Diverging Memories: Emotive Representation of East Germany and its Communist Past in the German Bundestag

Konrat Pekkip

2022-07-26

Introduction

The following contains the majority of the code supporting my M.A. thesis, submitted to the University of Chicago's Division of the Social Sciences in July of 2022. In my thesis, I conduct a sentiment analysis of all speeches given in the German *Bundestag* since German reunification in 1990. The question underlying this thesis is whether or not there are meaningful differences in the sentiments with which members of the German *Bundestag* (hereafter referred to as MdB) refer to East Germany as a region, its communist past, and the institutions of the German Democratic Republic (GDR). Please note that in order to run the code below, you have to first run the data cleaning and preprocessing steps laid out here and import the resulting data frame, `full_data.csv`. Furthermore, in order to read in the SentiMerge sentiment dictionary I use for the purposes of this research, you need to download its contents here. If you have trouble accessing any of the data or if you have any questions about the code, feel free to reach out to me at kpekkip@uchicago.edu.

Loading Packages and Data

The following three code chunks load required packages and read in all relevant datasets as well as the SentiMerge sentiment dictionary.

```
#load packages
library(tidyverse)
library(tidytext)
library(corpus)
library(quanteda)
library(quanteda.sentiment)
library(quanteda.textplots)
library(readtext)
library(spacyr)
library(pradadata)
library(DT)
library(sandwich)
library(lmtest)
library(robustbase)
library(psych)
```

```
#load preprocessed and cleaned data frame
full_data <- read_csv("../data/full_data.csv")

#subset only metadata
speeches_meta <- full_data %>%
  select(-speechContent) %>%
  mutate(docname = as.character(id))
```

```
#load sentiment dictionary
sentiment_dict <- read.delim("../data/sentimerge.txt") %>%
  rename(word = lemma) %>%
  mutate(sentweight = sentiment * weight)

#convert sentiment dictionary to quanteda dictionary object
qtd_sentiment_dict <- sentiment_dict %>%
  as.dictionary()
```

Full Data Tokenization

In the next code chunk, I tokenize the full speeches, remove stopwords, and unnest tokens in order to create a data frame consisting of one token per row – the tidytext format. I then merge the resulting data frame with the sentiment dictionary, assigning each token a sentiment score and calculating average sentiment scores per speaker as well as per speaker and electoral term. Note that the sentiment score averages calculated here are based on all speeches given in the German *Bundestag* in the desired time frame, resulting in baseline scores for each lawmaker and electoral term. These baseline scores will later be used to calculate relative sentiment averages for each topic.

```
#define german stopwords
german_stopwords <- data.frame(word = stopwords("de"), stringsAsFactors = FALSE)

#tokenize full speeches
full_df <- full_data %>%
  select(id, party, born_system, speechContent, year, politicianId, electoralTerm, born_gdr)

#unnest tokens
full_df <- unnest_tokens(tbl = full_df,
                         output = word,
                         input = speechContent)

#remove stopwords
full_df <- full_df %>%
  anti_join(german_stopwords, by = c("word"))
```

```
#merge with sentiment dictionary and calculate average sentiments per speaker and electoral term
full_df <- inner_join(full_df,
                      sentiment_dict,
                      by = "word") %>%
  group_by(id) %>%
  mutate(avg_sentiment = mean((sentiment * weight))) %>%
  ungroup() %>%
  group_by(politicianId) %>%
  mutate(speaker_sent_full = mean((sentiment * weight))) %>%
  ungroup() %>%
  group_by(politicianId, electoralTerm) %>%
  mutate(speaker_session_sent = mean((sentiment * weight)))
```

Subset Data Tokenizing

The `quanteda` package contains powerful tools for natural language processing in R, and in order to utilize it, the code below turns the existing data frame into a `quanteda` corpus object. The data frame containing full speeches is once again tokenized into unigrams (except for compound words), while stopwords are being removed. Note that the `padding` parameter can be set to `TRUE` or `FALSE` based on your preference.

```
#create vector of compound words for tokenization
compound_words <- c("Deutsche Demokratische Republik*", "Freie Deutsche Jugend*", "Arbeiter und Bauern S

#create quanteda corpus
qtd_speeches_corp <- corpus(full_data,
                            docid_field = "id",
                            text_field = "speechContent")
```

```
## Warning: NA is replaced by empty string
```

```
#tokenize speeches
qtd_tokens <- tokens(qtd_speeches_corp,
                     remove_punct = TRUE,
                     remove_numbers = TRUE,
                     remove_symbols = TRUE,
                     padding = FALSE,
                     split_hyphens = TRUE,
                     include_docvars = TRUE) %>%
  tokens_remove(stopwords("de"), padding = TRUE) %>%
  tokens_compound(pattern = phrase(compound_words), concatenator = " ")
```

