

Datenanalyse mit R

# # 13 Kommunikation von Ergebnissen 1

Tobias Wiß, Carmen Walenta und Felix Wohlgemuth

05.06.2020



Institut für  
Gesellschafts-  
und Sozialpolitik

# 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")
```

Wiederholung

(multiple) lineare Regression

# lineare Regression - Modell

Ein lineares Modell wird mit `lm(abhängige Variable ~ unabhängige Variable, data = Dataframe)` erstellt.

*Funktionen die nicht aus dem tidyverse stammen, wie zB `lm()`, können auch mit `%>%` kombiniert werden. Mit `.` werden die Daten aus `%>%` als Input festgelegt.*

```
lm1 <- cws_data %>%  
  filter(year == 1990) %>%  
  lm(family_pub ~ fempar, data = . )
```

# lineare Regression - Ergebnisse

`summary(lm1)` zeigt die Ergebnisse des linearen Regressionsmodells:

```
##
## Call:
## lm(formula = family_pub ~ fempar, data = .)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.35164 -0.54523 -0.03309  0.61028  1.32332
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.61101    0.27648   2.210   0.0389 *
## fempar        0.07225    0.01479   4.885 8.97e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7202 on 20 degrees of freedom
## Multiple R-squared:  0.544,    Adjusted R-squared:  0.5212
## F-statistic: 23.86 on 1 and 20 DF,  p-value: 8.967e-05
```

# lineare Regression - Ergebnisse

Alternativ kann auch `tidy()` und `glance()` aus dem `broom`-Paket verwendet werden.

Der Output ist ein dataframe und kann in einem eigenen Objekt gespeichert werden. Dadurch können wir den Output weiterverwenden und zB mit `kable()` aus dem `kableExtra`-Paket die Ergebnisse in einer html-Tabelle darstellen:

```
library(broom)
lm1_coeff <- tidy(lm1)

library(kableExtra)
kable(lm1_coeff) %>% kable_styling(bootstrap_options = "condensed")
```

term	estimate	std.error	statistic	p.value
(Intercept)	0.611015	0.2764764	2.210008	0.0389259
fempar	0.072246	0.0147908	4.884512	0.0000897

# lineare Regression - Ergebnisse

*Wie `kable()` genau funktioniert und wie Ergebnisse von R nach Word übertragen werden, behandeln wir nach der Wiederholung.*

```
lm1_fit <- glance(lm1)
kable(lm1_fit) %>% kable_styling(bootstrap_options = "condensed") %>
```

r.squared	adj.r.squared	sigma	statistic	p.value	df	logLik	A
0.5439876	0.5211869	0.7202122	23.85845	8.97e-05	2	-22.94763	51.895

# lineare Regression - Vorhersagen

Mit `augment()` werden dem ursprünglichen Datensatz (Werte der abhängigen und unabhängigen Variable(n) im Modell) die vorhergesagten Werte `.fitted` und die Residuen `.resid` hinzugefügt:

```
lm1_augment <- augment(lm1)
kable(lm1_augment) %>% kable_styling(bootstrap_options = "condensed")
```

<b>family_pub</b>	<b>fempar</b>	<b>.fitted</b>	<b>.se.fit</b>	<b>.resid</b>	<b>.hat</b>	<b>.sign</b>
1.510	6.76	1.0993976	0.2011455	0.4106024	0.0780009	0.73238
2.530	19.67	2.0320928	0.1652296	0.4979072	0.0526326	0.72954
2.211	8.49	1.2243831	0.1856472	0.9866169	0.0664440	0.70080
0.578	13.22	1.5661065	0.1573520	-0.9881065	0.0477335	0.70145
3.193	32.96	2.9922416	0.2998826	0.2007584	0.1733731	0.73718
3.140	31.50	2.8867625	0.2815508	0.2532375	0.1528244	0.73622



# lineare Regression - Vorhersagen

Mit `augment()` können auch für selbst gewählte Werte der unabhängigen Variable(n) die Werte der abhängigen Variable vorhergesagt werden. Dafür benötigen wir ein Dataframe mit neuen Werte für `fempar`:

