


Frühjahrstagung des VDSt
in Saarbrücken

Amt für Statistik Berlin-Brandenburg



Clusteranalyse mit der Open Source-Software R

Dienstag, 1. April 2008



Inhalt

- Statistiksistem R
- Clusteranalyse
- Kartierung der Ergebnisse
- Fazit



Statistiksystem R

R ist

- eine Umgebung für die statistische Analyse, graphische Exploration und Darstellung;
- ein Dialekt der Sprache S (AT&T Bell) und
- eine freie Implementierung dieser Sprache (kommerziell S-Plus).

- GNU General Public License
- Internetsoftware: Entwickler- und Nutzergemeinschaft, Internetbibliothek (anders als SAS, SPSS, Zwischenstellung Stata)
- Plattform: Windows, Linux und Mac OS X

- Anwendung: Ökonomie, Biologie, Geologie/Geodaten, Lehre („Statistiklabor“)



Statistiksystem R

Anlaufstelle ist der CRAN-Server:

- R-Programm, Erweiterungsmodule, Community

The R Project for Statistical Computing

Navigation Links:

- About R
 - [What is R?](#)
 - [Contributors](#)
 - [Screenshots](#)
 - [What's new?](#)
- Download
 - [CRAN](#)
- R Project Foundation
 - [Members & Donors](#)
 - [Mailing Lists](#)
 - [Bug Tracking](#)
 - [Developer Page](#)
 - [Conferences](#)
 - [Search](#)
- Documentation
 - [Manuals](#)
 - [FAQs](#)
 - [Newsletter](#)
 - [Wiki](#)
 - [Books](#)
 - [Certification](#)
 - [Other](#)
- Misc
 - [Bioconductor](#)
 - [Related Projects](#)
 - [Links](#)

Statistical Plots:

- PCA 5 vars:** `princomp(x = data, cor = cor)`. Includes a biplot with variables: Fertility, Examination, Education, Catholic, Agriculture. A bar chart shows the first three principal components account for 60% of the variance.
- Clustering 4 groups:** A dendrogram showing hierarchical clustering of data points into four groups.
- Factor 1 [41%]:** A normal distribution plot for the first principal component.
- Factor 3 [19%]:** A normal distribution plot for the third principal component.

Getting started:

- R is a free software environment for statistical computing and graphics. It compiles and runs on a wide variety of UNIX platforms, Windows and MacOS. To download R, please choose your preferred [CRAN mirror](#).
- If you have questions about R like how to download and install the software, or what the license terms are, please read our [answers to frequently asked questions](#) before you send an email.

News:

- [R project ideas](#) for the Google [Summer of Code 2008](#).
- [R version 2.6.2](#) has been released on 2008-02-08.
- [R News 7/3](#) has been published on 2007-12-18.
- [useR! 2008](#), the R user conference, will be held at Dortmund University, Germany, August 12-14, 2008.

This server is hosted by the [Department of Statistics and Mathematics](#) of the [WU Wien](#).

<http://www.r-project.org/>



Clusteranalyse: Ablauf

Daten einlesen:

```
Clusteranalyse_Fruehjahrstagung_2008-04_a.R  
# Lese Excel-Tabelle ein -----  
options(digits=3)  
setwd("D:/Buero/Statistik/AfS/Kommunalstat/Workplace/Cluster-AG/Daten/")  
  
library(RODBC)  
channel<-odbcConnectExcel("EWR200706E_Matrix.xls")  
sqlTables(channel)  
sqlQuery(channel, "select * from \"Tabelle1$\"")  
EWR07 <- sqlQuery(channel, "select * from \"Tabelle1$\"")  
odbcCloseAll()
```

Darstellung im Editor Tinn-R



Clusteranalyse: Ablauf

Variablen berechnen und Datei speichern:

```
Clusteranalyse_Fruehjahrstagung_2008-04_a.R |
ed<-edit(EWRO7)      # Daten im Editor anzeigen

# Anteilsvariablen berechnen -----

EWRO7$pE00U01 <- with(EWRO7, 100 * E_E00_01/E_E)
EWRO7$pE01U02 <- with(EWRO7, 100 * E_E01_02/E_E)
EWRO7$pE02U03 <- with(EWRO7, 100 * E_E02_03/E_E)

EWRO7$pE85U90 <- with(EWRO7, 100 * E_E85_90/E_E)
EWRO7$pE90U95 <- with(EWRO7, 100 * E_E90_95/E_E)
EWRO7$pE95U99 <- with(EWRO7, 100 * E_E95_110/E_E)

save(EWRO7, file="EWRO7.Rdata")

setwd("D:/Buero/Statistik/AfS/Kommunalstat/Workplace/Cluster-AG/Daten/")
load("EWRO7.Rdata")
```



