Datenanalyse mit R

11 Visuelle Darstellung mehrerer Variablen

Tobias Wiß, Carmen Walenta und Felix Wohlgemuth

15.05.2020

Daten für diese Woche

Wie in den letzten Wochen verwenden wir für diese Einheit die Daten des World Value Surveys Welle 6. Diese Woche erstellen wir aber unseren Datensatz direkt aus dem Welle 6 Datensatz und nicht aus den Time Series Datensatz.

- Damit können wir ein paar zusätzliche Variablen verwenden und einen verfeinerten Index zu Erwerbstätigkeit von Frauen erstellen.
- Die Reihenfolge der Antwortitems (Agree, Neither, Disagree) ist nun kohärent.
- Die Variablennamen haben sich leider geändert. Im Codebook auf moodle sind beide Variablennamen aufgelistet.

Sie können direkt mit dem verkürzten Datensatz arbeiten. Sie finden den Datensatz `wvs_short_w6.rds` und `wvs_short_w6.csv` auf moodle und das Codebook `wvs_short_w6_codebook.pdf`.

Auf den folgenden Folien befindet sich das Skript mit dem ich den Datensatz erstellt habe. Sie müssen den Code nicht ausführen.

WVS 6 Datensatz

```r
# Preliminaries
library(tidyverse)
library(naniar)
# install.packages("countrycode")
library(countrycode)

# import data
wvs_w6_data <- readRDS("_raw/F00007762-WV6_Data_R_v20180912.rds")

# reduce dataframe
# define NAs (exception Y003)
# recode countrycodes
wvs_w6_data <- wvs_w6_data %>%
  select(V2, V3, V258, S019, V240,V242, V57, V58,
         V248, V239, V229, V235, Y001, Y002, Y003,
         V4, V45, V102, V54, V50, V48, V47) %>%
  replace_with_na_at(.vars = variable.names(wvs_w6_data)[ variable.na
    condition = ~.x <= -1) %>%
  replace_with_na(replace = list(Y003 = -5)) %>%
  mutate(V2 = countrycode(V2, "wvs", "iso3c"))
```

WVS 6 Datensatz

```r
# create NEW women_index
wvs_w6_data <- wvs_w6_data %>%
  mutate(women_index = (
    ((V45 - 1) / 2) +
    ((V54 - 1) / 3) +
    ((((V48 - 1) / 2) -1 ) * -1) +
    ((V50 - 1) / 3) +
    ((V47 - 1) / 2)) / 5)

# set categorical variables
wvs_w6_data[c(5, 7:22)] <- lapply(wvs_w6_data[c(5, 7:22)] , function(
wvs_w6_data$V2 <- as.factor(wvs_w6_data$V2)

# export dataframe as .rds and .csv
saveRDS(wvs_w6_data, "data/wvs_short_w6.rds")
write_csv(wvs_w6_data, "data/wvs_short_w6.csv")
```

Daten für diese Woche

Da die einzige kontinuierliche Variable im WVS Datensatz unser Index ist, benötigen für die Visualisierung des Zusammenhangs zweier kontinuierlichen Variablen einen anderen Datensatz.

Der Comparative Welfare States 2020 Datensatz beinhaltet Variablen zu den Ausgaben für Sozialpolitik, aber auch sozioökonomische, makroökonomische, demographische und politische Variablen für 22 Länder von 1960 bis 2018.

Den Datensatz und das Codebook finden Sie unter: https://www.lisdatacenter.org/news-and-events/comparative-welfare-states-dataset-2020/

Wiederholung

Beziehungen zwischen Variablen

Zusammenhang zweier Variablen (absolut)

`table()` erstellt eine Kreuztabelle mit den Häufigkeiten aller Wertekombinationen mehrere Variablen.

Wir untersuchen den Zusammenhang zwischen `V54` "Being a housewife is just as fulfilling as working for pay" und `V45` "When jobs are scarce, men should have more right to a job than women".

```
# load data
wvs_w6_data <- readRDS("data/wvs_short_w6.rds")
# absolute Häufigkeiten
table(wvs_w6_data$V54, wvs_w6_data$V45)
```

```
##
##         1     2     3
##   1 10722  3269  7256
##   2 12755  6755 12610
##   3  7271  4327 10733
##   4  2429  1289  4582
```

Zusammenhang zweier Variablen (relativ)

Mit `prop.table()` werden relative Häufigkeiten berechnet.

```
prop.table(table(wvs_w6_data$V54, wvs_w6_data$V45))
```

```
##
##                1           2           3
##    1 0.12764590 0.03891759 0.08638301
##    2 0.15184885 0.08041858 0.15012262
##    3 0.08656158 0.05151313 0.12777685
##    4 0.02891736 0.01534560 0.05454892
```

Zusammenhang zweier Variablen (relativ)

Die Grundeinstellung zeigt Häufigkeiten im Verhältnis zu der Gesamtsumme. `, 1` in Relation zu Reihen- und `, 2` zur Spaltenvariable.

```
prop.table(table(wvs_w6_data$V54, wvs_w6_data$V45), 1)
```

```
##
##              1         2         3
##   1 0.5046359 0.1538570 0.3415070
##   2 0.3971046 0.2103051 0.3925903
##   3 0.3256012 0.1937665 0.4806323
##   4 0.2926506 0.1553012 0.5520482
```

```
prop.table(table(wvs_w6_data$V54, wvs_w6_data$V45), 2)
```

```
##
##               1          2          3
##   1 0.32317569 0.20901535 0.20624769
##   2 0.38445308 0.43190537 0.35843211
##   3 0.21915785 0.27666240 0.30507945
##   4 0.07321337 0.08241688 0.13024075
```

Chi-Quadrat-Test - Unabhängigkeitstest

Per `chisq.test(table())` können wir überprüfen, ob zwei Variablen unabhängig voneinander sind. Ein statistisch signifikanter Test bedeutet, dass die Variablen nicht unabhängig voneinander sind.

```
chisq.test(table(wvs_w6_data$V54, wvs_w6_data$V45))
```

```
##
##     Pearson's Chi-squared test
##
## data:  table(wvs_w6_data$V54, wvs_w6_data$V45)
## X-squared = 2319.9, df = 6, p-value < 2.2e-16
```

Da unser p-value niedriger als 0.05 ist, sind `V54` und `V45` nicht unabhängig voneinander.

Korrelationen

Ob kontinuierliche Variablen zusammenhängen, können wir mit `cor()` testen.

Korrelieren die jährlichen öffentliche Ausgaben für Familienpolitik `family_pub` mit dem Anteil linker Parteien im Parlament `leftseat` oder alternativ mit dem Anteil von Frauen im Parlament `fempar`.

```
# load CWS data
library(readxl)
cws_data <- read_excel("_raw/CWS-data-2020.xlsx")

# correlation
cor(cws_data$family_pub, cws_data$leftseat, use = "complete.obs")
```

```
## [1] 0.3187759
```

```
cor(cws_data$family_pub, cws_data$fempar, use = "complete.obs")
```

```
## [1] 0.5808422
```

Korrelationen

Die Korrelation der Ausgaben für Familienpolitik mit dem Anteil von Frauen im Parlament ist höher als mit dem Anteil linker Parteien im Parlament.

