Data Visualisation with R

Caroline Sporleder & Ines Rehbein

WS 09/10

Data Visualisation with R

- What is R?
  - Free software environment for statistical computing and graphics
  - Runs on UNIX/Linux, Windows and MacOS
  - `http://www.r-project.org`
- Tutorials:
  - Getting started (very basic introduction)
    `pages.pomona.edu/~jsh04747/courses/R.pdf`
  - An introduction to R (more detailed)
    `http://cran.r-project.org/doc/manuals/R-intro.html`
  - Yet another introduction to R
    `cran.r-project.org/doc/manuals/R-intro.pdf`
  - And another (very good) one
    `http://zoonek2.free.fr/UNIX/48_R`

Getting Started

- Running R
  - R [RET]
- Reading data: vectors
  - x <- c(1, 2, 3, 4, 5)
    > x
    [1] 1 2 3 4 5
  - x <- c(1:5)
    > x
    [1] 1 2 3 4 5
  - x <- c("one", "two", "three", "four", "five")
    > x
    [1] "one" "two" "three" "four" "five"
- Reading data from file
  - $ cat myfile
    1 2 3 4 5
    > x <- scan("myfile")
    > x
    [1] 1 2 3 4 5

Getting Started

- Running R
  - R [RET]
- Reading data: vectors
  - x <- c(1, 2, 3, 4, 5)

    > x
    [1] 1 2 3 4 5
  - x <- c(1:5)

    > x
    [1] 1 2 3 4 5
  - x <- c("one", "two", "three", "four", "five")

    > x
    [1] "one" "two" "three" "four" "five"
- Reading data from file
  - $ cat myfile

    1 2 3 4 5

    > x <- scan("myfile")

    > x
    [1] 1 2 3 4 5

Getting Started

- Running R
  - R [RET]
- Reading data: vectors
  - x <- c(1, 2, 3, 4, 5)
    > x
    [1] 1 2 3 4 5
  - x <- c(1:5)
    > x
    [1] 1 2 3 4 5
  - x <- c("one", "two", "three", "four", "five")
    > x
    [1] "one" "two" "three" "four" "five"
- Reading data from file
  - $ cat myfile
    1 2 3 4 5
    > x <- scan("myfile")
    > x
    [1] 1 2 3 4 5

Getting Started

- Running R
  - R [RET]
- Reading data: vectors
  - x <- c(1, 2, 3, 4, 5)

    > x

    [1] 1 2 3 4 5
  - x <- c(1:5)

    > x

    [1] 1 2 3 4 5
  - x <- c("one", "two", "three", "four", "five")

    > x

    [1] "one" "two" "three" "four" "five"
- Reading data from file
  - $ cat myfile

    1 2 3 4 5

    > x <- scan("myfile")

    > x

    [1] 1 2 3 4 5

Getting Started

- Running R
  - R [RET]
- Reading data: vectors
  - x <- c(1, 2, 3, 4, 5)
    > x
    [1] 1 2 3 4 5
  - x <- c(1:5)
    > x
    [1] 1 2 3 4 5
  - x <- c("one", "two", "three", "four", "five")
    > x
    [1] "one" "two" "three" "four" "five"
- Reading data from file
  - $ cat myfile
    1 2 3 4 5
    > x <- scan("myfile")
    > x
    [1] 1 2 3 4 5

Getting Started

- Running R
  - R [RET]
- Reading data: vectors
  - x <- c(1, 2, 3, 4, 5)
    > x
    [1] 1 2 3 4 5
  - x <- c(1:5)
    > x
    [1] 1 2 3 4 5
  - x <- c("one", "two", "three", "four", "five")
    > x
    [1] "one" "two" "three" "four" "five"
- Reading data from file
  - $ cat myfile
    1 2 3 4 5
    > x <- scan("myfile")
    > x
    [1] 1 2 3 4 5

Getting Started

- Running R
  - R [RET]
- Reading data: vectors
  - x <- c(1, 2, 3, 4, 5)
    > x
    [1] 1 2 3 4 5
  - x <- c(1:5)
    > x
    [1] 1 2 3 4 5
  - x <- c("one", "two", "three", "four", "five")
    > x
    [1] "one" "two" "three" "four" "five"
- Reading data from file
  - $ cat myfile
    1 2 3 4 5
    > x <- scan("myfile")
    > x
    [1] 1 2 3 4 5

