R code commentary and appendix

Contents

1 Data cleaning

1.1 Import the dataset and check variables

```r
# import the data
tw <-  read_csv("data/large_files/politicians_final_corrected.csv",
                show_col_types = FALSE )
#save(tw,file="data/tw.Rda")
kable(colnames(tw), col.names = "variables")
```

variables
tw_screen_name
nome
tweet_testo
creato_il
creato_il_code
url
party_id
genere
chamber
status

1.2 Adjust date.time format

```r
# RUN IN THIS ORDER !!
Sys.setlocale("LC_TIME", "C")
tw$date <- as.Date(strptime(tw$creato_il,"%a %b %d %H:%M:%S %z %Y",
                            tz = "CET"))
tw$date <- na.replace(tw$date, as.Date(tw$creato_il))
```

1.2.1 Check the conversion

```
check_dates <- tw %>% select(creato_il,date)
kable(head(check_dates), col.names = c("Old date", "New date"))
```

Old date	New date
2021-02-13	2021-02-13
2021-02-09	2021-02-09
2021-02-07	2021-02-07
2021-01-21	2021-01-21
2021-01-21	2021-01-21
2021-01-20	2021-01-20

```
kable(tail(check_dates), col.names = c("Old date", "New date"))
```

Old date	New date
Mon Dec 28 09:51:35 +0000 2020	2020-12-28
Tue Jul 20 11:15:44 +0000 2021	2021-07-20
Thu Nov 26 13:46:51 +0000 2020	2020-11-26
Fri Oct 15 17:28:57 +0000 2021	2021-10-15
Wed Jun 03 12:22:31 +0000 2020	2020-06-03
Fri Dec 03 21:01:20 +0000 2021	2021-12-03

1.3 Create the week variable

```
tw <- tw %>% mutate(week = cut.Date(date, breaks = "1 week", labels = FALSE))
```

1.3.1 Check the variable

Inspect the first and the last dates and check if the number of weeks is correct

```r
max(tw$date)
```

```
## [1] "2022-04-18"
```

```r
min(tw$date)
```

```
## [1] "2020-01-01"
```

```r
difftime(max(tw$date), min(tw$date), units = "weeks")
```

```
## Time difference of 119.7143 weeks
```

1.4 Create the month variable

```r
tw <- tw %>% mutate(month = cut.Date(date, breaks = "1 month", labels = FALSE))
```

1.4.1 Check the number of month

```r
max(tw$month)
```

```
## [1] 28
```

```r
length(seq(from = min(tw$date), to = max(tw$date), by = 'month'))
```

```
## [1] 28
```

1.5 Create the trimester variable

```r
tw <- tw %>% mutate(quarter = cut.Date(date, breaks = "1 quarter", labels = FALSE))
```

1.5.1 Check the number of trimesters

```r
max(tw$quarter)
```

```
## [1] 10
```

```r
length(seq.Date(from = min(tw$date), to = max(tw$date), by = 'quarter'))
```

```
## [1] 10
```

1.6 Create the year variables

```r
tw <- tw %>% mutate(year = cut.Date(date, breaks = "year", labels = FALSE))
```

1.6.1 Check the number of years

```r
max(tw$year)
```

```
## [1] 3
```

```r
length(seq.Date(from = min(tw$date), to = max(tw$date), by = 'year'))
```

```
## [1] 3
```

1.7 Count the number of missing values

```
sum(is.na(tw))
```

```
## [1] 154672
```

1.7.1 Inspect where are the missings

```
missings <- c(
sum(is.na(tw$tw_screen_name)),
sum(is.na(tw$nome)),
sum(is.na(tw$tweet_testo)),
sum(is.na(tw$creato_il)),
sum(is.na(tw$creato_il_code)),
sum(is.na(tw$url)),
sum(is.na(tw$party_id)),
sum(is.na(tw$genere)),
sum(is.na(tw$chamber)),
sum(is.na(tw$status)),
sum(is.na(tw$date)),
sum(is.na(tw$week)),
sum(is.na(tw$month)),
sum(is.na(tw$quarter)),
sum(is.na(tw$year)))


missing_df <- data.frame(colnames(tw), missings)
kable(missing_df)
```

colnames.tw.	missings
tw_screen_name	0
nome	0
tweet_testo	6494
creato_il	0
creato_il_code	0
url	148178
party_id	0
genere	0
chamber	0
status	0
date	0
week	0
month	0
quarter	0
year	0

1.7.2 Remove rows with missing tweets

```
sum(is.na(tw$tweet_testo))
```

```
## [1] 6494
```

```
tw <- tw %>% drop_na(tweet_testo)
```

1.8 Check that the variables make sense

```r
unique(tw$party_id)
```

```
##  [1] "PD"            "FDI"           "M5S"           "FI"            "REG_LEAGUES"
##  [6] "MISTO"         "LEGA"          "IV"            "INDIPENDENTE"  "CI"
## [11] "LEU"
```

```r
unique(tw$genere)
```

```
## [1] "male"   "female" "male "
```

```r
unique(tw$chamber)
```

```
## [1] "NotParl" "Senate"  "Camera"
```

```r
unique(tw$status)
```

```
## [1] "sottosegretario" "presregione"     "viceministro"    "ministro"
## [5] "segretario"      "Parl"
```

1.8.1 Adjust the variable genere

```r
# Remove space from genere variable [RUN ONLY ONCE!]
a <- unique(tw$genere)
a[3]
```

```
## [1] "male "
```

```r
which(tw$genere == a[3])
```

```
## [1] 33300 33301 33302 33303 33304
```

```
tw$genere <- gsub(a[3],"male",tw$genere)
```

1.8.2 Verify the substitution

```
which(tw$genere == a[3])
```

```
## integer(0)
```

```
unique(tw$genere)
```

```
## [1] "male"   "female"
```

Now all the variables are ready for next steps

1.9 Create a new dataset selecting only necessary informations

```
# Select variables for the analysis
dataset <- tw %>% select(nome, tweet_testo, genere, party_id,
                         chamber,status, date, week, month, quarter, year )
colnames(dataset)
```

```
##  [1] "nome"        "tweet_testo" "genere"      "party_id"    "chamber"
##  [6] "status"      "date"        "week"        "month"       "quarter"
## [11] "year"
```

1.10 Create the corpus

```r
corpus <- corpus(dataset, text = "tweet_testo")
ndoc(corpus)
```

```
## [1] 391197
```

1.11 Create the DFM

```r
# Split the corpus into single tokens (remain positional)
doc.tokens <- tokens(corpus,
                                 remove_punct = TRUE,
                                 remove_numbers = TRUE,
                                 remove_symbols = TRUE,
                                 remove_url = TRUE)

# Import my stopwords
my_word <- as.list(read_csv("data/it_stopwords_new_list.csv",
                              show_col_types = FALSE))

# Attach unrecognized symbols
my_list <- c(" ","c'è","+"," ", my_word$stopwords,
             stopwords('italian'), stopwords("english"))

# Save my_list
#save(my_list,file="data/my_list.Rda")

doc.tokens <- tokens_select(doc.tokens, my_list, selection='remove')

DFM <- dfm(doc.tokens, tolower = TRUE)
```

1.12 Remove the emoji

```
# Create a copy of the dfm
test <- DFM
# Remove from the copy all the non ASCII carachters
test@Dimnames$features <- gsub("[^\x01-\x7F]", "", test@Dimnames$features)

# Check the difference from the list of features before and after apply gsub
a <- unique(test@Dimnames$features)
b <- unique(DFM@Dimnames$features)
setdiff(b,a) #I have selected also words that must not be removed

# Create an object with the features after remove non ASCII characters
c <- test@Dimnames$features
# Create an object with the original features
d <- DFM@Dimnames$features

# Create the list of the removed features
diff <- setdiff(d,c)
emoji <- diff[diff %>% nchar() < 4]
emoji <- list(emoji)
# Now i can remove this list from the dfm
DFM <- dfm_remove(DFM, emoji)

#save(DFM,file="data/dfm.Rda")
```

Now the data are ready for the next analysis

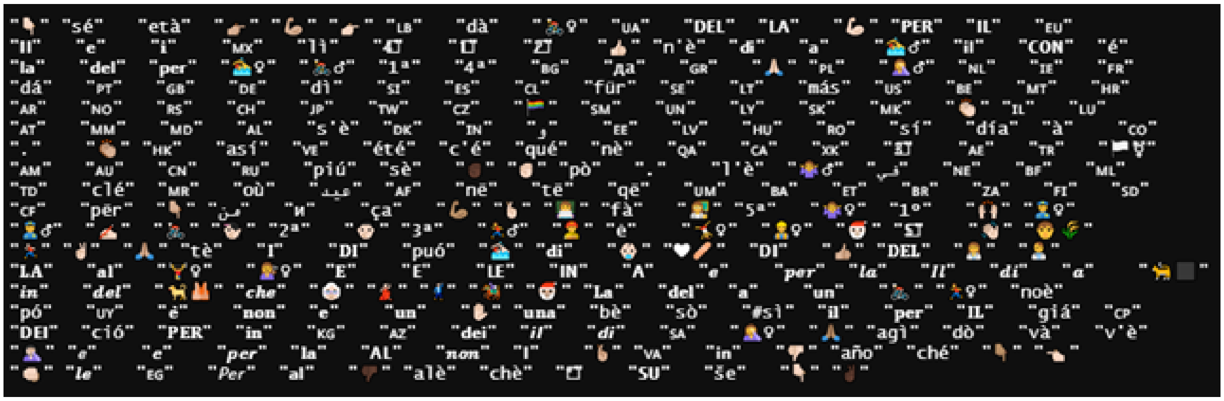


Figure 1: Emoji removed

2 Preliminar analysis

2.1 Who is inside this dataset?

```r
# Number of parliamentarians
n_parl <- length(unique(dataset$nome))
n_parl
```

```
## [1] 730
```

```r
# How many parliamentarians for each party_id?
n_parl_party <- dataset %>% select(party_id, nome) %>%
  group_by(party_id) %>% unique() %>% count() %>%
  arrange(desc(n))
kable(n_parl_party)
```

party_id	n
M5S	197
PD	144
LEGA	134
FI	96
MISTO	71
FDI	39
CI	17
LEU	15
REG_LEAGUES	7
INDIPENDENTE	6
IV	5

```
# Gender composition
n_gender <- dataset %>% select(genere, nome) %>%
  group_by(genere) %>% unique() %>% count()
kable(n_gender)
```

genere	n
female	258
male	472

```
# Wich is the period of analysis?
max(tw$date)
```

```
## [1] "2022-04-18"
```

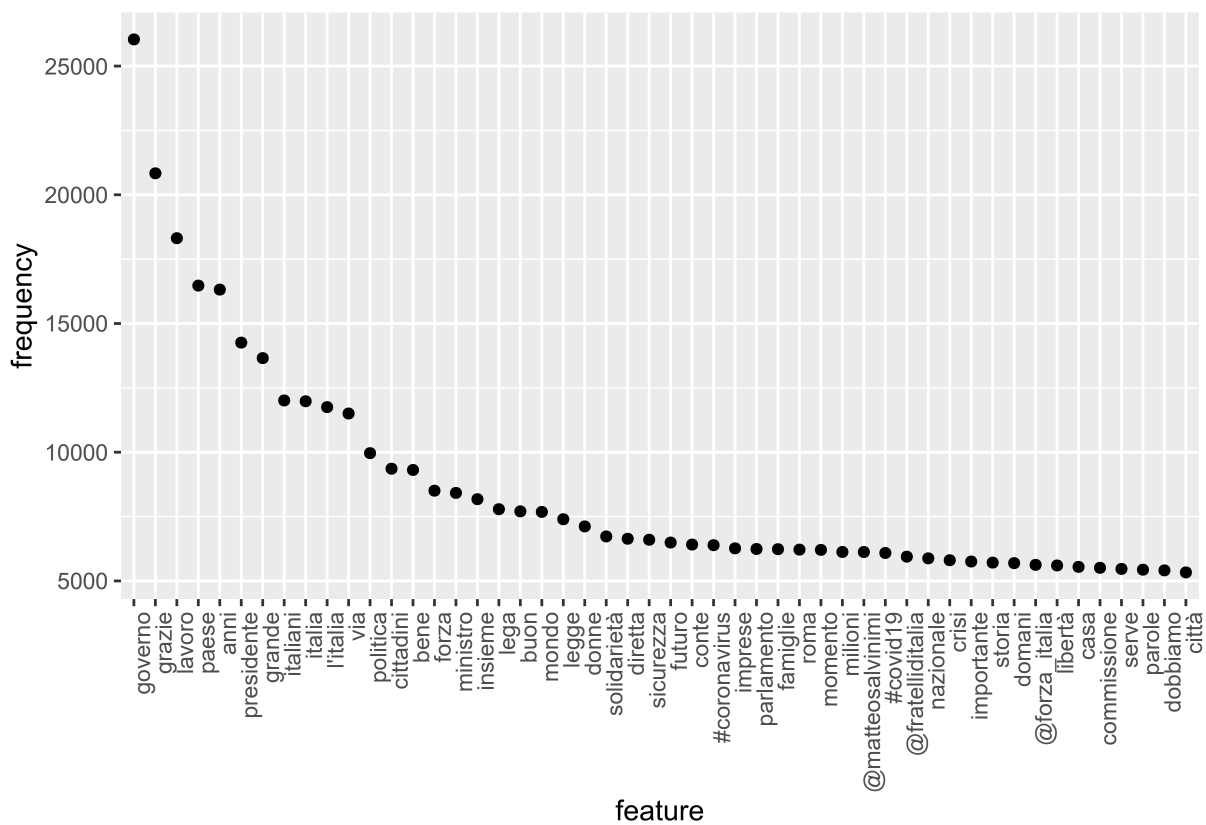
```
min(tw$date)
```

```
## [1] "2020-01-01"
```

2.2 Topfeatures frequency

```
# Textplotwordcloud
set.seed(123)
textplot_wordcloud(DFM, min_count = 20,max_words = 200,
     color = c('red', 'pink', 'green', 'purple', 'blue'))
```


```
# Plot frequency of the topfeatures in the DFM
features_dfm <- textstat_frequency(DFM, n = 50)
# Sort by reverse frequency order
features_dfm$feature <- with(features_dfm, reorder(feature, -frequency))
ggplot(features_dfm, aes(x = feature, y = frequency)) +
  geom_point() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```

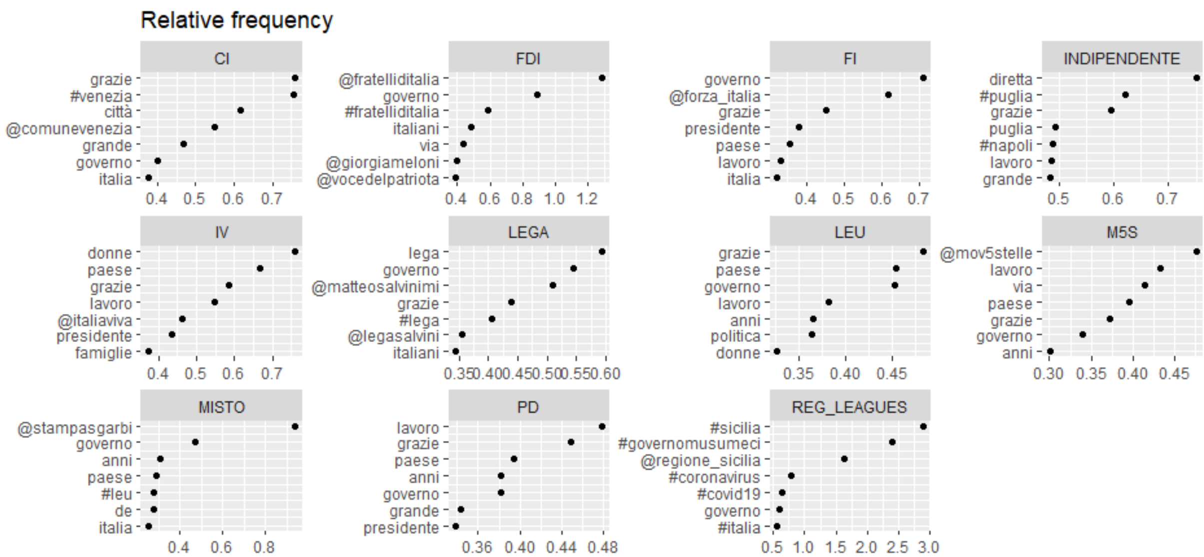


2.2.1 Relative frequency of the topfeatures by Party ID

```r
# group and weight the DFM
dfm_party_weight <- dfm_group(DFM, groups = party_id) %>%
  dfm_weight(scheme = "prop")
```

```r
# Plot relative frequency by party_id
freq_weight <- textstat_frequency(dfm_party_weight, n = 7, groups = party_id)



ggplot(data = freq_weight, aes(x = nrow(freq_weight):1, y = frequency)) +
     geom_point() +
     facet_wrap(~ group, scales = "free") +
```

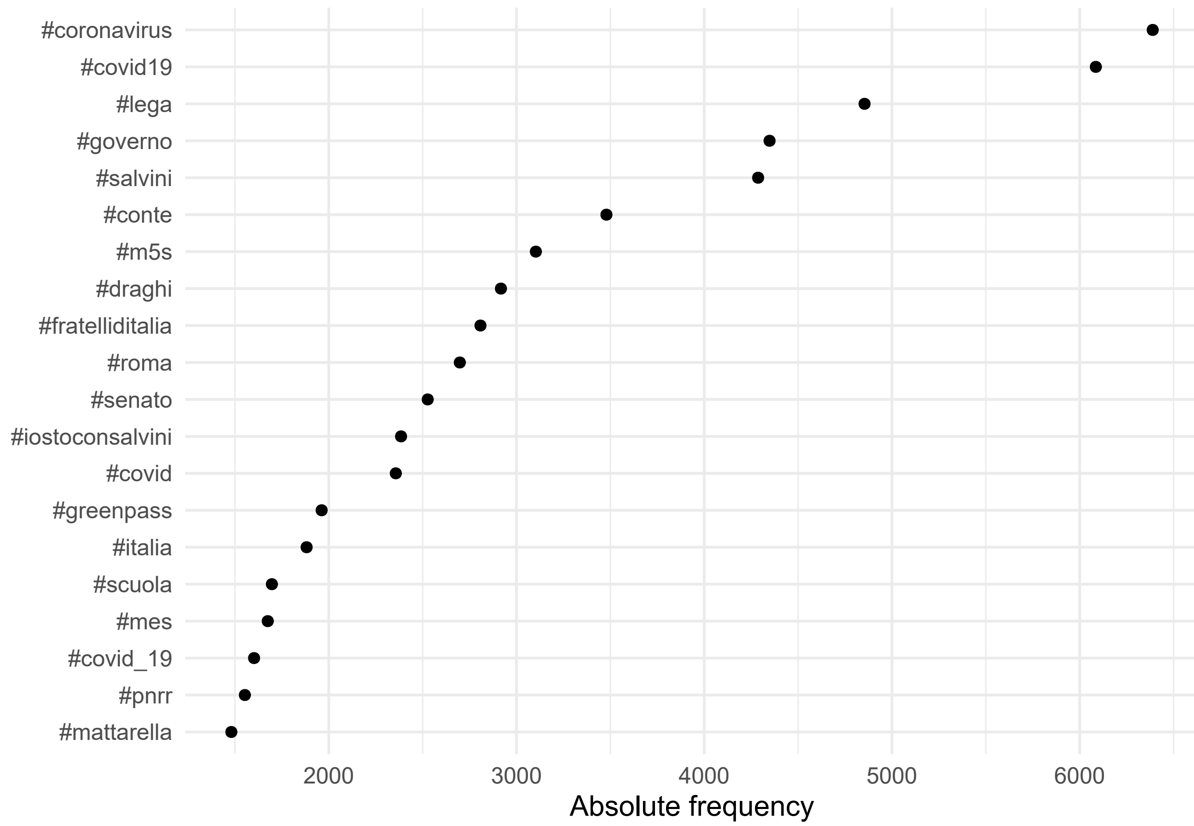
```
    coord_flip() +
    scale_x_continuous(breaks = nrow(freq_weight):1,
                       labels = freq_weight$feature) +
  ggtitle("Relative frequency") +
  labs(x = NULL, y = NULL)
```



2.3 Most common hashtag

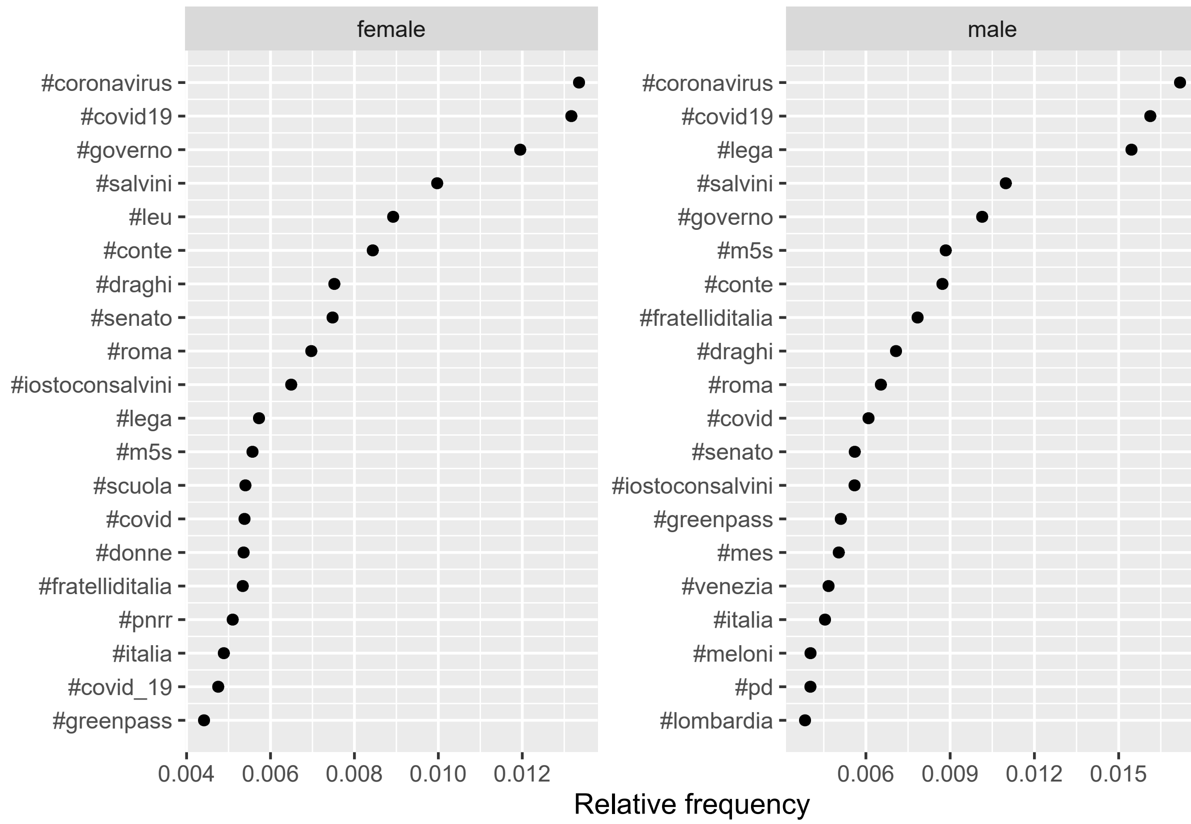
```
tag_dfm <- dfm_select(DFM, pattern = "#*")
toptag <- names(topfeatures(tag_dfm, 20))
toptag
```

```
##  [1] "#coronavirus"     "#covid19"        "#lega"           "#governo"
##  [5] "#salvini"         "#conte"          "#m5s"            "#draghi"
##  [9] "#fratelliditalia" "#roma"           "#senato"         "#iostoconsalvini"
## [13] "#covid"           "#greenpass"      "#italia"         "#scuola"
## [17] "#mes"             "#covid_19"       "#pnrr"           "#mattarella"
```



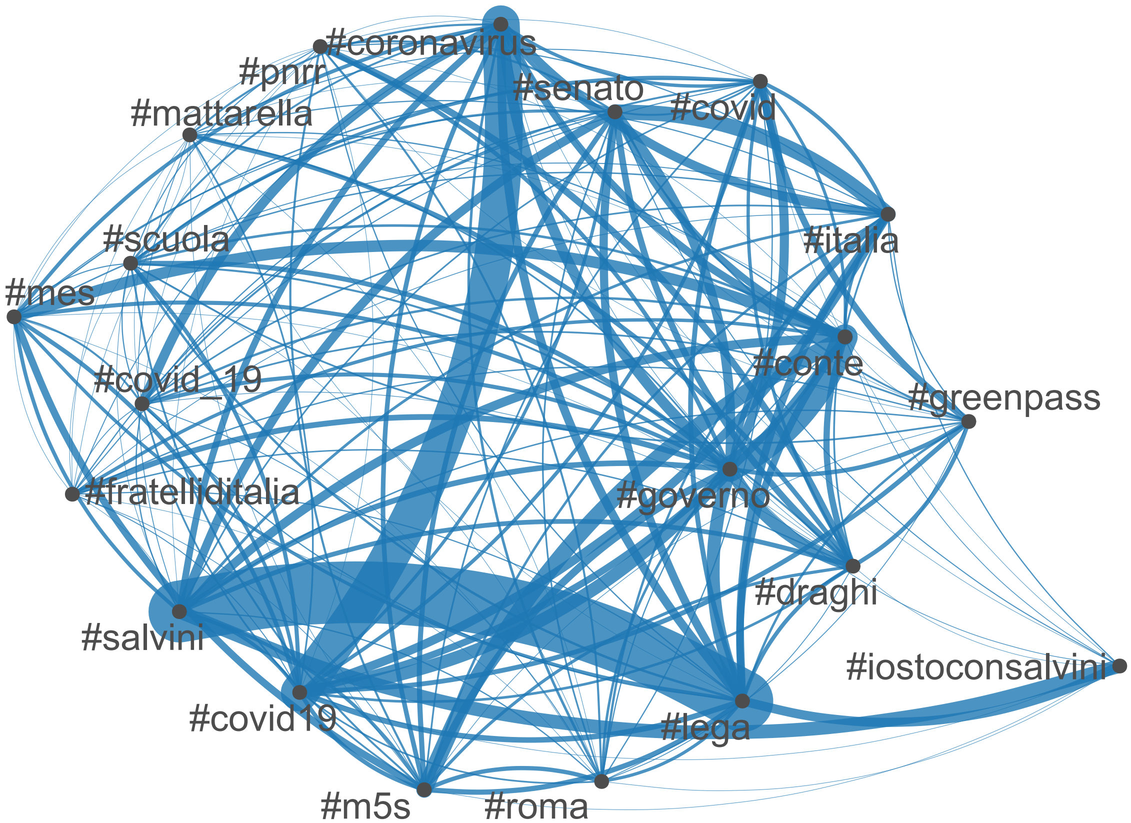
2.3.1 Most common hashtag by Gender

```
# group and weight the DFM
dfm_gender_weight <- dfm_group(tag_dfm, groups = genere) %>%
  dfm_weight(scheme = "prop")

tstat_freq <- textstat_frequency(dfm_gender_weight, n = 20,
                                 groups = dfm_gender_weight$genere)
```



2.3.2 Co-occurrence Plot of hashtags

```
tag_dfm_NOT_W <- dfm_select(DFM, pattern = "#*")
toptag_NOT <- names(topfeatures(tag_dfm_NOT_W, 20))

tag_fcm_NOT <- fcm(tag_dfm_NOT_W)
set.seed(666)
topgat_fcm_NOT <- fcm_select(tag_fcm_NOT, pattern = toptag_NOT)
textplot_network(topgat_fcm_NOT, min_freq = 0.1,
                 edge_alpha = 0.8, edge_size = 5)
```



2.4 Most frequently mentioned usernames

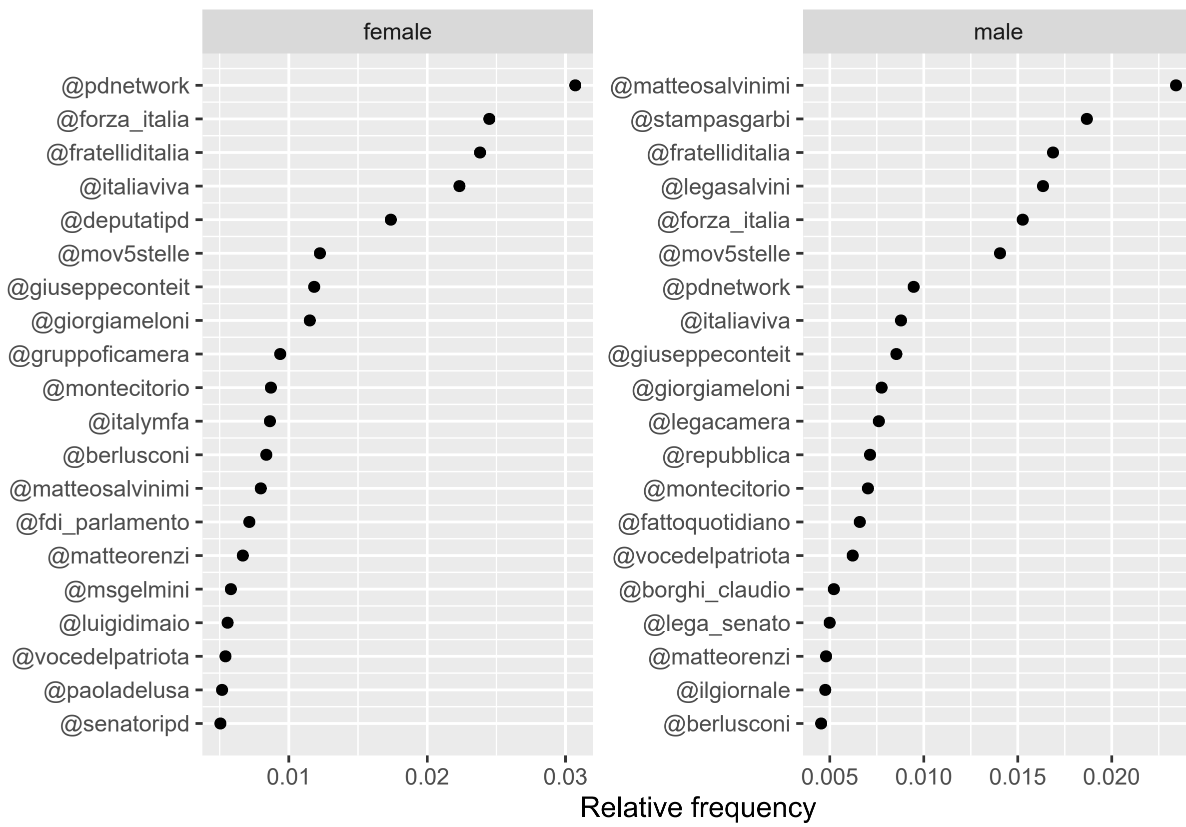
```
user_dfm <- dfm_select(DFM, pattern = "@*")
topuser <- names(topfeatures(user_dfm, 20, scheme = "docfreq"))
kable(topuser, col.names = "Most mentioned username")
```

Most mentioned username
@matteosalvinimi
@fratelliditalia
@forza_italia
@pdnetwork
@stampasgarbi
@mov5stelle
@legasalvini
@italiaviva
@giuseppeconteit
@giorgiameloni
@montecitorio
@deputatipd
@repubblica
@vocedelpatriota
@legacamera
@berlusconi
@matteorenzi
@fattoquotidiano
@enricoletta
@borghi_claudio

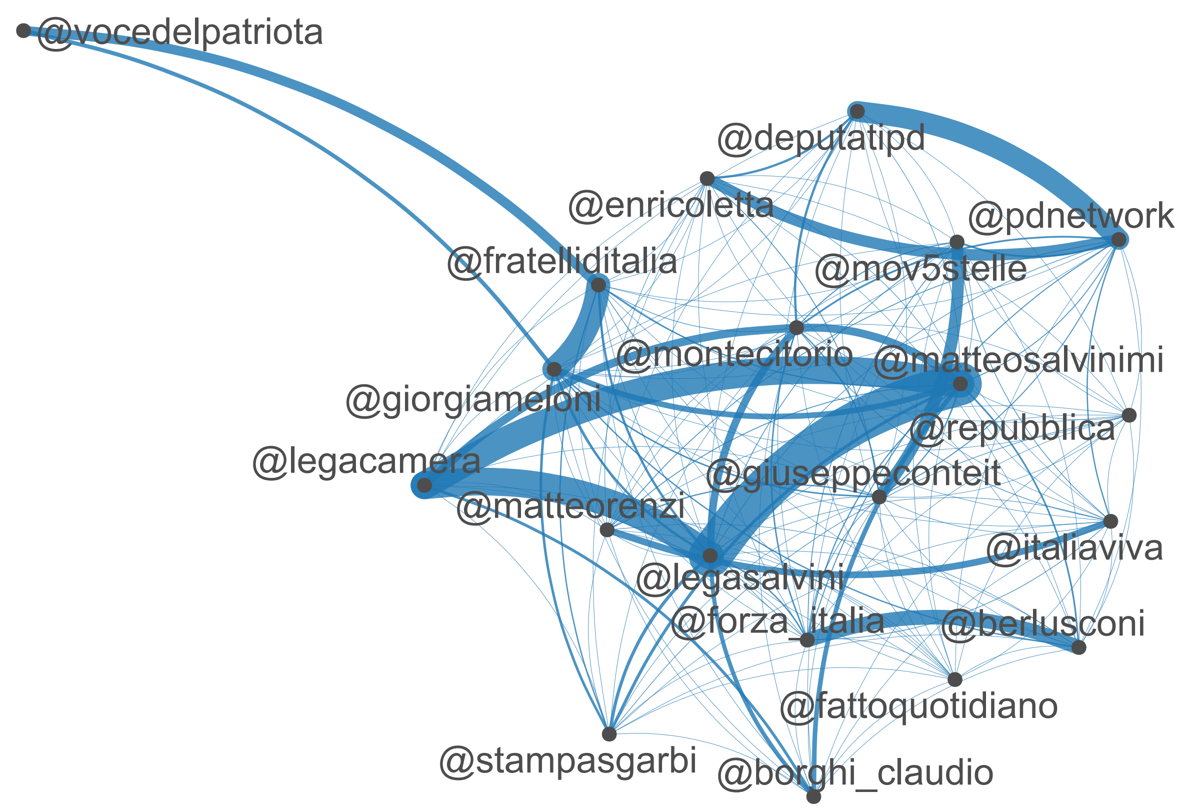
2.4.1 Most frequently mentioned usernames by gender

```
# group and weight the DFM
user_dfm_gender_weight <- dfm_group(user_dfm, groups = genere) %>%
  dfm_weight(scheme = "prop")


user_tstat_freq <- textstat_frequency(
  user_dfm_gender_weight,
```

```
  n = 20,
  groups = user_dfm_gender_weight$genere)
```



2.4.2 Co-occurrence plot of usernames



2.5 How many times a politician cite his/her party

```
party_citations <- data.frame(first = vector(), second = vector() )
system.time(
for (i in unique(tw$party_id))
{
  a <- paste("#", i ,sep = "")
  b <- tw %>% filter(grepl(a,tweet_testo)&party_id== i) %>% count()
  c <- tw %>% filter(party_id == i) %>% count()
  d <- (b/c) * 100
  party_citations <- rbind(party_citations, cbind(i,b,c,d))
```

```r
}
)

#save(party_citations, file = "data/party_citations.Rda")
```

```r
colnames(party_citations) <- c("party","n_citations", "Number of tweets", "perc")

kable(party_citations %>% arrange(desc(perc)),
      col.names = c("Party","Number of citations",
                    "number of tweets", "% of citations"))
```

Party	Number of citations	number of tweets	% of citations
M5S	1581	54418	2.9052887
LEGA	511	87162	0.5862647
FDI	131	36177	0.3621085
PD	179	91997	0.1945716
IV	5	3129	0.1597955
FI	62	65264	0.0949988
CI	1	6954	0.0143802
REG_LEAGUES	0	1398	0.0000000
MISTO	0	34644	0.0000000
INDIPENDENTE	0	2186	0.0000000
LEU	0	7868	0.0000000

In the above script i search the # for the parliamentary group, but is very unlikely, for example, that someone use the #IV for talking about the "Italia Viva" party, so i decided to enrich the dataframe creating a new variable with the name of the official twitter page for every party, and repeat the search using it.

I created the variable party_Page for only those parliamentary group that has a direct connection with a party (i excluded Reg_leagues, misto and indipendente)

2.5.1 Create the variable with the name of the official Twitter account

```
tw <- tw %>% mutate(party_page = if_else(party_id == "PD", "@pdnetwork",
if_else( party_id == "FDI", "@FratellidItalia",
    if_else(party_id == "M5S", "@Mov5Stelle",
      if_else(party_id == "FI", "@forza_italia",
        if_else(party_id == "LEGA", "@LegaSalvini",
            if_else(party_id == "CI", "@coraggio_italia",
              if_else(party_id == "LEU", "@liberi_uguali",
                                                      "NA"))))))))
```

2.5.2 Count for each party how many times a politician cite their respective party

```
party_citations_page <- data.frame(first = vector(), second = vector(),
                                   third = vector(), fourth = vector())
system.time(
  for (i in unique(tw$party_page))
  {
    b <- tw %>% filter(grepl(i,tweet_testo)&party_page== i) %>% count()
    c <- tw %>% filter(party_page == i) %>% count()
    d <- (b/c) * 100
    party_citations_page <- rbind(party_citations_page, cbind(i,b,c,d))


  }
)
# save(party_citations_page, file = "data/party_citations_page.Rda")
```

```
colnames(party_citations_page) <- c("party","n_citations",
                                    "Number of tweets", "perc")


kable(party_citations_page %>% filter(party != "NA") %>%
        arrange(desc(perc)),
      col.names = c("Party","Number of citations",
                    "number of tweets", "% of citations"))
```

Party	Number of citations	number of tweets	% of citations
@FratellidItalia	5842	36177	16.1483816
@forza_italia	5203	65264	7.9722358
@Mov5Stelle	3873	54418	7.1171304
@ItaliaViva	201	3129	6.4237776
@pdnetwork	4194	91997	4.5588443
@LegaSalvini	3364	87162	3.8594800
@coraggio_italia	131	6954	1.8838079
@liberi_uguali	16	7868	0.2033554

2.6 How many times the party leader is cited by his/her party

2.6.1 Create the variable with the official leader's account for every party

```
tw <- tw %>% mutate(party_leader =
if_else(party_id == "PD" & date < "2021-03-14", "@nzingaretti",
if_else(party_id == "PD" & date > "2021-03-14", "@EnricoLetta",
if_else( party_id == "FDI", "@GiorgiaMeloni",
if_else(party_id == "M5S" &date < "2020-01-22" , "@luigidimaio",
if_else(party_id == "M5S" &date > "2020-01-22" &date < "2021-08-06", "@vitocrimi",
if_else(party_id == "M5S" & date > "2021-08-061", "@GiuseppeConteIT",
if_else(party_id == "FI", "@berlusconi",
```

```r
if_else(party_id == "LEGA", "@matteosalvinimi",
if_else(party_id == "IV", "@matteorenzi",
if_else(party_id == "CI", "@LuigiBrugnaro",
if_else(party_id == "LEU", "@robersperanza",
"NA"))))))))))))
```

2.6.2 Count for each party how many times a politician cite his/ her party leader

```r
leader_citations <- data.frame(first = vector(), second = vector(),
                               third = vector(), fourth = vector())
system.time(
  for (i in unique(tw$party_leader))
  {
    b <- tw %>% filter(grepl(i,tweet_testo)&party_leader== i) %>% count()
    c <- tw %>% filter(party_leader == i) %>% count()
    d <- (b/c) * 100
    leader_citations <- rbind(leader_citations, cbind(i,b,c,d))

  }
)
 #save(leader_citations, file = "data/leader_citations.Rda")
```

```r
colnames(leader_citations) <- c("leader","n_citations",
                                "Number of tweets", "perc")


kable(leader_citations %>% filter(leader != "NA") %>%
        arrange(desc(perc)),
      col.names = c("Leader","Number of citations",
                    "Number of tweets", "% of citations"))
```

Leader	Number of citations	Number of tweets	% of citations
@matteosalvinimi	4826	87162	5.5368165
@GiorgiaMeloni	1745	36177	4.8235066
@GiuseppeConteIT	444	15517	2.8613778
@luigidimaio	30	1184	2.5337838
@berlusconi	1533	65264	2.3489213
@EnricoLetta	709	44520	1.5925427
@matteorenzi	46	3129	1.4701182
@nzingaretti	475	47305	1.0041222
@robersperanza	45	7868	0.5719370
@vitocrimi	107	37544	0.2849989
@LuigiBrugnaro	19	6954	0.2732240

2.7 How many times a politician cite itself in the tweet

```
self_citations <- data.frame(first = vector(), second = vector() )
system.time(
for (i in unique(tw$tw_screen_name))
{
  a <- paste("@", i ,sep = "")
  b <- tw %>% filter(grepl(a,tweet_testo) & tw_screen_name== i) %>% count()
  c <- tw %>% filter(tw_screen_name == i) %>% count()
  d <- (b/c) * 100
  self_citations <- rbind(self_citations, cbind(i,b,c,d))
}
)
#save(self_citations, file = "data/self_citations.Rda")
```

```r
colnames(self_citations) <- c("Politician","n_citations",
                              "Number of tweets", "perc")

kable(self_citations %>% filter(n_citations > 2) %>%
        arrange(desc(perc)),
      col.names = c("Politician","Number of citations",
                    "Number of tweets", "% of citations"))
```

Politician	Number of citations	Number of tweets	% of citations
wandaferro1	32	55	58.1818182
FrassinettiP	32	163	19.6319018
albertlaniece	51	282	18.0851064
Luca_Sut	20	341	5.8651026
DalilaNesci	17	341	4.9853372
PatassiniTullio	13	714	1.8207283
matteodallosso	3	170	1.7647059
sbonaccini	33	2884	1.1442441
sfnlcd	9	1308	0.6880734
gianluc_ferrara	3	560	0.5357143
adolfo_urso	7	1966	0.3560529
gualtierieurope	4	1432	0.2793296
MassimoUngaro	3	1135	0.2643172
EugenioGiani	3	1235	0.2429150
pierofassino	3	1255	0.2390438
ecdelre	4	2113	0.1893043
guglielmopicchi	3	3234	0.0927644

3 Dictionary analysis

3.1 Create the dictionary

I imported the excel file with the words for the dictionaries, excluding NA's.

