Statistische Datenanalyse

Berliner Hochschule für Technik: B.Sc. Angewandte Mathematik

Arne Maaß

2024-06-02

Table of contents

Projektbericht Statistische Datenanalyse

1. Maximum-Likelihood-Schätzung (5P)

Erläutern Sie das Prinzip der Maximum-Likelihood-Schätzung an einem Beispiel.

- *Ziehen Sie sich (in R) eine Stichprobe aus einer der folgenden Verteilungen: Geometrisch (Achtung: R definiert die Verteilung etwas anders als wir in der Formelsammlung), Poisson, Exponential.*

oder Modifikation zur R-Stichprobe: Wenn Sie einen passenden Datensatz finden mit einer Variable, die zu einer dieser Verteilungen passt, dürfen Sie auch diese Daten anstatt der Stichprobe verwenden.

```r
set.seed(102377) # Für Reproduzierbarkeit

n <- 100 # Größe der Stichprobe
prob <- 0.3 # Erfolgswahrscheinlichkeit
sample <- rgeom(n, prob) + 1
# Zieht eine Stichprobe und addiert 1, da rgeom von 0 beginnt
```

- *Schreiben Sie die Herleitung der Schätzung in den wesentlichen Schritten auf.*

Wahrscheinlichkeitsfunktion aufstellen:

Die Wahrscheinlichkeitsfunktion einer geometrischen Verteilung (die bei 1 beginnt) für eine einzelne Beobachtung ist gegeben durch:

$$P(X = x) = (1 - p)^{x-1}p$$

Likelihood-Funktion aufstellen:

Für eine Stichprobe von n unabhängigen Beobachtungen x_1 ,x_2 ,...,x_n

ist die Likelihood-Funktion das Produkt der einzelnen Wahrscheinlichkeiten

$$L(p) = \prod_{i=1}^{n}(1 - p)^{x_i-1}p$$

Log-Likelihood-Funktion:

Die Log-Likelihood-Funktion, die der natürliche Logarithmus der Likelihood-Funktion ist, vereinfacht die Berechnungen, da sie uns erlaubt mit einer Summe statt einem Produkt zu arbeiten und ist gegeben durch:

$$l(p) = \ln(L(p)) = \sum_{i=1}^{n}[(x_i - 1)\ln(1 - p) + \ln(p)]$$

Ableitung der Log-Likelihood-Funktion:

Um den Wert von p zu finden, der $l(p)$ maximiert, leiten wir $l(p)$ nach p ab:

$$\frac{dl(p)}{dp} = \sum_{i=1}^{n} \left[\frac{-(x_i - 1)}{1 - p} + \frac{1}{p} \right]$$

Bestimmung des Maximums:

Wir setzen die Ableitung gleich Null und lösen die Gleichung nach p

$$0 = \sum_{i=1}^{n} \left[\frac{-(x_i - 1)}{1 - p} + \frac{1}{p} \right]$$

Durch Umformen erhalten wir die MLE-Schätzung für $\hat{p}$. Die genaue Lösung hängt von der spezifischen Form der Gleichung ab und führt zu:

$$\hat{p} = \frac{n}{\sum_{i=1}^{n} x_i}$$

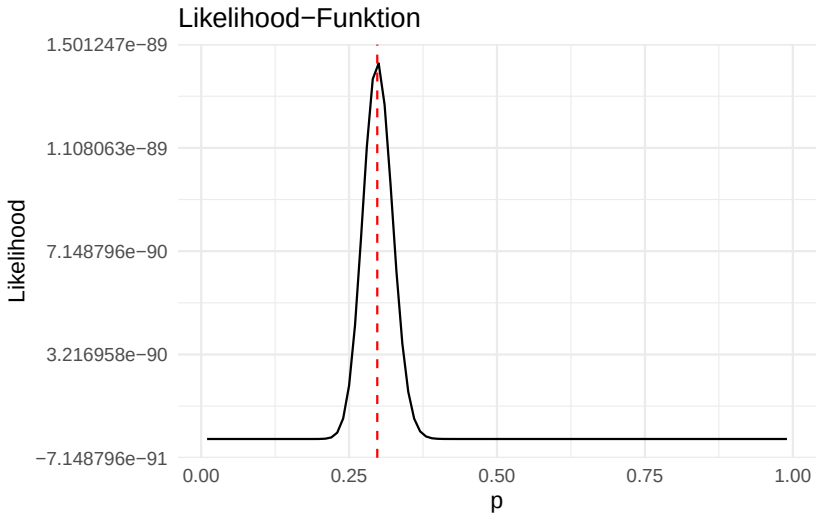
- *Wenden Sie die hergeleitete Schätzung auf Ihre Daten an und stellen Sie die sowohl die Likelihood- als auch die Log-Likelihood-Funktion (inklusive der eingezeichneten Schätzung) passend grafisch dar.*

```r
# Maximum-Likelihood-Schätzung für p
p_hat <- 1 / mean(sample)

# Likelihood- und Log-Likelihood-Funktion definieren
likelihood <- function(p, data) prod(dgeom(data - 1, prob = p))
log_likelihood <- function(p, data) sum(dgeom(data - 1, prob = p, log = TRUE))

# Werte für p über einen Bereich generieren
p_vals <- seq(0.01, 0.99, by = 0.01)

# Likelihood und Log-Likelihood für jeden Wert von p berechnen
likelihood_vals <- sapply(p_vals, likelihood, data = sample)
log_likelihood_vals <- sapply(p_vals, log_likelihood, data = sample)

# Daten für die Plots vorbereiten
plot_data <- tibble(p = p_vals, likelihood = likelihood_vals, log_likelihood = log_likelih

# Likelihood-Plot erstellen
```

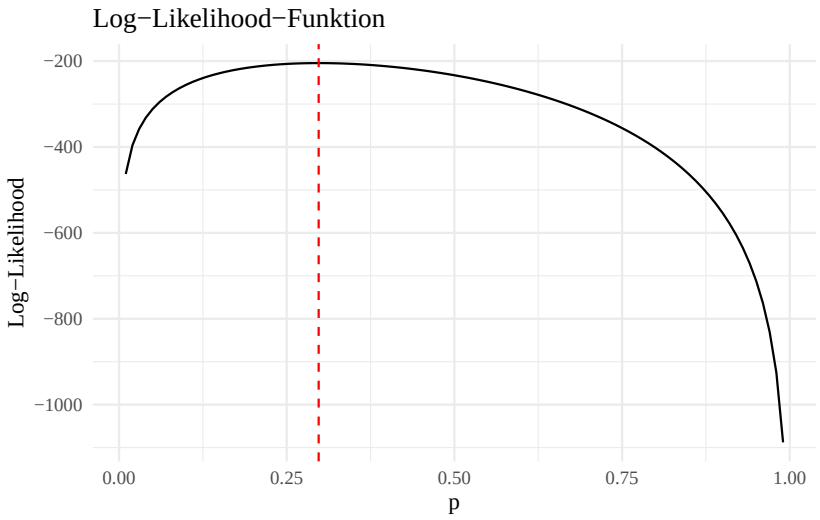
```r
  p1 <- ggplot(plot_data, aes(x = p, y = likelihood)) +
    geom_line() +
    geom_vline(xintercept = p_hat, linetype = "dashed", color = "red") +
    labs(title = "Likelihood-Funktion", x = "p", y = "Likelihood") +
    theme_minimal()
    theme(text = element_text(family = "serif"))
```

```
List of 1
 $ text:List of 11
  ..$ family       : chr "serif"
  ..$ face         : NULL
  ..$ colour       : NULL
  ..$ size         : NULL
  ..$ hjust        : NULL
  ..$ vjust        : NULL
  ..$ angle        : NULL
  ..$ lineheight   : NULL
  ..$ margin       : NULL
  ..$ debug        : NULL
  ..$ inherit.blank: logi FALSE
  ..- attr(*, "class")= chr [1:2] "element_text" "element"
 - attr(*, "class")= chr [1:2] "theme" "gg"
 - attr(*, "complete")= logi FALSE
 - attr(*, "validate")= logi TRUE
```

```r
  # Log-Likelihood-Plot erstellen
  p2 <- ggplot(plot_data, aes(x = p, y = log_likelihood)) +
    geom_line() +
    geom_vline(xintercept = p_hat, linetype = "dashed", color = "red") +
    labs(title = "Log-Likelihood-Funktion", x = "p", y = "Log-Likelihood") +
    theme_minimal()+
    theme(text = element_text(family = "serif"))

  # Plots anzeigen
  p1
```




```
p2
```



2. Schätztheorie (5P)

Diskutieren Sie die Eigenschaften des Mittelwerts als Schätzung für den Erwartungswert der Verteilung.

- *Suchen Sie sich eine der folgenden Verteilungen aus: Bernoulli, Geometrisch, Poisson, Exponential - allerdings nicht dieselbe Verteilung wie in Themenabschnitt 1.!*

Hier wurde eine exonential Verteilung gewählt!

