Dict Analysis Explain

Create the dictionary

```r
# 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"
```

```r
# 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 <-
  dictionary(list(populism =
                    dict$Decadri_Boussalis
                  [!is.na(dict$Decadri_Boussalis)]))

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)]))
# Create an extra dictionary combining 3 components into one
d_b_g_Populism <-
  dictionary(list(populism =
                    c(Decadri_Boussalis_Grundl$people,
                      Decadri_Boussalis_Grundl$common_will,
                      Decadri_Boussalis_Grundl$elite)))
```

```
dictionaries <- c("Rooduijn_Pauwels_Italian", "Grundl_Italian_adapted",
                  "Decadri_Boussalis","Decadri_Boussalis_Grundl","d_b_g_Populism")
n.words <- c(length(Rooduijn_Pauwels_Italian$populism),
           length(Grundl_Italian_adapted$populism),
           length(Decadri_Boussalis$populism),
           (length(Decadri_Boussalis_Grundl$people)+
              length(Decadri_Boussalis_Grundl$common_will)+
              length(Decadri_Boussalis_Grundl$elite)),
           length(d_b_g_Populism$populism)
           )
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	25
Decadri_Boussalis_Grundl	77
d_b_g_Populism	77

Load DFM

Group and weight the dfm

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

Decadri_Boussalis_Grundl

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

##Qui trasformo la DFM in un dataset normale, che è alla fine quello che ci serve ora. Nota che devi fare cbind per riagganciare le docvars ai valori della DFM (non c'è bisogno di fare merge perché documenti e docvars della medesima DFM sono nello stesso ordine. . .)

```
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)
```

Ora che hai un dataframe "normale" puoi ragionare in questo modo: la tua unit of analysis ora è partitoxquarter. Quindi ogni volta che vuoi fare un'analisi su una unit più grande puoi fare la media. Qui sotto un esempio con la componente people, ovviamente il ragionamento vale ugualmente per le altre e per populism.

1. Qual è l'andamento nel tempo del livello di populismo, generale e delle sue singole componenti?

Per rispondere, non ci serve il dettaglio dei gruppi, ma solo il tempo.

```r
#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
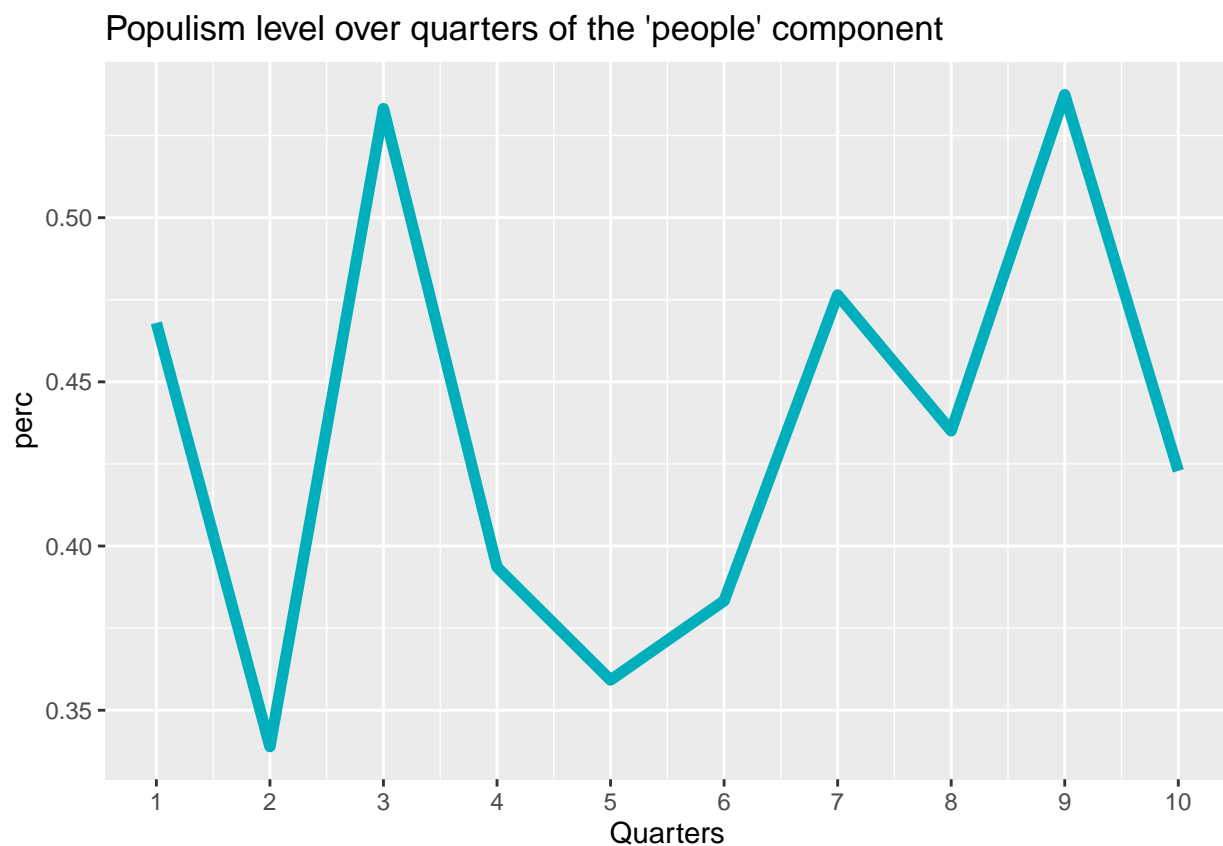
```r
############
#Over time COMMON WILL (quarters)
data_quarter_common <- aggregate(x = data_dict1$common_will,  # Specify data column
          by = list(data_dict1$quarter),  # Specify group indicator
          FUN = mean) # Specify function (i.e. mean)
data_quarter_common$perc <- data_quarter_common$x * 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_quart
  labs(title = "Populism level over quarters of the 'common_will' component")
plot_common
```

