

01. Einfuehrung in R

Aufgabe 1 | Festival-Datei herunterladen

- Ladet von folgender Webseite die Datei `festival.txt` herunter: https://github.com/ch-bu/tutorat_statistik/blob/master/01_Einfuehrung/Tutorat/festival.txt.
- Speichert die Datei auf einen Ordner eures PCs.

Bei der Datei handelt es sich um Daten von Festivalbesuchern. Die Datei hat 5 Variablen. * `thicknumb`: ID der Person * `gender`: Geschlecht der Person ("Male", "Female") * `day1`: Sauberkeit am ersten Tag des Festivals (0 = "Du stinkst ganz fuerchterlich" bis 4 = "Du riechst so angenehm wie Sonnencreme") * `day2`: Sauberkeit am zweiten Tag des Festivals * `day3`: Sauberkeit am dritten Tag des Festivals

Aufgabe 2 | Working-Directory bestimmen

Bestimmt zunaechst das Working-Directory auf den Ordner, in dem die `festival.txt` Datei liegt. Benutzt hierfuer die Funktion `setwd()`. Hierfuer drueckt ihr `Strg + Umschalt + K` auf eurem Keyboard.

In eurer Console sollte nun ein Befehl ausgefuehrt sein, der in etwa so aussieht:

```
setwd("C:/Users/Christian/Dropbox/Christian_HiWi/_Statistik I_Tutorat/01_Einfuehrung/Tutorat")
```

Aufgabe 3 | Festival-Datei in R einlesen

Als naechstes laden wir die Datei in R. Versichert euch zunaechst mit folgendem Befehl, dass ihr euer Arbeitsverzeichnis richtig bestimmt habt:

```
getwd()
```

```
## [1] "C:/Users/Christian/Dropbox/Christian_HiWi/_Statistik I_Tutorat/01_Einfuehrung/Tutorat"
```

Jetzt koennen wir die Datei laden. Hierfuer benutzen wir die `read.table()` Funktion.

```
festival <- read.table("festival.txt", header = TRUE, sep = "\t")
```

- `sep = "\t"` bedeutet, dass die Werte der Datei durch einen Tabstop getrennt sind.
- `festival` ist ein beliebiger Name. Wichtig ist nur, dass er nicht mit einer Zahl beginnt.

Aufgabe 4 | Den Datensatz kennenlernen

Nachdem der Datensatz geladen ist, koennen wir ihn betrachten. Zunaechst moechten wir wissen, welche Variablen im Dataframe sind.

```
names(festival)
```

```
## [1] "ticknumb" "gender" "day1" "day2" "day3"
```