- *Mit dem Mittelwert $\overline{X}$ schätzen Sie nun den Erwartungswert Ihrer gewählten Verteilung. Erläutern Sie daran die Begriffe Bias, MSE und Konsistenz.*

Bias

Der Bias eines Schätzers ist die Differenz zwischen dem Erwartungswert des Schätzers und dem wahren Wert des Parameters. Ein unverzerrter Schätzer hat einen Erwartungswert, der gleich dem wahren Parameterwert ist. Ist der Mittelwert $\overline{X}$ der Exponentialverteilung ein unverzerrter Schätzer für den Erwartungswert μ , so gilt $E(\overline{X}) = \mu$. Dies wird für identisch und unabhängig verteilte (i.i.d.) Stichproben mit $n > 30$ angenommen, aufgrund des Zentralen Grenzwertsatzes, der besagt, dass der Mittelwert solcher Stichproben gegen eine Normalverteilung konvergiert, unabhängig von der ursprünglichen Verteilung der Variablen, solange die ursprüngliche Verteilung einen endlichen Erwartungswert und eine endliche Varianz aufweist

MSE

Der Mean Squared Error (MSE) eines Schätzers ist der Erwartungswert des quadratischen Fehlers.

$$\text{MSE}(\hat{\theta}) = E[(\hat{\theta} - \theta)^2$$

MSE kombiniert die Varianz des Schätzers und seinen quadrierten Bias:

$$MSE(X) = Bias(X)^2 + Var(X)$$

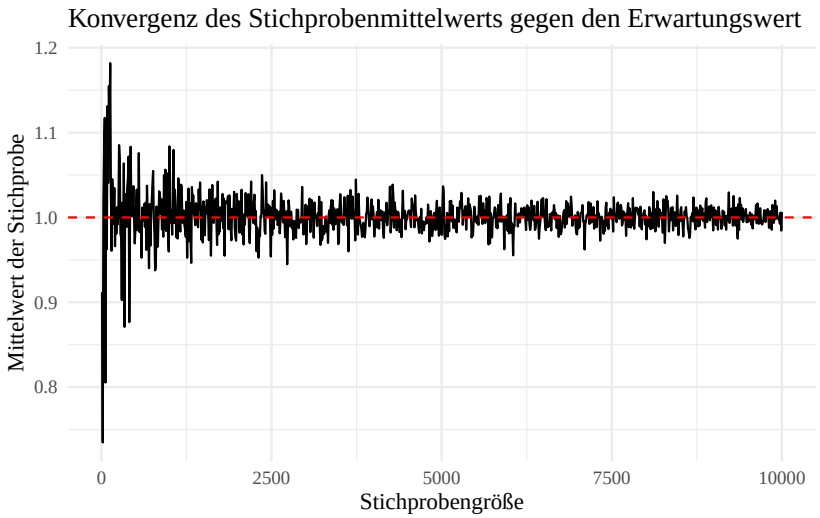
Für einen unverzerrten Schätzer wie $\overline{X}$ ist der MSE gleich der Varianz des Schätzers, da $Bias = 0$, aufgrund der Erwartunstreue.

Konsistenz

Ein Schätzer ist konsistent, wenn er mit zunehmender Stichprobengröße gegen den wahren Parameterwert konvergiert. Anders ausgedrückt ist Mittelwert $\overline{X}$ ein konsistenter Schätzer für den Erwartungswert, da mit zunehmender Stichprobengröße die Varianz von $\overline{X}$ gegen null konvergiert während der Bias bereits null ist wegen der iid-Stichprobe.

- *Überlegen Sie sich wie Sie mit Hilfe von Stichprobenziehungen in R (grafisch) illustrieren können, dass $\overline{X}$ gegen den "wahren" Erwartungswert konvergiert.*

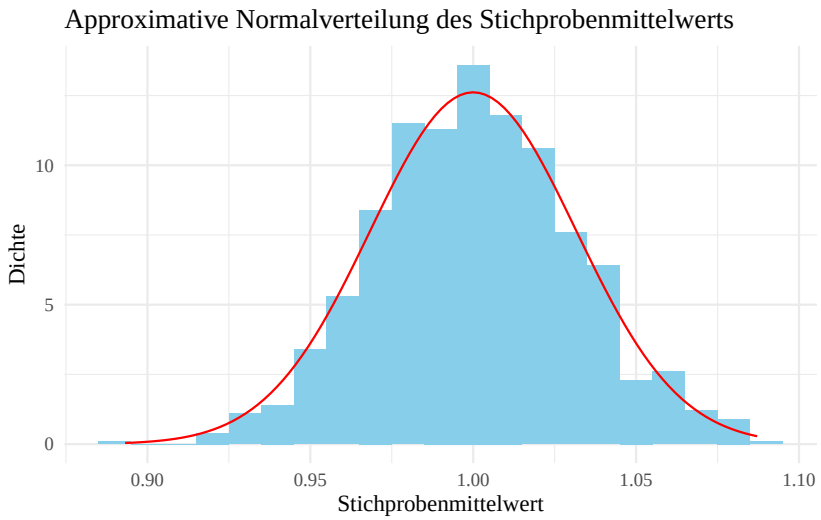
```r
# Parameter setzen
lambda <- 1
mu <- 1 / lambda # wahrer Wert

# Stichprobengrößen
sample_sizes <- seq(10, 10000, by = 10)

# Simulation der Mittelwerte
set.seed(102377)
means <- sapply(sample_sizes, function(n) {
  mean(rexp(n, rate = lambda))
})

data_means <- data.frame(SampleSize = sample_sizes, Mean = means)

# Plot der Konvergenz
ggplot(data_means, aes(x = SampleSize, y = Mean)) +
  geom_line() +
  geom_hline(yintercept = mu, linetype = "dashed", color = "red") +
  labs(title = "Konvergenz des Stichprobenmittelwerts gegen den Erwartungswert",
       x = "Stichprobengröße", y = "Mittelwert der Stichprobe") +
  theme_minimal()+
  theme(text = element_text(family = "serif"))
```



- *Überlegen Sie sich außerdem, wie Sie mit Hilfe von Stichprobenziehungen in R (grafisch) illustrieren können, dass $\overline{X}$ eine approximative Normalverteilung hat. Welche approxi-*

mative Normalverteilung sollte das bei Ihnen sein - in Abhängigkeit vom Parameter der Verteilung?

```r
set.seed(102377)
num_simulations <- 1000
simulated_means <- replicate(num_simulations, mean(rexp(1000, rate = lambda)))

# Daten vorbereiten
data_simulated_means <- data.frame(Mean = simulated_means)

# Histogramm mit einer Normalverteilungskurve
ggplot(data_simulated_means, aes(x = Mean)) +
  geom_histogram(aes(y = after_stat(density)), binwidth = 0.01, fill = "skyblue") +
  stat_function(fun = dnorm, args = list(mean = mu, sd = (1 / lambda) / sqrt(1000)), color
  labs(title = "Approximative Normalverteilung des Stichprobenmittelwerts",
       x = "Stichprobenmittelwert", y = "Dichte") +
  theme_minimal() +
  theme(text = element_text(family = "serif"))
```



3. Hypothesentests (5P)

Hier benötigen Sie einen -gern selbst gewählten- Datensatz mit quantitativ-stetige Variablen und zumindest auch einem qualititativ-binären Merkmal.

Für die kommenden Aufgabe wird die Welle 10 des European Social Surveys verwendet. Ziel dieser Untersuchung wird es nachzuvollziehen welche Faktoren mit einer linken oder rechten

politischen Gesinnung einhergehen (In Anlehnung an die kürzlich veröffentliche Studie der Financial Times: https://www.ft.com/content/29fd9b5c-2f35-41bf-9d4c-994db4e12998) . Dafür schauen wir lediglich nach Korrelation.

Vorab werden ein paar Datenvorbereitungsschritte durchlaufen und Zielgruppe eingeschränkt.

Wir schauen uns drei Alters-Kohorten an und lediglich Leute die auch im Befragunsland sozialisiert wurden. Die Links-Rechts-Skala wird Dummy-Kodiert mit links <= 5 und rechts >5.