Getting Started

- Running R
  - R [RET]
- Reading data: vectors
  - x <- c(1, 2, 3, 4, 5)
    > x
    [1] 1 2 3 4 5
  - x <- c(1:5)
    > x
    [1] 1 2 3 4 5
  - x <- c("one", "two", "three", "four", "five")
    > x
    [1] "one" "two" "three" "four" "five"
- Reading data from file
  - $ cat myfile
    1 2 3 4 5
    > x <- scan("myfile")
    > x
    [1] 1 2 3 4 5

Getting Started

- Running R
  - R [RET]
- Reading data: vectors
  - x <- c(1, 2, 3, 4, 5)
    > x
    [1] 1 2 3 4 5
  - x <- c(1:5)
    > x
    [1] 1 2 3 4 5
  - x <- c("one", "two", "three", "four", "five")
    > x
    [1] "one" "two" "three" "four" "five"
- Reading data from file
  - $ cat myfile
    1 2 3 4 5
    > x <- scan("myfile")
    > x
    [1] 1 2 3 4 5

Getting Started

- Running R
  - R [RET]
- Reading data: vectors
  - x <- c(1, 2, 3, 4, 5)
    > x
    [1] 1 2 3 4 5
  - x <- c(1:5)
    > x
    [1] 1 2 3 4 5
  - x <- c("one", "two", "three", "four", "five")
    > x
    [1] "one" "two" "three" "four" "five"
- Reading data from file
  - $ cat myfile
    1 2 3 4 5
    > x <- scan("myfile")
    > x
    [1] 1 2 3 4 5

Getting Started (2)

- Simple statistics with R
  - > y <- c(1.5, 2.3, 2.5, 2.8, 3)
  - length(y)
  - mean(y)
  - min(y)
  - max(y)
  - median(y)
  - var(y)
  - sd(y)

  - What is sd?
    > help(sd)

Getting Started (2)

- Simple statistics with R
  - > y <- c(1.5, 2.3, 2.5, 2.8, 3)
  - length(y)
  - mean(y)
  - min(y)
  - max(y)
  - median(y)
  - var(y)
  - sd(y)

  - What is sd?
    > help(sd)

Getting Started (2)

- Simple statistics with R
  - > y <- c(1.5, 2.3, 2.5, 2.8, 3)
  - length(y)
  - mean(y)
  - min(y)
  - max(y)
  - median(y)
  - var(y)
  - sd(y)

  - What is sd?
    > help(sd)

Getting Started (2)

- Simple statistics with R
  - > y <- c(1.5, 2.3, 2.5, 2.8, 3)
  - length(y)
  - mean(y)
  - min(y)
  - max(y)
  - median(y)
  - var(y)
  - sd(y)

  - What is sd?
    > help(sd)

Getting Started (2)

- Simple statistics with R
  - > y <- c(1.5, 2.3, 2.5, 2.8, 3)
  - length(y)
  - mean(y)
  - min(y)
  - max(y)
  - median(y)
  - var(y)
  - sd(y)

  - What is sd?
    > help(sd)

Getting Started (2)

- Simple statistics with R
  - > y <- c(1.5, 2.3, 2.5, 2.8, 3)
  - length(y)
  - mean(y)
  - min(y)
  - max(y)
  - median(y)
  - var(y)
  - sd(y)

  - What is sd?
    > help(sd)

Getting Started (2)

- Simple statistics with R
  - > y <- c(1.5, 2.3, 2.5, 2.8, 3)
  - length(y)
  - mean(y)
  - min(y)
  - max(y)
  - median(y)
  - var(y)
  - sd(y)

  - What is sd?
    > help(sd)

Getting Started (2)

- Simple statistics with R
  - > y <- c(1.5, 2.3, 2.5, 2.8, 3)
  - length(y)
  - mean(y)
  - min(y)
  - max(y)
  - median(y)
  - var(y)
  - sd(y)

  - What is sd?
    > help(sd)

Getting Started (2)

- Simple statistics with R
  - > y <- c(1.5, 2.3, 2.5, 2.8, 3)
  - length(y)
  - mean(y)
  - min(y)
  - max(y)
  - median(y)
  - var(y)
  - sd(y)

  - What is sd?
    > help(sd)

Getting Started (3)