Eine Variable kann in R durch ein `$`-Zeichen aufgerufen werden.

```
festival$day1
```

```
## [1] 2.64 0.97 0.84 3.03 0.88 0.85 1.56 3.02 2.29 1.11 2.17 0.82 1.41 1.76
## [15] 1.38 2.79 1.50 1.91 2.32 2.05 2.17 2.05 1.61 1.66 2.30 2.76 1.44 1.06
## [29] 3.23 0.97 2.57 0.26 0.47 1.73 1.94 1.91 2.08 1.91 1.42 1.50 0.11 1.67
## [43] 2.08 2.05 2.00 1.52 1.58 1.28 1.88 1.32 2.09 2.00 2.64 0.85 2.47 1.79
## [57] 1.64 1.32 2.97 1.44 2.02 1.79 1.34 2.29 1.66 0.60 1.76 1.50 2.08 1.00
## [71] 1.73 1.05 2.81 1.52 1.47 2.64 2.20 0.55 2.29 2.00 2.23 2.45 1.20 2.91
## [85] 1.14 1.88 0.94 1.85 2.58 0.61 0.70 1.38 1.94 2.29 1.59 2.46 1.67 2.02
## [99] 1.50 2.70 1.61 2.29 0.97 1.85 2.76 1.64 1.17 1.57 2.23 2.05 2.05 2.94
## [113] 2.39 1.94 2.12 1.11 0.97 1.35 2.81 2.50 1.87 1.33 1.26 1.44 0.55 1.75
## [127] 2.08 0.85 2.52 3.00 1.41 1.08 1.20 1.94 2.26 1.41 2.50 2.17 1.82 1.44
## [141] 1.66 1.82 1.26 2.67 1.47 1.84 2.58 1.73 1.23 2.32 2.67 1.02 1.66 1.88
## [155] 1.91 1.64 1.34 1.85 2.08 1.02 1.79 1.94 3.26 1.14 1.50 2.03 2.24 1.11
## [169] 2.21 1.94 2.41 0.88 1.17 2.23 1.64 2.14 0.11 2.17 1.67 1.00 0.88 2.20
## [183] 2.17 2.32 1.64 3.00 2.38 1.60 1.58 2.61 1.44 1.57 2.32 1.14 1.93 2.47
## [197] 2.29 1.00 1.58 2.44 0.83 2.71 1.73 1.58 1.50 1.05 2.05 2.63 2.55 2.00
## [211] 2.00 1.32 3.14 1.44 1.85 1.41 1.94 2.91 1.85 1.70 2.23 1.11 1.47 2.20
## [225] 1.82 1.42 2.44 2.66 1.52 1.35 1.29 2.32 0.78 2.84 0.97 1.52 1.70 0.94
## [239] 1.41 1.79 1.08 1.47 1.79 2.00 0.76 2.20 0.94 1.38 1.38 0.32 2.58 0.51
## [253] 0.32 0.91 1.51 1.47 2.50 2.26 2.81 1.87 2.00 2.23 2.00 1.41 1.64 1.64
## [267] 1.26 1.52 2.44 2.18 3.02 1.02 2.88 1.54 1.64 2.44 1.29 1.61 1.77 0.91
## [281] 0.85 0.85 1.50 1.05 3.38 1.42 1.85 1.91 0.82 1.32 2.23 1.47 2.70 1.58
## [295] 1.00 1.44 2.00 1.60 2.32 3.41 2.02 0.64 3.58 1.50 1.08 1.52 1.26 1.68
## [309] 1.47 1.47 1.67 2.47 1.82 2.17 3.21 1.60 0.32 0.55 1.42 1.14 2.64 2.58
## [323] 2.02 2.00 2.90 1.82 0.50 1.53 2.48 2.05 2.52 1.88 2.73 2.88 1.67 1.93
## [337] 1.67 1.20 2.75 1.94 0.59 1.50 1.58 2.23 2.35 2.55 1.55 2.31 2.23 0.67
## [351] 2.51 1.08 2.44 0.23 2.17 1.90 1.67 2.00 2.44 1.44 0.82 2.50 1.82 1.97
## [365] 2.52 0.05 2.08 2.39 1.45 2.58 2.12 2.02 1.78 0.73 2.26 2.79 0.43 0.52
## [379] 2.32 2.22 0.58 2.00 0.70 1.00 0.30 1.52 1.58 2.34 0.79 2.26 2.35 1.70
## [393] 3.09 1.52 0.35 2.70 1.64 0.82 