```
# import dictionaries file
dict <-  read_excel("data/populism_dictionaries.xlsx")
variable.names(dict)
```

```
## [1] "Rooduijn_Pauwels_Italian"
## [2] "Grundl_Italian_adapted"
## [3] "Decadri_Boussalis"
## [4] "Decadri_Boussalis_Grundl_People"
## [5] "Decadri_Boussalis_Grundl_Common Will"
## [6] "Decadri_Boussalis_Grundl_Elite"
```

```
# create the dictionary
Rooduijn_Pauwels_Italian <-
  dictionary(list(populism =
                    (dict$Rooduijn_Pauwels_Italian
                     [!is.na(dict$Rooduijn_Pauwels_Italian)])))

Grundl_Italian_adapted <-
  dictionary(list(populism =
                    dict$Grundl_Italian_adapted
                  [!is.na(dict$Grundl_Italian_adapted)]))



Decadri_Boussalis_Grundl <-
  dictionary(list(people =
```

```
                      dict$Decadri_Boussalis_Grundl_People
                    [!is.na(dict$Decadri_Boussalis_Grundl_People)],
                    common_will =
                      dict$`Decadri_Boussalis_Grundl_Common Will`
                    [!is.na(dict$`Decadri_Boussalis_Grundl_Common Will`)],
                    elite =
                      dict$Decadri_Boussalis_Grundl_Elite
                    [!is.na(dict$Decadri_Boussalis_Grundl_Elite)]))
```

```
dictionaries <- c("Rooduijn_Pauwels_Italian", "Grundl_Italian_adapted"
                  ,"Decadri_Boussalis_Grundl")
n.words <- c(
  length(Rooduijn_Pauwels_Italian$populism),
  length(Grundl_Italian_adapted$populism),
  (length(Decadri_Boussalis_Grundl$people)+
     length(Decadri_Boussalis_Grundl$common_will)+
     length(Decadri_Boussalis_Grundl$elite))
            )

number_of_words <- data.frame(dictionaries,n.words)

kable(number_of_words)
```

dictionaries	n.words
Rooduijn_Pauwels_Italian	18
Grundl_Italian_adapted	135
Decadri_Boussalis_Grundl	77

3.1.1 Group and weight the dfm

```
# By party & quarter
dfm_weigh_p_quart <- dfm_group(DFM, groups = interaction(party_id, quarter))%>%
  dfm_weight(scheme = "prop")
```

Apply the dictionaries

3.2 Decadri_Boussalis_Grundl

```
# Dictionary analysis with Decadri_Boussalis_Grundl
# By quarter
dfm_dict1  <- dfm_lookup(dfm_weigh_p_quart, dictionary = Decadri_Boussalis_Grundl)
```

3.2.1 Transform the DFM into an ordinary dataframe

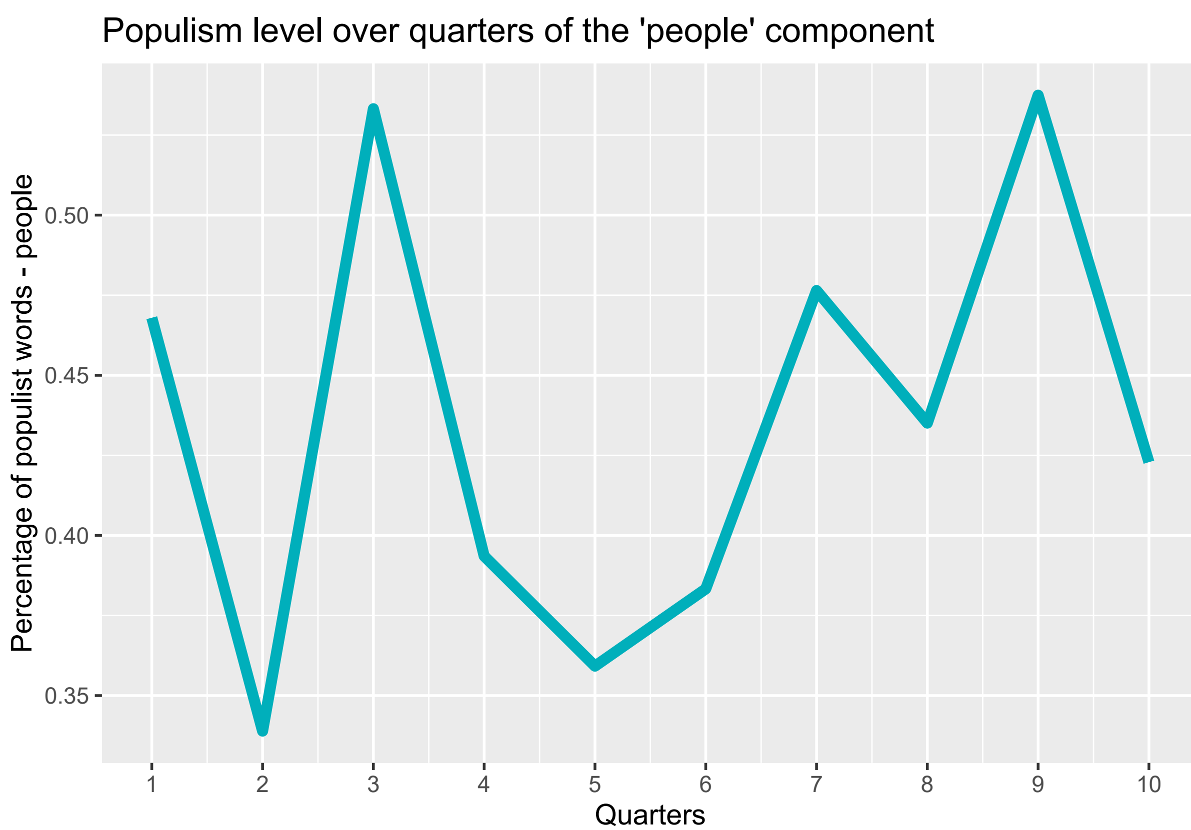
```
data_dict1 <- dfm_dict1 %>%
  quanteda::convert(to = "data.frame") %>%
  cbind(docvars(dfm_dict1))

# Add variable with general level of populism
data_dict1 <- data_dict1 %>%
  mutate(populism = (people + common_will + elite) * 100)
```

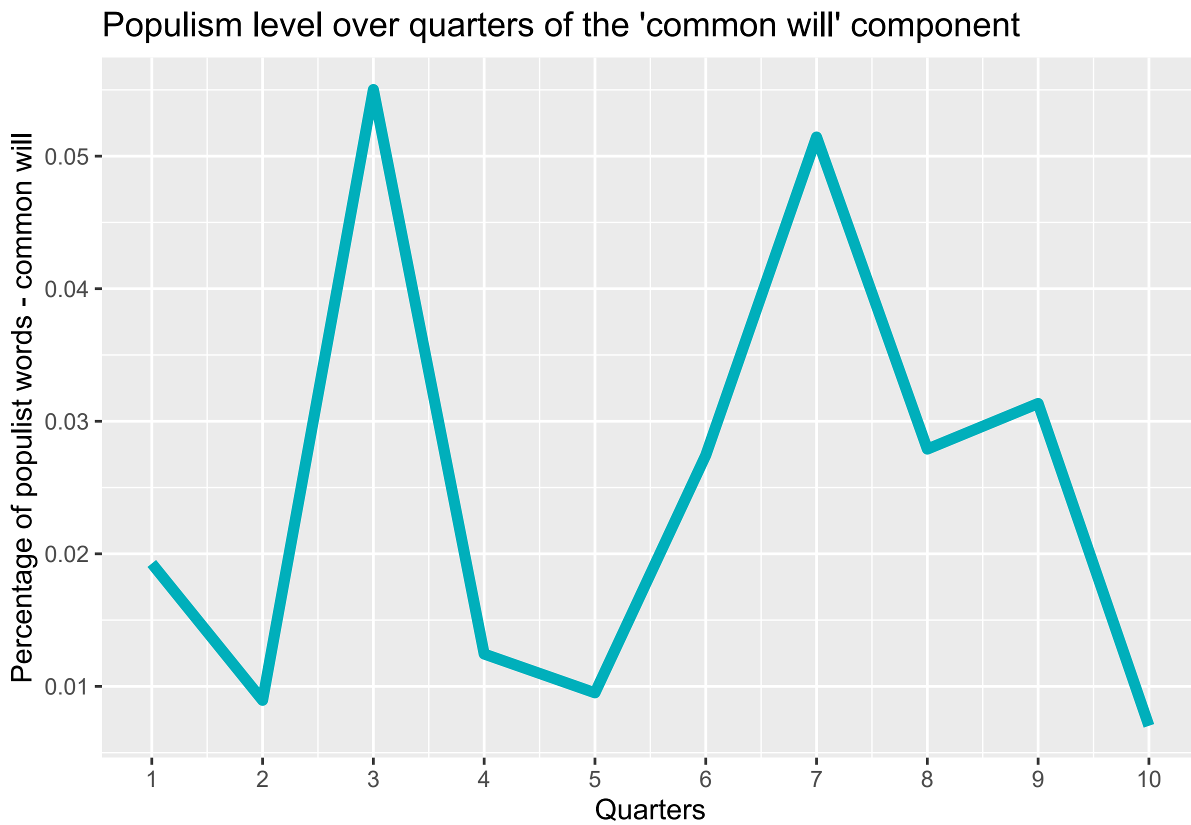
3.2.2 Level of populism in time

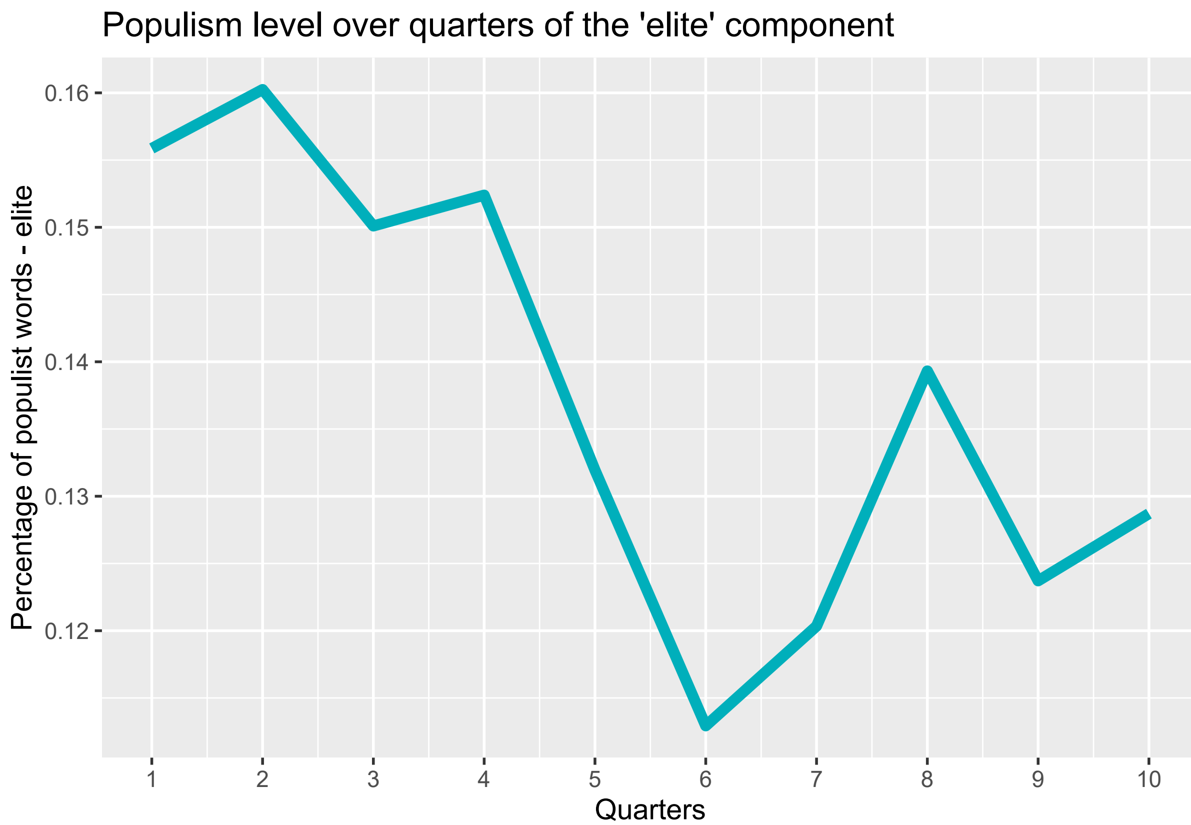
The code is only shown for the "PEOPLE" component but is identical for the others

```
#Over time PEOPLE (quarters)
data_quarter_people <- aggregate(x = data_dict1$people,  # Specify data column
          by = list(data_dict1$quarter),  # Specify group indicator
          FUN = mean) # Specify function (i.e. mean)
data_quarter_people$perc <- data_quarter_people$x * 100

# plot the level of the "people" component in time
```

```
plot_people <- ggplot(data = data_quarter_people, aes(x = Group.1, y = perc))+
  geom_line(color = "#00AFBB", size = 2)+
  scale_x_continuous("Quarters", labels = as.character(data_quarter_people$Group.1),
                     breaks = data_quarter_people$Group.1)+
  ylab("Percentage of populist words - people")+
  labs(title = "Populism level over quarters of the 'people' component")
plot_people
```

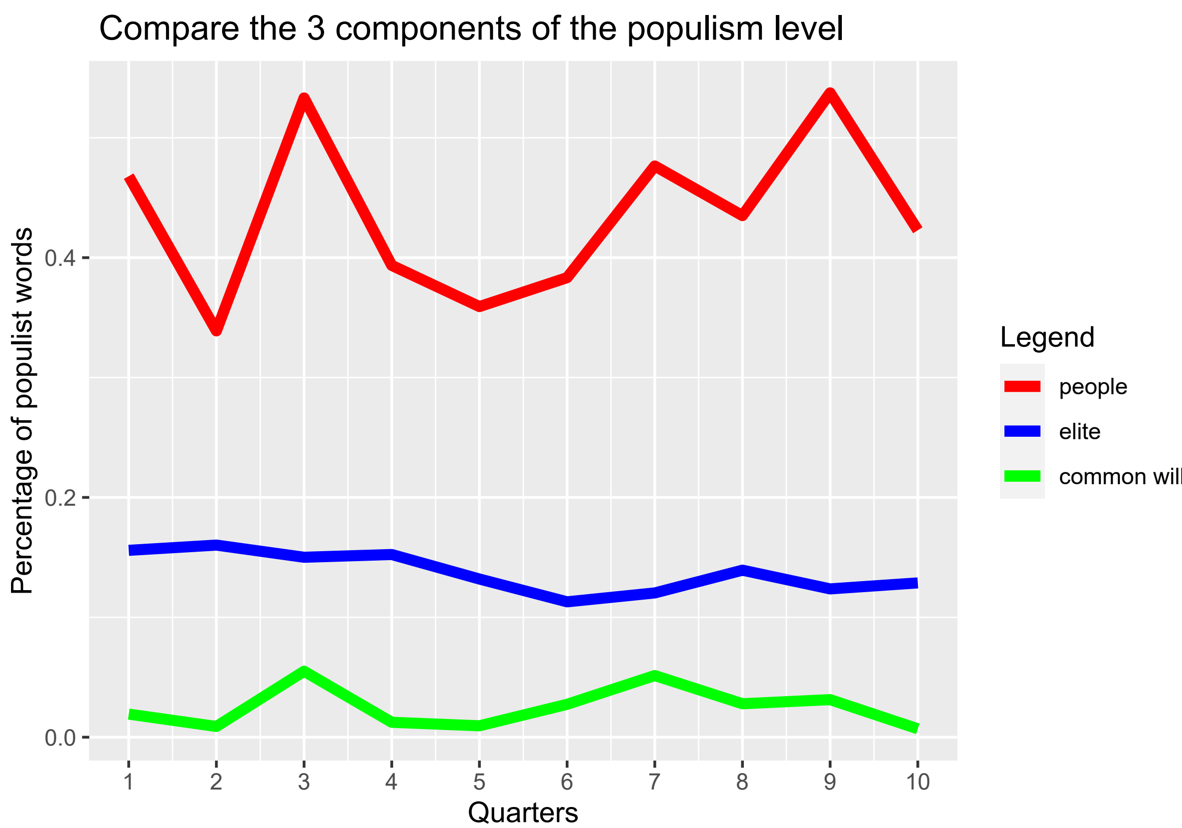



Populism level over quarters of the 'common will' component
Percentage of populist words - common will
0.05
0.04
0.03
0.02
0.01
1
2
3
4
5
6
7
8
9
10
Quarters



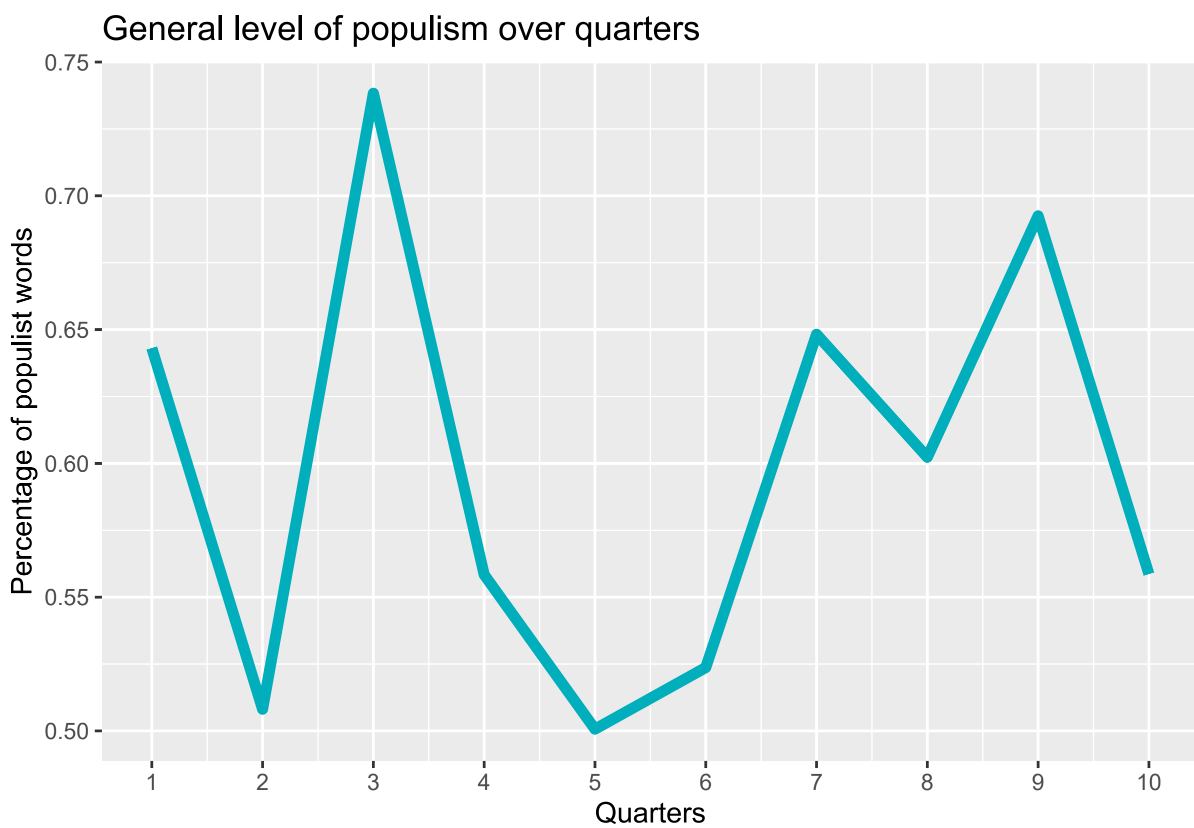

```
# compare the levels
p <- ggplot() +
  # plot people
  geom_line(data = data_quarter_people,
            aes(x = Group.1, y = perc, color = "people"), size = 2) +
  # plot common will
  geom_line(data = data_quarter_common,
            aes(x = Group.1, y = perc, color = "common will"), size = 2) +
  # plot elite
  geom_line(data = data_quarter_elite,
            aes(x = Group.1, y = perc, color = "elite"), size = 2) +
  scale_color_manual(name='Legend',
                     breaks=c('people', 'elite', 'common will'),
                     values=c('people'='red', 'elite'='blue', 'common will'='green'))+
```

```
  scale_x_continuous("Quarters", labels = as.character(data_quarter_people$Group.1),
                     breaks = data_quarter_people$Group.1)+
  ylab("Percentage of populist words")+
  labs(title = " Compare the 3 components of the populism level")

p
```

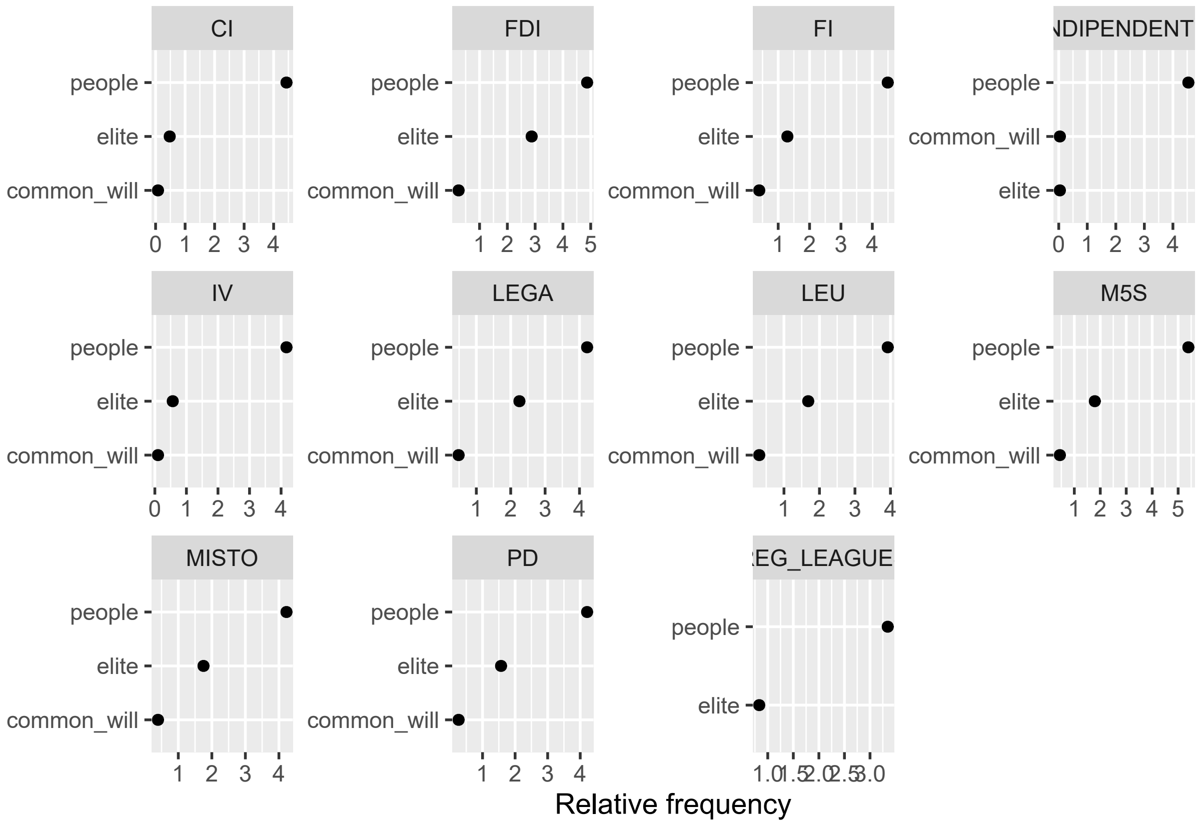



```
#Over time general level populism (quarters)
data_quarter_general <- aggregate(x = data_dict1$populism,  # Specify data column
          by = list(data_dict1$quarter),  # Specify group indicator
          FUN = mean) # Specify function (i.e. mean)
data_quarter_general$perc <- data_quarter_general$x
```

```r
# plot the level of populism
plot_general <- ggplot(data = data_quarter_general, aes(x = Group.1, y = perc))+
  geom_line(color = "#00AFBB", size = 2)+
  scale_x_continuous("Quarters", labels = as.character(data_quarter_general$Group.1),
                     breaks = data_quarter_general$Group.1)+
  ylab("Percentage of populist words")+
  labs(title = "General level of populism over quarters")
plot_general
```



3.2.3 Frequencies of the 3 components of populism for each parliamentary group

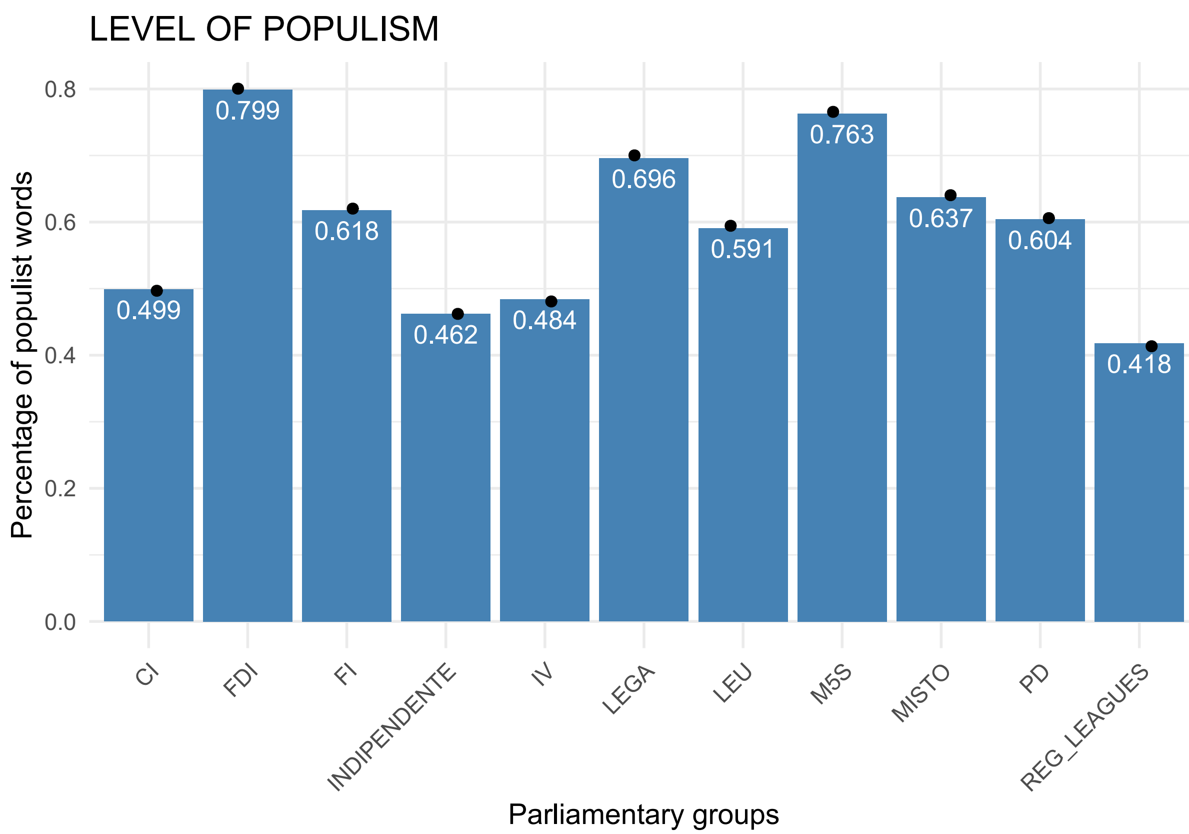


3.2.4 Ranking of parliamentary groups according to their level of populism

The code is only shown for the main "POPULISM" indicator but is identical for the single components

```
#By party no time (quarters)


# POPULISM
data_party <- aggregate(x = data_dict1$populism,  # Specify data column
          by = list(data_dict1$party_id),  # Specify group indicator
          FUN = mean) # Specify function (i.e. mean)
data_party$perc <- round(data_party$x,3)
```

```
ggplot(data=data_party, aes(x=Group.1, y=perc)) +
  geom_bar(stat="identity", fill="steelblue")+
  geom_text(aes(label=perc), vjust=1.6, color="white", size=3.5)+
  theme_minimal()+
  geom_jitter(width=0.15)+
  theme(axis.text.x = element_text(angle = 45, hjust=1))+
  ylab("Percentage of populist words") +
  xlab("Parliamentary groups")+
  labs(title = "LEVEL OF POPULISM")
```



```r
kable(data_party %>% select(Group.1, perc) %>%  arrange(desc(perc)),
      col.names = c("Party","Perc"),
      caption = "Populism")%>%
  kable_styling(latex_options = "HOLD_position")
```

Table 1: Populism

Party	Perc
FDI	0.799
M5S	0.763
LEGA	0.696
MISTO	0.637
FI	0.618
PD	0.604
LEU	0.591
CI	0.499
IV	0.484
INDIPENDENTE	0.462
REG_LEAGUES	0.418

Table 2: People

Party	Perc
M5S	0.539
FDI	0.487
INDIPENDENTE	0.454
FI	0.449
CI	0.444
MISTO	0.423
LEGA	0.422
PD	0.421
IV	0.417
LEU	0.392
REG_LEAGUES	0.335

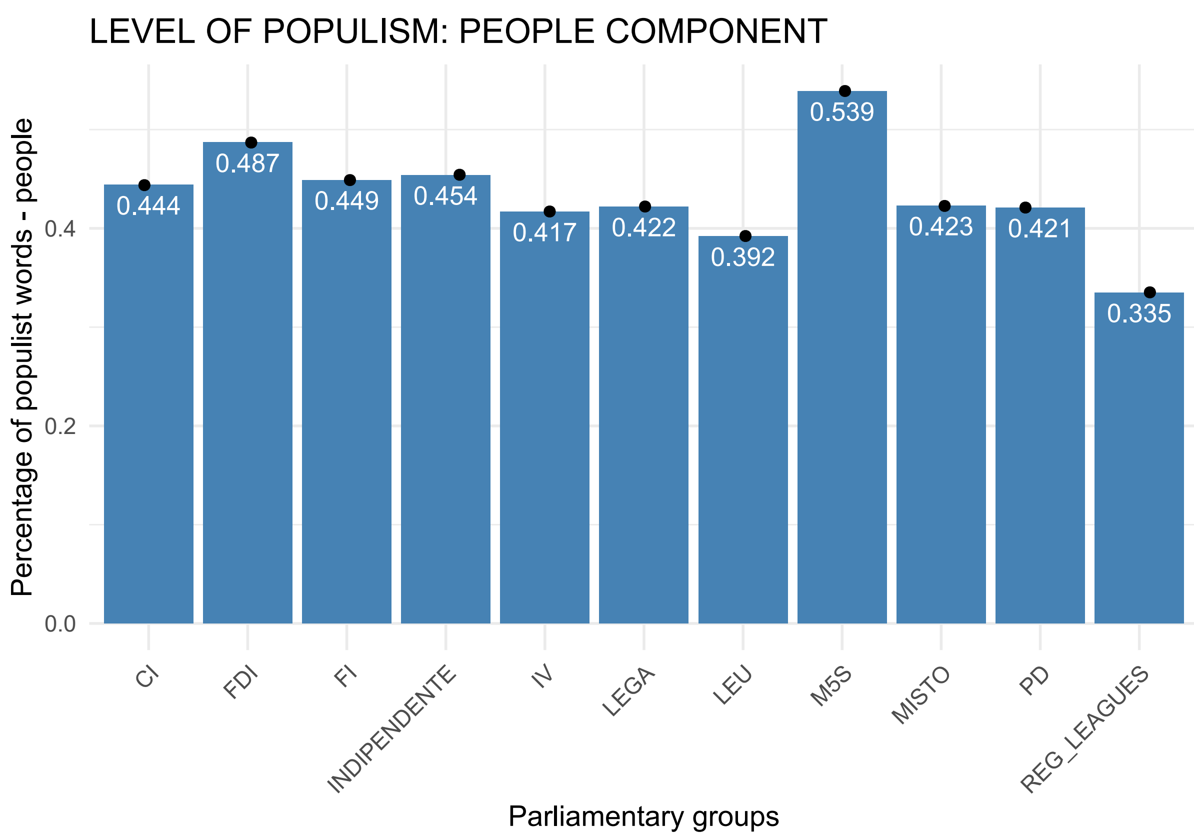


Table 3: Common will

Party	Perc
LEGA	0.048
M5S	0.044
FI	0.040
MISTO	0.039
LEU	0.030
PD	0.027
FDI	0.025
IV	0.010
CI	0.008
INDIPENDENTE	0.004
REG_LEAGUES	0.000

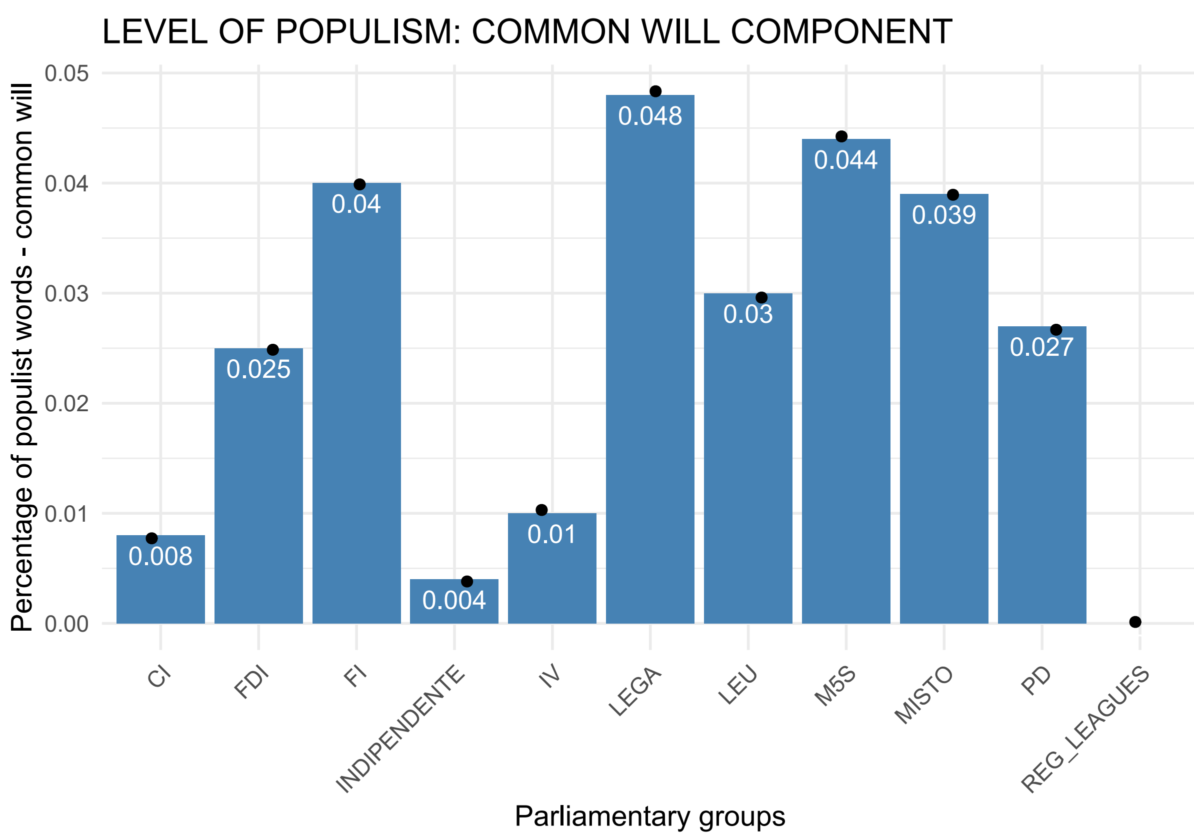
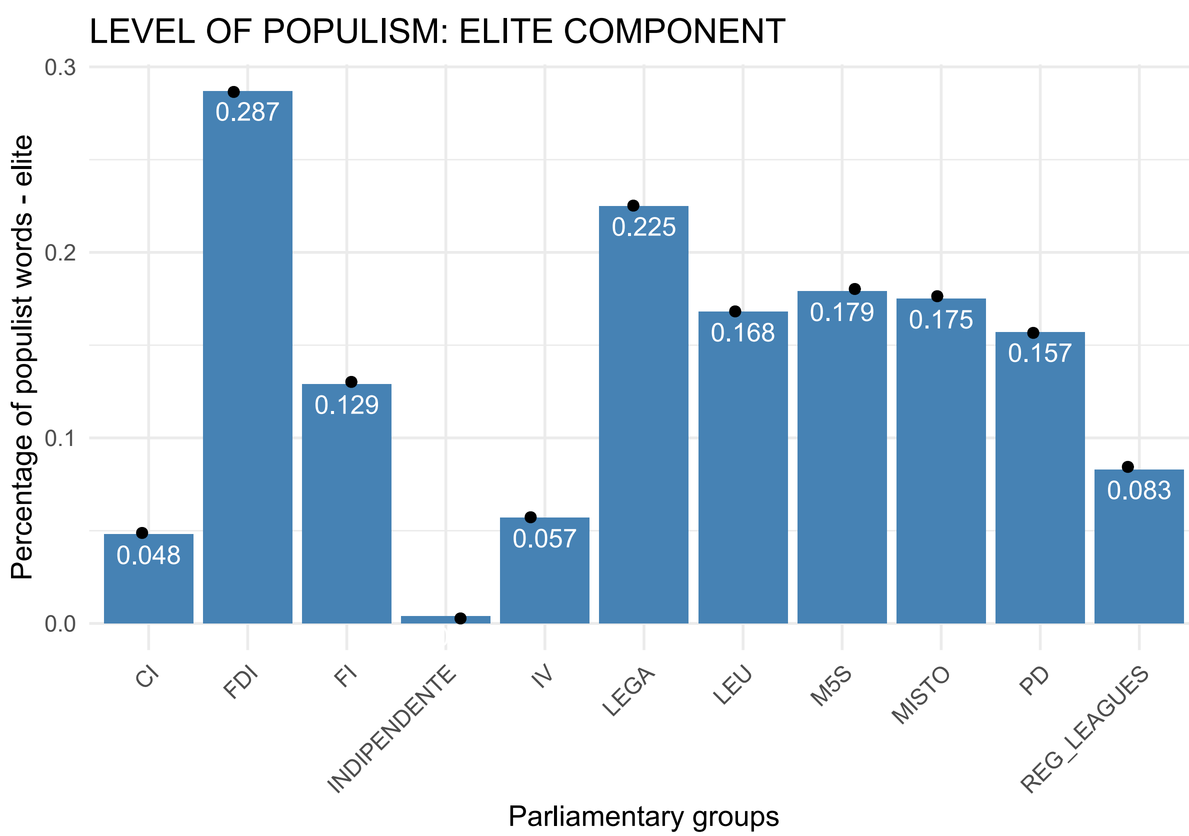


Table 4: Elite

Party	Perc
FDI	0.287
LEGA	0.225
M5S	0.179
MISTO	0.175
LEU	0.168
PD	0.157
FI	0.129
REG_LEAGUES	0.083
IV	0.057
CI	0.048
INDIPENDENTE	0.004



3.2.5 Bivariate regression for check t-test

