

Datenanalyse mit R

14 Kommunikation von Ergebnissen 2

Tobias Wiß, Carmen Walenta und Felix Wohlgemuth

19.06.2020



Institut für
Gesellschafts-
und Sozialpolitik

Vielen Dank für Ihre rege Teilnahme am R Kurs dieses Semester! Es hat mich gefreut, dass Sie das moodle-Forum und auch die R Sprechstunde genutzt haben. Danke für die spannenden Diskussionen und hoffentlich konnte ich Ihnen bei Ihren Fragen weiterhelfen.

In der heutigen Einheit wiederholen wir nochmal wie Ergebnisse aus R in Word berichtet werden und zusätzlich werden wir noch kurz in R Markdown einsteigen. Diese Woche gibt es keine Übungsaufgabe und auch keine R Sprechstunde.

Daten für diese Woche

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/>

Sie finden die Daten und das Codebook natürlich auch auf moodle.

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

# optional variable preparation
cws_data <- cws_data %>%
  mutate(fem_employment = 100 * (flabfo / tlabfo))
```

Wiederholung

Kommunikation von Ergebnissen mit
stargazer()

R und Word

Um die Ergebnisse von R mit einer Tabelle in Word zu präsentieren, kann mit `stargazer()` eine html-Tabelle erstellt werden. In einem Browser kann dann die ganze Tabelle markiert, kopiert und dann in Word eingefügt werden. `stargazer()` speichert die exportierte html-Tabelle in der vorher festgelegten Working Directory.

Das Layout und Aussehen der Tabelle kann einfach in Word bearbeitet werden. Der Vorteil hierbei ist, dass Änderungen an der Tabelle direkt gesehen werden und dass die Tabelle an den Stil des Textes angepasst werden kann. Der Nachteil ist, dass bei kleinen Änderungen an den Daten der ganze Anpassungsprozess in Word wiederholt werden muss.

Das Aussehen und Layout der html-Tabelle kann auch gleich in `stargazer()` festgelegt werden. Der Anpassungsprozess ist somit einfacher wiederhol- und nachvollziehbar. Der letzte Feinschliff muss dann in Word gemacht werden.

Tabelle Deskriptive Statistik

Wenn in `stargazer()` ein Dataframe ohne weitere Spezifikationen als Input angegeben wird, erstellt `stargazer()` eine Tabelle mit deskriptiven Statistikmaßen für alle Variablen im Dataframe. Deshalb muss der Dataframe nur die Variablen enthalten, die man auch im Bericht darstellen möchte.

stargazer() erwartet ein Dataframe als Input, deshalb kann vorher mit `str()` geprüft werden, ob das Inputobjekt ein Dataframe ist und gegebenenfalls umgewandelt werden. Da wir mit Funktionen aus dem tidyverse gearbeitet haben, ist das Datenobjekt ein tibble und kein klassischer Dataframe.

```
library(stargazer)
cws_data_90 <- cws_data %>%
  filter(year == 1990) %>%
  select(family_pub, socx_pub, fempar, rtcrseat, leftseat, ud, fem_er)
str(cws_data_90)
```

```
## tibble [22 x 8] (S3: tbl_df/tbl/data.frame)
## $ family_pub      : num [1:22] 1.51 2.53 2.211 0.578 3.193 ...
## $ socx_pub        : num [1:22] 13.1 23.1 24.4 17.5 21.9 ...
## $ fempar          : num [1:22] 6.76 19.67 8.49 13.22 32.96 ...
## $ rtcrseat        : num [1:22] 0 0 0 0 0 2.5 0 45.2 0 0 ...
```

Tabelle Deskriptive Statistik

```
cws_data_90 <- as.data.frame(cws_data_90)

