Programmieren mit R für Einsteiger

3. Tabellen / 3.5 Fehldaten

Berry Boessenkool 2022-02-25 11:41

Fehlende Werte (Missing values, NA=not available)

```
df <- data.frame(x=11:20, y=21:30)
df[3,2] <- NA
```

```
df
##      x  y
## 1   11 21
## 2   12 22
## 3   13 NA
## 4   14 24
## 5   15 25
## 6   16 26
## 7   17 27
## 8   18 28
## 9   19 29
## 10 20 30
```

```
is.na(df)
##              x     y
##  [1,] FALSE FALSE
##  [2,] FALSE FALSE
##  [3,] FALSE  TRUE
##  [4,] FALSE FALSE
##  [5,] FALSE FALSE
##  [6,] FALSE FALSE
##  [7,] FALSE FALSE
##  [8,] FALSE FALSE
##  [9,] FALSE FALSE
## [10,] FALSE FALSE
```

```
na.omit(df)
##      x  y
## 1   11 21
## 2   12 22
## 4   14 24
## 5   15 25
## 6   16 26
## 7   17 27
## 8   18 28
## 9   19 29
## 10 20 30
```

na.rm-Argument

```
df
##       x  y
## 1    11 21
## 2    12 22
## 3    13 NA
## 4    14 24
## 5    15 25
## 6    16 26
## 7    17 27
## 8    18 28
## 9    19 29
## 10   20 30
```

```
mean(df$x)
## [1] 15.5
```

```
mean(df$y)
## [1] NA
```

Mittelwert der nicht-NA Einträge:

```
mean(df$y, na.rm=TRUE)
## [1] 25.77778
```

Nah dran am Original:

```
mean(21:30)
## [1] 25.5
```

Für Summe gefährlich (wächst mit Anzahl):

```
sum(df$y, na.rm=TRUE)
## [1] 232
```

```
sum(21:30) # na.rm unterschätzt Summe !!
## [1] 255
```

NA-Imputation: fehlende Werte mit Schätzungen füllen

Mit Mittelwert / Median / Min / Max / ... der anderen Werte füllen:

```
df$y[is.na(df$y)] <- mean(df$y, na.rm=TRUE)
```

```
df$y[is.na(df$y)] <- median(df$y, na.rm=TRUE)
```

Letzte Beobachtung fortsetzen (locf: last observation carried forwards):

```
df[3,2] <- NA
zoo::na.locf(df$y)
##  [1] 21 22 22 24 25 26 27 28 29 30
```

Linear interpolieren:

```
approx(df$y, n=length(df$y))$y
##  [1] 21 22 23 24 25 26 27 28 29 30
```

```
zoo::na.approx(df$y) # weniger Tippen :)
##  [1] 21 22 23 24 25 26 27 28 29 30
```

Komplexe (multivariate) Modellierung -> Statistikkurse

Zusammenfassung

Fehldaten managen:

- `NA`, `is.na`
- `na.omit`
- `mean` / `median` / `sum` /...(na.rm=TRUE)

NA-imputation:

- `x[is.na(x)] <- median(x, na.rm=TRUE)`
- `x <- zoo::na.locf(x)`
- `x <- zoo::na.approx(x)`