Populism level over quarters of the 'common_will' component
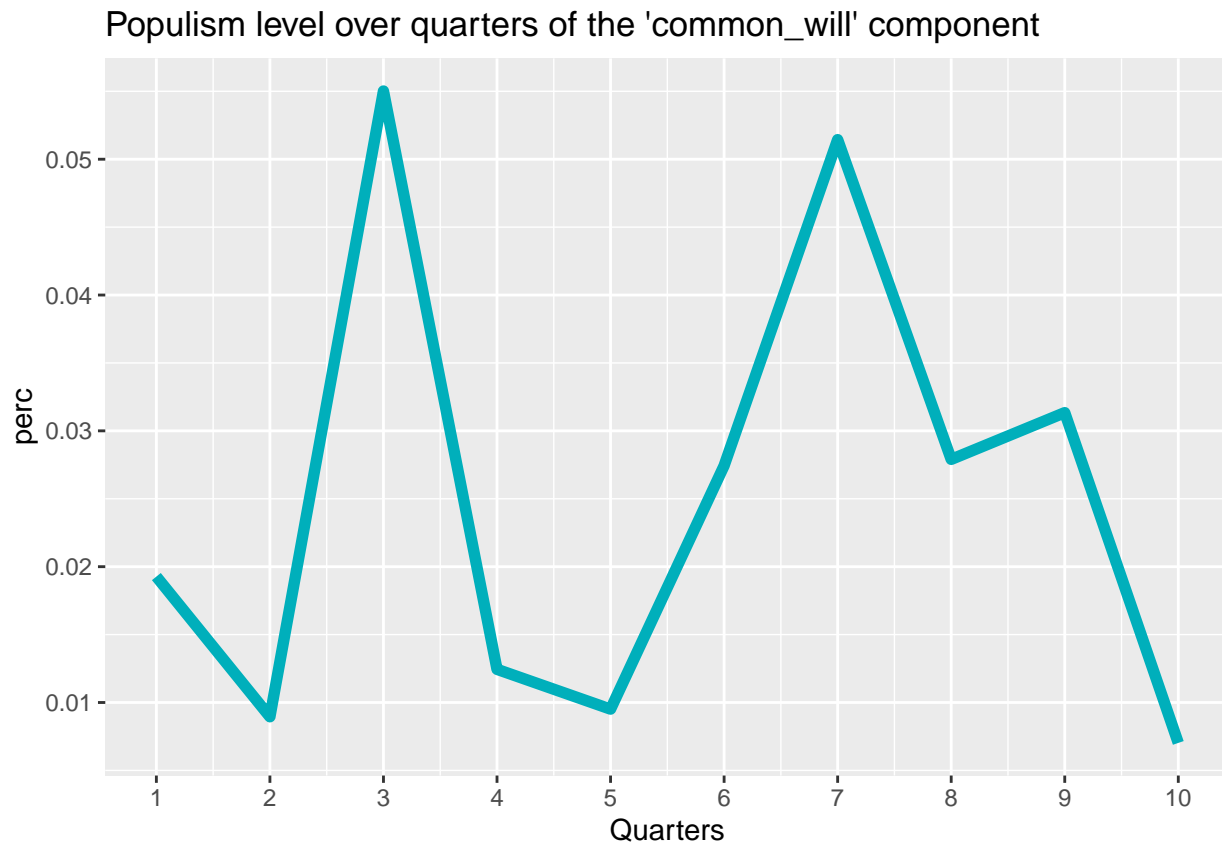

```
############
#Over time ELITE (quarters)
data_quarter_elite <- aggregate(x = data_dict1$elite,  # Specify data column
         by = list(data_dict1$quarter),  # Specify group indicator
         FUN = mean) # Specify function (i.e. mean)
data_quarter_elite$perc <- data_quarter_elite$x * 100

# 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_quart
  labs(title = "Populism level over quarters of the 'elite' component")
plot_elite
```