```
new_fempar <- data.frame("fempar" = c(0, 25, 50, 75, 100))  
lm1_predict <- augment(lm1, newdata = new_fempar)  
kable(lm1_predict)
```

fempar	.fitted	.se.fit
0	0.611015	0.2764764
25	2.417164	0.2076937
50	4.223313	0.5322544
75	6.029461	0.8927001
100	7.835610	1.2585678

# lineare Regression - Vorhersagen

Für Vorhersagen kann auch der `predict()` Befehl verwendet werden. Dem Output fehlen jedoch die selbst gewählten Werte und der Output ist auch kein Dataframe: (Mehr Infos: <http://www.sthda.com/english/articles/40-regression-analysis/166-predict-in-r-model-predictions-and-confidence-intervals/>)

```
lm1_predict2 <- predict(lm1, newdata = new_fempar, interval = "confic  
lm1_predict2 <- data.frame(new_fempar, lm1_predict2)  
kable(lm1_predict2)
```

fempar	fit	lwr	upr
0	0.611015	0.0342953	1.187735
25	2.417164	1.9839224	2.850405
50	4.223313	3.1130494	5.333576
75	6.029461	4.1673217	7.891601
100	7.835610	5.2102839	10.460937

# lineare Regression - Annahmen

Annahme	Überprüfungsmethode
Linearität des Zusammenhangs	Scatterplot
Keine extremen Ausreißer	Scatterplot
Normalverteilung der Residuen	Histogramm der Residuen
Konstante Varianz (Homoskedastizität)	Scatterplot der Residuen und vorhergesagten y-Werten
Unabhängigkeit der Beobachtungen	theoretische/konzeptionelle Überprüfung

# multiple lineare Regression

Für ein Regressionsmodell mit mehreren unabhängigen Variablen werden in `lm()` mehrere Variablen getrennt durch `+` angegeben.

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

# multiple linear model specification
# for 3 different years
cws_data_90 <- cws_data %>% filter(year == 1990)
lm_90 <- lm(family_pub ~ socx_pub + fempar + rtcseat + leftseat + t  

cws_data_00 <- cws_data %>% filter(year == 2000)
lm_00 <- lm(family_pub ~ socx_pub + fempar + rtcseat + leftseat + t  

cws_data_10 <- cws_data %>% filter(year == 2010)
lm_10 <- lm(family_pub ~ socx_pub + fempar + rtcseat + leftseat + t
```

Um die Ergebnisse der multiplen Regressionsmodelle anzuzeigen, wird auch `summary()` oder `tidy()` und `glance()` verwendet.

