

# Dictionary Analysis

Riccardo Ruta

5/2022

## Contents

<b>Dictionary analysis</b>	<b>1</b>
Create the dictionary . . . . .	1
Decadri_Boussalis_Grundl . . . . .	4
Rooduijn_Pauwels_Italian . . . . .	20
Grundl_Italian_adapted . . . . .	23
Compare the general level of populism in time for the dictionaries . . . . .	26
DA SISTEMARE LA COMPARAZIONE TRA DIZIONARI ! . . . . .	26
Compare how the dictionaries score for the most populist parliamentary group . . . . .	26

## Dictionary analysis

At the level of political parties, which ones make most use of populist rhetoric?

At the level of individual politicians, which ones make most use of populist rhetoric?

I use 3 dictionary to perform the analysis

- Rooduijn & Pauwels: Rooduijn, M., and T. Pauwels. 2011. “Measuring Populism: Comparing Two Methods of Content Analysis.” *West European Politics* 34 (6): 1272–1283.
- Decadri & Boussalis: Decadri, S., & Boussalis, C. (2020). Populism, party membership, and language complexity in the Italian chamber of deputies. *Journal of Elections, Public Opinion and Parties*, 30(4), 484-503.
- Grundl: Gründl J. Populist ideas on social media: A dictionary-based measurement of populist communication. *New Media & Society*. December 2020.
- Decadri & Boussalis + Grundl: this is simply a more extended version of the D&B dictionary, which also contains some terms taken from Grundl.

## 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

## 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

Decadri\_Boussalis\_Grundl

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

Transform the DFM into an ordinary dataframe

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

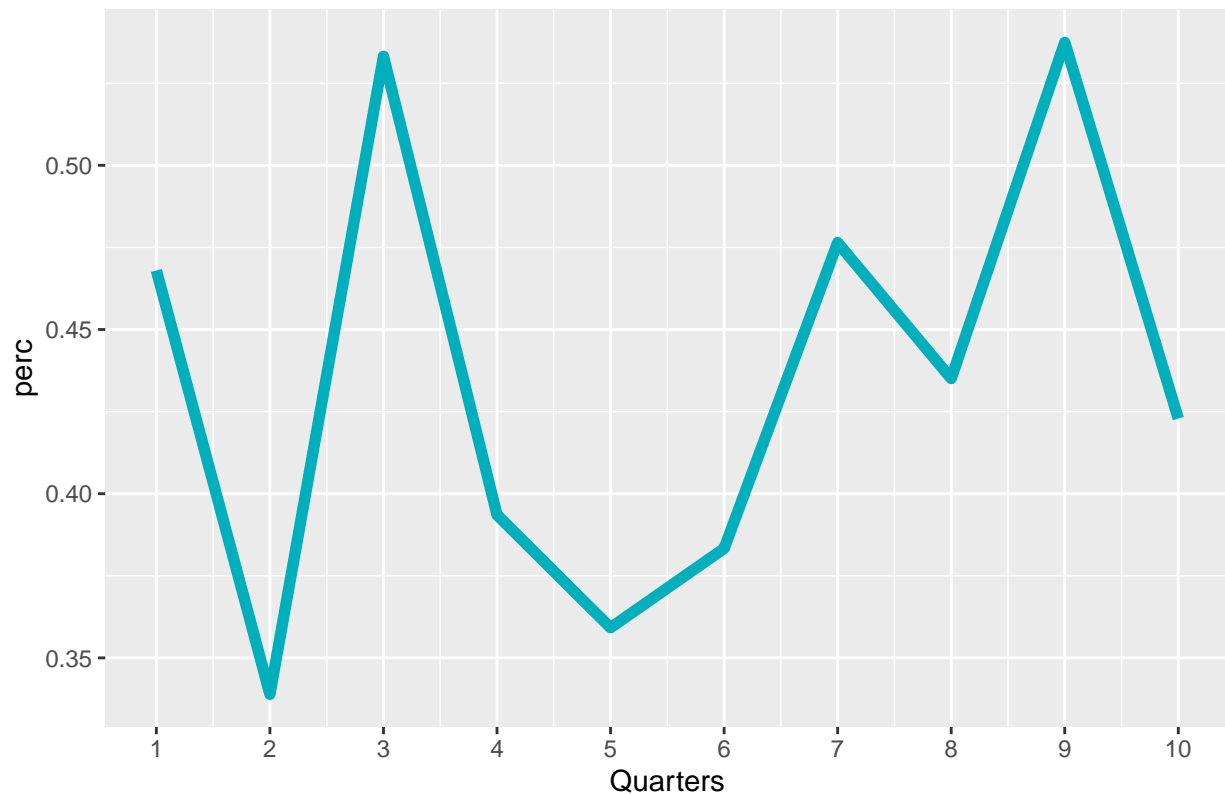
# Add variable with general level of populism
data_dict1 <- data_dict1 %>% mutate(populism = (people + common_will + elite) * 100)
```

Level of populism in time

```
#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_quar
  labs(title = "Populism level over quarters of the 'people' component")
plot_people
```

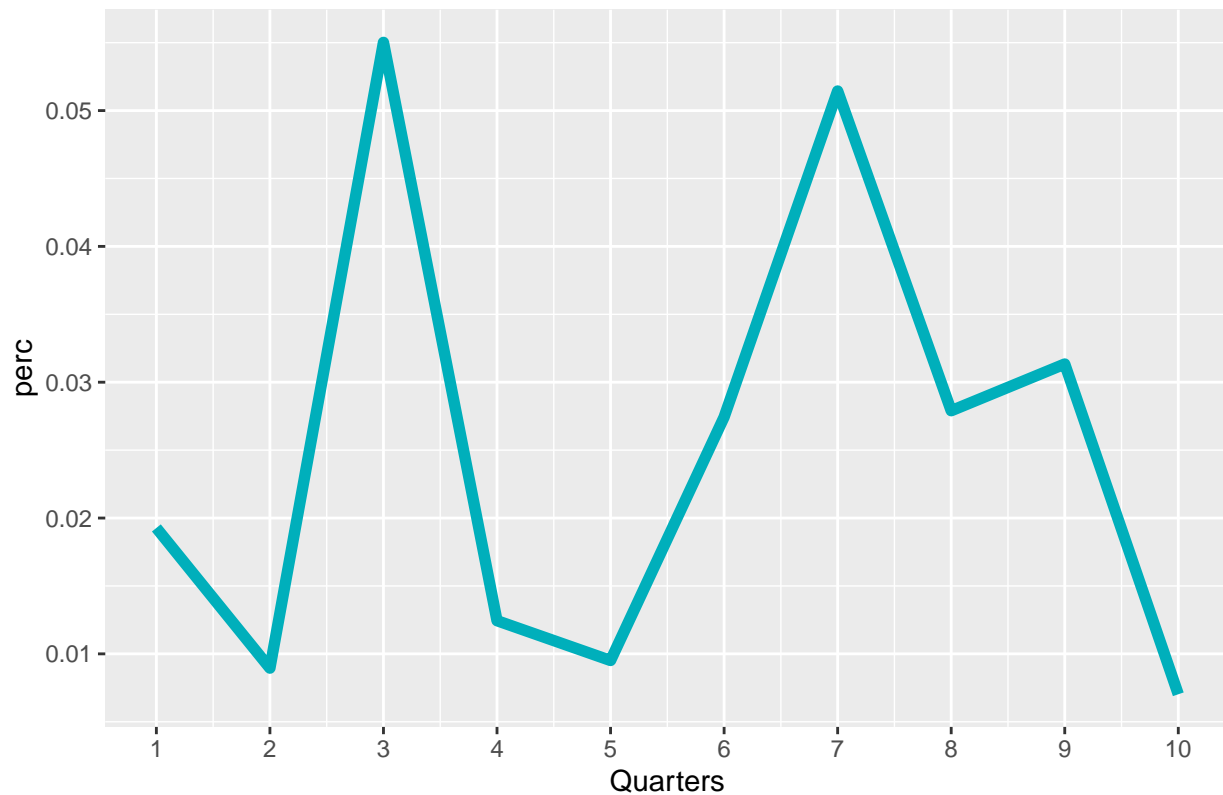
Populism level over quarters of the 'people' component



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

# plot the level of the "common will" component in time
plot_common <- ggplot(data = data_quarter_common, aes(x = Group.1, y = perc))+
  geom_line(color = "#00AFBB", size = 2)+
  scale_x_continuous("Quarters", labels = as.character(data_quarter_common$Group.1), breaks = data_quarter_common$Group.1)
labs(title = "Populism level over quarters of the 'common_will' component")
plot_common
```