## Populism level over quarters of the 'elite' component
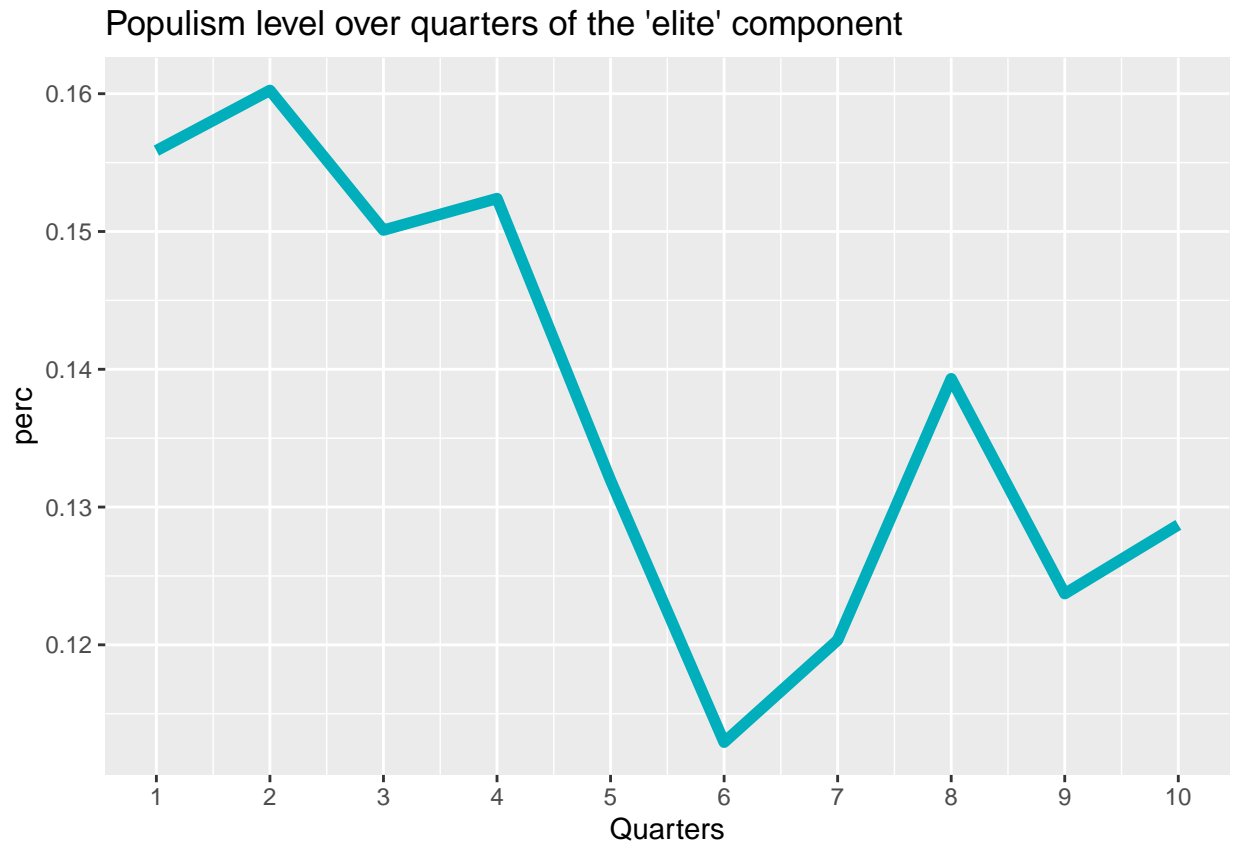

```
#########
# compare the levels
p <- ggplot() +
  # plot people
  geom_line(data = data_quarter_people, aes(x = Group.1, y = perc, color = "people"), size = 2) +
  # plot common will
  geom_line(data = data_quarter_common, aes(x = Group.1, y = perc, color = "common will"), size = 2) +
  # plot elite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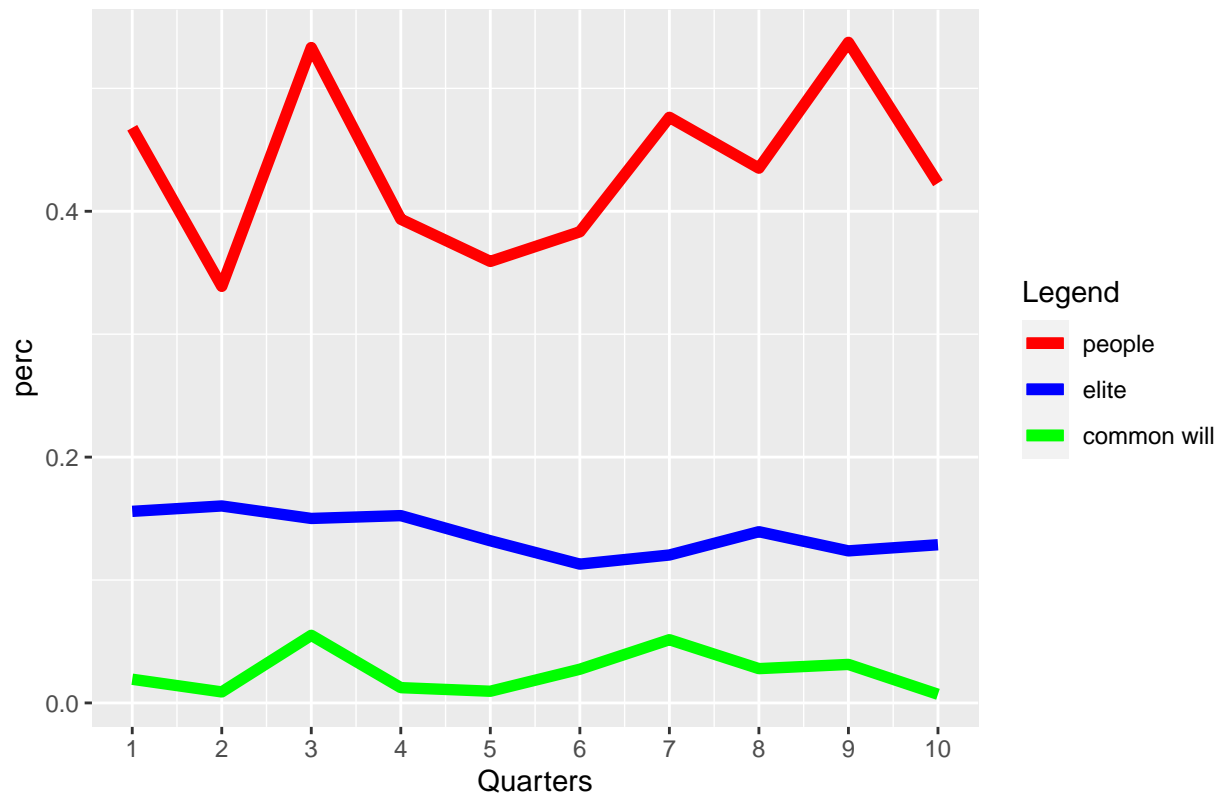

```r
################
#Over time general level populism (quarters)
data_quarter_general <- aggregate(x = data_dict1$populism,  # Specify data column
          by = list(data_dict1$quarter),  # Specify group indicator
          FUN = mean) # Specify function (i.e. mean)
data_quarter_general$perc <- data_quarter_general$x