Wie genau der Zusammenhang zwischen den Variablen ist, erkennt man am besten mit Visualisierungen.

Visuelle Darstellung mehrerer Variablen

Zusammenhang zwischen Variablen visualisieren

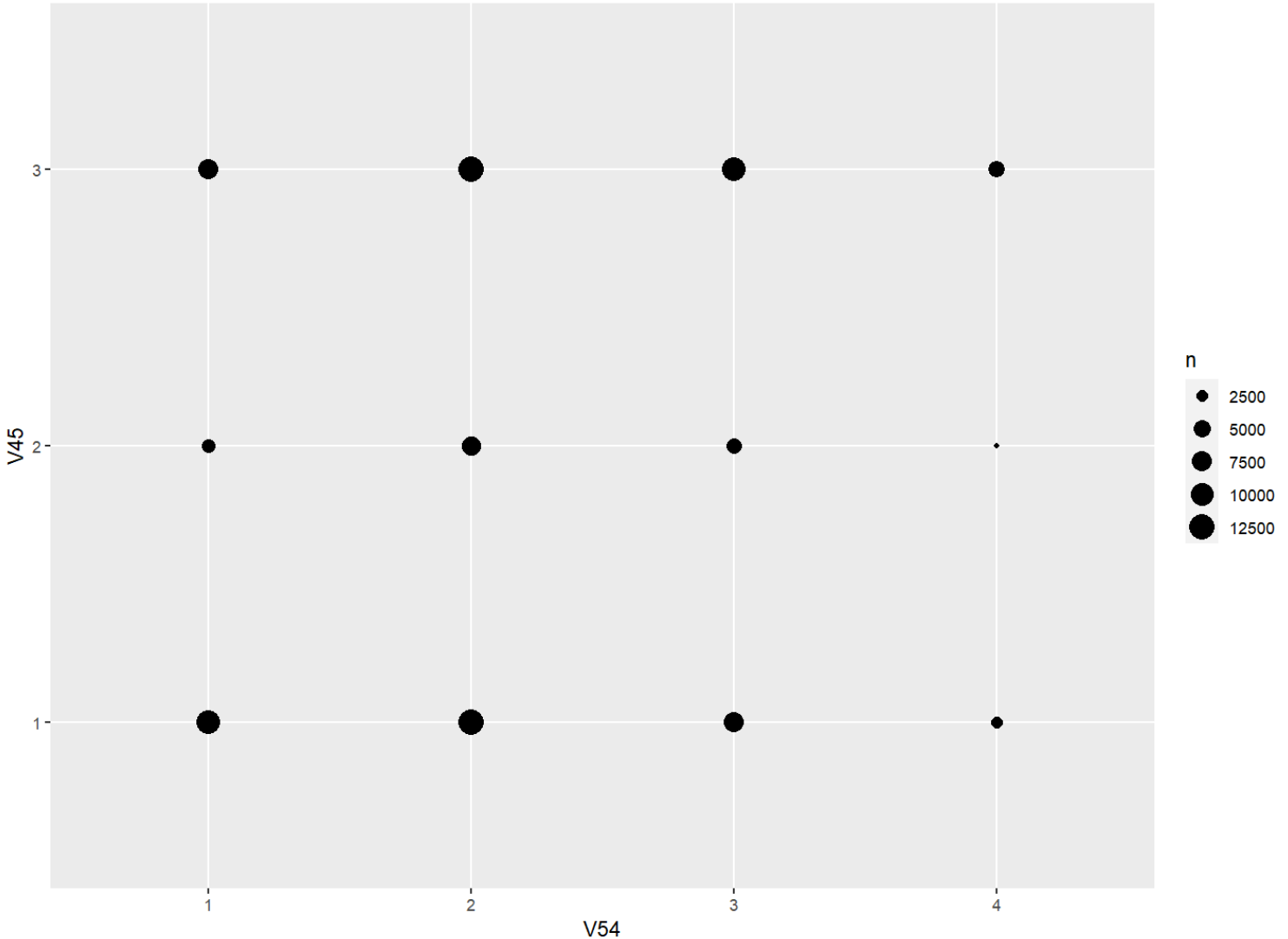
`chisq.test()` und `cor()` zeigt uns die Stärke des Zusammenhangs zwischen Variablen unterschiedlichen Typs.

Manchmal sind die Tests nicht ganz eindeutig oder wir wollen erstmal die Variablen finden, die eventuell zusammenhängen. Das geht am besten mit der Visualisierung der Zusammenhangs mehrerer Variablen.

Zusammenhang kategorialer Variablen

Mit `geom_count()` visualisieren wir die Werte von `table()`. Je größer der Punkt, desto größer ist die Häufigkeit der Kombination der Werte zweier Variablen.

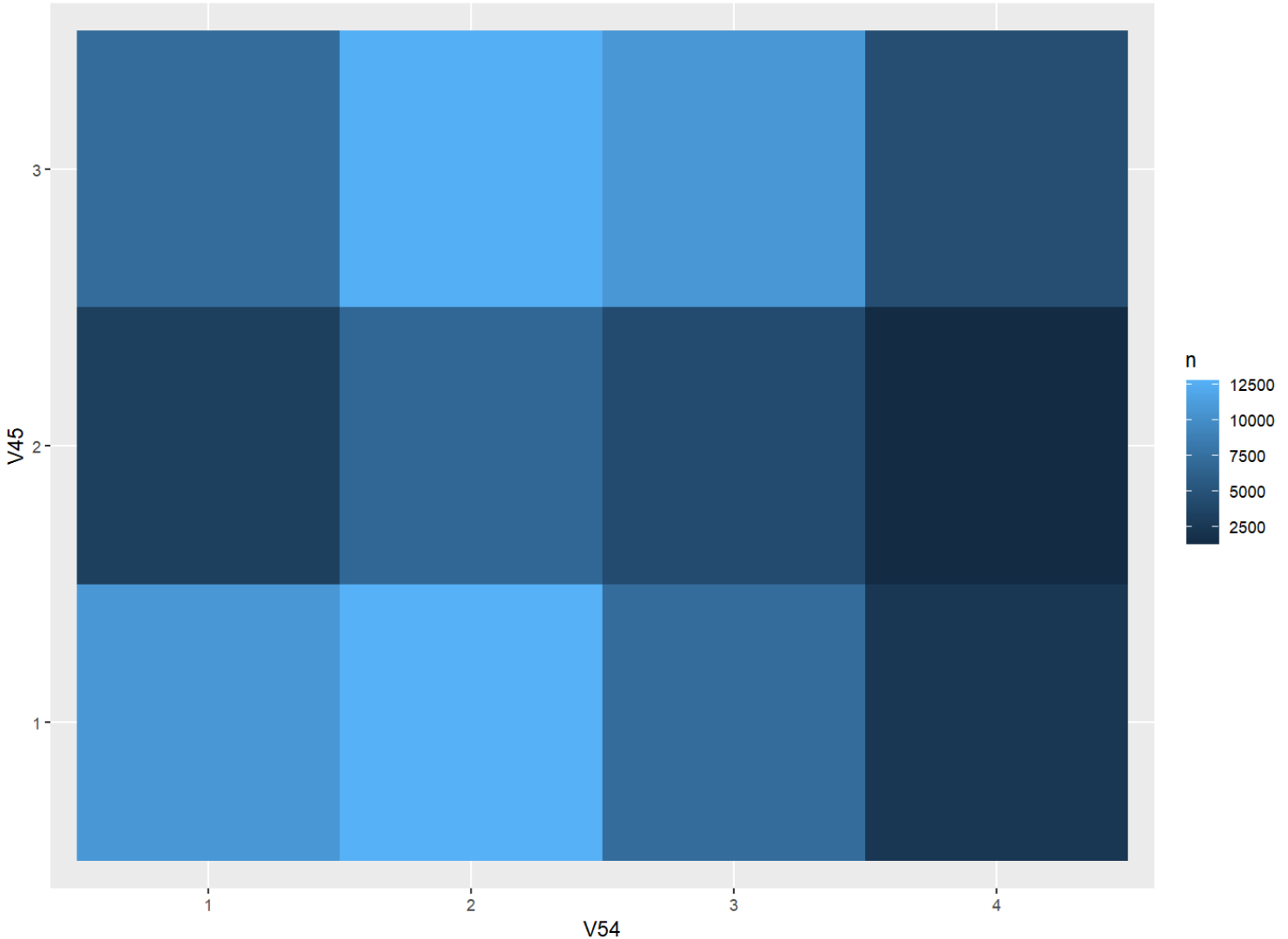
```
wvs_w6_data %>%
  drop_na(V54, V45) %>%
  ggplot() +
  geom_count(aes(x = V54, y = V45))
```