Folgende variablen werden für die Analyse ausgewählt:

label	Inhalt
lrscale	Links/Rechts Skala
lrdummy	Links Rechts Dummy
gndr	Geschlecht
trstprl	Vertrauen in das nationale Parlament
netusoft	Internetnutzung
stflife	Lebenszufriedenheit
health	Subjektive Gesundheit
rlgdgr	Religiösität
wrclmch	Sorgen um Klimawandel
chldo12_dummy	Eigenes Kind über 12
cohort	Geburtskohorte nach Jahrzehnten
cntry	Land der Befragung (synonym mit Citizenship und Geburtsland durch implementierten Filter)

```
# df <- read.csv("C:/Users/user/OneDrive/Desktop/BHT/StatDA/PROJEKT/data/ESS10.csv")
df <- read_parquet("C:/Users/user/OneDrive/Desktop/BHT/StatDA/PROJEKT/data/ess_sda.parquet


#Stichprobenziehung um Replikationen zu vermeidne:
set.seed(102377)
df <- initial_split(prop = 0.8, data=df) %>% training(.)

df <- df %>%
  # Wir schauen uns 3 Kohorten an:
  mutate(cohort = case_when(yrbrn %in% c(1980:1989) ~ "1980s",
                            yrbrn %in% c(1990:1999) ~ "1990s",
                            yrbrn %in% c(2000:2009) ~ "2000s")) %>%
  filter(!is.na(cohort)) %>%
  # Leute die im Befragunsland sozialisiert wurden
  filter(ctzcntr == 1 & brncntr == 1) %>%
```

```
  # Filtern von NA- Werten:
  filter(lrscale <= 10) %>%
  mutate(dummy_lr = as.factor(ifelse(lrscale<=5, 0, 1))) %>%
  filter(trstprl %in% (0:10)) %>%
  filter(netusoft %in% (1:5)) %>%
  filter(stflife <= 10) %>%
  filter(health <= 5) %>%
  filter(rlgdgr <= 10) %>%
  filter(wrclmch <= 5) %>%
  mutate(chldo12_dummy = ifelse(chldo12 %in% c(1,2,3,4,5,6), 1,
                                ifelse(chldo12 == 0, 0, NA))) %>%
  select(cntry, lrscale, dummy_lr, trstprl, cohort, gndr, netusoft, stflife, health, rlgdg
  na.omit()
```

- *Wenden Sie mindestens 3 verschiedene Tests direkt auf die Variablen an. Mindestens einer der Tests soll dabei auch ein einseitiger sein.*

```
# Ein-Stichproben T-Test für Vertrauen in das nationale Parlament
t_test_trstprl <- t.test(trstprl ~ dummy_lr, data=df)

# Einseitiger T-Test für Lebenszufriedenheit zwischen linker und rechter Gesinnung
t_test_stflife <- t.test(stflife ~ dummy_lr, data=df, alternative="greater")

# Chi-Quadrat-Test für `gender` und `dummy_lrs`
tbl_gender_lrscale <- table(df$gndr, df$dummy_lr)
chi_sq_test <- chisq.test(tbl_gender_lrscale)

# Ergebnisse
list(t_test_trstprl = t_test_trstprl,
     t_test_stflife = t_test_stflife,
     chi_sq_test = chi_sq_test)
```

```
$t_test_trstprl

    Welch Two Sample t-test

data:  trstprl by dummy_lr
t = -8.8925, df = 5485.4, p-value < 2.2e-16
alternative hypothesis: true difference in means between group 0 and group 1 is not equal to
95 percent confidence interval:
 -0.6824395 -0.4358968
```

```
sample estimates:
mean in group 0 mean in group 1
       4.522411        5.081579


$t_test_stflife

    Welch Two Sample t-test

data:  stflife by dummy_lr
t = -7.6597, df = 5965.4, p-value = 1
alternative hypothesis: true difference in means between group 0 and group 1 is greater than
95 percent confidence interval:
 -0.4184603        Inf
sample estimates:
mean in group 0 mean in group 1
       7.236728        7.581203


$chi_sq_test

    Pearson's Chi-squared test with Yates' continuity correction

data:  tbl_gender_lrscale
X-squared = 47.435, df = 1, p-value = 5.687e-12
```

- *Was sind jeweils Hypothese und Alternative, was ist das Testergebnis? (Schön wäre es, wenn die Hypothese nicht immer abgelehnt wird, experimentieren Sie ggf. etwas mit Ihrern Datenbeispielen herum.)*

1. **T-Test für unabhängige Stichproben für Vertrauen in das nationale Parlament**:

   Nullhypothese (H0): Es gibt keinen Unterschied im Mittelwert des Vertrauens in das nationale Parlament zwischen den Gruppen mit linker (5) und rechter (>5) politischer Gesinnung. **Alternativhypothese (H1):** Es gibt einen Unterschied im Mittelwert des politischen Vertrauens zwischen Gruppen mit linker und rechter politischer Gesinnung.

   Ergebnis: Der sehr kleine p-Wert deutet darauf hin, dass die Nullhypothese mit einer hohen Konfidenz abgelehnt wird. Es gibt einen statistisch signifikanten Unterschied im Vertrauen zum nationalen Parlament zwischen den Gruppen mit linker und rechter politischer Gesinnung in den Daten, wobei die linke Gruppe ein niedrigeres durchschnittliches Vertrauen aufweist.

2. **Einseitiger T-Test für Lebenszufriedenheit zwischen linker und rechter Gesinnung:**

   Nullhypothese (H0): Es gibt keinen Unterschied in der Lebenszufriedenheit zwischen den Gruppen mit linker und rechter politischer Gesinnung, oder die Lebenszufriedenheit ist bei linker Gesinnung niedriger.

   Alternativhypothese (H1): Die Lebenszufriedenheit ist in der Gruppe mit rechter politischer Gesinnung geringer als in der Gruppe mit linker politischer Gesinnung.

   Testergebnis: Der p-werit von 1 weist darauf hin, dass die Alternativhypothese abgelehnt wird und die Nullhypothese unterstützt: Die Gruppe mit linker politischer Gesinnung weist eine geringere durchschnittliche Lebenszufriedenheit auf als die Gruppe mit rechter Gesinnung.

3. **Pearson's Chi-squared Test für Geschlecht und politische Gesinnung**

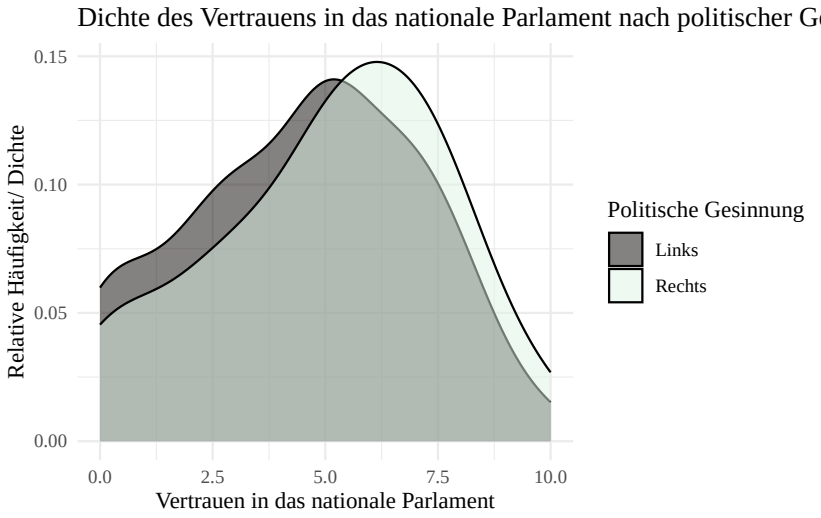
   Nullhypothese (H0): Es gibt keine Assoziation zwischen Geschlecht und politischer Gesinnung (links/rechts).

   Alternativhypothese (H1): Es gibt eine Assoziation zwischen Geschlecht und politischer Gesinnung.

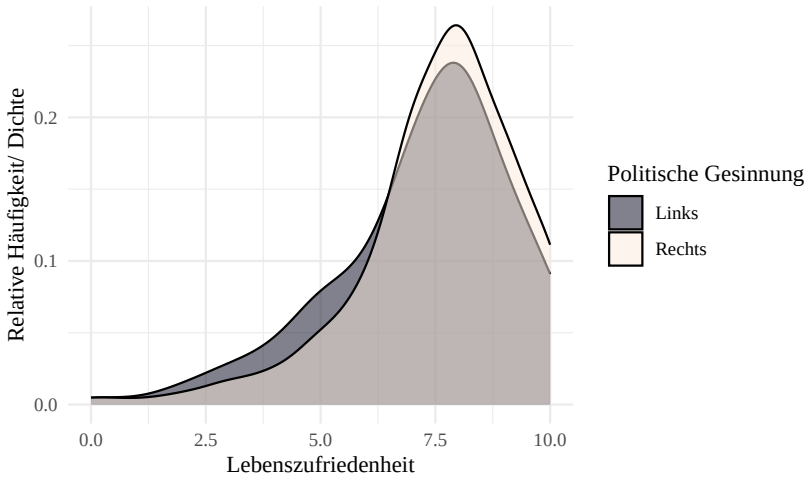
   Testergebnis: Der p-Wert ist extrem klein, was darauf hinweist, dass die Nullhypothese abgelehnt wird. Es gibt eine signifikante Assoziation zwischen Geschlecht und politischer Gesinnung in den Daten.