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

General level of populism over quarters
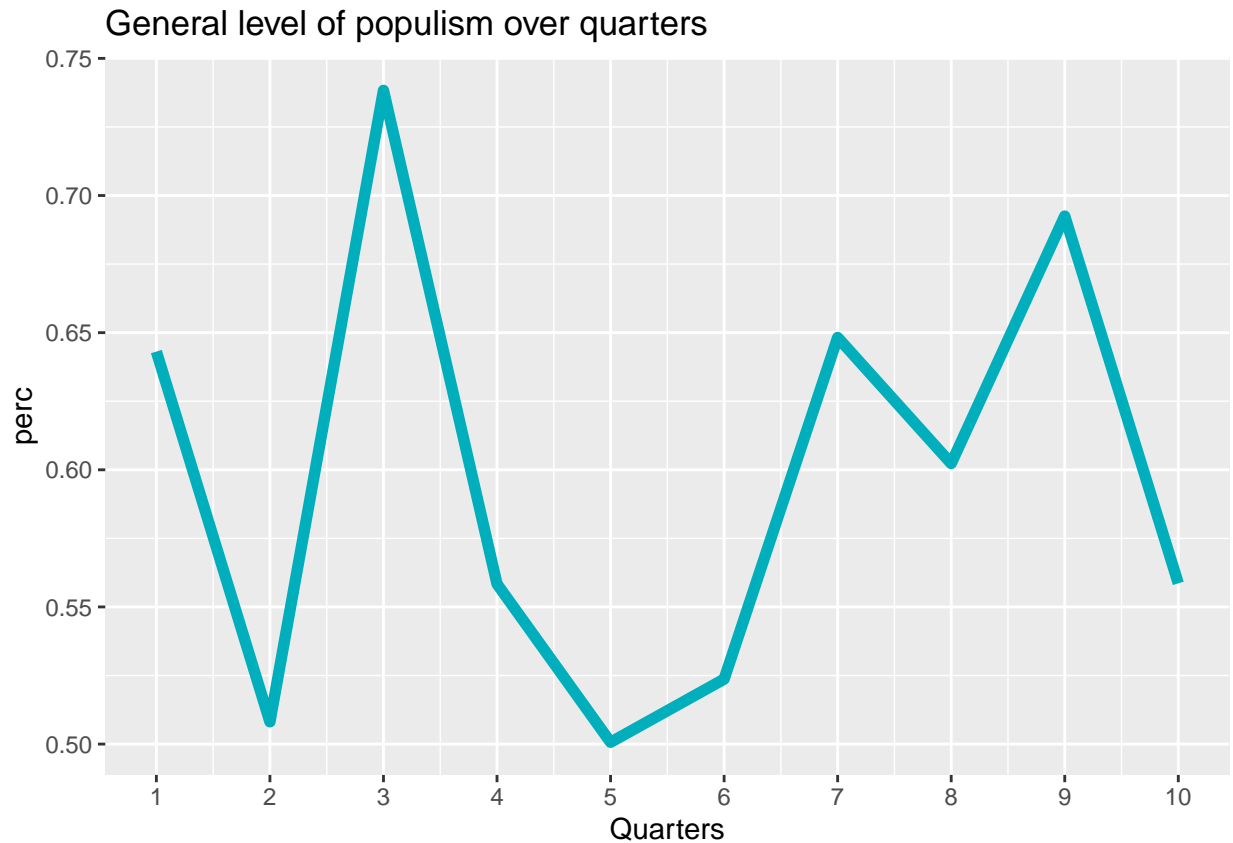
## 2a. Quali sono i gruppi parlamentari più populisti e quale componente di populismo prevale per ciascuno di essi?

Per capire quali sono i più populisti, possiamo fare dei ranking, guardare come si colloca ogni gruppo rispetto alla media/mediana/quintili, stimare dei t-test per la significatività delle differenze nei valori medi tra i diversi partiti.