V45
3
2
1
1
2
3
4
V54
n
2500
5000
7500
10000
12500

Zusammenhang kategorialer Variablen

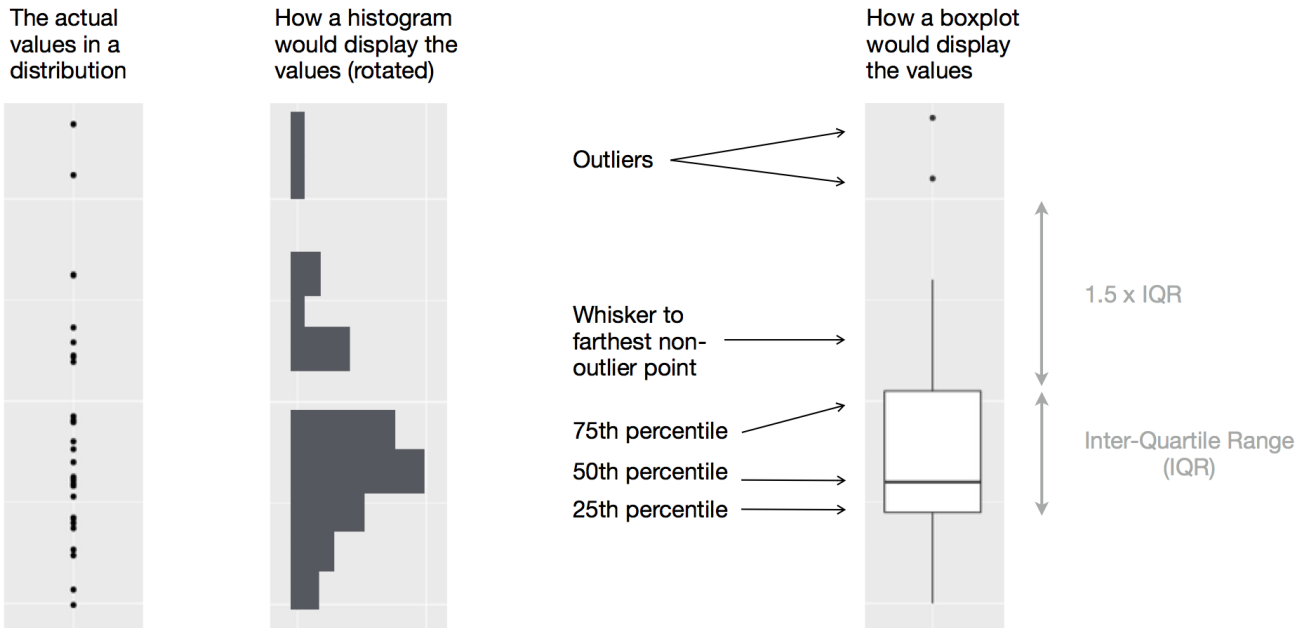
Mit der Kombination von `count()` und `geom_tile(fill = n)` erstellen wir ein Plot, bei dem wir die Häufigkeit anhand der Helligkeit der Farbe erkennen.

```
wvs_w6_data %>%
  drop_na(V54, V45) %>%
  count(V54, V45) %>%
  ggplot() +
  geom_tile(aes(x = V54, y = V45, fill = n))
```


V45
3
2
1
1
2
3
4
V54
n
12500
10000
7500
5000
2500

Zusammenhang kategorialer und kontinuierlicher Variablen

Ein Boxplot illustriert die Verteilung der Werte einer Variable.



Grafik: https://r4ds.had.co.nz/tidy-data.html

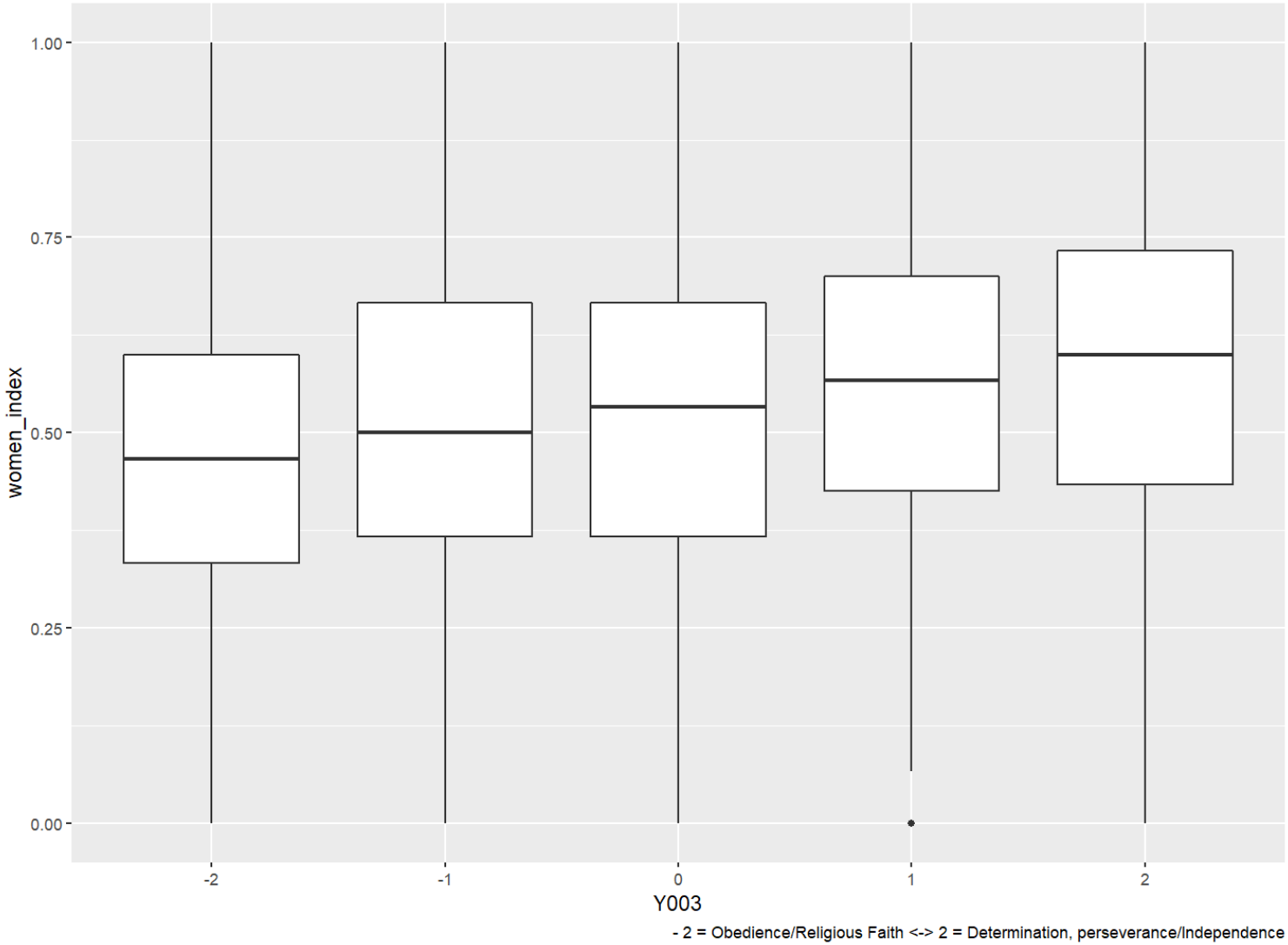
Zusammenhang kategorialer und kontinuierlicher Variablen