# multiple lineare Regression

*Folgende Variablen werden im Beispiel verwendet:*

	<b>Name</b>	<b>Inhalt</b>
AV	family_pub	Public expenditure on family benefits (% GDP)
UV1	socx_pub	Total public social expenditure (% GDP)
UV2	fempar	Share of seats in parliament held by women
UV3	rtcrseat	Share of seats in parliament won by parties classified as right Christian
UV4	leftseat	Share of seats in parliament won by parties classified as left
UV5	ud	Union density (net union membership as a % of employed wage and salary earners)
UV6	fem_employment	Share of female labour force (flabfo), as % of total labour force (tlabfo)
UV7	ptemp_f	Part-time employment for females, all ages, as a % of employment

```
##
## Call:
## lm(formula = family_pub ~ socx_pub + fempar + rtcseat + leftseat +
##      ud + fem_employment + ptemp_f, data = cws_data_90)
##
## Residuals:
##      Min        1Q      Median        3Q       Max
## -0.64488 -0.07919  0.03027   0.08847   0.91149
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -5.163106   1.744368  -2.960   0.0160 *
## socx_pub       0.102210   0.041446   2.466   0.0358 *
## fempar        -0.018443   0.022457  -0.821   0.4327
## rtcseat        0.007053   0.012889   0.547   0.5975
## leftseat       0.012418   0.013082   0.949   0.3673
## ud             0.030625   0.010711   2.859   0.0188 *
## fem_employment 0.074036   0.042623   1.737   0.1164
## ptemp_f        0.012267   0.012824   0.957   0.3638
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5356 on 9 degrees of freedom
## (5 observations deleted due to missingness)
## Multiple R-squared:  0.8701,    Adjusted R-squared:  0.769
## F-statistic: 8.609 on 7 and 9 DF,  p-value: 0.00225
```

```
##
## Call:
## lm(formula = family_pub ~ socx_pub + fempar + rtcseat + leftseat +
##      ud + fem_employment + ptemp_f, data = cws_data_00)
##
## Residuals:
##      Min        1Q      Median        3Q       Max
## -0.75829 -0.51161  0.04386   0.42261   1.15733
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -4.442503   3.055356  -1.454   0.174
## socx_pub       0.088712   0.054493   1.628   0.132
## fempar        -0.015505   0.024798  -0.625   0.545
## rtcseat       -0.010555   0.023117  -0.457   0.657
## leftseat       0.008798   0.014443   0.609   0.555
## ud            0.021958   0.012426   1.767   0.105
## fem_employment 0.071316   0.077449   0.921   0.377
## ptemp_f       0.027980   0.018149   1.542   0.151
##
## Residual standard error: 0.7445 on 11 degrees of freedom
## (3 observations deleted due to missingness)
## Multiple R-squared:  0.6101,    Adjusted R-squared:  0.362
## F-statistic: 2.459 on 7 and 11 DF,  p-value: 0.0881
```

```
##
## Call:
## lm(formula = family_pub ~ socx_pub + fempar + rtcseat + leftseat +
##      ud + fem_employment + ptemp_f, data = cws_data_10)
##
## Residuals:
##      Min        1Q    Median        3Q       Max
## -0.99162 -0.37908  0.02536  0.35363  1.45718
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -14.42880    5.50698   -2.620   0.0212 *
## socx_pub       0.09968    0.05682    1.754   0.1029
## fempar        -0.06430    0.03201   -2.009   0.0658 .
## rtcseat       -0.01467    0.02421   -0.606   0.5551
## leftseat       0.02237    0.01914    1.168   0.2637
## ud            0.04138    0.01587    2.608   0.0217 *
## fem_employment 0.27079    0.11430    2.369   0.0340 *
## ptemp_f        0.05985    0.02448    2.445   0.0295 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8345 on 13 degrees of freedom
## (1 observation deleted due to missingness)
## Multiple R-squared:  0.58,    Adjusted R-squared:  0.3538
## F-statistic: 2.565 on 7 and 13 DF,  p-value: 0.06773
```



# Kommunikation von Ergebnissen 1

# R und Word

Die Zusammenfassung der Regressionsergebnisse mit `summary()` beinhaltet alle wichtigen Informationen über unser Modell, doch leider sieht die Tabelle nicht so aus als ob sie direkt in Word kopiert werden kann. Natürlich können alle Outputs von R aus der Konsole kopiert und dann in Word oder Excel in Form gebracht werden. Aber es gibt eine Reihe von Befehlen, die das einfacher macht.

Um einfache Tabellen und Regressionstabellen von R nach Word zu kopieren, benötigen wir ein Zwischenschritt. Mit `stargazer()` oder `kable()` erstellen wir zuerst html-Tabellen und speichern diese ab. html-Tabellen können einfach in einem Browser geöffnet, makiert und dann nach Word kopiert werden. Das Layout und Aussehen der kopierten Tabelle kann dann direkt in Word geändert und somit dem Textstil angepasst werden.