2.73 2.23 1.06 2.05 1.73 0.93 2.50 1.44
## [407] 2.88 0.67 1.85 1.21 1.06 0.61 2.00 1.17 1.48 1.55 3.29 1.47 0.96 1.00
## [421] 1.47 2.55 0.44 2.35 1.71 1.84 1.11 1.38 0.88 0.94 1.91 2.76 1.55 2.67
## [435] 1.03 2.50 1.64 2.26 2.14 0.52 1.08 1.69 2.73 1.91 1.73 3.21 2.11 2.05
## [449] 2.17 2.17 2.30 2.56 2.11 1.70 1.23 3.20 2.02 2.64 2.52 1.61 1.50 1.15
## [463] 1.82 1.50 2.32 2.92 1.41 1.35 0.61 0.73 2.23 1.32 2.94 1.61 1.00 3.15
## [477] 2.88 2.09 1.32 1.47 1.61 2.20 2.78 2.06 0.47 2.87 1.14 3.32 2.08 2.38
## [491] 2.08 1.85 1.38 1.14 1.58 1.23 2.53 0.67 0.73 1.34 2.14 1.00 1.35 1.94
## [505] 0.50 3.08 2.88 1.91 1.41 2.02 0.76 1.94 0.67 2.41 2.17 2.67 1.94 2.05
## [519] 2.17 0.47 0.62 2.00 0.45 2.29 2.55 0.82 3.12 2.50 1.79 2.28 0.58 2.50
## [533] 1.41 2.14 0.76 1.79 1.02 2.62 0.88 1.58 2.20 1.14 1.47 1.41 1.44 1.23
## [547] 1.82 2.44 1.94 2.41 2.27 1.79 1.88 1.85 2.21 1.97 2.51 2.05 1.29 2.05
## [561] 2.23 1.76 1.05 1.79 1.02 2.76 1.67 2.85 0.23 1.90 1.23 1.97 1.50 3.69
## [575] 0.50 2.18 2.17 1.58 2.88 2.52 2.20 1.73 2.23 1.97 1.20 2.00 1.91 0.81
## [589] 1.31 0.38 1.97 0.38 2.11 3.20 0.02 2.56 2.02 2.30 2.02 2.05 1.70 1.61
## [603] 0.73 2.50 2.18 2.46 1.50 1.73 1.44 1.64 2.02 1.20 0.38 1.58 1.67 1.00
```

```
## [617] 2.58 2.82 2.29 1.14 1.64 1.82 3.32 3.32 1.85 2.29 1.47 2.08 2.20 1.06
## [631] 0.97 2.00 1.67 2.94 1.55 0.88 1.35 0.61 1.00 1.52 1.00 1.76 2.52 2.00
## [645] 2.63 0.73 1.58 0.58 1.67 1.47 1.81 1.91 1.06 1.47 2.52 1.85 3.44 1.55
## [659] 2.29 1.76 1.90 2.52 2.52 2.82 2.02 1.29 1.26 0.94 2.00 0.73 2.26 2.23
## [673] 2.35 0.55 1.85 0.67 1.85 1.23 2.35 1.35 1.94 1.55 1.29 2.17 1.91 2.88
## [687] 2.36 2.36 2.20 2.17 0.52 0.32 1.52 2.00 1.32 2.05 1.73 1.94 1.81 0.90
## [701] 1.58 2.29 2.57 1.58 2.33 3.15 2.29 0.82 1.93 1.82 1.96 1.32 1.02 1.14
## [715] 2.32 2.16 2.42 1.14 1.55 1.17 1.00 1.05 1.38 1.93 2.73 2.02 2.81 2.47
## [729] 1.35 2.08 2.50 2.45 2.17 1.70 0.70 1.51 1.23 2.14 1.14 0.96 1.52 0.52
## [743] 1.56 3.29 0.45 2.63 1.70 3.11 1.82 1.58 2.73 1.50 1.78 2.02 0.67 1.41
## [757] 0.90 1.23 2.70 1.97 0.84 1.79 2.84 2.02 1.64 1.08 2.97 0.94 2.97 0.97
## [771] 1.47 2.61 1.73 3.38 3.17 2.20 2.14 1.29 3.21 2.67 1.85 1.35 2.14 1.24
## [785] 2.02 2.32 1.08 1.14 2.14 2.88 1.35 1.00 2.02 0.64 0.29 1.73 1.82 2.11
## [799] 1.23 0.64 2.23 2.44 1.17 0.61 0.52 2.91 2.61 1.47 1.28 1.26
```