Eigentlich haben wir schon die Verteilung einer kontinuierlichen Variable in Beziehung zu kategorialen Variablen gesetzt. Wir haben Boxplots getrennt für Schweden und Deutschland sowie für Männer und Frauen in einem Plot erstellt.

Das gleiche können wir auch für jedes Antwortitem oder Level anderer kategorialer Variablen getrennt machen. Wir erzeugen ein Boxplot der Verteilung von `women_index` für jedes Level von `Y003` "Autonomy Index".

```
wvs_w6_data %>%
  drop_na(Y003, women_index) %>%
  ggplot() +
  geom_boxplot(aes(x = Y003, y = women_index)) +
  labs(caption = "- 2 = Obedience/Religious Faith <-> 2 = Determinat
```

Um so höher der Wert des Autonomy Index umso höher ist der mittlere Wert des women_index. Interessanterweise ist die Streuung des women_index bei allen levels des Autonomy Index ähnlich.


1.00
0.75
0.50
0.25
0.00
women_index
-2
-1
0
1
2
Y003
- 2 = Obedience/Religious Faith <-> 2 = Determination, perseverance/Independence

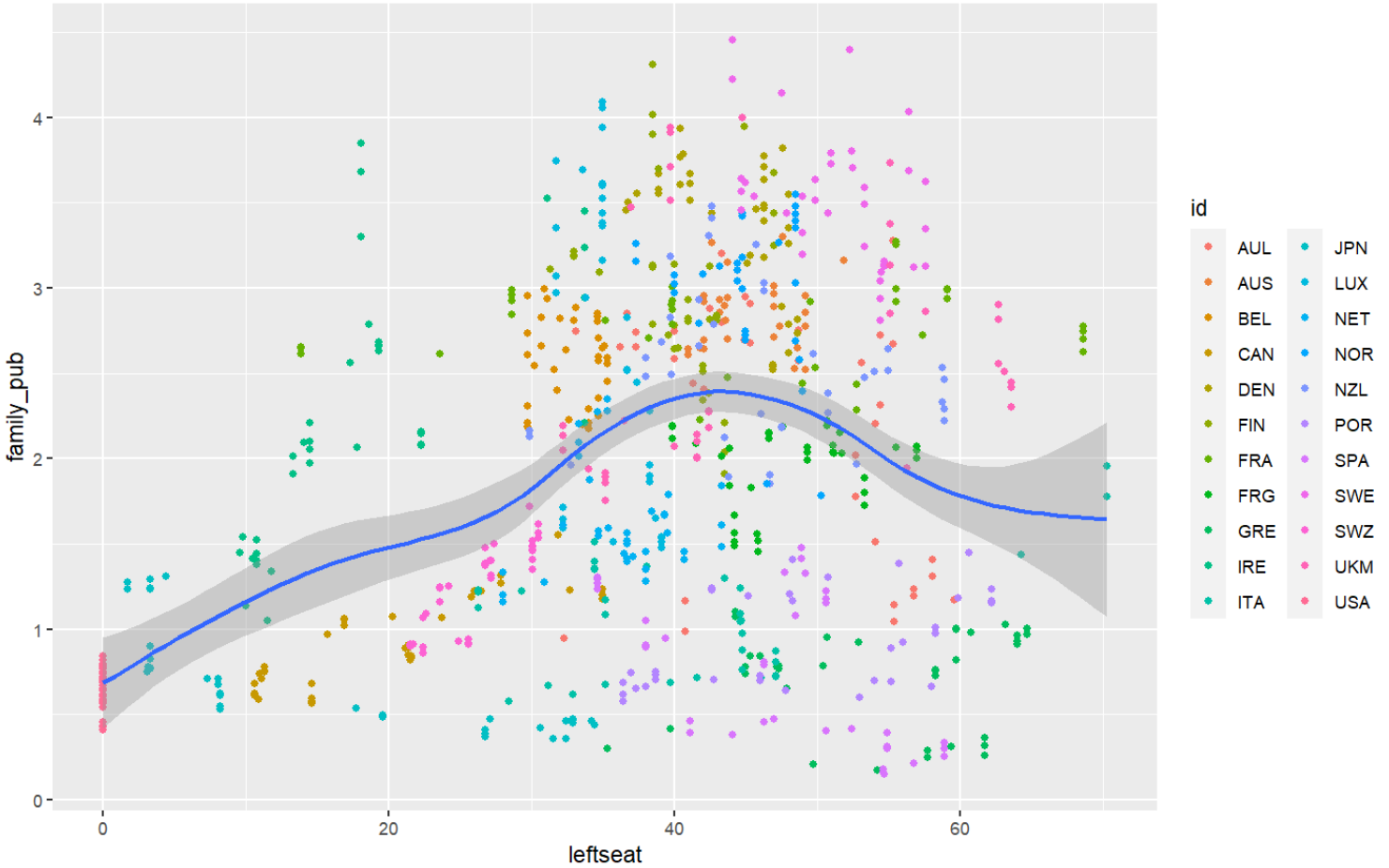
Zusammenhang kontinuierlicher Variablen

`geom_point()` erstellt ein Streudiagramm für jede Beobachtung der zwei Variablen. Zusätzlich erstellen wir mit `geom_smooth()` eine Regressionslinie der zwei Variablen.

