Programming and Analyzing in R: Data, Models, and Plots

Wouter van Atteveldt http://vanatteveldt.com

May 2016

Overview

- Session I
  - Data types
  - Transforming Data
- Session II
  - Basic modelling
  - Visualization

Primitives

- `numeric`
  - `integer`
- `character`
- `Date`
  - (and many variants)
- `factor`
  - "value labels"
  - faster than strings
  - automatically created in data frames
  - `as.factor()`, `factor()` functions
  - zero values in tables: `droplevels()`

Collections

- Data frame
  - rectangular, type per column
  - use `[rows, columns]`, `[columns]`
    - or `$column` / `[[column]]`
- List
  - any element per item
  - use `$item` / `[[item]]` or `[items]`
- Matrix
  - rectangular, one type
  - use `[rows, columns]`

Selecting columns is confusing

- Lists and data frames:
  - `d$a` returns a vector
  - `d[["a"]]` returns a vector
  - `d["a"]` returns a data frame
- Data frames and matrices:
  - `d[, c("a","b")]` returns a data frame
  - `d[, "a"]` returns a vector
  - `d[, "a", drop=F]` returns a data frame

Creating/transforming data types

- as.numeric, as.integer, as.character
- factor, as.factor
- data.frame, as.data.frame
- matrix, as.matrix
- list, as.list

Transforming data

- Rename, recode, compute
- Combining data
  - adding rows/columns
  - merging data
  - matching data
- Aggregating data
- Cases to variables
  - 'reshape' package

Renaming columns

- `colnames(d) = c("jaar", "nl", "us")`
- `colnames(d)[1] = "jaar"`
- `d = rename(d, c("old"="new"))`
  - package `plyr`

Recoding data

```
x$cat[x$US > 0.5] = "hoog"

x$cat = ifelse(x$US > 0.5, "hoog", "laag")

library(car)
x$period = recode(x$year,
 "lo:1945='pre'; 1945:1980='rec'; else='mod'")

x$period = cut(x$year, c(1900,1945,1980,2015),
               labels=c('pre', 'rec', 'mod'))
```

Working with strings

```
x$name = paste(x$firstname, x$surname)
x$name = tolower(x$name)
x$name = trimws(x$name)

x$height = sub(",", ".", x$height)

x[grepl("van der", x$name, ignore.case=T)
```

Ordering data

```
x = x[order(x$age), ]
x = x[order(-x$age), ]

library(plyr)
x = arrange(x, age)
```

Hands-on 2a

- `3_organizing.html`

Combining data

- Adding rows: `rbind`
- Adding columns: `cbind`

Merging data

- Merge on common key column(s)
  - `merge(a, b)`
  - `merge(a, b, by="year")`
  - `merge(a, b, by="year", all.x=T)`
  - `merge(a, b, by.x="year", by.y="jaar")`

Matching data

- Match returns the position of elements in another list
- `match(a$year, b$year)`
- `a$fr = b$fr[match(a$year, b$year)`

Reshape and aggregate

- Package `reshape2`:
- `melt(data, id.vars=)`
  - Variables to cases
- `dcast(data, rows~cols, value.var=)`
  - Cases to variables / tabulation
  - `fun.aggregate=mean`
- `aggregate(data, by=)`
  - Aggregating

Simple two variable tests

- `t.test(x, y, paired=.)`
- `cor.test(x, y)`
- `chisq.test(x,y)`

Linear models (and anova/ancova)

- `lm(y ~ x1 + x2)`
- `lm(Y ~ x1*x2)`
- `lm(Y ~ x1 + x2 + x1:x2)`
- `lm(Y ~ x1 + cat)`
- `lm(Y ~ x1 + cat - 1)`
- `aov(y ~ X)`

Modeling results

- `m = lm(..)`
- `summary(m)`
- `anova(m)`
- `coef(m)`
- `fitted(m)`
- `residuals(m)`
- `plot(m)`
- `anova(m1, m2)`
- `http://www.statmethods.net/stats/rdiagnostics.html`

Hands-on 2b

- `4_transforming.html`
- `5_modeling.html`