

Datenvisualisierung mit R

Übungsaufgabe vom 22.05.2025

Armin Schäfer

Einleitung

In dieser Musterlösung zeige ich, wie die Grafik erstellt wurde. Der Code ist nicht ideal, sollte aber dem Stand Ihres Wissens entsprechen. Später im Semester werde ich eine weitere Version hochladen, die zwar anspruchsvoller, aber weniger fehleranfällig ist. Die Annotationen im Code sind mit ChatGPT erstellt und dann von mir angepasst worden.

Packages (gegebenfalls) installieren und anschließend laden

Einige von Ihnen haben einen anderen Weg gewählt, die Packages zu öffnen. Dies ist der von mir empfohlene Weg, um Ihr Skript reproduzierbar zu halten.

```
if(!require("pacman")) {install.packages("pacman");library(pacman)}  
p_load(here, scales, rio, tidyverse)  
  
# Daten einlesen mit dem RIO-package und dem "import"-Befehl  
  
ces <- import(here("data", "ces_usa.csv"))
```

Voraussetzungen, um ein PDF erstellen zu können

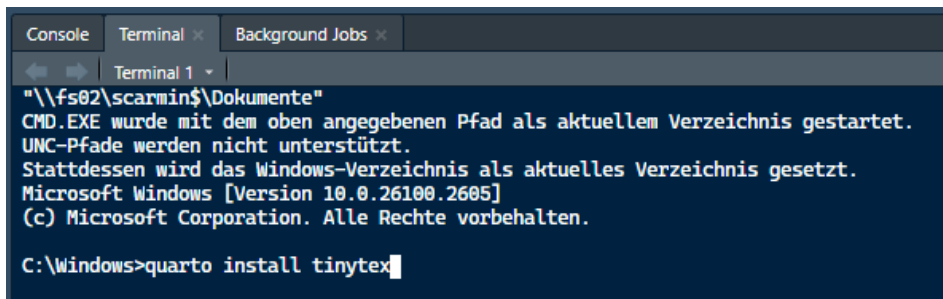
Als Sie versucht haben, ein PDF-Dokument mit Quarto zu erstellen, haben Sie womöglich die folgende Fehlermeldung erhalten. Um PDFs zu erzeugen, muss ein Hilfsprogramm vorhanden sein, dass Latex-Dokumente lesen und in PDFs umwandeln kann (das geschieht im Hintergrund).

```
Rendering PDF
running xelatex - 1

No TeX installation was detected.

Please run 'quarto install tinytex' to install TinyTex.
If you prefer, you may install TeXLive or another TeX distribution.
```

Um das Problem zu beheben, müssen Sie folgende Schritte durchführen: Starten Sie R *ohne ein Projekt oder ein Skript zu öffnen*. Gehen Sie zum Feld “**Terminal**” neben dem Reiter für die Console. Tippen Sie dort “**quarto install tinytex**” ein. Nach der Installation sollte es möglich sein, ein Quarto-Dokument als PDF ausgeben zu lassen.



```
Console Terminal × Background Jobs ×
Terminal 1
"\\fs02\scarmin$\Dokumente"
CMD.EXE wurde mit dem oben angegebenen Pfad als aktuellem Verzeichnis gestartet.
UNC-Pfade werden nicht unterstützt.
Stattdessen wird das Windows-Verzeichnis als aktuelles Verzeichnis gesetzt.
Microsoft Windows [Version 10.0.26100.2605]
(c) Microsoft Corporation. Alle Rechte vorbehalten.

C:\Windows>quarto install tinytex
```

Die Daten vorbereiten

```
ces <- ces %>%

# Creating new columns 'ideo_ego_fac' and 'ideo_scotus_fac'
# based on conditions using case_when()
mutate(
  ideo_ego_fac = case_when(
    CC22_340a == 1 ~ "very liberal",
    CC22_340a == 2 ~ "liberal",
    CC22_340a == 3 ~ "somewhat liberal",
    CC22_340a == 4 ~ "middle of the road",
    CC22_340a == 5 ~ "somewhat conservative",
    CC22_340a == 6 ~ "conservative",
    CC22_340a == 7 ~ "very conservative"),
  ideo_scotus_fac = case_when(
    CC22_340g == 1 ~ "very liberal",
    CC22_340g == 2 ~ "liberal",
    CC22_340g == 3 ~ "somewhat liberal",
    CC22_340g == 4 ~ "middle of the road",
```

```

CC22_340g == 5 ~ "somewhat conservative",
CC22_340g == 6 ~ "conservative",
CC22_340g == 7 ~ "very conservative")) %>%

# Filtering out rows where either 'ideo_ego_fac'
# or 'ideo_scotus_fac' is NA
filter(!is.na(ideo_ego_fac), !is.na(ideo_scotus_fac)) %>%

# Grouping the data by 'ideo_ego_fac' and 'ideo_scotus_fac' columns
group_by(ideo_ego_fac, ideo_scotus_fac) %>%

# Summarizing the number of observations in each group
summarize(N = n()) %>%

# Calculating frequency and percentage based on the
# number of observations in each group
mutate(
  freq = N / sum(N),
  pct = round((freq * 100), 2)
)