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


LEVEL OF POPULISM
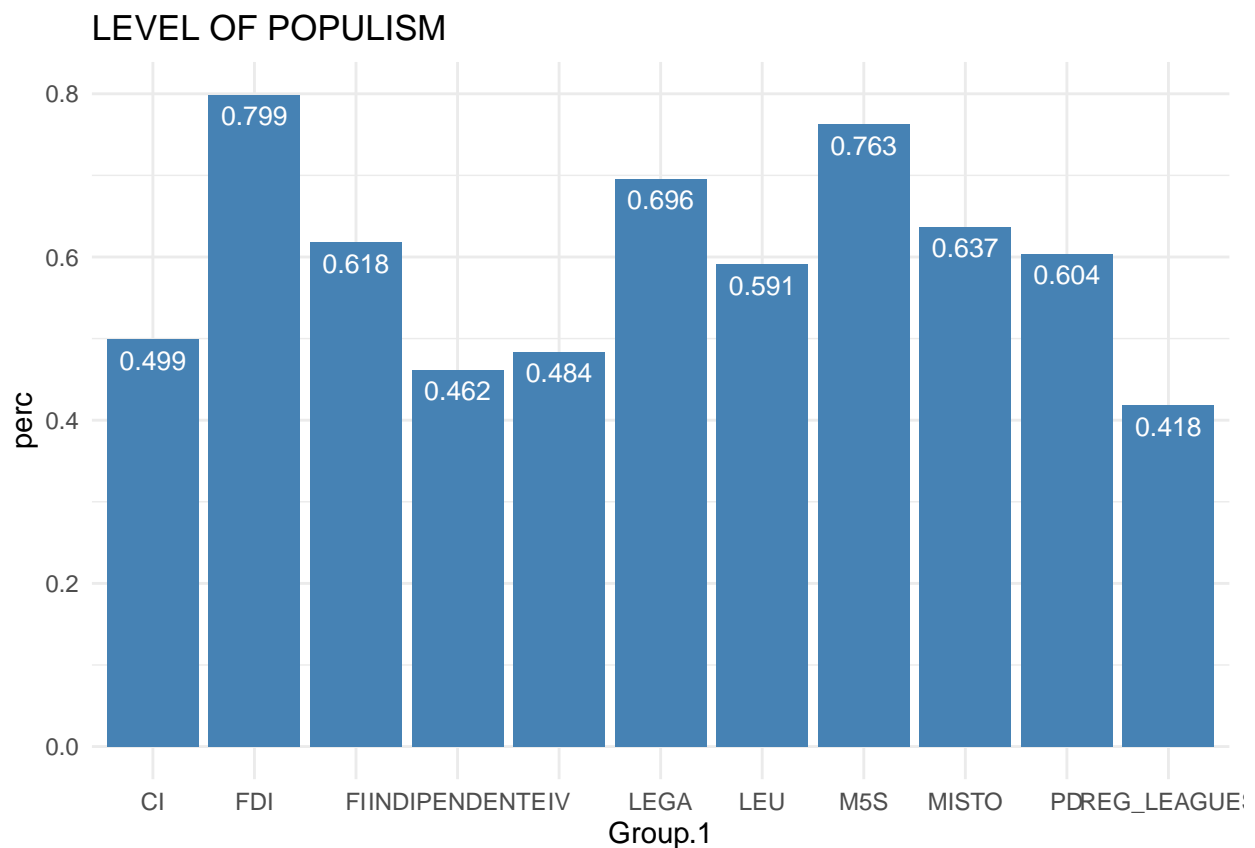
```
# PEOPLE
data_party_people <- aggregate(x = data_dict1$people,  # Specify data column
        by = list(data_dict1$party_id),  # Specify group indicator
        FUN = mean) # Specify function (i.e. mean)
data_party_people$perc <- round(data_party_people$x * 100,3)
kable(data_party_people %>% select(Group.1, 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()+
  labs(title = "LEVEL OF POPULISM: PEOPLE COMPONENT")
```


