Datenaufbereitung

Jan-Philipp Kolb

21 Juni 2017

Data Frames

Beispieldaten einlesen:

```
library(foreign)
dat<-read.dta("https://github.com/Japhilko/RSocialScience/
                raw/master/data/GPanel.dta")
```

Den Datensatz anschauen

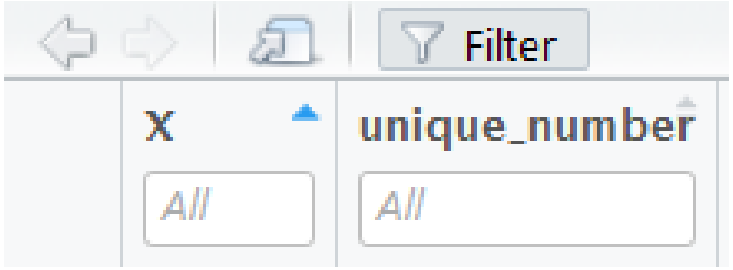

Figure 1:

```
typeof(dat)
```

```
## [1] "list"
```

In Dataframe übertragen

Diese beiden Vektoren zu einem data.frame verbinden:

```
Daten <- data.frame(dat)
```

Anzahl der Zeilen/Spalten herausfinden

```
nrow(Daten) # Zeilen
```

```
## [1] 100
```

```
ncol(Daten) # Spalten
```

```
## [1] 23
```

Die Daten anschauen

- die ersten zeilen anschauen

```
head(Daten)
```

- Übersicht mittels Rstudio

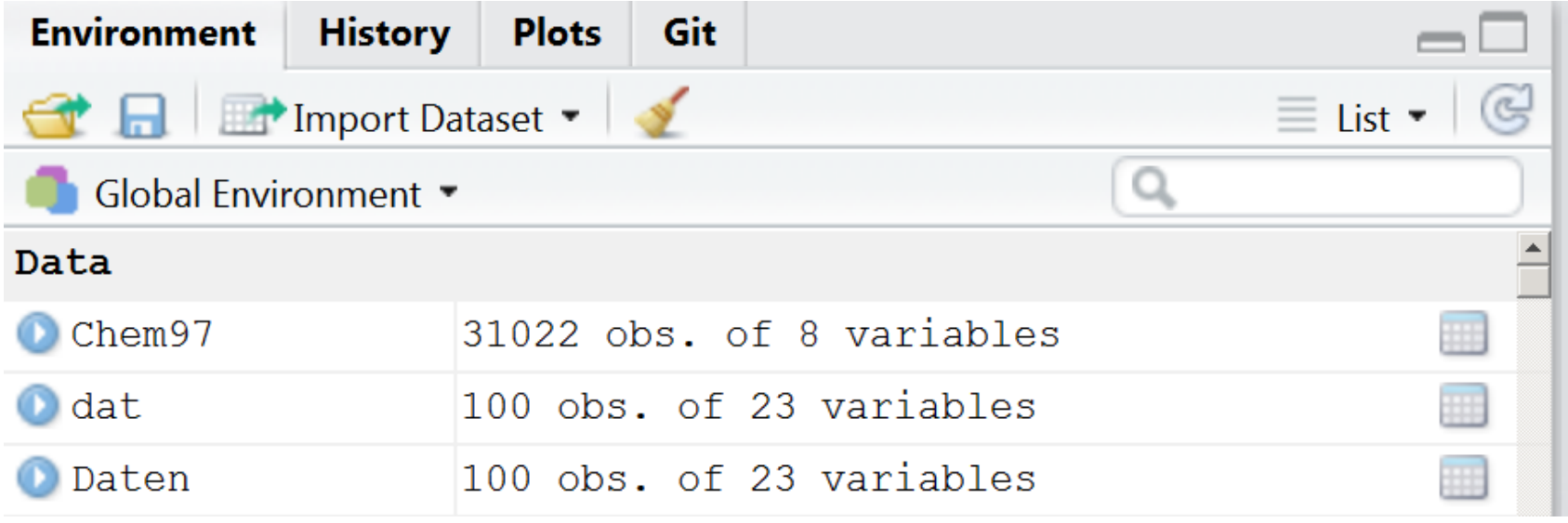

Figure 2:

Indizieren

Indizieren eines dataframe:

```
Daten[1,1]
```

```
## [1] Eher zufrieden
## 7 Levels: Item nonresponse Sehr zufrieden ... Weiß nicht
```

```
Daten[2,]
```

```
##         a11c019a         a11c020a         a11c021a
## 2 Sehr zufrieden Eher unzufrieden Eher unzufrieden Stimm
##         a11c023a                  a11c024a
## 2 Stimme eher zu Stimme voll und ganz zu Eher niedrigere
##             a11c026a                    a11c027a a11c028a
## 2 Mehrmals die Woche Mindestens einmal im Monat  Täglich
##                     a11c031a          a11c032a a11c033a
## 2 Mindestens einmal im Monat Mehrmals im Monat Seltener
##                                 a11c034a bazq020a a11d054
```

