Datenanalyse mit R

10 Beschreibende Statistik: mehrere Variablen

Tobias Wiß, Carmen Walenta und Felix Wohlgemuth

08.05.2020

JKU Institut für Gesellschafts- und Sozialpolitik

Wiederholung

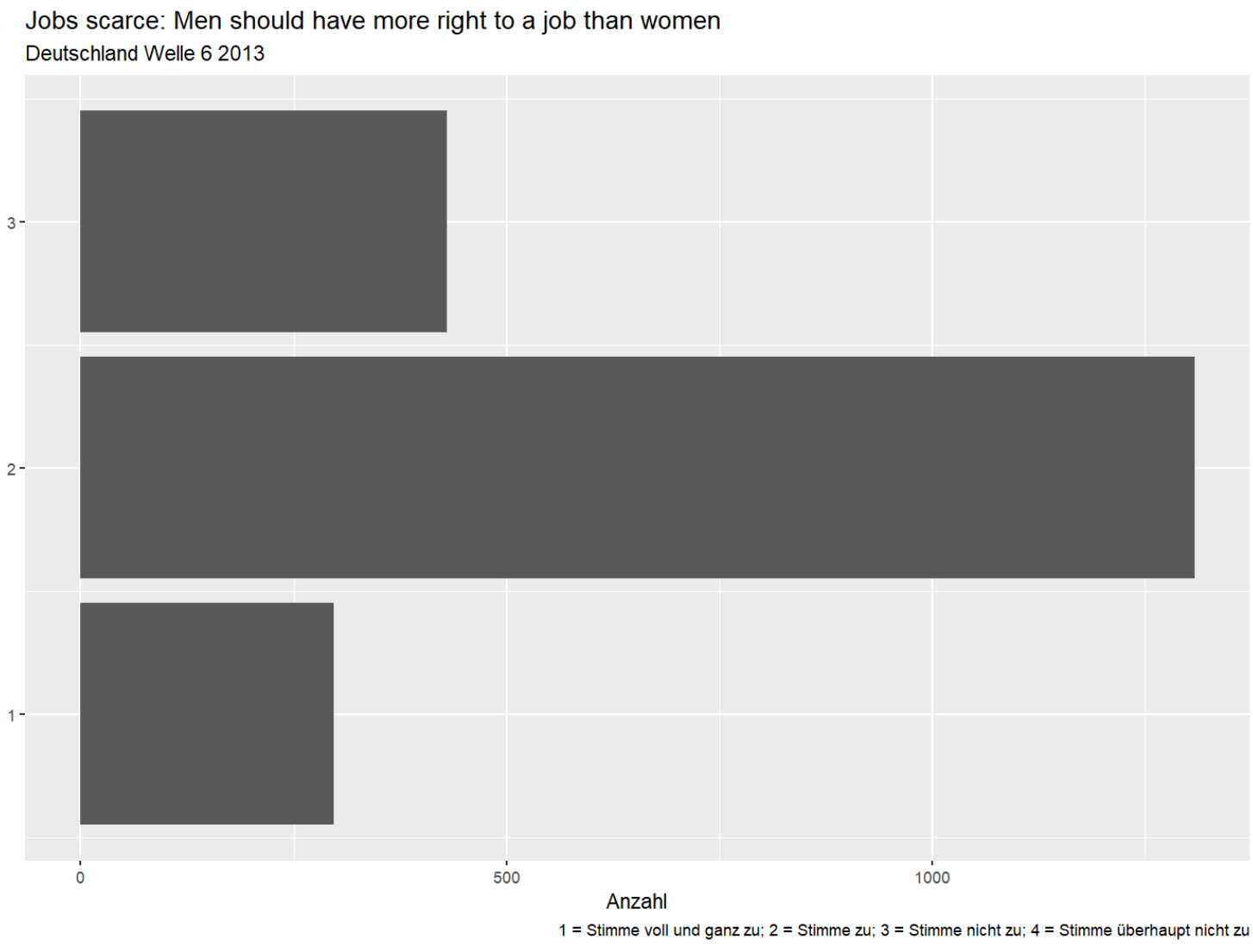
Visualisierungen mit ggplot2

Balkendiagramm geom_bar

Balkendiagramme werden per `geom_bar()` erstellt.
`geom_bar` erwartet **eine** Variable entweder auf der x-Achse oder y-Achse.
Diese wird in den aesthetics `aes()` in `ggplot()` oder in `geom_bar()` definiert.

```
# load data & packages
library(tidyverse)
wvs_data <- readRDS("_raw/wvs_short.rds")
# bar plot Germany & variable y = C001
wvs_data %>%
  filter(S003 == "DEU") %>%
  drop_na(C001) %>%
  ggplot(aes(y = C001)) +
  geom_bar() +
  labs(title = "Jobs scarce: Men should have more right to a job thar
  subtitle = "Deutschland Welle 6 2013",
  caption = "1 = Stimme voll und ganz zu; 2 = Stimme zu; 3 = Stimme r
  x = "Anzahl", y = "")
```

Um den Plot besser zu verstehen, haben wir mit `labs()` Titel und Achsenbeschriftungen festgelegt.


Jobs scarce: Men should have more right to a job than women
Deutschland Welle 6 2013
3
2
1
0
500
1000
Anzahl
1 = Stimme voll und ganz zu; 2 = Stimme zu; 3 = Stimme nicht zu; 4 = Stimme überhaupt nicht zu

Verteilung von kontinuierlichen Variablen visualisieren

Um die Verteilung von kontinuierlichen (numerischen) Variablen darzustellen, sind Histogramme, Dichteplots und Boxplots geeignet.

Im WVS sind wenige kontinuierliche Variablen. Daher bilden wir einen Index, der die Antworten von drei Variablen zu dem Thema "Erwerbsarbeit von Frauen" zusammenfasst.

```
wvs_data_index <- wvs_data %>%
  # answer "neither" transformed to NA
  naniar::replace_with_na(replace = list(C001 = 3)) %>%
  # create women_index variable
  mutate(C001 = (((C001 - 1) / 1) - 1) * -1 ) %>%
  mutate(D057 = (((D057 - 1) / 3) - 1) * -1 ) %>%
  mutate(D063_B = (D063_B - 1) / 2) %>%
  # create women_index variable
  mutate(women_index = (C001 + D057 + D063_B) / 3)
```

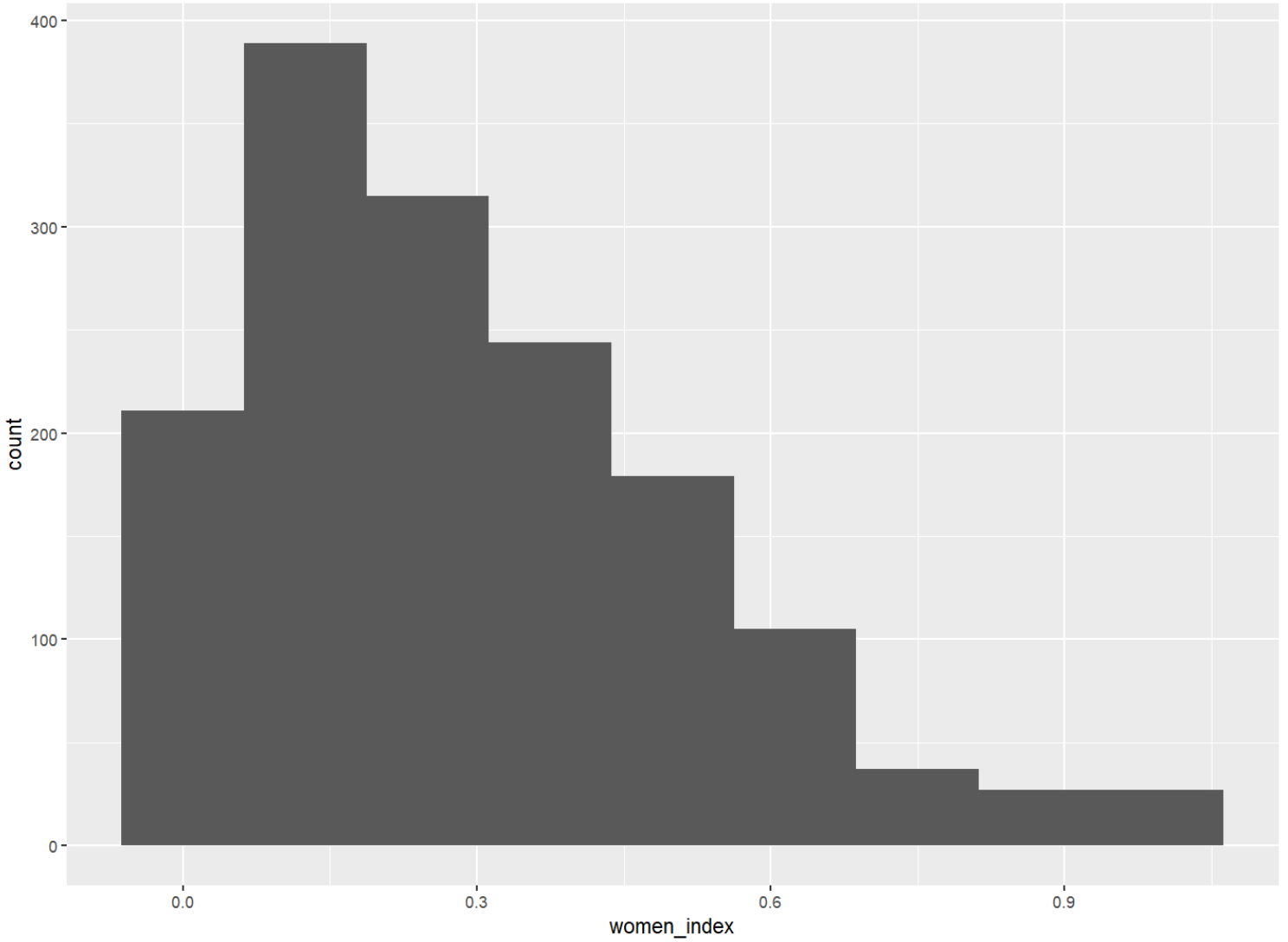
Histogramm geom_histogram()