Locating Keywords-In-Context

In the next few chunks of code, I locate context-specific keywords in all speeches, and extract a window of five words preceding and succeeding each keyword (10 words total per keyword). The resulting data frames containing the keywords-in-context are merged with the respective metadata and converted to the tidytext format by unnesting tokens. Finally, I calculate average sentiment scores by speaker and electoral term to retrieve topic-specific sentiment scores per legislator. These steps are taken with regard to GDR-specific keywords in the first code chunk, and repeated with regard to East Germany-specific keywords and East German *Bundesländer*-specific keywords in the second and third code chunk, respectively.

```
#define keywords
ddr_words <- c("DDR", "D.D.R.", "Deutsche Demokratische Republik", "SED", "S.E.D.", "Stasi", "Staatssich

#identify keywords-in-context
ddr_df <- kwic(qtd_tokens,
               pattern = ddr_words,
               valuetype = "fixed",
               window = 5) #you can change the window parameter to add more/fewer words

#merge with metadata
ddr_df <- left_join(ddr_df, speeches_meta, by = "docname")

#join pre and post columns into one
ddr_df <- ddr_df %>%
  as_tibble() %>%
  unite(col = "merged",
        pre, post,
```

```
        sep = " ")

#tokenize in tidytext format
ddr_df <- unnest_tokens(tbl = ddr_df,
                        output = word,
                        input = merged)

#merge with sentiment dictionary, calculate legislator-session sentiment scores
ddr_df <- inner_join(ddr_df,
                     sentiment_dict,
                     by = "word") %>%
  group_by(id) %>%
  mutate(avg_sentiment = mean((sentiment * weight))) %>%
  ungroup() %>%
  group_by(politicianId) %>%
  mutate(speaker_sent_ddr = mean((sentiment * weight))) %>%
  ungroup() %>%
  group_by(politicianId, electoralTerm) %>%
  mutate(speaker_session_sent_ddr = mean((sentiment * weight)))
```

```
#define keywords
ostdeutschland_words <- c("Neue Bundesländer", "Neue Länder", "Neuen Ländern", "Ostdeutschland", "der Os

#identify keywords-in-context
ostdeutschland_df <- kwic(qtd_tokens,
                          pattern = ostdeutschland_words,
                          valuetype = "fixed",
                          window = 5)

#merge with metadata
ostdeutschland_df <- left_join(ostdeutschland_df, speeches_meta, by = "docname")

#join pre and post columns into one
ostdeutschland_df <- ostdeutschland_df %>%
  as_tibble() %>%
  unite(col = "merged",
        pre, post,
        sep = " ")

#tokenize in tidytext format
ostdeutschland_df <- unnest_tokens(tbl = ostdeutschland_df,
                                   output = word,
                                   input = merged)

#merge with sentiment dictionary, calculate legislator-session sentiment scores
ostdeutschland_df <- inner_join(ostdeutschland_df,
                                sentiment_dict,
                                by = "word") %>%
  group_by(id) %>%
  mutate(avg_sentiment = mean((sentiment * weight)))%>%
  ungroup() %>%
  group_by(politicianId) %>%
  mutate(speaker_sent_ostdeutschland = mean((sentiment * weight))) %>%
  ungroup() %>%
```

```
  group_by(politicianId, electoralTerm) %>%
  mutate(speaker_session_sent_ostdeutschland = mean((sentiment * weight)))
```

```
#define keywords
bundesland_words <- c("Brandenburg", "Thüringen", "Sachsen", "Anhalt", "Mecklenburg", "Vorpommern")

#identify keywords-in-context
bundesland_df <- kwic(qtd_tokens,
                      pattern = bundesland_words,
                      valuetype = "fixed",
                      window = 5)

#merge with metadata
bundesland_df <- left_join(bundesland_df, speeches_meta, by = "docname")

#join pre and post columns into one
bundesland_df <- bundesland_df %>%
  as_tibble() %>%
  unite(col = "merged",
        pre, post,
        sep = " ")

#tokenize in tidytext format
bundesland_df <- unnest_tokens(tbl = bundesland_df,
                               output = word,
                               input = merged)

#merge with sentiment dictionary, calculate legislator-session sentiment scores
bundesland_df <- inner_join(bundesland_df,
                            sentiment_dict,
                            by = "word") %>%
  group_by(id) %>%
  mutate(avg_sentiment = mean((sentiment * weight))) %>%
  ungroup() %>%
  group_by(politicianId) %>%
  mutate(speaker_sent_bundesland = mean((sentiment * weight))) %>%
  ungroup() %>%
  group_by(politicianId, electoralTerm) %>%
  mutate(speaker_session_sent_bundesland = mean((sentiment * weight)))
```