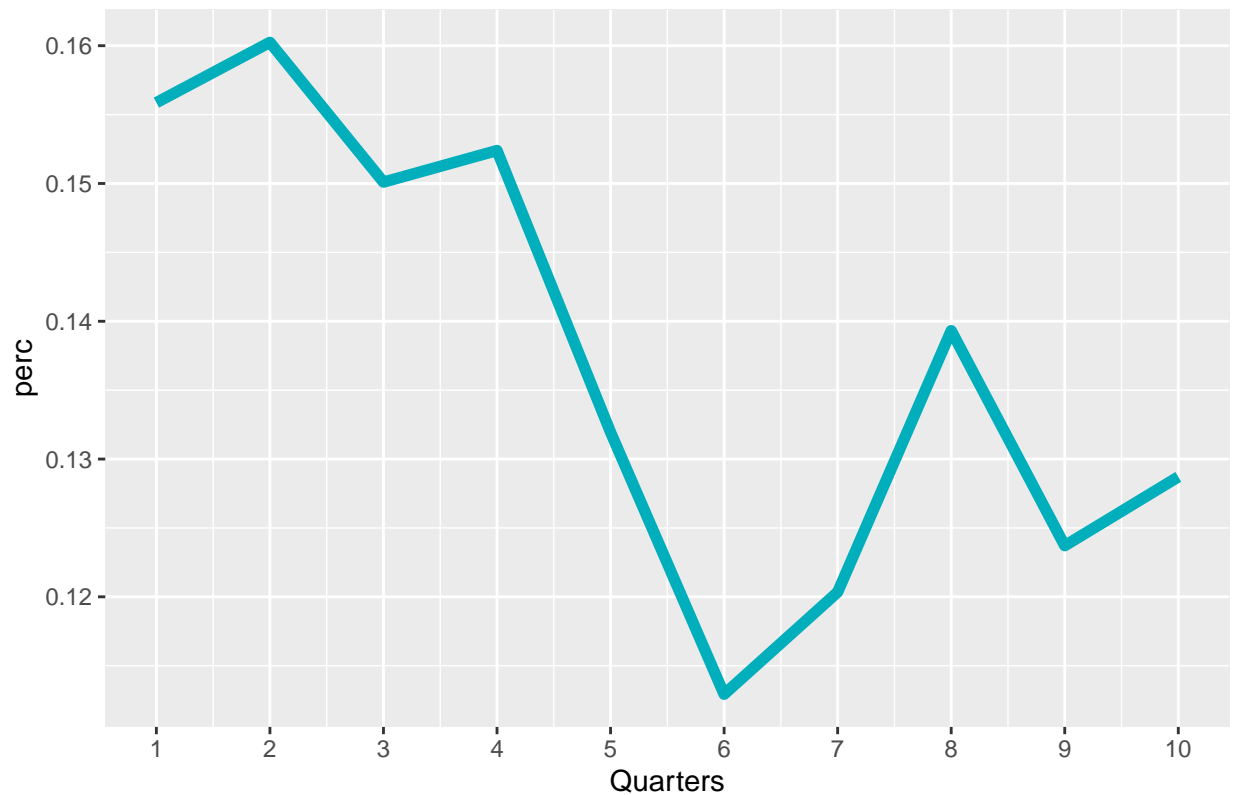
Populism level over quarters of the 'common\_will' component



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

# plot the level of the "ELITE" component in time
plot_elite <- ggplot(data = data_quarter_elite, aes(x = Group.1, y = perc))+
  geom_line(color = "#00AFBB", size = 2)+
  scale_x_continuous("Quarters", labels = as.character(data_quarter_elite$Group.1), breaks = data_quarter_elite$Group.1)
  labs(title = "Populism level over quarters of the 'elite' component")
plot_elite
```

Populism level over quarters of the 'elite' component



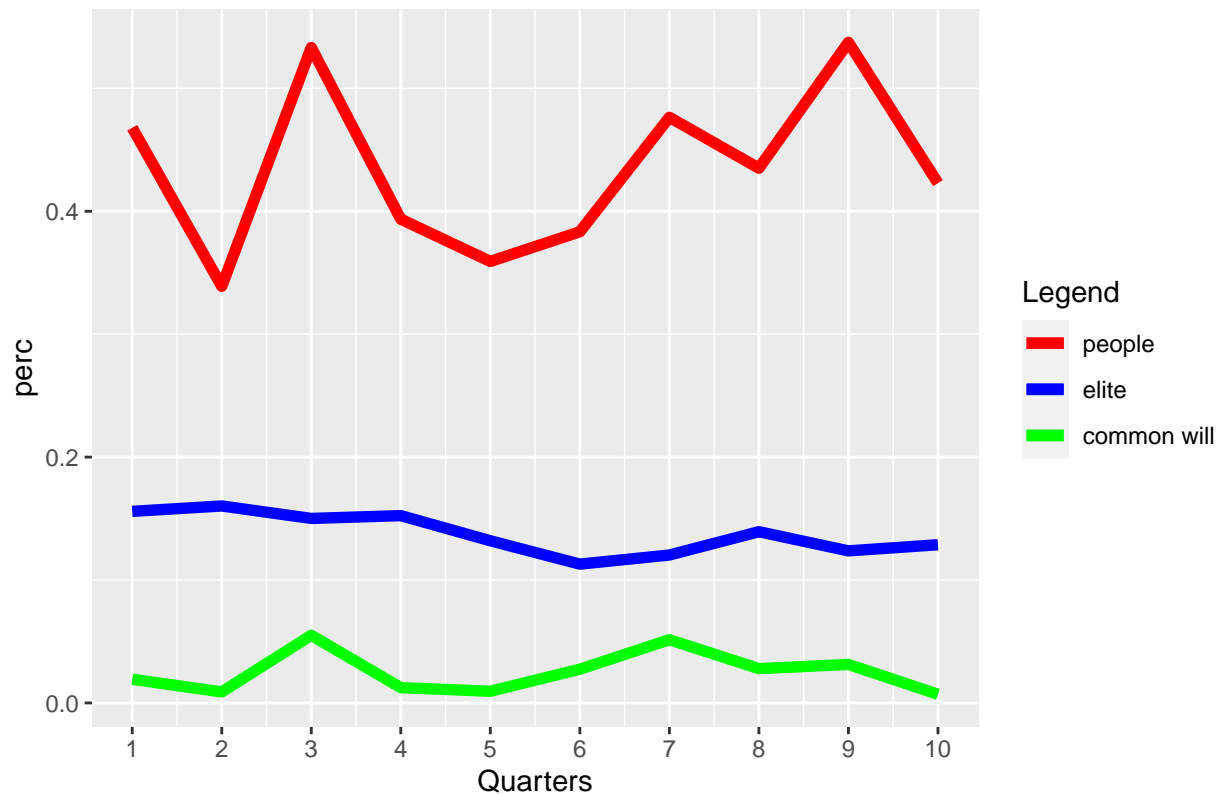
```
#####
# compare the levels
p <- ggplot() +
  # plot people
  geom_line(data = data_quarter_people, aes(x = Group.1, y = perc, color = "people"), size = 2) +
  # plot common will
  geom_line(data = data_quarter_common, aes(x = Group.1, y = perc, color = "common will"), size = 2) +
  # plot elite
  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_quar

  labs(title = " Compare the 3 components of the populism level")