Die Syntax von `geom_histogram()` ist sehr ähnlich zu `geom_bar()`.

Ein wichtiger Unterschied ist, dass R die Säulenbreite und damit die Anzahl der Säulen automatisch festlegt. Wir können diese aber auch manuell festlegen mit `binwidth = Säulenbreite in Einheit der Variable`.

```
wvs_data_index %>%
  filter(S003 == "DEU") %>%
  ggplot(aes(x = women_index)) +
  geom_histogram(binwidth = 0.125) # bin width 0.125 = 8 bars
```

```
## Warning: Removed 512 rows containing non-finite values (stat_bin).
```



Dichteplots geom_density()

Histogramme fassen Werte in Säulen zusammen und reduzieren damit die Komplexität der Verteilung. Dichteplots per `geom_density()` zeigen ein detaillierteres Bild.

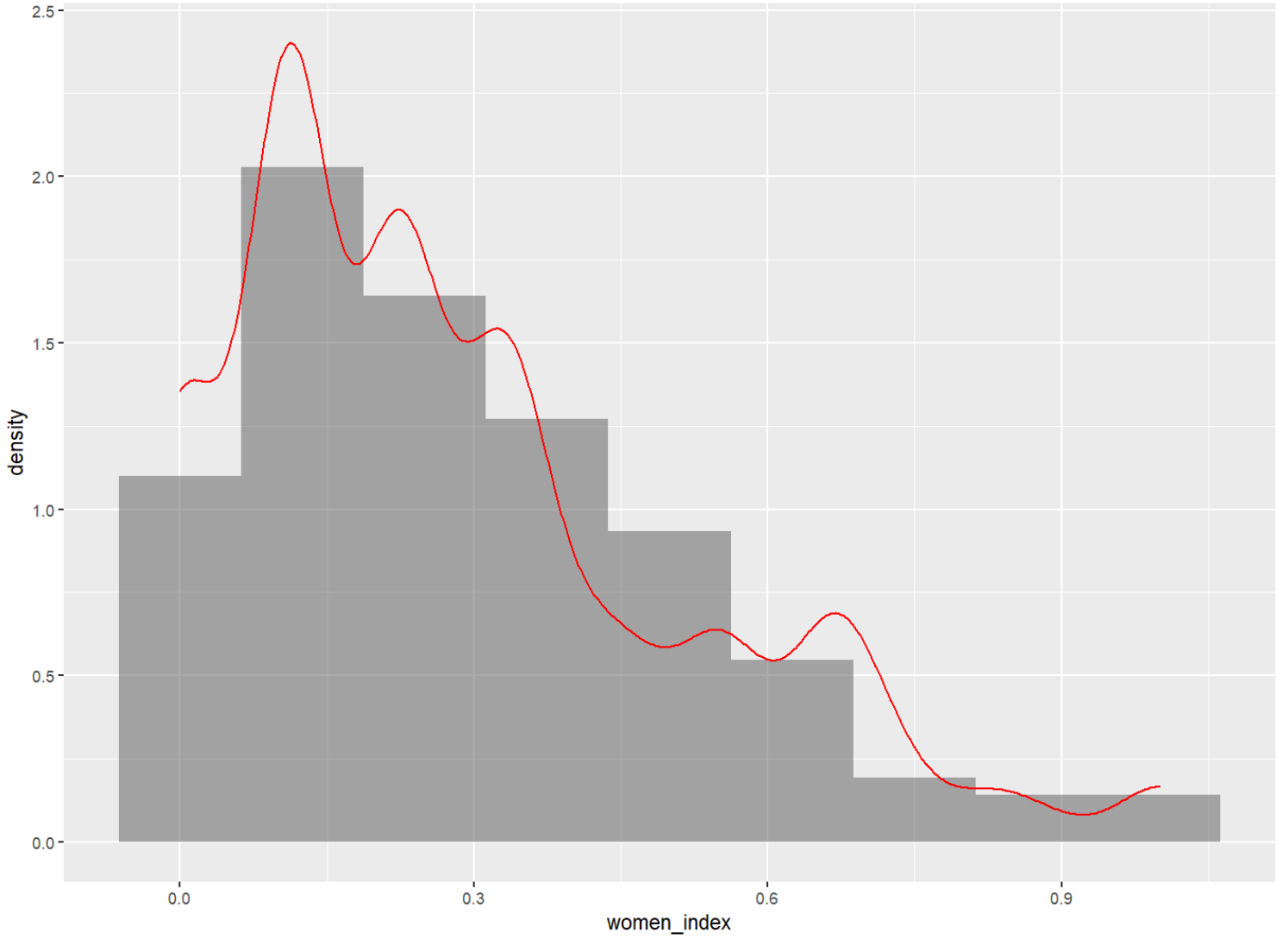
Da ggplot2 mit Ebenen arbeitet, können wir den Dichteplot auch auf das Histogramm legen. Falls wir nur den Dichteplot haben möchten, lassen wir die Zeile `geom_histogram()` einfach weg.

```
wvs_data_index %>%
  filter(S003 == "DEU") %>%
  ggplot(aes(x = women_index)) +
  geom_histogram(aes(y = ..density..), binwidth = 0.125, alpha = 0.5)
  geom_density(colour = "red")
```

`geom_histogramm()` zeigt normalerweise die Anzahl der Ausprägungen auf der y-Achse. `geom_density()` zeigt die Dichte auf der y-Achse. Um die zwei Ebenen in einem Plot anzuzeigen, müssen wir die Skala des Histogramms per `aes(y = ..density..)` ändern. Alle andere Optionen ändern das Aussehen der Plots.

```
## Warning: Removed 512 rows containing non-finite values (stat_bin).

## Warning: Removed 512 rows containing non-finite values (stat_density).
```



Boxplots geom_boxplot()

Boxplots plotten die Werte von Ausreißern, das 25%- , das 50%- und das 75%-Quantil von numerischen kontinuierlichen Variablen.