Zusammenhang zwischen öffentlichen Ausgaben für Familienpolitik und Anteil von linken Parteien im Parlament.

```
cws_data %>%
  ggplot(aes(x = leftseat, y = family_pub)) +
  geom_point(aes(colour = id)) +
  geom_smooth()
```

```
## geom_smooth() using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
## Warning: Removed 517 rows containing non-finite values (stat_smooth).
## Warning: Removed 517 rows containing missing values (geom_point).
```



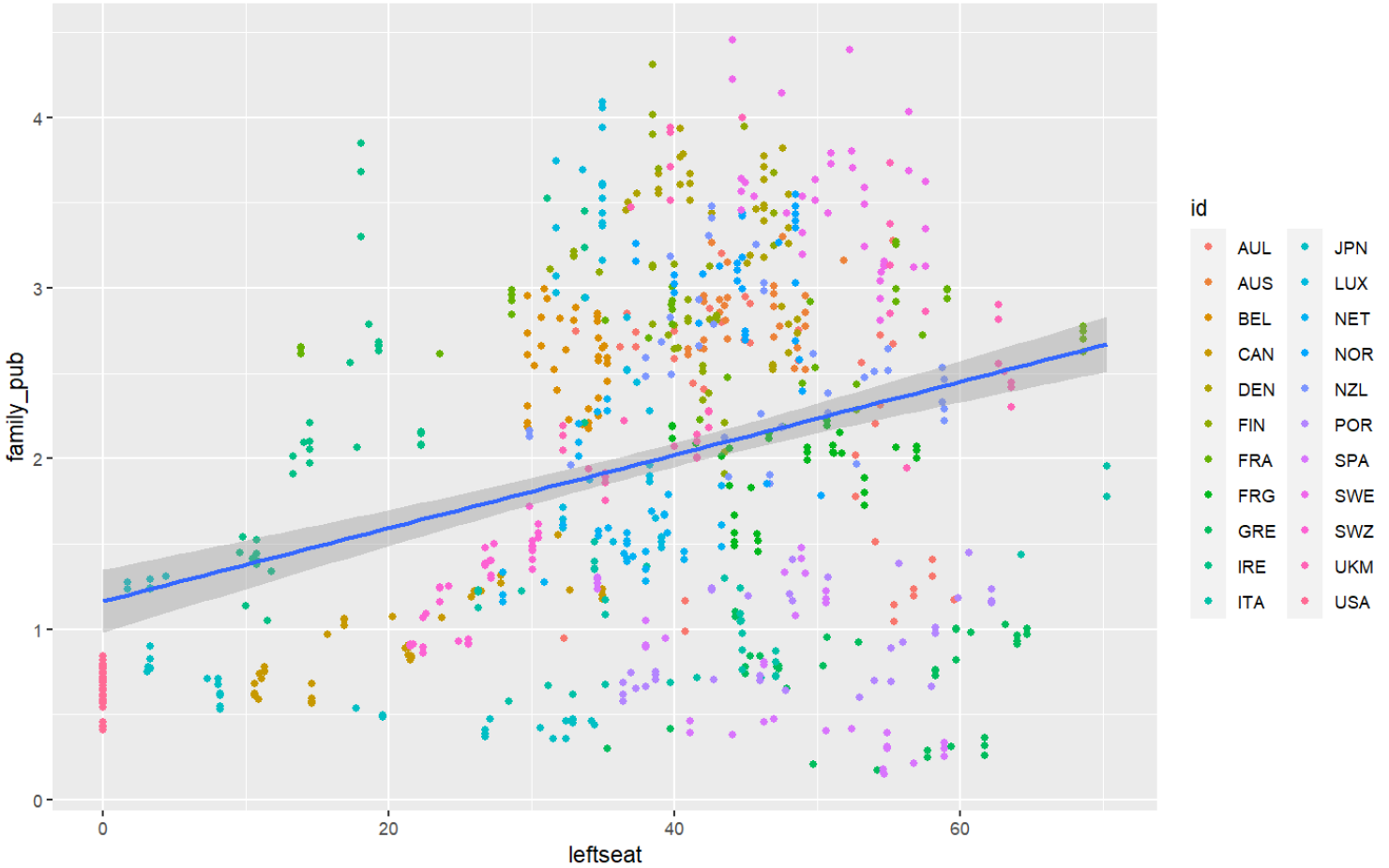
Zusammenhang kontinuierlicher Variablen

`geom_smooth()` wählt eine Regressionslinie, die am besten zu den Daten passt. Jedoch ist die Steigung der Regressionslinie schwierig zu interpretieren.

Per `geom_smooth(method = "lm")` können wir zB eine lineare Regressionslinie erstellen.

```
cws_data %>%
  ggplot(aes(x = leftseat, y = family_pub)) +
  geom_point(aes(colour = id)) +
  geom_smooth(method = "lm")
```

```
## geom_smooth() using formula 'y ~ x'
## Warning: Removed 517 rows containing non-finite values (stat_smooth).
## Warning: Removed 517 rows containing missing values (geom_point).
```



Zusammenhang kontinuierlicher Variablen

Der `cor()` Befehl hat uns schon gezeigt, dass die Ausgaben für Familienpolitik stärker mit dem Anteil von Frauen im Parlament als mit dem Anteil von linken Parteien im Parlament zusammenhängt.

```
cor(cws_data$family_pub, cws_data$leftseat, use = "complete.obs")
```

```
## [1] 0.3187759
```

```
cor(cws_data$family_pub, cws_data$fempar, use = "complete.obs")
```

```
## [1] 0.5808422
```

```
cws_data %>%
  ggplot(aes(x = fempar, y = family_pub)) +
  geom_point(aes(colour = id)) +
  geom_smooth(method = "lm")
```

```
## geom_smooth() using formula 'y ~ x'
## Warning: Removed 517 rows containing non-finite values (stat_smooth).
## Warning: Removed 517 rows containing missing values (geom_point).
```

Visualisierung vieler Datenpunkte

In den letzten Plots haben wir alle Beobachtungen verwendet und keine Rücksicht auf die unterschiedlichen Jahre genommen. Das heißt der Wert von Deutschland 1980 hat gleich viel Einfluss auf die Regressionslinie als USA 2015.

Vielleicht hat sich die Beziehung zwischen den Variablen über die Jahre verändert.

Eine Möglichkeit mit vielen Datenpunkten in einem Plot umzugehen, ist die Unterteilung anhand einer dritten Variable.

- Im WVS Datensatz könnten wir nach Länder unterteilen.
- Beim CWS Datensatz bietet es sich an nach Jahrzehnten zu unterteilen und zu schauen ob sich die Beziehung zwischen dem Anteil der Frauen im Parlament und den öffentlichen Ausgaben für Familienpolitik verändert hat.