```

Grafik erstellen

```

# Create a ggplot with specified aesthetics
ggplot(data = ces, aes(
  x = fct_relevel(ideo_ego_fac,
    "very liberal", "liberal", "somewhat liberal",
    "middle of the road", "somewhat conservative",
    "conservative", "very conservative"),
  y = pct,
  fill = fct_relevel(ideo_scotus_fac,
    "very liberal", "liberal", "somewhat liberal",
    "middle of the road", "somewhat conservative",
    "conservative", "very conservative"))) +

# Add a column chart layer
geom_col() +

# Use viridis color scale with reversed direction
scale_fill_viridis_d(direction = -1) +

```

```

# Set y-axis breaks at intervals of 10
scale_y_continuous(breaks = seq(0, 100, by = 10)) +

# Flip the coordinates to create a horizontal bar chart
coord_flip() +

# Customize plot labels
labs(
  fill = NULL, # Remove fill legend title
  x = "Ideological self-placement", # X-axis label
  y = "Perceived position SCOTUS") + # Y-axis label

# Use a minimal theme
theme_minimal() +

# Customize additional theme elements
theme(
  legend.position = "top", # Move legend to the top
  axis.title = element_text(face = "bold"), # Bold axis titles
  legend.text = element_text(size = 6), # Set legend text size
  legend.key.size = unit(.3, 'cm'), # Set legend key size
  plot.title = element_text(size = 10, face = "bold")) +

# Reverse the order of items in the legend
guides(fill = guide_legend(reverse = TRUE))

```

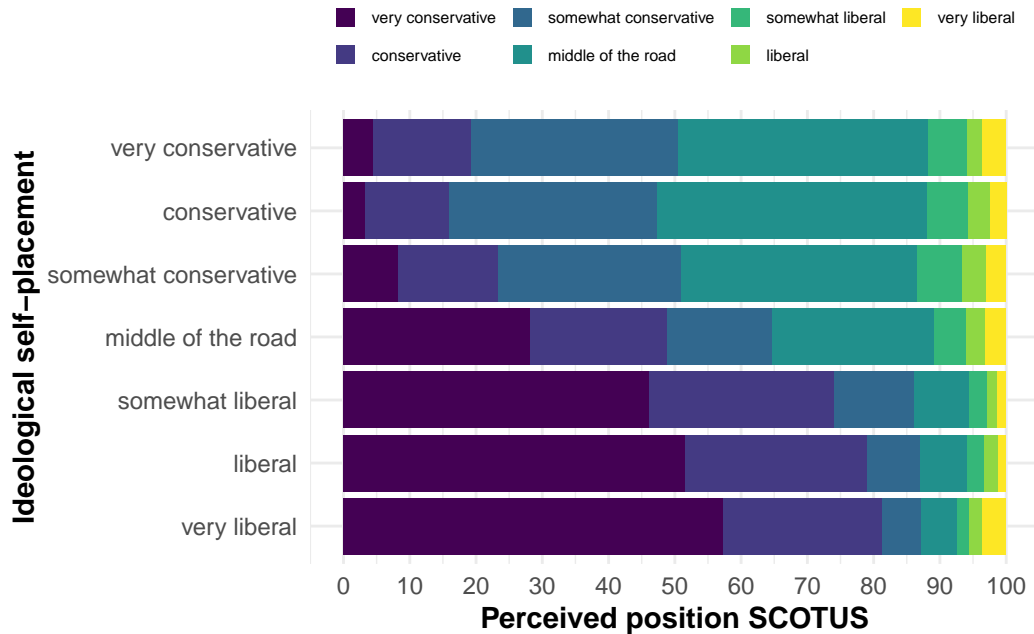


Figure 1: Die Sicht der US-Bürger:innen auf den Supreme Court.

Erklärung (ChatGPT)

`ggplot` initializes the plot with specified aesthetics.

`geom_col()` adds a column chart layer to the plot.

`scale_fill_viridis_d(direction = -1)` sets the fill color scale with a reversed direction.

`scale_y_continuous(breaks = seq(0, 100, by = 10))` sets y-axis breaks at intervals of 10.

`coord_flip()` flips the coordinates to create a horizontal bar chart.

`labs()` sets plot labels, and `fill = NULL` removes the fill legend title.

`theme_minimal()` applies a minimal theme to the plot.

`theme()` is used to further customize various theme elements such as legend position, axis titles, legend text size, legend key size, and plot title.

`guides(fill = guide_legend(reverse = TRUE))` reverses the order of items in the legend.