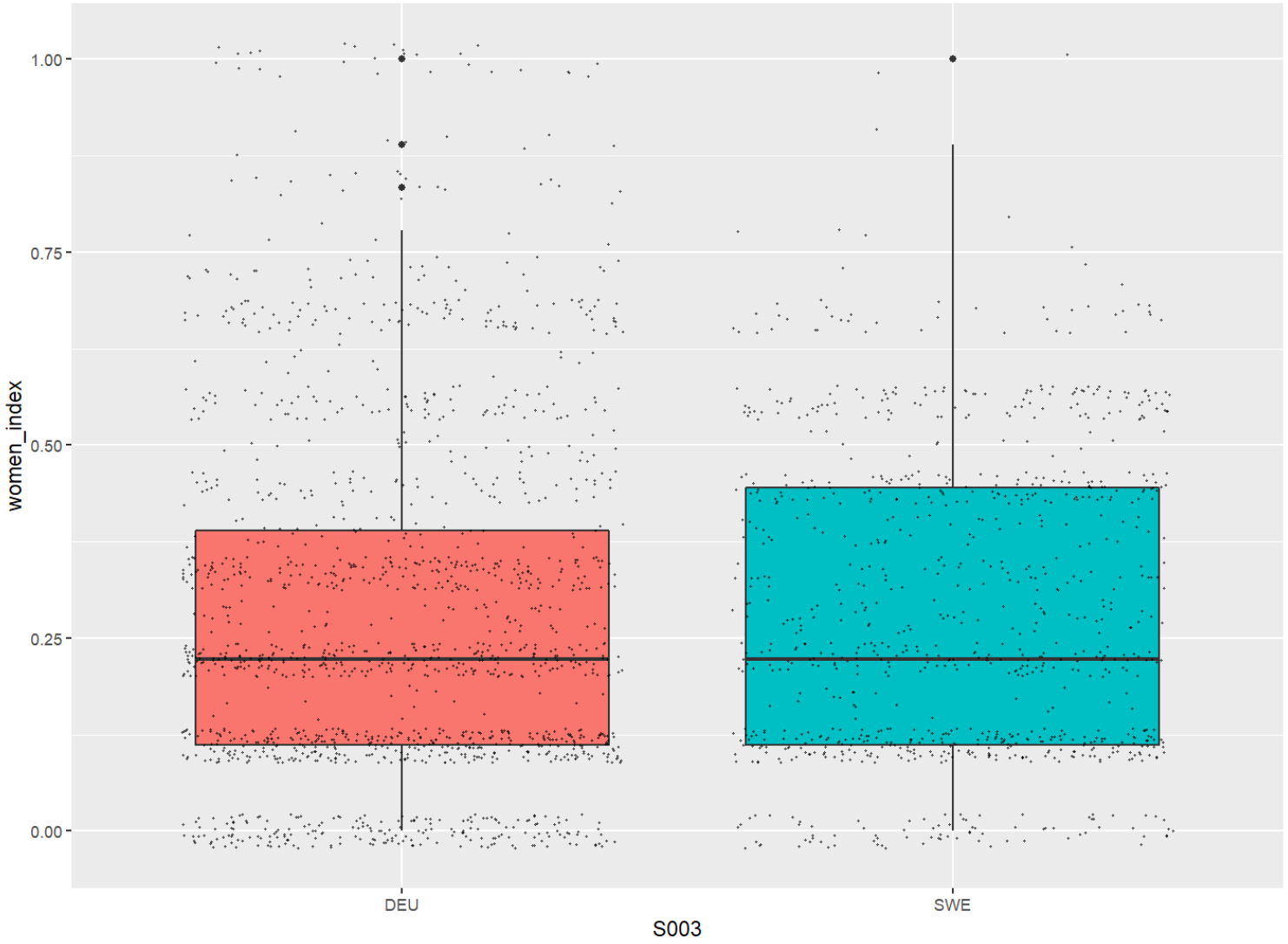
Die Variable für das Boxplot wird am besten auf der y-Achse angezeigt. Es kann eine andere Variable zusätzlich auf der x-Achse angezeigt werden, diese darf aber nicht kontinuierlich sein, sondern muss kategorial sein. Dann wird pro Wert der zweiten Variable ein Boxplot angezeigt.

```
wvs_data_index %>%
  ggplot(aes(y = women_index, x = S003, fill= S003)) +
  geom_boxplot() +
  geom_jitter(color="black", size=0.4, alpha=0.5) +
  theme(legend.position="none")
```

Mit `geom_jitter()` zeigen wir zusätzlich alle Beobachtungen

```
## Warning: Removed 768 rows containing non-finite values (stat_boxplot).

## Warning: Removed 768 rows containing missing values (geom_point).
```



R Sprechstunde Inhalte

Folgende Punkte haben wir diese Woche in der R Sprechstunde (07.05.) besprochen:

R Sprechstunde Inhalte

1. End- und Anfangspunkte der x- oder y-Achse festlegen.

2. Angezeigte Werte von YEAR auf der x- oder y-Achse:
   a. Umwandlung YEAR zu Faktor-Variable.
   b. Skala der y-Achse definieren, so dass nur ganze Jahreszahlen angezeigt werden.

3. Antwortitems umbenennen.

```r
# Vorbereitung
## Pakte und Daten laden
library(tidyverse)
library(naniar)

## wvs data
wvs_data <- readRDS("_raw/wvs_short.rds")
## Index erstellen
wvs_data <- wvs_data %>%
  # answer "neither" transformed to NA
  replace_with_na(replace = list(C001 = 3)) %>%
  # 0-1 scale of each variable
  mutate(C001_trans = (((C001 - 1) / 1) - 1) * -1 ) %>%
  mutate(D057_trans = (((D057 - 1) / 3) - 1) * -1 ) %>%
  mutate(D063_B_trans = (D063_B - 1) / 2) %>%
  # create women_index variable
  mutate(women_index = (C001_trans + D057_trans + D063_B_trans) / 3)
  select(-C001_trans, -D057_trans, -D063_B_trans)

## SOCX Data
socx_data <- read_csv("_raw/SOCX_AGG_20042020191205895.csv")
## yearly relative expenses
socx_data <- socx_data %>%
  mutate(cash_prop = (family_cash_total_pct_gdp / family_total_total_
  mutate(services_prop = (family_service_total_pct_gdp / family_total
```

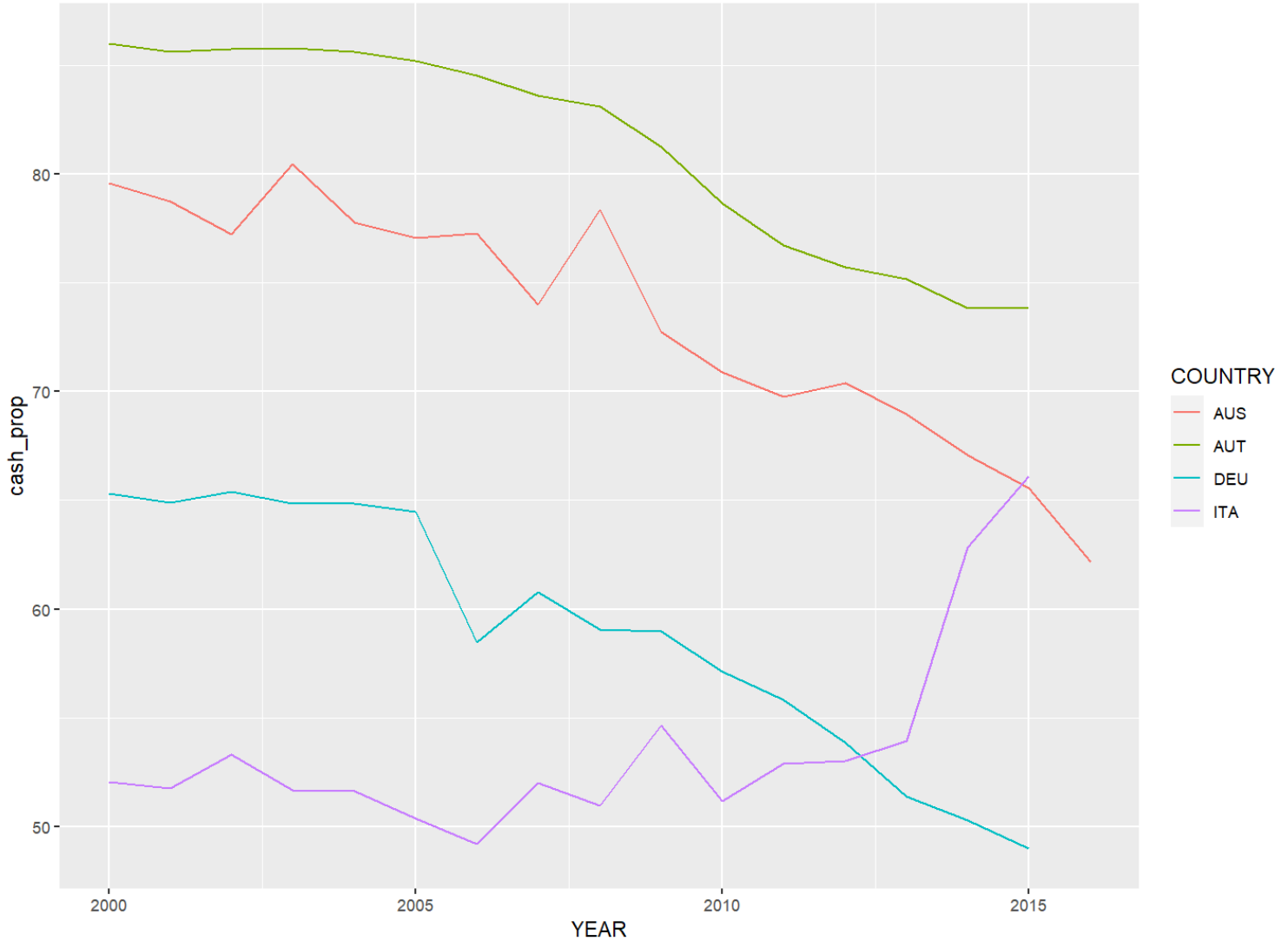
1. End- und Anfangspunkte der x- oder y-Achse festlegen