- *Erläutern Sie anhand von passenden explorativen Grafiken (am besten mit dem R-Paket ggplot2), dass diese Testergebnisse für Ihre Daten sinnvoll erscheinen.*

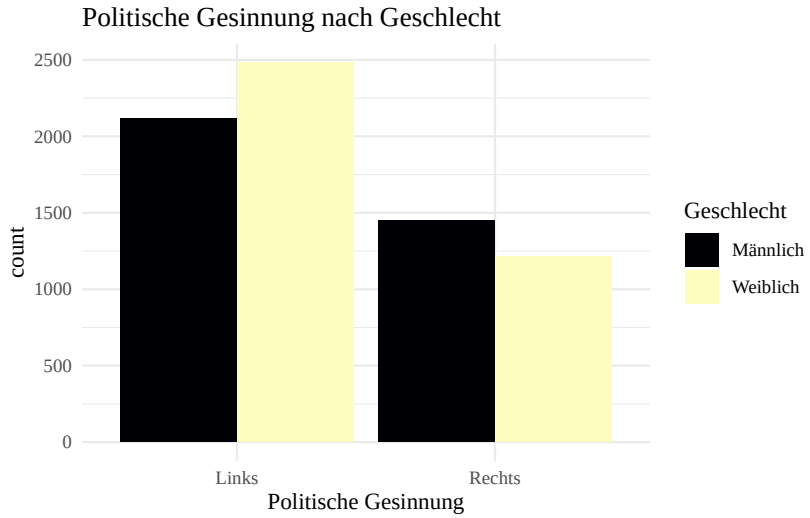
```r
# Vertrauen in das nationale Parlament
ggplot(df, aes(x = trstprl, fill = factor(dummy_lr, labels = c("Links", "Rechts")))) +
  geom_density(alpha = 0.5, adjust = 2) +
  scale_fill_viridis(option = "G" ,discrete=T) +
  labs(title = "Dichte des Vertrauens in das nationale Parlament nach politischer Gesinnun
       x = "Vertrauen in das nationale Parlament",
       y = "Relative Häufigkeit/ Dichte",
       fill = "Politische Gesinnung") +
  theme_minimal() +
  theme(text = element_text(family = "serif"))
```




```r
# Lebenszufriedenheit
ggplot(df, aes(x = stflife, fill = factor(dummy_lr, labels = c("Links", "Rechts")))) +
  geom_density(alpha = 0.5, adjust = 2) +
  scale_fill_viridis(option = "F" ,discrete=T) +
  labs(title = "Dichte der Lebenszufriedenheit nach politischer Gesinnung",
       x = "Lebenszufriedenheit",
       y = "Relative Häufigkeit/ Dichte",
       fill = "Politische Gesinnung") +
  theme_minimal() +
  theme(text = element_text(family = "serif"))
```



```r
# Geschlecht und politische Gesinnung
ggplot(df, aes(x = factor(dummy_lr, labels = c("Links", "Rechts")), fill = factor(gndr, la
  geom_bar(position = "dodge") +
  scale_fill_viridis(option = "A" ,discrete=T) +
  labs(title = "Politische Gesinnung nach Geschlecht",
       x = "Politische Gesinnung",
       fill = "Geschlecht") +
  theme_minimal() +
  theme(text = element_text(family = "serif"))
```



4. Regressionsmodelle - LM/GLM (10P)

Hier benötigen Sie einen -gern selbst gewählten- Datensatz mit möglichst Variablen verschiedener Merkmalstypen darin

Sie sollen verschiedene Regressionsmodelle schätzen und vergleichen bzw. optimale Modelle finden..

- *Beginnen Sie zunächst mit einer ausführlicheren explorativen (grafischen) Analyse Ihrer erklärenden Variablen und deren einzelner Auswirkung auf die abhängige Variable. Hierzu sollten Sie vorrangin ggplot2 verwenden um die Grafiken zu erstellen.*

```r
# # Fastest and easiest way:
# DataExplorer::create_report(df)

# Übersicht
```

Table 2: Korrelationsmatrix

	lrscale	trstprl	gndr	netusoft	stflife	health	rlgdgr	wrclmch	chldo12_dummy
lrscale	1.000	0.033	-0.074	-0.026	0.101	-0.049	0.082	-0.131	0.036
trstprl	0.033	1.000	-0.001	0.074	0.251	-0.029	-0.055	0.071	-0.081
gndr	-0.074	-0.001	1.000	0.037	0.003	0.033	0.108	0.139	0.113
netusoft	-0.026	0.074	0.037	1.000	0.133	-0.078	-0.075	0.047	-0.075
stflife	0.101	0.251	0.003	0.133	1.000	-0.224	-0.006	0.011	-0.034
health	-0.049	-0.029	0.033	-0.078	-0.224	1.000	-0.003	0.004	0.081
rlgdgr	0.082	-0.055	0.108	-0.075	-0.006	-0.003	1.000	-0.001	0.062
wrclmch	-0.131	0.071	0.139	0.047	0.011	0.004	-0.001	1.000	-0.030
chldo12_dummy	0.036	-0.081	0.113	-0.075	-0.034	0.081	0.062	-0.030	1.000

Table 3: Kovarianzmatrix

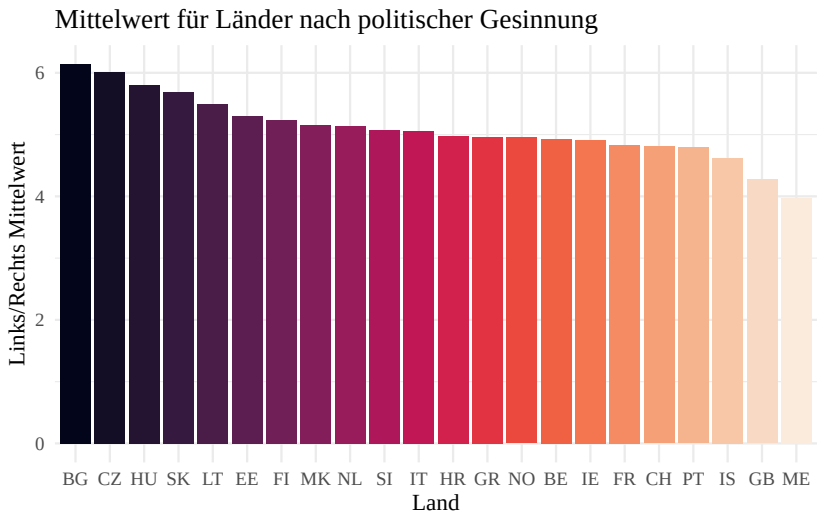
	lrscale	trstprl	gndr	netusoft	stflife	health	rlgdgr	wrclmch	chldo12_dummy
lrscale	4.860	0.189	-0.082	-0.035	0.422	-0.078	0.552	-0.273	0.024
trstprl	0.189	6.682	-0.002	0.116	1.231	-0.054	-0.434	0.172	-0.063
gndr	-0.082	-0.002	0.250	0.011	0.003	0.012	0.165	0.066	0.017
netusoft	-0.035	0.116	0.011	0.367	0.153	-0.035	-0.138	0.027	-0.014
stflife	0.422	1.231	0.003	0.153	3.603	-0.311	-0.037	0.020	-0.020
health	-0.078	-0.054	0.012	-0.035	-0.311	0.534	-0.006	0.003	0.018
rlgdgr	0.552	-0.434	0.165	-0.138	-0.037	-0.006	9.373	-0.002	0.058
wrclmch	-0.273	0.172	0.066	0.027	0.020	0.003	-0.002	0.887	-0.009
chldo12_dummy	0.024	-0.063	0.017	-0.014	-0.020	0.018	0.058	-0.009	0.092

```
# Korrelationsmatrix
kable(as.data.frame(cor(as.matrix(select_if(df, is.numeric)))),digits = 3, format = "latex
```

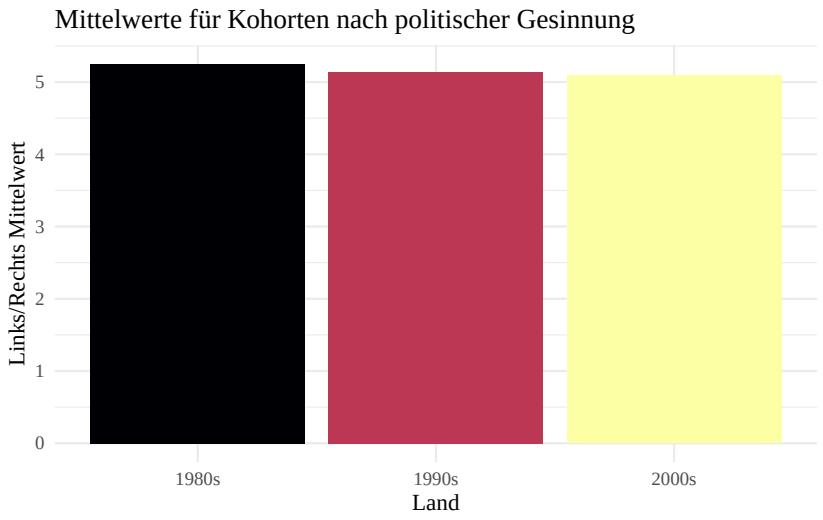
```
# Kovarianzmatrix
kable(cov(select_if(df, is.numeric)),digits = 3, format = "latex", caption= "Kovarianzmatr
```

```
# Visuelle Exploration

# Länder
df %>%
  group_by(cntry) %>%
  summarise(mean_value = mean(as.numeric(lrscale), na.rm = TRUE)) %>%
  ungroup() %>%
  mutate(cntry = reorder(cntry, -mean_value)) %>%
  ggplot(aes(x = cntry, y = mean_value, fill = cntry)) +
  geom_bar(stat = "identity", position = position_dodge()) +
  scale_fill_viridis_d(option = "F") +
  labs(title = "Mittelwert für Länder nach politischer Gesinnung",
```