Ihr seht alle 810 Werte dieser Variable. Jeder Wert ist die Sauberkeit einer Person am ersten Tag des Festivals.
Den kompletten Datensatz koennen wir mit folgender Funktion ansehen:

```
View(festival)
```

Damit wir nicht immer `festival$` vor einer Variable schreiben muessen, benutzen wir die `attach()` Funktion. Nachdem wir sie benutzt haben, genuegt es, `day1` fuer die Spalte des Dataframes zu schreiben.

```
attach(festival)
day1
```

```
## [1] 2.64 0.97 0.84 3.03 0.88 0.85 1.56 3.02 2.29 1.11 2.17 0.82 1.41 1.76
## [15] 1.38 2.79 1.50 1.91 2.32 2.05 2.17 2.05 1.61 1.66 2.30 2.76 1.44 1.06
## [29] 3.23 0.97 2.57 0.26 0.47 1.73 1.94 1.91 2.08 1.91 1.42 1.50 0.11 1.67
## [43] 2.08 2.05 2.00 1.52 1.58 1.28 1.88 1.32 2.09 2.00 2.64 0.85 2.47 1.79
## [57] 1.64 1.32 2.97 1.44 2.02 1.79 1.34 2.29 1.66 0.60 1.76 1.50 2.08 1.00
## [71] 1.73 1.05 2.81 1.52 1.47 2.64 2.20 0.55 2.29 2.00 2.23 2.45 1.20 2.91
## [85] 1.14 1.88 0.94 1.85 2.58 0.61 0.70 1.38 1.94 2.29 1.59 2.46 1.67 2.02
## [99] 1.50 2.70 1.61 2.29 0.97 1.85 2.76 1.64 1.17 1.57 2.23 2.05 2.05 2.94
## [113] 2.39 1.94 2.12 1.11 0.97 1.35 2.81 2.50 1.87 1.33 1.26 1.44 0.55 1.75
## [127] 2.08 0.85 2.52 3.00 1.41 1.08 1.20 1.94 2.26 1.41 2.50 2.17 1.82 1.44
## [141] 1.66 1.82 1.26 2.67 1.47 1.84 2.58 1.73 1.23 2.32 2.67 1.02 1.66 1.88
## [155] 1.91 1.64 1.34 1.85 2.08 1.02 1.79 1.94 3.26 1.14 1.50 2.03 2.24 1.11
## [169] 2.21 1.94 2.41 0.88 1.17 2.23 1.64 2.14 0.11 2.17 1.67 1.00 0.88 2.20
## [183] 2.17 2.32 1.64 3.00 2.38 1.60 1.58 2.61 1.44 1.57 2.32 1.14 1.93 2.47
## [197] 2.29 1.00 1.58 2.44 0.83 2.71 1.73 1.58 1.50 1.05 2.05 2.63 2.55 2.00
## [211] 2.00 1.32 3.14 1.44 1.85 1.41 1.94 2.91 1.85 1.70 2.23 1.11 1.47 2.20
## [225] 1.82 1.42 2.44 2.66 1.52 1.35 1.29 2.32 0.78 2.84 0.97 1.52 1.70 0.94
## [239] 1.41 1.79 1.08 1.47 1.79 2.00 0.76 2.20 0.94 1.38 1.38 0.32 2.58 0.51
## [253] 0.32 0.91 1.51 1.47 2.50 2.26 2.81 1.87 2.00 2.23 2.00 1.41 1.64 1.64
## [267] 1.26 1.52 2.44 2.18 3.02 1.02 2.88 1.54 1.64 2.44 1.29 1.61 1.77 0.91
## [281] 0.85 0.85 1.50 1.05 3.38 1.42 1.85 1.91 0.82 1.32 2.23 1.47 2.70 1.58
## [295] 1.00 1.44 2.00 1.60 2.32 3.41 2.02 0.64 3.58 1.50 1.08 1.52 1.26 1.68
## [309] 1.47 1.47 1.67 2.47 1.82 2.17 3.21 1.60 0.32 0.55 1.42 1.14 2.64 2.58
## [323] 2.02 2.00 2.90 1.82 0.50 1.53 2.48 2.05 2.52 1.88 2.73 2.88 1.67 1.93
## [337] 1.67 1.20 2.75 1.94 0.59 1.50 1.58 2.23 2.35 2.55 1.55 2.31 2.23 0.67
## [351] 2.51 1.08 2.44 0.23 2.17 1.90 1.67 2.00 2.44 1.44 0.82 2.50 1.82 1.97
```

```