Vergleich der relativen Ausgaben für Sachleistungen (cash_prop) von vier Ländern.

```
socx_data %>%
  filter(COUNTRY %in% c("AUT", "AUS", "DEU", "ITA")) %>%
  ggplot(aes(x = YEAR, y = cash_prop, colour = COUNTRY)) +
  geom_line()
```

Da keiner der Werte unter circa 50% fällt wird dieser Bereich nicht angezeigt.

```
## Warning: Removed 1 row(s) containing missing values (geom_path).
```



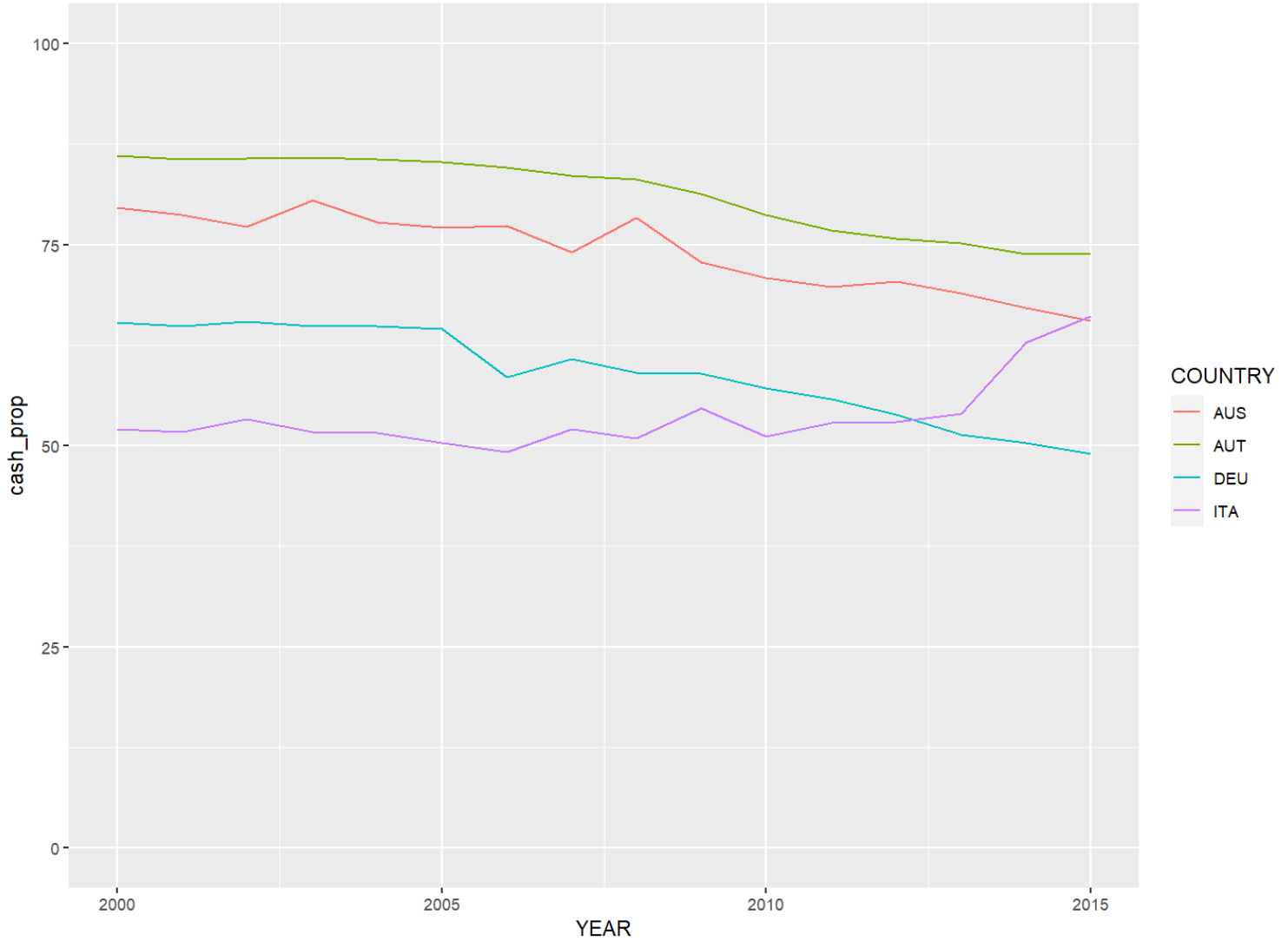
1. Die End- und Anfangspunkte der x- oder y-Achse festlegen

Mit `xlim()` und `ylim()` kann der Start- und Endpunkt der Skalen festgelegt werden.

```
socx_data %>%
  filter(COUNTRY %in% c("AUT", "AUS", "DEU", "ITA")) %>%
  ggplot(aes(x = YEAR, y = cash_prop, colour = COUNTRY)) +
  geom_line() +
  xlim(2000, 2015) +
  ylim(0, 100)
```

Mehr Infos: https://ggplot2.tidyverse.org/reference/lims.html

```
## Warning: Removed 2 row(s) containing missing values (geom_path).
```