```r
# bivariate regression for check t-test
data_dict1$factor_party <- as.factor(data_dict1$party_id)
data_dict1$factor_party <- relevel(data_dict1$factor_party, ref = "PD")

data_dict1$factor_quarter <- as.factor(data_dict1$quarter)
data_dict1$factor_quarter <- relevel(data_dict1$factor_quarter, ref = "8")

a3 <- lm(populism ~  factor_quarter + factor_party, data_dict1 )

summary(a3)
```

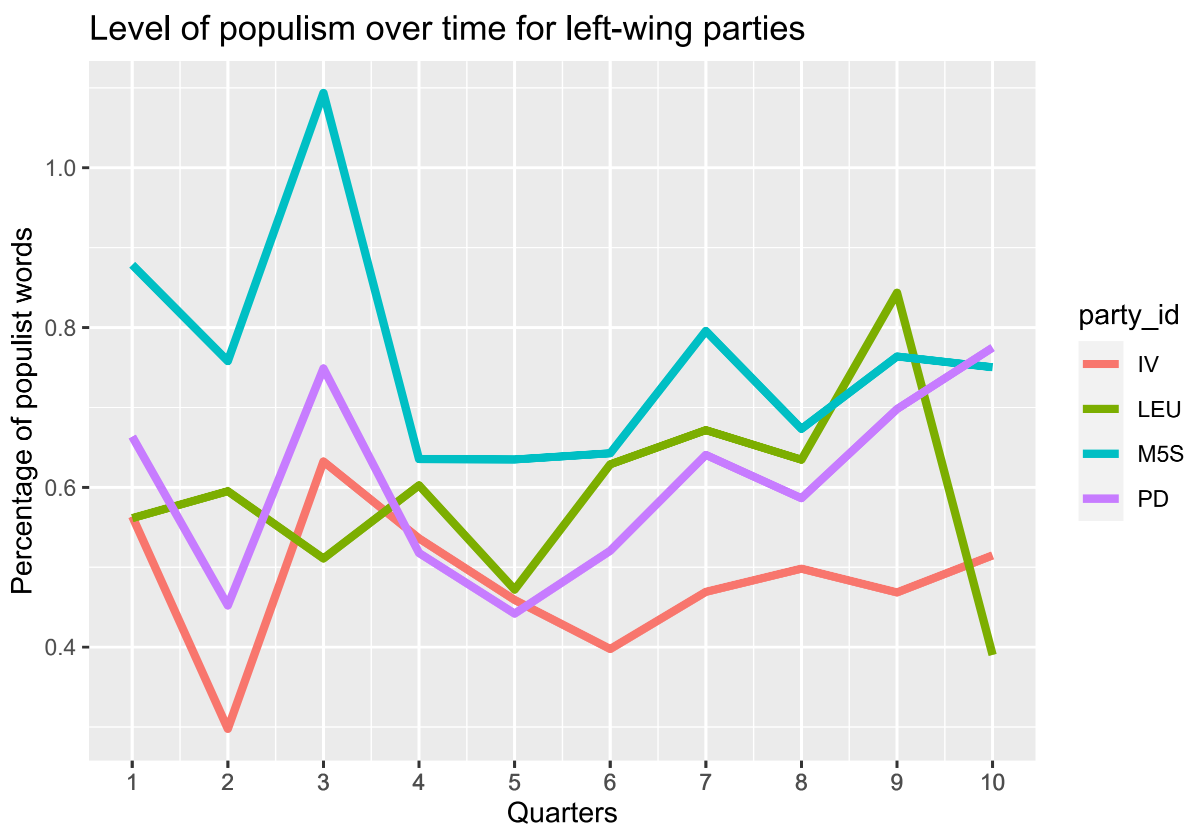
```
##
## Call:
## lm(formula = populism ~ factor_quarter + factor_party, data = data_dict1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.30617 -0.06571  0.00588  0.05535  0.32599
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 0.60934    0.05058  12.046  < 2e-16 ***
## factor_quarter1             0.04082    0.05058   0.807 0.421838
## factor_quarter2            -0.09418    0.05058  -1.862 0.065878 .
## factor_quarter3             0.13606    0.05058   2.690 0.008522 **
## factor_quarter4            -0.04390    0.05058  -0.868 0.387769
## factor_quarter5            -0.10164    0.05058  -2.009 0.047500 *
## factor_quarter6            -0.07861    0.05058  -1.554 0.123684
```

```
## factor_quarter7           0.04596    0.05058   0.909 0.365971
## factor_quarter9           0.09022    0.05058   1.783 0.077879 .
## factor_quarter10         -0.04369    0.05058  -0.864 0.390079
## factor_partyCI           -0.10503    0.05305  -1.980 0.050793 .
## factor_partyFDI           0.19458    0.05305   3.668 0.000414 ***
## factor_partyFI            0.01356    0.05305   0.256 0.798859
## factor_partyINDIPENDENTE -0.14233    0.05305  -2.683 0.008687 **
## factor_partyIV           -0.12078    0.05305  -2.277 0.025184 *
## factor_partyLEGA          0.09147    0.05305   1.724 0.088134 .
## factor_partyLEU          -0.01339    0.05305  -0.252 0.801282
## factor_partyM5S           0.15814    0.05305   2.981 0.003698 **
## factor_partyMISTO         0.03265    0.05305   0.615 0.539799
## factor_partyREG_LEAGUES  -0.18644    0.05305  -3.514 0.000693 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1186 on 90 degrees of freedom
## Multiple R-squared:  0.6326, Adjusted R-squared:  0.5551
## F-statistic: 8.157 on 19 and 90 DF,  p-value: 1.35e-12
```

3.2.6 Trends in the level of populism for each parliamentary group over time

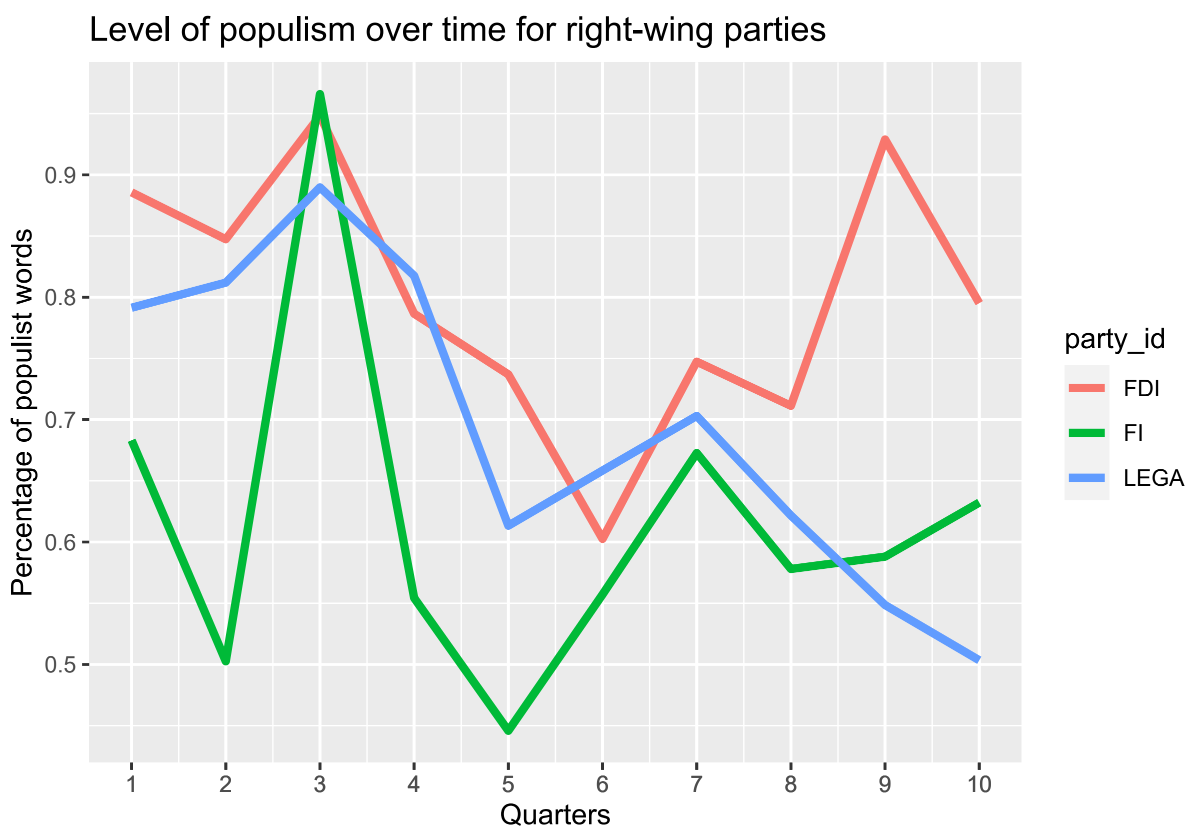
```r
#By party & time (quarters)
parties_time <- data_dict1 %>% select(populism, party_id, quarter)


right_party <- data_dict1 %>% select(populism, party_id, quarter) %>%
  filter(party_id == "FDI"|party_id =="FI"|party_id =="LEGA")
left_party <- data_dict1 %>% select(populism, party_id, quarter) %>%
  filter(party_id == "LEU"|party_id =="M5S"|party_id =="PD"|party_id =="IV")
```

```r
# Left parties in time
ggplot(left_party, aes(x=quarter, y=populism, color=party_id)) +
  geom_line(size=1.5)+
  scale_x_continuous("Quarters", labels = as.character(left_party$quarter),
                     breaks = left_party$quarter)+
  ylab("Percentage of populist words")+
  ggtitle("Level of populism over time for left-wing parties")
```




```r
# Right parties in time
ggplot(right_party, aes(x=quarter, y=populism, color=party_id)) +
  geom_line(size=1.5)+
  scale_x_continuous("Quarters", labels = as.character(right_party$quarter),
                     breaks = right_party$quarter)+
```

```
  ylab("Percentage of populist words")+
  ggtitle("Level of populism over time for right-wing parties")
```



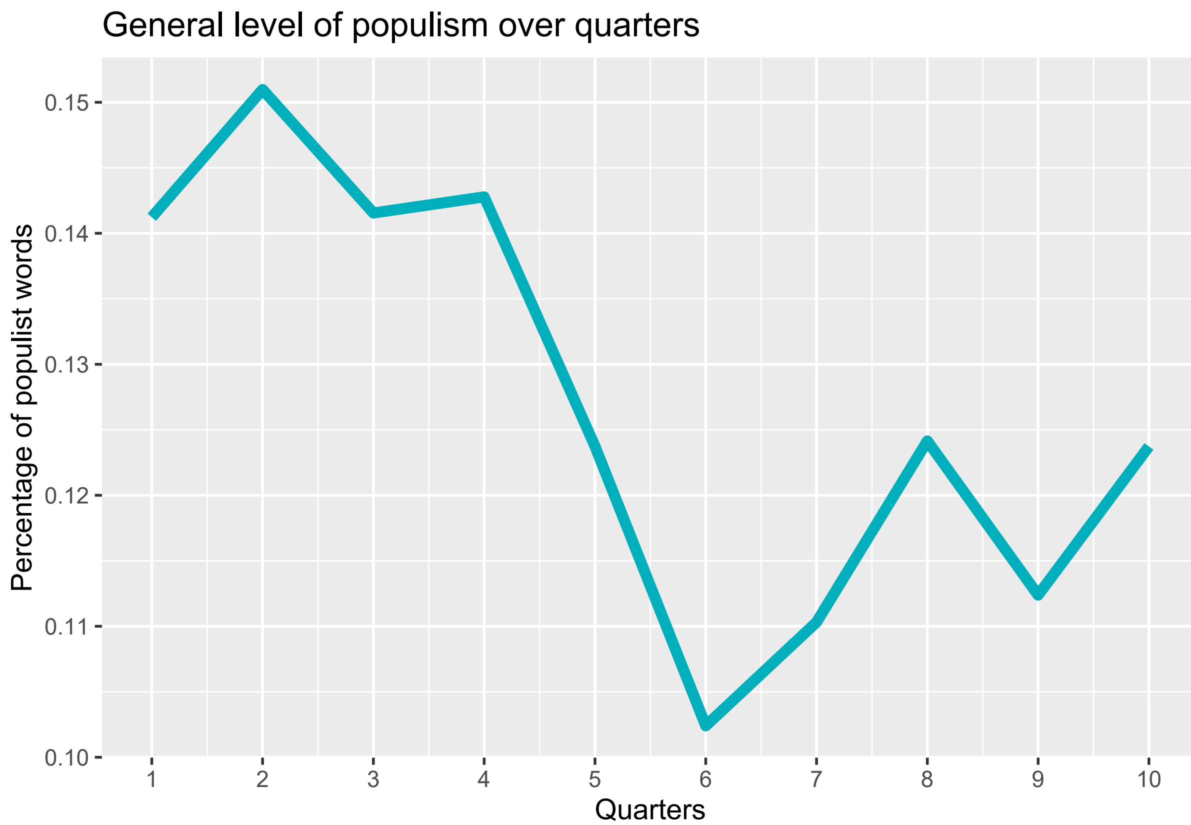
3.3 Rooduijn_Pauwels_Italian

```
# Dictionary analysis with Rooduijn_Pauwels_Italian
# By quarter
dfm_dict2  <- dfm_lookup(dfm_weigh_p_quart,
                         dictionary = Rooduijn_Pauwels_Italian)
```

```
data_dict2 <- dfm_dict2 %>%
  quanteda::convert(to = "data.frame") %>%
  cbind(docvars(dfm_dict2))
```

3.3.1 Level of populism over time

```
#Over time general level populism (quarters)
data_quarter_general2 <- aggregate(x = data_dict2$populism,  # Specify data column
          by = list(data_dict2$quarter),  # Specify group indicator
          FUN = mean) # Specify function (i.e. mean)
data_quarter_general2$perc <- data_quarter_general2$x * 100

# plot the level of populism
plot_general2 <- ggplot(data = data_quarter_general2, aes(x = Group.1, y = perc))+
  geom_line(color = "#00AFBB", size = 2)+
  scale_x_continuous("Quarters", labels = as.character(data_quarter_general2$Group.1),
  ylab("Percentage of populist words")+
  labs(title = "General level of populism over quarters")
plot_general2
```

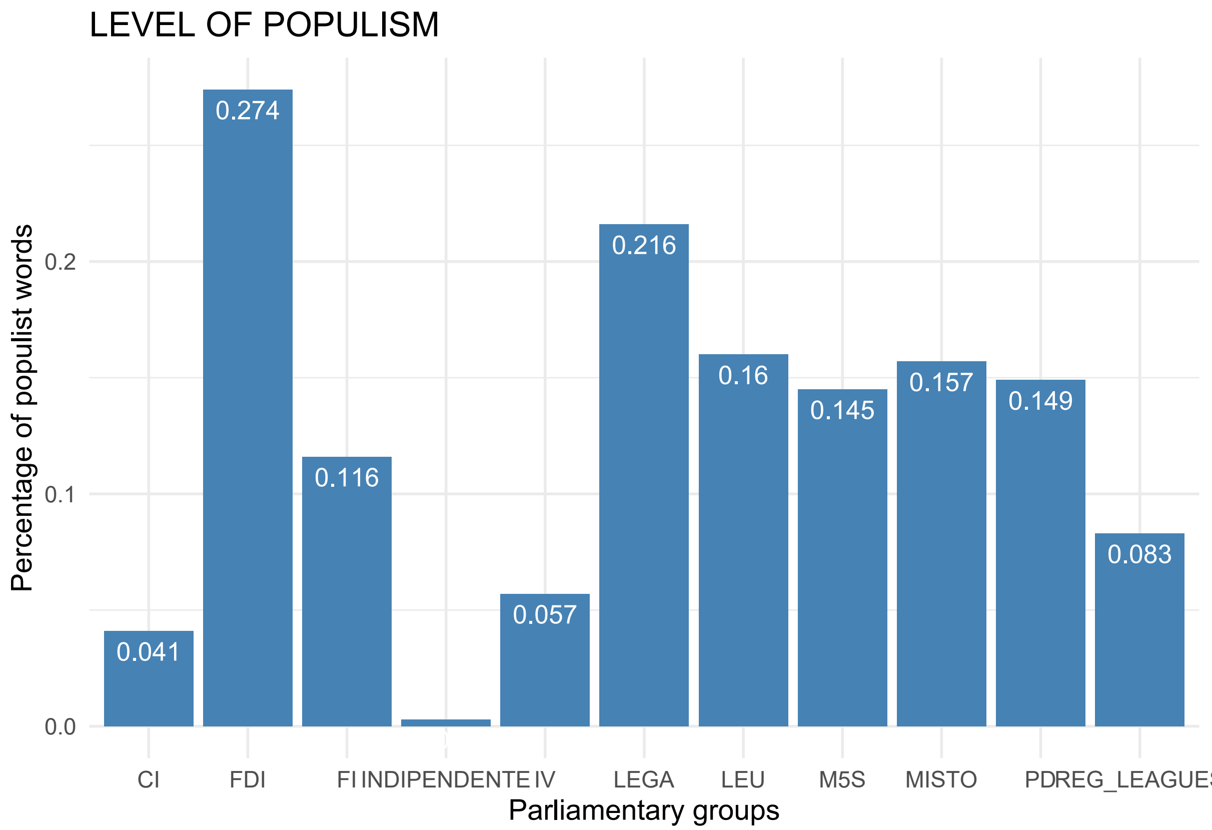


3.3.2 Ranking of parliamentary groups according their populism level

```
# POPULISM
data_party2 <- aggregate(x = data_dict2$populism,  # Specify data column
          by = list(data_dict2$party_id),  # Specify group indicator
          FUN = mean) # Specify function (i.e. mean)
data_party2$perc <- round(data_party2$x * 100 ,3)
kable(data_party2 %>% select(Group.1, perc) %>%  arrange(desc(perc)))
```

Group.1	perc
FDI	0.274
LEGA	0.216
LEU	0.160
MISTO	0.157
PD	0.149
M5S	0.145
FI	0.116
REG_LEAGUES	0.083
IV	0.057
CI	0.041
INDIPENDENTE	0.003

```
ggplot(data=data_party2, aes(x=Group.1, y=perc)) +
  geom_bar(stat="identity", fill="steelblue")+
  geom_text(aes(label=perc), vjust=1.6, color="white", size=3.5)+
  theme_minimal()+
  ylab("Percentage of populist words")+
  xlab("Parliamentary groups")+
  labs(title = "LEVEL OF POPULISM")
```


LEVEL OF POPULISM
Percentage of populist words
0.0
0.1
0.2
0.041
0.274
0.116
0.057
0.216
0.16
0.145
0.157
0.149
0.083
CI
FDI
FI
INDIPENDENTE
IV
LEGA
LEU
M5S
MISTO
PD
REG_LEAGUE
Parliamentary groups

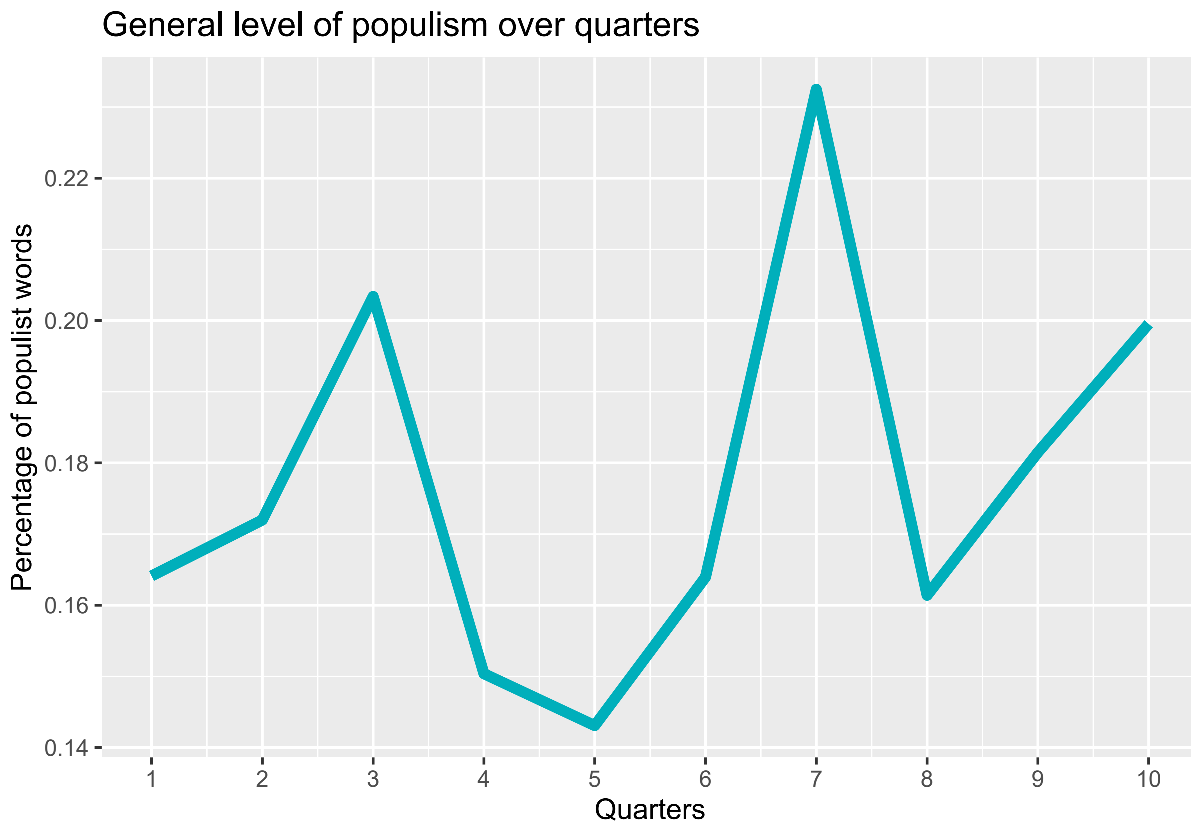
3.4 Grundl_Italian_adapted

```
# Dictionary analysis with Rooduijn_Pauwels_Italian
# By quarter
dfm_dict3  <- dfm_lookup(dfm_weigh_p_quart,
                         dictionary = Grundl_Italian_adapted)
```

```
data_dict3 <- dfm_dict3 %>%
  quanteda::convert(to = "data.frame") %>%
  cbind(docvars(dfm_dict3))
```

3.4.1 Level of populism in time

```
#Over time general level populism (quarters)
data_quarter_general3 <- aggregate(x = data_dict3$populism,  # Specify data column
          by = list(data_dict3$quarter),  # Specify group indicator
          FUN = mean) # Specify function (i.e. mean)
data_quarter_general3$perc <- data_quarter_general3$x * 100

# plot the level of populism
plot_general3 <- ggplot(data = data_quarter_general3, aes(x = Group.1, y = perc))+
  geom_line(color = "#00AFBB", size = 2)+
  scale_x_continuous("Quarters", labels = as.character(data_quarter_general3$Group.1),
  ylab("Percentage of populist words")+
  labs(title = "General level of populism over quarters")
plot_general3
```

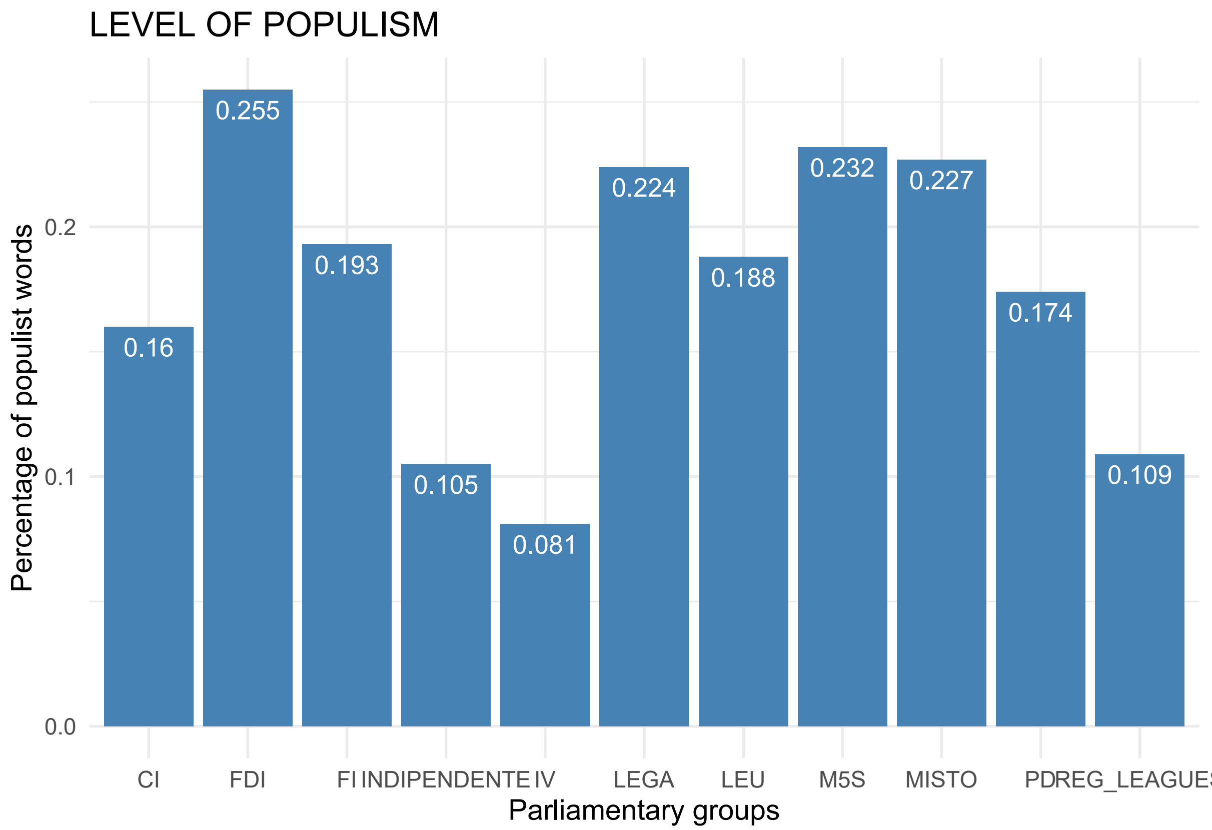


3.4.2 Most populist parliamentary group

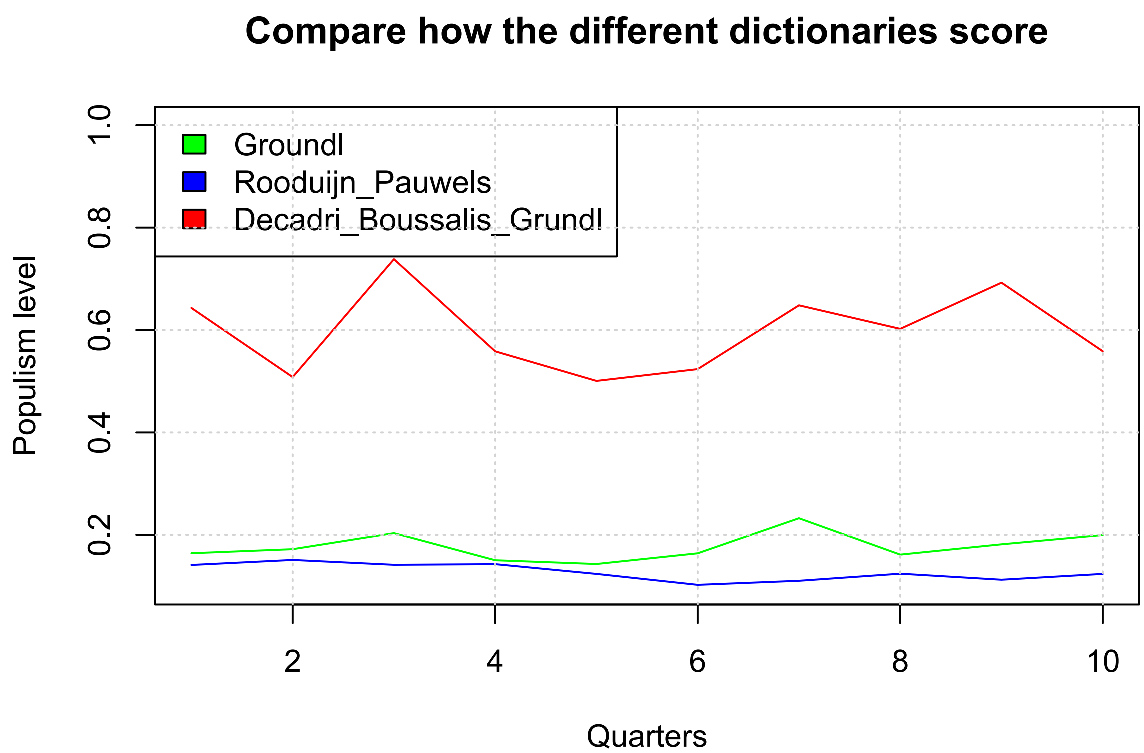
```
# POPULISM
data_party3 <- aggregate(x = data_dict3$populism,  # Specify data column
          by = list(data_dict3$party_id),  # Specify group indicator
          FUN = mean) # Specify function (i.e. mean)
data_party3$perc <- round(data_party3$x * 100 ,3)
kable(data_party3 %>% select(Group.1, perc) %>%  arrange(desc(perc)))
```

Group.1	perc
FDI	0.255
M5S	0.232
MISTO	0.227
LEGA	0.224
FI	0.193
LEU	0.188
PD	0.174
CI	0.160
REG_LEAGUES	0.109
INDIPENDENTE	0.105
IV	0.081

```
ggplot(data=data_party3, aes(x=Group.1, y=perc)) +
  geom_bar(stat="identity", fill="steelblue")+
  geom_text(aes(label=perc), vjust=1.6, color="white", size=3.5)+
  theme_minimal()+
  ylab("Percentage of populist words")+
  xlab("Parliamentary groups")+
  labs(title = "LEVEL OF POPULISM")
```


LEVEL OF POPULISM
Percentage of populist words
0.0
0.1
0.2
0.16
0.255
0.193
0.105
0.081
0.224
0.188
0.232
0.227
0.174
0.109
CI
FDI
FI
INDIPENDENTE
IV
LEGA
LEU
M5S
MISTO
PD
REG_LEAGUES
Parliamentary groups

3.5 Compare the general level of populism over time for the dictionaries



4 Emotion analysis

4.1 Import the LIWC2007_Dictionary

```
# Data
load("data/dfm.Rda")

# Dictionary LWIC Complete
LWIC_ITA <- dictionary(file = "data/large_files/Italian_LIWC2007_Dictionary.dic",
                       format = "LIWC")
```

```
## note: removing empty key: Formale
```

```
## note: removing empty key: Passivo
```

```
emotions <- c("Emo_Pos", "Emo_Neg", "Ansia", "Rabbia", "Tristezza", "Ottimismo" )

# Count the number of words
n.words <- c(
length(LWIC_ITA[["Emo_Pos"]]),
length(LWIC_ITA[["Emo_Neg"]]),
length(LWIC_ITA[["Ansia"]]),
length(LWIC_ITA[["Rabbia"]]),
length(LWIC_ITA[["Tristez"]]),
length(LWIC_ITA[["Ottimis"]])
)

num_words <-  data.frame(emotions,n.words)

# Extracting only the keys we need
```

```r
myLWIC_ITA <- dictionary(list(positive = LWIC_ITA[["Emo_Pos"]],
                              negative = LWIC_ITA[["Emo_Neg"]],
                              anxiety = LWIC_ITA[["Ansia"]],
                              anger = LWIC_ITA[["Rabbia"]],
                              sadness = LWIC_ITA[["Tristez"]],
                              optimism = LWIC_ITA[["Ottimis"]]
                              ))

myLWIC_ITA_sent <- dictionary(list(positive = LWIC_ITA[["Emo_Pos"]],
                              negative = LWIC_ITA[["Emo_Neg"]]))
```

```r
kable(num_words)
```

emotions	n.words
Emo_Pos	200
Emo_Neg	663
Ansia	65
Rabbia	227
Tristezza	226
Ottimismo	93

4.2 Group and weight the dfm

```r
# By party & quarter
dfm_weigh_p_quart <- dfm_group(DFM, groups = interaction(party_id, quarter))%>%
  dfm_weight(scheme = "prop")
```

4.3 Apply the dictionary

```
# Apply Dictionary to DFM
DFM_emotions <- dfm_lookup(dfm_weigh_p_quart,
                           dictionary = myLWIC_ITA)
DFM_emotions
```

```
## Document-feature matrix of: 110 documents, 6 features (0.76% sparse) and 3 docvars.
##                  features
## docs                 positive   negative     anxiety       anger     sadness
##   CI.1            0.008060854 0.02236603 0.003405995 0.006471390 0.004541326
##   FDI.1           0.006416312 0.02893245 0.002834199 0.011061250 0.006140765
##   FI.1            0.006498830 0.02547256 0.003243474 0.007675035 0.006974064
##   INDIPENDENTE.1 0.005129667 0.01567398 0.001994870 0.005984611 0.003989741
##   IV.1            0.008545455 0.02309091 0.003272727 0.009272727 0.006000000
##   LEGA.1          0.006352373 0.02593448 0.003005565 0.008426081 0.006194876
##                  features
## docs                optimism
##   CI.1            0.01089918
##   FDI.1           0.01487955
##   FI.1            0.01447089
##   INDIPENDENTE.1 0.01025933
##   IV.1            0.01600000
##   LEGA.1          0.01257350
## [ reached max_ndoc ... 104 more documents ]
```

4.3.1 Transform the DFM into an ordinary dataframe

```r
data_dict_emo <- DFM_emotions %>%
  quanteda::convert(to = "data.frame") %>%
  cbind(docvars(DFM_emotions))

# Create a new variable with the difference of negative - positive emotions
data_dict_emo$negative_prevalence <-
  (data_dict_emo$negative - data_dict_emo$positive)