LEVEL OF POPULISM: PEOPLE COMPONENT
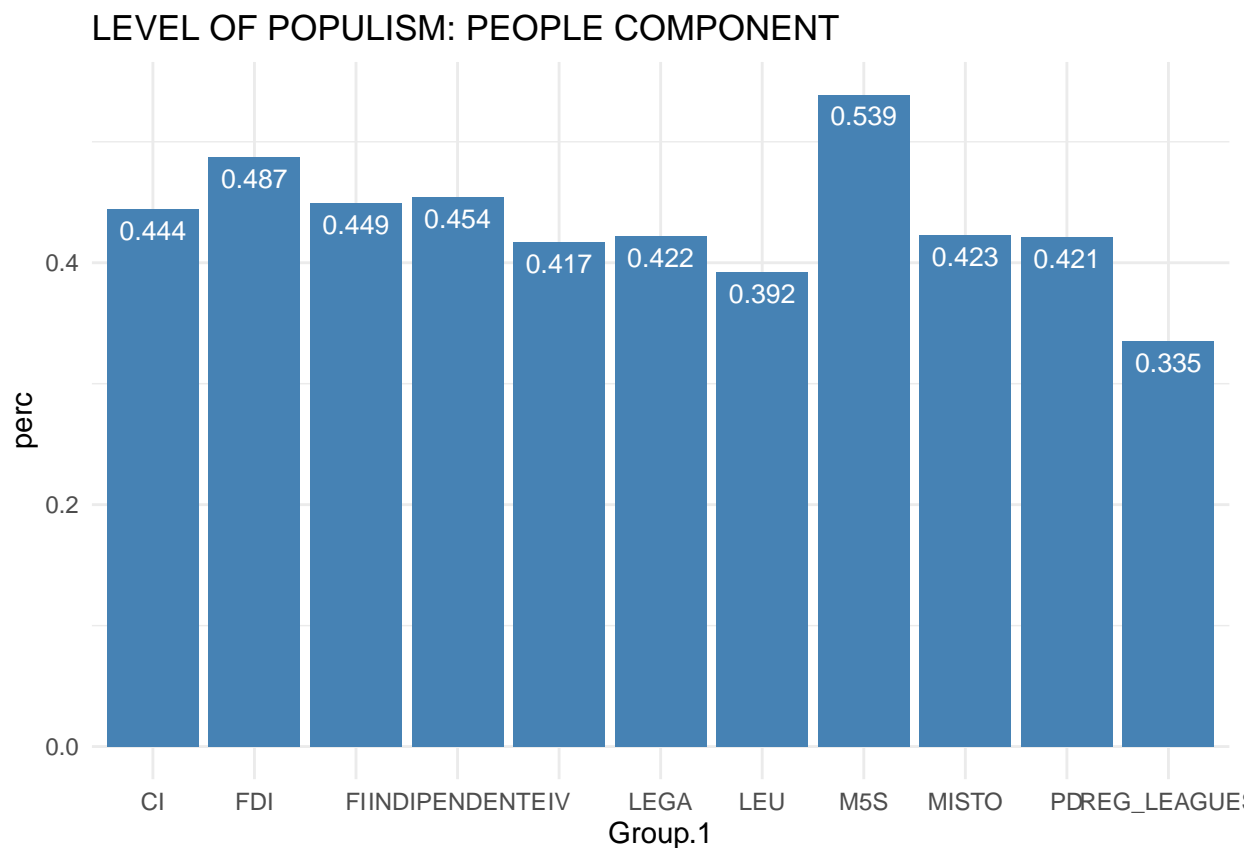
```
# COMMON WILL
data_party_common <- aggregate(x = data_dict1$common_will,  # Specify data column
          by = list(data_dict1$party_id),  # Specify group indicator
          FUN = mean) # Specify function (i.e. mean)
data_party_common$perc <- round(data_party_common$x * 100,3)
kable(data_party_common %>% 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()+
  labs(title = "LEVEL OF POPULISM: COMMON WILL COMPONENT")
```
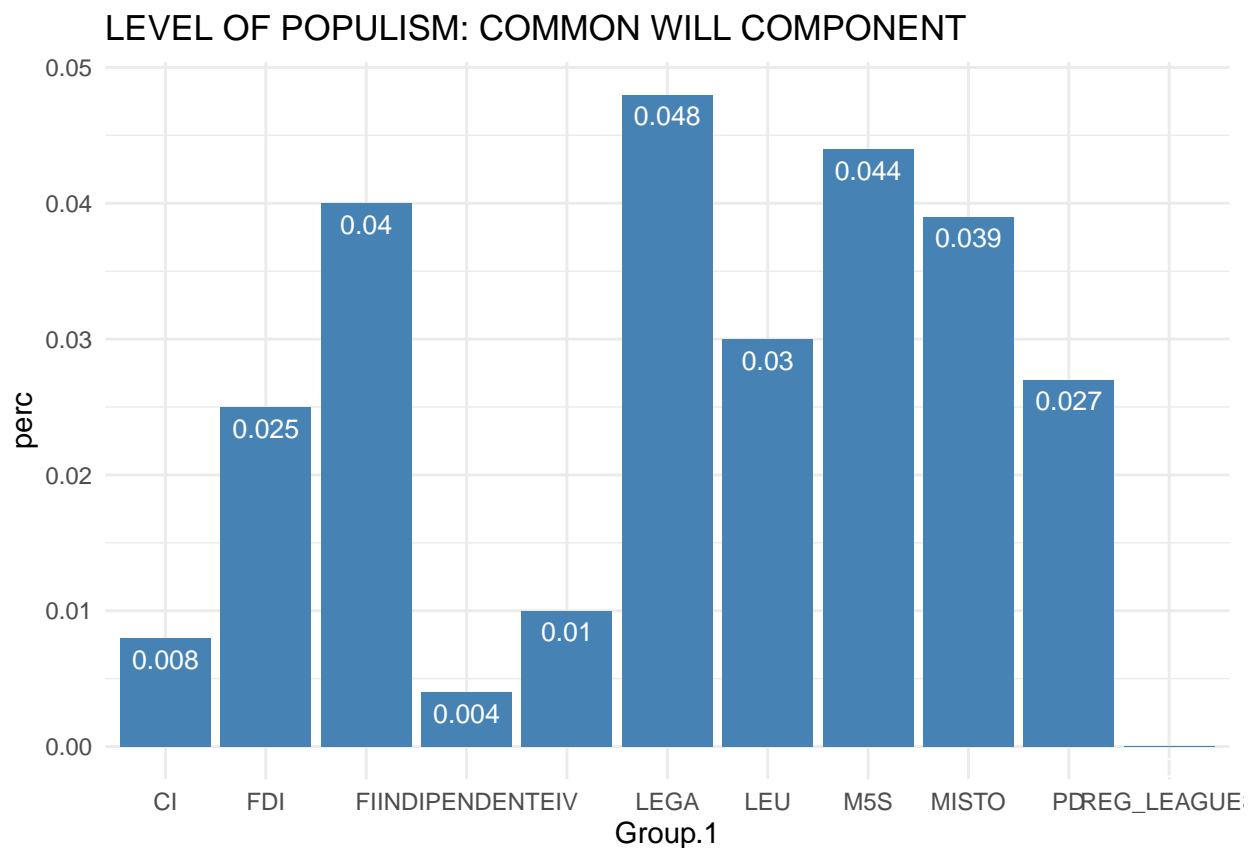


```
# ELITE
data_party_elite <- aggregate(x = data_dict1$elite,  # Specify data column
          by = list(data_dict1$party_id),  # Specify group indicator
          FUN = mean) # Specify function (i.e. mean)
data_party_elite$perc <- round(data_party_elite$x * 100,3)
kable(data_party_elite %>% select(Group.1, 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()+
  labs(title = "LEVEL OF POPULISM: ELITE COMPONENT")
```