Operatoren um Subset für Datensatz zu bekommen

Diese Operatoren eignen sich gut um Datensätze einzuschränken

```
Dauer <- as.numeric(Daten$bazq020a)
```

```
## Warning: NAs durch Umwandlung erzeugt
```

```
head(Daten[Dauer>20,])
```

```
##          a11c019a          a11c020a          a11c021a
## 2  Sehr zufrieden Eher unzufrieden Eher unzufrieden Stir
## 3  Eher zufrieden   Sehr zufrieden Eher unzufrieden Stir
## 5  Eher zufrieden   Eher zufrieden   Eher zufrieden
## NA            <NA>              <NA>              <NA>
## 9  Sehr zufrieden   Eher zufrieden   Sehr zufrieden
## 15 Sehr zufrieden   Sehr zufrieden   Sehr zufrieden
##                a11c023a                a11c024a
## 2       Stimme eher zu Stimme voll und ganz zu
## 3  Stimme eher nicht zu          Stimme eher zu
```

Einschränken mit dem Paket `tidyverse`

- einfacher geht es mit dem Paket `tidyverse`

```
library(tidyverse)
```

```
## Loading tidyverse: ggplot2
## Loading tidyverse: tibble
## Loading tidyverse: tidyr
## Loading tidyverse: readr
## Loading tidyverse: purrr
## Loading tidyverse: dplyr

## Conflicts with tidy packages ----------------------------

## filter(): dplyr, stats
## lag():    dplyr, stats
```

```
filter(Daten, Dauer>20)
```

```
##                                a11c019a
```

Datensätze einschränken

```
SEX <- Daten$a11d054a

Daten[SEX=="Männlich",]
```

```
##                                a11c019a
## 1                    Eher zufrieden Weder zufrieden noch
## 2                    Sehr zufrieden                  Eher
## 3                    Eher zufrieden                    Se
## 4                    Eher zufrieden                    Se
## 5                    Eher zufrieden                    Eh
## 7                    Eher zufrieden                    Eh
## 9                    Sehr zufrieden                    Eh
## 12                   Sehr zufrieden                    Eh
## 15                   Sehr zufrieden                    Se
## 16                   Sehr zufrieden                    Se
## 17 Weder zufrieden noch unzufrieden                  Eher
## 18                   Eher zufrieden                    Se
## 20                   Eher zufrieden                    Eh
```

Weitere wichtige Optionen

```r
# Ergebnis in ein Objekt speichern
subDat <- Daten[Dauer>20,]
```

```r
# mehrere Bedingungen können mit
# & verknüpft werden:
Daten[Dauer>18 & SEX=="Männlich",]
```

```
##                          a11c019a
## 2          Sehr zufrieden
## 3          Eher zufrieden
## 5          Eher zufrieden
## 9          Sehr zufrieden
## 15         Sehr zufrieden
## 18         Eher zufrieden
## 20         Eher zufrieden
## 29         Sehr zufrieden
## 41         Sehr zufrieden
## 48         Eher zufrieden
```

Die Nutzung einer Sequenz

```
Daten[1:3,]
```

```
##          a11c019a                                a11c020a
## 1 Eher zufrieden Weder zufrieden noch unzufrieden   Sehr
## 2 Sehr zufrieden                 Eher unzufrieden Eher u
## 3 Eher zufrieden                   Sehr zufrieden Eher u
##                 a11c022a                a11c023a
## 1 Stimme eher nicht zu       Stimme eher zu          Sti
## 2 Stimme eher nicht zu       Stimme eher zu Stimme voll
## 3 Stimme eher nicht zu Stimme eher nicht zu          Sti
##                          a11c025a          a11c026a
## 1 Eher niedrigeren Lebensstandard          Seltener
## 2 Eher niedrigeren Lebensstandard Mehrmals die Woche
## 3        Denselben Lebensstandard           Täglich
##                       a11c027a a11c028a a11c029a
## 1         Mehrmals die Woche  Täglich  Täglich
## 2 Mindestens einmal im Monat  Täglich  Täglich
## 3          Mehrmals im Monat      Nie  Täglich Mehrmals
```

Variablen Labels

```
library(foreign)
dat <- read.dta("https://github.com/Japhilko/RSocialScience
```

```
attributes(dat)
```

```
var.labels <- attr(dat,"var.labels")
```