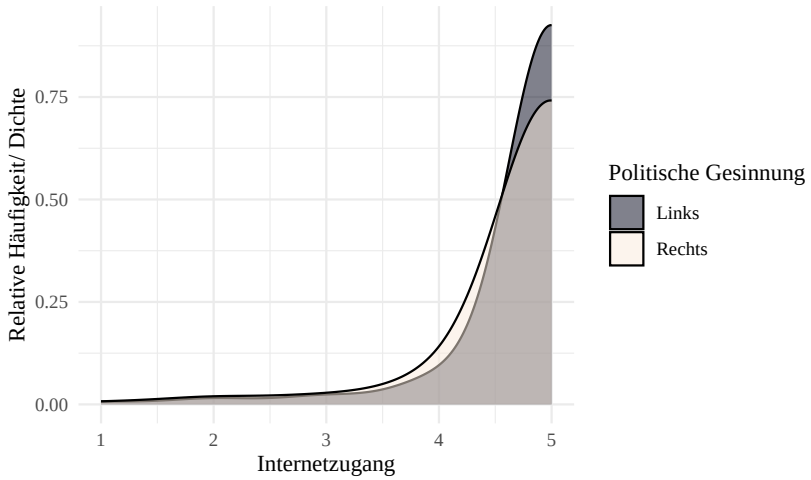
```
      x = "Land",
      y = "Links/Rechts Mittelwert") +
  theme_minimal() +
  theme(text = element_text(family = "serif"),
        legend.position = "none")
```



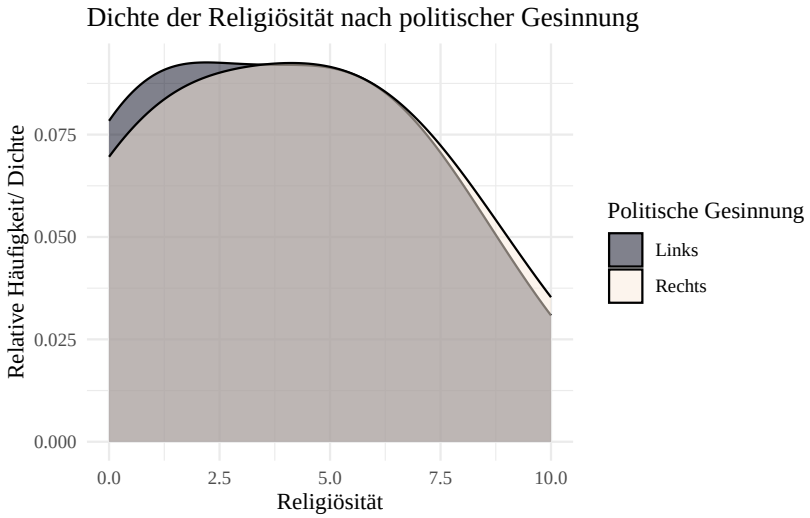

```
# Kohorte
df %>%
  group_by(cohort) %>%
  summarise(mean_value = mean(as.numeric(lrscale), na.rm = TRUE)) %>%
  ungroup() %>%
  ggplot(aes(x = cohort, y = mean_value, fill = factor(cohort))) +
  geom_bar(stat = "identity", position = position_dodge()) +
  scale_fill_viridis_d(option = "B") +
  labs(title = "Mittelwerte für Kohorten nach politischer Gesinnung",
       x = "Land",
       y = "Links/Rechts Mittelwert") +
  theme_minimal() +
  theme(text = element_text(family = "serif"),
        legend.position = "none")
```



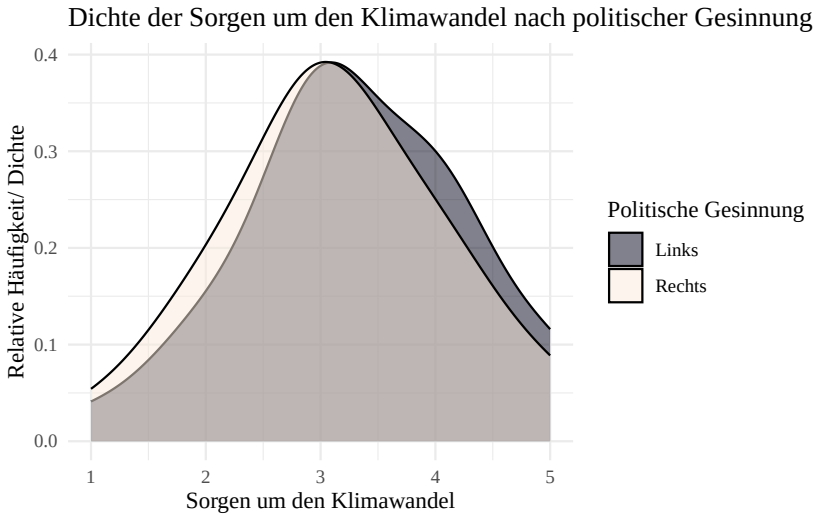

```
# Internetzugang
ggplot(df, aes(x = netusoft, fill = factor(dummy_lr, labels = c("Links", "Rechts")))) +
  geom_density(alpha = 0.5, adjust = 4) +
  scale_fill_viridis(option = "F" ,discrete=T) +
  labs(title = "Dichte für Internetzugang nach politischer Gesinnung",
       x = "Internetzugang",
       y = "Relative Häufigkeit/ Dichte",
       fill = "Politische Gesinnung") +
  theme_minimal() +
  theme(text = element_text(family = "serif"))
```



```r
#Religiösität
ggplot(df, aes(x = rlgdgr, fill = factor(dummy_lr, labels = c("Links", "Rechts")))) +
  geom_density(alpha = 0.5, adjust = 4) +
  scale_fill_viridis(option = "F" ,discrete=T) +
  labs(title = "Dichte der Religiösität nach politischer Gesinnung",
       x = "Religiösität",
       y = "Relative Häufigkeit/ Dichte",
       fill = "Politische Gesinnung") +
  theme_minimal() +
  theme(text = element_text(family = "serif"))
```




```r
# Klimawandel
ggplot(df, aes(x = wrclmch, fill = factor(dummy_lr, labels = c("Links", "Rechts")))) +
  geom_density(alpha = 0.5, adjust = 4) +
  scale_fill_viridis(option = "F" ,discrete=T) +
  labs(title = "Dichte der Sorgen um den Klimawandel nach politischer Gesinnung",
       x = "Sorgen um den Klimawandel",
       y = "Relative Häufigkeit/ Dichte",
       fill = "Politische Gesinnung") +
  theme_minimal() +
  theme(text = element_text(family = "serif"))
```



- *Schätzen Sie dann unterschiedliche Regressionsmodelle und vergleichen Sie sie mit verschiedenen Ansätzen (im Sinne einer in-sample und out-of-sample Modellvalidierung).*

Regressionsmodelle

OLS

Die klassische Methode der kleinsten Quadrate (OLS - Ordinary Least Squares) ist ein Verfahren zur Schätzung der unbekannten Parameter in einem linearen Regressionsmodell. OLS wählt die Parameter (Regressionskoeffizienten), die die Summe der quadrierten Residuen (Unterschiede zwischen beobachteten und durch das Modell vorhergesagten Werten) minimieren. Diese Methode wird aufgrund ihrer Einfachheit und der Tatsache, dass sie unter bestimmten Bedingungen die besten unverzerrten Schätzer liefert, häufig verwendet.

Die Schätzung erfolgt durch Minimierung der Quadratsumme der Residuen:

$$Q(\beta) = \sum_i (y_i - \beta_0 - \beta_1 x_{i1} - ... - \beta_m x_{im})^2$$

In Matrixnotation kann dies als

$$\hat{\beta} = (X^T X)^{-1} X^T Y$$

geschrieben werden,.

Logit

Im Gegensatz zu OLS, die auf ununterbrochene abhängige Variablen angewendet wird, wird die logistische Regression für binäre (0/1, Ja/Nein) Antworten verwendet. Die Wahrscheinlichkeit des Eintretens des Ereignisses wird durch die logistische Funktion modelliert, die auch als Sigmoid-Funktion bekannt ist:

$$P(Y = 1|X) = \frac{e^{X\beta}}{1 + e^{X\beta}}$$

Zur Schätzung der Koeffizienten verwendet die logistische Regression die Maximum-Likelihood-Methode, welche die Wahrscheinlichkeit, dass die beobachteten Daten unter dem gegebenen Modell auftreten, maximiert.

Lasso

Ridge und Lasso nehmen je einen L1 oder L2 Regularisierungsterm mit in das Modell auf.