# Transform the proportion into percentage
data_dict_emo <- data_dict_emo %>%
  mutate(positive =  positive * 100,
                  negative = negative * 100,
                  anxiety = anxiety * 100,
                  anger = anger *100,
                  sadness = sadness * 100,
                  optimism = optimism *100,
                  negative_prevalence = negative_prevalence *100)
```

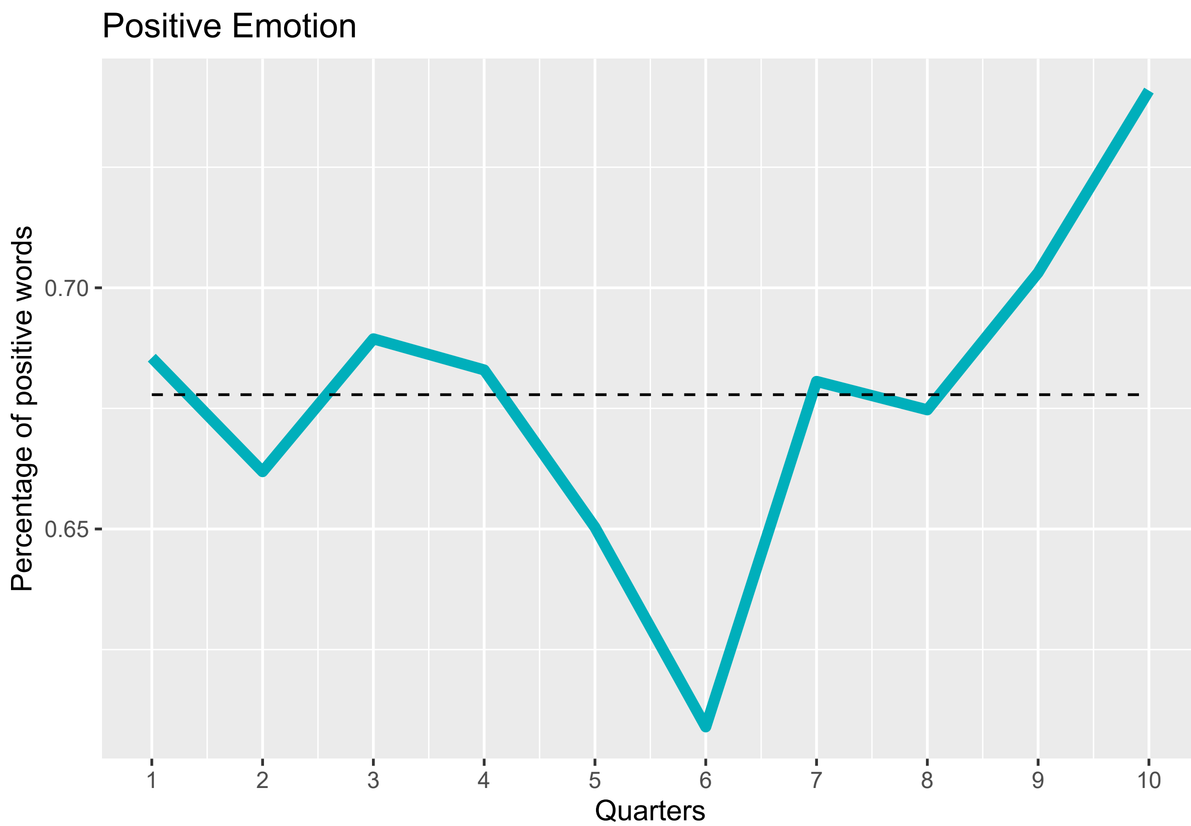
4.4 Percentage of the emotions in time

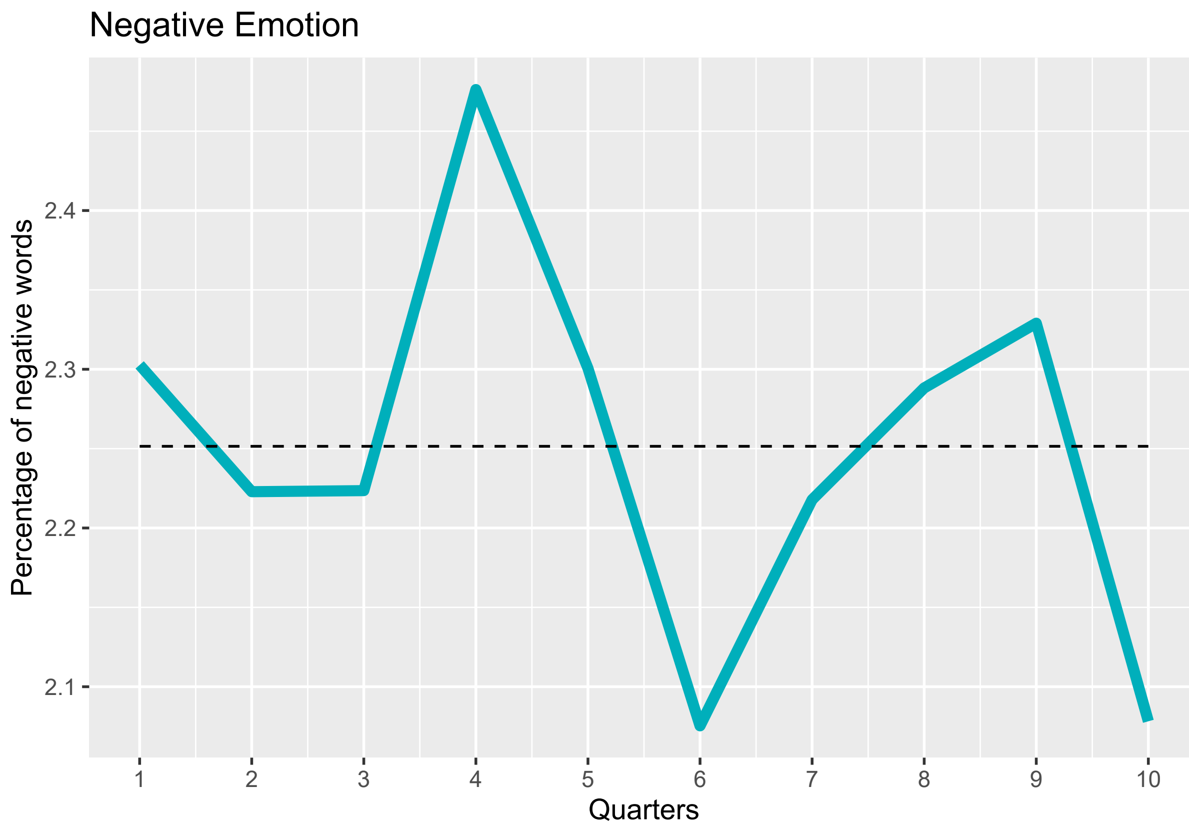
These are the start and end dates of the quarters covered by the analysis

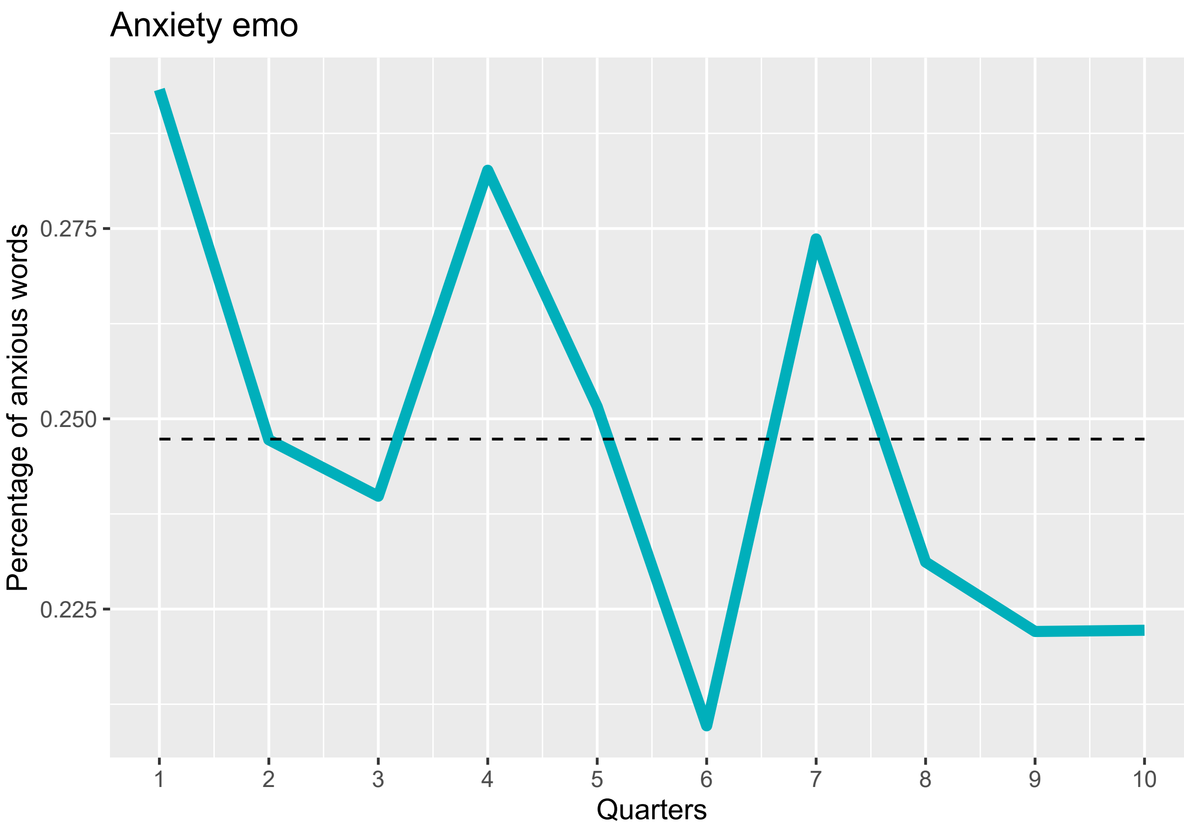
Trimester	from	to
1	01 January 2020	31 March 2020
2	01 April 2020	30 June 2020
3	01 July 2020	30 September 2020
4	01 October 2020	31 December 2020
5	01 January 2021	31 March 2021
6	01 April 2021	30 June 2021
7	01 July 2021	30 September 2021
8	01 October 2021	31 December 2021
9	01 January 2022	31 March 2022
10	01 April 2022	18 April 2022

The code is only shown for 'positive' but is identical for all emotions

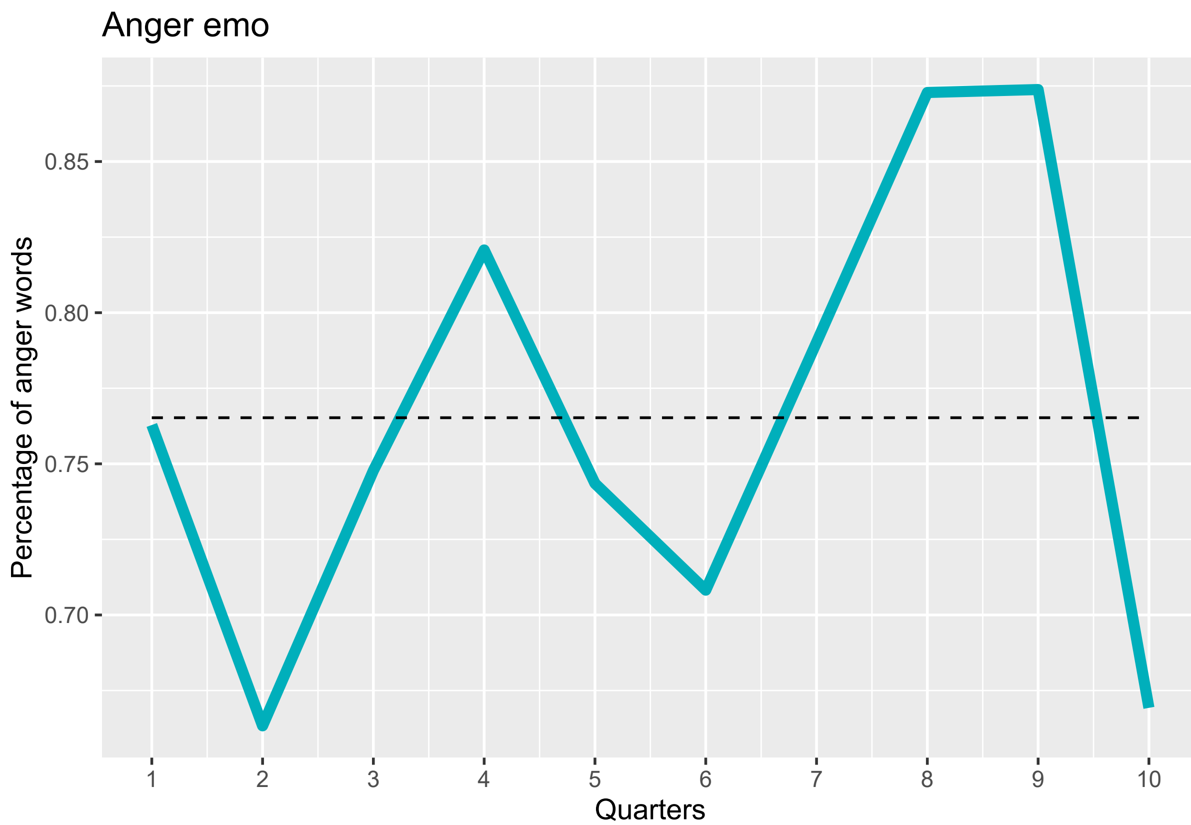
```
#Over time POSITIVE (quarters)
data_quarter_positive <- aggregate(x = data_dict_emo$positive,  # Specify data column
          by = list(data_dict_emo$quarter),  # Specify group indicator
          FUN = mean) # Specify function (i.e. mean)
data_quarter_positive$perc <- data_quarter_positive$x


mean_positive <- mean(data_quarter_positive$perc)


# plot
plot_positive <- ggplot(data = data_quarter_positive, aes(x = Group.1, y = perc))+
  geom_line(color = "#00AFBB", size = 2)+
  scale_x_continuous("Quarters", labels = as.character(data_quarter_positive$Group.1),
  geom_line(aes(x = Group.1, y = mean(perc)),linetype = "dashed")+
  ylab("Percentage of positive words")+
  labs(title = "Positive Emotion")
plot_positive
```

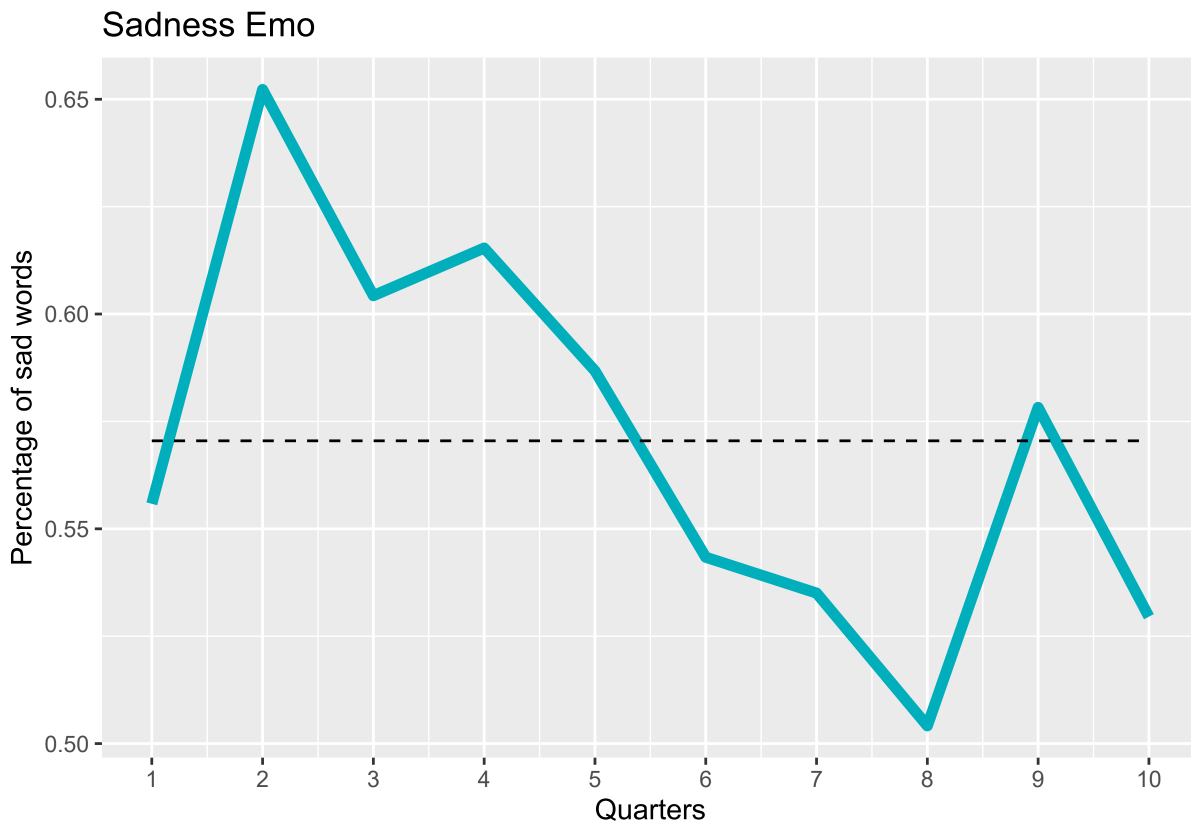

Positive Emotion
Percentage of positive words
0.70
0.65
1
2
3
4
5
6
7
8
9
10
Quarters

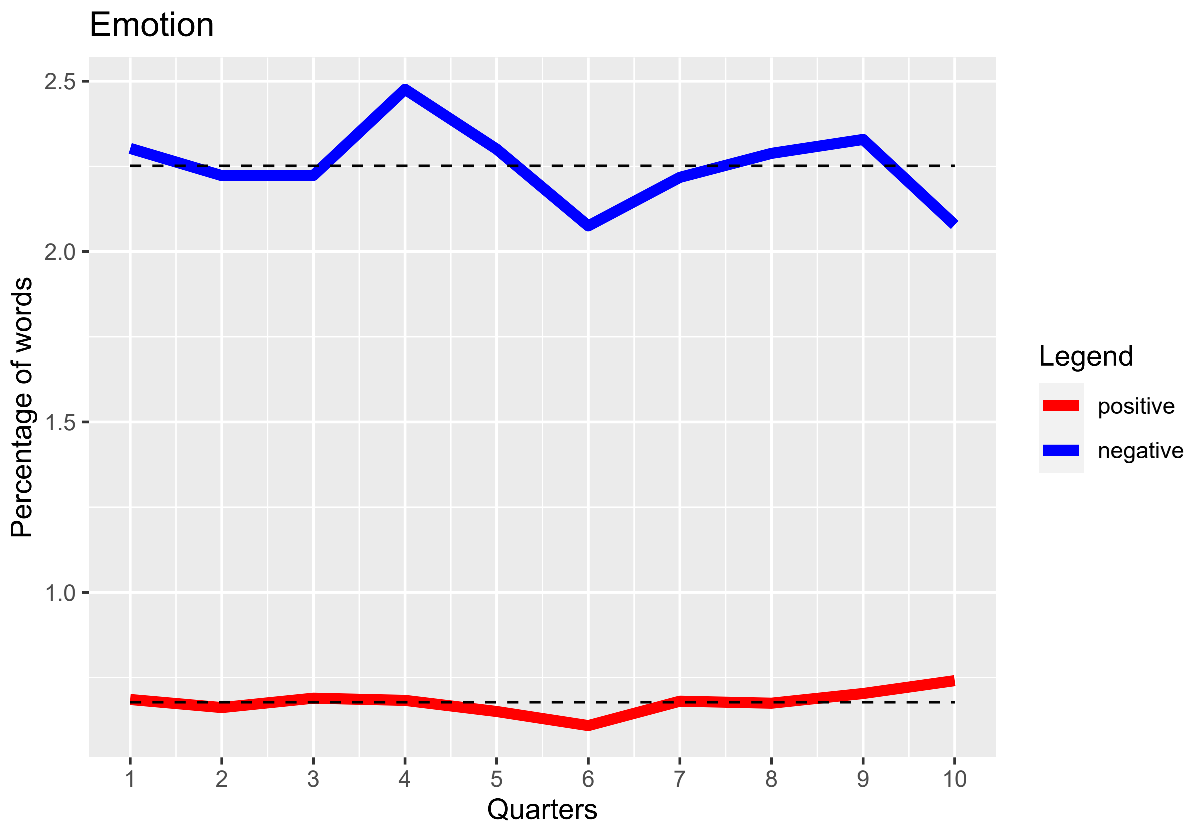

Negative Emotion
Percentage of negative words
2.4
2.3
2.2
2.1
1
2
3
4
5
6
7
8
9
10
Quarters

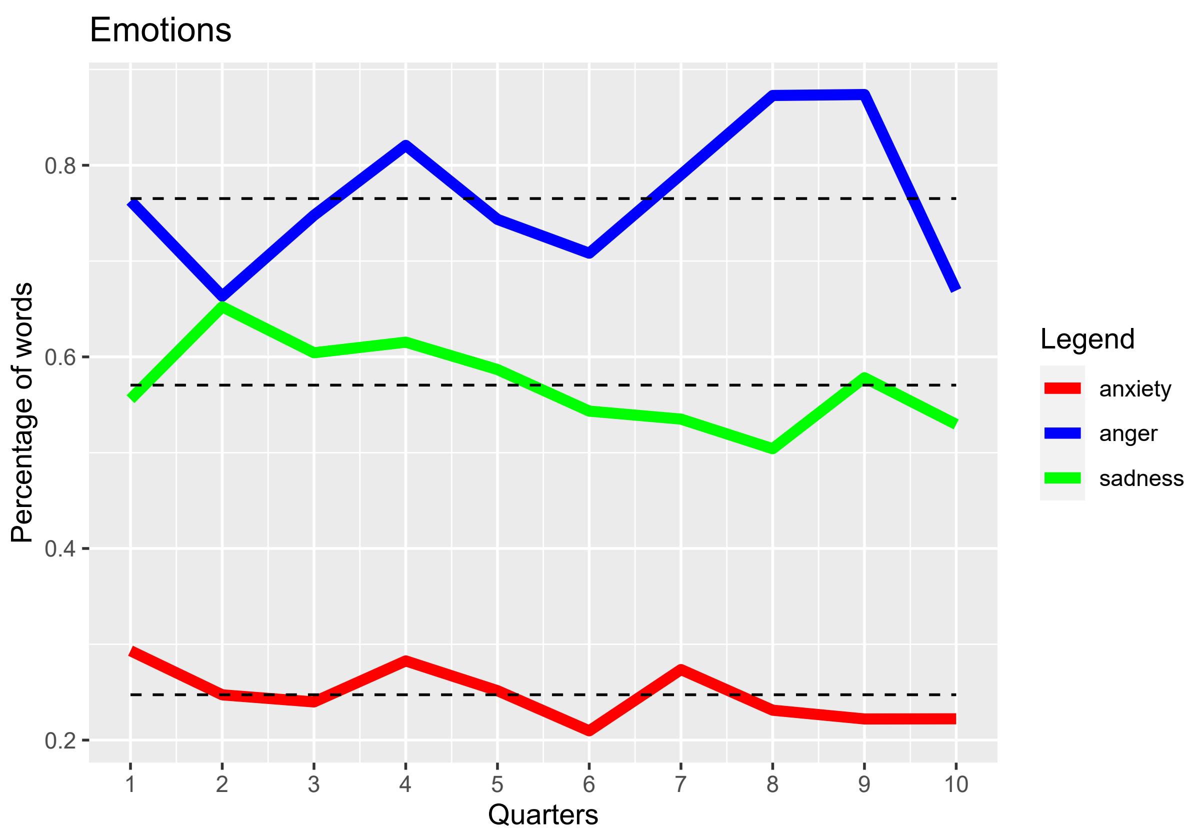


Anxiety emo
Percentage of anxious words
0.275
0.250
0.225
1
2
3
4
5
6
7
8
9
10
Quarters

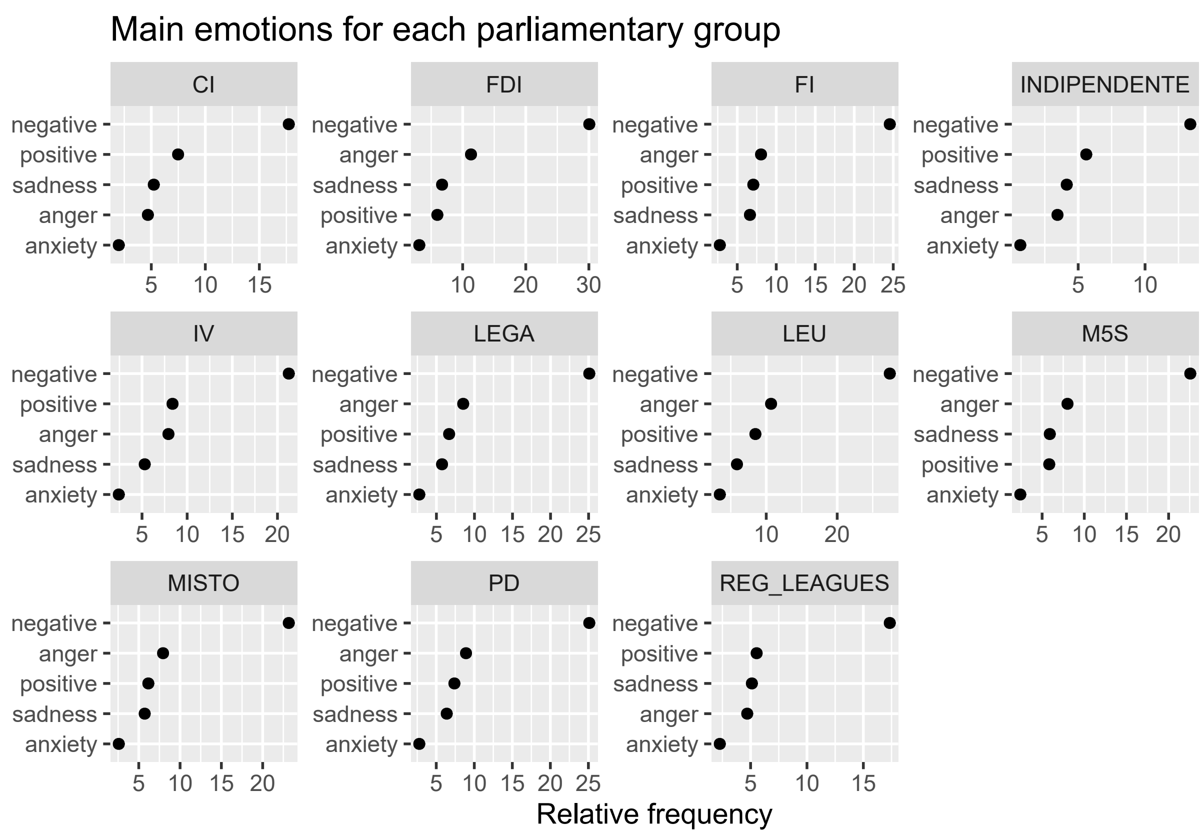

Anger emo
0.85
0.80
0.75
0.70
Percentage of anger words
1
2
3
4
5
6
7
8
9
10
Quarters


Sadness Emo
Percentage of sad words
0.65
0.60
0.55
0.50
1
2
3
4
5
6
7
8
9
10
Quarters


Emotion
2.5
2.0
1.5
1.0
Percentage of words
1
2
3
4
5
6
7
8
9
10
Quarters
Legend
positive
negative


Emotions
Percentage of words
0.8
0.6
0.4
0.2
1
2
3
4
5
6
7
8
9
10
Quarters
Legend
anxiety
anger
sadness

4.5 Main emotion for each parliamentary group



The code is only shown for 'positive' but is identical for all emotions

```
# POSITIVE
data_party_positive <- aggregate(x = data_dict_emo$positive,  # Specify data column
          by = list(data_dict_emo$party_id),  # Specify group indicator
          FUN = mean) # Specify function (i.e. mean)
data_party_positive$perc <- round(data_party_positive$x,3)


kable(data_party_positive %>%
        select(Group.1, perc) %>%
        arrange(desc(perc)), caption = "POSITIVE")
```

Table 5: POSITIVE

Group.1	perc
LEU	0.847
IV	0.838
CI	0.748
PD	0.738
FI	0.706
LEGA	0.667
MISTO	0.616
FDI	0.598
M5S	0.584
INDIPENDENTE	0.560
REG_LEAGUES	0.554

```
ggplot(data=data_party_positive, aes(x=Group.1, y=perc)) +
  geom_bar(stat="identity", fill="steelblue")+
  geom_text(aes(label=perc), vjust=0, color="black", size=3.5)+
  geom_abline(slope=0, intercept= mean(data_party_positive$perc),lty=2) +
  theme_minimal()+
  xlab("Parliamentary group")+
  labs(title = "Positive Emotion")+
  coord_flip()
```

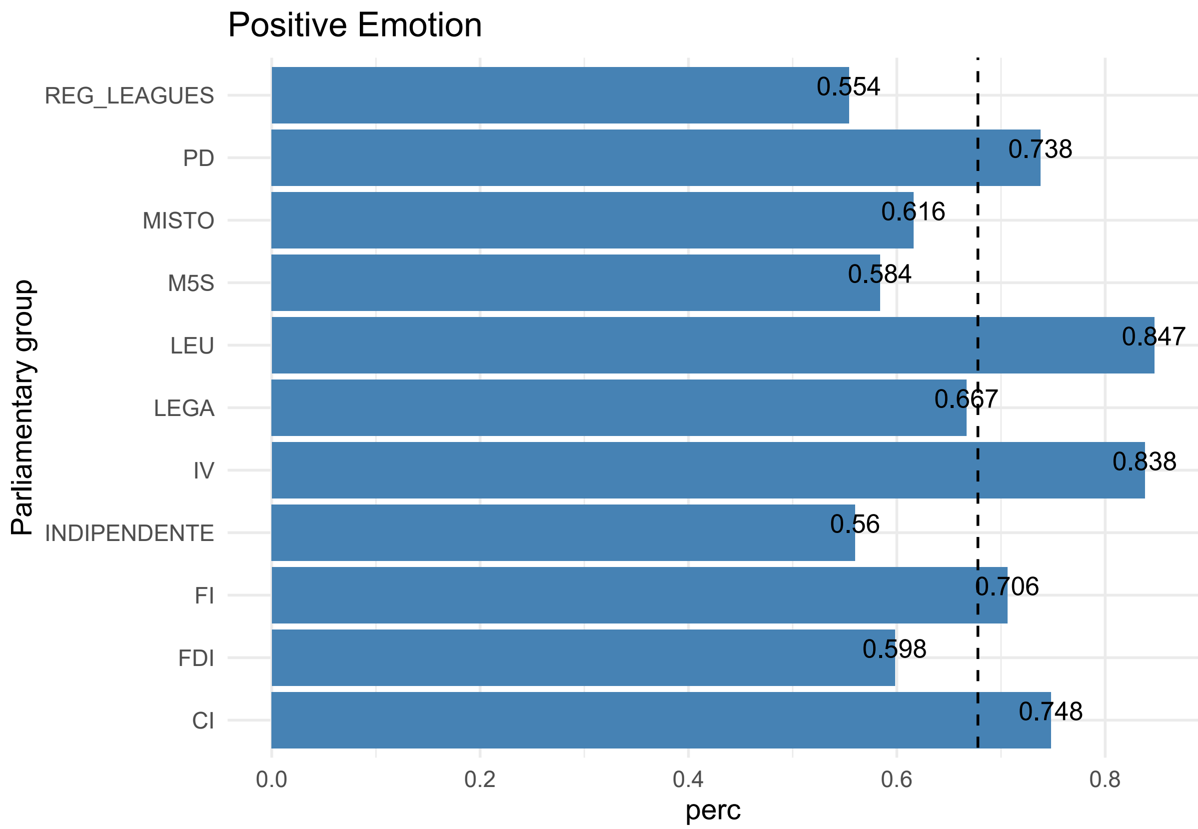

Positive Emotion
Parliamentary group
REG_LEAGUES
PD
MISTO
M5S
LEU
LEGA
IV
INDIPENDENTE
FI
FDI
CI
0.554
0.738
0.616
0.584
0.847
0.667
0.838
0.56
0.706
0.598
0.748
0.0
0.2
0.4
0.6
0.8
perc

Table 6: NEGATIVE

Group.1	perc
FDI	3.006
LEU	2.741
PD	2.512
LEGA	2.509
FI	2.455
MISTO	2.316
M5S	2.257
IV	2.125
CI	1.772
REG_LEAGUES	1.734
INDIPENDENTE	1.338

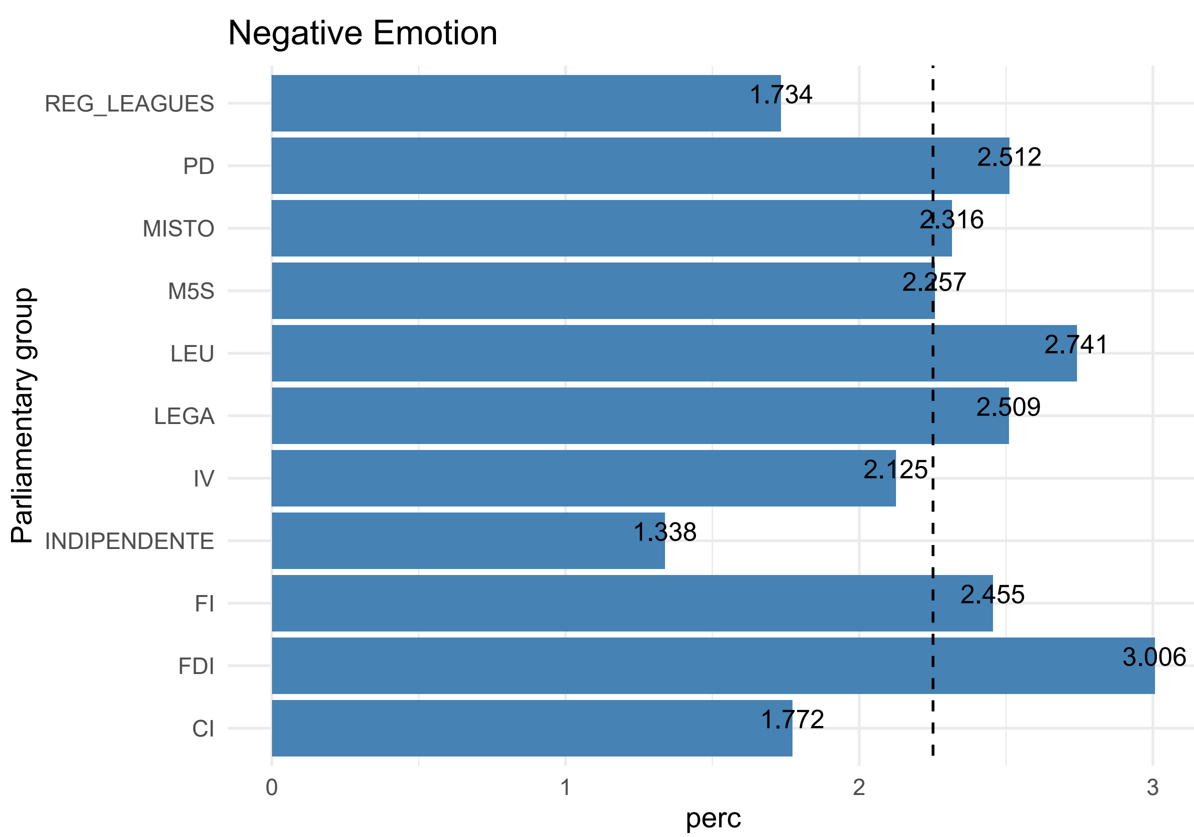


Table 7: ANXIETY

Group.1	perc
LEU	0.345
FDI	0.312
PD	0.277
FI	0.276
LEGA	0.275
MISTO	0.258
IV	0.243
M5S	0.241
REG_LEAGUES	0.227
CI	0.199
INDIPENDENTE	0.067

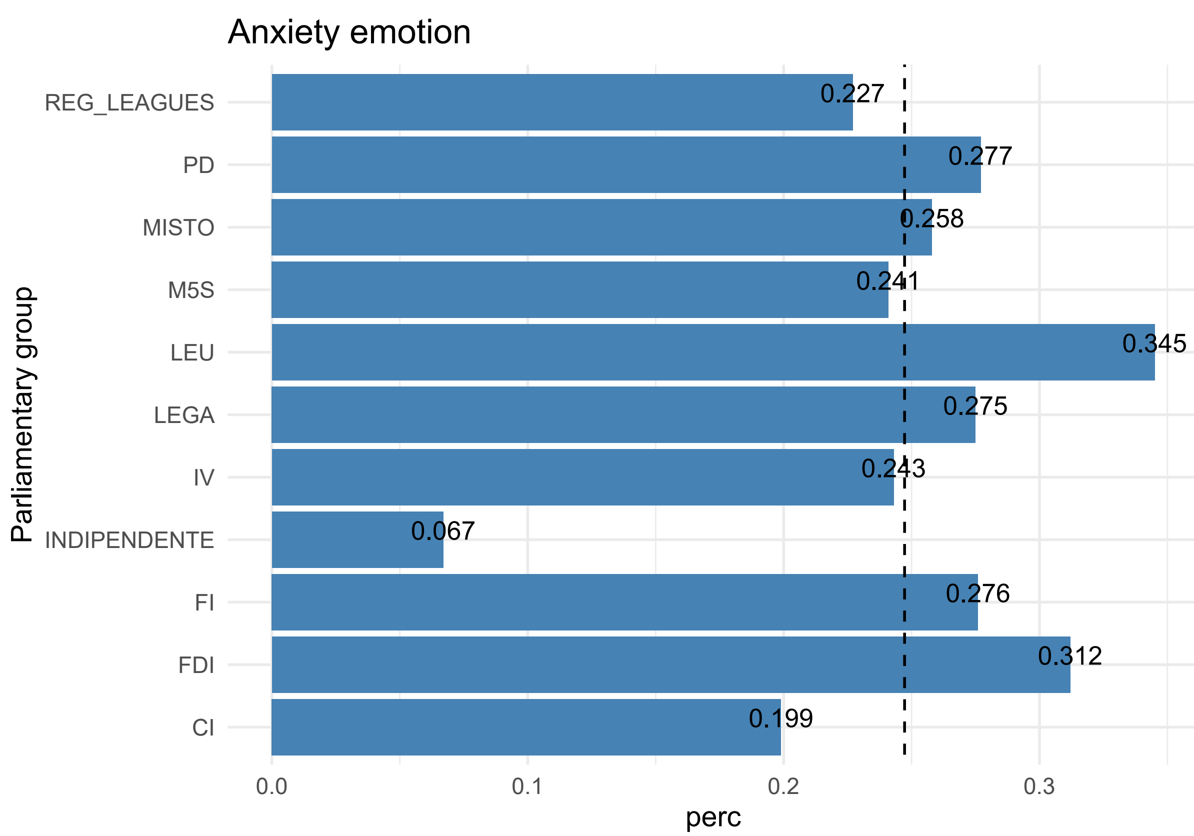


Table 8: ANGER

Group.1	perc
FDI	1.132
LEU	1.068
PD	0.891
LEGA	0.852
FI	0.805
M5S	0.801
MISTO	0.794
IV	0.793
REG_LEAGUES	0.470
CI	0.468
INDIPENDENTE	0.345

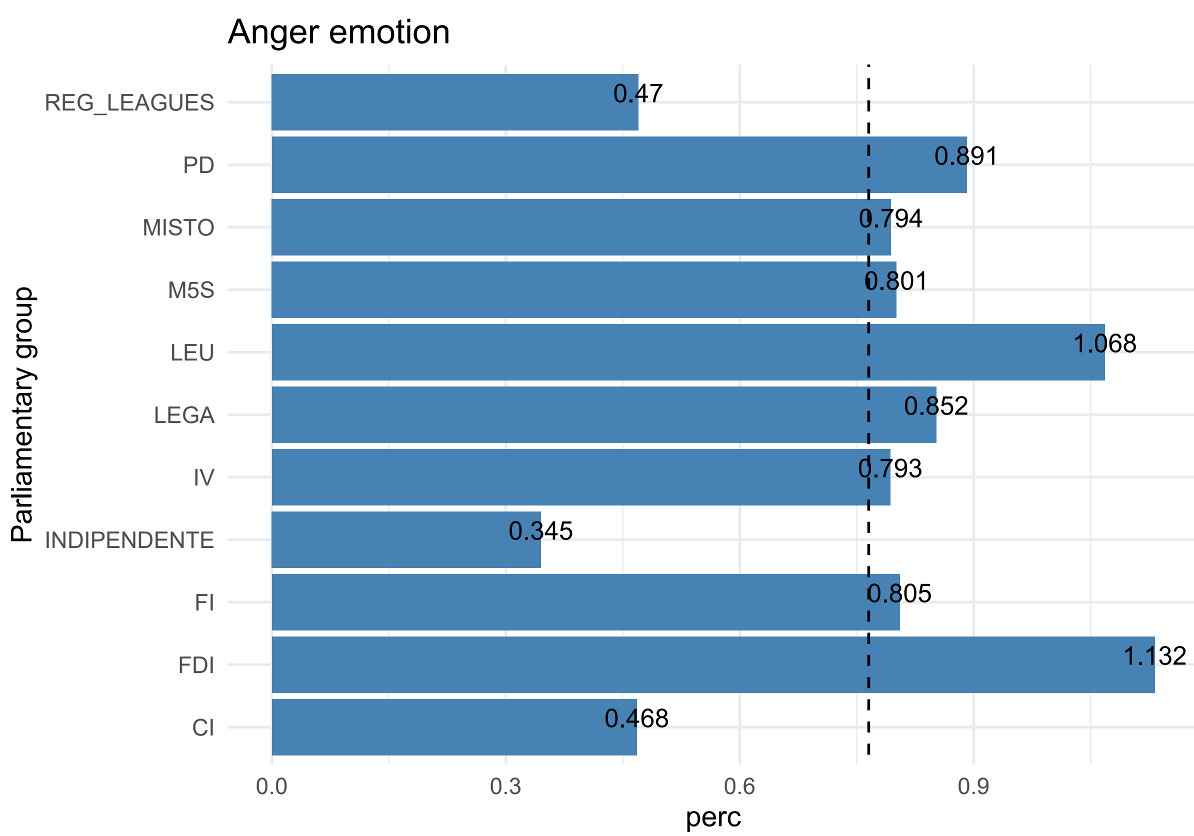
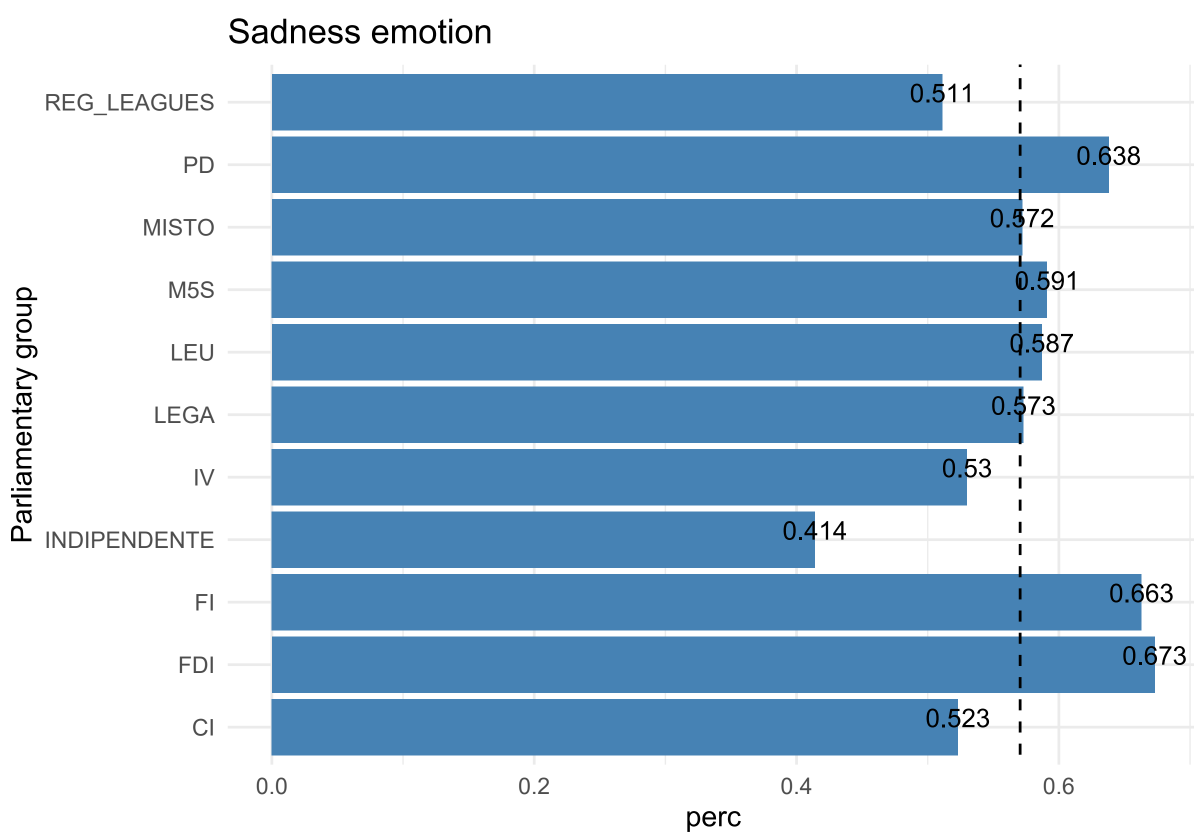


Table 9: SADNESS

Group.1	perc
FDI	0.673
FI	0.663
PD	0.638
M5S	0.591
LEU	0.587
LEGA	0.573
MISTO	0.572
IV	0.530
CI	0.523
REG_LEAGUES	0.511
INDIPENDENTE	0.414



4.5.1 Are the average values of positive/negative emotions for each party statistically different from each other?

The reference category is PD