stargazer(cws_data_90,
  digits = 2,
  decimal.mark = ",",
  title = "Deskriptive Statistik",
  covariate.labels = c(
    "Öffentliche Ausgaben für Familienpolitik <i>(% BIP)</i>",
    "Öffentliche Sozialausgaben insgesamt <i>(% BIP)</i>",
    "Anteil weibliche Parlamentsabgeordnete",
    "Anteil Abgeordnete rechter christlicher Parteien",
    "Anteil Abgeordnete linker Parteien",
    "Anteil Gewerkschaftsmitglieder <i>(% Angestellten)</i>",
    "Anteil Frauen an Erwerbstätigen",
    "Anteil Teilzeibesetzten an weiblichen Erwerbstätigen"),
  type = "html",
  out= "descriptives_90.html")
```

In der Konsole wird der Code der html-Tabelle ausgegeben. In einem html-Dokument, wie diesen Folien, kann die Tabelle mit dem Code direkt angezeigt werden. Für Word müssen Sie die exportierte html-Datei mit einem Browser öffnen und in Word kopieren.

Deskriptive Statistik

Statistic	N	Mean	St. Dev.	Min	Pctl(25)	Pctl(75)	Max
öffentliche Ausgaben für Familienpolitik (% <i>BIP</i>)	22	1,73	1,04	0,31	0,73	2,49	4,03
öffentliche Sozialausgaben insgesamt (% <i>BIP</i>)	22	18,92	4,75	10,93	15,11	22,85	27,24
Anteil weibliche Parlamentsabgeordnete	22	15,54	10,63	2,34	7,10	20,32	38,40
Anteil Abgeordnete rechter christlicher Parteien	22	2,17	9,63	0	0	0	45
Anteil Abgeordnete linker Parteien	22	38,96	15,06	0,00	34,30	48,48	56,40
Anteil Gewerkschaftsmitglieder (% <i>Angestellten</i>)	22	40,69	19,59	9,83	26,34	50,73	81,50
Anteil Frauen an Erwerbstätigen	20	42,06	3,84	34,97	40,05	44,57	47,99
Anteil Teilzeibeschäftigten an weiblichen Erwerbstätigen	18	25,49	11,27	10,62	18,54	29,76	52,51

Regressionstabelle

Wenn ein Regressionsmodell als Input in `stargazer()` angegeben wird, erstellt `stargazer()` eine Regressionstabelle. Wie bei der Tabelle mit deskriptiven Statistikmaßen, kann der Inhalt und das Aussehen der Tabelle direkt in `stargazer()` verändert werden. Falls das nicht gemacht wird, verwendet `stargazer()` einfach die Variablennamen. Die Regressionskoeffizienten und die Parameter der Modellgüte werden dabei nicht verändert.

```
lm_90 <- lm(family_pub ~ socx_pub + fempar + rtcseat + leftseat + u  
stargazer(lm_90,  
  column.labels = c("Jahr = 1990"),  
  dep.var.labels = c("Öffentliche Ausgaben für Familienpolitik <i>"),  
  covariate.labels = c("Öffentliche Sozialausgaben insgesamt <i>(%  
    "Anteil weibliche Parlamentsabgeordnete",  
    "Anteil Abgeordnete rechter christlicher Parteien",  
    "Anteil Abgeordnete linker Parteien",  
    "Anteil Gewerkschaftsmitglieder <i>(% Angestellten)</i>",  
    "Anteil Frauen an Erwerbstätigen",  
    "Anteil Teilzeibesäftigten an weiblichen Erwerbstätigen",  
    "Konstante"),  
  no.space = TRUE, type = "html", out = "lm_90_table.html")
```

	<i>Dependent variable:</i>
	öffentliche Ausgaben für Familienpolitik (% BIP)
	Jahr = 1990
öffentliche Sozialausgaben insgesamt (% BIP)	0.102** (0.041)
Anteil weibliche Parlamentsabgeordnete	-0.018 (0.022)
Anteil Abgeordnete rechter christlicher Parteien	0.007 (0.013)
Anteil Abgeordnete linker Parteien	0.012 (0.013)
Anteil Gewerkschaftsmitglieder (% Angestellten)	0.031** (0.011)
Anteil Frauen an Erwerbstätigen	0.074 (0.043)
Anteil Teilzeibesetzten an weiblichen Erwerbstätigen	0.012 (0.013)
Konstante	-5.163** (1.744)
Observations	17
R ²	0.870
Adjusted R ²	0.769
Residual Std. Error	0.536 (df = 9)
F Statistic	8.609*** (df = 7; 9)
Note:	*p<0.1; **p<0.05; ***p<0.01

Regressionstabelle

`stargazer()` kann auch die Ergebnisse mehrerer Regressionsmodelle in einer Tabelle gemeinsam anzeigen. Damit können Modelle verglichen werden. Dazu werden einfach als Input mehrere Regressionmodelle angegeben. Die Reihenfolge im Input legt die Reihenfolge in der Tabelle fest.