Visualisierung vieler Datenpunkte

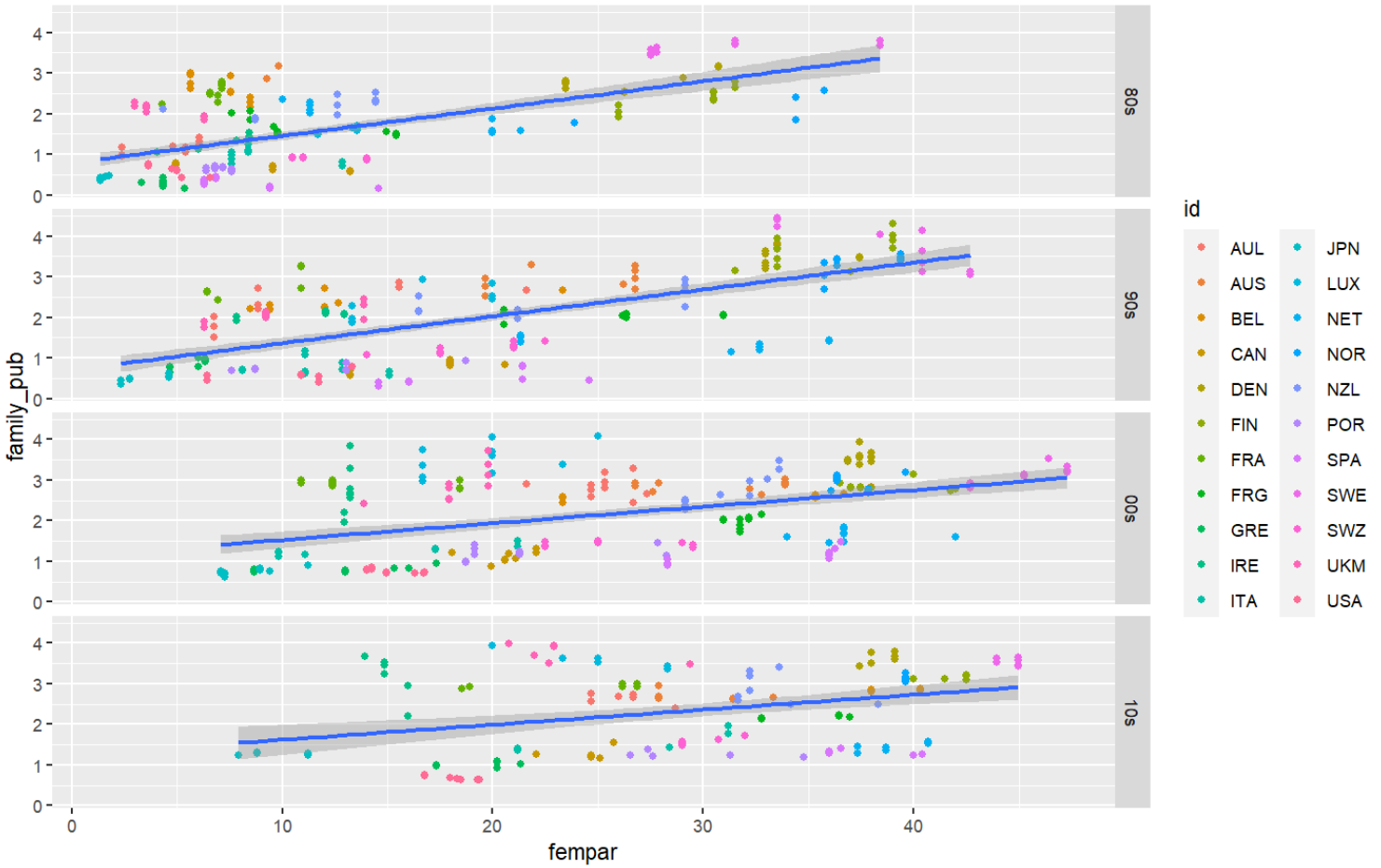
Das machen wir mit `facet_grid()`. Da keine Werte für die Zeit 1960 - 1979 existieren, lasse ich die zwei Jahrzehnte weg.

```
# create categorical decade variable
cws_data <- cws_data %>%
  mutate(year_10s = cut(year, breaks = c(1960, 1969, 1979, 1989, 199
                        labels = c("60s", "70s", "80s", "90s", "00s",
                        include.lowest = TRUE))


cws_data %>%
  filter(year_10s %in% c("80s", "90s", "00s", "10s")) %>%
  ggplot(aes(y = family_pub, x = fempar)) +
  geom_point(aes(colour = id)) +
  geom_smooth(method = "lm") +
  facet_grid(
    rows = vars(year_10s))
```

Der Zusammenhang scheint über alle Jahrzehnte ähnlich zu sein. Jedoch gibt es große Unterschiede bei der Verteilung des Anteils von Frauen im Parlament über die Jahrzehnte.

```
## geom_smooth() using formula 'y ~ x'
## Warning: Removed 76 rows containing non-finite values (stat_smooth).
## Warning: Removed 76 rows containing missing values (geom_point).
```