- Genauso funktioniert es auch mit dem Paket haven

```
library(haven)
dat2 <- read_dta(
"https://github.com/Japhilko/RSocialScience/blob/master/dat
var.labels2 <- attr(dat,"var.labels")
```

Die Spaltennamen umbenennen

- Mit `colnames` bekommt man die Spaltennamen angezeigt

```
colnames(dat)
```

```
##   [1] "a11c019a" "a11c020a" "a11c021a" "a11c022a" "a11c02
##   [7] "a11c025a" "a11c026a" "a11c027a" "a11c028a" "a11c02
##  [13] "a11c031a" "a11c032a" "a11c033a" "a11c034a" "bazq02
##  [19] "a11d056z" "a11d092a" "a11c100a" "a11c111a" "a11c10
```

- So kann man die Spaltennamen umbenennen:

```
colnames(dat) <-var.labels
```

- Analog geht das für die Reihennamen

```
rownames(dat)
```

```
##    [1] "1"   "2"   "3"   "4"   "5"   "6"   "7"   "8"   "9
##   [12] "12"  "13"  "14"  "15"  "16"  "17"  "18"  "19"  "2
```

Indizieren

- Das Dollarzeichen kann man auch nutzen um einzelne Spalten anzusprechen

```
head(dat$a11c019a)
```

```
## [1] Eher zufrieden Sehr zufrieden Eher zufrieden Eher zu
## [5] Eher zufrieden Sehr zufrieden
## 7 Levels: Item nonresponse Sehr zufrieden ... Weiß nicht
```

```
dat$a11c019a[1:10]
```

```
##  [1] Eher zufrieden Sehr zufrieden Eher zufrieden Eher z
##  [5] Eher zufrieden Sehr zufrieden Eher zufrieden Eher z
##  [9] Sehr zufrieden Sehr zufrieden
## 7 Levels: Item nonresponse Sehr zufrieden ... Weiß nicht
```

Auf Spalten zugreifen

- Wie bereits beschrieben kann man auch Zahlen nutzen um auf die Spalten zuzugreifen

```
head(dat[,1])
head(dat,"a11c019a"]) # liefert das gleiche Ergebnis
```

Exkurs - Labels wie verwenden

Tools for Working with Categorical Variables (Factors)

```
library("forcats")
```

- `fct_collapse` - um Faktorlevel zusammenzufassen
- `fct_count` - um die Einträge in einem Faktor zu zählen
- `fct_drop` - Unbenutzte Levels raus nehmen

Rekodieren

```
library(car)
```

```
##
## Attaching package: 'car'

## The following object is masked from 'package:dplyr':
##
##     recode

## The following object is masked from 'package:purrr':
##
##     some
```

```
head(dat$a11c020a)
```

```
## [1] Weder zufrieden noch unzufrieden Eher unzufrieden
## [3] Sehr zufrieden                   Sehr zufrieden
## [5] Eher zufrieden                   Eher zufrieden
```

Das Paket tibble

```
install.packages("tibble")
```

```
library(tibble)
gpanel1 <- as_tibble(dat)
gpanel1
```

```
## # A tibble: 100 × 23
##          a11c019a                                a11c020a
## *          <fctr>                                  <fctr>
## 1  Eher zufrieden Weder zufrieden noch unzufrieden   Seh
## 2  Sehr zufrieden                 Eher unzufrieden Eher
## 3  Eher zufrieden                   Sehr zufrieden Eher
## 4  Eher zufrieden                   Sehr zufrieden   Ehe
## 5  Eher zufrieden                   Eher zufrieden   Ehe
## 6  Sehr zufrieden                   Eher zufrieden   Ehe
## 7  Eher zufrieden                   Eher zufrieden   Ehe
## 8  Eher zufrieden                   Eher zufrieden   Ehe
## 9  Sehr zufrieden                   Eher zufrieden   Seh
```

Schleifen

```
erg <- vector()

for (i in 1:ncol(dat)){
  erg[i] <- length(table(dat[,i]))
}
```

Fehlende Werte ausschließen

- Mathe-Funktionen haben in der Regel einen Weg, um fehlende Werte in ihren Berechnungen auszuschließen.
- `mean()`, `median()`, `colSums()`, `var()`, `sd()`, `min()` und ‘max() all take the na.rm argument.

Fehlende Werte umkodieren

```
Daten$bazq020a[Daten$bazq020a==-99] <- NA
```

- Quick-R zu fehlenden Werten
- Fehlende Werte rekodieren

Mit Strings arbeiten

```
gsub("l","L","Hallo Welt")
```

```
## [1] "HaLLo WeLt"
```

- Natural Language Processing - Tutorial auf der UseR 2017

Weitere Links

- Tidy data - das Paket `tidyr`
- Die `tidyverse` Sammlung
- Data wrangling with R and RStudio