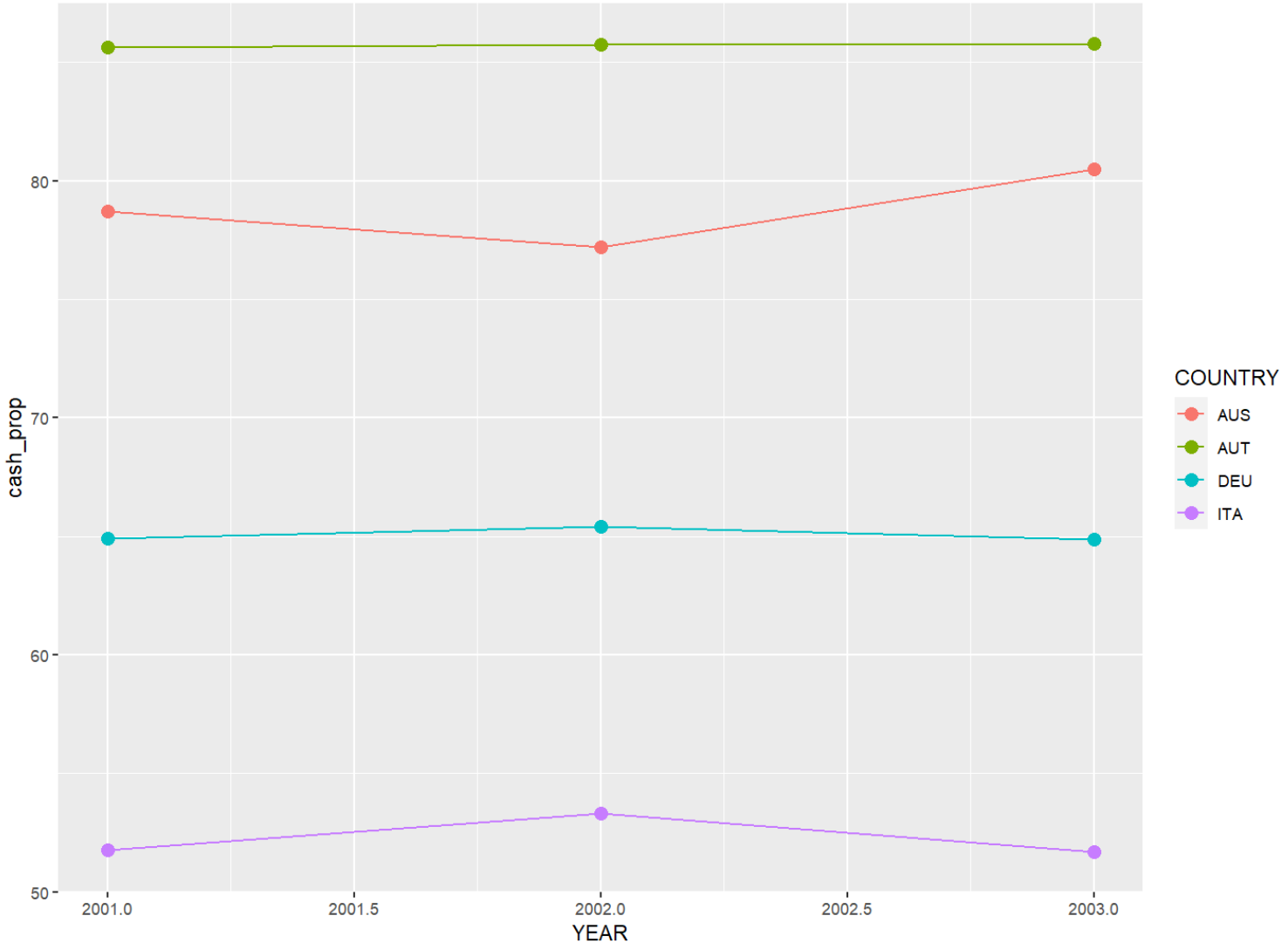


2. Angezeigte Werte von YEAR auf der x- oder y-Achse

ggplot wählt passende Werte für die Achsenbeschriftung.

Da `YEAR` als numerische Variable definiert ist, kann das dazu führen, dass ggplot einfach 2001.5 anzeigt. Obwohl es keine Beobachtungen für Halbjahre im Datensatz gibt.

```
socx_data %>%
  filter(YEAR >= 2001 & YEAR < 2004) %>%
  filter(COUNTRY %in% c("AUT", "AUS", "DEU", "ITA")) %>%
  ggplot(aes(x = YEAR, y = cash_prop, colour = COUNTRY)) +
  geom_point(size = 3) +
  geom_line()
```


cash_prop
80
70
60
50
2001.0
2001.5
2002.0
2002.5
2003.0
YEAR
COUNTRY
AUS
AUT
DEU
ITA

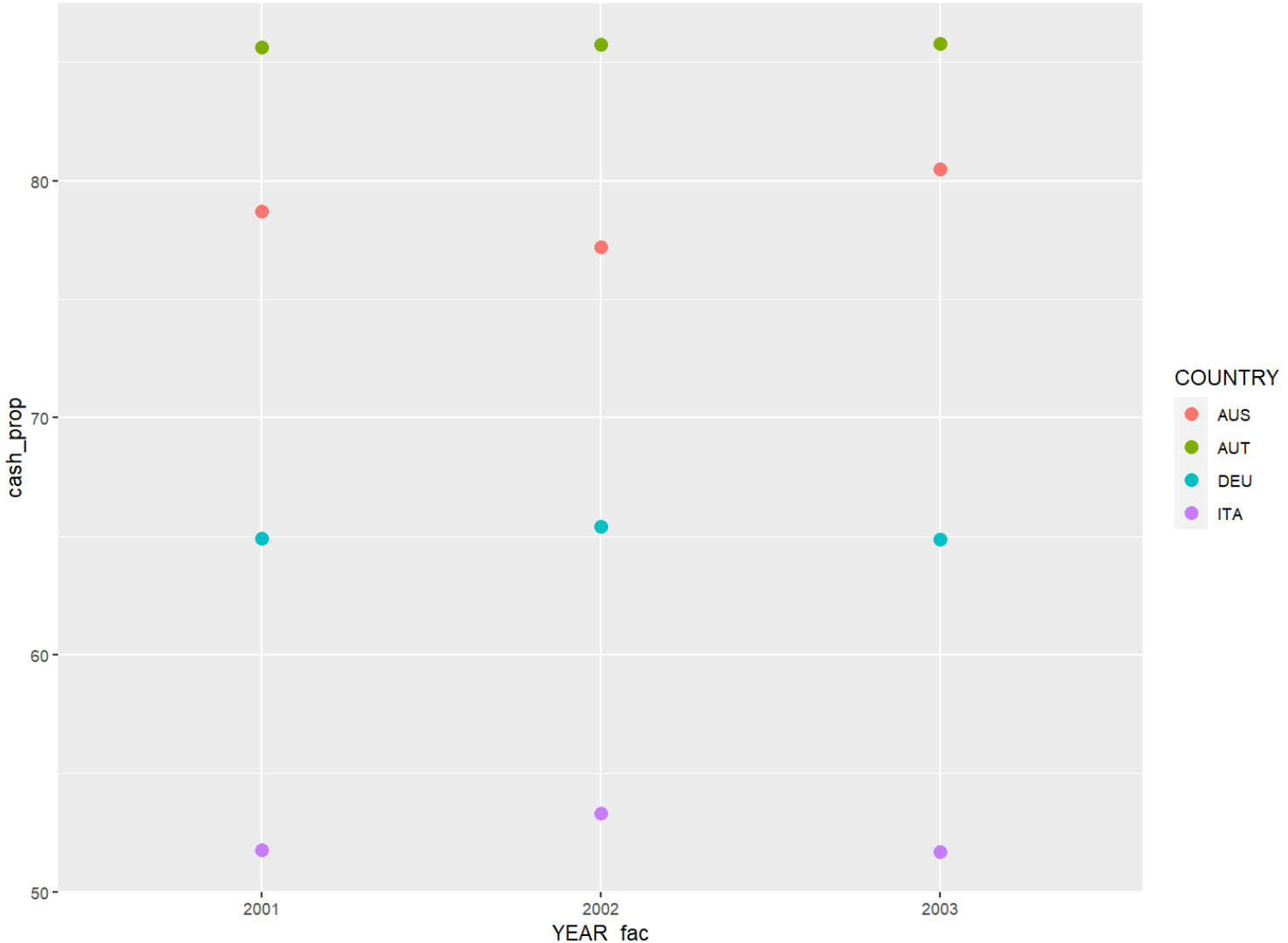
2a. Umwandlung YEAR zu Faktor-Variable

Eine Möglichkeit das Problem zu lösen, ist YEAR in eine Faktor-Variable umzuwandeln.

YEAR ist eine ordinalskalierte Variable, deshalb nutzen wir `as.ordered()` und nicht `as.factor()`.

```
socx_data <- socx_data %>%
  mutate(YEAR_fac = as.ordered(YEAR))

socx_data %>%
  filter(YEAR_fac %in% c(2001, 2002, 2003)) %>%
  filter(COUNTRY %in% c("AUT", "AUS", "DEU", "ITA")) %>%
  ggplot(aes(x = YEAR_fac, y = cash_prop, colour = COUNTRY)) +
  geom_point(size = 3)
```

