

Eurostat: Anträge aus Syrien in Deutschland, nach Alter und Geschlecht

refugee datathon munich

August, 2018

Beispiel für eine Eurostat-Analyse mit R

Asylanträge syrischer Staatsangehöriger in Deutschland

Zuerst werden Bibliotheken geladen, um Eurostat-Daten zu holen und ansprechende Visualisierungen zu erzeugen.

```
library(eurostat)
library(dplyr)
library(ggplot2)
library(scales)
```

Nun holen wir die Daten direkt von Eurostat. Dafür brauchen wir nur den Namen des Datensets.

```
#migr_asyappctza=get_eurostat("migr_asyappctza")
# oder, falls es bereits vorbereitend gesichert wurde:
migr_asyappctza = readRDS(file = "/tmp/migr_asyappctza_20180824.rds")
```

Ein erster Blick in die Daten:

```
glimpse(migr_asyappctza)
```

```
## Observations: 3,917,944
## Variables: 8
## $ citizen <fctr> AD, AD, AD, AD, AD, AD, AD, AD, AD, AD, AD, AD, AD, ...
## $ sex <fctr> F, F, F, F, F, F, F, F, F, F, F, F, F, F, F, F, F, F, F, F...
## $ unit <fctr> PER, PER, PER, PER, PER, PER, PER, PER, PER, PER, PER, PER, PE...
## $ age <fctr> TOTAL, TOTAL, TOTAL, TOTAL, TOTAL, TOTAL, TOTAL, TOTAL, TOT...
## $ asyl_app <fctr> ASY_APP, ASY_APP, ASY_APP, ASY_APP, ASY_APP, ASY_APP, ASY_APP...
## $ geo <fctr> AT, BE, BG, CH, CY, CZ, DE, DK, EE, EL, ES, EU28, FI...
## $ time <date> 2017-01-01, 2017-01-01, 2017-01-01, 2017-01-01, 2017...
## $ values <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
```

Wir reduzieren das auf:

1. Anträge syrischer Staatsangehöriger
2. Anträge in Deutschland
3. nur Anträge vom Typ "ASY_APP"
4. keine Totale bezüglich Geschlecht und Alter, und
5. keine sich überschneidenden Altersklassen

Die letzten drei Punkte beheben Mehrfachzählungen. Für die Übersichtlichkeit beschränken wir uns außerdem auf Spalten von Interesse.

```
de_sy_migr_asyappctza <- subset(migr_asyappctza,
                                citizen == "SY" &
                                asyl_app == "ASY_APP" &
                                geo == "DE" &
```

```
sex != "T" &
age != "Y_LT18" &
age != "TOTAL")[c(2,4,7:8)]
```