*Alternativ können wir auch direkt Latex-Tabellen erstellen oder in R mit RMarkdown den Bericht schreiben. In der nächsten Sitzung wird es eine kurze Einführung in R Markdown geben. Alle Funktionen von heute können auch in R Markdown verwendet werden.*

# Kommunikation - Deskriptive Statistik

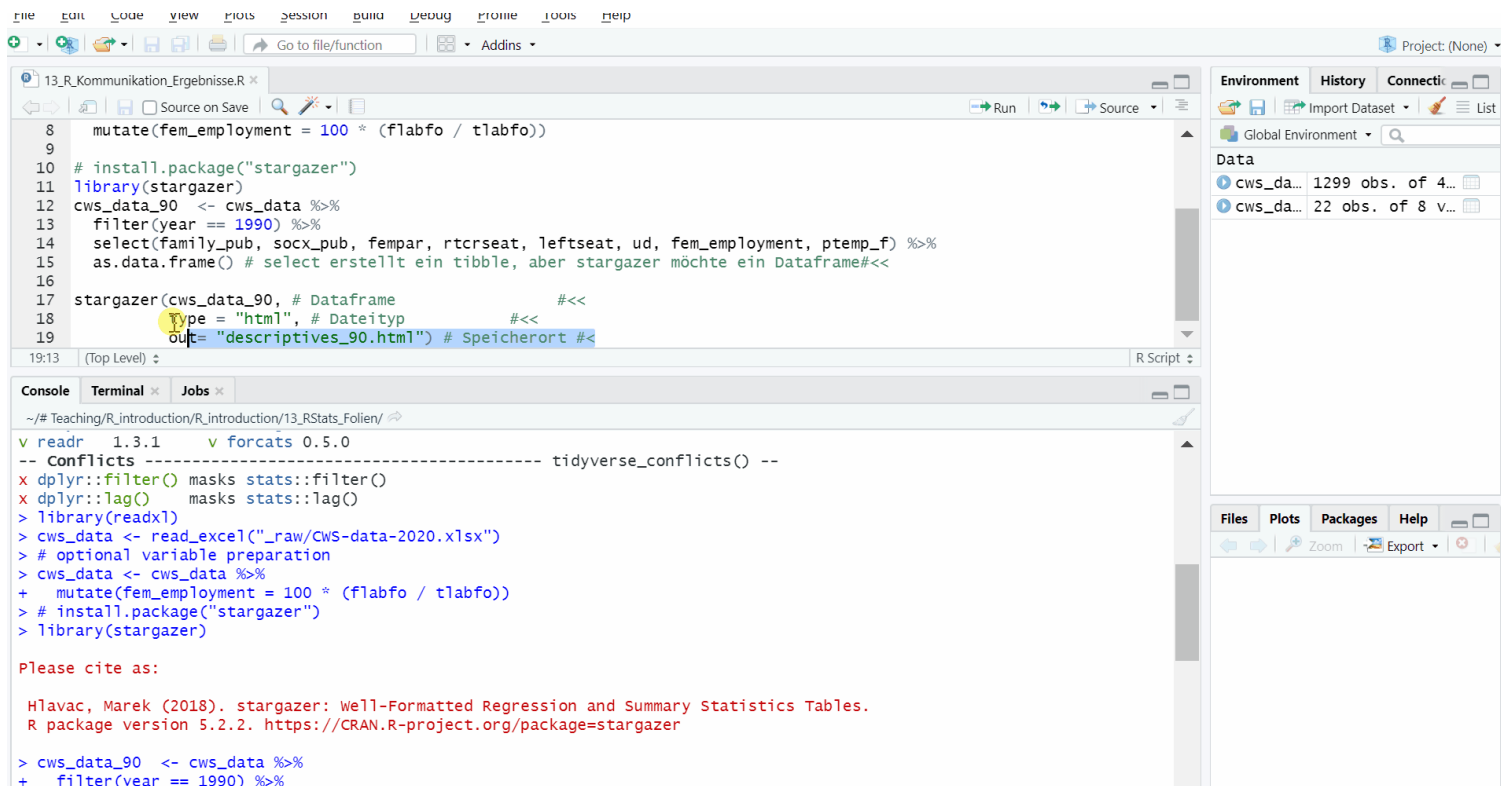
Wenn in `stargazer()` ein Dataframe ohne weitere Spezifikation angegeben wird, wird automatisch eine Tabelle mit deskriptiven Statistikmaßen erstellt:

```
# install.packages("stargazer")
library(stargazer)
cws_data_90 <- cws_data %>%
  filter(year == 1990) %>%
  select(family_pub, socx_pub, fempar, rtcseat, leftseat, ud, fem_er
as.data.frame() # select erstellt ein tibble, aber stargazer arbeitet mit Dataframes

stargazer(cws_data_90, # Dataframe
          type = "html", # Dateityp
          out= "descriptives_90.html") # Speicherort
```