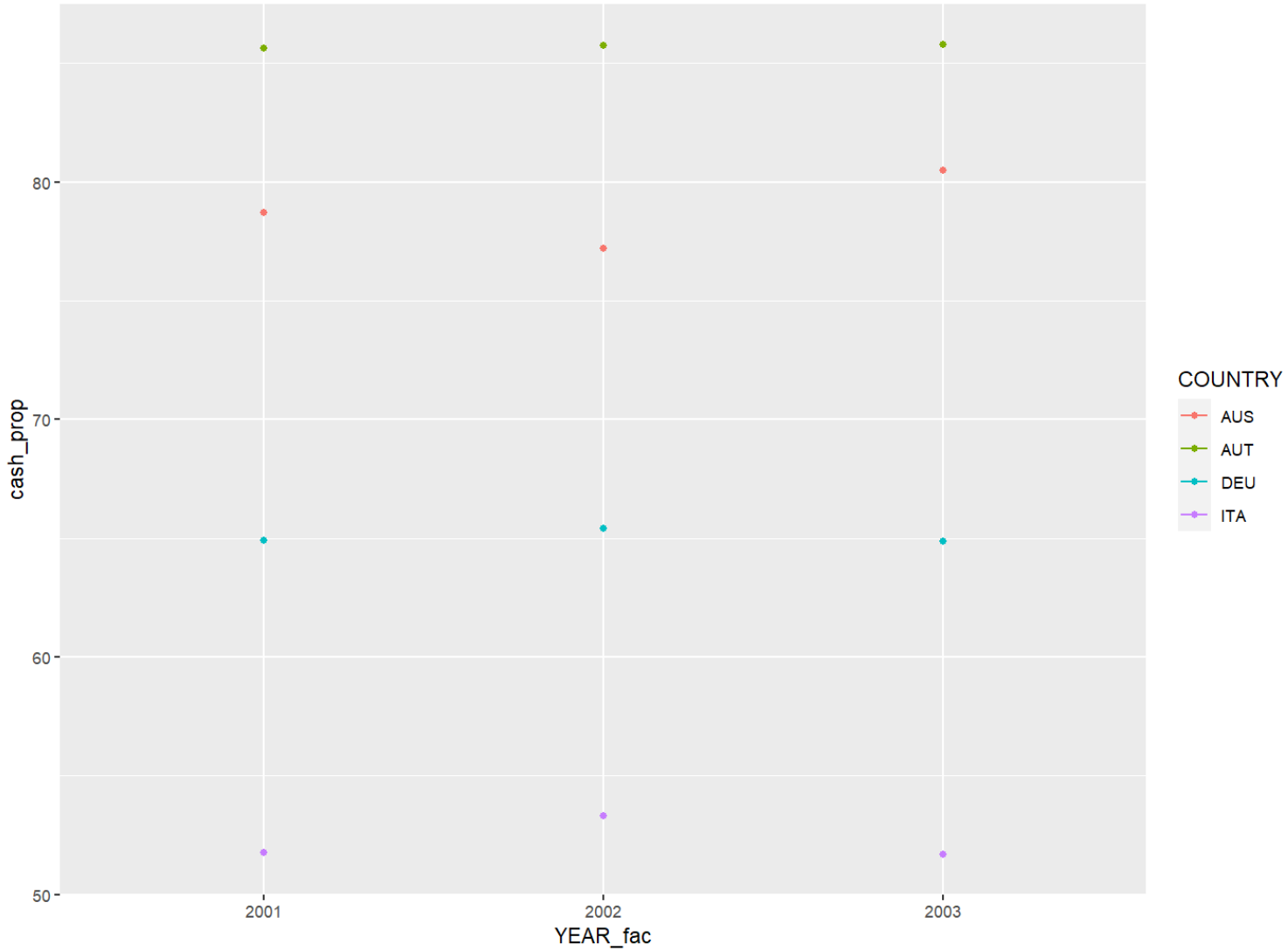

cash_prop
80
70
60
50
2001
2002
2003
YEAR_fac
COUNTRY
AUS
AUT
DEU
ITA

2a. Umwandlung YEAR zu Faktor-Variable

Leider funktionieren einige geom_functions nicht mit Faktor-Variablen zB `geom_line()`.

```
socx_data %>%
  filter(YEAR_fac %in% c(2001, 2002, 2003)) %>%
  filter(COUNTRY %in% c("AUT", "AUS", "DEU", "ITA")) %>%
  ggplot(aes(x = YEAR_fac, y = cash_prop, colour = COUNTRY)) +
  geom_point(size = 3) +
  geom_line()
```

```
## geom_path: Each group consists of only one observation. Do you need to adj
## the group aesthetic?
```



2b. Skala der y-Achse definieren, so dass nur ganze Jahreszahlen angezeigt werden

Wir können auch einfach die auf den Achsen angezeigten Werte festlegen. Dazu müssen wir `YEAR` nicht umwandeln. Das funktioniert mit `scale_x_continous()`, `scale_y_continous()` aber auch `scale_x_discrete()`, `scale_y_discrete()`

Folgende Optionen können mit den Funktionen verändert werden:

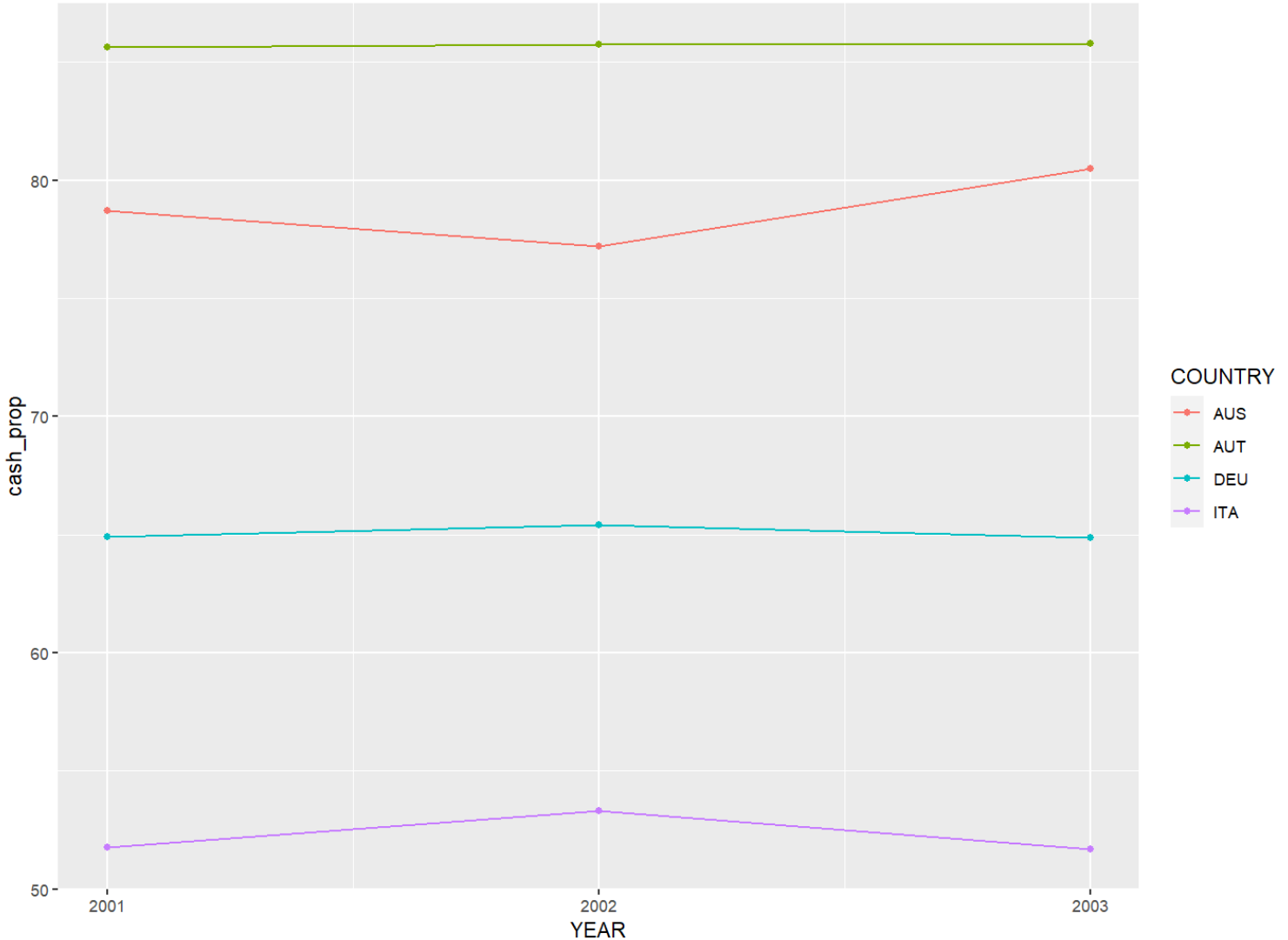
- **name** : x or y axis labels
- **breaks** : control the breaks in the guide (axis ticks, grid lines, …):
  - *NULL* : hide all breaks
  - *waiver()* : the default break computation
  - a character or numeric vector specifying the breaks to display
- **labels** : labels of axis tick marks. Allowed values are :
  - *NULL* for no labels
  - *waiver()* for the default labels
  - character vector to be used for break labels
- **limits** : a numeric vector specifying x or y axis limits (min, max)
- **trans for axis transformations**. Possible values are “log2”, “log10”, “sqrt”, etc.

