Programmieren mit R für Einsteiger

2. Datentypen / 2.4 Kategorien

Berry Boessenkool 2022-02-25 11:40

Factors = kategoriale Variablen

```
factor(c("Boot","Auto","Zug","Auto","Boot"))
## [1] Boot Auto Zug  Auto Boot
## Levels: Auto Boot Zug
```

Standardmäßig alphabetisch sortiert. Für ordinalskalierte Kategorien:

```
factor(c("Boot","Auto","Zug","Auto","Boot"),
       levels=c("Boot", "Zug", "Auto"))
## [1] Boot Auto Zug  Auto Boot
## Levels: Boot Zug Auto
```

```
state.region[37] # eingebauter Datensatz mit factors
## [1] West
## Levels: Northeast South North Central West
```

```
class(state.region)
## [1] "factor"
```

```
levels(state.region)
## [1] "Northeast"     "South"         "North Central"
## [4] "West"
```

Häufigkeitstabellen

```
table(state.region) # Anzahl Vorkommen pro Wert
## state.region
##     Northeast         South North Central          West
##             9            16            12            13
```

```
noten <- c(3,5,2,3,1,2,2,3,5,2,2,2,1,3,4,4,2,4,3,6,3,1)
table(noten, dnn=NULL) # dnn: Dimensionsnamen weglassen
## 1 2 3 4 5 6
## 3 7 6 3 2 1
```

```
names(table(noten))
## [1] "1" "2" "3" "4" "5" "6"
```

```
geschlecht <- c("m","m","m","w","d","w","w","m","m","d","w",
                "d","w","m","w","w","w","w","w","d","m","w")
table(geschlecht, noten) # Kreuztabelle (contingency table)
##           noten
## geschlecht 1 2 3 4 5 6
##          d 1 2 0 0 0 1
##          m 0 1 4 0 2 0
##          w 2 4 2 3 0 0
```

tagged (grouped) **apply**: eine Funktion gruppenweise anwenden

```
head(state.name) # Zeichenkette
## [1] "Alabama"    "Alaska"     "Arizona"    "Arkansas"
## [5] "California" "Colorado"
```

```
head(state.region) # Kategorie (factor)
## [1] South West  West  South West  West
## Levels: Northeast South North Central West
```

```
nchar(state.name[1:6])
## [1]  7  6  7  8 10  8
```

```
tapply(X=state.name, INDEX=state.region, FUN=nchar)
## $Northeast
## [1] 11  5 13 13 10  8 12 12  7
## $South
##  [1]  7  8  8  7  7  8  9  8 11 14  8 14  9  5  8 13
## $`North Central`
##  [1]  8  7  4  6  8  9  8  8 12  4 12  9
## $West
##  [1]  6  7 10  8  6  5  7  6 10  6  4 10  7
```

tagged (grouped) **apply**: aggregieren

```
mean_charlen <- function(x) mean(nchar(x))
```

```
mean_charlen(state.name)
## [1] 8.44
```

```
tapply(X=state.name, INDEX=state.region, FUN=mean_charlen)
##     Northeast         South North Central          West
##     10.111111      9.000000      7.916667      7.076923
```

Wenn die Funktion immer genau einen einzigen Wert ausgibt, simplifiziert `tapply` das Ergebnis.

Hier wird die Dimension auf einen Vektor* mit 4 Werten reduziert.
*: technisch gesehen ein Array

Für kleine, temporäre Sachen muss die Funktion nicht separat erstellt werden, sondern kann namenslos verwendet werden (anonymous function):

```
tapply(X=state.name, INDEX=state.region,
       FUN=function(x) mean(nchar(x))   )
```

Internals und Lesetipp

Intern sind Kategorien als Zahlen hinterlegt:

```
as.numeric(state.region)[c(37, 7, 27)]
## [1] 4 1 3
```

Falls Zahlen als Factors eingelesen werden, würde `as.numeric(x)` die Levels geben, erst `as.numeric(as.character(x))` die eigentlichen Zahlen:

```
as.numeric(as.factor(19:17))
## [1] 3 2 1

# NICHT 19,18,17, sondern deren Levels
```

Mehr Details zu factors in Advanced R.

Zusammenfassung

Kategoriale Variablen in R:

- factor, `levels`
- `table` für Häufigkeitstabelle
- `tapply(Werte, Kategorien, Funktion)`
- Praktisch um Daten zu gruppieren, zB farblich in Grafiken
- intern als Integers hinterlegt, vorsicht mit `as.numeric`