=mresults2$classification,
    pch=clusym[mresults2$classification]))
```

Stability

General principle for stability assessment

- Generate several new datasets out of the original one.
- Cluster all these new datasets.
- Define statistic to formalise how similar new clusterings are to the original one.
- If they are very similar, it's stable.

Fang and Wang (2012): Quantify overall stability of clustering, estimate number of clusters by optimising stability,
R-function nselectboot (fpc).

Cluster-wise stability assessment (Hennig (2007))
Many clusterings are unstable in one way or another.
Want to know which clusters are stable
$\Rightarrow$ here *cluster-wise* methodology,
`clusterboot` in package fpc.

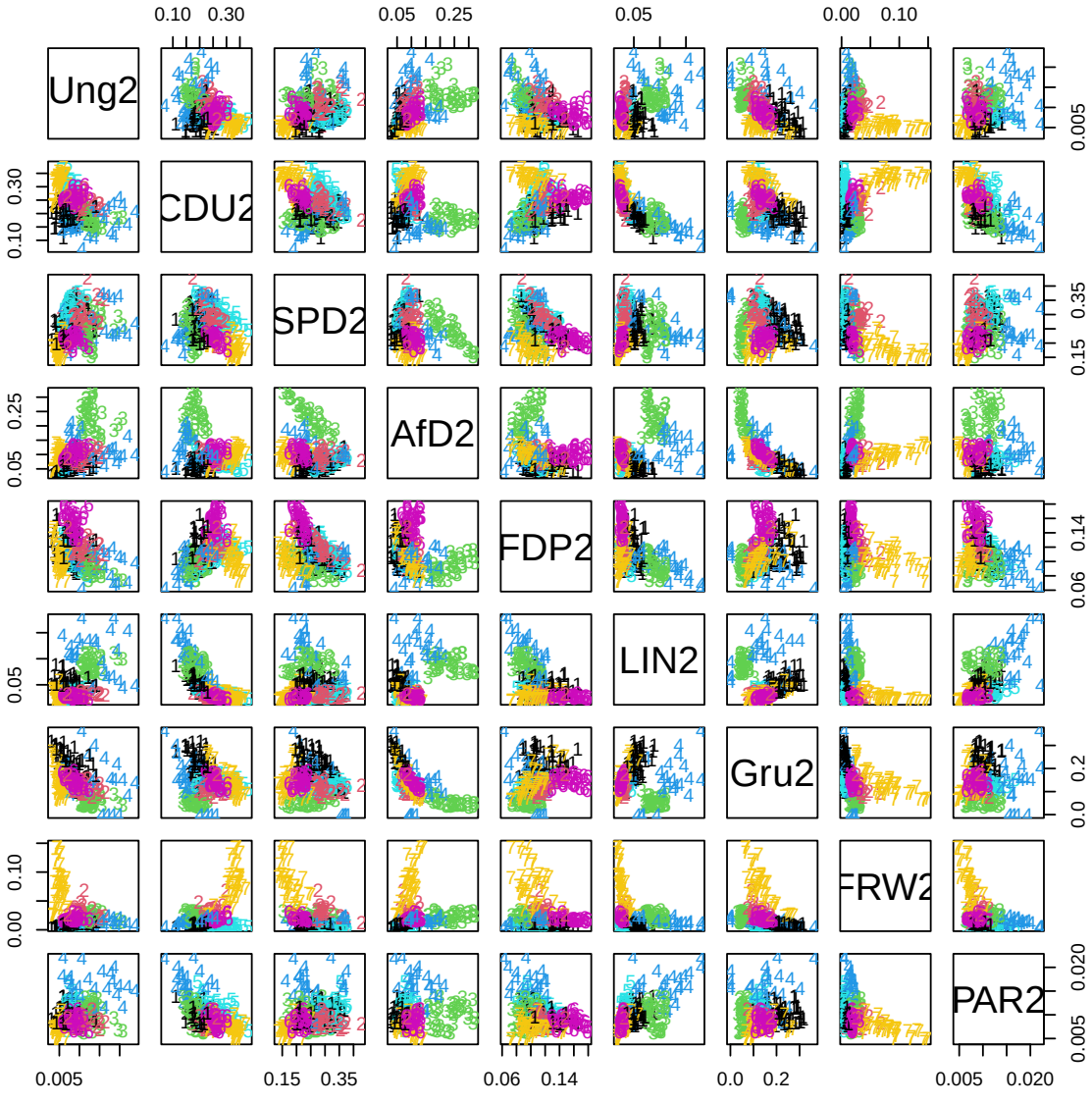
Can also assess stability including
estimating number of clusters in other ways
(as this may be source for instability).