2b. Skala der y-Achse definieren, so dass nur ganze Jahreszahlen angezeigt werden

YEAR ist eine numerische kontinuierliche Variable, daher benötigen wir `scale_x_continuous()`.

Um nur ausgewählte Werte als Beschriftung auf der x-Achse anzuzeigen, nutzen wir `breaks =`

```
socx_data %>%
  filter(YEAR %in% c(2001, 2002, 2003)) %>%
  filter(COUNTRY %in% c("AUT", "AUS", "DEU", "ITA")) %>%
  ggplot(aes(x = YEAR, y = cash_prop, colour = COUNTRY)) +
  geom_point() +
  geom_line() +
  scale_x_continuous(breaks = c(2001, 2002, 2003))
```

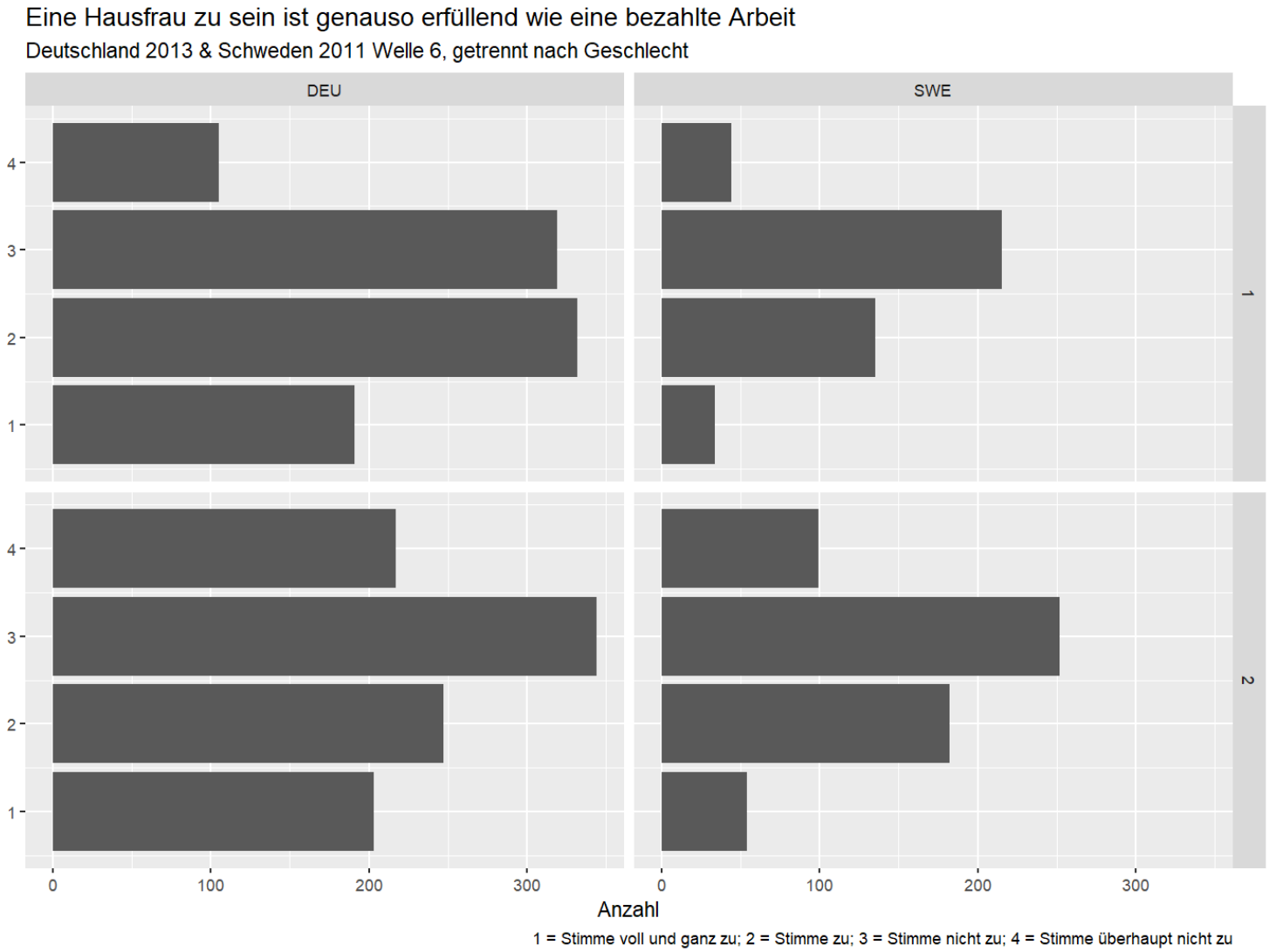

cash_prop
80
70
60
50
2001
2002
2003
YEAR
COUNTRY
AUS
AUT
DEU
ITA

3. Antwortitems umbenennen

In den WVS-Daten werden die Antwortmöglichkeiten der einzelnen Fragen nur als Nummern dargestellt. Das führt dazu, dass es manchmal nicht eindeutig ist wofür 1 oder 2 steht

```
wvs_data %>%
  ggplot(aes(y = D057)) + # define y-axis variable & data
  geom_bar() + # set boxplot  as geom_function
  facet_grid(
    cols = vars(S003),
    rows = vars(X001)) + # set facet variables
  labs(title = "Eine Hausfrau zu sein ist genauso erfüllend wie eine
       subtitle = "Deutschland 2013 & Schweden 2011 Welle 6, getrennt
       caption = "1 = Stimme voll und ganz zu; 2 = Stimme zu; 3 = St
       y = "", x = "Anzahl")
```

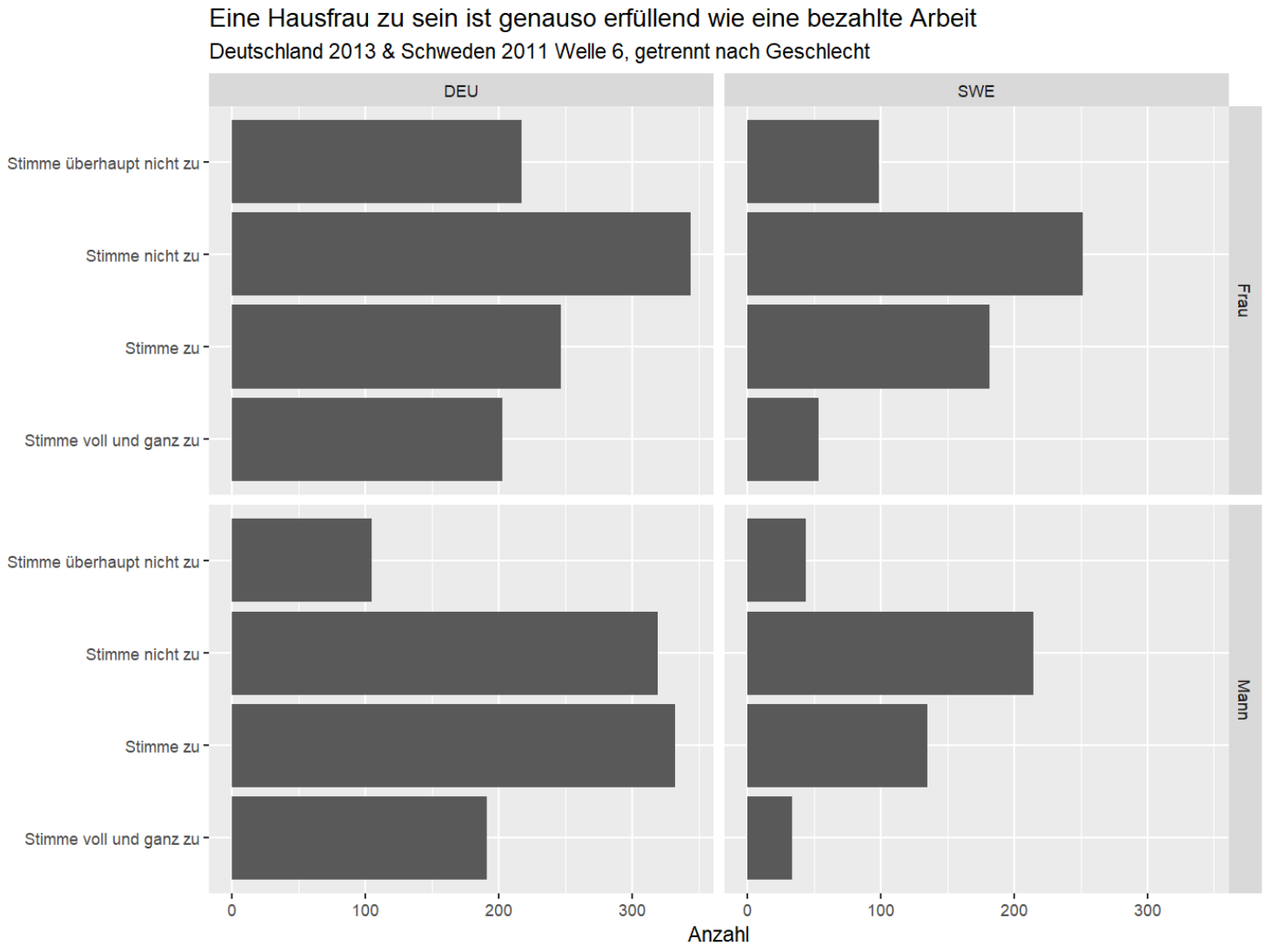
```
## Warning: Removed 279 rows containing non-finite values (stat_count).
```