LEVEL OF POPULISM: ELITE COMPONENT
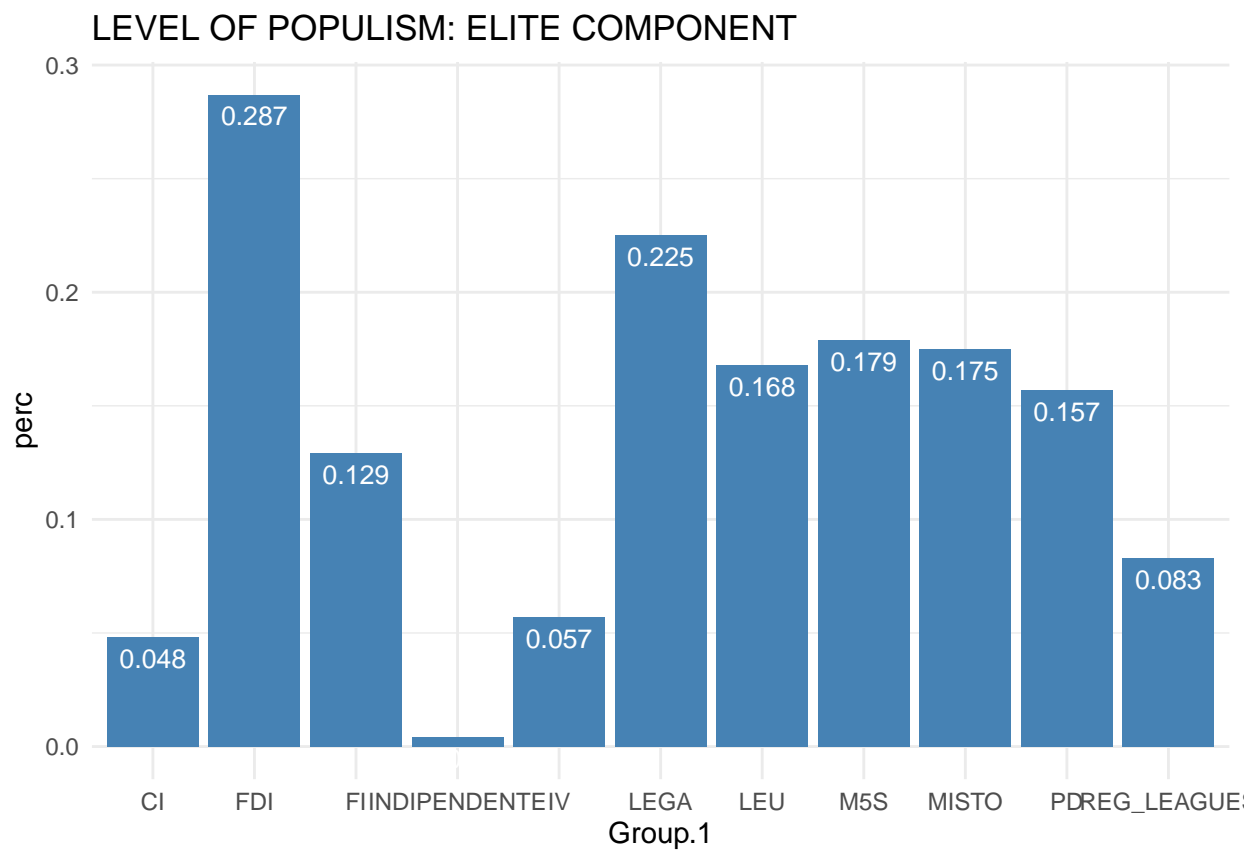
```
###########

#kable(data_dict1)
```

IL valore medio di populismo per ogni partito è statisticamente diverso dal livello di populismo medio del PD ?

```
# regression bivariate 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")

a <- lm(populism ~ factor_party, data_dict1)

b <- lm(people ~ factor_party, data_dict1)
c <- lm(common_will ~ factor_party, data_dict1)
d <- lm(elite ~ factor_party, data_dict1)
summary(a)
```

```
##
## Call:
## lm(formula = populism ~ factor_party, data = data_dict1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.34517 -0.08652 -0.01238  0.08042  0.37004
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)               0.60444    0.04404  13.725  < 2e-16 ***
## factor_partyCI           -0.10503    0.06228  -1.686  0.09487 .
## factor_partyFDI           0.19458    0.06228   3.124  0.00234 **
## factor_partyFI            0.01356    0.06228   0.218  0.82810
## factor_partyINDIPENDENTE -0.14233    0.06228  -2.285  0.02443 *
## factor_partyIV           -0.12078    0.06228  -1.939  0.05532 .
## factor_partyLEGA          0.09147    0.06228   1.469  0.14511
## factor_partyLEU          -0.01339    0.06228  -0.215  0.83018
## factor_partyM5S           0.15814    0.06228   2.539  0.01267 *
## factor_partyMISTO         0.03265    0.06228   0.524  0.60126
## factor_partyREG_LEAGUES  -0.18644    0.06228  -2.994  0.00348 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1393 on 99 degrees of freedom
## Multiple R-squared:  0.4431, Adjusted R-squared:  0.3868
## F-statistic: 7.876 on 10 and 99 DF,  p-value: 3.393e-09
```

```
summary(b)
```

```
##
## Call:
## lm(formula = people ~ factor_party, data = data_dict1)
##
## Residuals:
##         Min        1Q     Median        3Q        Max
```

```
## -0.0033715 -0.0005879 -0.0000602  0.0006587  0.0037806
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                4.211e-03  3.933e-04  10.707   <2e-16 ***
## factor_partyCI             2.243e-04  5.562e-04   0.403   0.6877
## factor_partyFDI            6.588e-04  5.562e-04   1.184   0.2391
## factor_partyFI             2.800e-04  5.562e-04   0.503   0.6158
## factor_partyINDIPENDENTE   3.299e-04  5.562e-04   0.593   0.5545
## factor_partyIV            -4.025e-05  5.562e-04  -0.072   0.9425
## factor_partyLEGA           1.008e-05  5.562e-04   0.018   0.9856
## factor_partyLEU           -2.876e-04  5.562e-04  -0.517   0.6062
## factor_partyM5S            1.182e-03  5.562e-04   2.125   0.0361 *
## factor_partyMISTO          1.438e-05  5.562e-04   0.026   0.9794
## factor_partyREG_LEAGUES   -8.646e-04  5.562e-04  -1.555   0.1232
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.001244 on 99 degrees of freedom
## Multiple R-squared:  0.1496, Adjusted R-squared:  0.06374
## F-statistic: 1.742 on 10 and 99 DF,  p-value: 0.08175
```

```
summary(c)
```

```
##
## Call:
## lm(formula = common_will ~ factor_party, data = data_dict1)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -4.647e-04 -1.291e-04 -4.012e-05  8.635e-05  1.129e-03
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                2.651e-04  8.777e-05   3.020  0.00322 **
## factor_partyCI            -1.841e-04  1.241e-04  -1.483  0.14123
## factor_partyFDI           -1.906e-05  1.241e-04  -0.154  0.87828
## factor_partyFI             1.311e-04  1.241e-04   1.056  0.29361
## factor_partyINDIPENDENTE  -2.249e-04  1.241e-04  -1.812  0.07298 .
## factor_partyIV            -1.658e-04  1.241e-04  -1.336  0.18460
## factor_partyLEGA           2.199e-04  1.241e-04   1.772  0.07955 .
## factor_partyLEU            3.855e-05  1.241e-04   0.311  0.75681
## factor_partyM5S            1.783e-04  1.241e-04   1.436  0.15406
## factor_partyMISTO          1.292e-04  1.241e-04   1.041  0.30052
## factor_partyREG_LEAGUES   -2.651e-04  1.241e-04  -2.135  0.03519 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0002776 on 99 degrees of freedom
## Multiple R-squared:  0.2794, Adjusted R-squared:  0.2066
## F-statistic: 3.839 on 10 and 99 DF,  p-value: 0.0002074
```

```
summary(d)
```

```
##
## Call:
## lm(formula = elite ~ factor_party, data = data_dict1)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -1.152e-03 -3.462e-04 -4.012e-05  2.026e-04  1.840e-03
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)               0.0015683  0.0001691   9.276 4.14e-15 ***
## factor_partyCI           -0.0010904  0.0002391  -4.561 1.46e-05 ***
## factor_partyFDI           0.0013061  0.0002391   5.462 3.50e-07 ***
## factor_partyFI           -0.0002755  0.0002391  -1.152  0.25205
## factor_partyINDIPENDENTE -0.0015282  0.0002391  -6.392 5.41e-09 ***
## factor_partyIV           -0.0010017  0.0002391  -4.190 6.08e-05 ***
## factor_partyLEGA          0.0006847  0.0002391   2.864  0.00511 **
## factor_partyLEU           0.0001151  0.0002391   0.482  0.63120
## factor_partyM5S           0.0002210  0.0002391   0.924  0.35752
## factor_partyMISTO         0.0001830  0.0002391   0.765  0.44598
## factor_partyREG_LEAGUES  -0.0007347  0.0002391  -3.073  0.00274 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0005346 on 99 degrees of freedom
## Multiple R-squared:  0.7123, Adjusted R-squared:  0.6832
## F-statistic: 24.51 on 10 and 99 DF,  p-value: < 2.2e-16
```