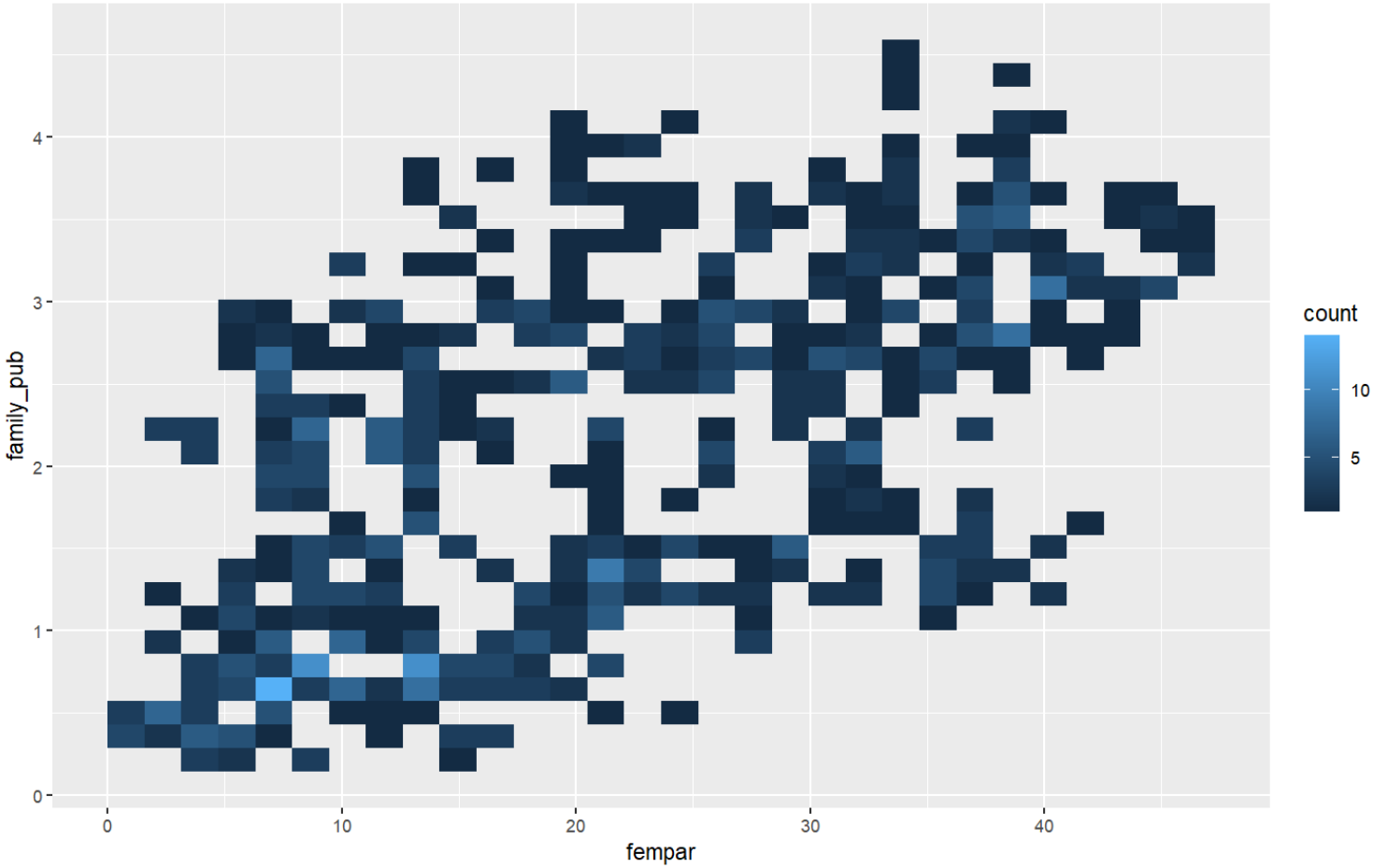
Visualisierung vieler Datenpunkte

Eine weitere Möglichkeit mit einer Vielzahl von Datenpunkten umzugehen, ist ein Heatmap zu erstellen. Mit `geom_bin2d()` erstellen wir Plot in dem wir die Anzahl von Datenpunkten anhand der Helligkeit der Farbe erkennen.

```
cws_data %>%
  ggplot(aes(y = family_pub, x = fempar)) +
  geom_bin2d()
```

`geom_bin2d()` eignet sich am besten für Datensätze mit sehr vielen Datenpunkten. Die WVS Daten wären perfekt dafür, nur haben wir dort leider nur eine kontinuierliche Variable.

```
## Warning: Removed 517 rows containing non-finite values (stat_bin2d).
```

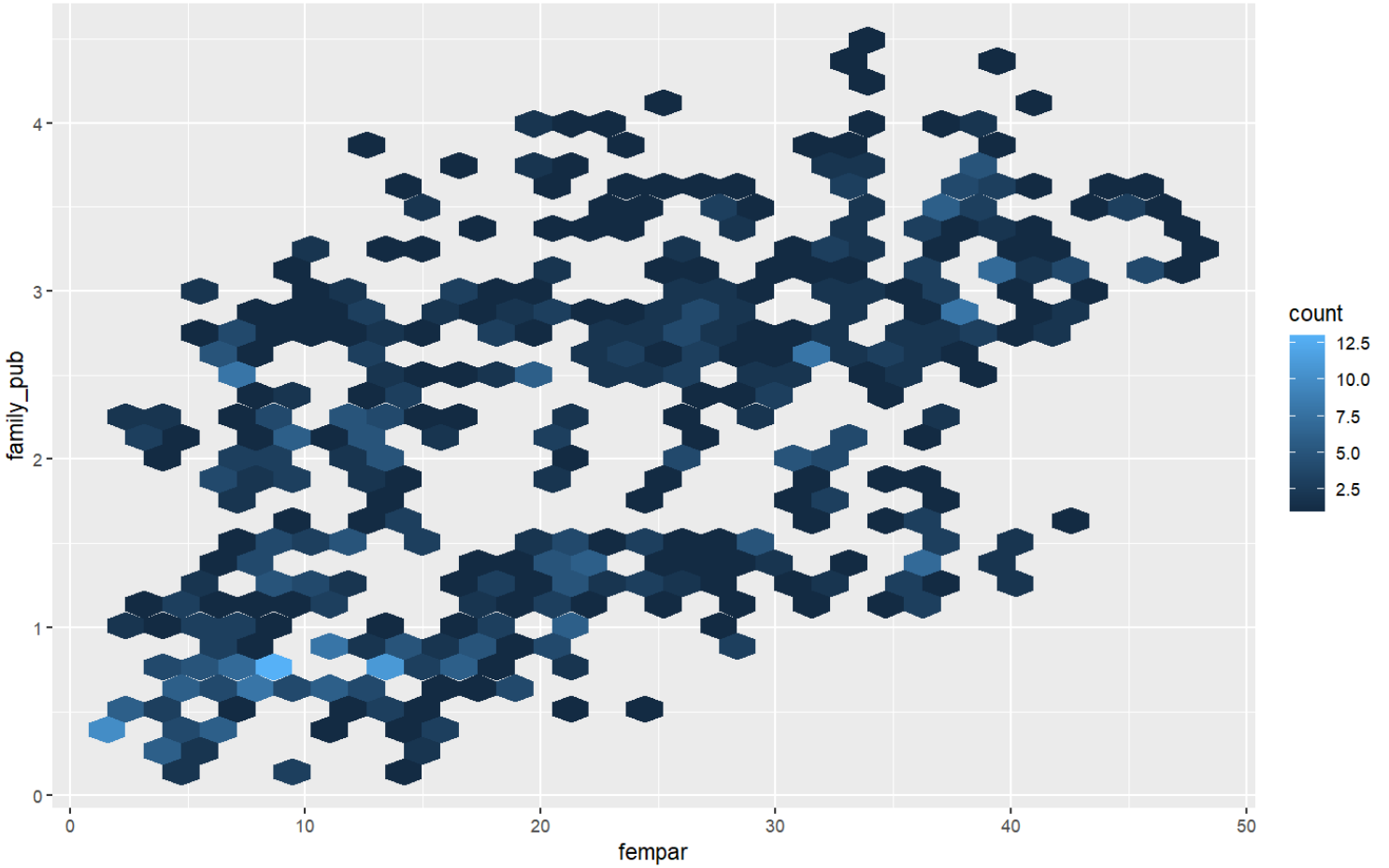


Visualisierung vieler Datenpunkte

Das gleiche geht auch mit Hexagone anstatt Quadraten. Dafür verwenden wir `geom_hex()` aus dem hexbin-Paket.

```
# install.packages("hexbin")
library(hexbin)
cws_data %>%
  ggplot(aes(y = family_pub, x = fempar)) +
  geom_hex()
```

```
## Warning: Removed 517 rows containing non-finite values (stat_binhex).
```