p
```

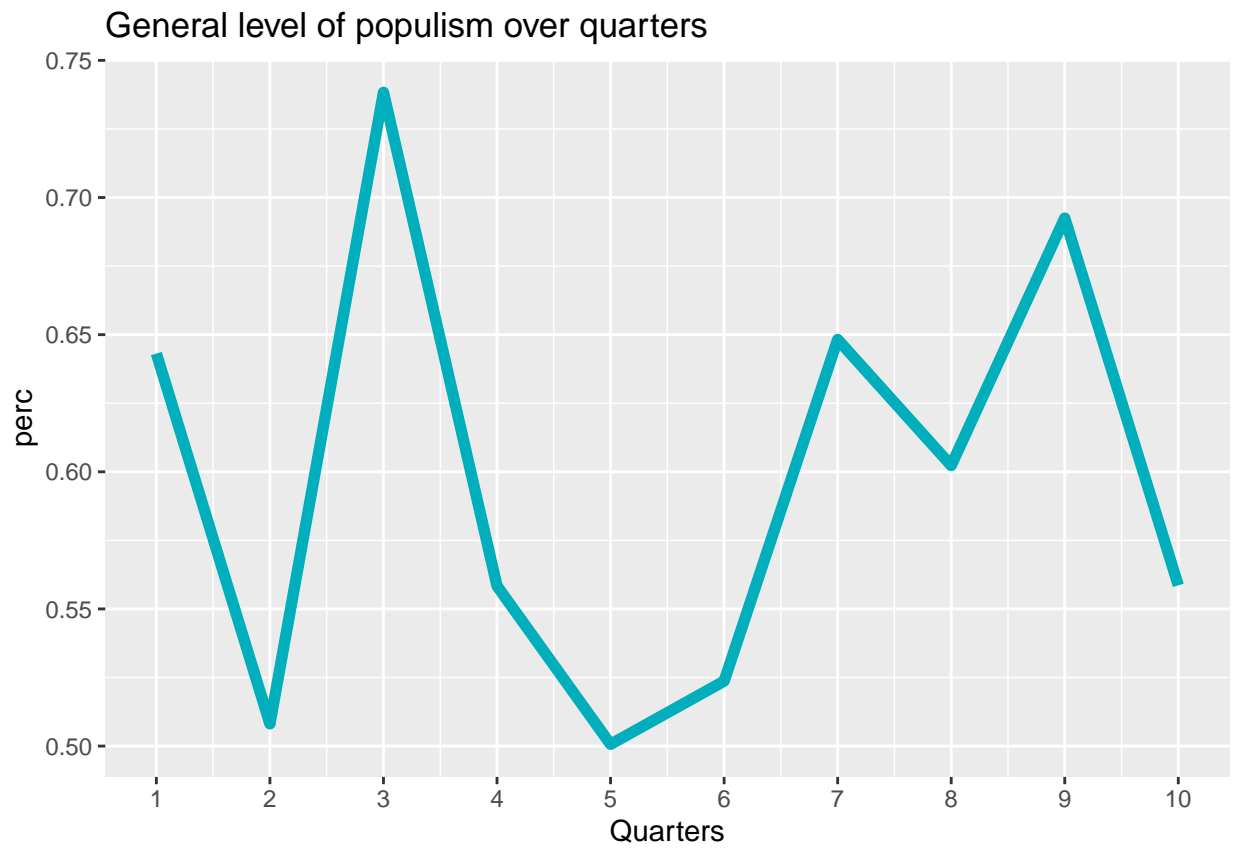
Compare the 3 components of the populism level



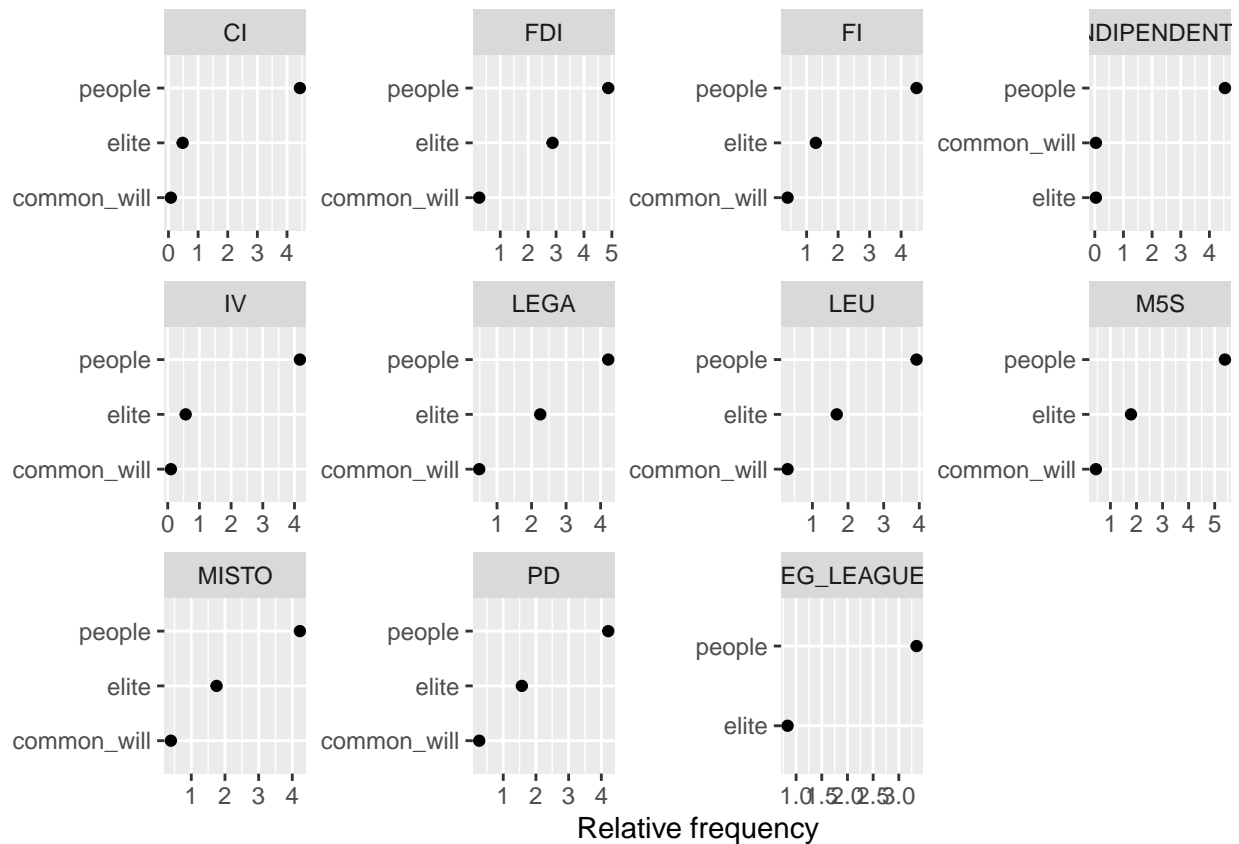
```
#####
#Over time general level populism (quarters)
data_quarter_general <- aggregate(x = data_dict1$populism, # Specify data column
  by = list(data_dict1$quarter), # Specify group indicator
  FUN = mean) # Specify function (i.e. mean)
data_quarter_general$perc <- data_quarter_general$x