1. Use the Jaccard coefficient

$$\gamma(C, D) = \frac{|C \cap D|}{|C \cup D|}.$$

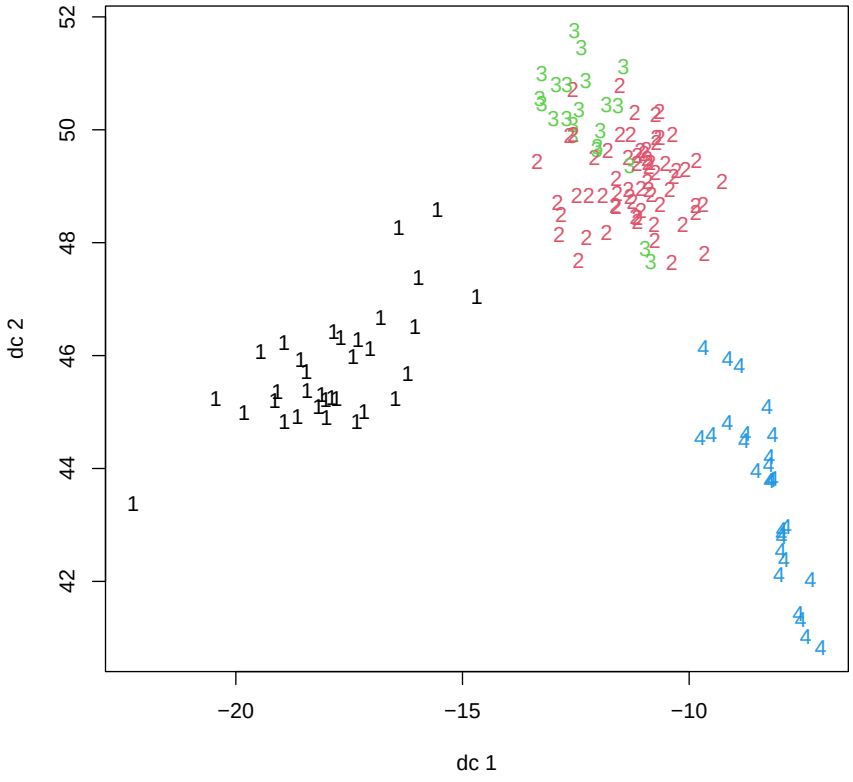
   to measure similarity between two subsets of a set.
2. Repeat B times steps 2-4:
   draw bootstrap datasets from original one,
3. apply the same clustering method to them.
4. For $C \in \mathcal{C}$ record $m_i = \max_{D \neq C} \gamma(C, D)$
5. Use $\bar{\gamma} = \frac{1}{B} \sum_{i=1}^{B} m_i$ to assess stability of C.

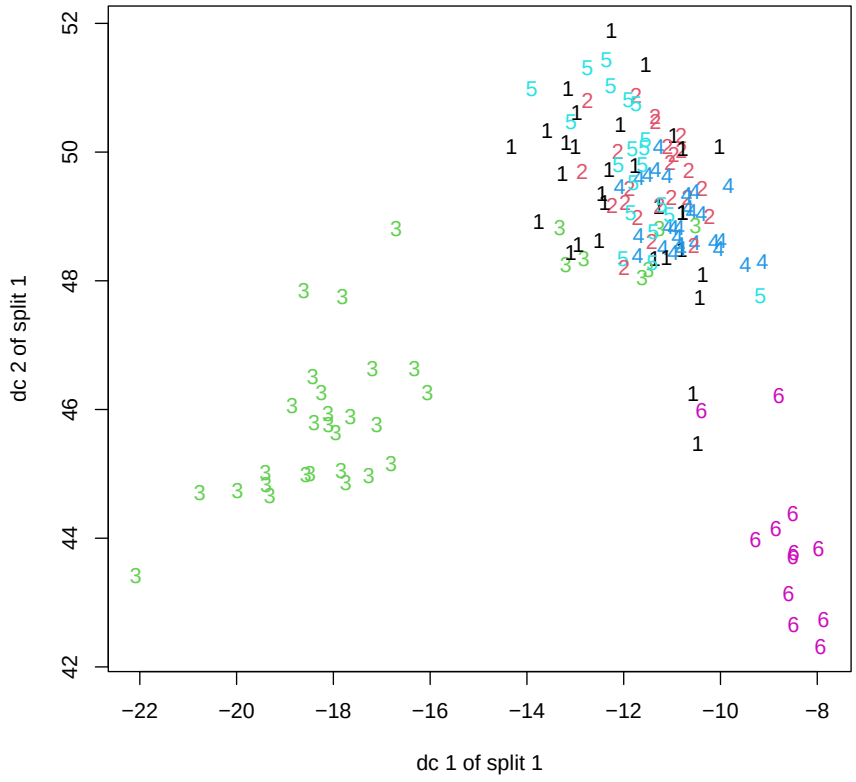
```
set.seed(667744)
cb1 <- clusterboot(wresults2,clustermethod=noisemclustCBI,nnk=0)
> cb1
* Cluster stability assessment *
Cluster method:  mclustBIC
Full clustering results are given as parameter result
of the clusterboot object, which also provides further statistics
of the resampling results.