Clusteranalyse: Ablauf

Deskriptive Statistiken berechnen:

```
print(weighted.mean(pE00U01,E_E) # gewichteter Mittelwert  
print(c(SD=sd(pE00U01), IQR=IQR(pE00U01), Range=diff(range(pE00U01))))
```

Ergebnis:

```
[1] 1.66  
      SD   IQR Range  
0.556 0.595 4.688
```



Clusteranalyse: Ablauf

Histogramme:

```
# Histogramm 1 Sturges -----  
  
layout(matrix(1:30,6,5), widths=c(1,1,1,1,1), heights=c(3,3,3,3,3,3))  
par(col.axis="blue", mar=c(4,4,2,1), oma=c(1,1,1,1), cex=0.6, las=1)  
  
i <- 0  
repeat{  
  i <- i + 1  
  hist(data.matrix(EWRO7[i+40]), breaks = "Sturges", probability=FALSE, main="", xlab=names(EWRO7[i+40]))  
  if(i==30) break  
}
```




Clusteranalyse: Ablauf

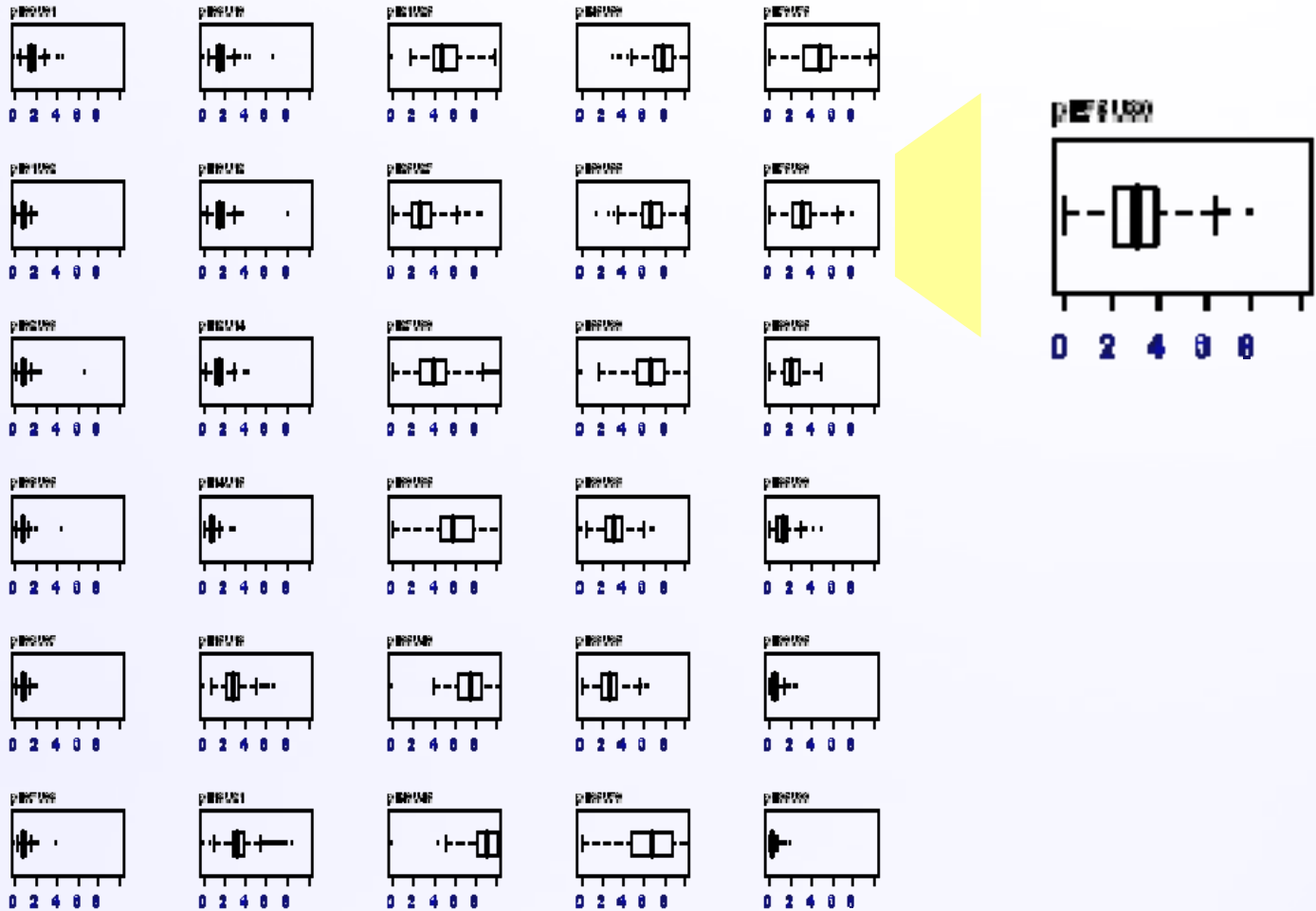
Boxplot:

```
# Boxplot -----  
  
layout(matrix(1:30,6,5), widths=c(1,1,1,1,1), heights=c(3,3,3,3,3))  
par(col.axis="blue", mar=c(4,4,2,1), oma=c(1,1,1,1), cex=0.6, las=1)  
  
i <- 0  
repeat(  
  i <- i + 1  
  {boxplot(EWRO7[i+40], pars = list(boxwex = 0.8, staplewex = 0.6, outwex = 0.8),  
horizontal=TRUE, xlab="", ylim = c(0, 10), notch = FALSE, border = par("fg"))  
  title(names(EWRO7[i+40]), cex.main=0.8, adj=0)  
  }  
  if(i==30) break  
}
```