# 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)
labs(title = "General level of populism over quarters")
plot_general
```





## Main component for each parliamentary group



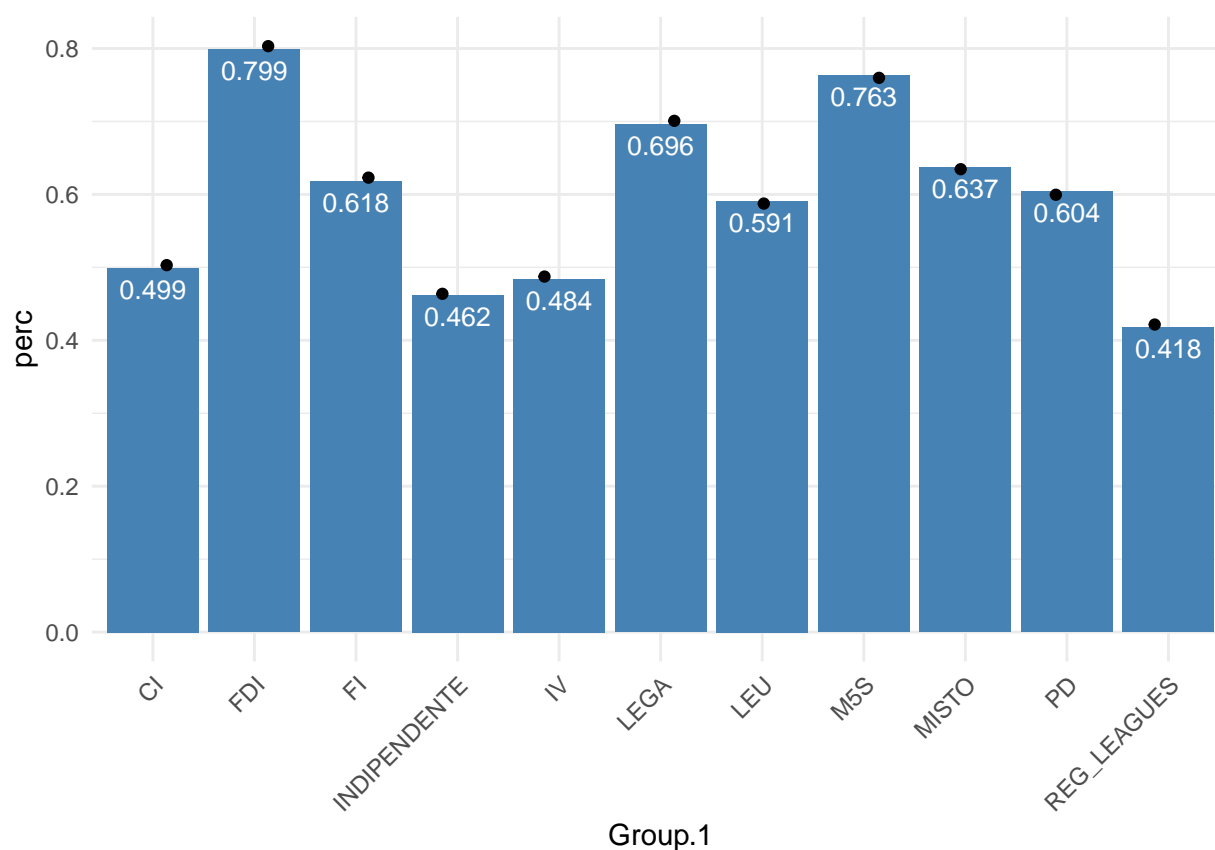
## Most populist parliamentary group

```
#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)
kable(data_party %>% select(Group.1, perc) %>% arrange(desc(perc)))
```

Group.1	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

```
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))
```



```
labs(title = "LEVEL OF POPULISM")
```

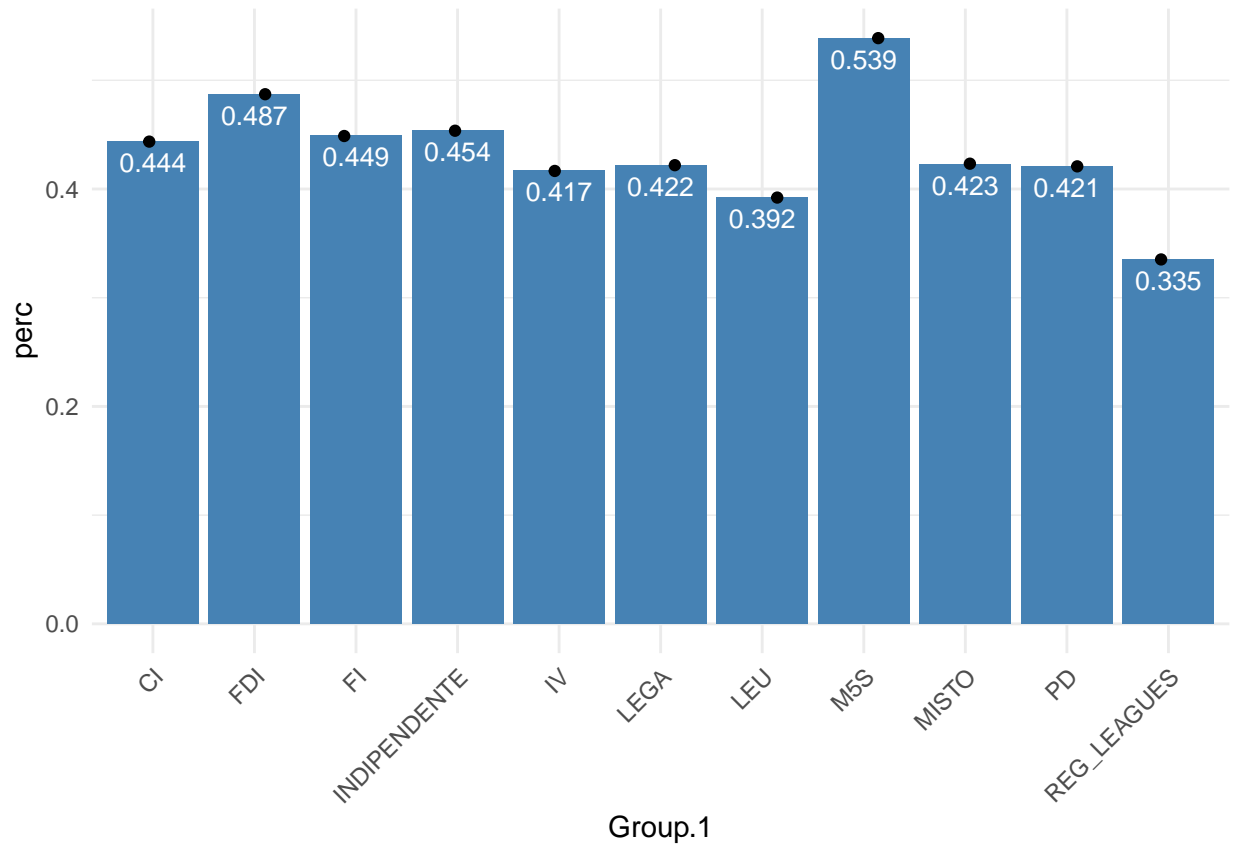
```
## $title
## [1] "LEVEL OF POPULISM"
##
```

```
## attr("class")
## [1] "labels"
```

```
# PEOPLE
data_party_people <- aggregate(x = data_dict1$people, # Specify data column
                               by = list(data_dict1$party_id), # Specify group indicator
                               FUN = mean) # Specify function (i.e. mean)
data_party_people$perc <- round(data_party_people$x * 100,3)
kable(data_party_people %>% select(Group.1, perc)%>% arrange(desc(perc)))
```

Group.1	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

```
ggplot(data=data_party_people, 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))
```



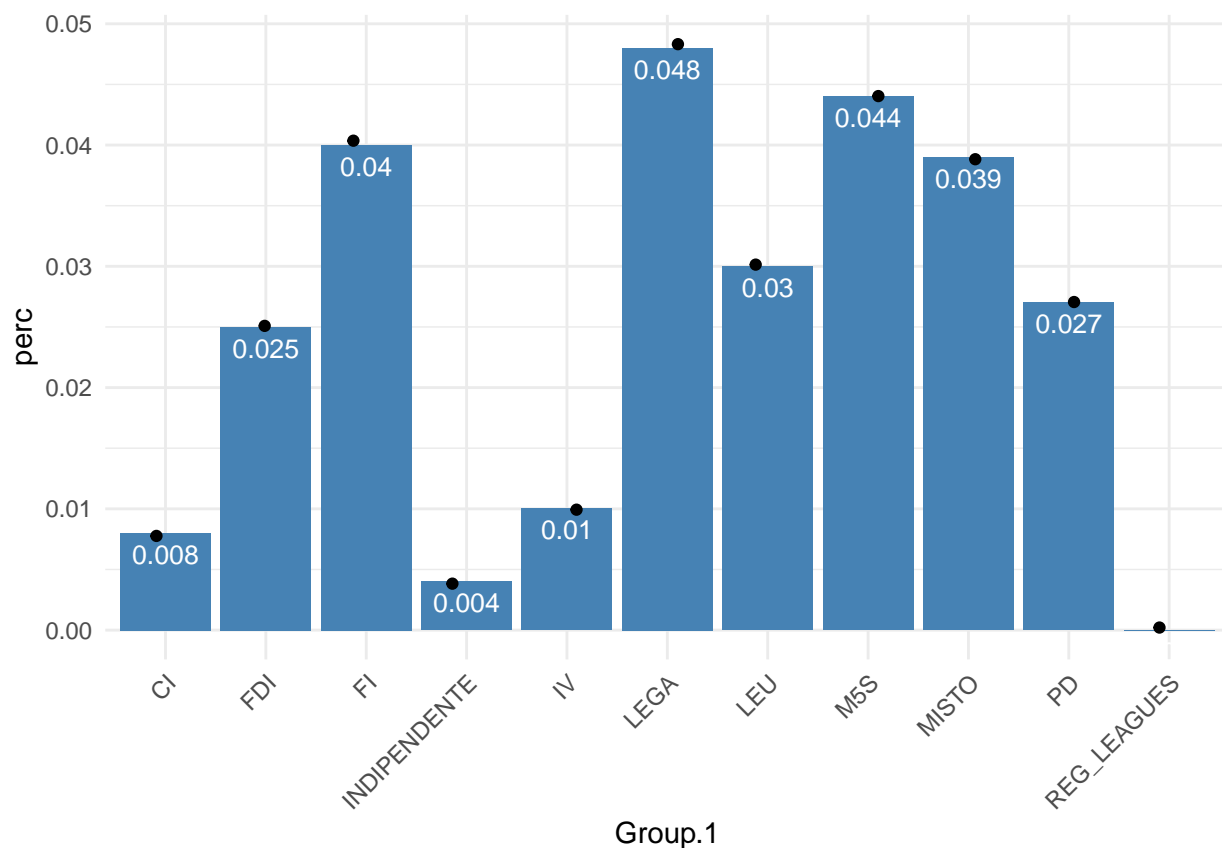
```
labs(title = "LEVEL OF POPULISM: PEOPLE COMPONENT")
```