Number of resampling runs:  100

Number of clusters found in data:  7

 Clusterwise Jaccard bootstrap (omitting multiple points) mean:
[1] 0.5195668 0.4349201 0.7497076 0.5724221 0.6190044 0.7037806 0.8748211
dissolved:
[1] 41 59  3 40 20 19  0
recovered:
[1]  3  0 45 19 12 54 99
```


Ung2
CDU2
SPD2
AfD2
FDP2
LIN2
Gru2
FRW2
PAR2
0.10 0.30
0.05 0.25
0.05
0.00 0.10
0.005
0.10 0.30
0.15 0.35
0.05 0.25
0.06 0.14
0.05
0.0 0.2
0.00 0.10
0.005 0.020
0.005
0.15 0.35
0.06 0.14
0.0 0.2
0.005 0.020

Could also split data set
and assess visual stability of clustering (Ullmann et al. (2022)):

```
set.seed(77665544)
splitd <- sample(299,150)
wrsplit1 <- wresults2[splitd,]
wrsplit2 <- wresults2[-splitd,]
wrm1 <- Mclust(wrsplit1)
wrm2 <- Mclust(wrsplit2)

dcws1 <- discrcoord(wrsplit1,wrm1$classification)
plot(dcws1$proj,col=wrm1$classification,pch=clusym[wrm1$classification],
  xlab="dc 1",ylab="dc 2")
plot(as.matrix(wrsplit2) %*% dcws1$units,col=wrm2$classification,
  pch=clusym[wrm2$classification],xlab="dc 1 of split 1",
  ylab="dc 2 of split 1")
```


dc 2
dc 1
52
50
48
46
44
42
−20
−15
−10


dc 2 of split 1
dc 1 of split 1
52
50
48
46
44
42
−22
−20
−18
−16
−14
−12
−10
−8

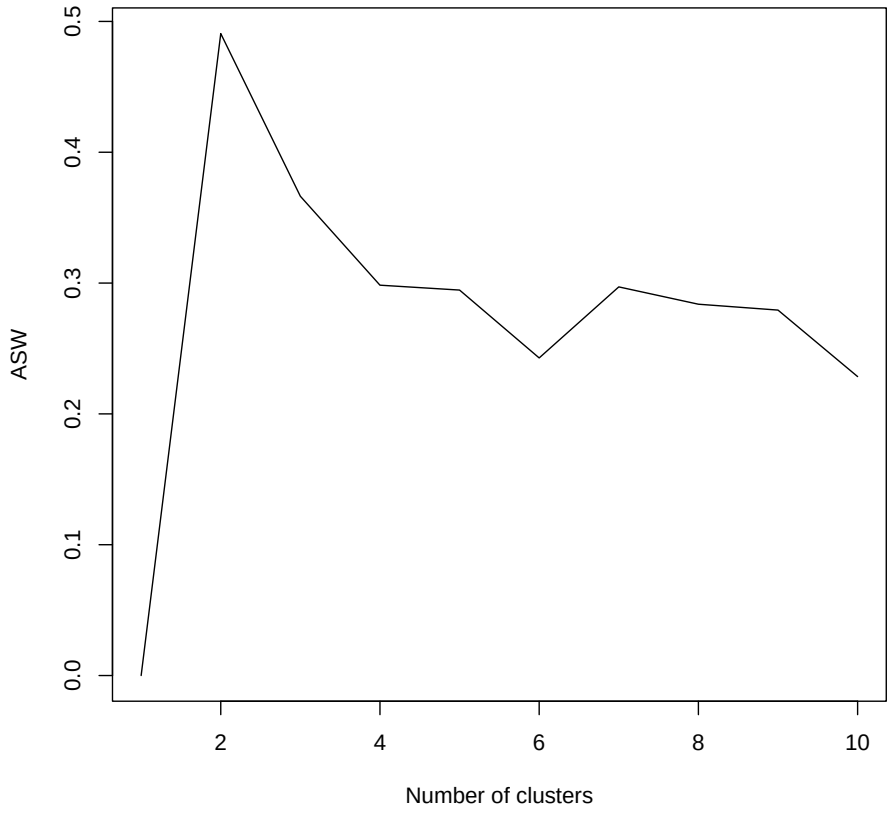
Overall, clusters 3 (strong AfD constituencies in the east), cluster 7 (Bavaria, with strong Freie Waehler) stable and clear, cluster 6 (strong FDP in Baden-Wuerttemberg) less separated.