- Plotting data
  - > dotchart(y)
  - > plot(x,y)
  - > plot(x,y, type="l")
  - plot(x, y, type="l", xlab="X-Axis", ylab="Y-Axis", main="My beautiful plot")

- Combine 2 vectors into a matrix
  - > matrix <- rbind(x, y)
    > matrix

Getting Started (3)

- Plotting data
  - > dotchart(y)
  - > plot(x,y)
  - > plot(x,y, type="l")
  - plot(x, y, type="l", xlab="X-Axis", ylab="Y-Axis", main="My beautiful plot")

- Combine 2 vectors into a matrix
  - > matrix <- rbind(x, y)
    > matrix

Getting Started (3)

- Plotting data
  - > dotchart(y)
  - > plot(x,y)
  - > plot(x,y, type="l")
  - plot(x, y, type="l", xlab="X-Axis", ylab="Y-Axis", main="My beautiful plot")

- Combine 2 vectors into a matrix
  - > matrix <- rbind(x, y)
    > matrix

Getting Started (3)

- Plotting data
  - > dotchart(y)
  - > plot(x,y)
  - > plot(x,y, type="l")
  - plot(x, y, type="l", xlab="X-Axis", ylab="Y-Axis", main="My beautiful plot")

- Combine 2 vectors into a matrix
  - > matrix <- rbind(x, y)
    > matrix

Getting Started (3)

- Plotting data
  - > dotchart(y)
  - > plot(x,y)
  - > plot(x,y, type="l")
  - plot(x, y, type="l", xlab="X-Axis", ylab="Y-Axis", main="My beautiful plot")
- Combine 2 vectors into a matrix
  - > matrix <- rbind(x, y)
    > matrix

Getting Started (4)

- Reading data from file: tables
  - > data <- read.table("data.POS", header=TRUE)
- How to show the whole table?
  > data
- How to show the first three rows?
  > data[1:3,]
- How to show the first five columns?
  > data[,1:5]
- How to show row 3 and 4 for columns 5 to 10?
  > data[3:4,5:10]
- How to show row 3 and 4 for columns 5, 6 and 10?
  > data[3:4,c(5,6,10)]

Getting Started (4)

- Reading data from file: tables
  - > data <- read.table("data.POS", header=TRUE)
- How to show the whole table?
  > data
- How to show the first three rows?
  > data[1:3,]
- How to show the first five columns?
  > data[,1:5]
- How to show row 3 and 4 for columns 5 to 10?
  > data[3:4,5:10]
- How to show row 3 and 4 for columns 5, 6 and 10?
  > data[3:4,c(5,6,10)]

Getting Started (4)

- Reading data from file: tables
  - > data <- read.table("data.POS", header=TRUE)
- How to show the whole table?
  > data
- How to show the first three rows?
  > data[1:3,]
- How to show the first five columns?
  > data[,1:5]
- How to show row 3 and 4 for columns 5 to 10?
  > data[3:4,5:10]
- How to show row 3 and 4 for columns 5, 6 and 10?
  > data[3:4,c(5,6,10)]

Getting Started (4)

- Reading data from file: tables
  - > data <- read.table("data.POS", header=TRUE)
- How to show the whole table?
  > data
- How to show the first three rows?
  > data[1:3,]
- How to show the first five columns?
  > data[,1:5]
- How to show row 3 and 4 for columns 5 to 10?
  > data[3:4,5:10]
- How to show row 3 and 4 for columns 5, 6 and 10?
  > data[3:4,c(5,6,10)]

Getting Started (4)

- Reading data from file: tables
  - > data <- read.table("data.POS", header=TRUE)
- How to show the whole table?
  > data
- How to show the first three rows?
  > data[1:3,]
- How to show the first five columns?
  > data[,1:5]
- How to show row 3 and 4 for columns 5 to 10?
  > data[3:4,5:10]
- How to show row 3 and 4 for columns 5, 6 and 10?
  > data[3:4,c(5,6,10)]

Getting Started (4)

- Reading data from file: tables
  - > data <- read.table("data.POS", header=TRUE)
- How to show the whole table?
  > data
- How to show the first three rows?
  > data[1:3,]
- How to show the first five columns?
  > data[,1:5]
- How to show row 3 and 4 for columns 5 to 10?
  > data[3:4,5:10]
- How to show row 3 and 4 for columns 5, 6 and 10?
  > data[3:4,c(5,6,10)]

