Data cleaning and Preliminar analysis

Riccardo Ruta

5/2022

Contents

Data cleaning

Import the dataset and check variables

```r
# import the data
tw <-  read_csv("data/large_files/politicians_final_corrected.csv",
                show_col_types = FALSE )

kable(colnames(tw), col.names = "variables")
```

variables
tw_screen_name
nome
tweet_testo
creato_il
creato_il_code
url
party_id
genere
chamber
status

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))
```

Check the conversion

```r
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

```r
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

Create the week variable

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

Check the variable

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

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

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

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

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

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

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

Create the month variable

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

Check the number of month

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

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

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

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

Create the trimester variable

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

Check the number of trimesters

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

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

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

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

Create the year variables

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

Check the number of years

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

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

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

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

Count the number of missing values

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

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

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

From this check I'll obtain 148178 urls missing, this variable is not collected properly and we will not use in the analysis, and also results 6494 tweets missings, those are the cases when someone post only images or video without text, so the extraction is correct.

Remove rows with missing tweets

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

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

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

Check that the variables make sense

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

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

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

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

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

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

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

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

Adjust the variable genere

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

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

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

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

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

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

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"
```

Create the corpus

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

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

Create the DFM

```
# 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)
```

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
```

```r
# 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")
```

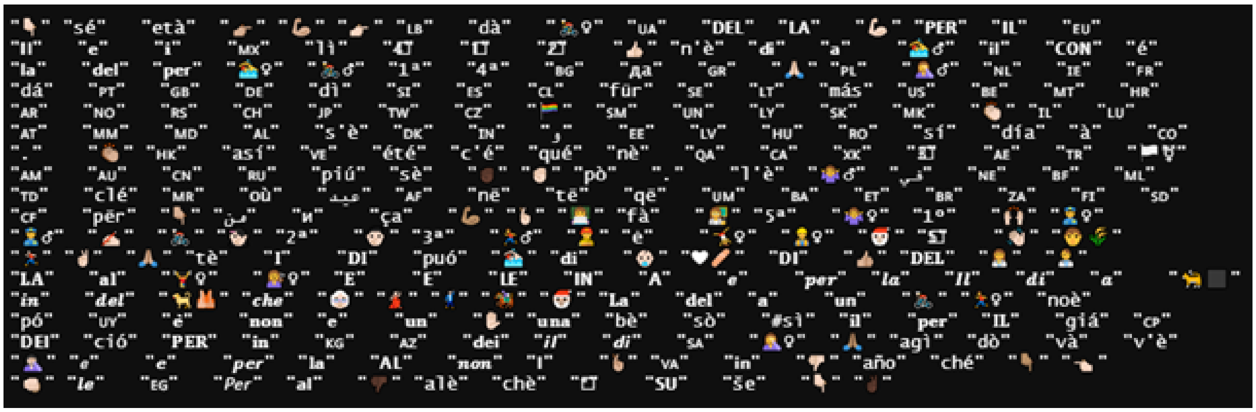


Figure 1: Emoji removed

Now the data are ready for the next analysis

Preliminar analysis

Who is inside this dataset?

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

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

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

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