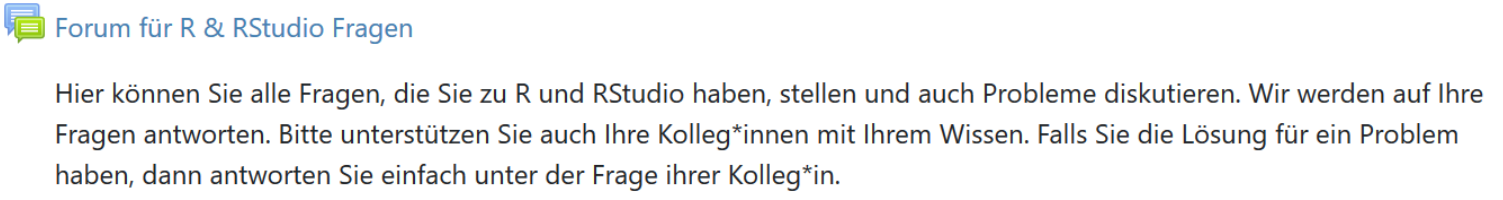
3. Antwortitems umbenennen

Wir können die Antwortitems per `recode()` direkt in der Variable umbenennen, dadurch wird `X001` zur Character-Variable und `D507` in Kombination mit `as.ordered()` zu ordinalskalierten Character-Variable.

```
wvs_data <- wvs_data %>%
  mutate(X001_chr = recode(X001, "1" = "Mann", "2" = "Frau")) %>%
  mutate(D057_chr = recode(as.ordered(D057),
                           "1" = "Stimme voll und ganz zu",
                           "2" = "Stimme zu",
                           "3" = "Stimme nicht zu",
                           "4" = "Stimme überhaupt nicht zu"))
wvs_data %>%
  drop_na(D057_chr) %>%
  ggplot(aes(y = D057_chr)) +
  geom_bar() +
  facet_grid(cols = vars(S003), rows = vars(X001_chr)) + # set facet
  labs(title = "Eine Hausfrau zu sein ist genauso erfüllend wie eine
       subtitle = "Deutschland 2013 & Schweden 2011 Welle 6, getrennt
       y = "", x = "Anzahl")
```


Eine Hausfrau zu sein ist genauso erfüllend wie eine bezahlte Arbeit
Deutschland 2013 & Schweden 2011 Welle 6, getrennt nach Geschlecht
DEU
SWE
Frau
Mann
Stimme überhaupt nicht zu
Stimme nicht zu
Stimme zu
Stimme voll und ganz zu
0
100
200
300
Anzahl

Falls Sie noch Fragen haben, nutzen Sie das **Forum** auf moodle und unterstützen Sie Ihre Kolleg*innen mit Ihrem Wissen!



Nutzen Sie auch unsere **R Sprechstunde**.
Jeden Donnerstag von 11:00 bis 11:45 auf zoom oder im Anschluss an die zoom-Sitzung zu Familienpolitik (Link finden Sie auf moodle).

Beziehungen zwischen Variablen

Kreuztabelle zweier Variablen (absolut)

`table()` erstellt eine Kreuztabelle, welche die absoluten Häufigkeiten der Kombinationen aller Werte zweier Variablen enthält.

```
# absolute Häufigkeiten
table(wvs_data$C001, # erste Variable (Reihen)
      wvs_data$D057) # zweite Variable(Spalten)
```

```
##
##       1   2   3   4
##   1 108  89  72  39
##   2 297 641 891 375
```

Obwohl die Variablen beide numerisch sind, interpretiert R sie richtig als kategoriale Variablen.

Kreuztabelle zweier Variablen (relativ)

`prop.table` erstellt eine Kreuztabelle, welche die relativen Häufigkeiten der Kombinationen aller Werte zweier Variablen enthält.

```
# relative Häufigkeit
# im Verhältnis zur Summe aller Beobachtungen
prop.table(table(wvs_data$C001, wvs_data$D057))
```

```
##
##                1          2          3          4
##   1 0.04299363 0.03542994 0.02866242 0.01552548
##   2 0.11823248 0.25517516 0.35469745 0.14928344
```

Kreuztabelle zweier Variablen (relativ)

```
# im Verhältnis zur Randsumme je Reihe
prop.table(table(wvs_data$C001, wvs_data$D057), 1)
```

```
##
##             1         2         3         4
##   1 0.3506494 0.2889610 0.2337662 0.1266234
##   2 0.1347550 0.2908348 0.4042650 0.1701452
```

```
# im Verhältnis zur Randsumme je Spalte
prop.table(table(wvs_data$C001, wvs_data$D057), 2)
```

```
##
##              1          2          3          4
##   1 0.26666667 0.12191781 0.07476636 0.09420290
##   2 0.73333333 0.87808219 0.92523364 0.90579710
```

Chi-Quadrat-Test - Unabhängigkeitstest

Per `chisq.test(table())` können wir überprüfen, ob die zwei Variablen unabhängig voneinander sind.

Ein signifikanter Test bedeutet, dass die beide Variablen nicht unabhängig voneinander sind.

```
chisq.test(table(wvs_data$C001, wvs_data$D057))
```

```
##
##     Pearson's Chi-squared test
##
## data:  table(wvs_data$C001, wvs_data$D057)
## X-squared = 101.73, df = 3, p-value < 2.2e-16
```

Da unser p-value niedriger als 0.05 ist, sind C001 und D057 nicht unabhängig voneinander.

Kreuztabellen drei Variablen

Kreuztabellen können aber auch mehr als zwei Variablen berücksichtigen. Fügen wir `S003` hinzu, werden zwei Kreuztabellen erstellt. Eine für Deutschland und eine für Schweden.

```
table(wvs_data$C001,
      wvs_data$D057,
      wvs_data$S003)
```

```
## , ,  = DEU
##
##
##       1   2   3   4
##   1 103  81  65  39
##   2 221 350 449 236
##
## , ,  = SWE
##
##
##       1   2   3   4
##   1   5   8   7   0
##   2  76 291 442 139
```

Korrelationen

Ob kontinuierliche Variablen zusammenhängen, können wir mit `cor()` testen. Korrelieren `X047` "Höhe des Einkommens" und der `women_index`?

Da wir fehlende Werte im Datensatz haben, müsse wir mit `use = "complete.obs"` spezifizieren, dass nur nicht-fehlende Werte berücksichtigt werden.

```
cor(wvs_data$X047, wvs_data$women_index, use = "complete.obs")
```

```
## [1] -0.04388441
```

Die Korrelation der zwei Variablen ist negativ aber auch sehr schwach.

Der Zusammenhang zwischen Variablen sieht man am besten mit Plots. Nächste Woche werden wir die Zusammenhänge von kategorialen und kontinuierlichen Variablen visualisieren.

Übung 10

- Verwenden Sie den wvs_short Datensatz.
  (oder auch den Originaldatensatz, wenn sie andere Variablen und Zeiträume verwenden wollen)
- Wählen Sie zwei kategoriale Variablen:
  - erstellen Sie eine Kreuztabelle mit absoluten Häufigkeiten.
  - erstellen Sie eine Kreuztabelle mit relativen Häufigkeiten.
- Testen Sie die Korrelation zweier Variablen Ihrer Wahl.
- Laden Sie Ihr Skript bis zum 15.05. 12:00 auf moodle hoch.