```
lm_00 <- cws_data %>% filter(year == 2000) %>% lm(family_pub ~ socx_f  
lm_10 <- cws_data %>% filter(year == 2010) %>% lm(family_pub ~ socx_f  
  
stargazer(lm_90, lm_00, lm_10,  
  column.labels = c("Jahr = 1990", "Jahr = 2000", "Jahr = 2010"),  
  dep.var.labels = c("öffentliche Ausgaben für Familienpolitik <i>(</i>  
  covariate.labels = c("öffentliche Sozialausgaben insgesamt <i>(</i>(  
    "Anteil weibliche Parlamentsabgeordnete",  
    "Anteil Abgeordnete rechter christlicher Parteien",  
    "Anteil Abgeordnete linker Parteien",  
    "Anteil Gewerkschaftsmitglieder <i>(</i>(Angestellten)</i>",  
    "Anteil Frauen an Erwerbstätigen",  
    "Anteil Teilzeibesäftigten an weiblichen Erwerbstätigen",  
    "Konstante"),  
  no.space = TRUE, type = "html", out = "lm_table_2.html")
```

	<i>Dependent variable:</i>		
	öffentliche Ausgaben für Familienpolitik (% BIP)		
	Jahr = 1990 (1)	Jahr = 2000 (2)	Jahr = 2010 (3)
öffentliche Sozialausgaben insgesamt (% BIP)	0.102** (0.041)	0.089 (0.054)	0.100 (0.057)
Anteil weibliche Parlamentsabgeordnete	-0.018 (0.022)	-0.016 (0.025)	-0.064* (0.032)
Anteil Abgeordnete rechter christlicher Parteien	0.007 (0.013)	-0.011 (0.023)	-0.015 (0.024)
Anteil Abgeordnete linker Parteien	0.012 (0.013)	0.009 (0.014)	0.022 (0.019)
Anteil Gewerkschaftsmitglieder (% Angestellten)	0.031** (0.011)	0.022 (0.012)	0.041** (0.016)
Anteil Frauen an Erwerbstätigen	0.074 (0.043)	0.071 (0.077)	0.271** (0.114)
Anteil Teilzeibesetzten an weiblichen Erwerbstätigen	0.012 (0.013)	0.028 (0.018)	0.060** (0.024)
Konstante	-5.163** (1.744)	-4.443 (3.055)	-14.429** (5.507)
Observations	17	19	21
R ²	0.870	0.610	0.580
Adjusted R ²	0.769	0.362	0.354
Residual Std. Error	0.536 (df = 9)	0.744 (df = 11)	0.835 (df = 13)
F Statistic	8.609*** (df = 7; 9)	2.459* (df = 7; 11)	2.565* (df = 7; 13)
Note:	* p<0.1; ** p<0.05; *** p<0.01		

Kommunikation von Ergebnissen 2

ggplot2 und Word

Visualisierungen, die mit `ggplot()` erstellt werden, werden im "Plots" Fenster unten rechts angezeigt. Dort kann der Plot mit *Export > Save as Image* extern abgespeichert und dann einfach als Grafik in Word eingefügt werden. Die Größe kann in Word verändert werden, aber das Seitenverhältnis und die Auflösung sollte schon beim Export richtig definiert werden.

Zum Beispiel kann das Ergebnis dreier linearen Regressionsmodelle in einem Plot dargestellt werden. Zuerst erstellen wir die Regressionstabelle und dann auf Basis der drei Modelle die Visualisierung der Datenpunkten, Residuen und prognostizierten Werte.

```
lm_90_fempar <- cws_data %>% filter(year == 1990) %>% lm(family_pub ~  
lm_00_fempar <- cws_data %>% filter(year == 2000) %>% lm(family_pub ~  
lm_10_fempar <- cws_data %>% filter(year == 2010) %>% lm(family_pub ~  
  
stargazer(lm_90_fempar, lm_00_fempar, lm_10_fempar,  
  column.labels = c("Jahr = 1990", "Jahr = 2000", "Jahr = 2010"),  
  dep.var.labels = c("Öffentliche Ausgaben für Familienpolitik <i>in %</i>"),  
  covariate.labels = c("Anteil weibliche Parlamentsabgeordnete", "Jahr"),  
  type = "html", out = "lm_fempar.html")
```

	<i>Dependent variable:</i>		
	öffentliche Ausgaben für Familienpolitik (% BIP)		
	Jahr = 1990 (1)	Jahr = 2000 (2)	Jahr = 2010 (3)
Anteil weibliche Parlamentsabgeordnete	0.072*** (0.015)	0.044** (0.017)	0.028 (0.024)
Konstante	0.611** (0.276)	1.018** (0.434)	1.643** (0.697)
Observations	22	22	22
R ²	0.544	0.243	0.066
Adjusted R ²	0.521	0.205	0.019
Residual Std. Error (df = 20)	0.720	0.835	1.070
F Statistic (df = 1; 20)	23.858***	6.429**	1.409
<i>Note:</i>		* p<0.1; ** p<0.05; *** p<0.01	

ggplot2 und Word

Für die Visualisierung müssen wir zuerst mit `augment()` die Residuen und prognostizierten Werte berechnen. Dann bilden wir ein gemeinsames Dataframe mit `bind_rows()`, das der Input für `ggplot()` ist. Mit `facet_wrap()` werden drei einzelne Plots in einer Visualisierung erstellt.