Getting Started (4)

- Reading data from file: tables
  - > data <- read.table("data.POS", header=TRUE)
- How to show the whole table?
  > data
- How to show the first three rows?
  > data[1:3,]
- How to show the first five columns?
  > data[,1:5]
- How to show row 3 and 4 for columns 5 to 10?
  > data[3:4,5:10]
- How to show row 3 and 4 for columns 5, 6 and 10?
  > data[3:4,c(5,6,10)]

Getting Started (4)

- Reading data from file: tables
  - > data <- read.table("data.POS", header=TRUE)
- How to show the whole table?
  > data
- How to show the first three rows?
  > data[1:3,]
- How to show the first five columns?
  > data[,1:5]
- How to show row 3 and 4 for columns 5 to 10?
  > data[3:4,5:10]
- How to show row 3 and 4 for columns 5, 6 and 10?
  > data[3:4,c(5,6,10)]

Getting Started (4)

- Reading data from file: tables
  - > data <- read.table("data.POS", header=TRUE)
- How to show the whole table?
  > data
- How to show the first three rows?
  > data[1:3,]
- How to show the first five columns?
  > data[,1:5]
- How to show row 3 and 4 for columns 5 to 10?
  > data[3:4,5:10]
- How to show row 3 and 4 for columns 5, 6 and 10?
  > data[3:4,c(5,6,10)]

Getting Started (4)

- Reading data from file: tables
  - > data <- read.table("data.POS", header=TRUE)
- How to show the whole table?
  > data
- How to show the first three rows?
  > data[1:3,]
- How to show the first five columns?
  > data[,1:5]
- How to show row 3 and 4 for columns 5 to 10?
  > data[3:4,5:10]
- How to show row 3 and 4 for columns 5, 6 and 10?
  > data[3:4,c(5,6,10)]

Getting Started (4)

- Reading data from file: tables
  - > data <- read.table("data.POS", header=TRUE)
- How to show the whole table?
  > data
- How to show the first three rows?
  > data[1:3,]
- How to show the first five columns?
  > data[,1:5]
- How to show row 3 and 4 for columns 5 to 10?
  > data[3:4,5:10]
- How to show row 3 and 4 for columns 5, 6 and 10?
  > data[3:4,c(5,6,10)]

Getting Started (4)

- Reading data from file: tables
  - > data <- read.table("data.POS", header=TRUE)
- How to show the whole table?
  > data
- How to show the first three rows?
  > data[1:3,]
- How to show the first five columns?
  > data[,1:5]
- How to show row 3 and 4 for columns 5 to 10?
  > data[3:4,5:10]
- How to show row 3 and 4 for columns 5, 6 and 10?
  > data[3:4,c(5,6,10)]

Getting Started (5)

- Executing files with R commands
  - $ cat names.row
    row <- c("A.1", "A.2", "A.3", "A.4", "A.5", "N.1", "N.2", "N.3", "N.4", "N.5", "O.1", "O.2", "O.3", "O.4", "O.5", "T.1", "T.2", "T.3", "T.4", "T.5", "W.1", "W.2", "W.3", "W.4", "W.5")
  - > source("names.row")
    > row
  - add row names to the table
    row.names(data) <- row
    > data

Getting Started (5)

- Executing files with R commands
  - $ cat names.row
    row <- c("A.1", "A.2", "A.3", "A.4", "A.5", "N.1", "N.2", "N.3", "N.4", "N.5", "O.1", "O.2", "O.3", "O.4", "O.5", "T.1", "T.2", "T.3", "T.4", "T.5", "W.1", "W.2", "W.3", "W.4", "W.5")
  - > source("names.row")
    > row
  - add row names to the table
    row.names(data) <- row
    > data

Getting Started (5)

- Executing files with R commands
  - $ cat names.row
    row <- c("A.1", "A.2", "A.3", "A.4", "A.5", "N.1", "N.2", "N.3", "N.4", "N.5", "O.1", "O.2", "O.3", "O.4", "O.5", "T.1", "T.2", "T.3", "T.4", "T.5", "W.1", "W.2", "W.3", "W.4", "W.5")
  - > source("names.row")
    > row
  - add row names to the table
    row.names(data) <- row
    > data

Getting Started (5)