## [365] 2.52 0.05 2.08 2.39 1.45 2.58 2.12 2.02 1.78 0.73 2.26 2.79 0.43 0.52
## [379] 2.32 2.22 0.58 2.00 0.70 1.00 0.30 1.52 1.58 2.34 0.79 2.26 2.35 1.70
## [393] 3.09 1.52 0.35 2.70 1.64 0.82 2.73 2.23 1.06 2.05 1.73 0.93 2.50 1.44
## [407] 2.88 0.67 1.85 1.21 1.06 0.61 2.00 1.17 1.48 1.55 3.29 1.47 0.96 1.00
## [421] 1.47 2.55 0.44 2.35 1.71 1.84 1.11 1.38 0.88 0.94 1.91 2.76 1.55 2.67
## [435] 1.03 2.50 1.64 2.26 2.14 0.52 1.08 1.69 2.73 1.91 1.73 3.21 2.11 2.05
## [449] 2.17 2.17 2.30 2.56 2.11 1.70 1.23 3.20 2.02 2.64 2.52 1.61 1.50 1.15
## [463] 1.82 1.50 2.32 2.92 1.41 1.35 0.61 0.73 2.23 1.32 2.94 1.61 1.00 3.15
## [477] 2.88 2.09 1.32 1.47 1.61 2.20 2.78 2.06 0.47 2.87 1.14 3.32 2.08 2.38
## [491] 2.08 1.85 1.38 1.14 1.58 1.23 2.53 0.67 0.73 1.34 2.14 1.00 1.35 1.94
## [505] 0.50 3.08 2.88 1.91 1.41 2.02 0.76 1.94 0.67 2.41 2.17 2.67 1.94 2.05
## [519] 2.17 0.47 0.62 2.00 0.45 2.29 2.55 0.82 3.12 2.50 1.79 2.28 0.58 2.50
## [533] 1.41 2.14 0.76 1.79 1.02 2.62 0.88 1.58 2.20 1.14 1.47 1.41 1.44 1.23
## [547] 1.82 2.44 1.94 2.41 2.27 1.79 1.88 1.85 2.21 1.97 2.51 2.05 1.29 2.05
## [561] 2.23 1.76 1.05 1.79 1.02 2.76 1.67 2.85 0.23 1.90 1.23 1.97 1.50 3.69
## [575] 0.50 2.18 2.17 1.58 2.88 2.52 2.20 1.73 2.23 1.97 1.20 2.00 1.91 0.81
## [589] 1.31 0.38 1.97 0.38 2.11 3.20 0.02 2.56 2.02 2.30 2.02 2.05 1.70 1.61
## [603] 0.73 2.50 2.18 2.46 1.50 1.73 1.44 1.64 2.02 1.20 0.38 1.58 1.67 1.00
## [617] 2.58 2.82 2.29 1.14 1.64 1.82 3.32 3.32 1.85 2.29 1.47 2.08 2.20 1.06
## [631] 0.97 2.00 1.67 2.94 1.55 0.88 1.35 0.61 1.00 1.52 1.00 1.76 2.52 2.00
## [645] 2.63 0.73 1.58 0.58 1.67 1.47 1.81 1.91 1.06 1.47 2.52 1.85 3.44 1.55
## [659] 2.29 1.76 1.90 2.52 2.52 2.82 2.02 1.29 1.26 0.94 2.00 0.73 2.26 2.23
## [673] 2.35 0.55 1.85 0.67 1.85 1.23 2.35 1.35 1.94 1.55 1.29 2.17 1.91 2.88
## [687] 2.36 2.36 2.20 2.17 0.52 0.32 1.52 2.00 1.32 2.05 1.73 1.94 1.81 0.90
## [701] 1.58 2.29 2.57 1.58 2.33 3.15 2.29 0.82 1.93 1.82 1.96 1.32 1.02 1.14
## [715] 2.32 2.16 2.42 1.14 1.55 1.17 1.00 1.05 1.38 1.93 2.73 2.02 2.81 2.47
## [729] 1.35 2.08 2.50 2.45 2.17 1.70 0.70 1.51 1.23 2.14 1.14 0.96 1.52 0.52
## [743] 1.56 3.29 0.45 2.63 1.70 3.11 1.82 1.58 2.73 1.50 1.78 2.02 0.67 1.41
## [757] 0.90 1.23 2.70 1.97 0.84 1.79 2.84 2.02 1.64 1.08 2.97 0.94 2.97 0.97
## [771] 1.47 2.61 1.73 3.38 3.17 2.20 2.14 1.29 3.21 2.67 1.85 1.35 2.14 1.24
## [785] 2.02 2.32 1.08 1.14 2.14 2.88 1.35 1.00 2.02 0.64 0.29 1.73 1.82 2.11
## [799] 1.23 0.64 2.23 2.44 1.17 0.61 0.52 2.91 2.61 1.47 1.28 1.26
```