Clusteranalyse: Ablauf

Boxplot:





Clusteranalyse: Ablauf

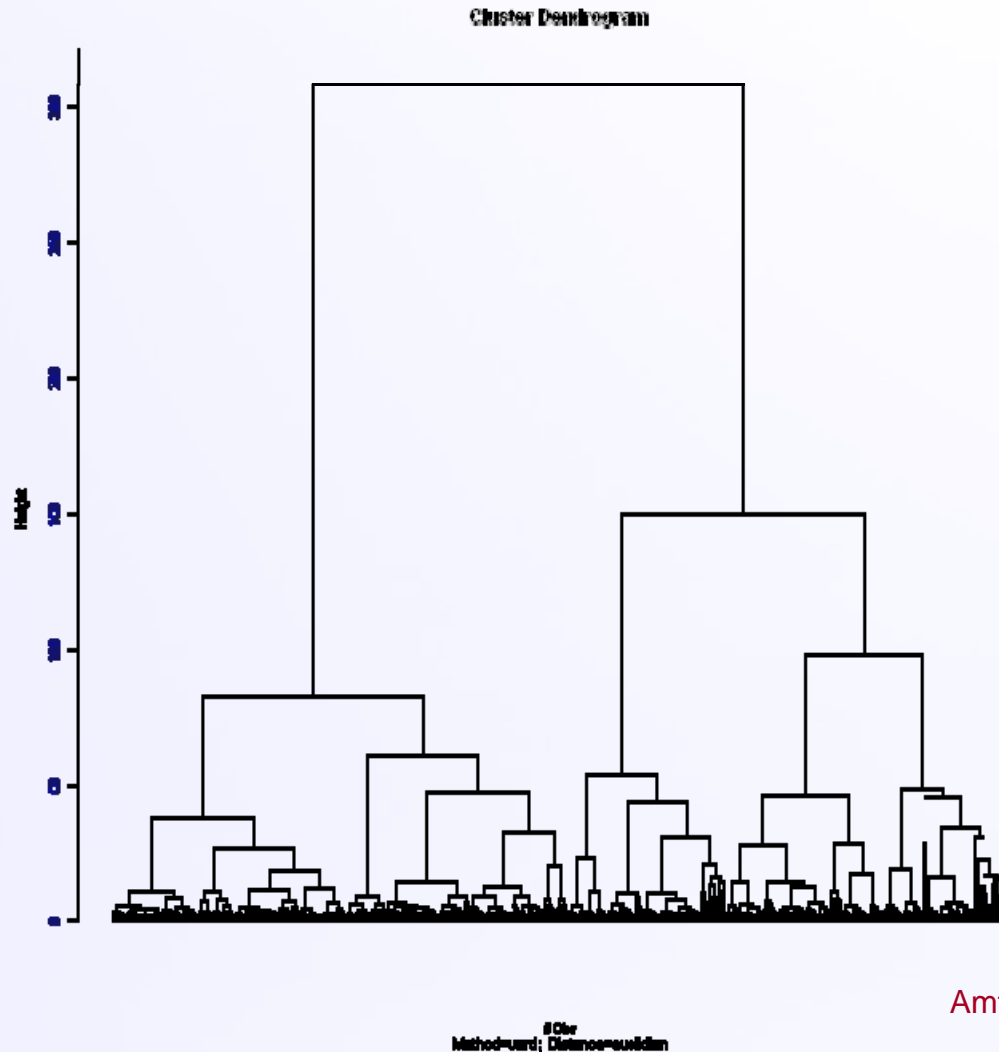
Cluster berechnen, Dendrogramm zeichnen:

```
# ----- HClust -----  
require(graphics)  
  
HClust.1 <- hclust(dist(scale(model.matrix(~-1 + as.matrix(EWRO7[41:70])^2, EWRO7))) , method= "ward")  
  
# ----- Dendrogramm 1  
par(col.axis="blue", cex=0.6)  
plot(HClust.1, main= "Cluster Dendrogram", xlab= "#Obs", sub="Method=ward; Distance=euclidian", hang=-1, labels=F)
```