- Executing files with R commands
  - $ cat names.row
    row <- c("A.1", "A.2", "A.3", "A.4", "A.5", "N.1", "N.2", "N.3", "N.4", "N.5", "O.1", "O.2", "O.3", "O.4", "O.5", "T.1", "T.2", "T.3", "T.4", "T.5", "W.1", "W.2", "W.3", "W.4", "W.5")
  - > source("names.row")
    > row
  - add row names to the table
    row.names(data) <- row
    > data

Getting Started (5)

- Executing files with R commands
  - $ cat names.row
    row <- c("A.1", "A.2", "A.3", "A.4", "A.5", "N.1", "N.2", "N.3", "N.4", "N.5", "O.1", "O.2", "O.3", "O.4", "O.5", "T.1", "T.2", "T.3", "T.4", "T.5", "W.1", "W.2", "W.3", "W.4", "W.5")
  - > source("names.row")
    > row
  - add row names to the table
    row.names(data) <- row
    > data

Getting Started (5)

- Executing files with R commands
  - $ cat names.row
    row <- c("A.1", "A.2", "A.3", "A.4", "A.5", "N.1", "N.2", "N.3", "N.4", "N.5", "O.1", "O.2", "O.3", "O.4", "O.5", "T.1", "T.2", "T.3", "T.4", "T.5", "W.1", "W.2", "W.3", "W.4", "W.5")
  - > source("names.row")
    > row
  - add row names to the table
    row.names(data) <- row
    > data

Getting Started (5)

- Executing files with R commands
  - $ cat names.row
    row <- c("A.1", "A.2", "A.3", "A.4", "A.5", "N.1", "N.2", "N.3", "N.4", "N.5", "O.1", "O.2", "O.3", "O.4", "O.5", "T.1", "T.2", "T.3", "T.4", "T.5", "W.1", "W.2", "W.3", "W.4", "W.5")
  - > source("names.row")
    > row
  - add row names to the table
    row.names(data) <- row
    > data

Multi-Dimensional Data

- Many variables, lots of data points in high-dimensional space
  - hard to interpret
  - hard to detect underlying patterns
- **Goal**: find the most important variables which explain a large part of your data
- Reduce the dimensions without loosing information, merge a high number of highly correlated variables into a smaller number of new, non-correlated variables

Multi-Dimensional Data

- Many variables, lots of data points in high-dimensional space
  - hard to interpret
  - hard to detect underlying patterns
- **Goal**: find the most important variables which explain a large part of your data
- Reduce the dimensions without loosing information, merge a high number of highly correlated variables into a smaller number of new, non-correlated variables

Principal Components Analysis (PCA)

PCA reduces complex, high-dimensional data and looks for underlying patterns

- Transform a high number of (possibly) correlating variables into a smaller number of non-correlating new variables (*eigen vectors*)
- Select the variables which describe the largest part of the variance in the data, combine them into a new variable
- PCA was successfully used for the analysis of register variation (*Biber 1998*) and for authorship detection (*Juala & Baayen 1998*)

PCA (in our experiment) is based on the frequency of POS-tags in text samples in order to describe the differences between different genres/domains

Principal Components Analysis (PCA)

PCA reduces complex, high-dimensional data and looks for underlying patterns

- Transform a high number of (possibly) correlating variables into a smaller number of non-correlating new variables (*eigen vectors*)
- Select the variables which describe the largest part of the variance in the data, combine them into a new variable
- PCA was successfully used for the analysis of register variation (*Biber 1998*) and for authorship detection (*Juala & Baayen 1998*)

PCA (in our experiment) is based on the frequency of POS-tags in text samples in order to describe the differences between different genres/domains

Principal Components Analysis (2)

- Data: samples from different domains (500 words/sample, POS-tagged)
  1. A: childrens books (L. Carroll, "Alice in Wonderland")
  2. N: newspaper (New York Times)
  3. W: newspaper (Wall Street Journal)
  4. O: fiction (H. MacMahon, "Orphans of the Strom", 1922)
  5. T: non-fiction (T. Smith, "What Germany Thinks", 1915)

- How to proceed:
  1. Standardise data ($z_{nj} = \frac{x_{nj} - \bar{x}_j}{sd_j}$)
     data matrix with $mean = 0$ and $sd = 1$
  2. Compute correlation matrix: Which of the variables show a high correlation to each other? (those are the ones we want to merge)
  3. Extract principal components
     (no math here, just the basic idea of PCA)