Regression Analysis

The following code centers around linear regression analysis of relative sentiment scores, i.e. the difference between a lawmakers average sentiment towards a given topic and a lawmakers overall/baseline average sentiment scores per electoral term. First, I select and factorize relevant independent and dependent variables and normalize average sentiment scores per term. In a second step, I run heteroskedasticity-robust linear regression analyses, once using `party` as an independent variables, and once using `partygroup` as an independent variable, grouping together all "mainstream" parties and testing the effects on the extreme right and left. These steps are first performed on the group of keywords pertaining to the GDR and its institutions, and then repeated on keywords relating to East Germany as a region and East German *Bundesländer*.

```
#subset full data to include one row per mp-session
full_prejoin <- full_df %>%
```

```
  unite(col = mp_session, electoralTerm, politicianId, sep = "_", remove = FALSE) %>%
  select(politicianId, electoralTerm, speaker_session_sent, mp_session) %>%
  distinct(mp_session, .keep_all = TRUE)
```

```
#regression using ddr sentiment subset
ddr_prejoin <- ddr_df %>%
  unite(col = mp_session, electoralTerm, politicianId, sep = "_", remove = FALSE) %>%
  select(politicianId, electoralTerm, speaker_session_sent_ddr, mp_session, party, born_system, born_gdr
  distinct(mp_session, .keep_all = TRUE)

ddr_normed <- left_join(ddr_prejoin, full_prejoin, by = "mp_session")

ddr_normed[c("speaker_session_sent_ddr","speaker_session_sent")] <- lapply(ddr_normed[c("speaker_session

ddr_normed <- ddr_normed %>%
  mutate(norm_sent = (speaker_session_sent_ddr - speaker_session_sent)) %>%
  mutate(partygroup = case_when(party == "CDU/CSU" | party == "FDP" | party == "SPD" | party == "Bündnis
                                party == "AfD" ~ "AfD",
                                party == "PDS/Die Linke" ~ "PDS/Die Linke"))

partygroupx = fct_relevel(as.factor(ddr_normed$partygroup), "Mainstream", after = 0)

partyx = fct_relevel(as.factor(ddr_normed$party), "PDS/Die Linke", after = 0)

born_gdrx = fct_relevel(as.factor(ddr_normed$born_gdr), "Elsewhere", after = 0)

electoralTermx = as.factor(ddr_normed$electoralTerm.y)

#(heteroskedasticity-robust) linear regression model with party specification

model_pddr <- lm(norm_sent ~ partyx + born_gdrx + electoralTermx, data = ddr_normed)

model_pddr_sum <- summary(model_pddr)

model_pddr_robust_clustered <- coeftest(model_pddr,
                                        vcov = vcovCL,
                                        type = "HC1",
                                        cluster = ~politicianId.x)

#(heteroskedasticity-robust) linear regression model with partygroup specification

model_pgddr <- lm(norm_sent ~ partygroupx + born_gdrx + electoralTermx, data = ddr_normed)

model_pgddr_sum <- summary(model_pgddr)

model_pgddr_robust_clustered <- coeftest(model_pgddr,
                                        vcov = vcovCL,
                                        type = "HC1",
                                        cluster = ~politicianId.x)

#print regression outputs

model_pddr_robust_clustered
```

```
##
## t test of coefficients:
##
##                                 Estimate Std. Error t value  Pr(>|t|)
## (Intercept)                     1.162435   0.117551  9.8887 < 2.2e-16 ***
## partyxAfD                       0.379668   0.190382  1.9942   0.04626 *
## partyxBündnis 90/Die Grünen    -0.311668   0.158719 -1.9636   0.04971 *
## partyxCDU/CSU                  -1.128047   0.115052 -9.8047 < 2.2e-16 ***
## partyxFDP                      -1.086382   0.133189 -8.1567 5.988e-16 ***
## partyxSPD                      -0.830931   0.116787 -7.1149 1.550e-12 ***
## born_gdrxEast Germany (not GDR) -0.223923   0.140057 -1.5988   0.11002
## born_gdrxGDR                   -0.094434   0.079305 -1.1908   0.23389
## electoralTermx13                0.046130   0.085867  0.5372   0.59117
## electoralTermx14               -0.147351   0.098443 -1.4968   0.13460
## electoralTermx15               -0.279266   0.140924 -1.9817   0.04765 *
## electoralTermx16               -0.633307   0.090827 -6.9726 4.200e-12 ***
## electoralTermx17               -0.408076   0.096631 -4.2230 2.517e-05 ***
## electoralTermx18               -0.929581   0.126917 -7.3243 3.455e-13 ***
## electoralTermx19               -0.779019   0.117324 -6.6399 4.024e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
model_pgddr_robust_clustered
```

```
##
## t test of coefficients:
##
##                                 Estimate Std. Error t value  Pr(>|t|)
## (Intercept)                     0.167096   0.059436  2.8114 0.0049807 **
## partygroupxAfD                  1.349027   0.165064  8.1727 5.261e-16 ***
## partygroupxPDS/Die Linke        0.948879   0.108315  8.7603 < 2.2e-16 ***
## born_gdrxEast Germany (not GDR) -0.240261   0.136042 -1.7661 0.0775331 .
## born_gdrxGDR                   -0.077423   0.080334 -0.9638 0.3352787
## electoralTermx13                0.104275   0.086198  1.2097 0.2265313
## electoralTermx14               -0.087524   0.098409 -0.8894 0.3738986
## electoralTermx15               -0.217312   0.139292 -1.5601 0.1188897
## electoralTermx16               -0.588750   0.093591 -6.2907 3.865e-10 ***
## electoralTermx17               -0.378505   0.099614 -3.7997 0.0001492 ***
## electoralTermx18               -0.878294   0.130516 -6.7294 2.212e-11 ***
## electoralTermx19               -0.755531   0.116080 -6.5087 9.534e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#regression using ostdeutschland sentiment subset
ostdeutschland_prejoin <- ostdeutschland_df %>%