```
## $title
## [1] "LEVEL OF POPULISM: PEOPLE COMPONENT"
##
## attr("class")
## [1] "labels"
```

```
# COMMON WILL
data_party_common <- aggregate(x = data_dict1$common_will, # Specify data column
                              by = list(data_dict1$party_id), # Specify group indicator
                              FUN = mean) # Specify function (i.e. mean)
data_party_common$perc <- round(data_party_common$x * 100, 3)
kable(data_party_common %>% select(Group.1, perc)%>% arrange(desc(perc)))
```

Group.1	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

```
ggplot(data=data_party_common, 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))
```



```
labs(title = "LEVEL OF POPULISM: COMMON WILL COMPONENT")
```

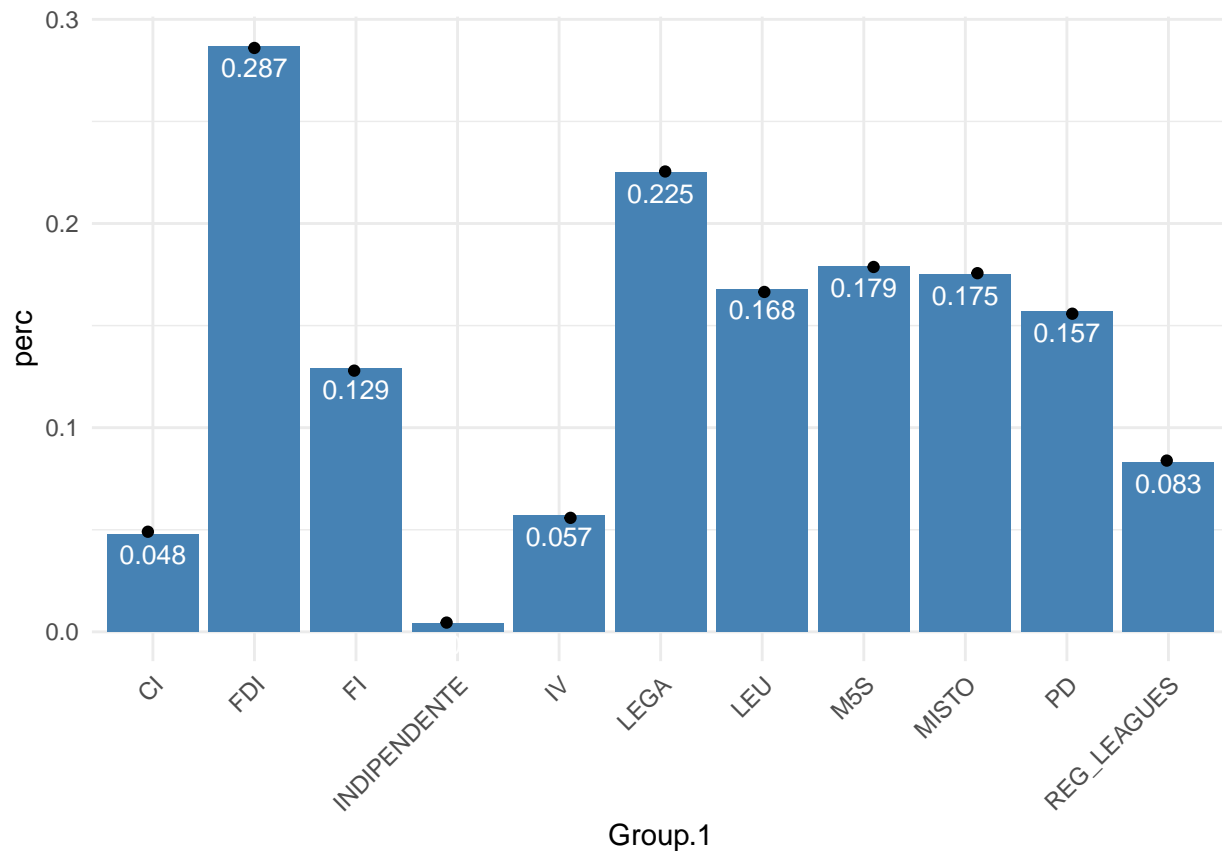
```
## $title
## [1] "LEVEL OF POPULISM: COMMON WILL COMPONENT"
##
```

```
## attr("class")
## [1] "labels"
```

```
# ELITE
data_party_elite <- aggregate(x = data_dict1$elite, # Specify data column
                             by = list(data_dict1$party_id), # Specify group indicator
                             FUN = mean) # Specify function (i.e. mean)
data_party_elite$perc <- round(data_party_elite$x * 100,3)
kable(data_party_elite %>% select(Group.1, perc)%>% arrange(desc(perc)))
```

Group.1	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

```
ggplot(data=data_party_elite, 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))
```



```
labs(title = "LEVEL OF POPULISM: ELITE COMPONENT")
```

```
## $title
## [1] "LEVEL OF POPULISM: ELITE COMPONENT"
##
## attr(,"class")
## [1] "labels"
```

Are the average values of populism for each party statistically different from each other? The reference category is PD

```
# 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
```

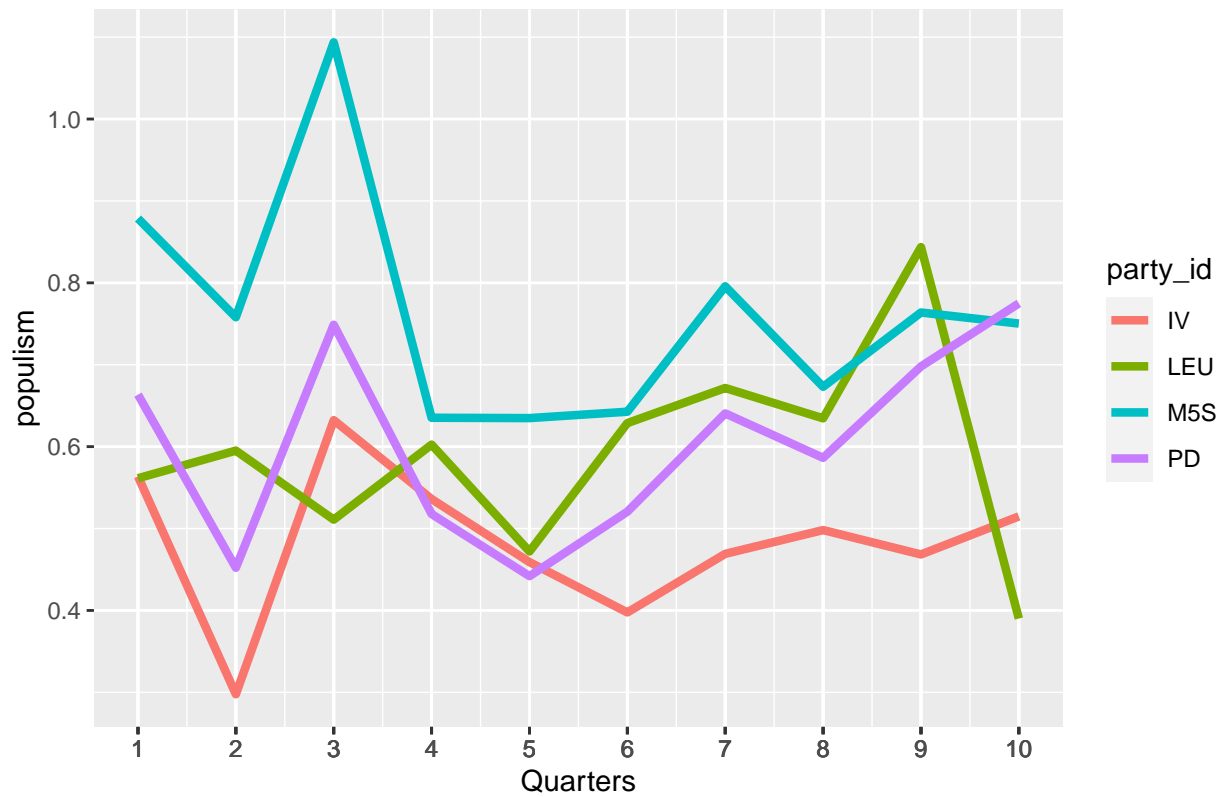
## Trends in the level of populism for each parliamentary group over time

