LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Statistische Software (R)

Paul Fink, M.Sc., Eva Endres, M.Sc.

Institut für Statistik
Ludwig-Maximilians-Universität München

Daten einlesen

Daten – DRY-Prinzip

Daten als eine Zusammenstellung von Informationen zu einem bestimmten Thema.

Am Sonntag 11.05.2014 hatte es in München Viktualienmarkt um Punkt 12 Uhr $15°\mathrm{C}$, während es am Samstag 10.05.2014 dort um Punkt 12 Uhr noch $20°\mathrm{C}$ waren. Am Freitag zuvor waren es zur selben Zeit am selben Ort allerdings erst $12°\mathrm{C}$. ...

Viel Irrelevantes und Informationen doppelt
$\Longrightarrow$ **DRY-Prinzip**

(Meta)Daten

Es geht hier um Temperaturbeobachtungen (gemessen in °C)
an einem bestimmten Ort (Viktualienmarkt in München)
zu jeweils einer bestimmten Zeit (12 Uhr)

Metadaten:
Wichtige Informationen über Daten; gleich für alle Beobachtungen

Rechteck-Schema:
Darstellung der eigentlich variablen Daten in einer Tabelle, mit einer Zeile je Beobachtung

Wochentag	Datum	Temperatur
Sonntag	11.05.2014	15
Samstag	10.05.2014	20
Freitag	09.05.2014	12
⋮	⋮	⋮

Datenformate - Textformat

Jedes Zeichen der Information wird als Text gespeichert.

Spezielle Zeichen trennen die Spalten innerhalb einer Zeile

Vorteil: Sehr einfaches Format, einfach zu editieren

Nachteil: Geht in der Regel nur für Datensätze im sog. Rechteck-Schema

Beispiel: CSV- oder Fixed-Width-Format

R: Einlesen wird direkt unterstützt

Datenformate - Binärformat

Zeichen werden unterschiedlich gespeichert, je nach Typ:
Ganze Zahl, reelle Zahl, Text, ...

$\Longrightarrow$ *Intelligentes* Speichern

Vorteil: Spart Platz, kann zur Vermeidung von Redundanz verwendet werden

Nachteil: Man braucht spezielle Software dazu, die auflöst, was als was gespeichert ist.

Beispiel: SPSS sav-Dateien, Excel-Spreadsheets

R: Einlesen wird durch Zusatzpakete unterstützt

Pfadangaben

Jede Datei auf dem Computer liegt an einem Ort in der Ordner-Baum-Struktur, identifiziert über den sogenannten *Pfad*.

Jede Ordnerebene wird über Pfadtrenner verbunden, bei MS Windows \, bei Mac und *nix-Systemen /

```
> # Geht auch unter Windows
> (pfad <- "Der/Pfad/zu/meiner/Datei")
[1] "Der/Pfad/zu/meiner/Datei"
> # Nur unter Windows
> (pfad_ms <- "Der\\Pfad\\zu\\meiner\\Datei")
[1] "Der\\Pfad\\zu\\meiner\\Datei"
> # Pfadtrenner automatisch eingefuegt
> (pfad_r <- file.path("Der", "Pfad", "zu", "meiner", "Datei"))
[1] "Der/Pfad/zu/meiner/Datei"
```

Relative vs. Absolute Pfadangabe

Bei der absoluten Angabe muss man immer im OS-Wurzelverzeichnis starten (Laufwerksbuchstabe bei Windows)

```
> "C:/Der/absoulte/Pfad/zu/meiner/Datei"
[1] "C:/Der/absoulte/Pfad/zu/meiner/Datei"
```

Bei der relativen Angabe nimmt R das aktuelle Arbeitsverzeichnis und geht von dort aus weiter

```
> "Der/relative/Pfad/zu/meiner/Datei"
[1] "Der/relative/Pfad/zu/meiner/Datei"
> # entspricht
> # file.path(getwd(),"Der/relative/Pfad/zu/meiner/Datei")
```

Spezielle Verzeichnisnamen:

`".."`	Verzeichnisebene oberhalb der aktuellen
`"."`	Aktueller Ordnerebene

Arbeitsverzeichnis

Verzeichnis, in dem man mit R arbeitet:

- Speicherort für History und Workspace
- Wurzelverzeichnis für relative Pfadangaben

```
> getwd() # Abfragen
> setwd("Pfad/zum/arbeits/verzeichnis")
> ?read.table
```

Einlesen in R - Textformat

DIE grundlegende Funktion in R zum Einlesen von Textformaten ist `read.table()`

Funktionen zum Einlesen von bestimmten Textformaten, rufen meist nur die Funktion `read.table()` mit anderen vorgegeben Argumenten auf, zum Beispiel `read.csv2()`

Hilfe: `?read.table`

Die Funktion liefert nach dem Einlesen ein Objekt vom Typ `data.frame` zurück

Einlesen in R - Textformat

Standardmäßig wandelt R beim Einlesen die Spalten in geeignete Formate um:

- Spalten mit Zahlen werden als `numeric` eingelesen
- Spalten mit Text werden als `factor` eingelesen

Wichtige Argumente für `read.table()`:

`header`:	Erste Zeile enthält Variablennamen
`sep`:	Trennzeichen zwischen den Spalten
`dec`:	Dezimaltrennzeichen (1.3 vs. 1,3)
`as.is`:	Keine automatische Umwandlung
`colClasses`:	Typ der Spalten vorgeben

Einlesen in R - Binärformat

Daten im Excel-Format *xls(x)* :

- Im csv-Format speichern und als Textformat einlesen
- Einlesen mit Funktion `read.xlsx()` aus Paket `openxlsx`

Daten in SPSS-Format *sav* :

- Einlesen mit Funktion `read.spss()` aus Paket `foreign`

TIPP: Einlesen aus Binärformaten vermeiden!

Aufgaben

1. Erstellen Sie einen leeren Ordner und setzen Sie das R-Arbeitsverzeichnis auf diesen Ordner.
2. Überprüfen Sie, ob der Wechsel des Arbeitsverzeichnisses erfolgreich war.
3. Speichern Sie sich das Zip-Archiv `beispieldaten.zip` von der Veranstaltungshomepage in den in 1. erstellten Ordner und entpacken Sie es.
4. Erstellen Sie einen Zeichenketten-Vektor `datensaetze`, der die vollen Dateinamen (also auch mit Endung) der Dateien aus dem entpackten Zip-Archiv enthält.

Aufgaben

5. Erstellen Sie ein R-Objekt `absPfade`, das die **absoluten** Pfade der entpackten Dateien enthält. Hinweis: `file.path()`
6. Lesen Sie die Beispieldaten in allen Formaten korrekt in R ein. Benutzen Sie dazu die absoluten Pfadangaben in `absPfade`.
7. Erstellen Sie ein R-Objekt `relPfade`, das die **relativen** Pfade der entpackten Dateien enthält. Hinweis: `file.path()`
8. Lesen Sie die Beispieldaten in allen Formaten korrekt in R ein. Benutzen Sie dazu die relativen Pfadangabe in `relPfade`.