```
# 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"
```

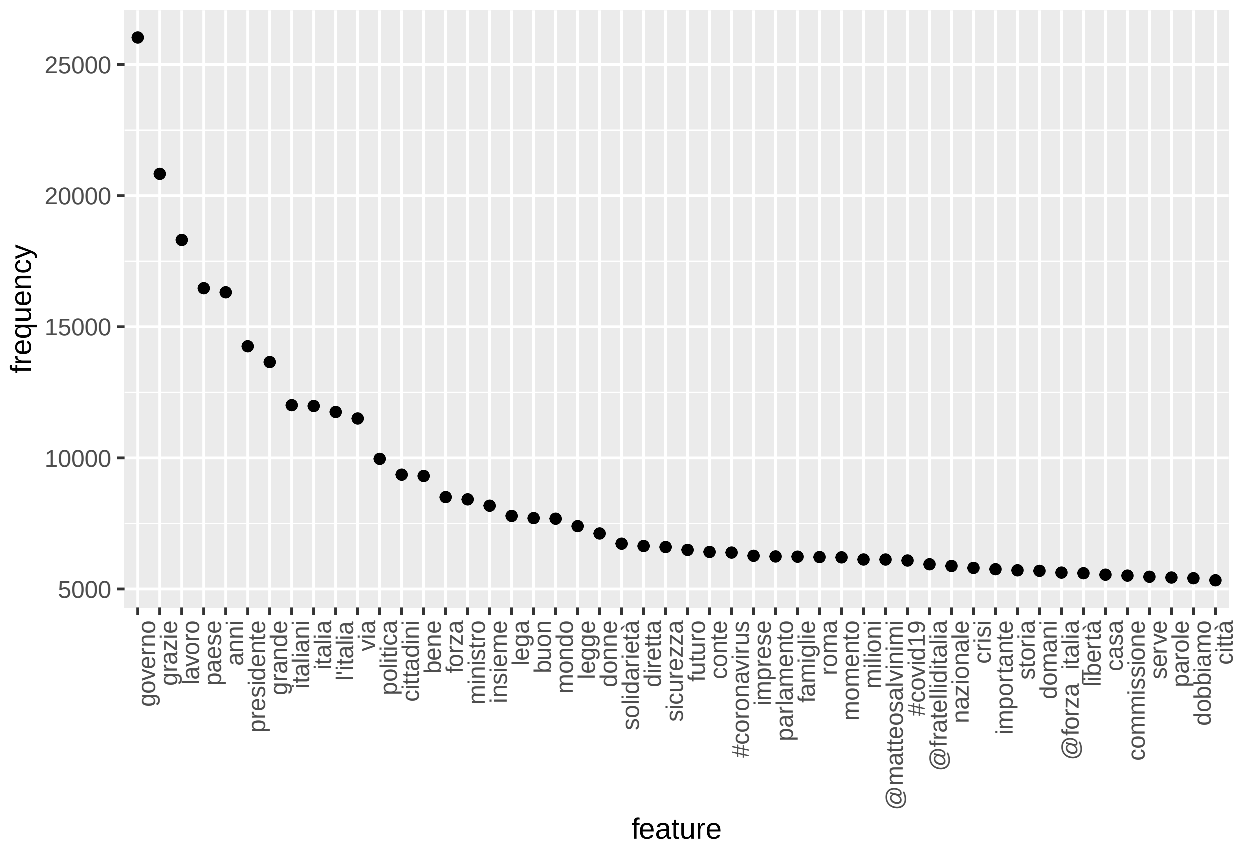
```
min(tw$date)
```

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

Topfeatures frequency

```
# Textplotwordcloud
set.seed(123)
textplot_wordcloud(DFM, min_count = 20,max_words = 300,
     color = c('red', 'pink', 'green', 'purple', 'blue'))
```

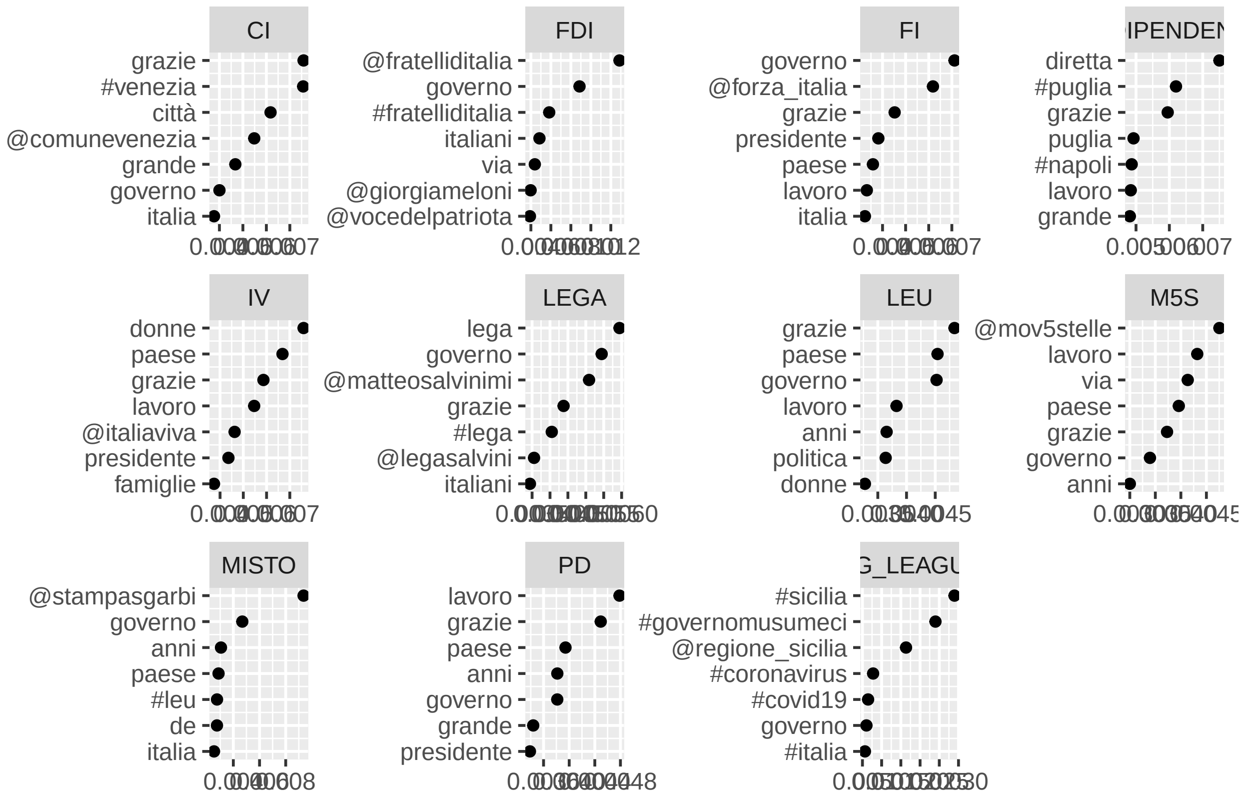