The others may well be partitioned in different ways.

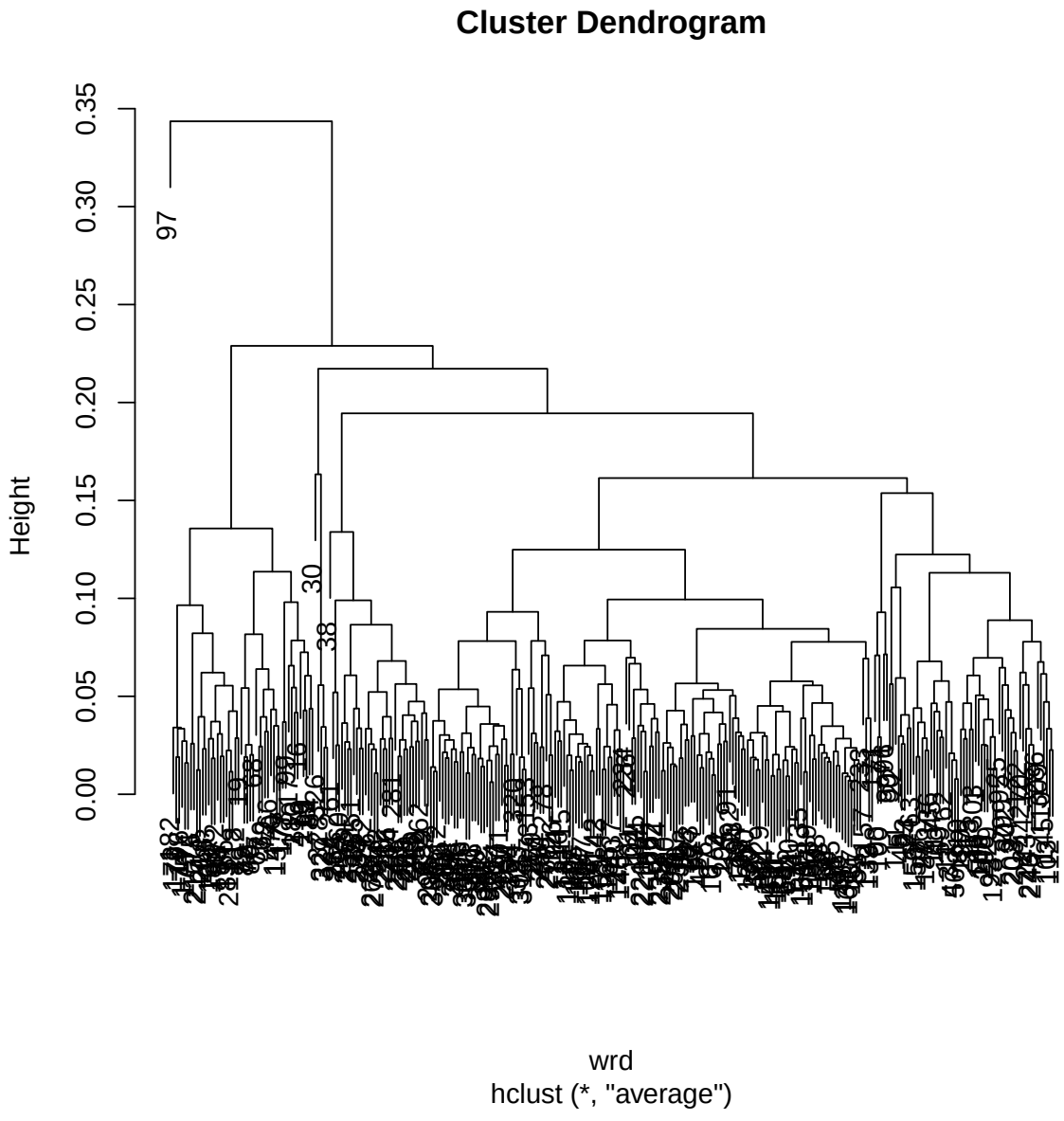
3.2 Average Linkage

Use Euclidean distance on *unscaled* data,
meaning that weaker parties have weaker influence on result.