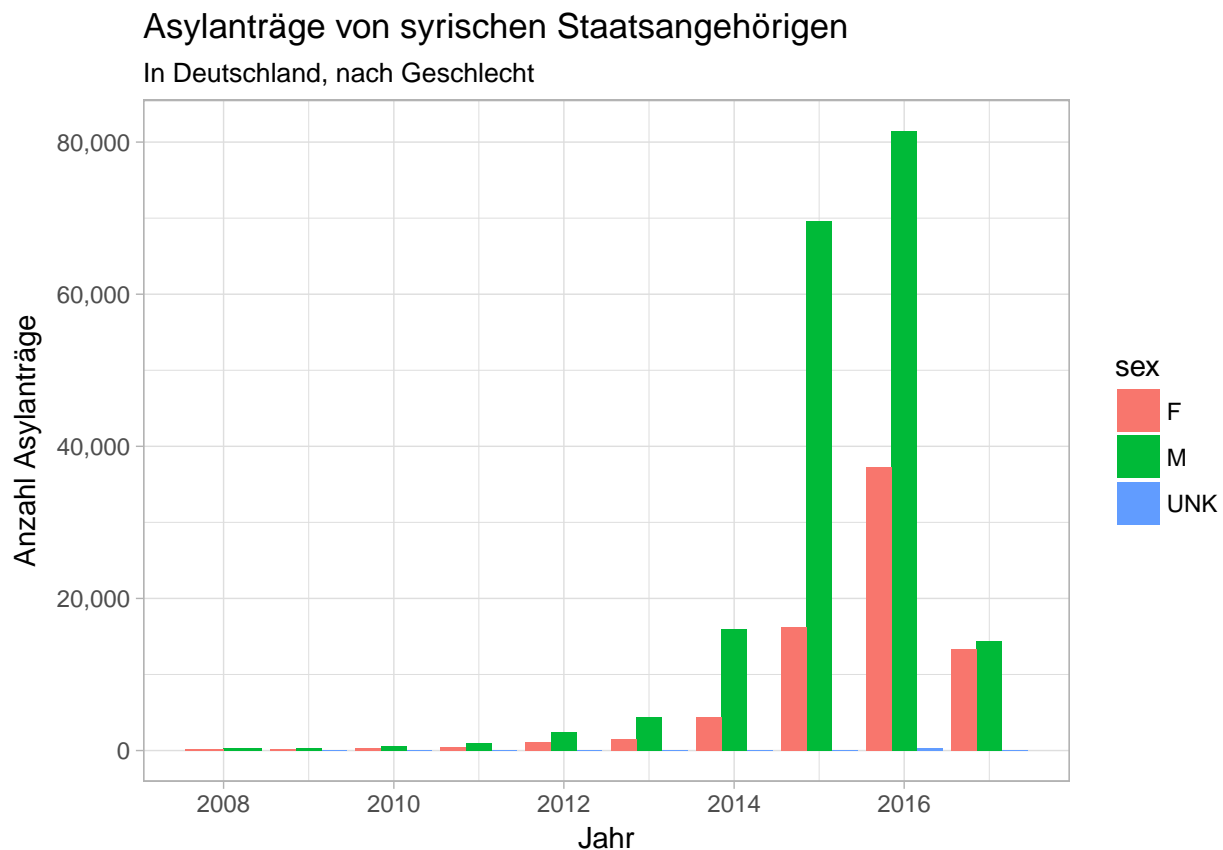
Jetzt sieht es schon besser aus:

```
glimpse(de_sy_migr_asyappctza)
```

```
## Observations: 174
## Variables: 4
## $ sex      <fctr> F, F, F, F, F, F, M, M, M, M, M, M, UNK, UNK, UNK, UNK...
## $ age      <fctr> UNK, Y14-17, Y18-34, Y35-64, Y_GE65, Y_LT14, UNK, Y14-...
## $ time     <date> 2017-01-01, 2017-01-01, 2017-01-01, 2017-01-01, 2017-0...
## $ values   <dbl> 0, 1380, 5765, 3890, 180, 13370, 0, 2010, 6005, 3260, 1...
```

Für die Visualisierung kommt das Paket `ggplot` zum Einsatz.

```
bar_sy_de_sex <- ggplot(de_sy_migr_asyappctza, aes(x=time, y=values, fill=sex)) +
  geom_bar(stat="identity", position="dodge") +
  scale_y_continuous(labels=comma) +
  scale_x_date() +
  ggtitle("Asylanträge von syrischen Staatsangehörigen",
    subtitle="In Deutschland, nach Geschlecht") +
  xlab("Jahr") +
  ylab("Anzahl Asylanträge")
bar_sy_de_sex + scale_fill_discrete() + theme_light()
```



Die Altersklassen können wir auch so darstellen. Allerdings müssen wir die Reihenfolge festlegen, damit sie aufsteigend erscheinen.

```

de_sy_migr_asyappctza$age <- factor(de_sy_migr_asyappctza$age,
                                   levels=c("Y_GE65", "Y35-64",
                                             "Y18-34", "Y14-17", "Y_LT14", "UNK"))
bar_sy_de_age <- ggplot(de_sy_migr_asyappctza, aes(x=time, y=values, fill=age)) +
  geom_bar(stat="identity") +
  scale_y_continuous(labels=comma) +
  scale_x_date() +
  ggtitle("Asylanträge von syrischen Staatsangehörigen",
          subtitle="In Deutschland, nach Alter") +
  xlab("Jahr") +
  ylab("Anzahl Asylanträge")
bar_sy_de_age + scale_fill_brewer() + theme_light()

```