  unite(col = mp_session, electoralTerm, politicianId, sep = "_", remove = FALSE) %>%
  select(politicianId, electoralTerm, speaker_session_sent_ostdeutschland, mp_session, party, born_gdr)
  distinct(mp_session, .keep_all = TRUE)

ostdeutschland_normed <- left_join(ostdeutschland_prejoin, full_prejoin, by = "mp_session")

ostdeutschland_normed[c("speaker_session_sent_ostdeutschland","speaker_session_sent")] <- lapply(ostdeu

ostdeutschland_normed <- ostdeutschland_normed %>%
  mutate(norm_sent = (speaker_session_sent_ostdeutschland - speaker_session_sent)) %>%
```

```
  mutate(partygroup = case_when(party == "CDU/CSU" | party == "FDP" | party == "SPD" | party == "Bündnis
                                party == "AfD" ~ "AfD",
                                party == "PDS/Die Linke" ~ "PDS/Die Linke"))

partygroupx = fct_relevel(as.factor(ostdeutschland_normed$partygroup), "Mainstream", after = 0)

partyx = fct_relevel(as.factor(ostdeutschland_normed$party), "PDS/Die Linke", after = 0)

born_gdrx = fct_relevel(as.factor(ostdeutschland_normed$born_gdr), "Elsewhere", after = 0)

electoralTermx = as.factor(ostdeutschland_normed$electoralTerm.y)

##(heteroskedasticity-robust) linear regression model with party specification

model_postdeutschland <- lm(norm_sent ~ partyx + born_gdrx + electoralTermx, data = ostdeutschland_norme

model_postdeutschland_sum <- summary(model_postdeutschland)

model_postdeutschland_robust_clustered <- coeftest(model_postdeutschland,
                                    vcov = vcovCL,
                                    type = "HC1",
                                    cluster = ~politicianId.x)


#(heteroskedasticity-robust) linear regression model with partygroup specification

model_pgostdeutschland <- lm(norm_sent ~ partygroupx + born_gdrx + electoralTermx, data = ostdeutschlan

model_pgostdeutschland_sum <- summary(model_pgostdeutschland)

model_pgostdeutschland_robust_clustered <- coeftest(model_pgostdeutschland,
                                    vcov = vcovCL,
                                    type = "HC1",
                                    cluster = ~politicianId.x)

#print regression outputs

model_postdeutschland_robust_clustered
```

```
##
## t test of coefficients:
##
##                                  Estimate Std. Error t value  Pr(>|t|)
## (Intercept)                     1.0761210  0.1228812  8.7574 < 2.2e-16 ***
## partyxAfD                      -0.0918397  0.3852280 -0.2384  0.811597
## partyxBündnis 90/Die Grünen    -0.3694894  0.1426750 -2.5897  0.009688 **
## partyxCDU/CSU                  -1.0826503  0.1191729 -9.0847 < 2.2e-16 ***
## partyxFDP                      -1.0303230  0.1342668 -7.6737 2.812e-14 ***
## partyxSPD                      -0.9457323  0.1194537 -7.9171 4.374e-15 ***
## born_gdrxEast Germany (not GDR) -0.1985209  0.1331748 -1.4907  0.136233
## born_gdrxGDR                    0.0079796  0.0784307  0.1017  0.918974
## electoralTermx13                0.0465298  0.0820205  0.5673  0.570590
## electoralTermx14               -0.1753196  0.0916608 -1.9127  0.055956 .
## electoralTermx15               -0.0244156  0.1348620 -0.1810  0.856357
```

```
## electoralTermx16                -0.6584419  0.1091512 -6.0324 1.982e-09 ***
## electoralTermx17                -0.2253302  0.1334031 -1.6891  0.091387 .
## electoralTermx18                -1.0564495  0.1502134 -7.0330 2.929e-12 ***
## electoralTermx19                -0.6848531  0.1381091 -4.9588 7.809e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
model_pgostdeutschland_robust_clustered
```

```
##
## t test of coefficients:
##
##                                  Estimate Std. Error t value  Pr(>|t|)
## (Intercept)                      0.077139   0.058056  1.3287   0.18413
## partygroupxAfD                   0.834745   0.374806  2.2271   0.02607 *
## partygroupxPDS/Die Linke         0.939943   0.109548  8.5802 < 2.2e-16 ***
## born_gdrxEast Germany (not GDR) -0.213184   0.139507 -1.5281   0.12667
## born_gdrxGDR                     0.011184   0.077370  0.1446   0.88508
## electoralTermx13                 0.100929   0.083353  1.2109   0.22612
## electoralTermx14                -0.111993   0.090559 -1.2367   0.21638
## electoralTermx15                 0.034195   0.133875  0.2554   0.79843
## electoralTermx16                -0.585773   0.110315 -5.3100 1.242e-07 ***
## electoralTermx17                -0.152088   0.135714 -1.1207   0.26260
## electoralTermx18                -1.012888   0.151326 -6.6934 2.959e-11 ***
## electoralTermx19                -0.612696   0.139038 -4.4067 1.116e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#regression using bundesland sentiment subset
bundesland_prejoin <- bundesland_df %>%