```
library(broom)
lm_90_aug <- augment(lm_90_fempar) %>% mutate(year = 1990)
lm_00_aug <- augment(lm_00_fempar) %>% mutate(year = 2000)
lm_10_aug <- augment(lm_10_fempar) %>% mutate(year = 2010)
lm_aug <- bind_rows(lm_90_aug, lm_00_aug, lm_10_aug)

ggplot(lm_aug , aes(x=fempar, y=family_pub)) +
  geom_segment(aes(xend = fempar, y = family_pub, yend = .fitted), col = "red") +
  geom_point() +
  geom_line(aes(x=fempar, y=.fitted)) +
  facet_wrap( ~ year, nrow = 3, ncol =1) +
  labs(title = "Lineare Regression",
        x = "Anteil weibliche Parlamentsabgeordnete",
        y = "Öffentliche Ausgaben für Familienpolitik (% BIP)")
```


ggplot2 und Word

Die Visualisierung erscheint unten rechts im "Plots" Fenster. Die Grafik kann nun extern gespeichert werden mit Klick auf *Export > Save as Image*. Hier kann auch die Größe in Pixel festgelegt werden. Welche Größe für das Word-Dokument passend ist, muss ausprobiert werden. Die aktuelle Grafik speichern wir als DIN A5 mit 72 ppi ab (420 x 595) (Tabelle mit Größen in Pixeln: <https://www.din-formate.de/reihe-a-din-groessen-mm-pixel-dpi.html>).

Alternativ kann im R Skript `ggsave()` genutzt werden. Der Vorteil ist, dass in `ggsave()` mit `units = c("in", "cm", "mm")` die Einheit und mit `dpi =` die Auflösung festgelegt werden kann. *Wenn in `ggsave()` kein Inputobjekt angegeben wird, dann wird der letzte Plot abgespeichert.*

```
ggsave("figures/plot_lm_fempar.png",  
       width = 148 , height = 210,  
       units = "mm", dpi = 300)
```

Die Grafikdatei kann dann einfach mit Drag and Drop in Word hineingezogen oder mit *Einfügen > Bilder > Dieses Gerät* in das Word-Dokument eingefügt werden. Falls die Größe nicht richtig passt, muss die Höhe und Breite in `ggsave()` geändert und die Funktion wiederholt werden.

R Markdown

kurze Einführung

R Markdown

Um die Ergebnisse eines Projektes in Word zu berichten, müssen wir leider eine paar Zwischenschritte machen (Tabellen in html abspeichern; Plots extern speichern und in Word einfügen). RStudio bietet mit R Markdown eine gute Alternative die Ergebnisse und den Bericht in einem Dokument zu erstellen.

R Markdown empfiehlt sich für kurze Projektberichte, insbesondere wenn der R Code angezeigt werden soll. Für Dokumente mit mehr Text als Tabellen / Grafiken ist Word oder Latex einfacher. Alle Folien des R Kurses sowie die Lösungen für die Übungen wurden mit R Markdown erstellt.

R Markdown

erste Schritte

Für ein neues R Markdown Dokument klicken Sie in RStudio auf *File > New File > R Markdown*. Im Fenster kann dann der Titel und der Autor_innennamen festgelegt werden. Als Output kann html, pdf oder Word gewählt werden:

- html funktioniert ohne vorherige Anpassungen und ist auch der Standard.
- Für ein pdf-Dokument muss entweder Latex installiert sein oder das Paket *TinyTex* in R. *TinyTex* muss nur einmal installiert und nicht extra geladen werden. Mehr Infos: <https://yihui.org/tinytex/>

```
install.packages('tinytex')  
tinytex::install_tinytex()
```

```
# to uninstall TinyTeX, run tinytex::uninstall_tinytex()
```

R Markdown

Es öffnet sich dann ein Beispieldokument, das schon mal grob zeigt wie ein Markdown Skript funktioniert.

Am Anfang des Dokumentes befinden sich die Metadaten des Dokuments (YAML);

```
---  
title: "Bericht"  
author: "Felix Wohlgemuth"  
date: "19 6 2020"  
output: pdf_document  
---
```

In diesem Fall wird ein pdf Dokument erstellt. Der output kann aber auch ein html-Dokument sein. Meine Lösungen der Übungen waren zB einfache html-Dokumente.

Markdown Sprache

Markdown ist eine vereinfachte Auszeichnungssprache mit eigenen Regeln. Das Aussehen des Texts wird in Word mit Einstellungen im Menü bearbeitet. In Markdown wird das Aussehen direkt im Text festgelegt.

Überschriften werden mit # festgelegt