Aufgabe 4 | Deskriptive Statistik

Wie sauber ist die sauberste Person am ersten Tag des Festivals?

```
max(day1)
```

```
## [1] 3.69
```

Auf einer Skale von 0 bis 4 hat die sauberste Person einen Wert von 3.69.

Wie dreckig ist die dreckigste Person am ersten Tag des Festivals?

Erinnert euch an die Funktion, die statt dem maximalen den minimalen Wert einer Variable findet.

```
min(day1)
```

```
## [1] 0.02
```

Wie sauber ist die sauberste Person am zweiten Tag des Festivals?

```
max(day2)
```

```
## [1] NA
```

Wenn wir diese Zeile ausführen, bekommen wir einen NA. Allerdings konnten wir unter `View()` sehen, dass es in der Variable Werte gibt. Das Ergebnis `[1] NA` kommt zu Stande, weil wir der Funktion nicht gesagt haben, dass es fehlende Werte in der Variablen gibt. Dies koennen wir auch durch eine Funktion herausfinden:

```
is.na(day2)
```

```
## [1] FALSE FALSE TRUE TRUE FALSE TRUE TRUE TRUE TRUE TRUE FALSE TRUE
## [12] FALSE TRUE FALSE FALSE TRUE TRUE FALSE TRUE TRUE FALSE TRUE
## [23] TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE FALSE FALSE TRUE TRUE
## [34] TRUE FALSE TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE FALSE FALSE
## [45] TRUE TRUE TRUE FALSE TRUE TRUE TRUE TRUE TRUE FALSE TRUE TRUE
## [56] TRUE TRUE TRUE TRUE FALSE TRUE TRUE TRUE TRUE FALSE TRUE TRUE
## [67] FALSE TRUE FALSE FALSE TRUE FALSE TRUE TRUE TRUE TRUE FALSE
## [78] FALSE TRUE TRUE TRUE TRUE TRUE TRUE FALSE FALSE TRUE TRUE TRUE
## [89] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE FALSE FALSE TRUE
## [100] FALSE TRUE TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE FALSE
## [111] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE FALSE TRUE TRUE
## [122] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [133] FALSE FALSE FALSE TRUE TRUE TRUE FALSE FALSE TRUE TRUE TRUE
## [144] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE FALSE TRUE
## [155] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE FALSE FALSE FALSE
## [166] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE FALSE TRUE TRUE
## [177] FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE FALSE
## [188] FALSE TRUE TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [199] TRUE TRUE TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE FALSE
## [210] TRUE TRUE TRUE TRUE TRUE FALSE FALSE TRUE TRUE TRUE TRUE
## [221] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE FALSE TRUE
## [232] TRUE TRUE TRUE TRUE TRUE TRUE FALSE TRUE TRUE TRUE FALSE TRUE
## [243] FALSE TRUE TRUE TRUE FALSE TRUE FALSE TRUE TRUE TRUE TRUE TRUE
## [254] FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE FALSE FALSE TRUE
## [265] TRUE TRUE TRUE FALSE FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [276] FALSE TRUE FALSE TRUE FALSE FALSE FALSE TRUE FALSE TRUE FALSE
## [287] TRUE FALSE FALSE FALSE FALSE FALSE TRUE TRUE TRUE TRUE TRUE
## [298] TRUE FALSE TRUE TRUE FALSE FALSE TRUE TRUE TRUE TRUE TRUE TRUE
## [309] FALSE TRUE FALSE FALSE TRUE TRUE TRUE FALSE TRUE TRUE TRUE TRUE
## [320] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [331] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE FALSE TRUE