  unite(col = mp_session, electoralTerm, politicianId, sep = "_", remove = FALSE) %>%
  select(politicianId, electoralTerm, speaker_session_sent_bundesland, mp_session, party, born_gdr) %>%
  distinct(mp_session, .keep_all = TRUE)

bundesland_normed <- left_join(bundesland_prejoin, full_prejoin, by = "mp_session")

bundesland_normed[c("speaker_session_sent_bundesland","speaker_session_sent")] <- lapply(bundesland_norm

bundesland_normed <- bundesland_normed %>%
  mutate(norm_sent = (speaker_session_sent_bundesland - speaker_session_sent)) %>%
  mutate(partygroup = case_when(party == "CDU/CSU" | party == "FDP" | party == "SPD" | party == "Bündnis
                                party == "AfD" ~ "AfD",
                                party == "PDS/Die Linke" ~ "PDS/Die Linke"))

partygroupx = fct_relevel(as.factor(bundesland_normed$partygroup), "Mainstream", after = 0)

partyx = fct_relevel(as.factor(bundesland_normed$party), "PDS/Die Linke", after = 0)

born_systemx = fct_relevel(as.factor(bundesland_normed$born_system), "Reunified Germany", after = 0)
```

```
## Warning: Unknown or uninitialised column: `born_system`.
```

```
## Warning: Unknown levels in `f`: Reunified Germany
```

```
born_gdrx = fct_relevel(as.factor(bundesland_normed$born_gdr), "Elsewhere", after = 0)
```

```r
electoralTermx = as.factor(bundesland_normed$electoralTerm.y)

#(heteroskedasticity-robust) linear regression model with party specification

model_pbundesland <- lm(norm_sent ~ partyx + born_gdrx + electoralTermx, data = bundesland_normed)

model_pbundesland_sum <- summary(model_pbundesland)

model_pbundesland_robust_clustered <- coeftest(model_pbundesland,
                                    vcov = vcovCL,
                                    type = "HC1",
                                    cluster = ~politicianId.x)

#(heteroskedasticity-robust) linear regression model with partygroup specification

model_pgbundesland <- lm(norm_sent ~ partygroupx + born_gdrx + electoralTermx, data = bundesland_normed

model_pgbundesland_sum <- summary(model_pgbundesland)

model_pgbundesland_robust_clustered <- coeftest(model_pgbundesland,
                                    vcov = vcovCL,
                                    type = "HC1",
                                    cluster = ~politicianId.x)

#print regression outputs

model_pbundesland_robust_clustered
```

```
##
## t test of coefficients:
##
##                                  Estimate Std. Error  t value  Pr(>|t|)
## (Intercept)                      1.500187   0.115598  12.9776 < 2.2e-16 ***
## partyxAfD                        0.138764   0.184383   0.7526   0.45178
## partyxBündnis 90/Die Grünen     -0.559959   0.121083  -4.6246 3.960e-06 ***
## partyxCDU/CSU                   -1.156152   0.101917 -11.3441 < 2.2e-16 ***
## partyxFDP                       -1.042531   0.119586  -8.7178 < 2.2e-16 ***
## partyxSPD                       -1.160329   0.104242 -11.1312 < 2.2e-16 ***
## born_gdrxEast Germany (not GDR) -0.239164   0.135373  -1.7667   0.07741 .
## born_gdrxGDR                    -0.171755   0.068452  -2.5091   0.01217 *
## electoralTermx13                -0.110734   0.083605  -1.3245   0.18547
## electoralTermx14                -0.257195   0.103914  -2.4751   0.01339 *
## electoralTermx15                -0.191121   0.126794  -1.5073   0.13186
## electoralTermx16                -0.878702   0.103633  -8.4790 < 2.2e-16 ***
## electoralTermx17                -0.593702   0.098030  -6.0563 1.621e-09 ***
## electoralTermx18                -1.168669   0.107090 -10.9130 < 2.2e-16 ***
## electoralTermx19                -0.959771   0.096336  -9.9628 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
model_pgbundesland_robust_clustered
```

```
##
## t test of coefficients:
```

```
##
##                                  Estimate Std. Error  t value  Pr(>|t|)
## (Intercept)                      0.381825   0.069763   5.4732 4.897e-08 ***
## partygroupxAfD                   1.204118   0.164955   7.2997 3.945e-13 ***
## partygroupxPDS/Die Linke         1.069387   0.094616  11.3024 < 2.2e-16 ***
## born_gdrxEast Germany (not GDR) -0.253965   0.136667  -1.8583   0.06326 .
## born_gdrxGDR                    -0.186194   0.069737  -2.6699   0.00764 **
## electoralTermx13                -0.062374   0.083643  -0.7457   0.45591
## electoralTermx14                -0.222699   0.103168  -2.1586   0.03098 *
## electoralTermx15                -0.128907   0.125122  -1.0302   0.30300
## electoralTermx16                -0.807298   0.104758  -7.7063 1.906e-14 ***
## electoralTermx17                -0.520705   0.098013  -5.3126 1.184e-07 ***
## electoralTermx18                -1.105253   0.108711 -10.1669 < 2.2e-16 ***
## electoralTermx19                -0.903275   0.097095  -9.3030 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Descriptive Statistics

In this final step, I calculate descriptive statistics both overall and per keyword group. These include, but are not limited to, the amount of speeches for each keyword group, the amount of speeches by party or birthplace, as well as the different distributions of relative sentiment scores by party and birthplace.