Bei der Lasso-Regression wird die L1-Regularisierung angewendet, indem ein Strafterm zur Verlustfunktion hinzugefügt wird, der die absolute Summe der Regressionskoeffizienten beinhaltet. Das Ziel der Lasso-Regression ist es, sowohl den Vorhersagefehler zu minimieren als auch die Summe der absoluten Werte der Regressionskoeffizienten zu kontrollieren. Dieser Ansatz führt dazu, dass einige der Koeffizienten im Modell auf Null gesetzt werden, was gleichzeitig eine Art von Variablenselektion darstellt. Die Lasso-Regression ist besonders nützlich in Situationen, wo wir viele Prädiktoren haben, von denen einige möglicherweise irrelevant sind für die Vorhersage der Zielvariable

$\text{Loss}(\theta) = \text{Original Loss}(\theta) + \lambda_1 \sum |\theta_i|$

Ridge

Bei der Ridge-Regression wird die L2-Regularisierung verwendet, indem ein Strafterm zur Verlustfunktion hinzugefügt wird, der die quadrierten Werte der Regressionskoeffizienten summiert. Der Hauptzweck der Ridge-Regression ist es, die Größe der Koeffizienten zu kontrollieren, um Multikollinearität zu bekämpfen und Überanpassung zu reduzieren. Im Gegensatz zur Lasso-Regression, die Koeffizienten auf Null setzen kann, bewirkt die Ridge-Regression, dass die Koeffizienten kleiner werden, aber selten exakt Null.

$\text{Loss}(\theta) = \text{Original Loss}(\theta) + \lambda_2 \sum \theta_i^2$

Anwendung

Die Inkorporierung von L1 oder L2 Regularisierung in ein Regressionsmodell erfolgt durch die Auswahl des Regularisierungsparameters λ , der bestimmt, wie stark die Regularisierung sein soll. Die Verlustfunktion, die minimiert wird, um die Regressionskoeffizienten zu schätzen, umfasst sowohl den Term für den Vorhersagefehler (z.B. die Summe der quadrierten Residuen) als auch den Regularisierungsterm.

Die Bestimmung von λ (und gegebenenfalls α in elastischen Netzen, die eine Kombination aus L1 und L2 Regularisierung darstellen) ist daher ein wesentlicher Schritt, um das Gleichgewicht

zwischen Bias und Varianz im Modell zu finden und die Generalisierbarkeit der Modellvorhersagen zu maximieren. Ein optimales λ lässt sich durch Kreuzvalidierung bestimmen.

```
#Logistische Regression
log_m<-glm(dummy_lr ~  factor(gndr)+ netusoft + stflife + health + rlgdgr +  wrclmch + chl
           family = binomial(link = "logit"), data = df)

#OLS
lin <- lm(lrscale ~  factor(gndr)+ netusoft + stflife + health + rlgdgr +  wrclmch + chldo

# Modelmatrix für Ridge und Lasso Regression:
x <- model.matrix(lrscale ~ factor(gndr) + netusoft + stflife + health + rlgdgr + wrclmch
y <- df$lrscale

set.seed(102377) # Für reproduzierbare Ergebnisse
#Kreuzvalidierung um Lambda über verschiedene Stichproben zu bestimmen
cv.lasso <- cv.glmnet(x, y, alpha=1)
lasso <- glmnet(x, y, alpha=1, lambda=cv.lasso$lambda.min)
#Ridge Regression
cv.ridge <- cv.glmnet(x, y, alpha=0)
ridge <- glmnet(x, y, alpha=0, lambda=cv.ridge$lambda.min)


# Evaluation der Koeffizienten
stargazer(log_m, lin,
          type = "latex",
          single.row = T,
          title = "Lineare und Logistische Regressionsergebnisse",
          dep.var.labels.include = FALSE,
          star.cutoffs = c(0.05, 0.01, 0.001),
          star.char = c("*", "**", "***")
          ,style = "default"
          ,omit = "cntry"
)
```

% Table created by stargazer v.5.2.3 by Marek Hlavac, Social Policy Institute. E-mail: marek.hlavac at gmail.com % Date and time: Di, Feb 06, 2024 - 21:36:13

```
kable(data.frame(
  Lasso= as.vector(coef(lasso)),
  Ridge= as.vector(coef(ridge)),
  OLS= coef(lin),
```

Table 4: Lineare und Logistische Regressionsergebnisse

	Dependent variable:	
	logistic	*OLS*
	(1)	(2)
factor(gndr)2	-0.374^{***} (0.052)	-0.365^{***} (0.051)
netusoft	-0.050 (0.043)	-0.003 (0.042)
stflife	0.099^{***} (0.015)	0.129^{***} (0.014)
health	-0.128^{***} (0.039)	-0.072 (0.037)
rlgdgr	0.076^{***} (0.009)	0.101^{***} (0.009)
wrclmch	-0.177^{***} (0.028)	-0.265^{***} (0.027)
chldo12_dummy	-0.019 (0.089)	0.201^{*} (0.087)
factor(cohort)1990s	-0.094 (0.059)	-0.004 (0.058)
factor(cohort)2000s	-0.202^{**} (0.073)	-0.111 (0.071)
Constant	-0.264 (0.296)	4.784^{***} (0.290)
Observations	7,256	7,256
R^2		0.102
Adjusted R^2		0.098
Log Likelihood	$-4{,}507.837$	
Akaike Inf. Crit.	9,077.674	
Residual Std. Error		2.094 (df = 7225)
F Statistic		27.311^{***} (df = 30; 7225)
Note:		*p<0.05; **p<0.01; ***p<0.001

```
  LOG= coef(log_m)
),format = "latex")
```

	Lasso	Ridge	OLS	LOG
(Intercept)	4.8074251	4.9376710	4.7837005	-0.2635795
factor(gndr)2	-0.3500918	-0.3491165	-0.3652843	-0.3737563
netusoft	0.0000000	-0.0080938	-0.0033617	-0.0503025
stflife	0.1269281	0.1241477	0.1291231	0.0993495
health	-0.0604699	-0.0731468	-0.0720429	-0.1277114
rlgdgr	0.0964179	0.0954804	0.1005683	0.0755372
wrclmch	-0.2612578	-0.2579513	-0.2650787	-0.1774596
chldo12_dummy	0.1823674	0.1947835	0.2006309	-0.0193552
factor(cohort)1990s	0.0000000	-0.0070933	-0.0038698	-0.0944921
factor(cohort)2000s	-0.0943957	-0.1121919	-0.1109199	-0.2017970
factor(cntry)BG	1.0949872	1.0289169	1.1639840	0.4430619
factor(cntry)CH	-0.2149758	-0.2849950	-0.1937811	-0.0886885
factor(cntry)CZ	1.2160465	1.1456416	1.2963444	0.8335348
factor(cntry)EE	0.2529627	0.2255278	0.3346327	-0.0466822
factor(cntry)FI	0.2298596	0.2032540	0.3124870	0.2396175
factor(cntry)FR	-0.1117021	-0.1783228	-0.0822117	-0.2321259
factor(cntry)GB	-0.3772792	-0.4544002	-0.3543155	-0.4833118
factor(cntry)GR	-0.1286203	-0.1995193	-0.1084394	-0.7524315
factor(cntry)HR	-0.1487171	-0.2185080	-0.1309859	-0.7811690
factor(cntry)HU	0.9806471	0.9193656	1.0605159	0.6167856
factor(cntry)IE	-0.0631393	-0.1455677	-0.0467818	-0.3980947
factor(cntry)IS	-0.3532757	-0.4245400	-0.3370562	-0.1377734
factor(cntry)IT	0.0876377	0.0556554	0.1601452	-0.0217747
factor(cntry)LT	0.4329950	0.3987656	0.5118276	-0.1034065
factor(cntry)ME	-1.2242914	-1.2683012	-1.2166408	-0.8771997
factor(cntry)MK	-0.0622256	-0.1471546	-0.0624021	-0.2390122
factor(cntry)NL	0.1485463	0.1231200	0.2291083	0.3486609
factor(cntry)NO	0.0563811	0.0332297	0.1416344	0.1816745
factor(cntry)PT	-0.0674860	-0.1439713	-0.0448291	-0.3885649
factor(cntry)SI	0.0000000	-0.0500235	0.0462743	-0.7114235
factor(cntry)SK	0.4915961	0.4524940	0.5645967	0.2788649

In Sample Evaluation:

Das binäre Modell wird durch eine Konfusionsmatrix evaluiert.