# Kommunikation - Deskriptive Statistik

Die html-Tabelle befindet sich unter dem Dateipfad, der bei `out` = angegeben wurde. Dann die html-Datei im Browser öffnen, markieren und in das Worddokument kopieren.



The screenshot shows the RStudio environment with a script editor, console, and environment pane. The script editor contains R code for data manipulation and HTML table generation using the `stargazer` package. The console shows the execution of the code, including package installation and data filtering. The environment pane shows the loaded packages and data objects.

```
13_R_Kommunikation_Ergebnisse.R x
8 mutate(fem_employment = 100 * (flabfo / tlabfo))
9
10 # install.package("stargazer")
11 library(stargazer)
12 cws_data_90 <- cws_data %>%
13   filter(year == 1990) %>%
14   select(family_pub, socx_pub, fempar, rtcseat, leftseat, ud, fem_employment, ptemp_f) %>%
15   as.data.frame() # select erstellt ein tibble, aber stargazer möchte ein Dataframe#<<
16
17 stargazer(cws_data_90, # Dataframe #<<
18           type = "html", # Dateityp #<<
19           out = "descriptives_90.html") # Speicherort #<<
19:13 (Top Level) ↕ R Script ↕
```

Console

```
~/Teaching/R_introduction/R_introduction/13_RStats_Folien/
v readr 1.3.1 v forcats 0.5.0
-- Conflicts ----- tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag() masks stats::lag()
> library(readxl)
> cws_data <- read_excel("_raw/CWS-data-2020.xlsx")
> # optional variable preparation
> cws_data <- cws_data %>%
+   mutate(fem_employment = 100 * (flabfo / tlabfo))
> # install.package("stargazer")
> library(stargazer)

Please cite as:
Hlavac, Marek (2018). stargazer: Well-Formatted Regression and Summary Statistics Tables.
R package version 5.2.2. https://CRAN.R-project.org/package=stargazer

> cws_data_90 <- cws_data %>%
+   filter(year == 1990) %>%
```

Environment

Global Environment
Data
cws_da... 1299 obs. of 4...
cws_da... 22 obs. of 8 v...

Files Plots Packages Help

Zoom Export

# Kommunikation - Deskriptive Statistik

Statistic	N	Mean	St. Dev.	Min	Pctl(25)	Pctl(75)	Max
family_pub	22	1.734	1.041	0.312	0.733	2.493	4.033
socx_pub	22	18.924	4.747	10.927	15.108	22.849	27.237
fempar	22	15.545	10.626	2.340	7.098	20.322	38.400
rtcrseat	22	2.168	9.626	0	0	0	45
leftseat	22	38.964	15.063	0.000	34.300	48.475	56.400
ud	22	40.695	19.587	9.827	26.340	50.726	81.505
fem_employment	20	42.064	3.841	34.966	40.046	44.569	47.986
ptemp_f	18	25.492	11.273	10.624	18.541	29.763	52.514

*N* : Anzahl der Beobachtungen; *St. Dev.* : Standardabweichung; *Min* : Minimum; *Max* : Maximum; *Pctl(25)* : 25%-Perzentil; *Mean* : arithmetisches Mittel (50%-Perzentil); *Pctl(75)* : 75%-Perzentil