```
#number of speeches by party

#ddr cluster

length(unique(ddr_df$id))
```

```
## [1] 7334
```

```
afdddr <- ddr_df[ddr_df$party == "AfD",]
length(unique(afdddr$id))
```

```
## [1] 210
```

```
length(unique(afdddr$politicianId))
```

```
## [1] 62
```

```
cduddr <- ddr_df[ddr_df$party == "CDU/CSU",]
length(unique(cduddr$id))
```

```
## [1] 2402
```

```
length(unique(cduddr$politicianId))
```

```
## [1] 488
```

```
fdpddr <- ddr_df[ddr_df$party == "FDP",]
length(unique(fdpddr$id))
```

```
## [1] 884
```

```
length(unique(fdpddr$politicianId))
```

```
## [1] 154
```

```
spdddr <- ddr_df[ddr_df$party == "SPD",]
length(unique(spdddr$id))
```

```
## [1] 1570
```

```
length(unique(spdddr$politicianId))
```

```
## [1] 346
```

```
grnddr <- ddr_df[ddr_df$party == "Bündnis 90/Die Grünen",]
length(unique(grnddr$id))
```

```
## [1] 810
```

```
length(unique(grnddr$politicianId))
```

```
## [1] 96
```

```
pdsddr <- ddr_df[ddr_df$party == "PDS/Die Linke",]
length(unique(pdsddr$id))
```

```
## [1] 1458
```

```
length(unique(pdsddr$politicianId))
```

```
## [1] 117
```

```
#ostdeutschland cluster

length(unique(ostdeutschland_df$id))
```

```
## [1] 5774
```

```
afdodl <- ostdeutschland_df[ostdeutschland_df$party == "AfD",]
length(unique(afdodl$id))
```

```
## [1] 40
```

```
length(unique(afdodl$politicianId))
```

```
## [1] 19
```

```
cduodl <- ostdeutschland_df[ostdeutschland_df$party == "CDU/CSU",]
length(unique(cduodl$id))
```

```
## [1] 1541
```

```
length(unique(cduodl$politicianId))
```

```
## [1] 347
```

```
fdpodl <- ostdeutschland_df[ostdeutschland_df$party == "FDP",]
length(unique(fdpodl$id))
```

```
## [1] 606
```

```
length(unique(fdpodl$politicianId))
```

```
## [1] 121
```

```
spdodl <- ostdeutschland_df[ostdeutschland_df$party == "SPD",]
length(unique(spdodl$id))
```

```
## [1] 2007
```

```
length(unique(spdodl$politicianId))
```

```
## [1] 346
```

```
grnodl <- ostdeutschland_df[ostdeutschland_df$party == "Bündnis 90/Die Grünen",]
length(unique(grnodl$id))
```

```
## [1] 646
```

```
length(unique(grnodl$politicianId))
```

```
## [1] 86
```

```
pdsodl <- ostdeutschland_df[ostdeutschland_df$party == "PDS/Die Linke",]
length(unique(pdsodl$id))
```

```
## [1] 934
```

```
length(unique(pdsodl$politicianId))
```

```
## [1] 110
```

```
#bundesland cluster

length(unique(bundesland_df$id))
```

```
## [1] 6814
```

```
afdbundesland <- bundesland_df[bundesland_df$party == "AfD",]
length(unique(afdbundesland$id))
```

```
## [1] 179
```

```
length(unique(afdbundesland$politicianId))
```

```
## [1] 54
```

```
cdubundesland <- bundesland_df[bundesland_df$party == "CDU/CSU",]
length(unique(cdubundesland$id))
```

```
## [1] 2202
```

```
length(unique(cdubundesland$politicianId))
```

```
## [1] 483
```

```
fdpbundesland <- bundesland_df[bundesland_df$party == "FDP",]
length(unique(fdpbundesland$id))
```

```
## [1] 728
```

```
length(unique(fdpbundesland$politicianId))
```

```
## [1] 164
```

```
spdbundesland <- bundesland_df[bundesland_df$party == "SPD",]
length(unique(spdbundesland$id))
```

```
## [1] 1596
```

```
length(unique(spdbundesland$politicianId))
```

```
## [1] 403
```

```
grnbundesland <- bundesland_df[bundesland_df$party == "Bündnis 90/Die Grünen",]
length(unique(grnbundesland$id))
```

```
## [1] 777
```

```
length(unique(grnbundesland$politicianId))
```

```
## [1] 126
```

```
pdsbundesland <- bundesland_df[bundesland_df$party == "PDS/Die Linke",]
length(unique(pdsbundesland$id))
```

```
## [1] 1332
```

```
length(unique(pdsbundesland$politicianId))
```

```
## [1] 140
```

```
#number of speeches by birthplace

#ddr cluster

length(unique(ddr_df$id))
```

```
## [1] 7334
```

```
gdrddr <- ddr_df[ddr_df$born_gdr == "GDR",]
length(unique(gdrddr$id))
```

```
## [1] 1969
```

```
length(unique(gdrddr$politicianId))
```

```
## [1] 196
```

```
egddr <- ddr_df[ddr_df$born_gdr == "East Germany (not GDR)",]
length(unique(egddr$id))
```

```
## [1] 647
```

```
length(unique(egddr$politicianId))
```

```
## [1] 85
```

```
westddr <- ddr_df[ddr_df$born_gdr == "Elsewhere",]
length(unique(westddr$id))
```

```
## [1] 4718
```

```
length(unique(westddr$politicianId))
```

```
## [1] 974
```

```
#ostdeutschland cluster

length(unique(ostdeutschland_df$id))
```

```
## [1] 5774
```

```
gdrodl <- ostdeutschland_df[ostdeutschland_df$born_gdr == "GDR",]
length(unique(gdrodl$id))
```

```
## [1] 1846
```

```
length(unique(gdrodl$politicianId))
```

```
## [1] 188
```

```
egodl <- ostdeutschland_df[ostdeutschland_df$born_gdr == "East Germany (not GDR)",]
length(unique(egodl$id))
```