```r
# bivariate regression for check t-test

# create the factor variables for party and quarter
data_dict_emo$factor_party <- as.factor(data_dict_emo$party_id)
data_dict_emo$factor_quarter <- as.factor(data_dict_emo$quarter)

# Check the mean values
summary(data_dict_emo$positive)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.3281  0.5863  0.6542  0.6778  0.7546  1.1593
```

```r
summary(data_dict_emo$negative)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.9522  1.9364  2.3318  2.2515  2.5867  3.2025
```

```r
# Set PD as reference category for party_id
data_dict_emo$factor_party <- relevel(data_dict_emo$factor_party, ref = "PD")
# Set 5 as reference category for quarter
data_dict_emo$factor_quarter <- relevel(data_dict_emo$factor_quarter, ref = "5")

# Run the regressions

# POSITIVE
positive_model <- lm(positive ~  factor_quarter + factor_party, data_dict_emo )
summary(positive_model)
```

```
##
## Call:
```

```
## lm(formula = positive ~ factor_quarter + factor_party, data = data_dict_emo)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.26194 -0.06684  0.00093  0.04680  0.33861
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)                0.710990   0.052210  13.618  < 2e-16 ***
## factor_quarter1            0.035165   0.052210   0.674  0.50234
## factor_quarter2            0.011541   0.052210   0.221  0.82556
## factor_quarter3            0.039079   0.052210   0.748  0.45611
## factor_quarter4            0.032630   0.052210   0.625  0.53358
## factor_quarter6           -0.041367   0.052210  -0.792  0.43026
## factor_quarter7            0.030252   0.052210   0.579  0.56376
## factor_quarter8            0.024362   0.052210   0.467  0.64191
## factor_quarter9            0.052797   0.052210   1.011  0.31462
## factor_quarter10           0.090541   0.052210   1.734  0.08632 .
## factor_partyCI             0.009462   0.054759   0.173  0.86321
## factor_partyFDI           -0.140003   0.054759  -2.557  0.01224 *
## factor_partyFI            -0.032835   0.054759  -0.600  0.55026
## factor_partyINDIPENDENTE -0.178239   0.054759  -3.255  0.00160 **
## factor_partyIV             0.099436   0.054759   1.816  0.07272 .
## factor_partyLEGA          -0.071907   0.054759  -1.313  0.19247
## factor_partyLEU            0.108649   0.054759   1.984  0.05029 .
## factor_partyM5S           -0.154273   0.054759  -2.817  0.00595 **
## factor_partyMISTO         -0.122489   0.054759  -2.237  0.02776 *
## factor_partyREG_LEAGUES  -0.184902   0.054759  -3.377  0.00109 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 0.1224 on 90 degrees of freedom
## Multiple R-squared:  0.4781, Adjusted R-squared:  0.3679
## F-statistic: 4.339 on 19 and 90 DF,  p-value: 1.009e-06
```

```r
#NEGATIVE
negative_model <- lm(negative ~  factor_quarter + factor_party, data_dict_emo )
summary(negative_model)
```

```
##
## Call:
## lm(formula = negative ~ factor_quarter + factor_party, data = data_dict_emo)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.79357 -0.14849  0.00431  0.15790  0.46872
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)                2.560662   0.108714  23.554  < 2e-16 ***
## factor_quarter1            0.002167   0.108714   0.020  0.98414
## factor_quarter2           -0.077716   0.108714  -0.715  0.47654
## factor_quarter3           -0.077039   0.108714  -0.709  0.48038
## factor_quarter4            0.175647   0.108714   1.616  0.10966
## factor_quarter6           -0.225225   0.108714  -2.072  0.04115 *
## factor_quarter7           -0.082757   0.108714  -0.761  0.44851
## factor_quarter8           -0.012345   0.108714  -0.114  0.90984
## factor_quarter9            0.028457   0.108714   0.262  0.79410
## factor_quarter10          -0.222362   0.108714  -2.045  0.04374 *
## factor_partyCI            -0.739253   0.114020  -6.484 4.70e-09 ***
## factor_partyFDI            0.494954   0.114020   4.341 3.71e-05 ***
```

```
## factor_partyFI           -0.056139   0.114020  -0.492  0.62366
## factor_partyINDIPENDENTE -1.173282   0.114020 -10.290  < 2e-16 ***
## factor_partyIV           -0.386425   0.114020  -3.389  0.00104 **
## factor_partyLEGA         -0.002478   0.114020  -0.022  0.98271
## factor_partyLEU           0.229343   0.114020   2.011  0.04727 *
## factor_partyM5S          -0.254663   0.114020  -2.233  0.02800 *
## factor_partyMISTO        -0.195756   0.114020  -1.717  0.08944 .
## factor_partyREG_LEAGUES  -0.777217   0.114020  -6.817 1.03e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.255 on 90 degrees of freedom
## Multiple R-squared:  0.8089, Adjusted R-squared:  0.7685
## F-statistic: 20.05 on 19 and 90 DF,  p-value: < 2.2e-16
```

4.6 Regressions

```
# import the populism dataset
load("data/data_dict1.Rda")

# add the level of populism in the dataframe with the emotions
data_dict_emo$populism <- data_dict1$populism

# Change the reference category for quarter as quarter 8
data_dict_emo$factor_quarter <- relevel(data_dict_emo$factor_quarter, ref = "8")

# Negative prevalence
negative_prevalence_model <- lm(negative_prevalence ~ factor_party +
                                  factor_quarter +
```

```
                                  populism, data_dict_emo)
summary(negative_prevalence_model)
```

```
##
## Call:
## lm(formula = negative_prevalence ~ factor_party + factor_quarter +
##     populism, data = data_dict_emo)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.83425 -0.13061 -0.01836  0.15555  0.69102
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)                1.457921   0.196396   7.423 6.51e-11 ***
## factor_partyCI            -0.687517   0.130189  -5.281 9.02e-07 ***
## factor_partyFDI            0.521583   0.136636   3.817 0.000249 ***
## factor_partyFI            -0.031204   0.127490  -0.245 0.807208
## factor_partyINDIPENDENTE -0.912110   0.132441  -6.887 7.79e-10 ***
## factor_partyIV            -0.415488   0.131061  -3.170 0.002090 **
## factor_partyLEGA           0.016135   0.129531   0.125 0.901148
## factor_partyLEU            0.128497   0.127488   1.008 0.316228
## factor_partyM5S           -0.192532   0.133586  -1.441 0.153021
## factor_partyMISTO         -0.092293   0.127711  -0.723 0.471778
## factor_partyREG_LEAGUES   -0.483682   0.135906  -3.559 0.000600 ***
## factor_quarter5            0.095929   0.124208   0.772 0.441968
## factor_quarter1           -0.020075   0.121951  -0.165 0.869623
## factor_quarter2            0.002328   0.123831   0.019 0.985041
## factor_quarter3           -0.158689   0.126302  -1.256 0.212250
## factor_quarter4            0.205304   0.122020   1.683 0.095969 .
```

```
## factor_quarter6          -0.101347   0.123132  -0.823 0.412663
## factor_quarter7          -0.103082   0.122068  -0.844 0.400675
## factor_quarter9          -0.040199   0.123641  -0.325 0.745849
## factor_quarter10         -0.250742   0.122015  -2.055 0.042810 *
## populism                  0.582670   0.253212   2.301 0.023721 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.285 on 89 degrees of freedom
## Multiple R-squared:  0.7629, Adjusted R-squared:  0.7096
## F-statistic: 14.32 on 20 and 89 DF,  p-value: < 2.2e-16
```

```r
# Negative emotion
negative_model <- lm(negative ~ factor_party +
                       factor_quarter +
                       populism, data_dict_emo)
summary(negative_model)
```

```
##
## Call:
## lm(formula = negative ~ factor_party + factor_quarter + populism,
##     data = data_dict_emo)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.82801 -0.13125  0.00941  0.12134  0.50310
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)               2.248269   0.171994  13.072  < 2e-16 ***
## factor_partyCI           -0.687535   0.114013  -6.030 3.65e-08 ***
```

```
## factor_partyFDI           0.399141   0.119659   3.336  0.00124 **
## factor_partyFI           -0.062815   0.111649  -0.563  0.57511
## factor_partyINDIPENDENTE -1.103196   0.115985  -9.511 3.28e-15 ***
## factor_partyIV           -0.326952   0.114777  -2.849  0.00545 **
## factor_partyLEGA         -0.047517   0.113437  -0.419  0.67631
## factor_partyLEU           0.235937   0.111648   2.113  0.03738 *
## factor_partyM5S          -0.332532   0.116988  -2.842  0.00555 **
## factor_partyMISTO        -0.211835   0.111843  -1.894  0.06147 .
## factor_partyREG_LEAGUES  -0.685412   0.119020  -5.759 1.19e-07 ***
## factor_quarter5           0.062394   0.108775   0.574  0.56768
## factor_quarter1          -0.005587   0.106799  -0.052  0.95840
## factor_quarter2          -0.018994   0.108445  -0.175  0.86136
## factor_quarter3          -0.131691   0.110609  -1.191  0.23698
## factor_quarter4           0.209609   0.106859   1.962  0.05294 .
## factor_quarter6          -0.174171   0.107833  -1.615  0.10981
## factor_quarter7          -0.093044   0.106902  -0.870  0.38644
## factor_quarter9          -0.003622   0.108279  -0.033  0.97339
## factor_quarter10         -0.188505   0.106855  -1.764  0.08114 .
## populism                  0.492414   0.221751   2.221  0.02892 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2496 on 89 degrees of freedom
## Multiple R-squared:  0.8189, Adjusted R-squared:  0.7782
## F-statistic: 20.12 on 20 and 89 DF,  p-value: < 2.2e-16
```

```
# Anxiety emotion
anxiety_model <- lm(anxiety ~ factor_party +
                      factor_quarter +
                      populism, data_dict_emo)
```

```r
summary(anxiety_model)
```

```
##
## Call:
## lm(formula = anxiety ~ factor_party + factor_quarter + populism,
##     data = data_dict_emo)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.203185 -0.030062 -0.006422  0.031150  0.241173
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                0.2688373  0.0478034   5.624 2.13e-07 ***
## factor_partyCI            -0.0792116  0.0316883  -2.500   0.0143 *
## factor_partyFDI            0.0378298  0.0332575   1.137   0.2584
## factor_partyFI            -0.0006212  0.0310313  -0.020   0.9841
## factor_partyINDIPENDENTE -0.2119155  0.0322366  -6.574 3.24e-09 ***
## factor_partyIV            -0.0357351  0.0319007  -1.120   0.2656
## factor_partyLEGA          -0.0010955  0.0315281  -0.035   0.9724
## factor_partyLEU            0.0681484  0.0310310   2.196   0.0307 *
## factor_partyM5S           -0.0338173  0.0325152  -1.040   0.3011
## factor_partyMISTO         -0.0182670  0.0310853  -0.588   0.5583
## factor_partyREG_LEAGUES  -0.0526060  0.0330799  -1.590   0.1153
## factor_quarter5            0.0190702  0.0302326   0.631   0.5298
## factor_quarter1            0.0626135  0.0296833   2.109   0.0377 *
## factor_quarter2            0.0148207  0.0301407   0.492   0.6241
## factor_quarter3            0.0104310  0.0307423   0.339   0.7352
## factor_quarter4            0.0509013  0.0297000   1.714   0.0900 .
## factor_quarter6           -0.0225554  0.0299707  -0.753   0.4537
```

```
## factor_quarter7           0.0430576  0.0297118   1.449   0.1508
## factor_quarter9          -0.0079431  0.0300946  -0.264   0.7924
## factor_quarter10         -0.0095388  0.0296988  -0.321   0.7488
## populism                 -0.0131192  0.0616326  -0.213   0.8319
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.06936 on 89 degrees of freedom
## Multiple R-squared:  0.5817, Adjusted R-squared:  0.4877
## F-statistic: 6.188 on 20 and 89 DF,  p-value: 6.176e-10
```

```r
# Anger emotion
anger_model <- lm(anger ~ factor_party +
                    factor_quarter +
                    populism, data_dict_emo)
summary(anger_model)
```

```
##
## Call:
## lm(formula = anger ~ factor_party + factor_quarter + populism,
##     data = data_dict_emo)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.32401 -0.07952  0.00037  0.06871  0.48334
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 0.88129    0.09360   9.415 5.19e-15 ***
## factor_partyCI             -0.40239    0.06205  -6.485 4.83e-09 ***
## factor_partyFDI             0.20315    0.06512   3.120  0.00244 **
```

```
## factor_partyFI           -0.08894    0.06076  -1.464  0.14678
## factor_partyINDIPENDENTE -0.51858    0.06312  -8.215 1.57e-12 ***
## factor_partyIV           -0.07514    0.06246  -1.203  0.23221
## factor_partyLEGA         -0.05692    0.06174  -0.922  0.35900
## factor_partyLEU           0.17934    0.06076   2.951  0.00404 **
## factor_partyM5S          -0.12072    0.06367  -1.896  0.06120 .
## factor_partyMISTO        -0.10324    0.06087  -1.696  0.09337 .
## factor_partyREG_LEAGUES  -0.38502    0.06477  -5.944 5.33e-08 ***
## factor_quarter5          -0.10977    0.05920  -1.854  0.06701 .
## factor_quarter1          -0.11785    0.05812  -2.028  0.04559 *
## factor_quarter2          -0.19139    0.05902  -3.243  0.00167 **
## factor_quarter3          -0.15128    0.06020  -2.513  0.01377 *
## factor_quarter4          -0.04364    0.05816  -0.750  0.45502
## factor_quarter6          -0.14951    0.05869  -2.548  0.01256 *
## factor_quarter7          -0.09150    0.05818  -1.573  0.11934
## factor_quarter9          -0.01639    0.05893  -0.278  0.78149
## factor_quarter10         -0.19516    0.05815  -3.356  0.00116 **
## populism                  0.19253    0.12068   1.595  0.11418
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1358 on 89 degrees of freedom
## Multiple R-squared:  0.8022, Adjusted R-squared:  0.7577
## F-statistic: 18.04 on 20 and 89 DF,  p-value: < 2.2e-16
```

```
# sadness emotion
sadness_model <- lm(sadness ~ factor_party +
                      factor_quarter +
                      populism, data_dict_emo)
summary(sadness_model)
```

```
##
## Call:
## lm(formula = sadness ~ factor_party + factor_quarter + populism,
##     data = data_dict_emo)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.36628 -0.04760  0.00219  0.04560  0.36965
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)               0.51962    0.08025   6.475 5.06e-09 ***
## factor_partyCI           -0.10570    0.05320  -1.987 0.049995 *
## factor_partyFDI           0.01902    0.05583   0.341 0.734222
## factor_partyFI            0.02387    0.05209   0.458 0.647930
## factor_partyINDIPENDENTE -0.21123    0.05412  -3.903 0.000184 ***
## factor_partyIV           -0.09736    0.05355  -1.818 0.072438 .
## factor_partyLEGA         -0.07186    0.05293  -1.358 0.178028
## factor_partyLEU          -0.04913    0.05209  -0.943 0.348193
## factor_partyM5S          -0.06025    0.05459  -1.104 0.272693
## factor_partyMISTO        -0.06847    0.05219  -1.312 0.192868
## factor_partyREG_LEAGUES  -0.11049    0.05553  -1.990 0.049710 *
## factor_quarter5           0.09126    0.05075   1.798 0.075556 .
## factor_quarter1           0.04824    0.04983   0.968 0.335682
## factor_quarter2           0.15611    0.05060   3.085 0.002710 **
## factor_quarter3           0.08862    0.05161   1.717 0.089436 .
## factor_quarter4           0.11495    0.04986   2.306 0.023463 *
## factor_quarter6           0.04591    0.05031   0.912 0.363979
## factor_quarter7           0.02701    0.04988   0.542 0.589448
## factor_quarter9           0.06648    0.05052   1.316 0.191568
```

```
## factor_quarter10          0.02911    0.04986   0.584 0.560799
## populism                  0.08471    0.10347   0.819 0.415138
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1164 on 89 degrees of freedom
## Multiple R-squared:  0.3902, Adjusted R-squared:  0.2532
## F-statistic: 2.847 on 20 and 89 DF,  p-value: 0.0003978
```

5 STM Topic model analysis

5.1 Preliminary steps

5.1.1 Load the data

```r
load("data/dfm.Rda")
load("data/dataset.Rda")
load("data/tw.Rda")
load("data/corpus.Rda")
```

5.1.2 Import the dictionaries

```r
# Import dictionaries file
dict <-  read_excel("data/populism_dictionaries.xlsx")

Decadri_Boussalis_Grundl <-
  dictionary(list(people =
                    dict$Decadri_Boussalis_Grundl_People
                  [!is.na(dict$Decadri_Boussalis_Grundl_People)],
                  common_will =
                    dict$`Decadri_Boussalis_Grundl_Common Will`
                  [!is.na(dict$`Decadri_Boussalis_Grundl_Common Will`)],
                  elite =
                    dict$Decadri_Boussalis_Grundl_Elite
                  [!is.na(dict$Decadri_Boussalis_Grundl_Elite)]))
```

5.1.3 Remove all the account's mentions

```
DFM@Dimnames$features <- gsub("^@", "", DFM@Dimnames$features)
```

5.1.4 Trim the data

```
# Remove text with less than 1 word
DFM <- dfm_subset(DFM, ntoken(DFM) > 1)

# Remove very short words
DFM <- dfm_remove(DFM, min_nchar=2)

# Dfm trimming:only words that occur in the top 20% of the distribution
# and in less than 10% of documents
DFM <- dfm_trim(DFM,min_termfreq = 0.8,
                termfreq_type = "quantile",
                max_docfreq = 0.1, docfreq_type = "prop")
```

5.1.5 Apply dictionary

```
# Apply Dictionary
DFMdict <- dfm_lookup(DFM, dictionary = Decadri_Boussalis_Grundl)

# Convert to a dataframe
DATAdictDFM <- DFMdict %>%
  quanteda::convert(to = "data.frame")
```

5.1.6 Create percentage for each components

```r
# RUN ONLY ONCE!
# Add variable with general level of populism & multiply all components by 100
DATAdictDFM <- DATAdictDFM %>%
  mutate(populism = (people + common_will + elite) * 100)

DATAdictDFM <- DATAdictDFM %>%
  mutate(people = people*100,
         common_will = common_will*100,
         elite = elite*100)
```

5.1.7 Add the percentage of populism to the original dfm

```r
docvars(DFM) <- cbind(docvars(DFM),DATAdictDFM)
```

5.1.8 Convert DFM to STM format

```r
myDFM = DFM
set.seed(123)
DfmStm <- quanteda::convert(myDFM, to = "stm", docvars = docvars(myDFM))
#save(DfmStm, file="data/DfmStm.Rda")
```

5.1.9 Import the original corpus and repeat the same cleanings

This is for search the documents after find a label for the topics

```
# Remove text with less than 1 word
corpus <- corpus_subset(corpus, ntoken(corpus) > 1)

subs_corpus <- corpus_subset(corpus,
                             docnames(corpus) %in% myDFM@Dimnames$docs)

subs_corpus <- as.character(subs_corpus)

subs_corpus <-  as.vector(subs_corpus)

#save(subs_corpus, file="data/subs_corpus.Rda")
```

5.2 Find best number of topics k

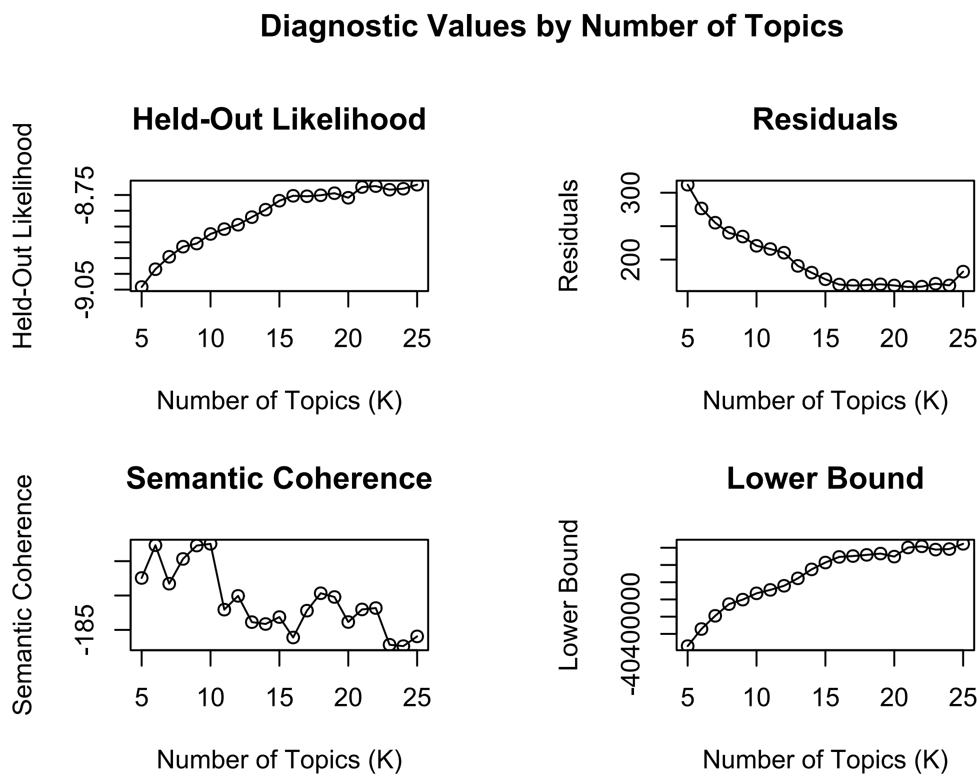
5.2.1 Search the best number of Topics comparing coherence and exclusivity values

K = 5:20

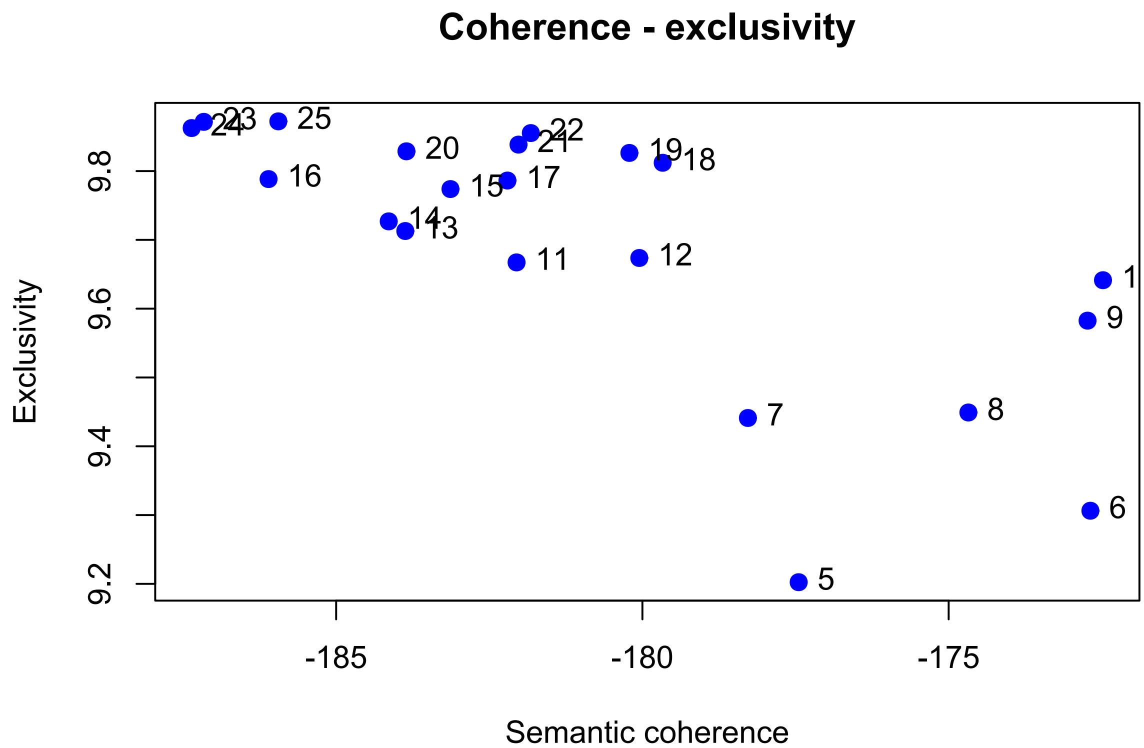
```
k <-c(5:20)
system.time(storageNoG <- searchK(DfmStm$documents,
                                 DfmStm$vocab,
                                 K = k,
                                 prevalence = ~ party_id + populism + s(quarter),
                                 data = DfmStm$meta, init.type = "Spectral"))
#save(storageNoG,file="data/storageNoG.Rda")
```

5.2.2 plot results

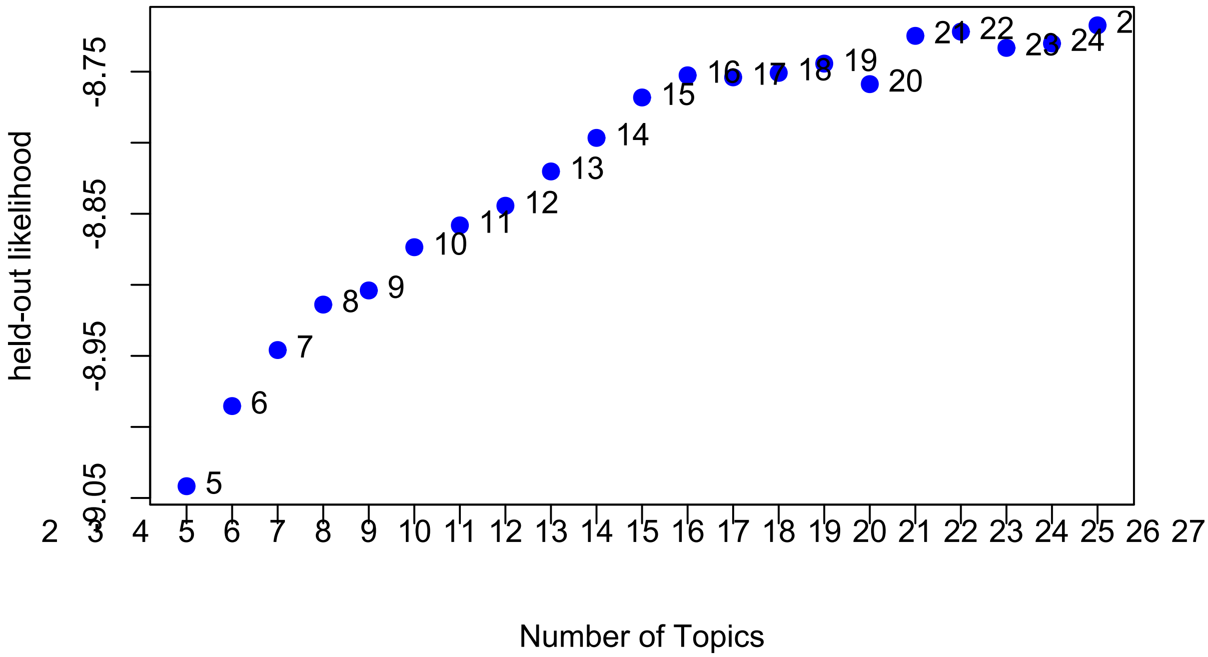
```
plot.searchK(storageNoG)
```




```
plot(storageNoG$results$semcoh, storageNoG$results$exclus,
     xlab= "Semantic coherence",
     ylab= "Exclusivity",
     col= "blue", pch = 19, cex = 1, lty = "solid", lwd = 2,
     main = "Coherence - exclusivity")
text(storageNoG$results$semcoh, storageNoG$results$exclus,
     labels=storageNoG$results$K, cex= 1, pos=4)
```




```
plot(storageNoG$results$K, storageNoG$results$heldout,
     xlab= "Number of Topics",
     ylab= "held-out likelihood",
     col= "blue", pch = 19, cex = 1, lty = "solid", lwd = 2, xaxt="n")
text(storageNoG$results$K, storageNoG$results$heldout,
     labels=storageNoG$results$K, cex= 1, pos=4)
xtick<-seq(2, 50, by=1)
axis(side=1, at=xtick, labels = FALSE)
text(x=xtick,  par("usr")[3],
     labels = xtick, pos = 1, xpd = TRUE)
```



held-out likelihood
-8.75
-8.85
-8.95
-9.05
2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27
Number of Topics

5.3 Run the analysis selecting k = 10

```
k = 10

mySTM10NoG <- stm(DfmStm$documents, vocab = DfmStm$vocab,
             K = k,
             prevalence = ~ party_id + populism + s(quarter),
             data = DfmStm$meta,
             init.type = "Spectral",
             verbose = TRUE)

 #save(mySTM10NoG,file="data/mySTM10NoG.Rda")
```

5.3.1 Label topics

The frequency/exclusivity (FREX) scoring summarizes words according to their probability of appearance under a topic and the exclusivity to that topic. These words provide more semantically intuitive representations of each topic.

```
labeledtpic <- labelTopics(mySTM10NoG, n=10)

#FREX
FREXmySTM10NoG <- t(as.matrix(labeledtpic[["frex"]]))
kable(FREXmySTM10NoG[,1:5], col.names = c("Topic1","Topic2","Topic3",
                                          "Topic4","Topic5"))
```

Topic1	Topic2	Topic3	Topic4	Topic5
regione	pass	legasalvini	libertà	domani
covid	ragazzi	vittime	diritti	città
vaccini	green	famiglia	democrazia	buona
personale	viva	pensiero	violenza	auguri
virus	#greenpass	foto	minacce	sindaco
medici	vittoria	ricordo	inaccettabile	mattina
sanitario	sport	memoria	costituzione	aspetto
numero	vince	abbraccio	#ddlzan	milano
lombardia	finale	pagina	umani	vediamo
coronavirus	vinto	legacamera	esteri	stasera

```
kable(FREXmySTM10NoG[,6:10], col.names = c("Topic6","Topic7","Topic8",
                                            "Topic9","Topic10"))
```

Topic6	Topic7	Topic8	Topic9	Topic10
fratelliditalia	presidente	conte	italiaviva	imprese
stampasgarbi	forza_italia	salvini	davvero	euro
giorgiameloni	pdnetwork	pd	giusto	lavoratori
#fratelliditalia	giuseppeconteit	#salvini	cambiare	decreto
meloni	deputatipd	#conte	matteorenzi	misure
vocedelpatriota	berlusconi	m5s	guardare	piano
fratelli	#mattarella	vergogna	maestro	risorse
#meloni	enricoletta	#iostoconsalvini	l'intervista	miliardi
ilgiornale	mattarella	vogliono	parità	bilancio
adnkronos	gruppoficamera	referendum	elenabonetti	servono

Looking at the FREX words, it is complicated to give a substantive interpretation of the content of the topics. We therefore made a second attempt using k = 18

5.4 Run the analysis selecting k = 18

```
k = 18

mySTM18NoG <- stm(DfmStm$documents, vocab = DfmStm$vocab,
             K = k,
             prevalence = ~ party_id + populism + s(quarter),
             data = DfmStm$meta,
             init.type = "Spectral",
             verbose = TRUE)

# save(mySTM18NoG,file="data/mySTM18NoG.Rda")
```

5.4.1 Label topics

The frequency/exclusivity (FREX) scoring summarizes words according to their probability of appearance under a topic and the exclusivity to that topic. These words provide more semantically intuitive representations of each topic.