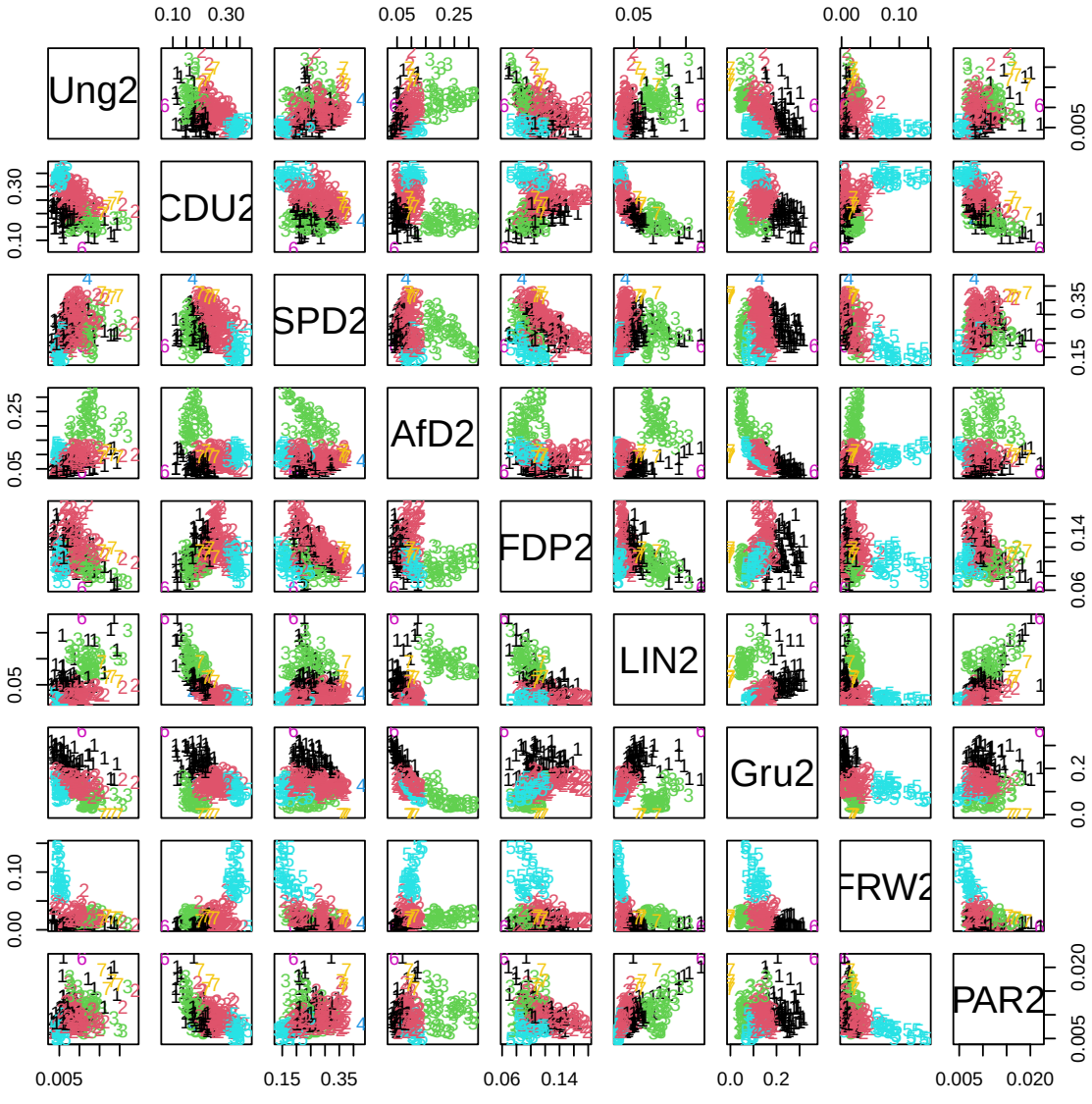
```
wrd <- dist(wresults2)
alresults <- hclust(wrd,method="average")

# Number of clusters,  ASW
maxnc <- 10
alasw <- rep(0,10)
kpartition <- list()
for (k in 2:maxnc){
  kpartition[[k]] <- cutree(alresults, k = k)