# Kommunikation - Deskriptive Statistik

Die Variablennamen und die Darstellung der Maße können nun in Word geändert werden oder auch gleich in `stargazer()`. Der Vorteil ist wie immer, dass im R-Skript alles nachvollziehbar ist. Folgende Optionen können in `stargazer()` verwendet werden:

- `digits =` : Anzahl der Nachkommastellen
- `title =` : Tabellentitel
- `covariate.labels = c()` : angezeigte Variablennamen  
(neue Namen müssen in der gleichen Reihenfolge sein wie die Variablen im Dataframe)
- `decimal.mark = ", "` : Deutsche Darstellung von Kommastellen

*Für die Darstellung von Regressionsergebnissen können noch weitere Optionen verwendet werden.*

# Kommunikation - Deskriptive Statistik

```
stargazer(cws_data_90,  
  digits = 2,  
  decimal.mark = ",",  
  title = "Deskriptive Statistik", # Tabellentitel  
  covariate.labels = c("öffentliche Ausgaben für Familienpol-  
    "öffentliche Sozialausgaben insgesamt  
    "Anteil weibliche Parlamentsabgeordnete  
    "Anteil Abgeordnete rechter christlich  
    "Anteil Abgeordnete linker Parteien",  
    "Anteil Gewerkschaftsmitglieder <i>(<strong>(%  
    "Anteil Frauen an Erwerbstätigen",  
    "Anteil Teilzeibesetzten an weiblich  
  type = "html",  
  out= "descriptives_90.html")
```

Da der Output eine html-Datei ist, können html-Tags wie `<i></i>` oder auch `<strong></strong>` verwendet werden

## 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



# Tabelle mit freiem Inhalt

Mit `kable()` aus dem `kableExtra`-Paket kann der Inhalt eines Dataframes als html-Tabelle exportiert werden. Der Inhalt kann zB ein gesamter Dataframes sein oder zB die Korrelation von `family_pub` und `fempar` je Level der `fed` Variable:

```
cws_cor <- cws_data %>%  
  filter(year == 1990) %>%  
  select(family_pub, socx_pub, fempar, rtcseat, leftseat, ud, fem_er  
  as.data.frame() %>%  
  group_by(fed) %>%  
  summarise(COR = cor(fempar, family_pub))  
  
kable(cws_cor) %>%  
  save_kable(file = "cor_fed.html")
```

*Mehr Infos zu `kable()`:*

[https://haozhu233.github.io/kableExtra/awesome\\_table\\_in\\_html.html](https://haozhu233.github.io/kableExtra/awesome_table_in_html.html)

# Tabelle mit freien Inhalt

fed	COR
0	0.7480076
1	1.0000000
2	0.9123703

Das Aussehen der Tabelle kann am einfachsten in Word bearbeitet werden.

# Regressionstabelle

Der eigentliche Verwendungszweck von `stargazer()` ist die Erstellung von Regressionstabellen. Dazu wird der Objektname des Regressionsmodells als Input angegeben:

```
lm_90 <- lm(family_pub ~ socx_pub + fempar + rtcseat + leftseat + u  
stargazer(lm_90,  
          type = "html",  
          out = "lm_90_table.html")
```

	<i>Dependent variable:</i>
	family_pub
socx_pub	0.102** (0.041)
fempar	-0.018 (0.022)
rtcrseat	0.007 (0.013)
leftseat	0.012 (0.013)
ud	0.031** (0.011)
fem_employment	0.074 (0.043)
ptemp_f	0.012 (0.013)
Constant	-5.163** (1.744)
Observations	17
R <sup>2</sup>	0.870
Adjusted R <sup>2</sup>	0.769
Residual Std. Error	0.536 (df = 9)
F Statistic	8.609*** (df = 7; 9)
<i>Note:</i>	* p<0.1; ** p<0.05; *** p<0.01