## [342] TRUE FALSE FALSE TRUE FALSE TRUE TRUE TRUE TRUE FALSE TRUE FALSE
## [353] TRUE FALSE TRUE FALSE FALSE FALSE TRUE TRUE TRUE TRUE TRUE TRUE
## [364] TRUE TRUE TRUE TRUE TRUE FALSE TRUE TRUE FALSE FALSE FALSE
## [375] FALSE TRUE FALSE FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [386] TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [397] TRUE TRUE TRUE TRUE TRUE FALSE FALSE FALSE FALSE TRUE TRUE
## [408] FALSE FALSE FALSE FALSE FALSE TRUE FALSE FALSE TRUE TRUE TRUE
## [419] TRUE FALSE TRUE FALSE FALSE FALSE FALSE FALSE FALSE TRUE TRUE
## [430] TRUE TRUE TRUE FALSE TRUE FALSE TRUE TRUE TRUE TRUE TRUE
```

```
## [441] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [452] TRUE FALSE TRUE TRUE TRUE TRUE TRUE FALSE TRUE FALSE FALSE
## [463] TRUE TRUE TRUE TRUE TRUE TRUE FALSE TRUE FALSE FALSE FALSE
## [474] TRUE TRUE FALSE TRUE FALSE FALSE FALSE TRUE TRUE TRUE TRUE
## [485] TRUE TRUE TRUE FALSE FALSE FALSE TRUE TRUE TRUE TRUE TRUE
## [496] FALSE TRUE TRUE TRUE TRUE FALSE TRUE TRUE FALSE TRUE TRUE
## [507] TRUE TRUE TRUE TRUE TRUE TRUE FALSE TRUE TRUE FALSE FALSE
## [518] FALSE TRUE TRUE FALSE TRUE TRUE FALSE TRUE TRUE FALSE FALSE
## [529] TRUE TRUE FALSE FALSE TRUE TRUE FALSE FALSE FALSE TRUE TRUE
## [540] FALSE TRUE TRUE FALSE FALSE FALSE FALSE TRUE TRUE FALSE TRUE
## [551] TRUE FALSE TRUE FALSE TRUE FALSE TRUE TRUE FALSE TRUE TRUE
## [562] TRUE TRUE TRUE FALSE FALSE FALSE TRUE FALSE TRUE FALSE TRUE
## [573] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [584] TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE FALSE
## [595] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE FALSE
## [606] FALSE FALSE FALSE TRUE FALSE FALSE FALSE FALSE FALSE FALSE
## [617] FALSE FALSE FALSE TRUE FALSE FALSE FALSE TRUE FALSE TRUE FALSE
## [628] FALSE TRUE FALSE FALSE TRUE FALSE TRUE TRUE FALSE FALSE FALSE
## [639] TRUE FALSE TRUE FALSE TRUE FALSE TRUE FALSE FALSE TRUE TRUE
## [650] TRUE TRUE TRUE TRUE FALSE TRUE FALSE TRUE FALSE FALSE TRUE
## [661] TRUE FALSE TRUE FALSE TRUE TRUE TRUE TRUE FALSE TRUE FALSE
## [672] TRUE TRUE FALSE TRUE TRUE FALSE FALSE TRUE TRUE TRUE TRUE
## [683] TRUE TRUE TRUE TRUE TRUE TRUE TRUE FALSE TRUE TRUE FALSE
## [694] FALSE FALSE TRUE TRUE TRUE TRUE FALSE FALSE FALSE TRUE TRUE
## [705] FALSE TRUE TRUE TRUE TRUE TRUE TRUE FALSE TRUE TRUE TRUE
## [716] TRUE FALSE TRUE FALSE TRUE FALSE FALSE TRUE FALSE TRUE TRUE
## [727] TRUE TRUE TRUE TRUE TRUE TRUE TRUE FALSE FALSE TRUE TRUE
## [738] FALSE FALSE TRUE FALSE TRUE TRUE FALSE TRUE TRUE TRUE TRUE
## [749] TRUE FALSE TRUE FALSE FALSE TRUE FALSE FALSE TRUE FALSE TRUE
## [760] TRUE TRUE TRUE TRUE TRUE FALSE TRUE TRUE TRUE FALSE TRUE
## [771] TRUE TRUE FALSE FALSE FALSE FALSE TRUE FALSE FALSE TRUE FALSE
## [782] TRUE FALSE FALSE FALSE TRUE FALSE FALSE FALSE TRUE FALSE FALSE
## [793] TRUE FALSE FALSE FALSE FALSE TRUE TRUE TRUE FALSE FALSE FALSE
## [804] FALSE TRUE FALSE FALSE TRUE TRUE TRUE
```

```
any(is.na(day2))
```

```
## [1] TRUE
```

Die erste Funktion zeigt uns fuer jeden Wert der Variable, ob es sich um einen NA handelt. Die zweite Funktion sagt uns, ob mindestens ein NA in der Variable steckt.