  alasw[k] <- summary(silhouette(kpartition[[k]],wrd))$avg.width
}
plot(1:10,alasw,type="l",xlab="Number of clusters",ylab="ASW") # 2 over 7

plot(alresults)
```

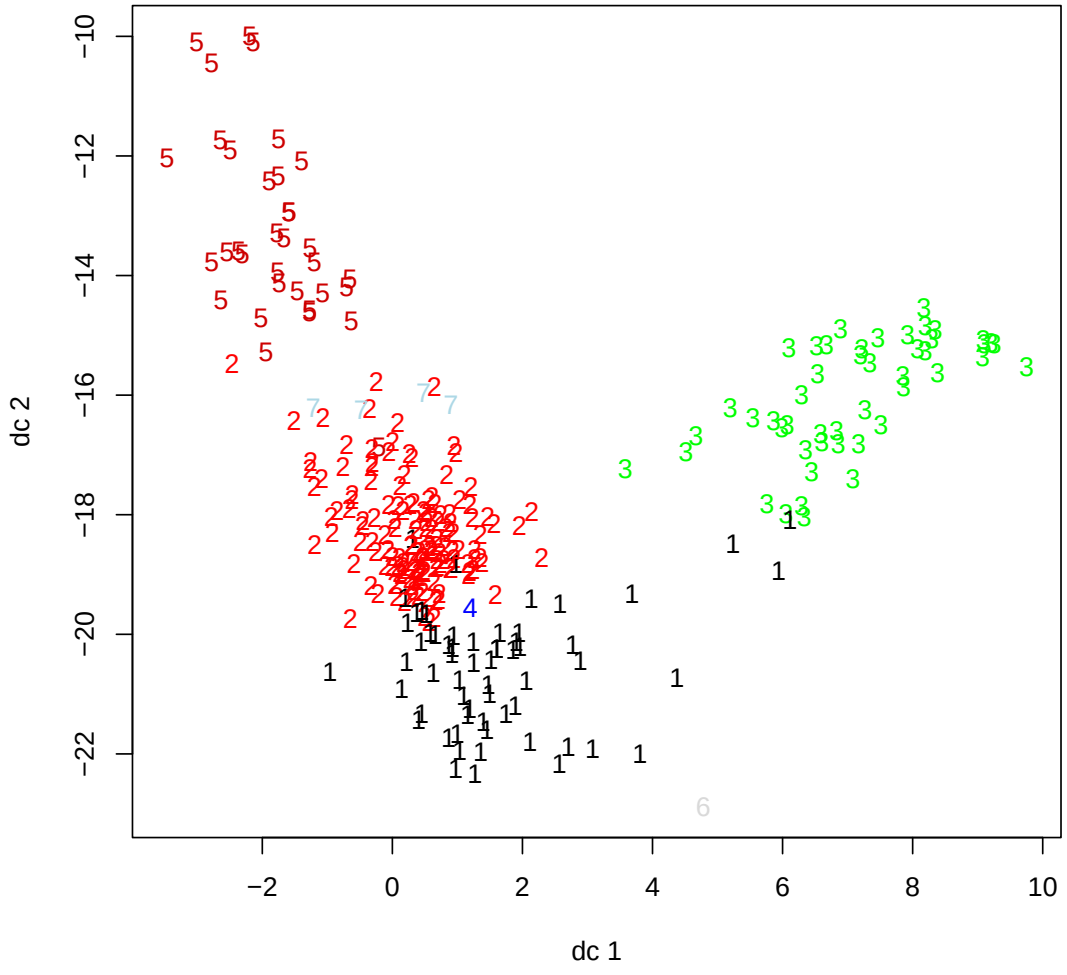


Problem with ASW: Often maximised at $k = 2$ or very low, ignoring finer structure. Use local optimum at 7!