# Regressionstabelle

Der Vorteil von `stargazer()` ist, dass der Befehl auch mehrere Modelle in einer Regressionstabelle darstellen kann. Damit ist ein Vergleich von Modellen möglich:

```
lm_90 <- lm(family_pub ~ socx_pub + fempar + rtcrseat + leftseat + u
lm_00 <- lm(family_pub ~ socx_pub + fempar + rtcrseat + leftseat + u
lm_10 <- lm(family_pub ~ socx_pub + fempar + rtcrseat + leftseat + u

stargazer(lm_90, lm_00, lm_10,
          type = "html",
          out = "lm_table.html")
```

	<i>Dependent variable:</i>		
	family_pub		
	(1)	(2)	(3)
socx_pub	0.102** (0.041)	0.089 (0.054)	0.100 (0.057)
fempar	-0.018 (0.022)	-0.016 (0.025)	-0.064* (0.032)
rtcrseat	0.007 (0.013)	-0.011 (0.023)	-0.015 (0.024)
leftseat	0.012 (0.013)	0.009 (0.014)	0.022 (0.019)
ud	0.031** (0.011)	0.022 (0.012)	0.041** (0.016)
fem_employment	0.074 (0.043)	0.071 (0.077)	0.271** (0.114)
ptemp_f	0.012 (0.013)	0.028 (0.018)	0.060** (0.024)
Constant	-5.163** (1.744)	-4.443 (3.055)	-14.429** (5.507)
Observations	17	19	21
R <sup>2</sup>	0.870	0.610	0.580
Adjusted R <sup>2</sup>	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

# Regressionstabelle

Zusätzlich zu den bekannten Optionen, kann mit folgenden Optionen das Aussehen von Regressionstabellen verändert werden:

- `column.lables = c()` : Modellnamen definieren
- `dep.var.labels = c()` : Abhängige Variablennamen definieren
- `no.space = TRUE` : Leerzeilen weglassen (hilft beim Kopieren in Word)

```
stargazer(lm_90, lm_00, lm_10,  
  column.labels = c("Jahr = 1990", "Jahr = 2000", "Jahr = 2010"),  
  dep.var.labels = c("öffentliche Ausgaben für Familienpolitik",  
    "öffentliche Sozialausgaben insgesamt",  
    "Anteil weibliche Parlamentsabgeordnete",  
    "Anteil Abgeordnete rechter christlicher Parteien",  
    "Anteil Abgeordnete linker Parteien",  
    "Anteil Gewerkschaftsmitglieder <i>(%</i>)",  
    "Anteil Frauen an Erwerbstätigen",  
    "Anteil Teilzeibeschäftigten 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 <sup>2</sup>	0.870	0.610	0.580
Adjusted R <sup>2</sup>	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		



# Übung 13

- Verwenden Sie Daten Ihrer Wahl (alternativ CWS Daten).
- Wählen Sie eine abhängige Variable und mehrere unabhängige Variablen für ein Regressionsmodell aus.
- Erstellen Sie eine Tabelle mit deskriptiven Statistikmaßen und kopieren Sie diese in ein Worddokument.
- Erstellen Sie ein Regressionsmodell mit `lm()`.
- Erstellen Sie für das Modell eine Regressionstabelle und kopieren Sie diese in das Worddokument.
- Laden Sie Ihr R-Skript und das Worddokument (gerne als pdf) bis zum 19.06.2020 12:00 auf moodle hoch.

nächste Woche:

# Kommunikation von Ergebnissen 2

In der letzten Einheit beschäftigen wir uns weiter mit der Kommunikation von Ergebnissen.

Falls Sie Fragen zur Darstellung Ihrer Ergebnisse in Word haben, dann können Sie mir gerne Fragen für die letzte Stunde schicken (bis spätestens 16.06.). Bitte schauen Sie sich vorher die Folien von dieser Woche an.

Ich werde Ihnen eine kurze Einführung in R Markdown geben.