Nachdem wir wissen, dass es NAs in der Variable gibt, koennen wir unseren Code verbessern.

```
max(day2, na.rm = T)
```

```
## [1] 3.44
```

na.rm = T sagt der Funktion, dass NAs in der Variable sind. Wir sehen, dass bereits am zweiten Tag die sauberste Person nicht mehr so sauber ist, wie die sauberste Person am ersten Tag.

Wie dreckig ist die dreckigste Person am zweiten Tag des Festivals?

```
min(day2, na.rm = T)
```

```
## [1] 0
```

Finde Heraus, ob es fehlende Wert in der Variable day3 gibt, ohne die Variable zu betrachten

```
any(is.na(day3))
```

```
## [1] TRUE
```

Und wie dreckig ist die dreckigste Person am dritten Tag des Festivals?

```
min(day3, na.rm = T)
```

```
## [1] 0.02
```

Aufgabe 5 | Teile des Datensatzes betrachten

Mit `max()` und `min()` finden wir die hoechsten und niedrigsten Werte einer Variablen. Wir wissen aber nicht, welche Personen diese Werte haben. Hierfuer gibt es die `which.max()` und `which.min()` Funktionen.

```
dirty <- which.min(day2)
dirty
```

```
## [1] 734
```

Mit `which.in(day2)` findet die Zeile der Person, die am zweiten Tag am dreckigsten ist. Wir speichern diesen Wert in der Variable `dirty`.

Wir wissen aber noch nicht, wie wir die Person aufgrund der Zahl finden. Wir koennten in `View()` die Zeile suchen. Das wuerde auf Dauer aber zu lange dauern. Alternativ koennen wir auch nur Ausschnitte des Dataframes betrachten.

```
festival[1,]
```

```
##   ticknumb gender day1 day2 day3
## 1      2111   Male 2.64 1.35 1.61
```

```
festival[1,2]
```

```
## [1] Male
## Levels: Female Male
```

```
festival[1, c(2:5)]
```

```
##   gender day1 day2 day3
## 1   Male 2.64 1.35 1.61
```

```
festival[c(1:5), ]
```

```
##   ticknumb gender day1 day2 day3
## 1      2111   Male 2.64 1.35 1.61
## 2      2229 Female 0.97 1.41 0.29
## 3      2338   Male 0.84   NA   NA
## 4      2384 Female 3.03   NA   NA
## 5      2401 Female 0.88 0.08   NA
```

```
festival[c(1:3), c(1,3,5)]
```

```
##   ticknumb day1 day3
## 1      2111 2.64 1.61
## 2      2229 0.97 0.29
## 3      2338 0.84   NA
```

Mit der eckigen Klammer koennen wir Teile des Dataframes betrachten. Das Komma in der Mitte trennt Zeilen von Spalten. Links vom Komma stehen die Zeilen, rechts die Spalten. Mit `c()` bestimmen wir die Anzahl der Zeilen und/oder Spalten, die wir betrachten wollen.

Zeige die Zeilen 300 bis 305 mit allen Spalten

```
festival[c(300:305),]
```

```
##   ticknumb gender day1 day2 day3
## 300      3371 Female 3.41   NA   NA
## 301      3372 Female 2.02   NA   NA
## 302      3373   Male 0.64 0.52   NA
## 303      3374   Male 3.58 3.35   NA
## 304      3375   Male 1.50   NA   NA
## 305      3376   Male 1.08   NA   NA
```

Zeige die Zeile 466 mit den Spalten 3 bis 5

```
festival[466, c(3, 5)]
```

```
##   day1 day3
## 466 2.92   NA
```


Wer ist die dreckigste Person am zweiten Tag des Festivals?

Erinnert euch, dass wir die Zeile der Person in der Variable `dirty` gespeichert haben. Diese koennen wir nun statt einer Zahl verwenden.

```
festival[dirty,]
```

```
##      ticknumb gender day1 day2 day3  
## 734      4444  Male  1.7   0   NA
```