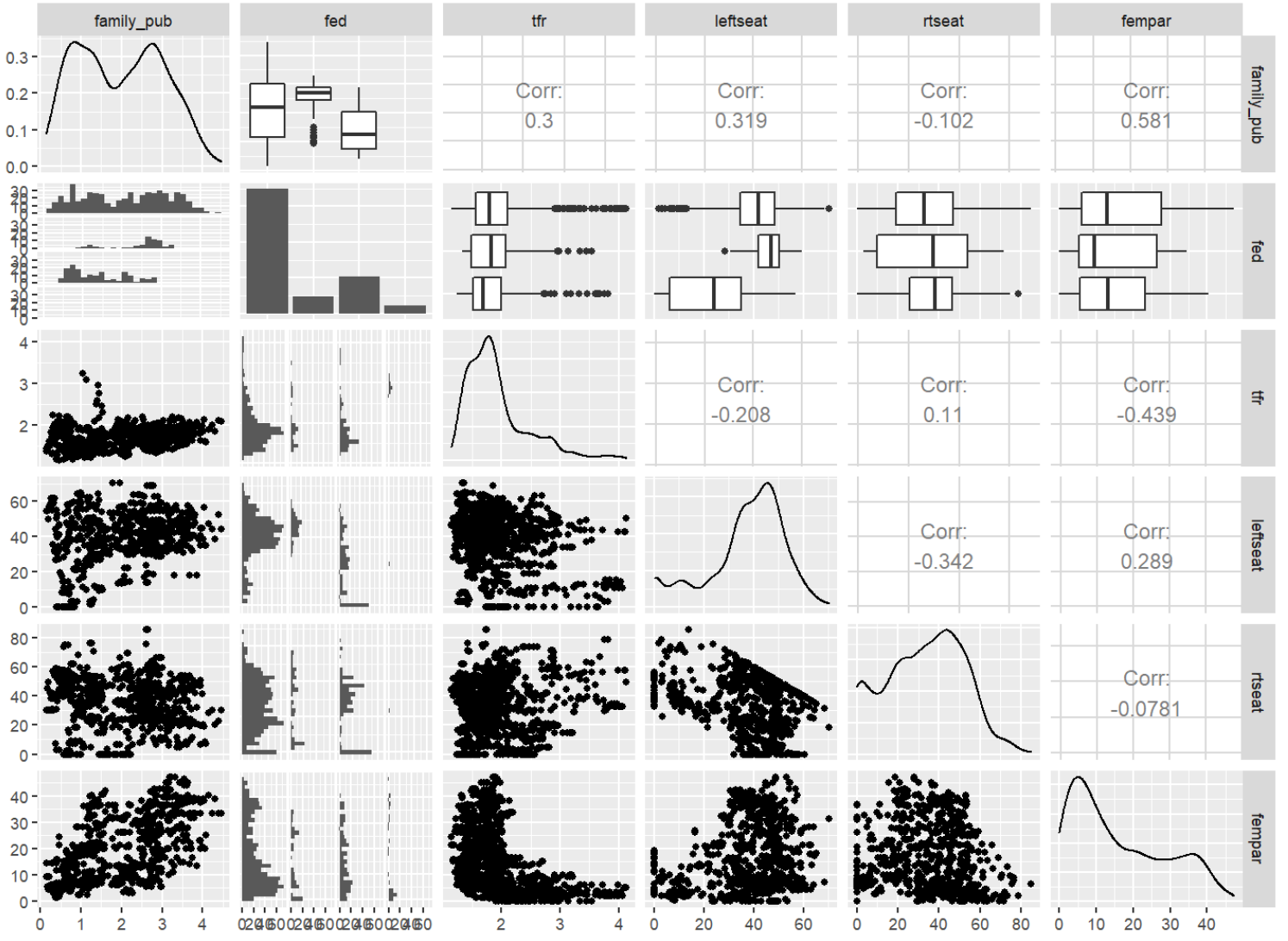


Explorative Plots

Am Anfang einer Analyse hilft es den Zusammenhang zwischen mehreren Variablen in einem Plot zu visualisieren. `ggpairs()` aus dem GGally-Paket wählt basierend auf den Variablentypen die passende `geom_functions`.

- Auf der Diagonalen wird die Verteilung der Einzelvariable visualisiert.
- Der Zusammenhang zweier Variablen wird in dem Feld visualisiert, wo sich die Reihe und Spalte der zwei Variablen trifft.
- Bei kontinuierlichen Variablen wird auch die Korrelation angezeigt.

```
# install.package("GGally")
library(GGally)
cws_data %>%
  mutate(fed = as.ordered(fed)) %>%
  ggpairs(columns = c("family_pub", "fed", "tfr", "leftseat", "rtseat
```


family_pub
fed
tfr
leftseat
rtseat
fempar
Corr:
0.3
Corr:
0.319
Corr:
-0.102
Corr:
0.581
Corr:
-0.208
Corr:
0.11
Corr:
-0.439
Corr:
-0.342
Corr:
0.289
Corr:
-0.0781

Übung 11

- Verwenden Sie entweder den wvs_short_w6 oder den CWS-data-2020 Datensatz.
- Wählen Sie bis zu 6 Variablen aus.
- Visualisieren Sie den Zusammenhang der 6 Variablen per `ggpairs()`.
- Wählen Sie anhand des Plots zwei Variablen aus.
- Visualisieren Sie den Zusammenhang der zwei Variablen mit einer passenden `geom_function`.
- Erklären Sie kurz in einem Kommentar wie die Variablen zusammenhängen.

Anhang

ggplot2 Theme Elements

theme(element_name = element_function())

- element_text()
- element_line()
- element_rect()
- element_blank()

Plot elements:

plot.background
element_rect()

plot.title
element_text()

plot.margin
margin()

Facetting elements:

strip.background
element_rect()

panel.spacing
unit()

strip.text
element_text()

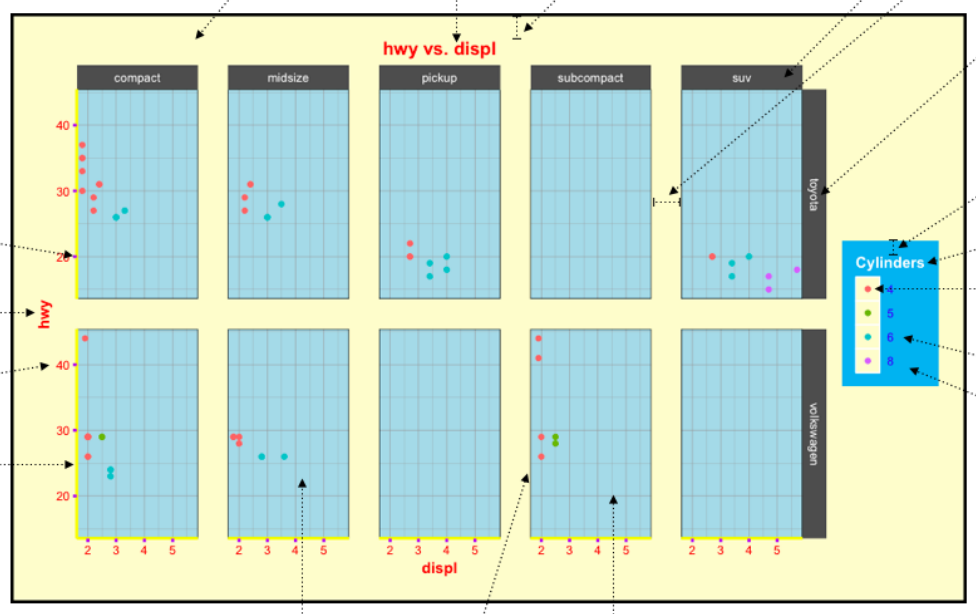
Axis elements:

axis.ticks
element_line()

axis.title
element_text()

axis.text
element_text()

axis.line
element_line()



Legend elements:

legend.margin
margin()

legend.title
element_text()

legend.key
element_rect()

legend.text
element_text()

legend.background
element_rect()

panel.background
element_rect()

panel.grid
element_line()

panel.border
element_rect(fill = NA)

Panel elements:

henrywang.nl

Derived from *"ggplot2: Elegant Graphics for Data Analysis"*