Clusteranalyse: Ablauf

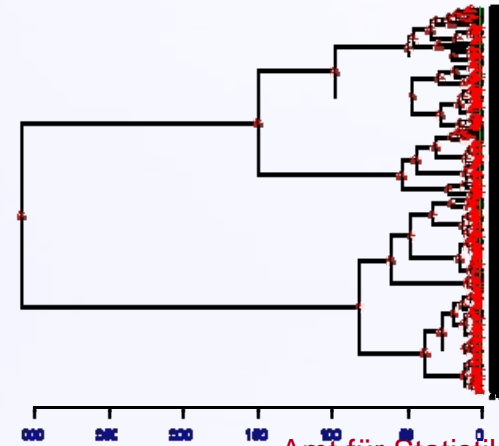
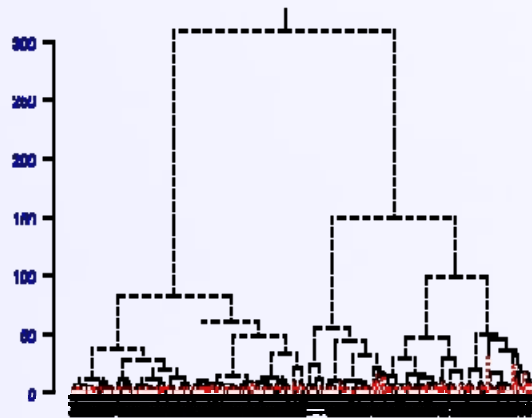
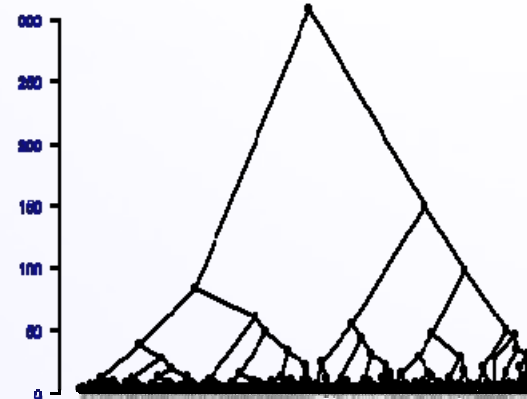
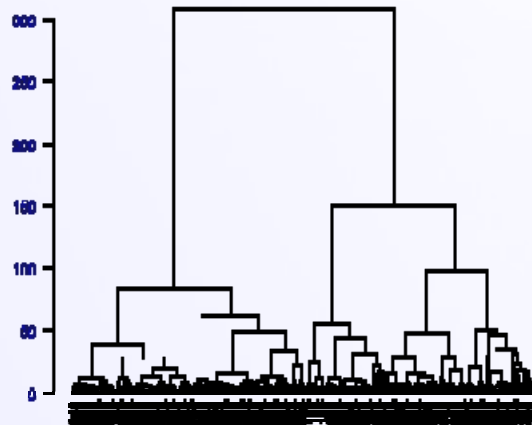
Dendrogramm:





Clusteranalyse: Ablauf

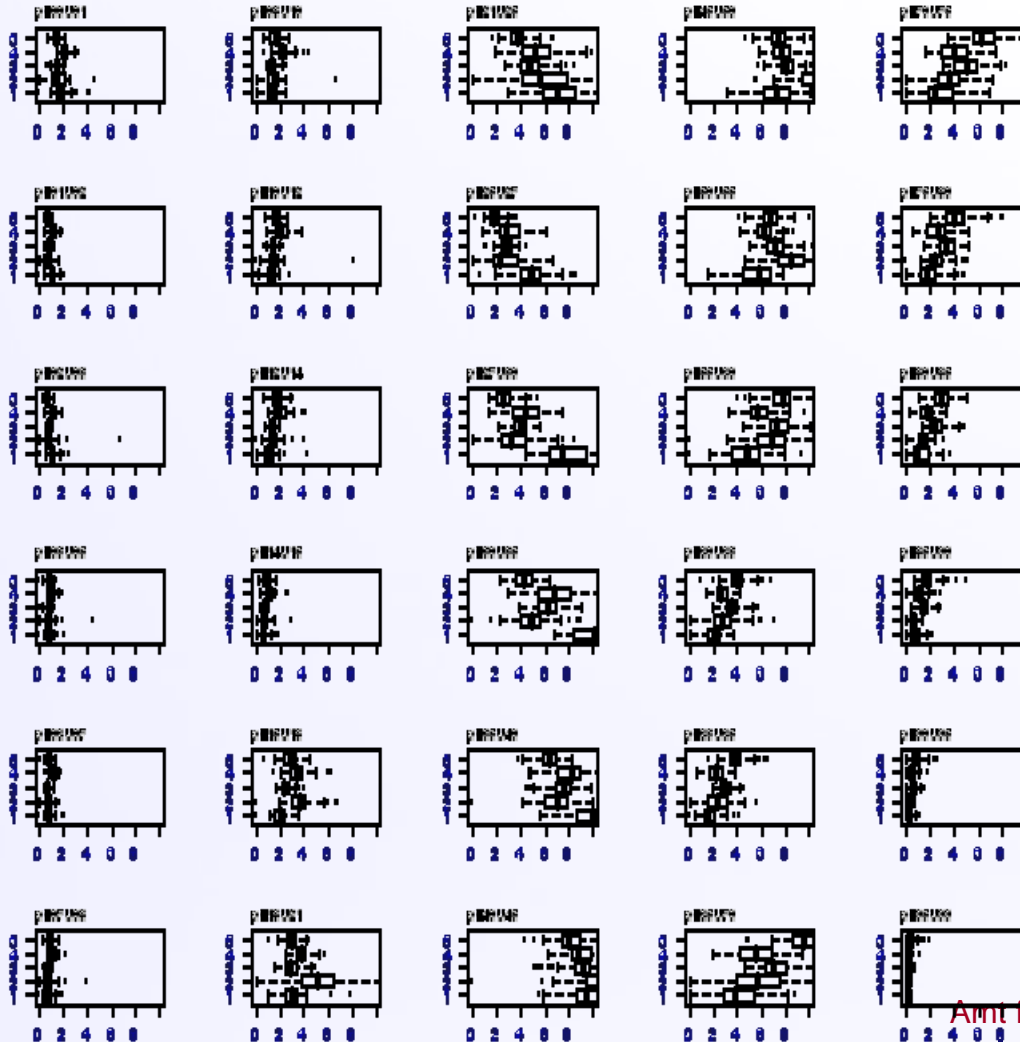
Dendrogramme:





Clusteranalyse: Ablauf

Clusteranalyse:



Clusteranalyse: Ablauf

Kartieren:

```
# ----- Mapping Faelle der nclust-Loesung -----  
library(rgdal)  
library(RColorBrewer)  
library(classInt)  
  
setwd("D:/Buero/Statistik/AfS/Kommunalstat/Workplace/Cluster-AG/MapInfo/")  
lor <- readOGR("LOR_PLR.TAB", "LOR_PLR")  
  
# ----- MAP  
plotvar <- clusternum # Vektor erzeugt mit cutree  
plotclr <- brewer.pal(nclust, "YlOrRd") # Blues, BuPu, PuOr  
# BuGn GnBu Greens Greys Oranges OrRd PuBu  
# PuBuGn PuRd Purples RdPu Reds YlGn YlGnBu Yl  
  
colnum <- classIntervals(plotvar, nclust, style="fixed", fixedBreaks = seq(1, nclust, by=1), all.inside=T)  
colcode <- findColours(colnum, plotclr)  
  
plot(lor, xlim=c(5000,48000), ylim=c(4000,35000))  
plot(lor, col=colcode, add=T)  
  
# points(lor$X, lor$Y, pch=16, col="red", cex=0.5)  
  
title("Clusterlösung", sub="LOR=447, Cluster=")  
legend(43000,35000, legend=names(attr(colcode, "palette")), fill=attr(colcode, "palette"), cex=0.6, bty="n")
```

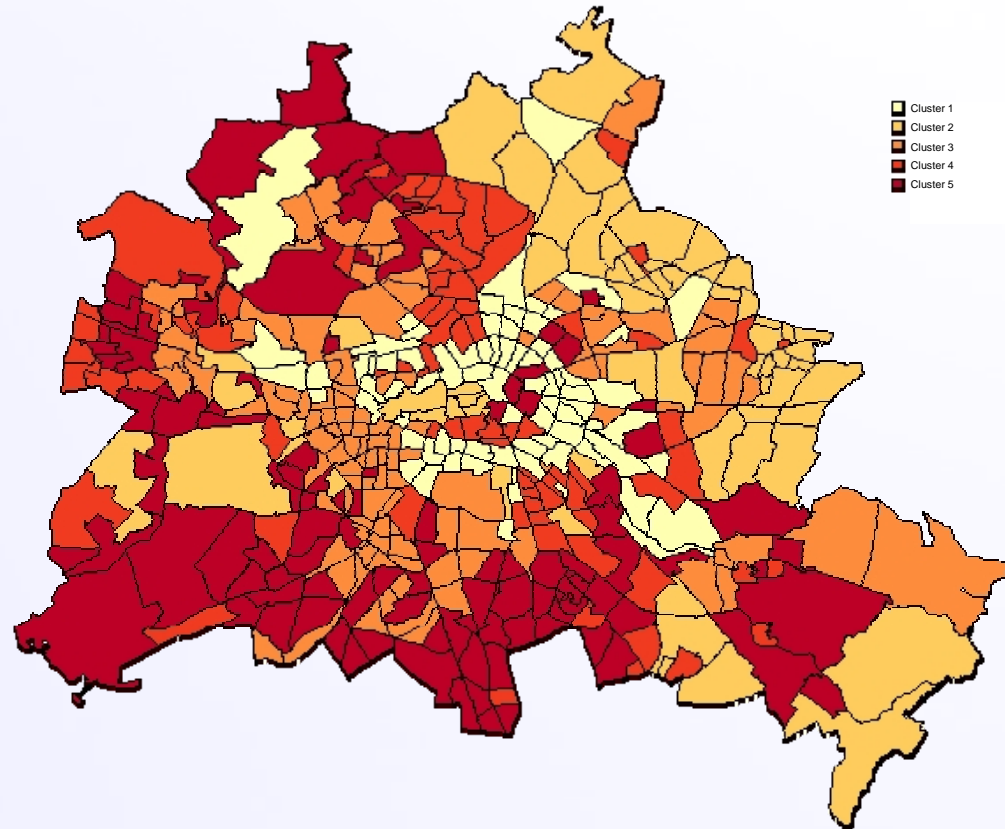
MapInfo-
Geometrie
einlesen



Clusteranalyse: Ablauf

Kartieren:

Clusterlösung





Fazit

- + Programm ist frei verfügbar
- + breites statistisches Instrumentarium
- + ständige Weiterentwicklung
- + vielfältige Grafikmöglichkeiten
- + MapInfo-, ESRI-Geometrien einlesbar
- + Steuerung über Syntax (leichte Wiederholbarkeit von Läufen)
- + grosse Nutzer- und Entwicklergruppe
- + vielfältige Hilfestellungen

- - nicht intuitiv
- - nur rudimentäre graphische Editoren (ebenfalls GNU)
- - Einarbeitungsaufwand
- - geringere Effizienz

- AfS: ergänzend zu anderen Statistikprogrammen