```
## [1] 408
```

```
length(unique(egodl$politicianId))
```

```
## [1] 68
```

```
westodl <- ostdeutschland_df[ostdeutschland_df$born_gdr == "Elsewhere",]
length(unique(westodl$id))
```

```
## [1] 3520
```

```
length(unique(westodl$politicianId))
```

```
## [1] 765
```

```
#bundesland cluster

length(unique(bundesland_df$id))
```

```
## [1] 6814
```

```
gdrbundesland <- bundesland_df[bundesland_df$born_gdr == "GDR",]
length(unique(gdrbundesland$id))
```

```
## [1] 2332
```

```
length(unique(gdrbundesland$politicianId))
```

```
## [1] 220
```

```
egbundesland <- bundesland_df[bundesland_df$born_gdr == "East Germany (not GDR)",]
length(unique(egbundesland$id))
```

```
## [1] 357
```

```
length(unique(egbundesland$politicianId))
```

```
## [1] 75
```

```
westbundesland <- bundesland_df[bundesland_df$born_gdr == "Elsewhere",]
length(unique(westbundesland$id))
```

```
## [1] 4125
```

```
length(unique(westbundesland$politicianId))
```

```
## [1] 1067
```

```
#distribution of relative sentiment by party and birthplace

#ddr cluster

length(ddr_normed$norm_sent)
```

```
## [1] 2031
```

```
summary(ddr_normed$norm_sent)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -7.41901 -0.73843  0.07773  0.00000  0.79307  8.07368
```

```
sd(ddr_normed$norm_sent)
```

```
## [1] 1.347271
```

```
describeBy(ddr_normed$norm_sent, ddr_normed$party)
```

```
##
##  Descriptive statistics by group
## group: AfD
##    vars  n mean   sd median trimmed  mad   min  max range skew kurtosis   se
## X1    1 62 0.74 1.03   0.66    0.72 0.98 -1.16 3.51  4.67 0.26    -0.53 0.13
## ------------------------------------------------------------
## group: Bündnis 90/Die Grünen
##    vars   n mean   sd median trimmed  mad   min  max range  skew kurtosis   se
## X1    1 161 0.46 1.41   0.55    0.53 1.08 -5.02 7.57  12.6 -0.05     5.78 0.11
## ------------------------------------------------------------
## group: CDU/CSU
##    vars   n  mean   sd median trimmed  mad   min  max range  skew kurtosis   se
## X1    1 831 -0.29 1.35  -0.27   -0.27 1.15 -7.42 8.07 15.49 -0.14     3.71 0.05
## ------------------------------------------------------------
## group: FDP
##    vars   n  mean   sd median trimmed  mad   min  max range  skew kurtosis   se
## X1    1 237 -0.22 1.17  -0.09   -0.18 0.91 -4.24 3.95  8.19 -0.29     1.31 0.08
## ------------------------------------------------------------
## group: PDS/Die Linke
##    vars   n mean   sd median trimmed  mad   min  max range  skew kurtosis   se
## X1    1 205 0.69 1.14   0.75    0.74 0.85 -5.94 4.61 10.55 -0.97     5.56 0.08
## ------------------------------------------------------------
## group: SPD
##    vars   n mean   sd median trimmed  mad   min max range  skew kurtosis   se
## X1    1 535 0.06 1.34   0.13     0.1 1.12 -6.62 4.5 11.13 -0.56     2.13 0.06
```

```
describeBy(ddr_normed$norm_sent, ddr_normed$born_gdr)
```

```
##
##  Descriptive statistics by group
## group: East Germany (not GDR)
##    vars   n mean  sd median trimmed  mad   min  max range  skew kurtosis   se
## X1    1 124 0.01 1.4  -0.04    0.04 1.02 -5.94 5.11 11.05 -0.51     3.79 0.13
## ------------------------------------------------------------
## group: Elsewhere
##    vars    n mean   sd median trimmed  mad   min  max range  skew kurtosis   se
## X1    1 1506    0 1.38   0.09    0.03 1.17 -7.42 8.07 15.49 -0.24     2.95 0.04
## ------------------------------------------------------------
## group: GDR
##    vars   n  mean   sd median trimmed  mad   min  max range  skew kurtosis   se
## X1    1 401 -0.02 1.21   0.07    0.05 0.98 -4.63 4.61  9.24 -0.62     1.72 0.06
```

```
#east germany cluster

length(ostdeutschland_normed$norm_sent)
```

```
## [1] 1698
```

```
summary(ostdeutschland_normed$norm_sent)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -9.87586 -0.69422  0.06868  0.00000  0.77824  6.03166
```