```
# Ebene 1
## Ebene 2
### Ebene 3
...
...
```

Ebene 1

Ebene 2

Ebene 3

Markdown Sprache

Text

fett wird mit `**TEXT**` festgelegt

kursiv wird mit `*TEXT*` festgelegt

~~durchgestrichen~~ wird mit `~~TEXT~~` festgelegt

```
**fett**
```

```
*kursiv*
```

```
~~durchgestrichen~~
```


Markdown Sprache

Listen

- Listen werden mit *, + oder - vor der Zeile festgelegt
- nummerierte Liste werden mit 1., 2.... festgelegt (die Zahl ist eigentlich egal, weil Markdown selbst nummeriert)

1. erstes Item

2. zweites Item

4. drittes Item

- * wenn eine ein Listenabsatz eingerückt wird, wird daraus eine Subliste
- + es ist egal ob *, +, - genutzt wird

- ungeordnetes Item

1. erstes Item

2. zweites Item

3. drittes Item

- wenn eine ein Listenabsatz eingerückt wird, wird daraus ein Sublisten Item
- es ist egal ob +, +, - genutzt wird

- ungeordnetes Item

Markdown Sprache

Tabellen

Für Tabellen gibt es ein einfaches Format:

```
| Tables          | Are              | Cool  |
| -----:      |:-----:        | -----:|
| col 3 is      | right-aligned   | $1600 |
| col 2 is      | centered        | $12   |
| zebra stripes | are neat        | $1     |
```

Die Leerzeichen sind egal, hauptsache die | sind an der richtigen Stelle und in der zweiten Zeile wurde mindestens ein - zwischen den | verwendet.

Tables	Are	Cool
col 3 is	right-aligned	\$1600
col 2 is	centered	\$12
zebra stripes	are neat	\$1

Markdown Sprache

Codes / Skripte

Das wichtigste für R Markdown sind die Code Chunks (im Beispiel Dokument sind schon drei Code Chunks mit unterschiedlichen Spezifikationen inkludiert). Code Chunks in R starten mit ````{r}` und enden mit `````

```
```{r}
library(tidyverse)
library(readxl)
cws_data <- read_excel("_raw/CWS-data-2020.xlsx")
```
```

Alles was innerhalb ````{r}` CODE ````` geschrieben wird, führt R aus und zeigt den Code sowie den Output. Wenn innerhalb der Klammer `{r, echo = FALSE}` definiert wird, dann wird nur der Output ohne den Code angezeigt.

R Markdown Setup

Am Anfang der R Markdown Datei befindet sich das YAML zwischen zwei ---.

Im Beispieldokument befindet sich an zweiter Stelle ein R Code Chunk, der das Verhalten aller folgenden Chunks definiert:

```
```${r} setup, include=FALSE}  
knitr::opts_chunk$set(
 echo = FALSE,
 warning = FALSE,
 message = FALSE)
```
```

Die Optionen legen fest, dass

- der R Code im Dokument nicht angezeigt wird (echo = FALSE).
- keine Warn- und andere Meldungen angezeigt werden (warning = FALSE, message = FALSE).

R Markdown

Auf moodle befindet sich ein fertiges R Markdown Skript mit Berechnungen und Grafiken der letzten Sitzung. Das Format von R Markdown Dokumenten ist .Rmd und kann in R Studio mit `Open File...` geöffnet werden.

Output / Dokument erstellen

Um den Output (pdf, html oder doc) zu erstellen, muss auf der Menüleiste über dem Skriptfenster auf "**Knit**" geklickt werden. Sie werden dann gefragt, in welchem Ordner und unter welchem Namen die Datei gespeichert werden soll. html-Dokumente werden ohne Problem erstellt. Für Pdfs wird entweder Latex benötigt oder das Paket TinyTex.

R Markdown

Cheatsheets

Cheatsheet für die Sprache Markdown: <https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet>

Cheatsheet für R Markdown mit Code Chunks Optionen:
<https://rstudio.com/wp-content/uploads/2016/03/rmarkdown-cheatsheet-2.0.pdf>

Infos zu den YAML Optionen: <https://bookdown.org/yihui/bookdown/yaml-options.html>

Sehr gute Präsentation zu R Markdown mit Übungsaufgaben:
<https://apreshill.github.io/rmd4cdc/>

R Markdown Homepage: <https://rmarkdown.rstudio.com/>