Principal Components Analysis (2)

- Data: samples from different domains (500 words/sample, POS-tagged)
  1. A: childrens books (L. Carroll, "Alice in Wonderland")
  2. N: newspaper (New York Times)
  3. W: newspaper (Wall Street Journal)
  4. O: fiction (H. MacMahon, "Orphans of the Strom", 1922)
  5. T: non-fiction (T. Smith, "What Germany Thinks", 1915)

- How to proceed:
  1. Standardise data ($z_{nj} = \frac{x_{nj} - \bar{x}_j}{sd_j}$)
     data matrix with $mean = 0$ and $sd = 1$
  2. Compute correlation matrix: Which of the variables show a high correlation to each other? (those are the ones we want to merge)
  3. Extract principal components
     (no math here, just the basic idea of PCA)

Principal Components Analysis (3)

- Tag your text samples (e.g. treetagger)
- Count the number of each POS tag in each of the samples (e.g. simple perl script *countPOS_en.pl*)
- Write frequencies for all POS tags into one file
- Use all variables (POS tags)? Select some of them?
- File *data.POS*
  > cat data.POS

Principal Components Analysis (3)

- Tag your text samples (e.g. treetagger)
- Count the number of each POS tag in each of the samples (e.g. simple perl script *countPOS_en.pl*)
- Write frequencies for all POS tags into one file
- Use all variables (POS tags)? Select some of them?
- File *data.POS*
  > cat data.POS

Principal Components Analysis (3)

- Tag your text samples (e.g. treetagger)
- Count the number of each POS tag in each of the samples (e.g. simple perl script *countPOS_en.pl*)
- Write frequencies for all POS tags into one file
- Use all variables (POS tags)? Select some of them?
- File *data.POS*
  > cat data.POS

Principal Components Analysis (3)

- Tag your text samples (e.g. treetagger)
- Count the number of each POS tag in each of the samples (e.g. simple perl script *countPOS_en.pl*)
- Write frequencies for all POS tags into one file
- Use all variables (POS tags)? Select some of them?
- File *data.POS*
  > cat data.POS

Principal Components Analysis (3)

- Tag your text samples (e.g. treetagger)
- Count the number of each POS tag in each of the samples (e.g. simple perl script *countPOS_en.pl*)
- Write frequencies for all POS tags into one file
- Use all variables (POS tags)? Select some of them?
- File *data.POS*
  > cat data.POS

Principal Components Analysis (4)

- Read the data
  > data <- read.table("data.POS", header=T)
- Read row names for data (stored in file names.row)
  > source("names.row")
- Display data
  > data
- Display row names
  > row
- Add row names to data
  > names.row(data) <- row
- Run a PCA
  > data.pca <- prcomp(data)

Principal Components Analysis (4)

- Read the data
  > data <- read.table("data.POS", header=T)
- Read row names for data (stored in file names.row)
  > source("names.row")
- Display data
  > data
- Display row names
  > row
- Add row names to data
  > names.row(data) <- row
- Run a PCA
  > data.pca <- prcomp(data)

Principal Components Analysis (4)

- Read the data
  > data <- read.table("data.POS", header=T)
- Read row names for data (stored in file names.row)
  > source("names.row")
- Display data
  > data
- Display row names
  > row
- Add row names to data
  > names.row(data) <- row
- Run a PCA
  > data.pca <- prcomp(data)

Principal Components Analysis (4)

- Read the data
  > data <- read.table("data.POS", header=T)
- Read row names for data (stored in file names.row)
  > source("names.row")
- Display data
  > data
- Display row names
  > row
- Add row names to data
  > names.row(data) <- row
- Run a PCA
  > data.pca <- prcomp(data)

Principal Components Analysis (4)

- Read the data
  > data <- read.table("data.POS", header=T)
- Read row names for data (stored in file names.row)
  > source("names.row")
- Display data
  > data
- Display row names
  > row
- Add row names to data
  > names.row(data) <- row
- Run a PCA
  > data.pca <- prcomp(data)

Principal Components Analysis (4)

- Read the data
  > data <- read.table("data.POS", header=T)
- Read row names for data (stored in file names.row)
  > source("names.row")
- Display data
  > data
- Display row names
  > row
- Add row names to data
  > names.row(data) <- row
- Run a PCA
  > data.pca <- prcomp(data)