```
labeledtpic <- labelTopics(mySTM18NoG, n=30)

#FREX
FREXmySTM18NoG <- t(as.matrix(labeledtpic[["frex"]]))
kable(FREXmySTM18NoG[,1:5], col.names = c("Topic1","Topic2","Topic3",
                                          "Topic4","Topic5")) %>%
  kable_styling(latex_options="scale_down")
```

```
kable(FREXmySTM18NoG[,6:10], col.names = c("Topic6","Topic7","Topic8",
                                           "Topic9","Topic10")) %>%
  kable_styling(latex_options="scale_down")
```

Topic1	Topic2	Topic3	Topic4	Topic5
great	dosi	pubblicata	maestro	anzaldi
muto	dose	foto	mancherà	lastampa
will	molecolari	#venetodaamare	addio	intervista
red_marxist	tamponi	vicenza	mancherai	adnkronos
government	vaccinale	italiachiama	viaggio	pierosansonetti
hope	terapie	#abruzzo	ciao	libero_official
together	vaccinati	laguna	#enniomorricone	adginforma
democracy	vaccinazione	pubblicato	musica	edicola
people	registrati	palazzo	sergio	ilmessaggeroit
good	gialla	chigi	artista	corriere
always	prossima	d'alfonso	#gigiproietti	lapresse_news
us	vaccino	#italianinelmondo	natale	radioradicale
even	intensive	treviso	#davidsassoli	repubblica
right	intensiva	ddl	#battiato	avvenire_nei
can	terapia	all'unanimità	curini	corriere
must	all'aperto	senato	#carlafracci	fuortes
today	decessi	dolomiti	icona	askanews_ita
really	#vaccino	ricevo	amato	ilprimaton
one	ricoveri	museo	onomastico	agenzia_dire
now	test	villa	poeta	agenzia_italia
rights	somministrazioni	canova	#monicavitti	ilriformista
years	rapidi	ig	#festadellarepubblica	messaggero
colinphoenix	settimana	neve	carla	il_piccolo
get	zona	belluno	lucianoghelfi	l'intervista
friend	prenotazioni	zan	simonamalpezzi	cda
newwaveandpunk	contagi	falsa	martina_carone	italpress
welikeduel	arancione	camera	sassoli	formichenews
make	guariti	mostra	appassionato	scanzi
work	ricoverati	deputati	#epifani	mattinodinapoli
ever	l'obbligo	pdabruzzo	eletta	ansaromalazio

Topic6	Topic7	Topic8	Topic9	Topic10
rapite	volontari	risposte	pour	strage
dall'oglio	vigili	economica	iva	scorta
#padredalloglio	infermieri	scelte	bollette	#foibe
paoladelusa	#forzearmate	responsabilità	avec	uccisi
tornino	poliziadistato	decisioni	fiscali	attanasio
sostenibilità	armate	sanitaria	autonomi	ucciso
ecologica	gdf	gestire	du	#giornodelricordo
transizione	svolto	concrete	cartelle	tragedia
ambientale	emergenzavvf	sociali	au	#giornatadellamemoria
sostenibile	ringraziare	tavolo	esattoriali	innocenti
climatici	penitenziaria	opposizioni	et	uccise
pianeta	ringraziamento	#lockdown	perduto	iacovacci
innovazione	dell'ordine	ripresa	dans	attentato
dell'ambiente	#vigilidelfuoco	affrontare	cassa	ferita
cambiamenti	divisa	#fase2	fiscale	l'orrore
angelazoppo	sm_difesa	l'emergenza	prestiti	vittime
loops40994697	_carabinieri_	necessarie	fatturato	#aldomoro
alatigiulio	italiannavy	ripartire	contributi	foibe
scureggione	ministerodifesa	collaborare	scadenze	#congo
l'ambiente	mamme	chiare	credito	#giornodellamemoria
gfi65	soccorso	giuste	pagamenti	ucraino
stretto	svolgono	soluzioni	tax	#falcone
massionline	compleanno	esecutivo	mutui	odio
renzo_pisu	#festadellamamma	uscire	tasse	#paoloborsellino
aledeniz	plauso	#smartworking	nous	violenza
gianni_dragoni	prezioso	servono	liquidità	uccisa
task	mille	messe	bonus	l'odio
sviluppo	#carabinieri	serve	sur	falcone
climatico	esercito	fronteggiare	prezzi	vittima
ugoarrigo	nonni	superare	#bollette	persero

Topic11	Topic12	Topic13	Topic14	Topic15
l'aggiornamento		diretta	#sanità	para
bollettino	5s	seguitemi	#quirinale	por
docenti	imbarazzante	aspetto	regione_sicilia	el
pittoni	grillini	ospite	#parlamento	giusvapulejo
scuola	franferrante	aggiornamenti	#quirinale2022	los
orizzontescuola	imbecille	streaming	#presidentedellarepubblica	las
ordinario	vabbè	interverrò	#governomusumeci	#amala
studenti	brutta	rete4	#verità	paolabottelli
orizzonte	capito	domattina	#rassegnastampa	tommasolabate
didattica	ridicolo	parleremo	#buongiorno	dispiace
#scuola	a_lisacorrado	seguiteci	#libero	italianos
scolastica	dica	rassegna	#salute	esta
scolastico	robdellaseta	seguite	#primapagina	hoy
concorsi	poltrone	facebook	#edicola	mariolavia
paritarie	l'unico	collegatevi	#giornali	nomfup
insegnanti	letta	sull'emergenza	#tempo	juve
asili	neanche	canale	#giornale	cottarellicpi
scolastici	legalemeglio	parteciperò	#consultazioni	exterior
azzolina	teatrino	mancate	#edicolalucidi	gobierno
campania	zingaretti	fb	coraggio_italia	adalucde
alunni	virologo	vediamo	#fiducia	gianlookingfor
azzolinalucia	povero	organizzato	#lavoro	appunto
scuole	talmente	seguire	eleggere	gracias
nido	leu	link	#maggioranza	#inter
precari	smentisce	l'audizione	#leu	ardigiorgio
classe	#arcuri	#danielasantanche	#europa	todos
#azzolina	giasilvestrini	raitre	#covid19italia	mancava
#dad	doveva	parlerò	#infrastrutture	lauracesaretti1
dell'istruzione	pur	conoscitiva	terni	grotondi
dad	monicafrassoni	stasera	#emergenza	il_cappellini

```
kable(FREXmySTM18NoG[,11:15], col.names = c("Topic11","Topic12","Topic13",
                                             "Topic14","Topic15")) %>%
  kable_styling(latex_options="scale_down")
```

```
kable(FREXmySTM18NoG[,16:18], col.names = c("Topic16","Topic17","Topic18"))
```

Topic16	Topic17	Topic18
#tokyo2020	#iostoconsalvini	youtube
#italiateam	molinaririk	speriamo
medaglie	maxromeomb	arrivato
tokyo	angelociocca	all'estero
atleti	#processateancheme	finito
olimpiadi	#salvinipremier	confermato
azzurri	patriziarametta	ricorso
alex	votalega	fatti
medaglia	lega_senato	deciso
argento	sbarchi	tratta
oro	a_gusmeroli	dato
bronzo	noiconsalvini	diversi
podio	ponytaele	germania
#borgonzonipresidente	lacavandoli	rapporti
#olimpiadi	lntoscana	riconosce
#euro2020	legacamera	bruxelles
#paralimpiadi	clandestina	penale
#paralympics	massimobitonci	l'ora
#jacobs	lucabattanta	sostengo
#berrettini	clandestini	lanciato
#tamberi	legasalvini	corruzione
vince	matteosalvinimi	sospeso
federica	#blocconavale	l'abbiamo
forza	#primagliitaliani	ilpost
azzurro	massimogara	tocca
#giochiolimpici	alex63roy	strada
fi_ultimissime	giuliocentemero	mediazione
paralimpiadi	lampedusa	passi
#pechino2022	albertobagnai	vedono
l'oro	alessia_smile6	appello

```r
# PROB
PROBmySTM18NoG <- t(as.matrix(labeledtpic[["prob"]]))
kable(PROBmySTM18NoG[,1:5], col.names = c("Topic1","Topic2","Topic3",
                                           "Topic4","Topic5")) %>%
  kable_styling(latex_options="scale_down")
```

```r
kable(PROBmySTM18NoG[,6:10], col.names = c("Topic6","Topic7","Topic8",
                                            "Topic9","Topic10")) %>%
  kable_styling(latex_options="scale_down")
```

```r
kable(PROBmySTM18NoG[,11:15], col.names = c("Topic11","Topic12","Topic13",
                                             "Topic14","Topic15")) %>%
  kable_styling(latex_options="scale_down")
```

```r
kable(PROBmySTM18NoG[,16:18], col.names = c("Topic16","Topic17","Topic18"))
```

Topic1	Topic2	Topic3	Topic4	Topic5
mov5stelle	covid	legge	presidente	via
leggi	dati	commissione	grande	stampasgarbi
parlamentari	vaccini	senato	anni	intervista
giorgiameloni	vaccino	camera	buona	rai
giuseppeconteit	#covid19	appena	pdnetwork	parlato
scritto	settimana	foto	mondo	repubblica
post	virus	aula	storia	vocedelpatriota
fratelliditalia	vaccinale	video	buon	fattoquotidiano
m5s_senato	#covid	palazzo	repubblica	pubblico
social	regioni	voto	italiana	politica
italymfa	#greenpass	lavori	donna	matteorenzi
amp	campagna	proposta	famiglia	sud
cina	numero	città	città	leggere
tweet	regione	deputati	cultura	corriere
usa	vaccinati	gruppo	uomo	draghi
l'ho	lombardia	approvato	festa	ilgiornale
#m5s	contagi	pubblicata	deputatipd	recovery
referendum	tamponi	testo	comunità	anzaldi
luigidimaio	zona	#ddlzan	politica	adnkronos
fdi_parlamento	casi	sede	enricoletta	direttore
corte	italia	venezia	mattarella	libero_official
costituzionale	mascherine	milano	italiano	lettera
italy	settimane	#venezia	tanti	quotidiano
news	positivi	ddl	auguri	giornalisti
letto	#vaccini	veneto	giornata	parlo
#referendum	dosi	firma	natale	lastampa
scrivere	test	centro	paese	riflessione
fake	#liguria	giunta	amico	nord
trump	primi	roma	persona	sole
#iovotono	numeri	approvata	simbolo	articolo

Topic6	Topic7	Topic8	Topic9	Topic10
presto	grazie	governo	imprese	solidarietà
sviluppo	lavoro	paese	milioni	libertà
territorio	donne	cittadini	euro	anni
settore	buon	crisi	decreto	parole
progetto	auguri	serve	miliardi	diritti
qualità	forze	l'italia	famiglie	vittime
ricerca	impegno	momento	lavoratori	democrazia
mondo	uomini	dare	fondo	violenza
tutela	servizio	responsabilità	tasse	guerra
importante	giornata	#coronavirus	riforma	giustizia
nuove	linea	sicurezza	bilancio	popolo
opportunità	medici	pandemia	fiscale	memoria
crescita	personale	salute	governo	vicinanza
investimenti	generale	misure	aziende	morte
comuni	sicurezza	bisogna	sostegno	minacce
futuro	cuore	italiani	misura	pensiero
transizione	civile	piano	emendamento	verità
turismo	tanti	politiche	stop	pace
infrastrutture	ringrazio	parlamento	soldi	dovere
sostenibile	dell'ordine	emergenza	dl	rispetto
progetti	anni	lavorare	aiuti	diritto
territori	operatori	dobbiamo	difficoltà	donne
italiana	protezione	ripartire	risorse	ricordo
modello	polizia	economica	aiutare	inaccettabile
centro	compleanno	bene	de	tragedia
sociale	paese	servono	approvato	dolore
digitale	nazionale	priorità	pagare	ricordare
produzione	sanitario	mettere	bonus	atto
città	difesa	devono	reddito	valori
fondamentale	l'impegno	lavoro	mld	mafia

Topic11	Topic12	Topic13	Topic14	Topic15
scuola	conte	diretta	paese	de
giovani	pd	domani	futuro	vero
scuole	governo	consiglio	#governo	ragione
ragazzi	vuole	punto	#draghi	credo
bambini	m5s	sera	sanità	vedere
settembre	sinistra	parlare	vogliamo	giusto
regione	pass	mattina	#italia	bene
figli	maggioranza	aspetto	#senato	penso
studenti	partito	stampa	pubblica	porta
#scuola	andare	amici	politica	carlocalenda
presenza	green	intervento	presidente	differenza
ministra	cose	vediamo	bisogno	bravo
#covid19	draghi	conferenza	#m5s	magari
notizie	#conte	ospite	costruire	posso
ministero	dovrebbe	pagina	#lavoro	dico
formazione	italiani	gt	dobbiamo	serie
ministro	voto	stasera	#leu	ovviamente
campania	parlamento	vista	#mes	peggio
classe	parla	incontro	#pnrr	guidocrosetto
distanza	problema	link	l'italia	gran
famiglie	destra	confronto	riforme	assolutamente
diritto	votare	libro	sistema	en
studio	sa	ministri	#sicilia	partita
dell'unità	capire	facebook	#recoveryfund	resto
bollettino	casa	presentazione	riforma	el
l'aggiornamento	male	pomeriggio	forza_italia	vedo
genitori	cittadinanza	coronavirus	italiaviva	spero
personale	pensa	insieme	giuseppeconteit	persona
scolastico	bene	seguire	#europa	sbagliato
riapertura	davvero	presidente	visione	azione_it

Topic16	Topic17	Topic18
italia	lega	bene
forza	matteosalvinimi	politica
forza_italia	salvini	fatti
sindaco	#lega	anni
insieme	italiani	dato
roma	fratelliditalia	anno
l'italia	#salvini	ministro
battaglia	legasalvini	europa
viva	matteo	strada
notizia	#fratelliditalia	casa
grande	ministro	realtà
bella	meloni	parlamento
elettorale	d'italia	risposta
complimenti	processo	deciso
coraggio	governo	tratta
vittoria	#iostoconsalvini	italiaviva
elezioni	legacamera	possibile
sport	fratelli	pochi
#roma	vergogna	movimento
squadra	borghi_claudio	all'estero
berlusconi	#meloni	scelta
candidato	testa	tante
risultato	lega_senato	unico
bocca	migranti	speriamo
piazza	piazza	arriva
centrodestra	confini	arrivato
campagna	casa	nuova
vincere	giorgia	continua
uniti	clandestini	roma
regionali	lamorgese	rispetto

5.4.2 Meaningfull labels with the first 10 FREX word associated

```
labeledtpic <- labelTopics(mySTM18NoG, n=10)


FREXmySTM18NoG <- t(as.matrix(labeledtpic[["frex"]]))
```

6) Sustainable energy	7) Categories involved in covid emergency	8) Economic relaunch
rapite	volontari	risposte
dall'oglio	vigili	economica
#padredalloglio	infermieri	scelte
paoladelusa	#forzearmate	responsabilità
tornino	poliziadistato	decisioni
sostenibilità	armate	sanitaria
ecologica	gdf	gestire
transizione	svolto	concrete
ambientale	emergenzavvf	sociali
sostenibile	ringraziare	tavolo

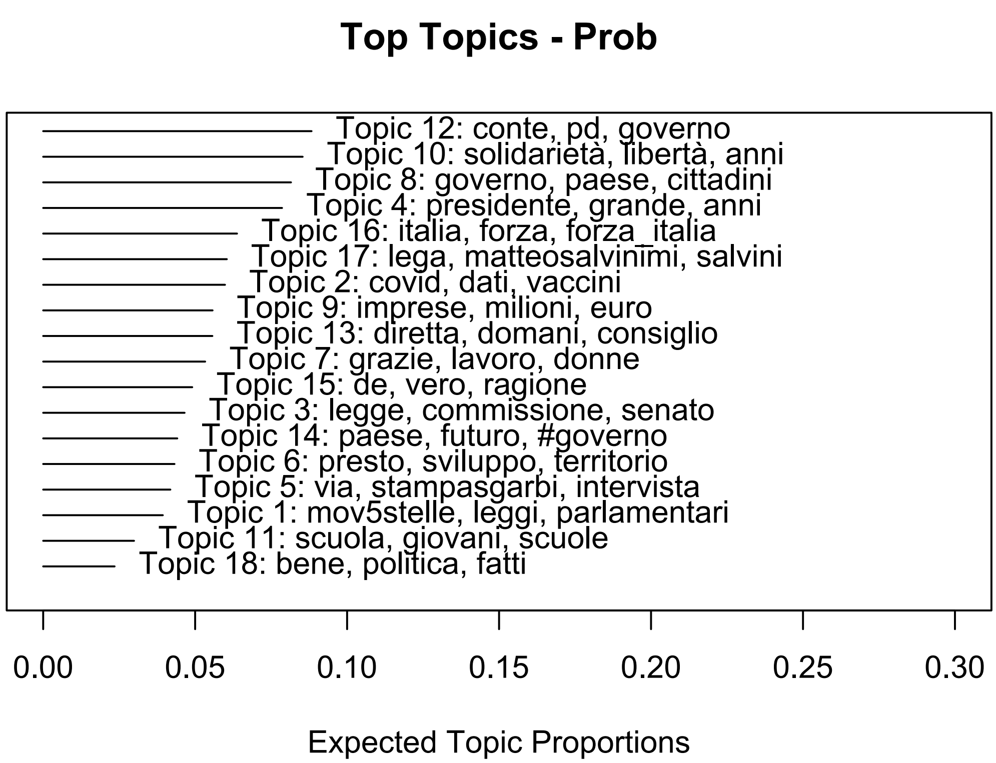
9) Economic hardship and taxes	10) Victims of violent deaths	11) Public education
pour	strage	l'aggiornamento
iva	scorta	bollettino
bollette	#foibe	docenti
avec	uccisi	pittoni
fiscali	attanasio	scuola
autonomi	ucciso	orizzontescuola
du	#giornodelricordo	ordinario
cartelle	tragedia	studenti
au	#giornatadellamemoria	orizzonte
esattoriali	innocenti	didattica

2) Covid-19	4) Epitaphs	5) Journals and media
dosi	maestro	anzaldi
dose	mancherà	lastampa
molecolari	addio	intervista
tamponi	mancherai	adnkronos
vaccinale	viaggio	pierosansonetti
terapie	ciao	libero_official
vaccinati	#enniomorricone	adginforma
vaccinazione	musica	edicola
registrati	sergio	ilmessaggeroit
gialla	artista	corriere

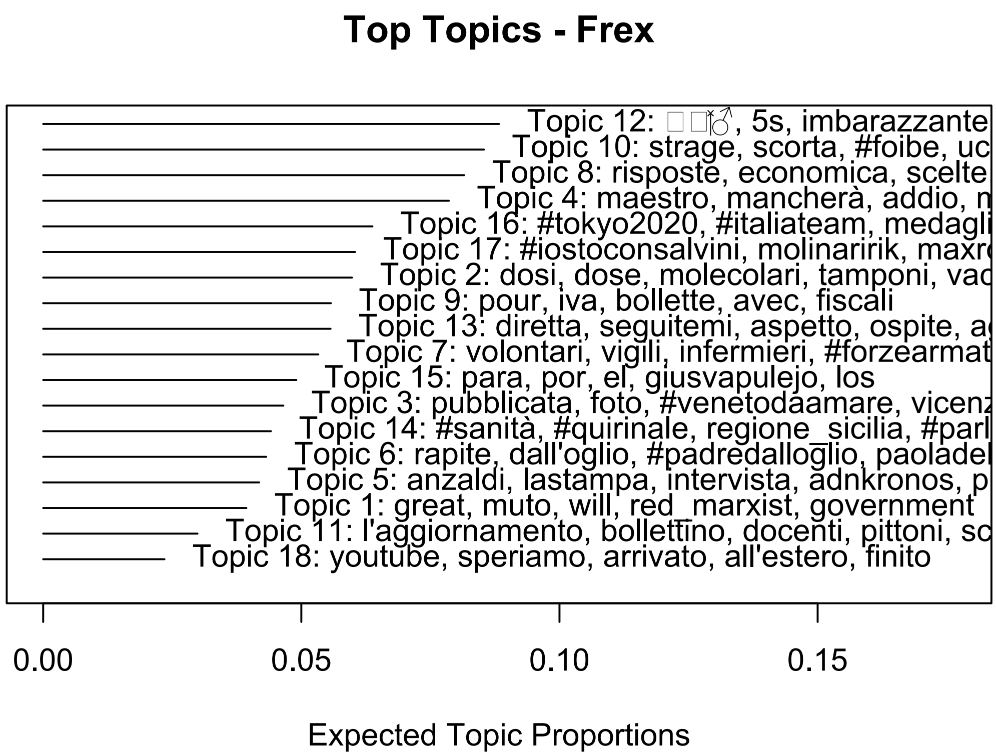
12) Anti-elitism	13) Social and TV live broadcasts	16) Olympics game	17) Right-wing party topics
	diretta	#tokyo2020	#iostoconsalvini
5s	seguitemi	#italiateam	molinaririk
imbarazzante	aspetto	medaglie	maxromeomb
grillini	ospite	tokyo	angelociocca
franferrante	aggiornamenti	atleti	#processateancheme
imbecille	streaming	olimpiadi	#salvinipremier
vabbè	interverrò	azzurri	patriziarametta
brutta	rete4	alex	votalega
capito	domattina	medaglia	lega_senato
ridicolo	parleremo	argento	sbarchi

5.4.3 Most frequent topic

```
plot(mySTM18NoG, type = "summary", xlim = c(0, .3),
     main = "Top Topics - Prob")
```



```r
# plot just frex words for each topic
plot(mySTM18NoG, type = "summary", labeltype = c("frex"), n=5,
     main = "Top Topics - Frex")
```



5.4.4 Which are the the most likely topics across our documents?

```r
tab <- table(apply(mySTM18NoG$theta,1,which.max))
topics_label <- c("01) Junk topic",
                  "02) Covid-19",
                  "03) Junk topic",
                  "04) Epitaphs",
                  "05) Journals and media",
                  "06) Sustainable energy",
```

```
                  "07) Categories involved in the covid emergency",
                  "08) Economic relaunch",
                  "09) Economic hardship and taxes",
                  "10) Victims of violent deaths",
                  "11) Public education",
                  "12) Anti-elitism",
                  "13) Social and TV live broadcasts",
                  "14) Junk topic",
                  "15) Junk topic",
                  "16) Olympics game",
                  "17) Right-wing parties topic",
                  "18) Junk topic")
tab <- as.matrix(tab)
tab2 <- cbind(topics_label,tab)
tab2 <- as.data.frame(tab2)
colnames(tab2) <- c("Topic label","Freq")
tab2$Freq <- as.numeric(tab2$Freq)
kable(tab2 %>% arrange(desc(Freq)))
```

	Topic label	Freq
12	12) Anti-elitism	41212
10	10) Victims of violent deaths	38252
8	08) Economic relaunch	31610
4	04) Epitaphs	30860
17	17) Right-wing parties topic	26510
16	16) Olympics game	23121
9	09) Economic hardship and taxes	22513
2	02) Covid-19	22233
13	13) Social and TV live broadcasts	20170
15	15) Junk topic	19391
7	07) Categories involved in the covid emergency	17417
6	06) Sustainable energy	16688
3	03) Junk topic	15645
5	05) Journals and media	14788
14	14) Junk topic	14516
1	01) Junk topic	13234
11	11) Public education	9454
18	18) Junk topic	627

5.4.5 Save them back in the original corpus

```
subs_corpus$topic <- apply(mySTM18NoG$theta,1,which.max)
```

5.4.6 Find the most associated document for each topics

This list of 18 items represent the respective document with highest theta for each topic ordered from 1 to 18.

```r
apply(mySTM18NoG$theta,2,which.max)
```

```
##  [1]  12710   1080  26346  52361  41589 198020 234701   8705  12415 248340
## [11]  80644 353132 200651 342504 267537 162724 199419  22068
```

```r
Tweet_number <- apply(mySTM18NoG$theta,2,which.max)


kable(cbind(topics_label,Tweet_number))
```

topics_label	Tweet_number
01) Junk topic	12710
02) Covid-19	1080
03) Junk topic	26346
04) Epitaphs	52361
05) Journals and media	41589
06) Sustainable energy	198020
07) Categories involved in the covid emergency	234701
08) Economic relaunch	8705
09) Economic hardship and taxes	12415
10) Victims of violent deaths	248340
11) Public education	80644
12) Anti-elitism	353132
13) Social and TV live broadcasts	200651
14) Junk topic	342504
15) Junk topic	267537
16) Olympics game	162724
17) Right-wing parties topic	199419
18) Junk topic	22068

5.5 Coefficients

```r
# set PD as reference
DfmStm$meta$party_id <- relevel(as.factor(DfmStm$meta$party_id), ref = "PD")

prep_K18NoG <- estimateEffect(1:18 ~ party_id + populism + s(quarter),
                       mySTM18NoG,metadata = DfmStm$meta,
                       uncertainty = "Global")

#save(prep_K18NoG,file="data/prep_K18NoG.Rda")
```

Regression coefficients for all topics are shown here

5.5.1 01) Junk topic

```r
summary(prep_K18NoG, topics = 1)
```

```
##
## Call:
## estimateEffect(formula = 1:18 ~ party_id + populism + s(quarter),
##     stmobj = mySTM18NoG, metadata = DfmStm$meta, uncertainty = "Global")
##
##
## Topic 1:
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)             3.977e-02  7.223e-04  55.061  < 2e-16 ***
## party_idCI             -2.106e-02  1.391e-03 -15.147  < 2e-16 ***
## party_idFDI             4.567e-03  7.736e-04   5.904 3.56e-09 ***
```

```
## party_idFI           -4.986e-03  5.682e-04  -8.776  < 2e-16 ***
## party_idINDIPENDENTE -2.388e-02  2.548e-03  -9.373  < 2e-16 ***
## party_idIV           -1.826e-02  2.153e-03  -8.483  < 2e-16 ***
## party_idLEGA         -1.195e-02  6.512e-04 -18.348  < 2e-16 ***
## party_idLEU          -1.537e-02  1.439e-03 -10.684  < 2e-16 ***
## party_idM5S           2.441e-02  7.343e-04  33.239  < 2e-16 ***
## party_idMISTO        -3.988e-03  7.314e-04  -5.452 4.98e-08 ***
## party_idREG_LEAGUES  -2.672e-02  2.941e-03  -9.088  < 2e-16 ***
## populism             -7.230e-05  6.305e-06 -11.468  < 2e-16 ***
## s(quarter)1           2.838e-02  1.330e-02   2.133 0.032910 *
## s(quarter)2          -1.526e-02  5.622e-03  -2.715 0.006623 **
## s(quarter)3           2.575e-02  2.872e-03   8.965  < 2e-16 ***
## s(quarter)4           1.151e-02  1.697e-03   6.785 1.17e-11 ***
## s(quarter)5          -1.999e-03  1.265e-03  -1.580 0.114110
## s(quarter)6           4.490e-03  1.172e-03   3.831 0.000128 ***
## s(quarter)7           5.908e-03  1.834e-03   3.221 0.001279 **
## s(quarter)8          -6.215e-03  1.769e-03  -3.512 0.000444 ***
## s(quarter)9          -7.725e-04  1.587e-03  -0.487 0.626490
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

5.5.2 02) Covid-19

```r
summary(prep_K18NoG, topics = 2)
```

```
##
## Call:
## estimateEffect(formula = 1:18 ~ party_id + populism + s(quarter),
##     stmobj = mySTM18NoG, metadata = DfmStm$meta, uncertainty = "Global")
##
##
## Topic 2:
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)            4.149e-02  8.804e-04  47.128  < 2e-16 ***
## party_idCI             1.805e-03  1.800e-03   1.003 0.316092
## party_idFDI           -6.548e-03  9.091e-04  -7.203 5.92e-13 ***
## party_idFI             1.045e-02  7.290e-04  14.330  < 2e-16 ***
## party_idINDIPENDENTE  4.496e-02  3.142e-03  14.307  < 2e-16 ***
## party_idIV            -1.083e-02  2.378e-03  -4.553 5.29e-06 ***
## party_idLEGA           1.983e-02  7.318e-04  27.104  < 2e-16 ***
## party_idLEU           -1.509e-02  1.511e-03  -9.987  < 2e-16 ***
## party_idM5S            4.103e-04  7.625e-04   0.538 0.590487
## party_idMISTO          1.978e-02  1.046e-03  18.907  < 2e-16 ***
## party_idREG_LEAGUES    2.341e-02  3.993e-03   5.862 4.57e-09 ***
## populism               7.144e-05  7.879e-06   9.067  < 2e-16 ***
## s(quarter)1            1.822e-01  1.280e-02  14.231  < 2e-16 ***
## s(quarter)2           -7.048e-02  5.691e-03 -12.386  < 2e-16 ***
## s(quarter)3            2.315e-02  3.225e-03   7.177 7.12e-13 ***
## s(quarter)4            7.312e-03  2.157e-03   3.391 0.000698 ***
```

```
## s(quarter)5          3.164e-02  1.508e-03  20.986  < 2e-16 ***
## s(quarter)6          1.346e-02  1.401e-03   9.613  < 2e-16 ***
## s(quarter)7          1.727e-02  1.855e-03   9.309  < 2e-16 ***
## s(quarter)8         -7.617e-04  2.161e-03  -0.353 0.724447
## s(quarter)9         -1.260e-02  1.737e-03  -7.253 4.09e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

5.5.3 03) Junk topic

```r
summary(prep_K18NoG, topics = 3)
```

```
##
## Call:
## estimateEffect(formula = 1:18 ~ party_id + populism + s(quarter),
##     stmobj = mySTM18NoG, metadata = DfmStm$meta, uncertainty = "Global")
##
##
## Topic 3:
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)             3.148e-02  7.290e-04  43.189  < 2e-16 ***
## party_idCI              7.255e-02  2.136e-03  33.967  < 2e-16 ***
## party_idFDI            -4.527e-03  8.588e-04  -5.271 1.35e-07 ***
## party_idFI             -2.892e-03  6.208e-04  -4.659 3.18e-06 ***
## party_idINDIPENDENTE -4.268e-03  2.594e-03  -1.645   0.0999 .
## party_idIV              1.578e-03  2.222e-03   0.710   0.4775
## party_idLEGA            2.023e-02  6.210e-04  32.571  < 2e-16 ***
## party_idLEU            -7.649e-03  1.422e-03  -5.379 7.51e-08 ***
## party_idM5S             6.548e-03  7.186e-04   9.113  < 2e-16 ***
## party_idMISTO           1.273e-02  8.906e-04  14.291  < 2e-16 ***
## party_idREG_LEAGUES     6.262e-03  3.591e-03   1.744   0.0812 .
## populism               -7.607e-05  6.703e-06 -11.348  < 2e-16 ***
## s(quarter)1             1.317e-02  1.193e-02   1.104   0.2697
## s(quarter)2            -1.345e-02  5.119e-03  -2.628   0.0086 **
## s(quarter)3             3.370e-02  2.795e-03  12.060  < 2e-16 ***
## s(quarter)4            -9.409e-03  1.669e-03  -5.639 1.71e-08 ***
```

```
## s(quarter)5          2.017e-02  1.270e-03  15.874  < 2e-16 ***
## s(quarter)6          1.472e-02  1.323e-03  11.123  < 2e-16 ***
## s(quarter)7          3.392e-02  1.609e-03  21.088  < 2e-16 ***
## s(quarter)8          2.942e-03  2.027e-03   1.451   0.1468
## s(quarter)9          2.202e-02  1.851e-03  11.897  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

5.5.4 04) Epitaphs

```r
summary(prep_K18NoG, topics = 4)
```

```
##
## Call:
## estimateEffect(formula = 1:18 ~ party_id + populism + s(quarter),
##     stmobj = mySTM18NoG, metadata = DfmStm$meta, uncertainty = "Global")
##
##
## Topic 4:
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)           8.350e-02  9.054e-04  92.228  < 2e-16 ***
## party_idCI            2.128e-02  2.274e-03   9.359  < 2e-16 ***
## party_idFDI          -4.149e-02  9.517e-04 -43.596  < 2e-16 ***
## party_idFI           -2.064e-02  8.591e-04 -24.023  < 2e-16 ***
## party_idINDIPENDENTE -1.630e-02  3.248e-03  -5.018 5.22e-07 ***
## party_idIV           -7.150e-03  2.624e-03  -2.725  0.00644 **
## party_idLEGA         -4.061e-02  7.099e-04 -57.203  < 2e-16 ***
## party_idLEU          -3.341e-02  1.842e-03 -18.137  < 2e-16 ***
## party_idM5S          -3.668e-02  7.732e-04 -47.431  < 2e-16 ***
## party_idMISTO        -3.674e-02  9.484e-04 -38.743  < 2e-16 ***
## party_idREG_LEAGUES  -3.186e-02  4.092e-03  -7.787 6.87e-15 ***
## populism             -2.239e-04  7.697e-06 -29.090  < 2e-16 ***
## s(quarter)1           2.293e-03  1.434e-02   0.160  0.87296
## s(quarter)2          -3.785e-03  6.127e-03  -0.618  0.53667
## s(quarter)3           2.202e-02  3.421e-03   6.436 1.23e-10 ***
## s(quarter)4           1.380e-02  2.040e-03   6.768 1.31e-11 ***
```

```
## s(quarter)5          2.382e-02  1.656e-03  14.383  < 2e-16 ***
## s(quarter)6          1.608e-02  1.589e-03  10.124  < 2e-16 ***
## s(quarter)7          1.804e-02  2.113e-03   8.534  < 2e-16 ***
## s(quarter)8          4.119e-02  2.311e-03  17.822  < 2e-16 ***
## s(quarter)9          2.743e-02  2.240e-03  12.246  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

5.5.5 05) Journals and media

```r
summary(prep_K18NoG, topics = 5)
```

```
##
## Call:
## estimateEffect(formula = 1:18 ~ party_id + populism + s(quarter),
##     stmobj = mySTM18NoG, metadata = DfmStm$meta, uncertainty = "Global")
##
##
## Topic 5:
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)            4.226e-02  7.212e-04  58.595  < 2e-16 ***
## party_idCI            -2.197e-02  1.400e-03 -15.695  < 2e-16 ***
## party_idFDI           -5.428e-03  8.091e-04  -6.709 1.97e-11 ***
## party_idFI            -1.715e-02  6.166e-04 -27.809  < 2e-16 ***
## party_idINDIPENDENTE -3.003e-02  2.272e-03 -13.214  < 2e-16 ***
## party_idIV            -1.346e-02  2.483e-03  -5.420 5.96e-08 ***
## party_idLEGA          -2.786e-02  5.344e-04 -52.128  < 2e-16 ***
## party_idLEU           -1.237e-02  1.532e-03  -8.071 7.00e-16 ***
## party_idM5S           -4.994e-03  6.527e-04  -7.652 1.98e-14 ***
## party_idMISTO          2.930e-02  8.749e-04  33.487  < 2e-16 ***
## party_idREG_LEAGUES   -1.077e-02  3.075e-03  -3.502 0.000461 ***
## populism              -8.650e-05  6.387e-06 -13.544  < 2e-16 ***
## s(quarter)1           -1.629e-02  1.030e-02  -1.581 0.113846
## s(quarter)2            1.929e-02  4.629e-03   4.167 3.09e-05 ***
## s(quarter)3            4.983e-03  2.462e-03   2.024 0.042993 *
## s(quarter)4            1.807e-02  1.587e-03  11.384  < 2e-16 ***
```

```
## s(quarter)5           1.021e-02  1.249e-03   8.177 2.93e-16 ***
## s(quarter)6           1.101e-02  1.327e-03   8.298  < 2e-16 ***
## s(quarter)7           3.342e-02  1.655e-03  20.194  < 2e-16 ***
## s(quarter)8          -3.824e-03  2.049e-03  -1.866 0.061987 .
## s(quarter)9           8.677e-03  1.509e-03   5.751 8.89e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

5.5.6 06) Sustainable energy

```r
summary(prep_K18NoG, topics = 6)
```

```
##
## Call:
## estimateEffect(formula = 1:18 ~ party_id + populism + s(quarter),
##     stmobj = mySTM18NoG, metadata = DfmStm$meta, uncertainty = "Global")
##
##
## Topic 6:
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)            5.311e-02  8.338e-04  63.690  < 2e-16 ***
## party_idCI             1.070e-02  1.944e-03   5.504 3.72e-08 ***
## party_idFDI           -2.796e-02  8.096e-04 -34.532  < 2e-16 ***
## party_idFI            -1.048e-02  8.022e-04 -13.063  < 2e-16 ***
## party_idINDIPENDENTE  4.738e-02  3.749e-03  12.639  < 2e-16 ***
## party_idIV             7.649e-03  2.888e-03   2.648  0.00809 **
## party_idLEGA          -1.863e-02  6.418e-04 -29.033  < 2e-16 ***
## party_idLEU            3.994e-02  1.959e-03  20.384  < 2e-16 ***
## party_idM5S            1.562e-02  7.455e-04  20.959  < 2e-16 ***
## party_idMISTO         -1.037e-02  9.232e-04 -11.228  < 2e-16 ***
## party_idREG_LEAGUES    3.475e-02  5.348e-03   6.498 8.13e-11 ***
## populism              -1.442e-04  6.905e-06 -20.888  < 2e-16 ***
## s(quarter)1           -3.316e-02  1.340e-02  -2.475  0.01333 *
## s(quarter)2            1.495e-02  6.120e-03   2.443  0.01456 *
## s(quarter)3            2.273e-03  3.150e-03   0.722  0.47043
## s(quarter)4           -9.643e-03  2.050e-03  -4.703 2.56e-06 ***
```

```
## s(quarter)5           2.144e-02  1.453e-03  14.759  < 2e-16 ***
## s(quarter)6          -1.104e-03  1.524e-03  -0.724  0.46901
## s(quarter)7           1.345e-02  2.064e-03   6.519 7.10e-11 ***
## s(quarter)8          -5.428e-03  2.171e-03  -2.500  0.01243 *
## s(quarter)9           1.389e-02  1.992e-03   6.972 3.13e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

5.5.7 07) Categories involved in the covid emergency

```r
summary(prep_K18NoG, topics = 7)
```

```
##
## Call:
## estimateEffect(formula = 1:18 ~ party_id + populism + s(quarter),
##     stmobj = mySTM18NoG, metadata = DfmStm$meta, uncertainty = "Global")
##
##
## Topic 7:
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)           6.786e-02  9.121e-04  74.399  < 2e-16 ***
## party_idCI            9.539e-03  1.780e-03   5.359 8.39e-08 ***
## party_idFDI          -1.508e-02  8.157e-04 -18.489  < 2e-16 ***
## party_idFI            1.429e-03  6.528e-04   2.189 0.028619 *
## party_idINDIPENDENTE  2.077e-02  3.246e-03   6.400 1.55e-10 ***
## party_idIV            3.286e-02  2.896e-03  11.346  < 2e-16 ***
## party_idLEGA         -6.556e-03  6.184e-04 -10.602  < 2e-16 ***
## party_idLEU           4.660e-03  1.730e-03   2.694 0.007070 **
## party_idM5S           9.556e-03  8.186e-04  11.675  < 2e-16 ***
## party_idMISTO        -2.005e-02  8.266e-04 -24.261  < 2e-16 ***
## party_idREG_LEAGUES   4.025e-04  4.271e-03   0.094 0.924923
## populism             -7.374e-05  6.933e-06 -10.635  < 2e-16 ***
## s(quarter)1           4.349e-02  1.225e-02   3.549 0.000387 ***
## s(quarter)2          -2.671e-02  5.392e-03  -4.953 7.30e-07 ***
## s(quarter)3          -1.204e-02  2.897e-03  -4.158 3.21e-05 ***
## s(quarter)4          -1.431e-02  1.932e-03  -7.404 1.33e-13 ***
```

```
## s(quarter)5          -5.240e-04  1.524e-03  -0.344 0.730969
## s(quarter)6          -1.987e-02  1.348e-03 -14.733  < 2e-16 ***
## s(quarter)7          -9.509e-03  1.817e-03  -5.234 1.66e-07 ***
## s(quarter)8          -1.561e-02  2.042e-03  -7.647 2.06e-14 ***
## s(quarter)9          -3.896e-03  1.763e-03  -2.209 0.027156 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

5.5.8 08) Economic relaunch

```r
summary(prep_K18NoG, topics = 8)
```

```
##
## Call:
## estimateEffect(formula = 1:18 ~ party_id + populism + s(quarter),
##     stmobj = mySTM18NoG, metadata = DfmStm$meta, uncertainty = "Global")
##
##
## Topic 8:
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)            1.096e-01  9.686e-04 113.167  < 2e-16 ***
## party_idCI            -5.213e-03  1.877e-03  -2.778  0.00547 **
## party_idFDI            1.038e-03  9.342e-04   1.111  0.26643
## party_idFI             7.875e-03  8.602e-04   9.156  < 2e-16 ***
## party_idINDIPENDENTE -8.187e-03  3.112e-03  -2.631  0.00853 **
## party_idIV             1.568e-02  2.804e-03   5.591 2.25e-08 ***
## party_idLEGA          -2.986e-02  6.855e-04 -43.563  < 2e-16 ***
## party_idLEU            1.548e-02  1.883e-03   8.223  < 2e-16 ***
## party_idM5S           -4.267e-03  8.624e-04  -4.948 7.50e-07 ***
## party_idMISTO         -6.885e-03  8.802e-04  -7.822 5.22e-15 ***
## party_idREG_LEAGUES   -4.986e-03  4.336e-03  -1.150  0.25021
## populism               1.782e-04  7.433e-06  23.977  < 2e-16 ***
## s(quarter)1            1.763e-01  1.301e-02  13.546  < 2e-16 ***
## s(quarter)2           -6.860e-02  5.605e-03 -12.238  < 2e-16 ***
## s(quarter)3           -2.265e-02  3.285e-03  -6.893 5.46e-12 ***
## s(quarter)4           -8.239e-03  2.200e-03  -3.745  0.00018 ***
```

```
## s(quarter)5          -3.520e-02  1.653e-03 -21.289  < 2e-16 ***
## s(quarter)6          -6.113e-02  1.383e-03 -44.215  < 2e-16 ***
## s(quarter)7          -5.238e-02  1.975e-03 -26.514  < 2e-16 ***
## s(quarter)8          -3.398e-02  2.159e-03 -15.739  < 2e-16 ***
## s(quarter)9          -4.886e-02  1.863e-03 -26.235  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

5.5.9 09) Economic hardship and taxes

```
summary(prep_K18NoG, topics = 9)
```

```
##
## Call:
## estimateEffect(formula = 1:18 ~ party_id + populism + s(quarter),
##     stmobj = mySTM18NoG, metadata = DfmStm$meta, uncertainty = "Global")
##
##
## Topic 9:
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)            5.106e-02  9.161e-04  55.741  < 2e-16 ***
## party_idCI            -5.886e-03  1.658e-03  -3.549 0.000386 ***
## party_idFDI            9.396e-03  8.949e-04  10.499  < 2e-16 ***
## party_idFI             2.000e-02  7.295e-04  27.411  < 2e-16 ***
## party_idINDIPENDENTE -1.472e-02  2.837e-03  -5.189 2.11e-07 ***
## party_idIV             1.314e-02  2.843e-03   4.622 3.80e-06 ***
## party_idLEGA           1.097e-02  6.776e-04  16.182  < 2e-16 ***
## party_idLEU            1.342e-02  1.779e-03   7.542 4.62e-14 ***
## party_idM5S            4.032e-02  8.682e-04  46.437  < 2e-16 ***
## party_idMISTO          6.231e-03  9.384e-04   6.640 3.15e-11 ***
## party_idREG_LEAGUES    1.114e-02  4.043e-03   2.756 0.005853 **
## populism               3.895e-05  8.102e-06   4.808 1.53e-06 ***
## s(quarter)1            2.917e-01  1.397e-02  20.880  < 2e-16 ***
## s(quarter)2           -9.134e-02  6.136e-03 -14.886  < 2e-16 ***
## s(quarter)3            4.468e-02  3.228e-03  13.842  < 2e-16 ***
## s(quarter)4           -3.345e-02  1.895e-03 -17.650  < 2e-16 ***
```