	Vorhergesagt Positiv	Vorhergesagt Negativ
Wahr Positiv	TP	FN
Wahr Negativ	FP	TN

Für die Vergleiche zwischen kontinuierliche Modelle wird der Anfangsbeschriebene mittlere quadratische Fehler verwendet. Zur Erinnerung:

$$\text{MSE}(\hat{\theta}) = E[(\hat{\theta} - \theta)^2$$

```r
# IN SAMPLE prediction für das Klassifikations-Modell
pred_log <- predict(log_m, type = "response")
pred_log <- ifelse(pred_log > 0.5, 1, 0)

cm_is<- confusionMatrix(data = factor(pred_log),
                reference = factor(df$dummy_lr),mode = "everything", positive = "1")
cm_is[2]
```

```
$table
          Reference
Prediction    0    1
         0 4061 2003
         1  535  657
```

```r
# IN SAMPLE prediction für die Linearen Modelle
pred_lin <- predict(lin)
pred_ridge <- predict(ridge, newx = x)
pred_lasso <- predict(lasso, newx = x)


mse_lin <- mean((pred_lin - df$lrscale)^2)
mse_lasso <- mean((pred_lasso - df$lrscale)^2)
mse_ridge <- mean((pred_ridge - df$lrscale)^2)


kable(data.frame(
  Lin = mse_lin,
  Lasso = mse_lasso,
  Ridge = mse_ridge), format = "latex", caption= "In Sample MSE-Werte")
```

Table 5: In Sample MSE-Werte

Lin	Lasso	Ridge
4.364394	4.365523	4.365538

Out of Sample Evaluation

Hier nehmen wir die 20 %, welche nicht in den Trainingsdaten enthalten sind. Sie simulieren nun bisher ungesehene Daten.

```
df <- initial_split(prop = 0.8, data=df) %>% training(.)
```

wird zu:

```
test<- initial_split(prop = 0.8, data=df) %>% testing(.)
```

```
set.seed(102377) # stellt sicher dass wir die gleiche Aufteilung der Daten haben
test<- initial_split(prop = 0.8, data=df) %>% training(.)


# OUT OF SAMPLE prediction für das Klassifikations-Modell
pred_log <- predict(log_m, type = "response", newdata = test)
pred_log <- ifelse(pred_log > 0.5, 1, 0)

cm_oos<- confusionMatrix(data = factor(pred_log),
                reference = factor(test$dummy_lr),mode = "everything", positive = "1")
cm_oos[2]
```

```
$table
          Reference
Prediction    0    1
         0 3269 1573
         1  443  519
```

```
# OUT OF SAMPLE Prediction für die Linearen Modelle
pred_lin <- predict(lin, newdata = test)


# Setup für Ridge und Lasso
x_test <- model.matrix(~ factor(gndr) + netusoft + stflife + health + rlgdgr + wrclmch + c
# OUT OF SAMPLE Prediction für Ridge und Lasso
pred_ridge <- predict(ridge, newx = x_test, s = cv.ridge$lambda.min)
```

Table 6: Out of Sample MSE-Werte

Lin	Lasso	Ridge
4.307819	4.309691	4.308453

```
pred_lasso <- predict(lasso, newx = x_test, s = cv.lasso$lambda.min)


mse_lin <- mean((pred_lin - test$lrscale)^2)
mse_lasso <- mean((pred_lasso - test$lrscale)^2)
mse_ridge <- mean((pred_ridge - test$lrscale)^2)

kable(data.frame(
  Lin = mse_lin,
  Lasso = mse_lasso,
  Ridge = mse_ridge), format = "latex", caption= "Out of Sample MSE-Werte")
```

- *Diskutieren Sie Ihre Ergebnisse.*

Diskussion

Die Visualisierung hat bedeutende Unterschiede entlang des Geschlechts und zwischen Ländern, auf die politische Gesinnung aufgezeigt. Erst vor kurzem hat die Financial Times eine Studie veröffentlicht, nach der die Schere bei den politischen Ansichten und Einstellungen zwischen Männern und Frauen zwischen 18 und 29 Jahren seit Jahrzehnten immer weiter auseinander geht.

Für die kontinuierlichen Variablen wurden in den Dichte-Plots subtilere Unterschiede beobachtet - die dennoch bedeutsam sein können. Der deutlichste Unterschied war im Vertrauen in das nationale Parlament zu sehen. Dies ist unter der Annahme, dass rechte Personen patriotischer sind, wenig überraschend. Dennoch ist besonders im deutschen Raum oft eher eine Entfremdung von der lokalen Regierung mit rechter Ideologie assoziiert. Um etwas mehr Sicherheit in solche Überlegungen zu bringen ist es wichtig für konfundierende Variablen zu kontrollieren. Regressionsmodelle sind dazu zwar auch nur bedingt in der Lage, erlauben aber in jedem Fall eine fundiertere Inferenz als absolute Häufigkeiten.

Die verschiedenen Regressionsmodelle erzeugen sehr ähnliche Koeffizienten. Einige Features erweisen sich hier als nicht Signifikant, trotz visueller Unterschiede, wie zum Beispiel Internetzugang. Dennoch sind die meisten Koeffizienten signifikant! Überraschend ist der starke positive Zusammenhang zwischen Lebenszufriedenheit und rechter Gesinnung. Eine gängige Annahme ist das rechte ideologie eine Reaktion auf Unzufriedenheiten sei, wie zum Beispiel bei negativer intergenerationaler sozialer Mobilität.

Laut der Financial Times Studie ist besonders Gen-Z von politischen Unterschieden entlang des Geschlechts betroffen. Zwischen den Kohorten im ESS sind jedoch kaum unterschiede zu beobachten.

Mit 0.102 ist der Anteil der erklärten Varianz zwar gering, was in sozial wissenschaftliche Fragestellungen jedoch kein Problem darstellt, sofern die meisten Features statistisch signifikant sind (Ozili, Person K, 2023, The acceptable R-square in empirical modelling for social science research, Munich Personal RePEc Archive).

Die Lasso und Ridge Regression veringern nur den Beitrag weniger Features. Eigentlich sollte die stärke dieser beiden Modelle in der Out of Sample Evaluation zum Vorschein kommen, da der Zweck des Regularisierungsterms, wie der Name sagt, ein regularisierbares Ergebniss ist. Hier unterscheiden die Werte sich jedoch kaum von der In Sample Evaluation. Das OLS Modell wäre mit dem geringsten MSE sogar vorzuziehen. Mit einer höheren Featureanzahl wären die Vorteile der Regularisierung aber warscheinlich beobachtbar.

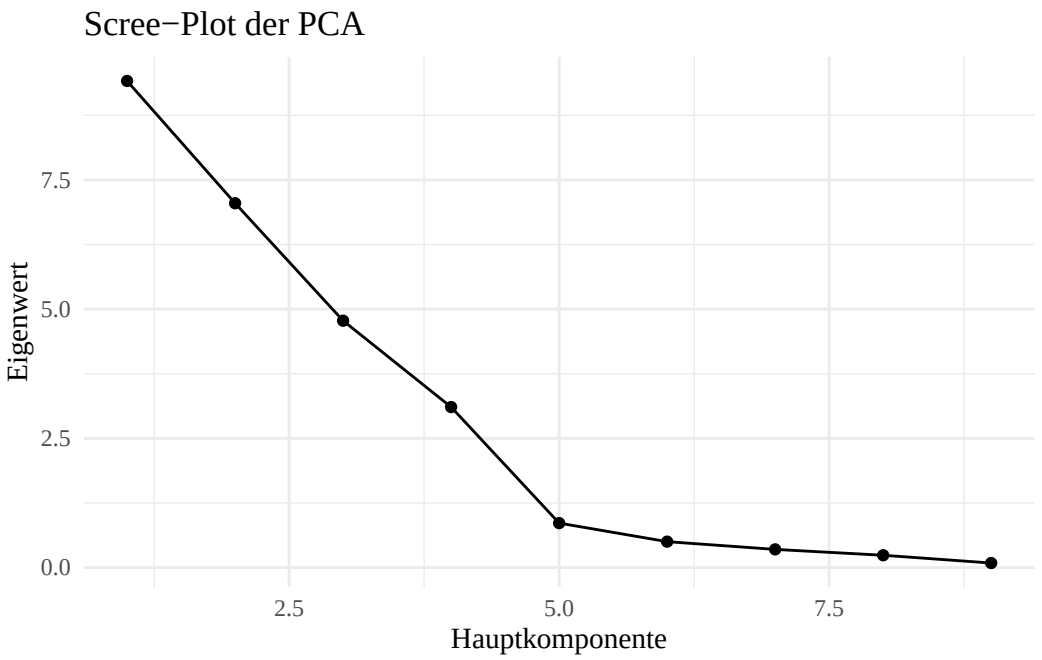
Hauptkomponentenanalyse - PCA (5P)

Hier benötigen Sie wieder einen -gern selbst gewählten- Datensatz, dieses Mal wären möglichst viele quantitativ-stetige Variablen von Vorteil. Sie sollen hier die Dimensionalität Ihrer Daten diskutieren.