Cluster Dendrogram
Height
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
97
30
38
wrd
hclust (*, "average")

```
al7result <- cutree(alresults,7)
pairs(wresults2,pch=clusym[al7result],col=al7result)
plotcluster(wresults2,al7result)
table(al7result) # still fairly imbalanced
# al7result
#   1   2   3   4   5   6   7
#  61 150  48   1  34   1   4

table(al7result,land)
# cluster 1 Hamburg, NRW and others
# cluster 2 very mixed
# cluster 3 East
# cluster 5 most of Bayern
# cluster 7 Saarland
```


Ung2
CDU2
SPD2
AfD2
FDP2
LIN2
Gru2
FRW2
PAR2
0.10 0.30
0.05 0.25
0.05
0.00 0.10
0.005
0.10 0.30
0.15 0.35
0.05 0.25
0.06 0.14
0.05
0.0 0.2
0.00 0.10
0.005 0.020
0.005
0.15 0.35
0.06 0.14
0.0 0.2
0.005 0.020



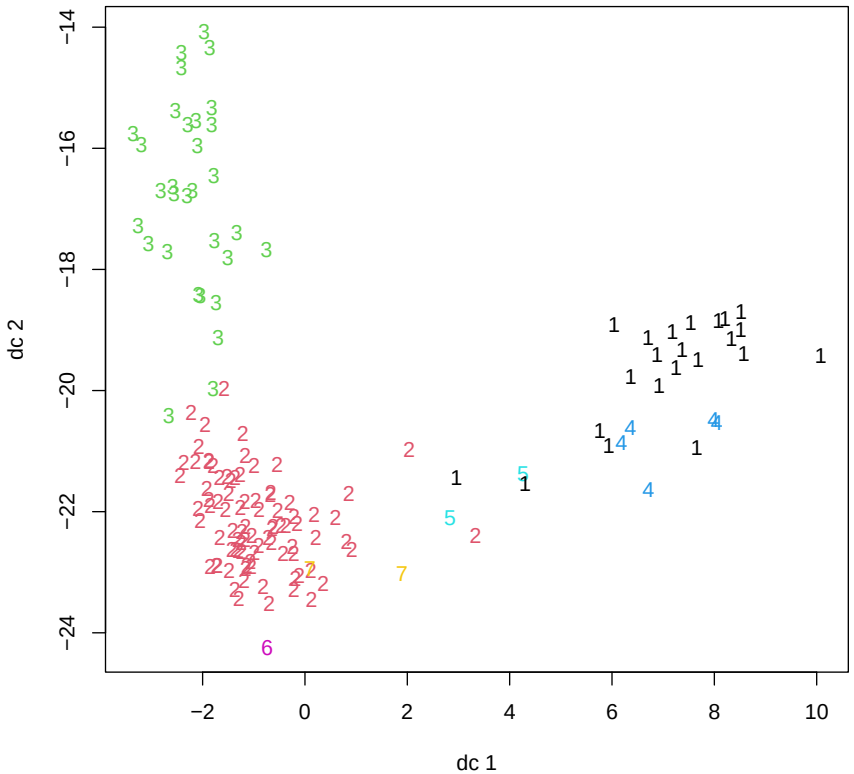
Looks good enough for all clusters.