```
#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")

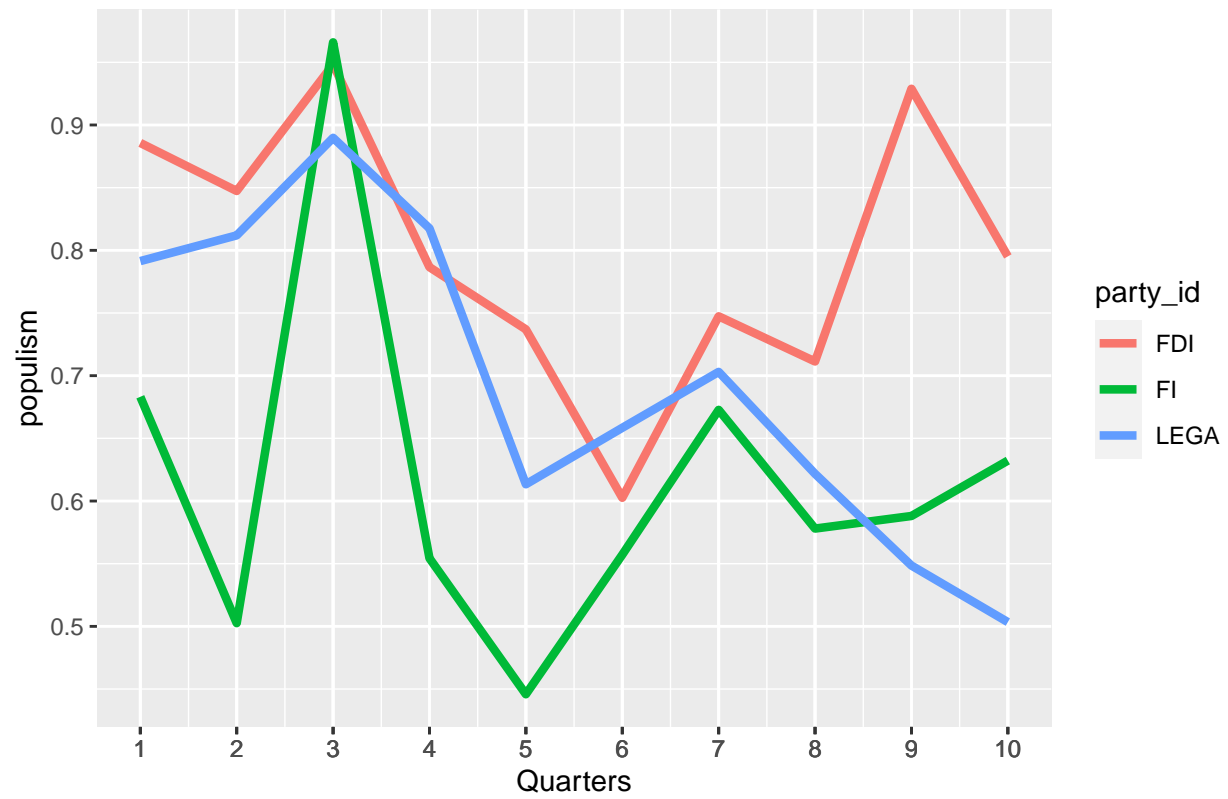
# Left parties in time
ggplot(left_party, aes(x=quarter, y=populism, color=party_id)) +
  geom_line(size=1.5)+
  scale_x_continuous("Quarters", labels = as.character(left_party$quarter), breaks = left_party$quarter)
  ggtitle("Level of populism in the quarters for left-wing parties")
```

Level of populism in the quarters for left-wing parties



```
# 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)
ggtitle("Level of populism in the quarters for right-wing parties")
```

Level of populism in the quarters for right-wing parties



## 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))

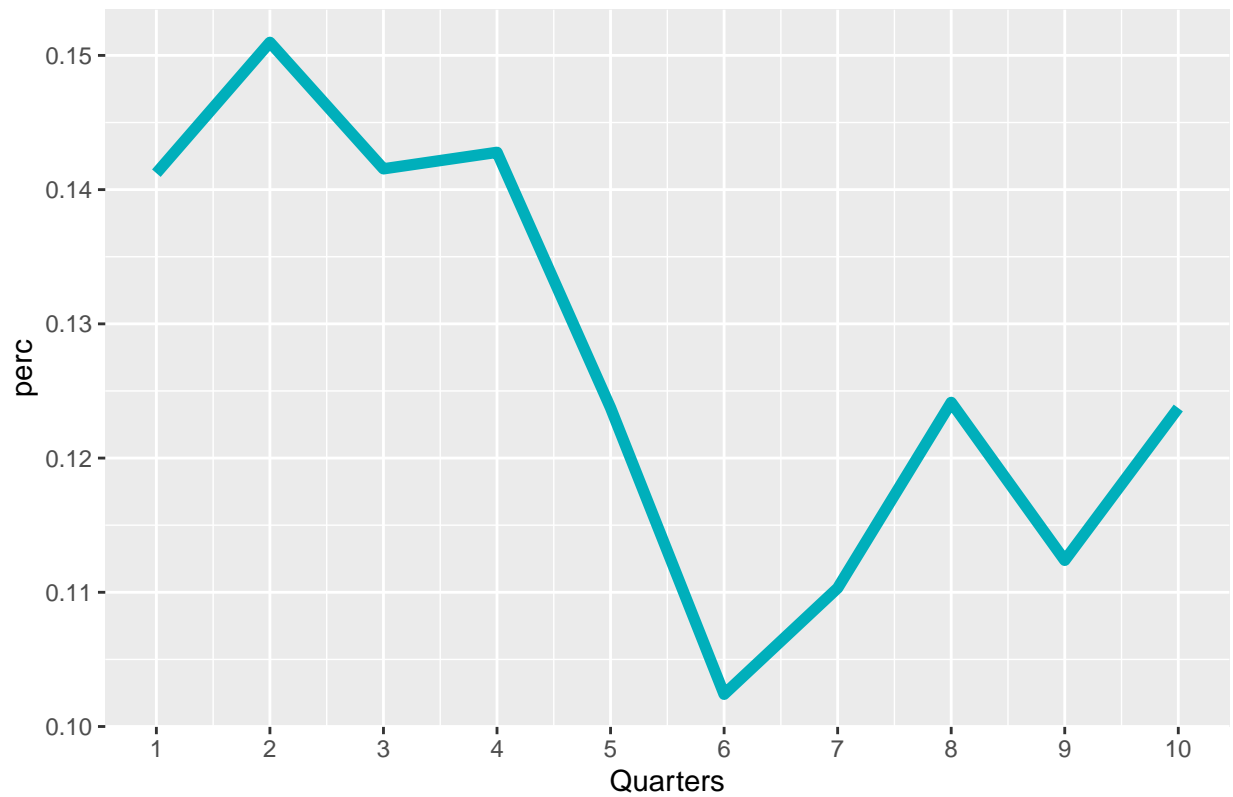
# Add variable with general level of populism
#data_dict2 <- data_dict2 %>% mutate(populism = (people + common_will + elite) * 100)
```