```
## s(quarter)5          -7.920e-03  1.547e-03  -5.120 3.05e-07 ***
## s(quarter)6          -3.383e-02  1.526e-03 -22.166  < 2e-16 ***
## s(quarter)7           3.294e-03  1.864e-03   1.768 0.077097 .
## s(quarter)8          -1.651e-02  2.188e-03  -7.544 4.58e-14 ***
## s(quarter)9           3.731e-03  1.938e-03   1.925 0.054267 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

5.5.10 10) Victims of violent deaths

```r
summary(prep_K18NoG, topics = 10)
```

```
##
## Call:
## estimateEffect(formula = 1:18 ~ party_id + populism + s(quarter),
##     stmobj = mySTM18NoG, metadata = DfmStm$meta, uncertainty = "Global")
##
##
## Topic 10:
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)            8.042e-02  9.762e-04  82.382  < 2e-16 ***
## party_idCI            -4.190e-02  2.008e-03 -20.862  < 2e-16 ***
## party_idFDI           -1.394e-02  1.141e-03 -12.219  < 2e-16 ***
## party_idFI            -2.769e-02  8.971e-04 -30.870  < 2e-16 ***
## party_idINDIPENDENTE -5.453e-02  3.463e-03 -15.743  < 2e-16 ***
## party_idIV            -1.428e-03  3.509e-03  -0.407   0.6841
## party_idLEGA          -3.397e-02  8.575e-04 -39.621  < 2e-16 ***
## party_idLEU            2.853e-02  2.569e-03  11.105  < 2e-16 ***
## party_idM5S           -1.284e-02  9.541e-04 -13.453  < 2e-16 ***
## party_idMISTO         -2.553e-02  1.196e-03 -21.350  < 2e-16 ***
## party_idREG_LEAGUES  -3.313e-02  4.504e-03  -7.356 1.90e-13 ***
## populism               3.117e-04  9.394e-06  33.180  < 2e-16 ***
## s(quarter)1            1.284e-02  1.734e-02   0.741   0.4588
## s(quarter)2           -6.420e-03  7.353e-03  -0.873   0.3826
## s(quarter)3            1.623e-02  3.847e-03   4.220 2.45e-05 ***
## s(quarter)4            2.250e-03  2.502e-03   0.899   0.3685
```

```
## s(quarter)5           2.587e-02  1.789e-03  14.460  < 2e-16 ***
## s(quarter)6           2.003e-02  1.665e-03  12.034  < 2e-16 ***
## s(quarter)7          -5.806e-03  2.220e-03  -2.616   0.0089 **
## s(quarter)8           6.090e-02  2.457e-03  24.781  < 2e-16 ***
## s(quarter)9           5.470e-02  2.556e-03  21.402  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

5.5.11 11) Public education

```r
summary(prep_K18NoG, topics = 11)
```

```
##
## Call:
## estimateEffect(formula = 1:18 ~ party_id + populism + s(quarter),
##     stmobj = mySTM18NoG, metadata = DfmStm$meta, uncertainty = "Global")
##
##
## Topic 11:
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)            3.579e-02  6.734e-04  53.141  < 2e-16 ***
## party_idCI            -1.396e-03  1.743e-03  -0.801 0.423143
## party_idFDI           -1.331e-02  7.347e-04 -18.116  < 2e-16 ***
## party_idFI            -6.545e-03  6.718e-04  -9.742  < 2e-16 ***
## party_idINDIPENDENTE  8.760e-02  3.871e-03  22.633  < 2e-16 ***
## party_idIV             2.406e-02  2.361e-03  10.190  < 2e-16 ***
## party_idLEGA          -7.458e-03  5.695e-04 -13.097  < 2e-16 ***
## party_idLEU           -9.759e-03  1.279e-03  -7.631 2.34e-14 ***
## party_idM5S            1.261e-03  6.943e-04   1.817 0.069271 .
## party_idMISTO         -1.960e-03  7.556e-04  -2.593 0.009501 **
## party_idREG_LEAGUES   -1.231e-02  3.125e-03  -3.938 8.22e-05 ***
## populism              -1.265e-04  5.173e-06 -24.457  < 2e-16 ***
## s(quarter)1           -4.459e-02  1.195e-02  -3.731 0.000191 ***
## s(quarter)2            3.844e-02  5.047e-03   7.616 2.62e-14 ***
## s(quarter)3            6.861e-03  2.628e-03   2.610 0.009049 **
## s(quarter)4            7.292e-03  1.760e-03   4.144 3.41e-05 ***
```

```
## s(quarter)5           3.348e-03  1.122e-03   2.984 0.002841 **
## s(quarter)6           7.625e-03  1.177e-03   6.480 9.21e-11 ***
## s(quarter)7          -2.087e-03  1.562e-03  -1.336 0.181581
## s(quarter)8           9.468e-03  1.815e-03   5.216 1.83e-07 ***
## s(quarter)9           3.045e-03  1.529e-03   1.992 0.046381 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

5.5.12 12) Anti-elitism

```r
summary(prep_K18NoG, topics = 12)
```

```
##
## Call:
## estimateEffect(formula = 1:18 ~ party_id + populism + s(quarter),
##     stmobj = mySTM18NoG, metadata = DfmStm$meta, uncertainty = "Global")
##
##
## Topic 12:
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)            5.103e-02  8.931e-04  57.138  < 2e-16 ***
## party_idCI            -2.581e-02  1.709e-03 -15.107  < 2e-16 ***
## party_idFDI            4.787e-02  1.064e-03  44.994  < 2e-16 ***
## party_idFI             2.080e-02  7.766e-04  26.779  < 2e-16 ***
## party_idINDIPENDENTE -5.279e-02  2.773e-03 -19.036  < 2e-16 ***
## party_idIV            -2.936e-02  2.475e-03 -11.860  < 2e-16 ***
## party_idLEGA           3.913e-02  7.937e-04  49.294  < 2e-16 ***
## party_idLEU            1.075e-02  1.744e-03   6.166 7.01e-10 ***
## party_idM5S           -8.031e-03  8.577e-04  -9.363  < 2e-16 ***
## party_idMISTO          2.991e-02  1.043e-03  28.674  < 2e-16 ***
## party_idREG_LEAGUES   -3.954e-02  3.454e-03 -11.447  < 2e-16 ***
## populism               3.871e-04  9.752e-06  39.697  < 2e-16 ***
## s(quarter)1           -1.084e-01  1.340e-02  -8.090 5.97e-16 ***
## s(quarter)2            6.294e-02  5.841e-03  10.776  < 2e-16 ***
## s(quarter)3           -3.460e-02  3.066e-03 -11.284  < 2e-16 ***
## s(quarter)4            4.570e-02  2.026e-03  22.554  < 2e-16 ***
```

```
## s(quarter)5          -3.033e-02  1.500e-03 -20.227  < 2e-16 ***
## s(quarter)6           2.375e-02  1.604e-03  14.804  < 2e-16 ***
## s(quarter)7           2.497e-02  2.059e-03  12.127  < 2e-16 ***
## s(quarter)8          -5.494e-03  2.433e-03  -2.258   0.0239 *
## s(quarter)9          -4.179e-04  2.036e-03  -0.205   0.8374
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

5.5.13 13) Social and TV live broadcasts

```
summary(prep_K18NoG, topics = 13)
```

```
##
## Call:
## estimateEffect(formula = 1:18 ~ party_id + populism + s(quarter),
##     stmobj = mySTM18NoG, metadata = DfmStm$meta, uncertainty = "Global")
##
##
## Topic 13:
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)            6.452e-02  8.661e-04  74.497  < 2e-16 ***
## party_idCI            -1.142e-02  1.640e-03  -6.958 3.45e-12 ***
## party_idFDI            2.154e-03  8.446e-04   2.550  0.01077 *
## party_idFI            -4.556e-03  7.494e-04  -6.080 1.20e-09 ***
## party_idINDIPENDENTE   3.917e-02  3.442e-03  11.381  < 2e-16 ***
## party_idIV             2.117e-02  2.818e-03   7.512 5.85e-14 ***
## party_idLEGA          -6.320e-03  6.529e-04  -9.681  < 2e-16 ***
## party_idLEU            1.580e-02  1.629e-03   9.696  < 2e-16 ***
## party_idM5S            4.571e-03  8.878e-04   5.149 2.62e-07 ***
## party_idMISTO          3.904e-03  9.396e-04   4.155 3.25e-05 ***
## party_idREG_LEAGUES   -7.715e-03  3.530e-03  -2.186  0.02882 *
## populism              -2.066e-04  6.458e-06 -31.992  < 2e-16 ***
## s(quarter)1           -2.228e-02  1.306e-02  -1.706  0.08796 .
## s(quarter)2            1.184e-03  5.736e-03   0.206  0.83652
## s(quarter)3           -1.595e-02  3.159e-03  -5.050 4.43e-07 ***
## s(quarter)4           -5.063e-03  1.964e-03  -2.578  0.00993 **
```

```
## s(quarter)5          -1.484e-03  1.521e-03  -0.975  0.32932
## s(quarter)6          -1.724e-02  1.290e-03 -13.364  < 2e-16 ***
## s(quarter)7           1.604e-03  1.957e-03   0.820  0.41242
## s(quarter)8          -1.577e-02  2.036e-03  -7.745 9.59e-15 ***
## s(quarter)9          -5.616e-03  1.710e-03  -3.284  0.00102 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

5.5.14 14) Junk topic

```
summary(prep_K18NoG, topics = 14)
```

```
##
## Call:
## estimateEffect(formula = 1:18 ~ party_id + populism + s(quarter),
##     stmobj = mySTM18NoG, metadata = DfmStm$meta, uncertainty = "Global")
##
##
## Topic 14:
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)             5.725e-02  9.351e-04  61.222  < 2e-16 ***
## party_idCI              1.926e-02  1.815e-03  10.610  < 2e-16 ***
## party_idFDI            -1.128e-02  8.600e-04 -13.118  < 2e-16 ***
## party_idFI              1.403e-02  8.038e-04  17.457  < 2e-16 ***
## party_idINDIPENDENTE  4.828e-03  2.828e-03   1.707 0.087759 .
## party_idIV              1.091e-02  2.436e-03   4.480 7.46e-06 ***
## party_idLEGA           -2.535e-02  6.679e-04 -37.963  < 2e-16 ***
## party_idLEU             6.289e-03  1.774e-03   3.545 0.000392 ***
## party_idM5S             8.759e-03  7.465e-04  11.733  < 2e-16 ***
## party_idMISTO           4.105e-03  8.938e-04   4.593 4.37e-06 ***
## party_idREG_LEAGUES    1.557e-01  4.723e-03  32.978  < 2e-16 ***
## populism               -9.792e-06  7.005e-06  -1.398 0.162155
## s(quarter)1            -2.033e-01  1.186e-02 -17.143  < 2e-16 ***
## s(quarter)2             8.861e-02  5.200e-03  17.041  < 2e-16 ***
## s(quarter)3            -5.690e-02  2.785e-03 -20.435  < 2e-16 ***
## s(quarter)4             3.977e-02  2.092e-03  19.012  < 2e-16 ***
```

```
## s(quarter)5          -1.481e-02  1.303e-03 -11.367  < 2e-16 ***
## s(quarter)6          -1.940e-02  1.326e-03 -14.632  < 2e-16 ***
## s(quarter)7          -2.989e-02  1.687e-03 -17.720  < 2e-16 ***
## s(quarter)8           3.374e-02  1.934e-03  17.443  < 2e-16 ***
## s(quarter)9          -1.922e-02  1.670e-03 -11.506  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

5.5.15 15) Junk topic

```
summary(prep_K18NoG, topics = 15)
```

```
##
## Call:
## estimateEffect(formula = 1:18 ~ party_id + populism + s(quarter),
##     stmobj = mySTM18NoG, metadata = DfmStm$meta, uncertainty = "Global")
##
##
## Topic 15:
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)             5.740e-02  8.020e-04  71.570  < 2e-16 ***
## party_idCI             -1.568e-02  1.712e-03  -9.156  < 2e-16 ***
## party_idFDI            -1.548e-03  8.308e-04  -1.863   0.0624 .
## party_idFI              7.207e-03  7.448e-04   9.677  < 2e-16 ***
## party_idINDIPENDENTE -3.752e-02  2.499e-03 -15.012  < 2e-16 ***
## party_idIV             -3.310e-02  2.149e-03 -15.403  < 2e-16 ***
## party_idLEGA           -6.455e-03  6.848e-04  -9.426  < 2e-16 ***
## party_idLEU            -2.188e-02  1.427e-03 -15.339  < 2e-16 ***
## party_idM5S            -2.777e-02  7.009e-04 -39.617  < 2e-16 ***
## party_idMISTO           7.864e-03  8.496e-04   9.257  < 2e-16 ***
## party_idREG_LEAGUES   -2.626e-02  3.237e-03  -8.111 5.02e-16 ***
## populism               -1.633e-04  6.565e-06 -24.878  < 2e-16 ***
## s(quarter)1            -2.363e-02  1.208e-02  -1.957   0.0503 .
## s(quarter)2             3.311e-04  5.448e-03   0.061   0.9515
## s(quarter)3            -1.635e-02  2.850e-03  -5.738 9.59e-09 ***
## s(quarter)4            -7.078e-04  1.952e-03  -0.363   0.7169
```

```
## s(quarter)5          -1.568e-02  1.304e-03 -12.020  < 2e-16 ***
## s(quarter)6           2.511e-02  1.452e-03  17.298  < 2e-16 ***
## s(quarter)7           1.579e-02  1.990e-03   7.937 2.08e-15 ***
## s(quarter)8           4.252e-03  2.264e-03   1.878   0.0604 .
## s(quarter)9           2.434e-03  1.651e-03   1.474   0.1405
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

5.5.16 16) Olympics game

```r
summary(prep_K18NoG, topics = 16)
```

```
##
## Call:
## estimateEffect(formula = 1:18 ~ party_id + populism + s(quarter),
##     stmobj = mySTM18NoG, metadata = DfmStm$meta, uncertainty = "Global")
##
##
## Topic 16:
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)            8.277e-02  9.054e-04  91.419  < 2e-16 ***
## party_idCI             2.621e-02  1.945e-03  13.477  < 2e-16 ***
## party_idFDI           -1.523e-02  9.020e-04 -16.881  < 2e-16 ***
## party_idFI             1.093e-02  7.193e-04  15.190  < 2e-16 ***
## party_idINDIPENDENTE  2.256e-02  3.454e-03   6.531 6.55e-11 ***
## party_idIV            -2.612e-03  2.595e-03  -1.007    0.314
## party_idLEGA          -3.736e-03  6.621e-04  -5.643 1.67e-08 ***
## party_idLEU           -2.525e-02  1.629e-03 -15.496  < 2e-16 ***
## party_idM5S           -2.141e-02  7.393e-04 -28.961  < 2e-16 ***
## party_idMISTO         -2.293e-02  8.429e-04 -27.205  < 2e-16 ***
## party_idREG_LEAGUES   -3.986e-02  3.363e-03 -11.851  < 2e-16 ***
## populism               3.233e-05  7.508e-06   4.306 1.66e-05 ***
## s(quarter)1           -2.641e-01  1.345e-02 -19.636  < 2e-16 ***
## s(quarter)2            6.577e-02  5.742e-03  11.455  < 2e-16 ***
## s(quarter)3           -4.197e-02  2.998e-03 -13.998  < 2e-16 ***
## s(quarter)4           -3.347e-02  1.869e-03 -17.914  < 2e-16 ***
```

```
## s(quarter)5          -2.866e-02  1.468e-03 -19.514  < 2e-16 ***
## s(quarter)6           3.937e-02  1.526e-03  25.804  < 2e-16 ***
## s(quarter)7          -5.056e-02  1.760e-03 -28.728  < 2e-16 ***
## s(quarter)8          -1.539e-02  2.033e-03  -7.569 3.77e-14 ***
## s(quarter)9          -1.956e-02  1.900e-03 -10.292  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

5.5.17 17) Right-wing parties topic

```
summary(prep_K18NoG, topics = 17)
```

```
##
## Call:
## estimateEffect(formula = 1:18 ~ party_id + populism + s(quarter),
##     stmobj = mySTM18NoG, metadata = DfmStm$meta, uncertainty = "Global")
##
##
## Topic 17:
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)           2.705e-02  8.315e-04  32.533  < 2e-16 ***
## party_idCI           -7.876e-03  1.513e-03  -5.207 1.92e-07 ***
## party_idFDI           9.296e-02  9.446e-04  98.413  < 2e-16 ***
## party_idFI            4.359e-03  6.370e-04   6.842 7.81e-12 ***
## party_idINDIPENDENTE -1.773e-02  2.505e-03  -7.077 1.47e-12 ***
## party_idIV           -9.605e-03  2.152e-03  -4.463 8.07e-06 ***
## party_idLEGA          1.312e-01  7.966e-04 164.679  < 2e-16 ***
## party_idLEU           5.859e-03  1.445e-03   4.056 5.00e-05 ***
## party_idM5S           3.315e-03  6.693e-04   4.954 7.28e-07 ***
## party_idMISTO         1.468e-02  8.196e-04  17.909  < 2e-16 ***
## party_idREG_LEAGUES   5.077e-03  3.389e-03   1.498   0.1341
## populism              1.592e-04  7.516e-06  21.181  < 2e-16 ***
## s(quarter)1          -2.743e-02  1.286e-02  -2.133   0.0329 *
## s(quarter)2           4.079e-03  5.654e-03   0.721   0.4706
## s(quarter)3           2.280e-02  3.039e-03   7.500 6.38e-14 ***
## s(quarter)4          -3.036e-02  1.906e-03 -15.932  < 2e-16 ***
```

```
## s(quarter)5           1.242e-03  1.457e-03   0.853   0.3938    
## s(quarter)6          -2.162e-02  1.542e-03 -14.017  < 2e-16 ***
## s(quarter)7          -1.922e-02  1.789e-03 -10.741  < 2e-16 ***
## s(quarter)8          -3.255e-02  2.020e-03 -16.113  < 2e-16 ***
## s(quarter)9          -2.685e-02  1.638e-03 -16.390  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

5.5.18 18) Junk topic

```r
summary(prep_K18NoG, topics = 18)
```

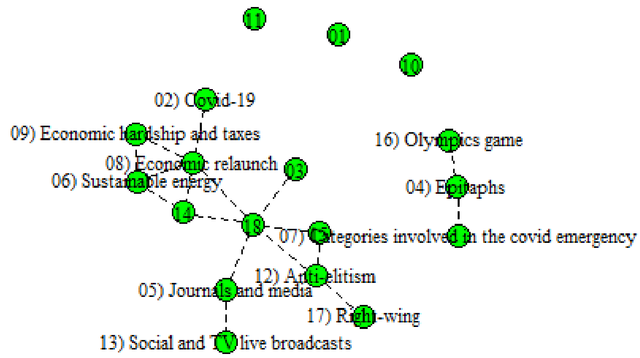
```
##
## Call:
## estimateEffect(formula = 1:18 ~ party_id + populism + s(quarter),
##     stmobj = mySTM18NoG, metadata = DfmStm$meta, uncertainty = "Global")
##
##
## Topic 18:
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)            2.362e-02  2.546e-04  92.797  < 2e-16 ***
## party_idCI            -3.237e-03  5.155e-04  -6.280 3.38e-10 ***
## party_idFDI           -1.660e-03  2.596e-04  -6.395 1.61e-10 ***
## party_idFI            -2.114e-03  2.476e-04  -8.537  < 2e-16 ***
## party_idINDIPENDENTE -7.411e-03  8.054e-04  -9.202  < 2e-16 ***
## party_idIV            -1.243e-03  8.439e-04  -1.473 0.140662
## party_idLEGA          -2.577e-03  2.084e-04 -12.366  < 2e-16 ***
## party_idLEU            1.373e-05  5.064e-04   0.027 0.978365
## party_idM5S            1.219e-03  2.349e-04   5.191 2.10e-07 ***
## party_idMISTO         -4.740e-05  2.875e-04  -0.165 0.869045
## party_idREG_LEAGUES   -3.446e-03  1.049e-03  -3.285 0.001020 **
## populism               4.098e-06  2.219e-06   1.847 0.064741 .
## s(quarter)1           -7.076e-03  3.967e-03  -1.784 0.074463 .
## s(quarter)2            4.528e-04  1.668e-03   0.272 0.785963
## s(quarter)3           -1.972e-03  8.963e-04  -2.200 0.027784 *
## s(quarter)4           -1.095e-03  5.735e-04  -1.909 0.056303 .
```

```
## s(quarter)5          -1.135e-03  4.871e-04  -2.331 0.019767 *
## s(quarter)6          -1.524e-03  4.244e-04  -3.592 0.000328 ***
## s(quarter)7           1.864e-03  6.133e-04   3.039 0.002370 **
## s(quarter)8          -1.110e-03  6.116e-04  -1.816 0.069433 .
## s(quarter)9           1.864e-03  6.191e-04   3.011 0.002601 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

5.6 Interpretation

5.6.1 Correlation between topics

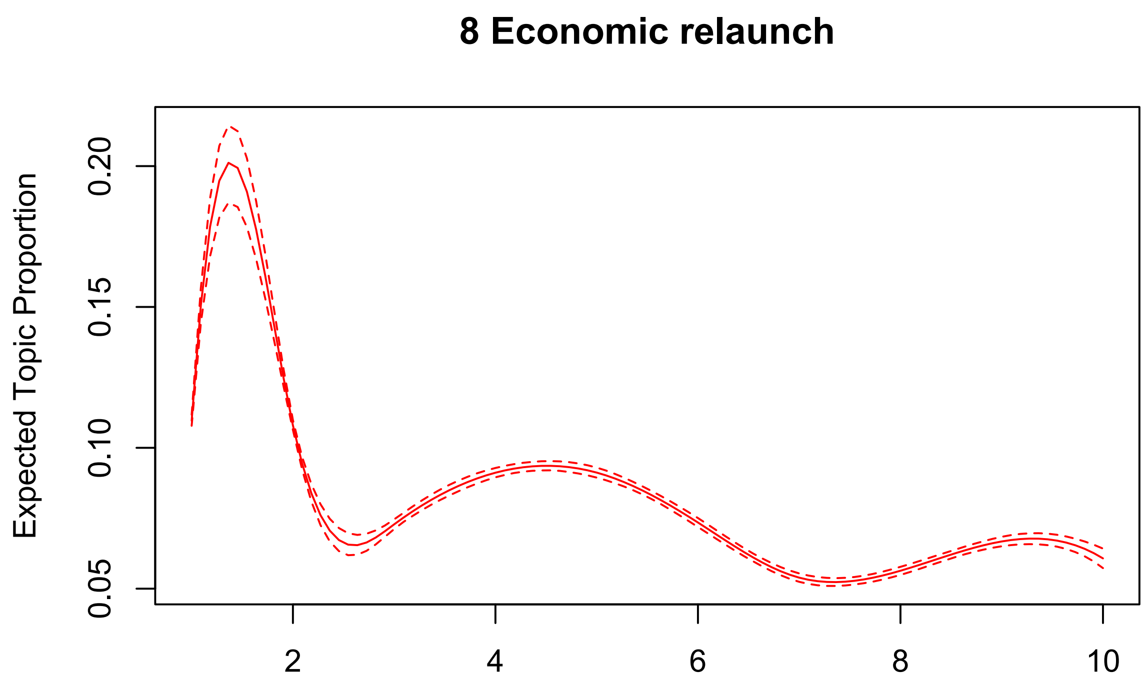
```r
mod.out.corr <- topicCorr(mySTM18NoG)

ll <- c("01","02) Covid-19","03",
        "04) Epitaphs",
        "05) Journals and media",
        "06) Sustainable energy",
        "07) Categories involved in the covid emergency",
        "08) Economic relaunch",
        "09) Economic hardship and taxes","10","11",
        "12) Anti-elitism",
        "13) Social and TV live broadcasts","14","15",
        "16) Olympics game",
        "17) Right-wing","18")

set.seed(10)
plot(mod.out.corr, vlabels = ll, layout = layout.auto )
```


11
01
10
02) Covid-19
09) Economic hardship and taxes
08) Economic relaunch
06) Sustainable energy
03
14
19
16) Olympics game
04) Epitaphs
07) Categories involved in the covid emergency
12) Anti-elitism
05) Journals and media
17) Right-wing
13) Social and TV live broadcasts

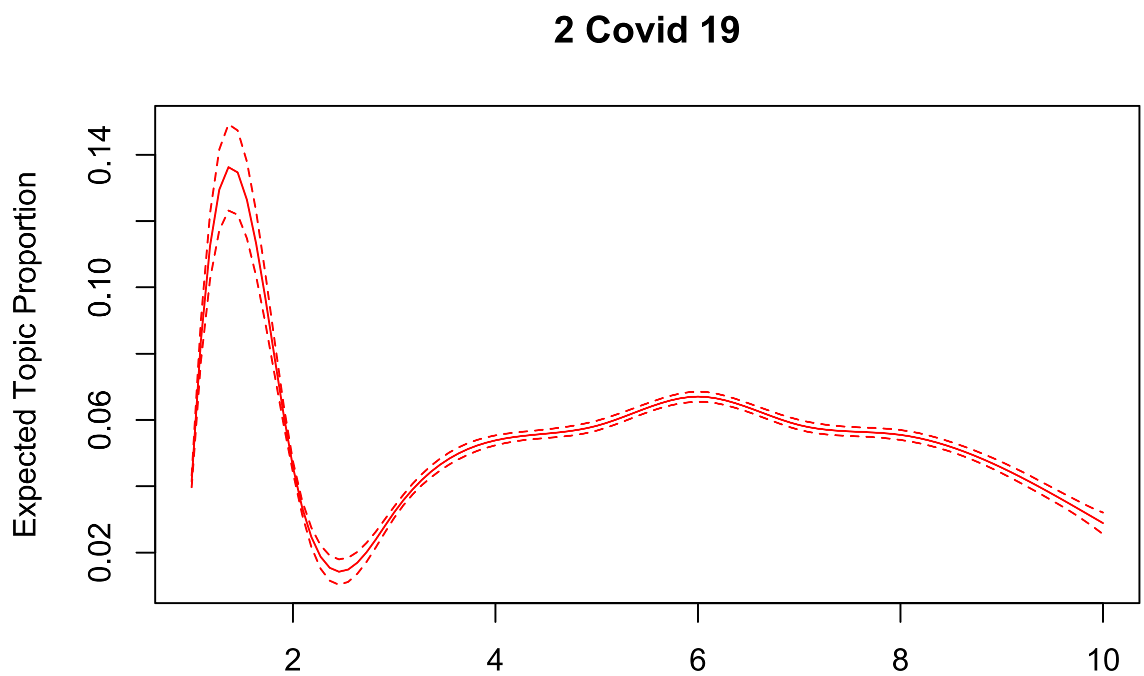
5.6.2 Topic variation over time

Covid cluster

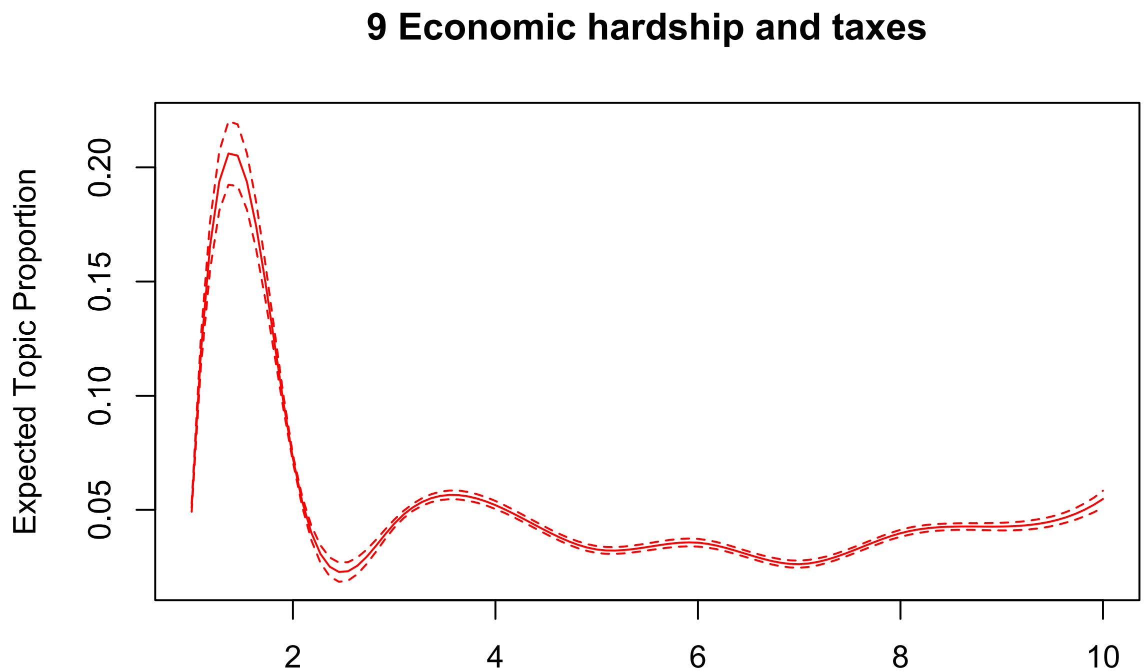
```
# TOPIC 8 Economic relaunch
plot(prep_K18NoG, "quarter", method = "continuous",
     topics = 8, printlegend = F, main = "8 Economic relaunch")
```



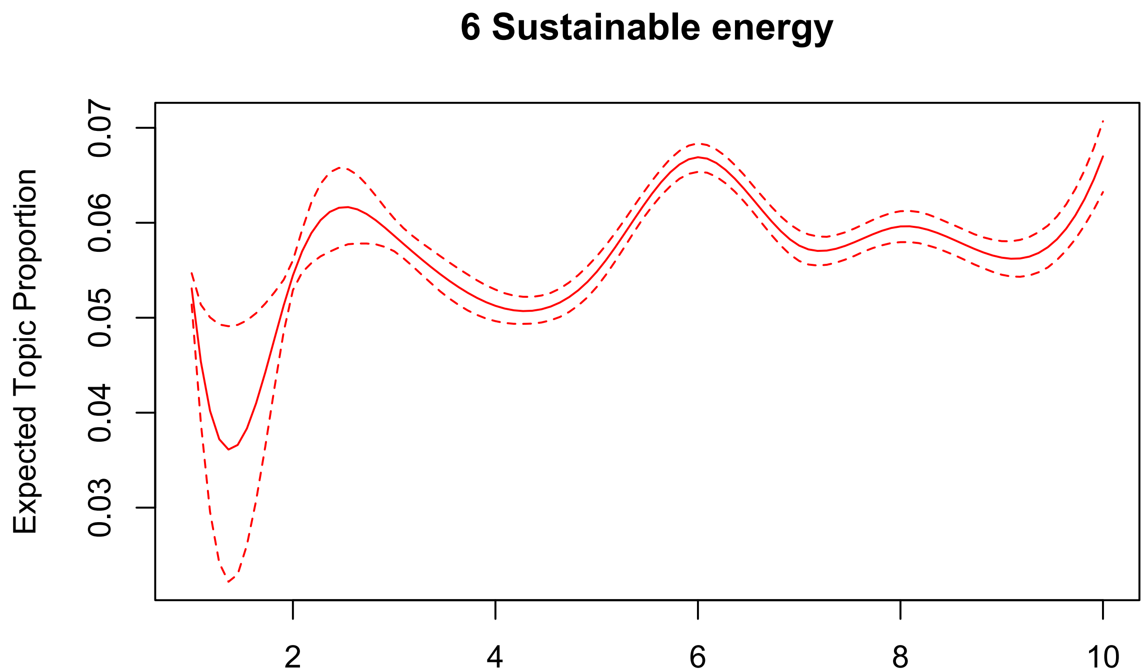
```r
# TOPIC 2 Covid 19
plot(prep_K18NoG, "quarter", method = "continuous",
     topics = 2, printlegend = F, main = "2 Covid 19")
```



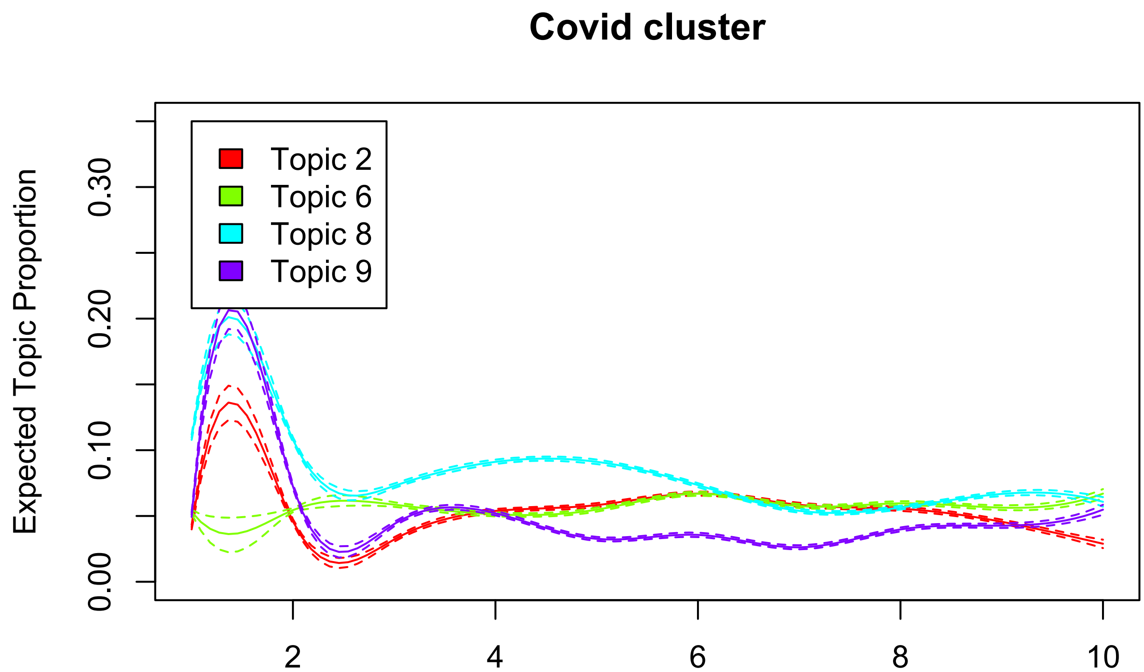

```r
# TOPIC 9 Economic hardship and taxes
plot(prep_K18NoG, "quarter", method = "continuous",
     topics = 9, printlegend = F, main = "9 Economic hardship and taxes")
```




```r
# TOPIC 6 Sustainable energy
plot(prep_K18NoG, "quarter", method = "continuous",
     topics = 6, printlegend = F, main = "6 Sustainable energy")
```

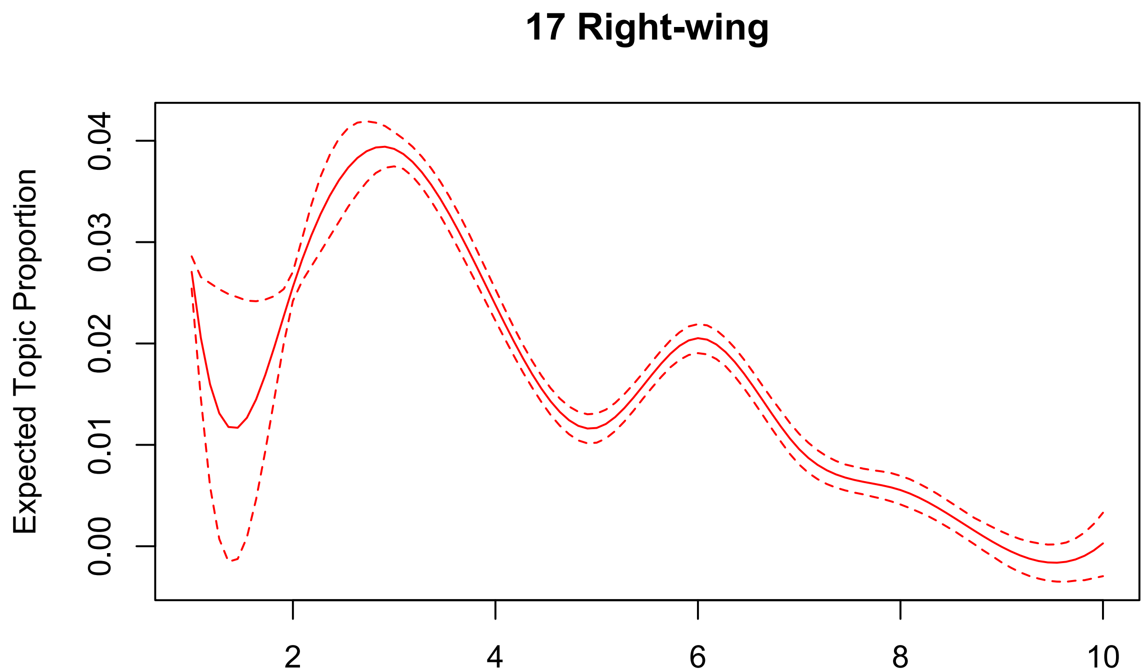