```
sd(ostdeutschland_normed$norm_sent)
```

```
## [1] 1.31799
```

```
describeBy(ostdeutschland_normed$norm_sent, ostdeutschland_normed$party)
```

```
##
##  Descriptive statistics by group
## group: AfD
##    vars  n mean   sd median trimmed  mad   min  max range skew kurtosis   se
## X1    1 19 0.29 1.58   0.32    0.26 0.89 -3.61 4.72  8.33 0.42     2.37 0.36
## ------------------------------------------------------------
## group: Bündnis 90/Die Grünen
##    vars   n mean   sd median trimmed mad   min  max range skew kurtosis  se
## X1    1 147 0.43 1.18    0.4    0.44 0.9 -3.28 4.95  8.24 0.12     2.09 0.1
## ------------------------------------------------------------
## group: CDU/CSU
##    vars   n  mean   sd median trimmed mad   min  max range  skew kurtosis   se
## X1    1 585 -0.22 1.36   -0.2    -0.2 1.2 -9.88 6.03 15.91 -0.76     5.84 0.06
## ------------------------------------------------------------
## group: FDP
##    vars   n  mean   sd median trimmed  mad   min  max range  skew kurtosis   se
## X1    1 185 -0.13 1.15  -0.05   -0.08 0.81 -6.12 2.83  8.95 -1.02     4.41 0.08
## ------------------------------------------------------------
## group: PDS/Die Linke
##    vars   n mean  sd median trimmed  mad   min  max range skew kurtosis   se
## X1    1 174 0.71 1.1   0.72     0.7 0.84 -3.39 4.26  7.64 0.02     1.21 0.08
## ------------------------------------------------------------
## group: SPD
##    vars   n  mean   sd median trimmed  mad   min  max range  skew kurtosis   se
## X1    1 588 -0.06 1.31   0.04   -0.01 1.13 -5.97 4.22 10.19 -0.39     1.55 0.05
```

```
describeBy(ostdeutschland_normed$norm_sent, ostdeutschland_normed$born_gdr)
```

```
##
##  Descriptive statistics by group
## group: East Germany (not GDR)
##    vars   n  mean   sd median trimmed  mad   min  max range  skew kurtosis   se
## X1    1 104 -0.05 1.24  -0.06   -0.05 1.09 -3.77 4.07  7.84 -0.04     1.16 0.12
## ------------------------------------------------------------
## group: Elsewhere
##    vars    n mean   sd median trimmed  mad   min  max range  skew kurtosis   se
## X1    1 1221    0 1.37   0.09    0.05 1.16 -9.88 6.03 15.91 -0.55      3.4 0.04
## ------------------------------------------------------------
## group: GDR
##    vars   n mean   sd median trimmed  mad   min  max range  skew kurtosis   se
## X1    1 373 0.02 1.18   0.03    0.03 0.91 -7.58 4.26 11.83 -0.54     5.09 0.06
```

```
#east german länder cluster
```

```
length(bundesland_normed$norm_sent)
```

```
## [1] 2330
```

```
summary(bundesland_normed$norm_sent)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -8.05772 -0.73489  0.06923  0.00000  0.79126  9.97278
```

```
sd(bundesland_normed$norm_sent)
```

```
## [1] 1.351789
```

```
describeBy(bundesland_normed$norm_sent, bundesland_normed$party)
```

```
##
##  Descriptive statistics by group
## group: AfD
##    vars  n mean   sd median trimmed  mad   min max range skew kurtosis   se
## X1    1 54 0.64 1.11   0.54     0.6 0.99 -1.49 4.2  5.69  0.5     0.68 0.15
## ------------------------------------------------------------
## group: Bündnis 90/Die Grünen
##    vars   n mean   sd median trimmed  mad   min  max range  skew kurtosis   se
## X1    1 242  0.3 1.26   0.42    0.41 0.99 -8.06 3.01 11.07 -2.21    11.47 0.08
## ------------------------------------------------------------
## group: CDU/CSU
##    vars   n  mean   sd median trimmed  mad   min max range skew kurtosis   se
## X1    1 853 -0.21 1.34  -0.18   -0.19 1.19 -7.63 6.1 13.73 -0.3     2.87 0.05
## ------------------------------------------------------------
## group: FDP
##    vars   n  mean   sd median trimmed mad   min  max range skew kurtosis   se
## X1    1 247 -0.04 1.25  -0.04   -0.04 0.9 -5.99 6.75 12.74 0.01     6.46 0.08
## ------------------------------------------------------------
## group: PDS/Die Linke
##    vars   n mean   sd median trimmed  mad  min  max range  skew kurtosis   se
## X1    1 255 0.75 1.11   0.76    0.76 0.84 -3.6 5.52  9.12 -0.03     2.48 0.07
## ------------------------------------------------------------
## group: SPD
##    vars   n  mean  sd median trimmed  mad   min  max range  skew kurtosis   se
## X1    1 679 -0.16 1.4  -0.19   -0.13 1.19 -7.88 9.97 17.85 -0.01     5.78 0.05
```

```
describeBy(bundesland_normed$norm_sent, bundesland_normed$born_gdr)
```

```
##
##  Descriptive statistics by group
## group: East Germany (not GDR)
##    vars   n mean   sd median trimmed  mad   min  max range  skew kurtosis   se
## X1    1 112 0.11 1.16   0.27    0.17 0.98 -3.57 2.87  6.45 -0.53     0.41 0.11
## ------------------------------------------------------------
## group: Elsewhere
##    vars    n mean   sd median trimmed  mad   min  max range  skew kurtosis   se
## X1    1 1757    0 1.42   0.09    0.03 1.18 -8.06 9.97 18.03 -0.31     4.46 0.03
## ------------------------------------------------------------
## group: GDR
##    vars   n  mean   sd median trimmed  mad   min  max range  skew kurtosis   se
## X1    1 461 -0.04 1.09   0.01       0 1.01 -5.99 2.87  8.86 -0.69     2.04 0.05
```