## Level of populism in 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$Group.1), breaks = data_quarter_general2$Group.1)
  labs(title = "General level of populism over quarters")
plot_general2
```

General level of populism over quarters

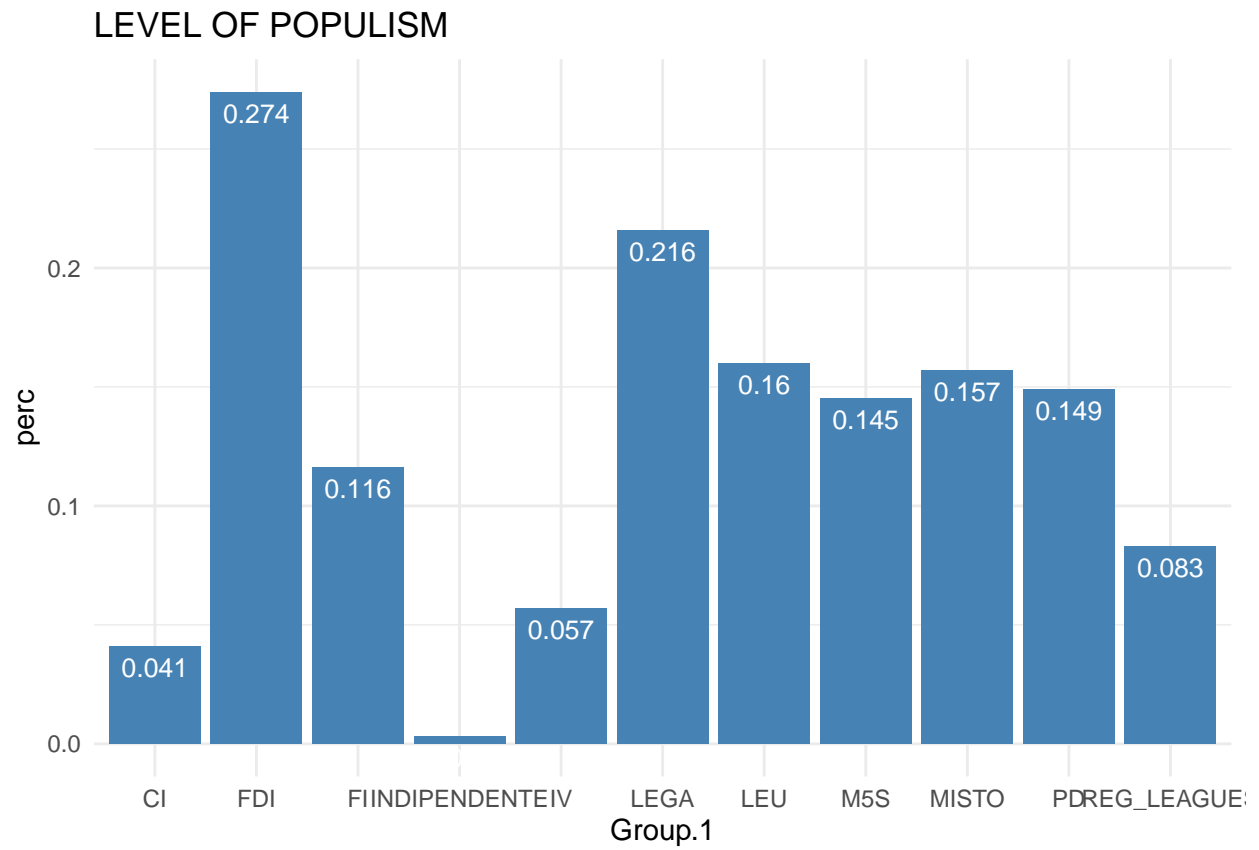


Most populist parliamentary group

```
# 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()+
  labs(title = "LEVEL OF POPULISM")
```



## 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))

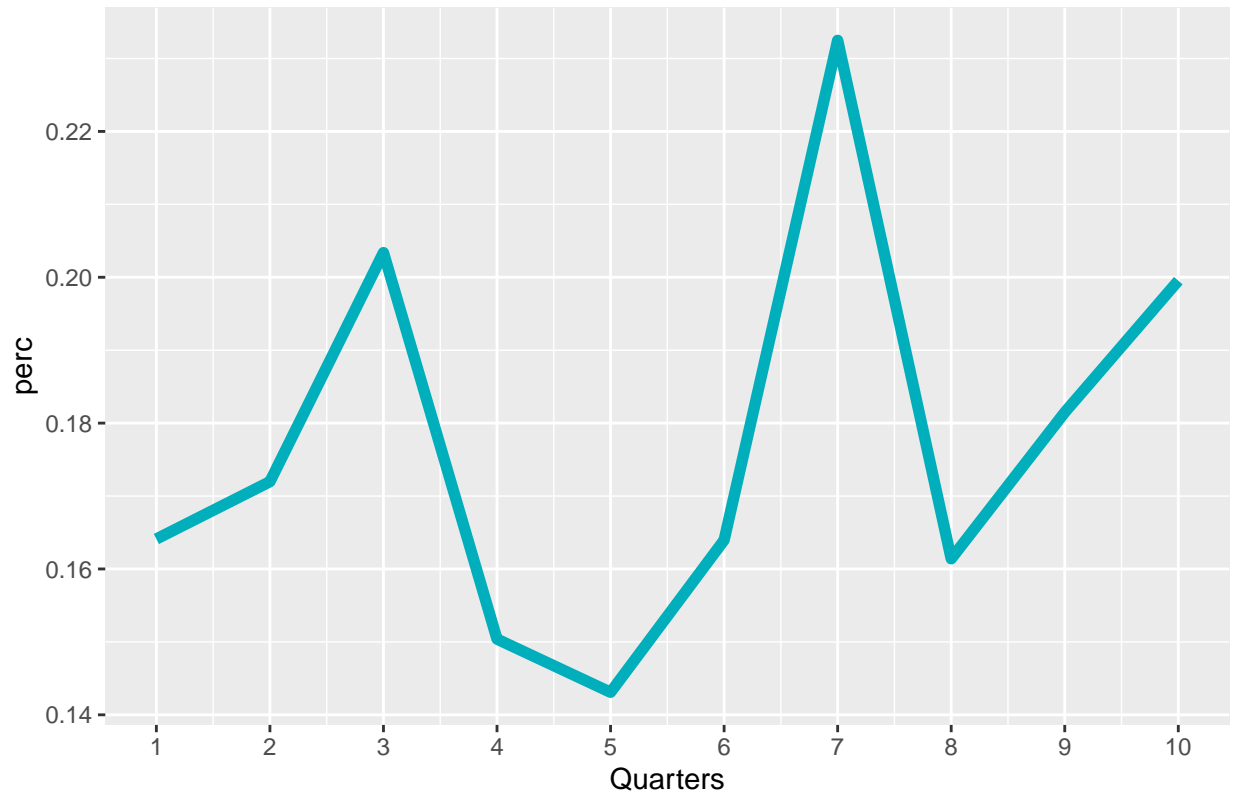
# Add variable with general level of populism
#data_dict2 <- data_dict2 %>% mutate(populism = (people + common_will + elite) * 100)
```