```r
# Covid cluster
plot(prep_K18NoG, "quarter", method = "continuous",
     topics = c(2,6,8,9), printlegend = T,
     ylim = c(0,0.35), main = "Covid cluster")
```

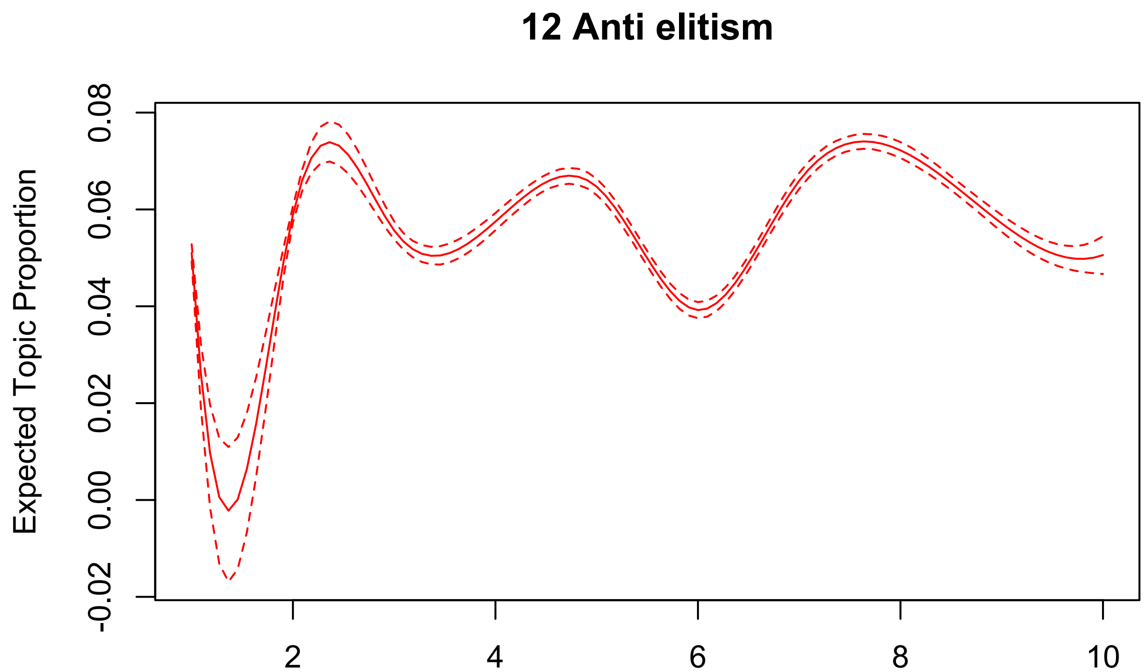


Populism cluster

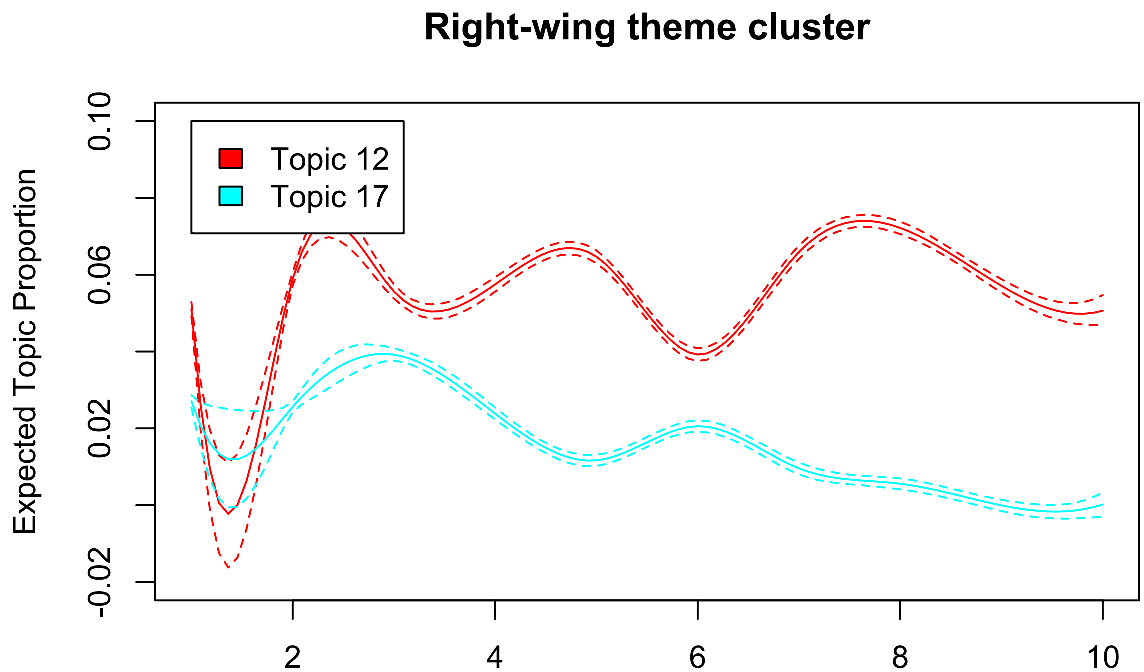
```
# TOPIC 17
plot(prep_K18NoG, "quarter", method = "continuous",
     topics = 17, printlegend = F, main = "17 Right-wing")
```




```
# TOPIC 12
plot(prep_K18NoG, "quarter", method = "continuous",
     topics = 12, printlegend = F, main = "12 Anti elitism")
```

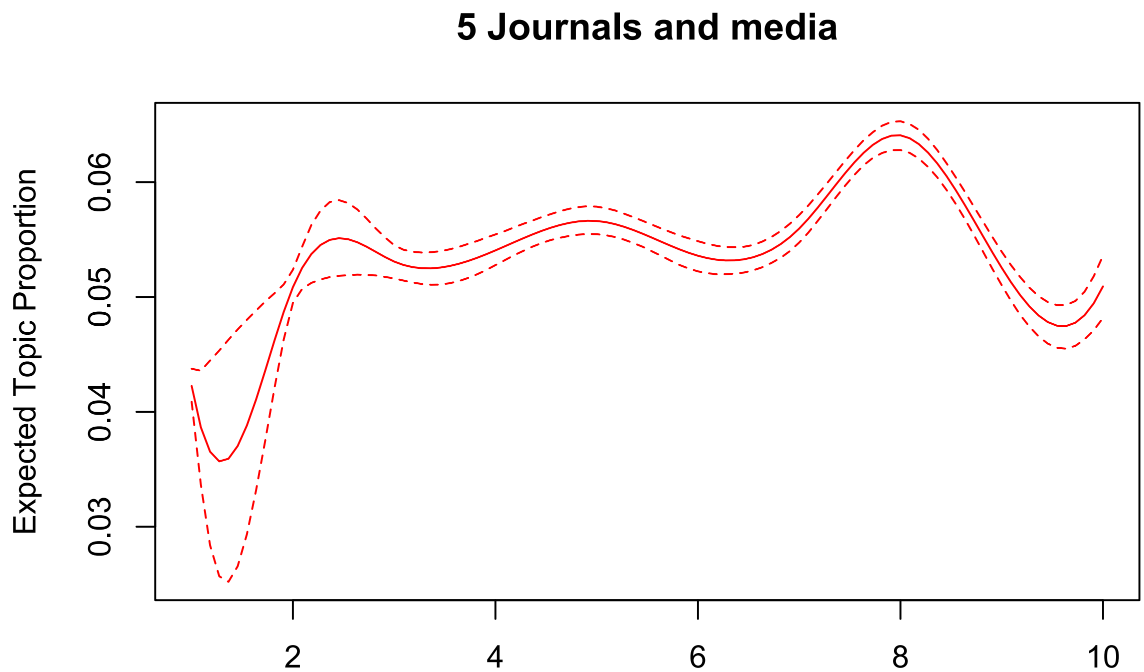



```r
# Right-wing theme cluster cluster
plot(prep_K18NoG, "quarter", method = "continuous",
     topics = c(12,17), printlegend = T,
     ylim = c(-0.02,0.1), main = "Right-wing theme cluster")
```

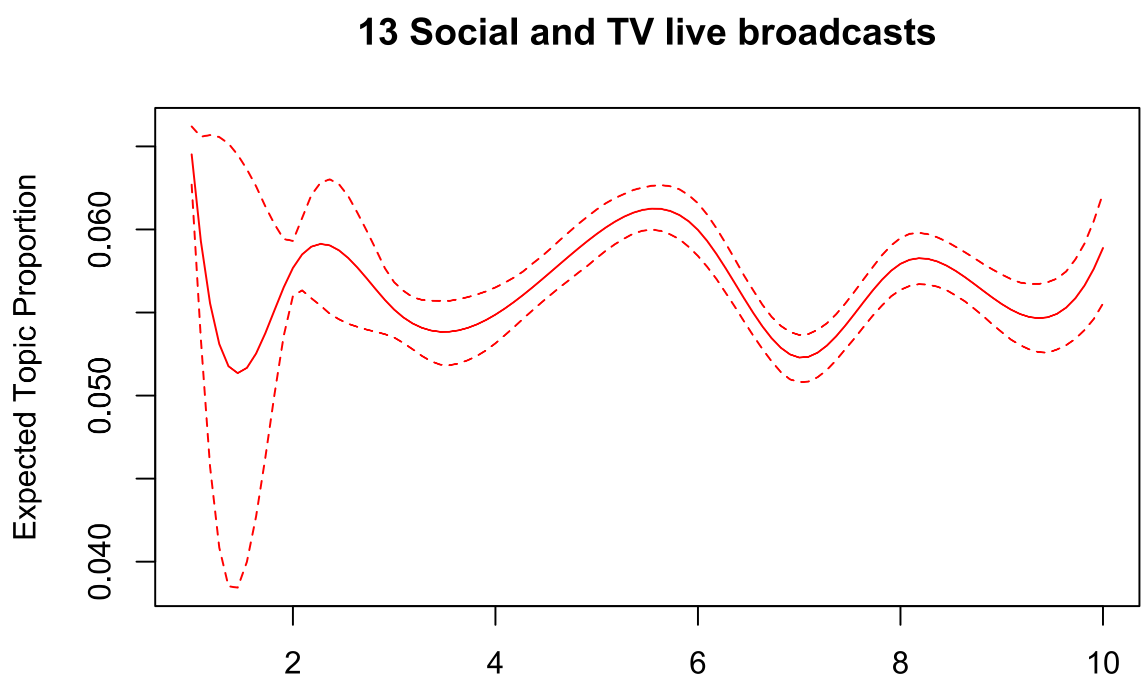


Communication cluster

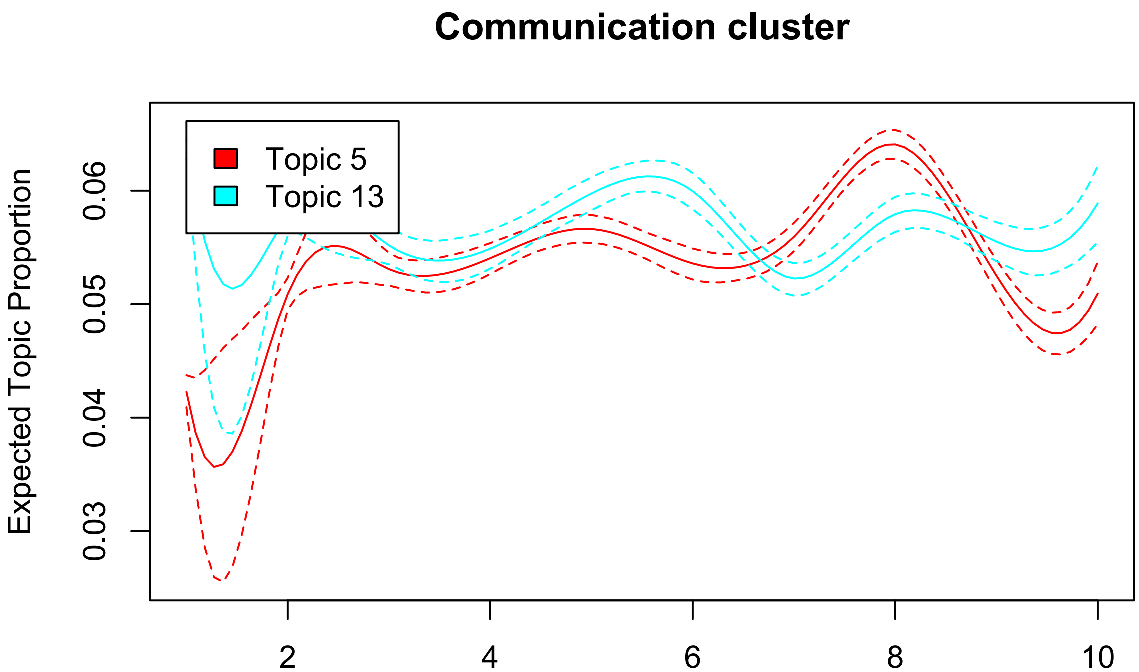
```
# TOPIC 5
plot(prep_K18NoG, "quarter", method = "continuous",
     topics = 5, printlegend = F, main = "5 Journals and media")
```




```
# TOPIC 13
plot(prep_K18NoG, "quarter", method = "continuous",
     topics = 13, printlegend = F, main = "13 Social and TV live broadcasts")
```




```r
# Communication cluster
plot(prep_K18NoG, "quarter", method = "continuous",
     topics = c(5,13), printlegend = T, main = "Communication cluster")
```



Communication cluster
Topic 5
Topic 13
Expected Topic Proportion
0.03
0.04
0.05
0.06
2
4
6
8
10

6 FER: Facial Emotion Recognition Analysis

Report on the analysis made with FER Python package

6.1 Import the datasets

```
# CONTE
Conte_07_03_22_00 <- read_csv("data/video_emotions/Conte_07-03-22_00.csv",
    col_types = cols(angry = col_number(),
        disgust = col_number(), fear = col_number(),
        happy = col_number(), sad = col_number(),
        surprise = col_number(), neutral = col_number()))
```

```
## New names:
## * `` -> `...1`
```

```
Conte_09_03_22_00 <- read_csv("data/video_emotions/Conte_09-03-22_00.csv",
    col_types = cols(angry = col_number(),
        disgust = col_number(), fear = col_number(),
        happy = col_number(), sad = col_number(),
        surprise = col_number(), neutral = col_number()))
```

```
## New names:
## * `` -> `...1`
```

```
Conte_22_02_22_00 <- read_csv("data/video_emotions/Conte_22-02-22_00.csv",
    col_types = cols(angry = col_number(),
        disgust = col_number(), fear = col_number(),
        happy = col_number(), sad = col_number(),
        surprise = col_number(), neutral = col_number()))
```

```
## New names:
## * `` -> `...1`
```

```r
Conte_23_02_22_00 <- read_csv("data/video_emotions/Conte_23-02-22_00.csv",
    col_types = cols(angry = col_number(),
        disgust = col_number(), fear = col_number(),
        happy = col_number(), sad = col_number(),
        surprise = col_number(), neutral = col_number()))
```

```
## New names:
## * `` -> `...1`
```

```r
Conte_23_02_22_01 <- read_csv("data/video_emotions/Conte_23-02-22_01.csv",
    col_types = cols(angry = col_number(),
        disgust = col_number(), fear = col_number(),
        happy = col_number(), sad = col_number(),
        surprise = col_number(), neutral = col_number()))
```

```
## New names:
## * `` -> `...1`
```

```r
Conte_24_02_22_01 <- read_csv("data/video_emotions/Conte_24-02-22_01.csv",
    col_types = cols(angry = col_number(),
        disgust = col_number(), fear = col_number(),
        happy = col_number(), sad = col_number(),
        surprise = col_number(), neutral = col_number()))
```

```
## New names:
## * `` -> `...1`
```

```r
Conte_28_02_22_00 <- read_csv("data/video_emotions/Conte_28-02-22_00.csv",
    col_types = cols(angry = col_number(),
        disgust = col_number(), fear = col_number(),
        happy = col_number(), sad = col_number(),
        surprise = col_number(), neutral = col_number()))
```

```
## New names:
## * `` -> `...1`
```

```r
# LETTA
Letta_03_03_22_00 <- read_csv("data/video_emotions/Letta_03-03-22_00.csv",
    col_types = cols(angry = col_number(),
        disgust = col_number(), fear = col_number(),
        happy = col_number(), sad = col_number(),
        surprise = col_number(), neutral = col_number()))
```

```
## New names:
## * `` -> `...1`
```

```r
Letta_06_04_22_00 <- read_csv("data/video_emotions/Letta_06-04-22_00.csv",
    col_types = cols(angry = col_number(),
        disgust = col_number(), fear = col_number(),
        happy = col_number(), sad = col_number(),
        surprise = col_number(), neutral = col_number()))
```

```
## New names:
## * `` -> `...1`
```

```r
# MELONI
Meloni_1_03_2022 <- read_csv("data/video_emotions/Meloni_1-03-2022.csv",
    col_types = cols(angry = col_number(),
        disgust = col_number(), fear = col_number(),
        happy = col_number(), sad = col_number(),
        surprise = col_number(), neutral = col_number()))
```

```
## New names:
## * `` -> `...1`
```

```r
Meloni_11_03_2022_02 <- read_csv("data/video_emotions/Meloni_11-03-2022_02.csv",
    col_types = cols(angry = col_number(),
        disgust = col_number(), fear = col_number(),
        happy = col_number(), sad = col_number(),
        surprise = col_number(), neutral = col_number()))
```

```
## New names:
## * `` -> `...1`
```

```r
Meloni_11_03_2022 <- read_csv("data/video_emotions/Meloni_11-03-2022.csv",
    col_types = cols(angry = col_number(),
        disgust = col_number(), fear = col_number(),
        happy = col_number(), sad = col_number(),
        surprise = col_number(), neutral = col_number()))
```

```
## New names:
## * `` -> `...1`
```

```r
Meloni_15_03_2022 <- read_csv("data/video_emotions/Meloni_15-03-2022.csv",
    col_types = cols(angry = col_number(),
        disgust = col_number(), fear = col_number(),
        happy = col_number(), sad = col_number(),
        surprise = col_number(), neutral = col_number()))
```

```
## New names:
## * `` -> `...1`
```

```r
Meloni_22_03_2022 <- read_csv("data/video_emotions/Meloni_22-03-2022.csv",
    col_types = cols(angry = col_number(),
        disgust = col_number(), fear = col_number(),
        happy = col_number(), sad = col_number(),
        surprise = col_number(), neutral = col_number()))
```

```
## New names:
## * `` -> `...1`
```

```r
Meloni_29_03_2022 <- read_csv("data/video_emotions/Meloni_29-03-2022.csv",
    col_types = cols(angry = col_number(),
        disgust = col_number(), fear = col_number(),
        happy = col_number(), sad = col_number(),
        surprise = col_number(), neutral = col_number()))
```

```
## New names:
## * `` -> `...1`
```

```r
Meloni_31_03_2022<- read_csv("data/video_emotions/Meloni_31-03-2022.csv",
    col_types = cols(angry = col_number(),
        disgust = col_number(), fear = col_number(),
```

```
        happy = col_number(), sad = col_number(),
        surprise = col_number(), neutral = col_number()))
```

```
## New names:
## * `` -> `...1`
```

```
# RENZI
Renzi_19_04_2022 <- read_csv("data/video_emotions/Renzi_19-04-2022.csv",
    col_types = cols(angry = col_number(),
        disgust = col_number(), fear = col_number(),
        happy = col_number(), sad = col_number(),
        surprise = col_number(), neutral = col_number()))
```

```
## New names:
## * `` -> `...1`
```

```
Renzi_30_03_2022 <- read_csv("data/video_emotions/Renzi_30-03-2022.csv",
    col_types = cols(angry = col_number(),
        disgust = col_number(), fear = col_number(),
        happy = col_number(), sad = col_number(),
        surprise = col_number(), neutral = col_number()))
```

```
## New names:
## * `` -> `...1`
```

```
# SALVINI
Salvini_08_03_2022 <- read_csv("data/video_emotions/Salvini_08-03-2022.csv",
    col_types = cols(angry = col_number(),
        disgust = col_number(), fear = col_number(),
        happy = col_number(), sad = col_number(),
        surprise = col_number(), neutral = col_number()))
```

```
## New names:
## * `` -> `...1`
```

```r
Salvini_08_04_2022_02 <- read_csv("data/video_emotions/Salvini_08-04-2022_02.csv",
    col_types = cols(angry = col_number(),
        disgust = col_number(), fear = col_number(),
        happy = col_number(), sad = col_number(),
        surprise = col_number(), neutral = col_number()))
```

```
## New names:
## * `` -> `...1`
```

```r
Salvini_16_03_2022 <- read_csv("data/video_emotions/Salvini_16-03-2022.csv",
    col_types = cols(angry = col_number(),
        disgust = col_number(), fear = col_number(),
        happy = col_number(), sad = col_number(),
        surprise = col_number(), neutral = col_number()))
```

```
## New names:
## * `` -> `...1`
```

6.2 Conte datasets

```r
#1
# Conte_07_03_22_00
Conte_07_03_22_00_prop <- c(
  angry <-  sum(Conte_07_03_22_00$angry),
  disgust <- sum(Conte_07_03_22_00$disgust),
  fear <- sum(Conte_07_03_22_00$fear),
  happy <- sum(Conte_07_03_22_00$happy),
```

```r
  sad <- sum(Conte_07_03_22_00$sad),
  surprise <- sum(Conte_07_03_22_00$surprise),
  meutral <- sum(Conte_07_03_22_00$neutral)
)
```

```r
#2
# Conte_09_03_22_00
Conte_09_03_22_00_prop <- c(
  angry <-  sum(Conte_09_03_22_00$angry),
  disgust <- sum(Conte_09_03_22_00$disgust),
  fear <- sum(Conte_09_03_22_00$fear),
  happy <- sum(Conte_09_03_22_00$happy),
  sad <- sum(Conte_09_03_22_00$sad),
  surprise <- sum(Conte_09_03_22_00$surprise),
  meutral <- sum(Conte_09_03_22_00$neutral)
)
```

```r
#3
# Conte_22_02_22_00
i = Conte_22_02_22_00
Conte_22_02_22_00_prop <- c(
  angry <-  sum(i$angry),
  disgust <- sum(i$disgust),
  fear <- sum(i$fear),
  happy <- sum(i$happy),
  sad <- sum(i$sad),
  surprise <- sum(i$surprise),
  meutral <- sum(i$neutral)
)
```

```r
#4
# Conte_23_02_22_00
i = Conte_23_02_22_00
Conte_23_02_22_00_prop <- c(
  angry <-  sum(i$angry),
  disgust <- sum(i$disgust),
  fear <- sum(i$fear),
  happy <- sum(i$happy),
  sad <- sum(i$sad),
  surprise <- sum(i$surprise),
  meutral <- sum(i$neutral)
)
```

```r
#5
# Conte_23_02_22_01
i = Conte_23_02_22_01
Conte_23_02_22_01_prop <- c(
  angry <-  sum(i$angry),
  disgust <- sum(i$disgust),
  fear <- sum(i$fear),
  happy <- sum(i$happy),
  sad <- sum(i$sad),
  surprise <- sum(i$surprise),
  meutral <- sum(i$neutral)
)
```

```r
#6
# Conte_24_02_22_01
i = Conte_24_02_22_01
Conte_24_02_22_01_prop <- c(
```

```
  angry <-  sum(i$angry),
  disgust <- sum(i$disgust),
  fear <- sum(i$fear),
  happy <- sum(i$happy),
  sad <- sum(i$sad),
  surprise <- sum(i$surprise),
  meutral <- sum(i$neutral)
)
```

```
#7
# Conte_28_02_22_00
i = Conte_28_02_22_00
Conte_28_02_22_00_prop <- c(
  angry <-  sum(i$angry),
  disgust <- sum(i$disgust),
  fear <- sum(i$fear),
  happy <- sum(i$happy),
  sad <- sum(i$sad),
  surprise <- sum(i$surprise),
  meutral <- sum(i$neutral)
)
```

```
conte <- rbind(Conte_07_03_22_00_prop,
               Conte_09_03_22_00_prop,
               Conte_22_02_22_00_prop,
               Conte_23_02_22_00_prop,
               Conte_23_02_22_01_prop,
               Conte_24_02_22_01_prop,
               Conte_28_02_22_00_prop
               )
```

```
emo_label <- colnames(Conte_07_03_22_00)[3:9]

colnames(conte) <- emo_label

conte <- as.data.frame(conte)

tot_conte <- max(Conte_07_03_22_00$...1) +
              max(Conte_09_03_22_00$...1) +
              max(Conte_22_02_22_00$...1) +
              max(Conte_23_02_22_00$...1) +
              max(Conte_23_02_22_01$...1) +
              max(Conte_24_02_22_01$...1) +
              max(Conte_28_02_22_00$...1)

conte[8,] <- c(sum(conte$angry)/tot_conte * 100,
               sum(conte$disgust)/tot_conte *100,
               sum(conte$fear)/tot_conte *100,
               sum(conte$happy)/tot_conte *100,
               sum(conte$sad)/tot_conte *100,
               sum(conte$surprise)/tot_conte * 100,
               sum(conte$neutral)/tot_conte *100)
```

6.3 Letta datasets

```
#1
# Letta_03_03_22_00
i = Letta_03_03_22_00
Letta_03_03_22_00_prop <- c(
  angry <-  sum(i$angry),
```

```
  disgust <- sum(i$disgust),
  fear <- sum(i$fear),
  happy <- sum(i$happy),
  sad <- sum(i$sad),
  surprise <- sum(i$surprise),
  meutral <- sum(i$neutral)
)
```

```
#2
# Letta_06_04_22_00
i = Letta_06_04_22_00
Letta_06_04_22_00_prop <- c(
  angry <-  sum(i$angry),
  disgust <- sum(i$disgust),
  fear <- sum(i$fear),
  happy <- sum(i$happy),
  sad <- sum(i$sad),
  surprise <- sum(i$surprise),
  meutral <- sum(i$neutral)
)
```

```
letta <- rbind(Letta_03_03_22_00_prop,
               Letta_06_04_22_00_prop
               )
colnames(letta) <- emo_label

letta <- as.data.frame(letta)

tot_letta <- max(Letta_03_03_22_00$...1) +
              max(Letta_06_04_22_00$...1)
```

```r
letta[3,] <- c(sum(letta$angry)/tot_letta * 100,
               sum(letta$disgust)/tot_letta *100,
               sum(letta$fear)/tot_letta *100,
               sum(letta$happy)/tot_letta *100,
               sum(letta$sad)/tot_letta *100,
               sum(letta$surprise)/tot_letta * 100,
               sum(letta$neutral)/tot_letta *100)
```

6.4 Meloni datasets

```r
#1
# Meloni_1_03_2022
i = Meloni_1_03_2022
Meloni_1_03_2022_prop <- c(
  angry <-  sum(i$angry),
  disgust <- sum(i$disgust),
  fear <- sum(i$fear),
  happy <- sum(i$happy),
  sad <- sum(i$sad),
  surprise <- sum(i$surprise),
  neutral <- sum(i$neutral)
)
```

```r
#2
# Meloni_11_03_2022_02
i = Meloni_11_03_2022_02
Meloni_11_03_2022_02_prop <- c(
  angry <-  sum(i$angry),
  disgust <- sum(i$disgust),
```

```
  fear <- sum(i$fear),
  happy <- sum(i$happy),
  sad <- sum(i$sad),
  surprise <- sum(i$surprise),
  meutral <- sum(i$neutral)
)
```

```
#3
# Meloni_11_03_2022
i = Meloni_11_03_2022
Meloni_11_03_2022_prop <- c(
  angry <-  sum(i$angry),
  disgust <- sum(i$disgust),
  fear <- sum(i$fear),
  happy <- sum(i$happy),
  sad <- sum(i$sad),
  surprise <- sum(i$surprise),
  meutral <- sum(i$neutral)
)
```

```
#4
# Meloni_15_03_2022
i = Meloni_15_03_2022
Meloni_15_03_2022_prop <- c(
  angry <-  sum(i$angry),
  disgust <- sum(i$disgust),
  fear <- sum(i$fear),
  happy <- sum(i$happy),
  sad <- sum(i$sad),
  surprise <- sum(i$surprise),
```

```
  meutral <- sum(i$neutral)
)
```

```
#5
# Meloni_22_03_2022
i = Meloni_22_03_2022
Meloni_22_03_2022_prop <- c(
  angry <-  sum(i$angry),
  disgust <- sum(i$disgust),
  fear <- sum(i$fear),
  happy <- sum(i$happy),
  sad <- sum(i$sad),
  surprise <- sum(i$surprise),
  meutral <- sum(i$neutral)
)
```

```
#6
# Meloni_29_03_2022
i = Meloni_29_03_2022
Meloni_29_03_2022_prop <- c(
  angry <-  sum(i$angry),
  disgust <- sum(i$disgust),
  fear <- sum(i$fear),
  happy <- sum(i$happy),
  sad <- sum(i$sad),
  surprise <- sum(i$surprise),
  meutral <- sum(i$neutral)
)
```

```r
#7
# Meloni_31_03_2022
i = Meloni_31_03_2022
Meloni_31_03_2022_prop <- c(
  angry <-  sum(i$angry),
  disgust <- sum(i$disgust),
  fear <- sum(i$fear),
  happy <- sum(i$happy),
  sad <- sum(i$sad),
  surprise <- sum(i$surprise),
  meutral <- sum(i$neutral)
)
```

```r
meloni <- rbind(Meloni_1_03_2022_prop,
               Meloni_11_03_2022_02_prop,
               Meloni_11_03_2022_prop,
               Meloni_15_03_2022_prop,
               Meloni_22_03_2022_prop,
               Meloni_29_03_2022_prop,
               Meloni_31_03_2022_prop
               )
colnames(meloni) <- emo_label

meloni <- as.data.frame(meloni)



tot_meloni <- max(Meloni_1_03_2022$...1) +
              max(Meloni_11_03_2022_02$...1)+
              max(Meloni_11_03_2022$...1)+
              max(Meloni_15_03_2022$...1)+
```

```
              max(Meloni_22_03_2022$...1)+
              max(Meloni_29_03_2022$...1)+
              max(Meloni_31_03_2022$...1)

meloni[8,] <- c(sum(meloni$angry)/tot_meloni * 100,
                sum(meloni$disgust)/tot_meloni *100,
                sum(meloni$fear)/tot_meloni *100,
                sum(meloni$happy)/tot_meloni *100,
                sum(meloni$sad)/tot_meloni *100 ,
                sum(meloni$surprise)/tot_meloni * 100,
                sum(meloni$neutral)/tot_meloni *100)
```

6.5 Renzi datasets

```
#1
# Renzi_19_04_2022
i = Renzi_19_04_2022
Renzi_19_04_2022_prop <- c(
  angry <-  sum(i$angry),
  disgust <- sum(i$disgust),
  fear <- sum(i$fear),
  happy <- sum(i$happy),
  sad <- sum(i$sad),
  surprise <- sum(i$surprise),
  meutral <- sum(i$neutral)
)
```

```
#2
# Renzi_30_03_2022
```

```
i = Renzi_30_03_2022
Renzi_30_03_2022_prop <- c(
  angry <-  sum(i$angry),
  disgust <- sum(i$disgust),
  fear <- sum(i$fear),
  happy <- sum(i$happy),
  sad <- sum(i$sad),
  surprise <- sum(i$surprise),
  meutral <- sum(i$neutral)
)
```

```
renzi <- rbind(Renzi_19_04_2022_prop,
               Renzi_30_03_2022_prop
               )
colnames(renzi) <- emo_label

renzi <- as.data.frame(renzi)

tot_renzi <- max(Renzi_19_04_2022$...1) +
              max(Renzi_30_03_2022$...1)

renzi[3,] <- c(sum(renzi$angry)/tot_renzi * 100,
               sum(renzi$disgust)/tot_renzi * 100,
               sum(renzi$fear)/tot_renzi * 100,
               sum(renzi$happy)/tot_renzi * 100,
               sum(renzi$sad)/tot_renzi * 100,
               sum(renzi$surprise)/tot_renzi * 100,
               sum(renzi$neutral)/tot_renzi * 100)
```

6.6 Salvini datasets

```
#1
# Salvini_08_03_2022
i = Salvini_08_03_2022
Salvini_08_03_2022_prop <- c(
  angry <-  sum(i$angry),
  disgust <- sum(i$disgust),
  fear <- sum(i$fear),
  happy <- sum(i$happy),
  sad <- sum(i$sad),
  surprise <- sum(i$surprise),
  meutral <- sum(i$neutral)
)
```

```
#2
# Salvini_08_04_2022_02
i = Salvini_08_04_2022_02
Salvini_08_04_2022_02_prop <- c(
  angry <-  sum(i$angry),
  disgust <- sum(i$disgust),
  fear <- sum(i$fear),
  happy <- sum(i$happy),
  sad <- sum(i$sad),
  surprise <- sum(i$surprise),
  meutral <- sum(i$neutral)
)
```

```
#3
# Salvini_16_03_2022
i = Salvini_16_03_2022
Salvini_16_03_2022_prop <- c(
  angry <-  sum(i$angry),
  disgust <- sum(i$disgust),
  fear <- sum(i$fear),
  happy <- sum(i$happy),
  sad <- sum(i$sad),
  surprise <- sum(i$surprise),
  meutral <- sum(i$neutral)
)
```

```
salvini <- rbind(Salvini_08_03_2022_prop,
               Salvini_08_04_2022_02_prop,
               Salvini_16_03_2022_prop
               )
colnames(salvini) <- emo_label

salvini <- as.data.frame(salvini)

tot_salvini <- max(Salvini_08_03_2022$...1) +
                max(Salvini_08_04_2022_02$...1)+
                max(Salvini_16_03_2022$...1)


salvini[4,] <- c(sum(salvini$angry)/tot_salvini * 100,
                 sum(salvini$disgust)/tot_salvini * 100,
               sum(salvini$fear)/tot_salvini * 100,
               sum(salvini$happy)/tot_salvini * 100,
               sum(salvini$sad)/tot_salvini * 100,
```

```
            sum(salvini$surprise)/tot_salvini * 100,
            sum(salvini$neutral)/tot_salvini * 100)
```

6.7 Create dataset with the proportion of the emotions registered for each leader

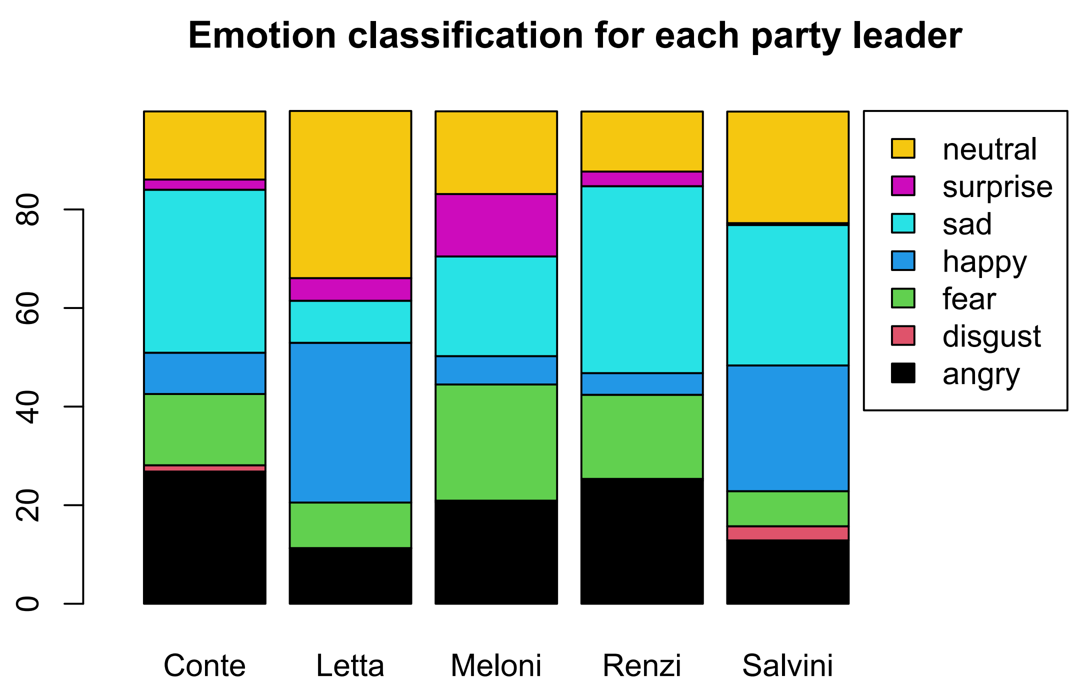
```
emotions <- rbind(conte[8,], letta[3,], meloni[8,],
                  renzi [3,], salvini[4,])

emotions <- as.data.frame(emotions)

rownames(emotions) <- c("Conte", "Letta","Meloni", "Renzi", "Salvini")

kable(emotions)
```

	angry	disgust	fear	happy	sad	surprise	neutral
Conte	26.82598	1.2745357	14.453157	8.376083	33.067107	2.0669418	13.82099
Letta	11.26310	0.0342991	9.236472	32.409459	8.532808	4.5901150	33.91223
Meloni	20.63115	0.2883622	23.570927	5.747407	20.226604	12.6588713	16.78353
Renzi	24.98139	0.3537467	17.060469	4.398206	37.923233	2.9666313	12.18218
Salvini	12.85566	2.8709759	7.116432	25.509843	28.479119	0.3918993	22.63165

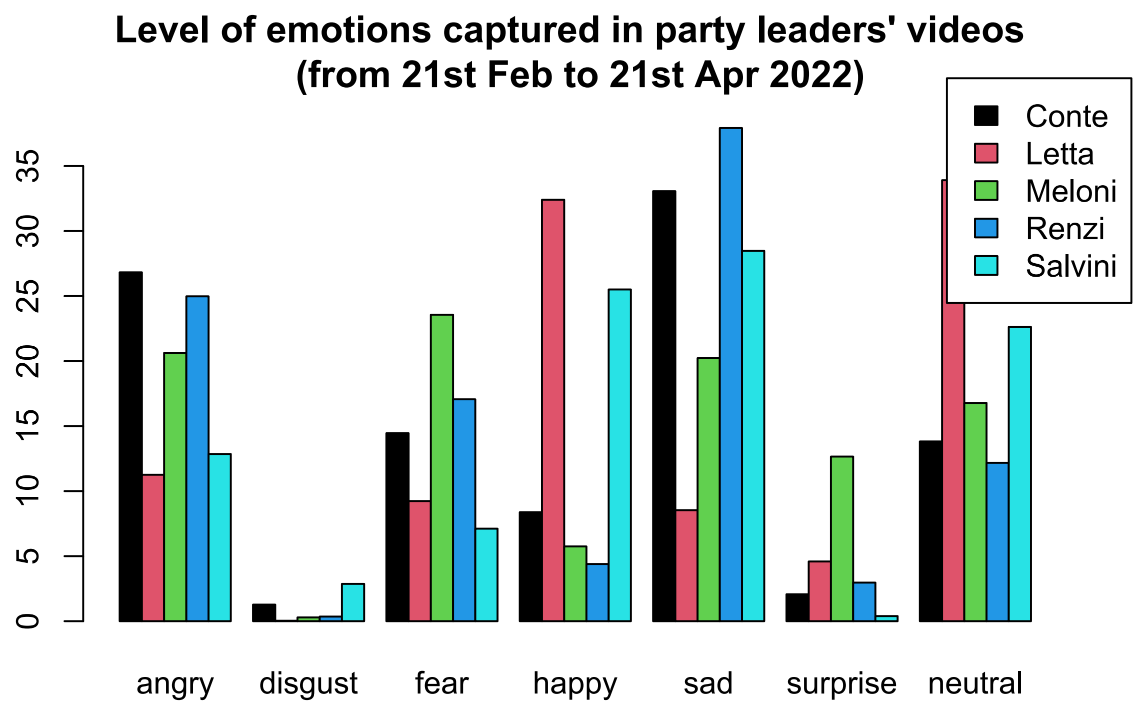
6.8 Results

```
matrix <- as.matrix(emotions)



barplot(t(matrix),
        col = 1:ncol(matrix),
```

```
        legend.text = TRUE,
        args.legend = list(x = "topright"),
         xlim=c(0,7.5),
        main = "Emotion classification for each party leader"
        )
```




```
barplot(matrix,
        col = 1:nrow(matrix),
        legend.text = TRUE,
        args.legend = list(x = "topright",
                           inset = c(- 0.065, -0.1)),
        beside = T,
        main = "Level of emotions captured in party leaders' videos \n (from 21st Feb
```


Level of emotions captured in party leaders' videos
(from 21st Feb to 21st Apr 2022)
Conte
Letta
Meloni
Renzi
Salvini
0
5
10
15
20
25
30
35
angry
disgust
fear
happy
sad
surprise
neutral

.

Un ringraziamento speciale va a tutto il team del progetto POPULITE e in particolare alle Professoresse Silvia Decadri e Fedra Negri che con i puntuali consigli e continuo incoraggiamento hanno reso questo lavoro possibile e a Vanessa Ferrara il cui sostegno è stato fondamentale nel raggiungimento del traguardo. Per ultimo ma di certo non meno importante a Gaia Rebecchi che mi ha aiutato a superare i momenti di sconforto e non ha mai smesso di motivarmi.