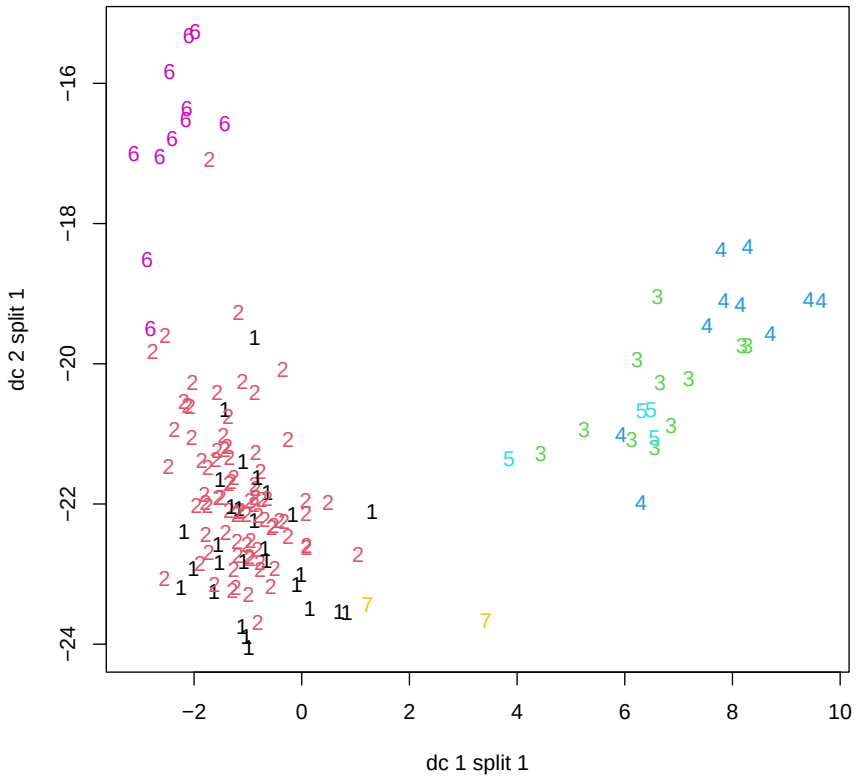
For stability fix $k = 7$ because ASW-estimation not reliable.

```
cbal <- clusterboot(wresults2,clustermethod=hclustCBI,k=7,
  scaling=FALSE,method="average")

# > cbal
# * Cluster stability assessment *
# Cluster method:  hclust/cutree
# Full clustering results are given as parameter result
# of the clusterboot object, which also provides further statistics
# of the resampling results.
#  Number of resampling runs:  100

# Number of clusters found in data:  7

# Clusterwise Jaccard bootstrap (omitting multiple points) mean:
# [1] 0.7179639 0.8703161 0.8226565 0.3275462 0.8547446 0.5125000 0.9493333
# dissolved:
# [1] 20  1  5 70  4 60  6
# recovered:
# [1] 56 83 72 30 85 40 93
```

1 not so stable, neither of course one-point clusters.

```
wral1 <- hclust(dist(wrsplit1),method="average")
wral2 <- hclust(dist(wrsplit2),method="average")
wral17 <- cutree(wral1,7)
wral27 <- cutree(wral2,7)

dcwral1 <- discrcoord(wrsplit1,wral17)
plot(dcwral1$proj,col=wral17,pch=clusym[wral17],xlab="dc 1",ylab="dc 2")
plot(as.matrix(wrsplit2) %*% dcwral1$units,col=wral27,pch=clusym[wral27],
  xlab="dc 1 split 1",ylab="dc 2 split 1")
```





Again points to three rough groups that are stable,
but not what happens within them.

4 Conclusion

- Very distinctive groups are in Bavaria,
  and in the East where the AfD is strong.
- Otherwise various groupings are possible,
  can be interpreted with care.
- No scaling of variables: More influence to
  larger numbers, stronger parties (may be appropriate).
- Focus here on visual validation and stability,
  also use external information.
- Number of clusters estimation is hard,
  and will always cause instability.

References I

Bouveyron, C., G. Celeux, T. B. Murphy, and A. E. Raftery (2019). *Model-based Clustering and Classification for Data Science*. Cambridge University Press.

Fang, Y. and J. Wang (2012, March). Selection of the number of clusters via the bootstrap method. *Computational Statistics and Data Analysis 56*(3), 468–477.

Hennig, C. (2004). Asymmetric linear dimension reduction for classification. *Journal of Computational and Graphical Statistics 13*(4), 930–945.

Hennig, C. (2007). Cluster-wise assessment of cluster stability. *Computational Statistics and Data Analysis 52*(1), 258–271.

Kaufman, L. and P. Rousseeuw (1990). *Finding Groups in Data*. Wiley, New York.

Rao, C. R. (1952). *Advanced Statistical Methods in Biometric Research*. Wiley, New York.

Ullmann, T., C. Hennig, and A.-L. Boulesteix (2022). Validation of cluster analysis results on validation data: A systematic framework. *WIREs Data Mining and Knowledge Discovery 12*(3), e1444.