## 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$Group.1), breaks = data_quarter_general3$Group.1)
  labs(title = "General level of populism over quarters")
plot_general3
```

### General level of populism over quarters



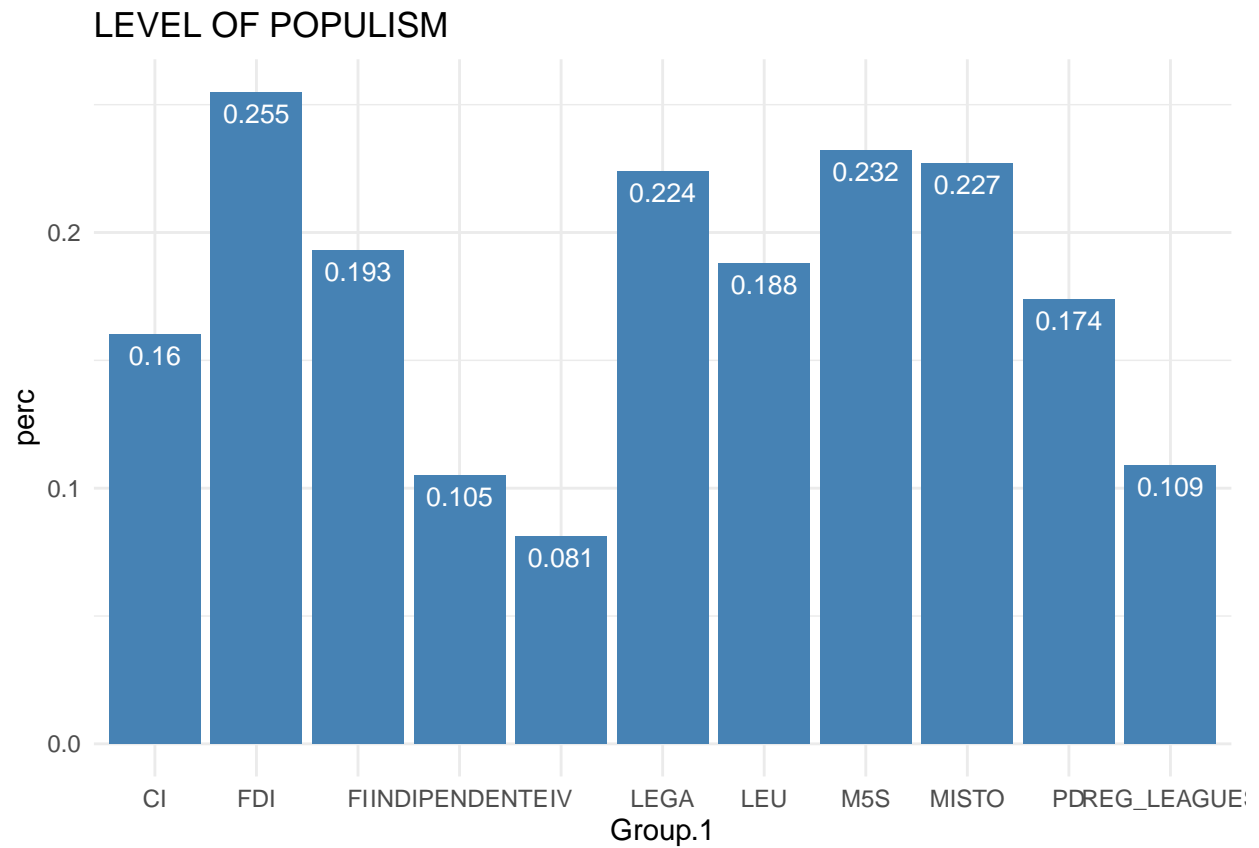
### Most populist parliamentary group

```
# 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

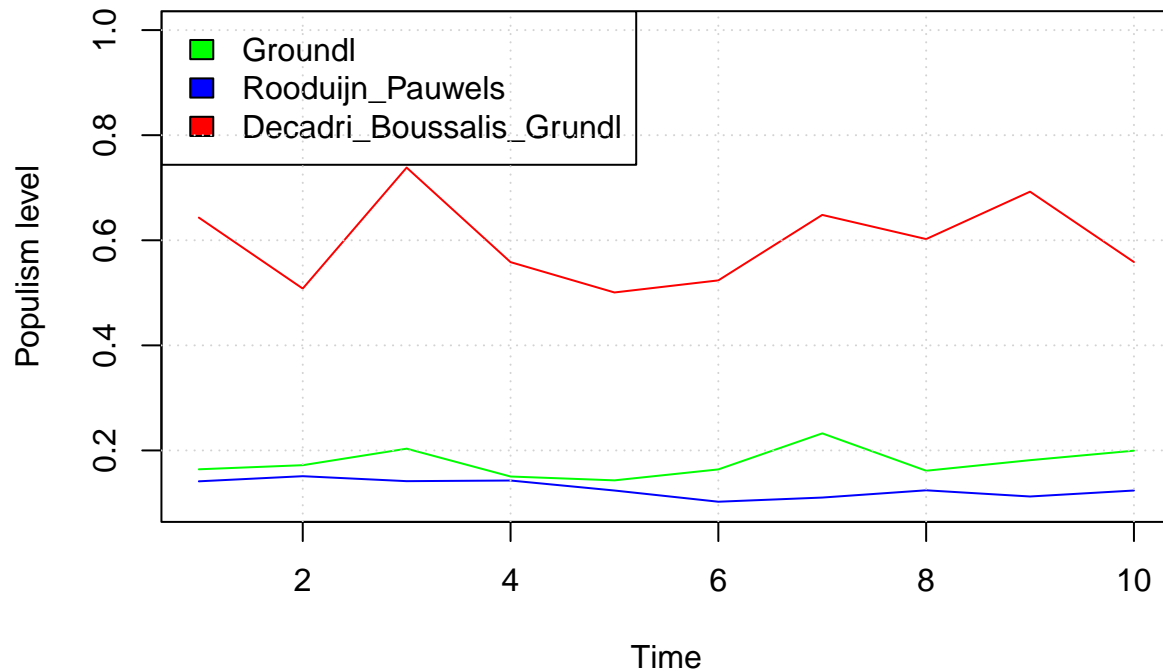


```
ggplot(data=data_party3, aes(x=Group.1, y=perc)) +
  geom_bar(stat="identity", fill="steelblue")+
  geom_text(aes(label=perc), vjust=1.6, color="white", size=3.5)+
  theme_minimal()+
  labs(title = "LEVEL OF POPULISM")
```



Compare the general level of populism in time for the dictionaries

### Compare how the different dictionaries score



DA SISTEMARE LA COMPARAZIONE TRA DIZIONARI !

Compare how the dictionaries score for the most populist parliamentary group

```
# Create the columns with the "populist score"
# 11 for the "most populist" and 1 for the least
dfm_dict1_tstat_party_filtered$my_rank <- rank(dfm_dict1_tstat_party_filtered$populism)
dfm_dict2_tstat_party$my_rank <- rank(dfm_dict2_tstat_party$frequency)
dict_3_tstat_party$my_rank <- rank(dict_3_tstat_party$frequency)
dict_4_tstat_party$my_rank <- rank(dict_4_tstat_party$frequency)

# define the parliamentary group list
party <- c("LEGA", "PD", "M5S", "FI", "FDI", "MISTO",
          "LEU", "CI", "IV", "INDIPENDENTE", "REG_LEAGUES")

# create an empty df
party_rank <- data.frame(first = vector(), second = vector(),
                        third = vector(), fourth = vector(), fifth = vector() )

# loop the rank for each parliamentary group
for (i in party)
{
```

```

rank_dict_1 <- (dfm_dict1_tstat_party_filtered %>% filter(group == i ) %>% .$my_rank)
rank_dict_2 <- (dfm_dict2_tstat_party %>% filter(group == i ) %>% .$my_rank)
rank_dict_3 <- (dict_3_tstat_party %>% filter(group == i ) %>% .$my_rank)
rank_dict_4 <- (dict_4_tstat_party %>% filter(group == i ) %>% .$my_rank)

party <- (i)
party_rank <- rbind(party_rank, cbind(party, rank_dict_1, rank_dict_2,
                                     rank_dict_3, rank_dict_4))
}

# change the format of the columns in numeric
party_rank$rank_dict_1 <- as.numeric(party_rank$rank_dict_1)
party_rank$rank_dict_2 <- as.numeric(party_rank$rank_dict_2)
party_rank$rank_dict_3 <- as.numeric(party_rank$rank_dict_3)
party_rank$rank_dict_4 <- as.numeric(party_rank$rank_dict_4)

# Create the column with the sum of the single score
party_rank$total_score <- rowSums(party_rank[, -1])
kable(party_rank %>% arrange(desc(total_score)), col.names = c("Party",
                                                             "Dec_Bous_Grun",
                                                             "Rood_Pau_it",
                                                             "Grun_it",
                                                             "Dec_Bous",
                                                             "Total"))

```