2b. Qual è l'andamento nel tempo del livello di populismo generale dei diversi gruppi parlamentari?

Per rispondere, ci servono il tempo e i gruppi.

```
#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")
```
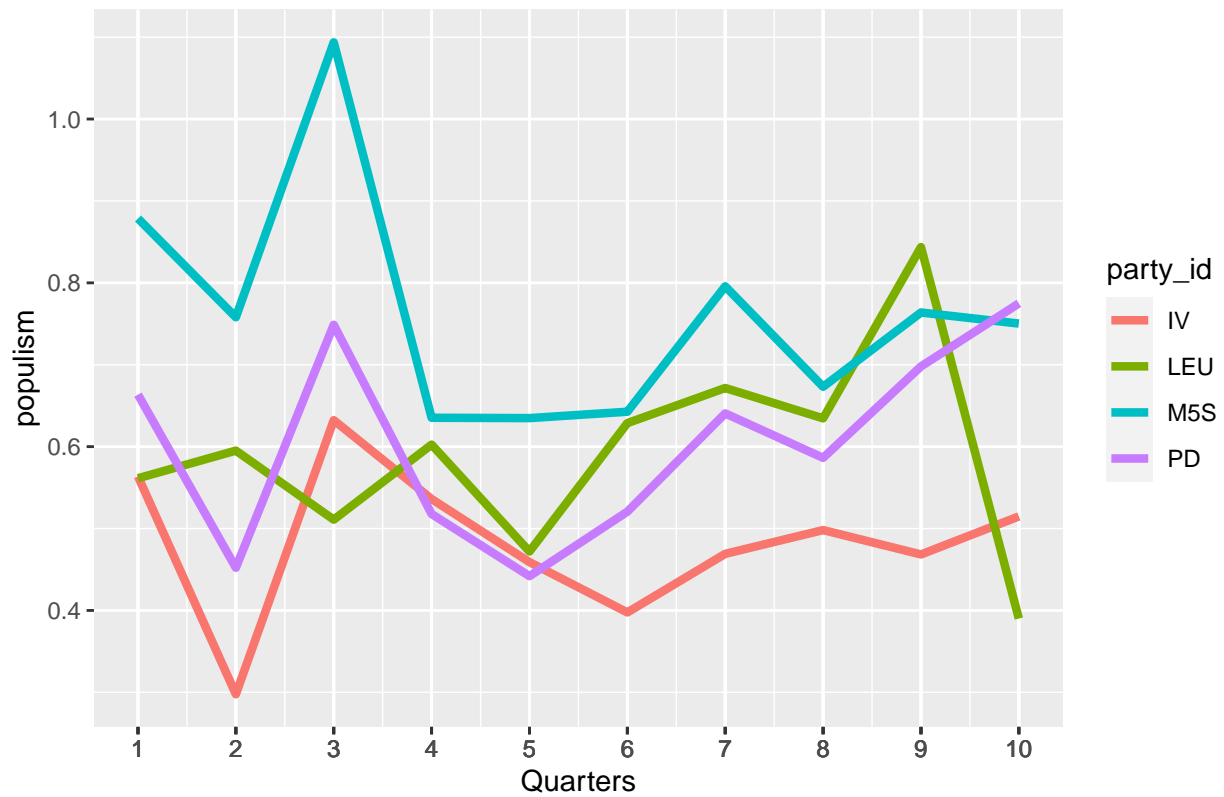
Level of populism in the quarters for left–wing parties


```
# 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$quarter), breaks = right_party$quarte
  ggtitle("Level of populism in the quarters for right-wing parties")
```

Level of populism in the quarters for right-wing parties