Principal Components Analysis (5)

- How many of the components should we consider?
  Run a scree-test
  $>$ screetest(data.pca)
- Scree-test shows eigen values (part of the variance in the data which can be explained by this component)
- *Loadings* describe the relation between a component and a feature (correlation between feature and component)
- *Scores* describe the strenght of the relation between all relevant features for one component and our subject (here: text sample)
- Each observation (data point) can be explained by the sum of the products of all its scores f and the loadings a for each component
  $z_{nj} = a_{j1} \times f_{n1} + a_{j1} \times f_{n1} + ... + a_{jq} \times f_{nq}$
  z: standardised variable, a: loading, f: score, j: feature (e.g. NN)
- We are looking for the components which explain the largest part of the variance in the data $>$ summary(data.pca)

Principal Components Analysis (5)

- How many of the components should we consider?
  Run a scree-test
  $>$ screetest(data.pca)
- Scree-test shows eigen values (part of the variance in the data which can be explained by this component)
- *Loadings* describe the relation between a component and a feature (correlation between feature and component)
- *Scores* describe the strenght of the relation between all relevant features for one component and our subject (here: text sample)
- Each observation (data point) can be explained by the sum of the products of all its scores f and the loadings a for each component
  $z_{nj} = a_{j1} \times f_{n1} + a_{j1} \times f_{n1} + ... + a_{jq} \times f_{nq}$
  z: standardised variable, a: loading, f: score, j: feature (e.g. NN)
- We are looking for the components which explain the largest part of the variance in the data $>$ summary(data.pca)

Principal Components Analysis (5)

- How many of the components should we consider?
  Run a scree-test
  $>$ screetest(data.pca)
- Scree-test shows eigen values (part of the variance in the data which can be explained by this component)
- *Loadings* describe the relation between a component and a feature (correlation between feature and component)
- *Scores* describe the strenght of the relation between all relevant features for one component and our subject (here: text sample)
- Each observation (data point) can be explained by the sum of the products of all its scores f and the loadings a for each component
  $z_{nj} = a_{j1} \times f_{n1} + a_{j1} \times f_{n1} + ... + a_{jq} \times f_{nq}$
  z: standardised variable, a: loading, f: score, j: feature (e.g. NN)
- We are looking for the components which explain the largest part of the variance in the data
  $>$ summary(data.pca)

Principal Components Analysis (5)

- How many of the components should we consider?
  Run a scree-test
  > screetest(data.pca)
- Scree-test shows eigen values (part of the variance in the data which can be explained by this component)
- *Loadings* describe the relation between a component and a feature (correlation between feature and component)
- *Scores* describe the strenght of the relation between all relevant features for one component and our subject (here: text sample)
- Each observation (data point) can be explained by the sum of the products of all its scores f and the loadings a for each component
  $z_{nj} = a_{j1} \times f_{n1} + a_{j1} \times f_{n1} + ... + a_{jq} \times f_{nq}$
  z: standardised variable, a: loading, f: score, j: feature (e.g. NN)
- We are looking for the components which explain the largest part of the variance in the data
  > summary(data.pca)

Principal Components Analysis (6)

- Now let's look at the components
  $>$ biplot(data.pca)
  projects the data along the dimensions for the first two principal components

- More components
  $>$ biplot(data.pca, choices=3:4)
  look at the third and fourth component
- Red arrows show variables and their loadings along the two components (long arrow $\Rightarrow$ strong (positive or negative) loading)

Principal Components Analysis (6)

- Now let's look at the components
  $>$ biplot(data.pca)
  projects the data along the dimensions for the first two principal components

- More components
  $>$ biplot(data.pca, choices=3:4)
  look at the third and fourth component
- Red arrows show variables and their loadings along the two components (long arrow $\Rightarrow$ strong (positive or negative) loading)

Principal Components Analysis (7)

RB	adverb
NP	proper noun, singular
CC	coordinating conjunction
PP	personal pronoun
CD	cardinal number
VBD	verb, past tense
DT	determiner
VV	verb, base form
IN	preposition/subordinating conjunction
VVD	verb, past tense
JJ	adjective
VVG	verb, present participle or gerund
MD	modal
VVN	verb, past participle
NN	noun, singular or mass
WRB	Wh-adverb