- *Sie sollten ebenfalls zunächst kurz eine explorative (grafische) Analyse Ihrer Variablen vornehmen. Legen Sie dabei besonderes Augenmerk auf Ausreißer und ggf. engere Zusammenhänge (z.B. Korrelationen) zwischen den Variablen.*
- *Führen Sie nun eine Hauptkomponentenanalyse durch, stellen Sie die Ergebnisse grafisch dar (inkl. Screeplot der Eigenwerte) und diskutieren Sie die Ergebnisse. Versuchen Sie die wichtigsten Hauptkomponenten zu interpretieren.*

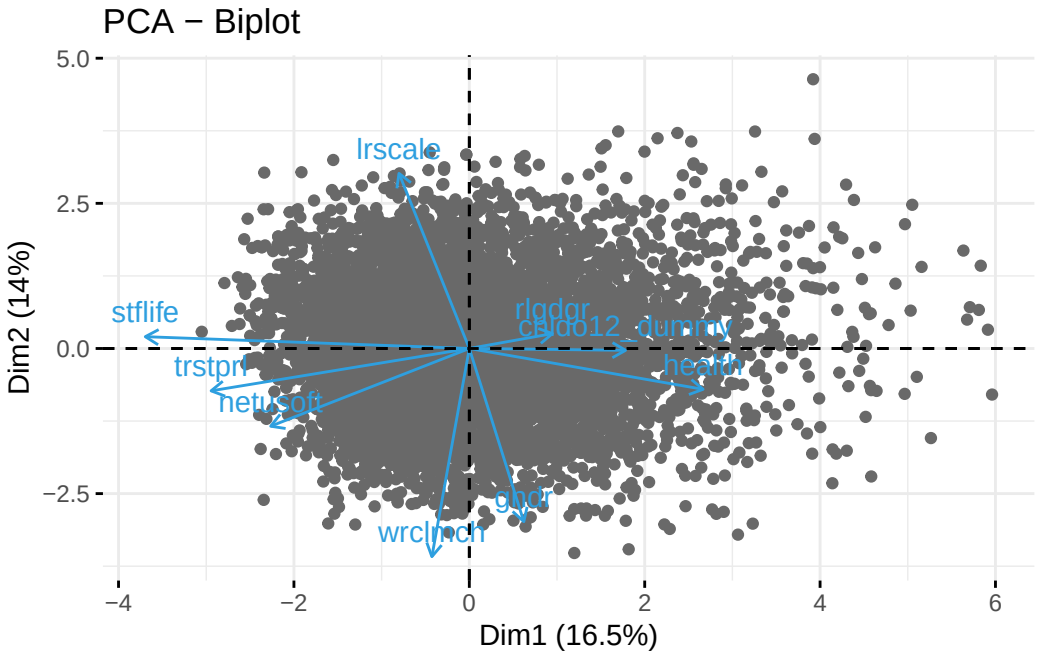
```
# Selbstgemachte Hauptkomponenten Analyse
x <- df %>% select(-cntry,-cohort,-dummy_lr)

s <- cov(x) *(n-1)/n ## Kovarianzmatrix - hier mit Faktor 1/n

e <- eigen(s) ## Eigenwerte und -vektoren berechnen

gamma <- e$vectors ## Matrix der Eigenvektoren (Spalten von gamma) = loadings

lambda <- e$values ## Vector der Eigenwerte

# # erzeugt wieder die Kovarianzmatrix:
# gamma %*% diag(lambda) %*% t(gamma)
```

```r
## Visualisierung

# Erstellen eines Dataframes für ggplot
eigen_values_df <- data.frame(Komponente = seq_along(lambda), Eigenwert = lambda)

# Erstellen des Scree-Plots
ggplot(eigen_values_df, aes(x = Komponente, y = Eigenwert)) +
  geom_line() + geom_point() +
  theme_minimal() +
  theme(text = element_text(family = "serif"))+
  labs(title = "Scree-Plot der PCA", x = "Hauptkomponente", y = "Eigenwert")
```




```r
# R's funktion um die PCA direkt durchzuführen
comps <- prcomp(x = x,
                center = T,
                scale. = T)

fviz_pca_biplot(comps, repel = F,
                col.var = "#2E9FDF", # Variables color
```

```
              col.ind = "#696969",  # Individuals color
              label = "var"
              )
```



Screeplot

Der Scree-Plot ist ein Diagramm, das die Eigenwerte abbildet, welche die Varianz darstellen, die von jeder Hauptkomponente aufgenommen wird . Es hilft bei der Entscheidung, wie viele Hauptkomponenten für weitere Analysen behalten werden sollten. Typischerweise sucht man nach einem “Ellbogen” im Plot. Aus dem Screeplot geht hervor, dass unser numerischer Datenanteil circa 4 Dimensionen hat.

Biplot

Ein PCA-Biplot ist eine Darstellung, die sowohl die Position der Beobachtungen in Bezug auf die ersten zwei Hauptkomponenten als auch die Ladungen (Eigenvektoren) der Variablen auf diesen Hauptkomponenten zeigt. Anders gesagt lässt sich ablesen wie die Variablen zu den Dimensionen beitragen.

Session Info

```
R version 4.2.2 (2022-10-31 ucrt)
Platform: x86_64-w64-mingw32/x64 (64-bit)
Running under: Windows 10 x64 (build 19045)

Matrix products: default

locale:
[1] LC_COLLATE=German_Germany.utf8  LC_CTYPE=German_Germany.utf8
[3] LC_MONETARY=German_Germany.utf8 LC_NUMERIC=C
[5] LC_TIME=German_Germany.utf8

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods   base

other attached packages:
 [1] factoextra_1.0.7   FactoMineR_2.8     viridis_0.6.3      viridisLite_0.4.2
 [5] hrbrthemes_0.8.0   stargazer_5.2.3    glmnet_4.1-7       Matrix_1.5-1
 [9] kableExtra_1.3.4   caret_6.0-94       lattice_0.20-45    yardstick_1.2.0
[13] workflowsets_1.0.1 workflows_1.1.3    tune_1.1.1         rsample_1.1.1
[17] recipes_1.0.6      parsnip_1.1.0      modeldata_1.1.0    infer_1.0.4
[21] dials_1.2.0        scales_1.2.1       broom_1.0.4        tidymodels_1.1.0
[25] arrow_14.0.0.1     lubridate_1.9.2    forcats_1.0.0      stringr_1.5.0
[29] dplyr_1.1.2        purrr_1.0.1        readr_2.1.4        tidyr_1.3.0
[33] tibble_3.2.1       ggplot2_3.4.2      tidyverse_2.0.0

loaded via a namespace (and not attached):
  [1] backports_1.4.1         systemfonts_1.0.4       plyr_1.8.8
  [4] splines_4.2.2           listenv_0.9.0           TH.data_1.1-2
  [7] digest_0.6.31           foreach_1.5.2           htmltools_0.5.5
 [10] fansi_1.0.4             magrittr_2.0.3          cluster_2.1.4
 [13] tzdb_0.3.0              globals_0.16.2          gower_1.0.1
 [16] extrafont_0.19          sandwich_3.0-2          extrafontdb_1.0
 [19] svglite_2.1.1           hardhat_1.3.0           timechange_0.2.0
 [22] gfonts_0.2.0            colorspace_2.1-0        ggrepel_0.9.3
 [25] rvest_1.0.3             xfun_0.39               crayon_1.5.2
 [28] jsonlite_1.8.4          zoo_1.8-12              survival_3.4-0
 [31] iterators_1.0.14        glue_1.6.2              gtable_0.3.3
 [34] emmeans_1.8.5           ipred_0.9-14            webshot_0.5.4
 [37] car_3.1-2               Rttf2pt1_1.3.12         future.apply_1.10.0
 [40] shape_1.4.6             abind_1.4-5             fontquiver_0.2.1
```

```
 [43] mvtnorm_1.1-3           rstatix_0.7.2          Rcpp_1.0.10
 [46] xtable_1.8-4            proxy_0.4-27           flashClust_1.01-2
 [49] GPfit_1.0-8             bit_4.0.5              DT_0.27
 [52] stats4_4.2.2            lava_1.7.2.1           fontLiberation_0.1.0
 [55] prodlim_2023.03.31      htmlwidgets_1.6.2      httr_1.4.6
 [58] ellipsis_0.3.2          farver_2.1.1           pkgconfig_2.0.3
 [61] multcompView_0.1-9      nnet_7.3-18            utf8_1.2.3
 [64] crul_1.3                labeling_0.4.2         tidyselect_1.2.0
 [67] rlang_1.1.1             DiceDesign_1.9         reshape2_1.4.4
 [70] later_1.3.1             munsell_0.5.0          tools_4.2.2
 [73] cli_3.6.1               generics_0.1.3         evaluate_0.21
 [76] fastmap_1.1.1           yaml_2.3.7             ModelMetrics_1.2.2.2
 [79] knitr_1.42              bit64_4.0.5            future_1.32.0
 [82] nlme_3.1-162            mime_0.12              leaps_3.1
 [85] xml2_1.3.4              compiler_4.2.2         rstudioapi_0.15.0
 [88] curl_5.0.0              ggsignif_0.6.4         e1071_1.7-13
 [91] lhs_1.1.6               stringi_1.7.12         gdtools_0.3.3
 [94] fontBitstreamVera_0.1.1 vctrs_0.6.2            pillar_1.9.0
 [97] lifecycle_1.0.3         furrr_0.3.1            estimability_1.4.1
[100] data.table_1.14.8       httpuv_1.6.10          R6_2.5.1
[103] promises_1.2.0.1        gridExtra_2.3          parallelly_1.35.0
[106] codetools_0.2-18        MASS_7.3-58.1          assertthat_0.2.1
[109] withr_2.5.0             httpcode_0.3.0         multcomp_1.4-23
[112] parallel_4.2.2          hms_1.1.3              grid_4.2.2
[115] rpart_4.1.19            timeDate_4022.108      coda_0.19-4
[118] class_7.3-20            rmarkdown_2.21         carData_3.0-5
[121] ggpubr_0.6.0            pROC_1.18.0            scatterplot3d_0.3-44
[124] shiny_1.7.4
```