```r
# Plot frequency of the topfeatures in the DFM
features_dfm <- textstat_frequency(DFM, n = 50)
# Sort by reverse frequency order
features_dfm$feature <- with(features_dfm, reorder(feature, -frequency))
ggplot(features_dfm, aes(x = feature, y = frequency)) +
  geom_point() +
  theme(axis.text.x =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



Relative frequency of the topfeatures by Party ID

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

# 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)
```

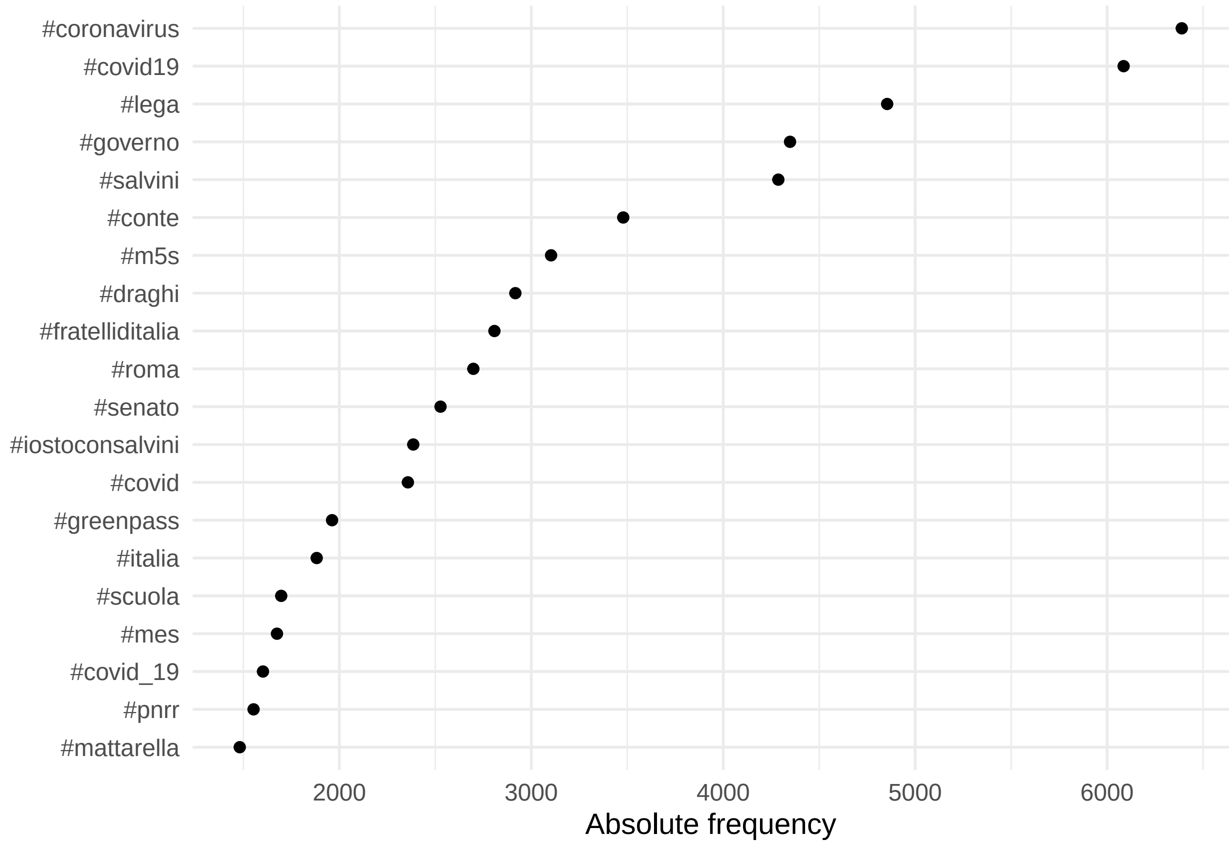
Relative frequency



Most common hashtag

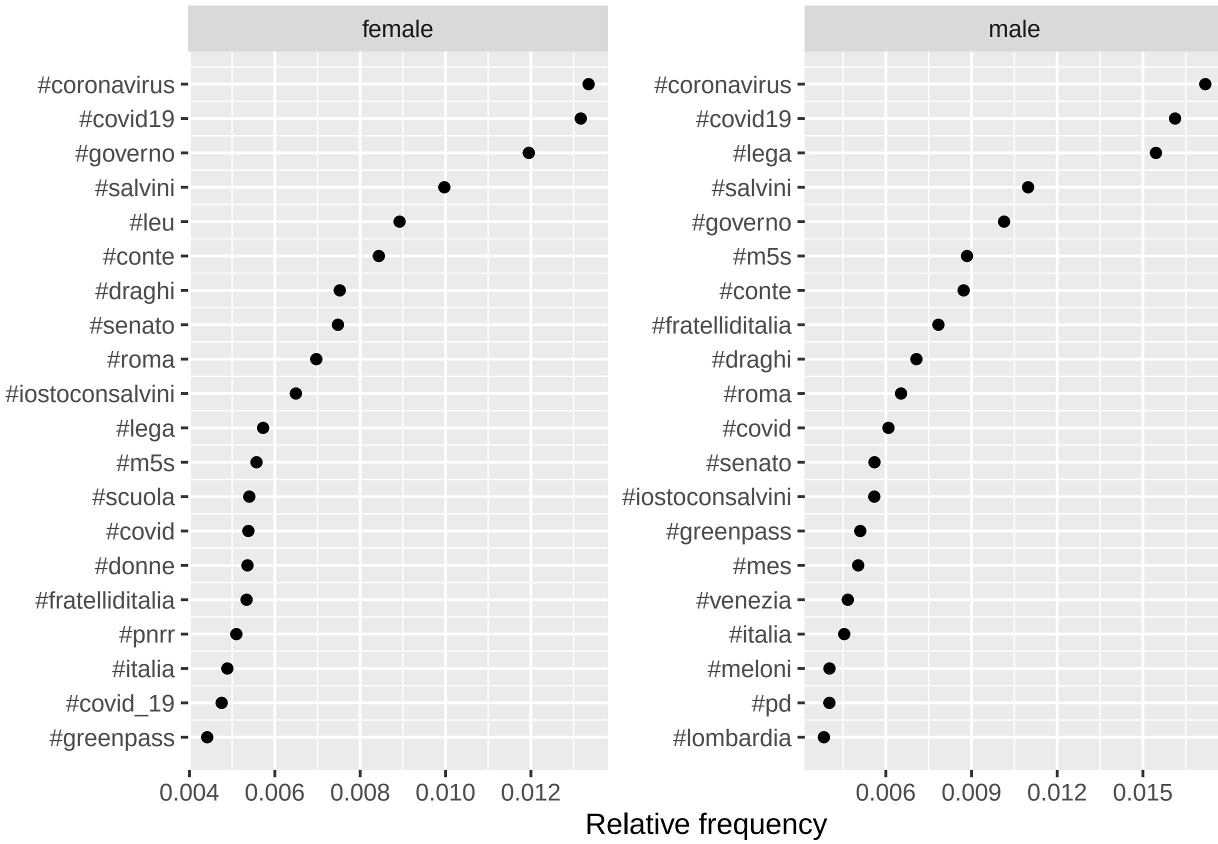
```
tag_dfm <- dfm_select(DFM, pattern = "#*")
toptag <- names(topfeatures(tag_dfm, 20))
toptag
```

```
##  [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"
```



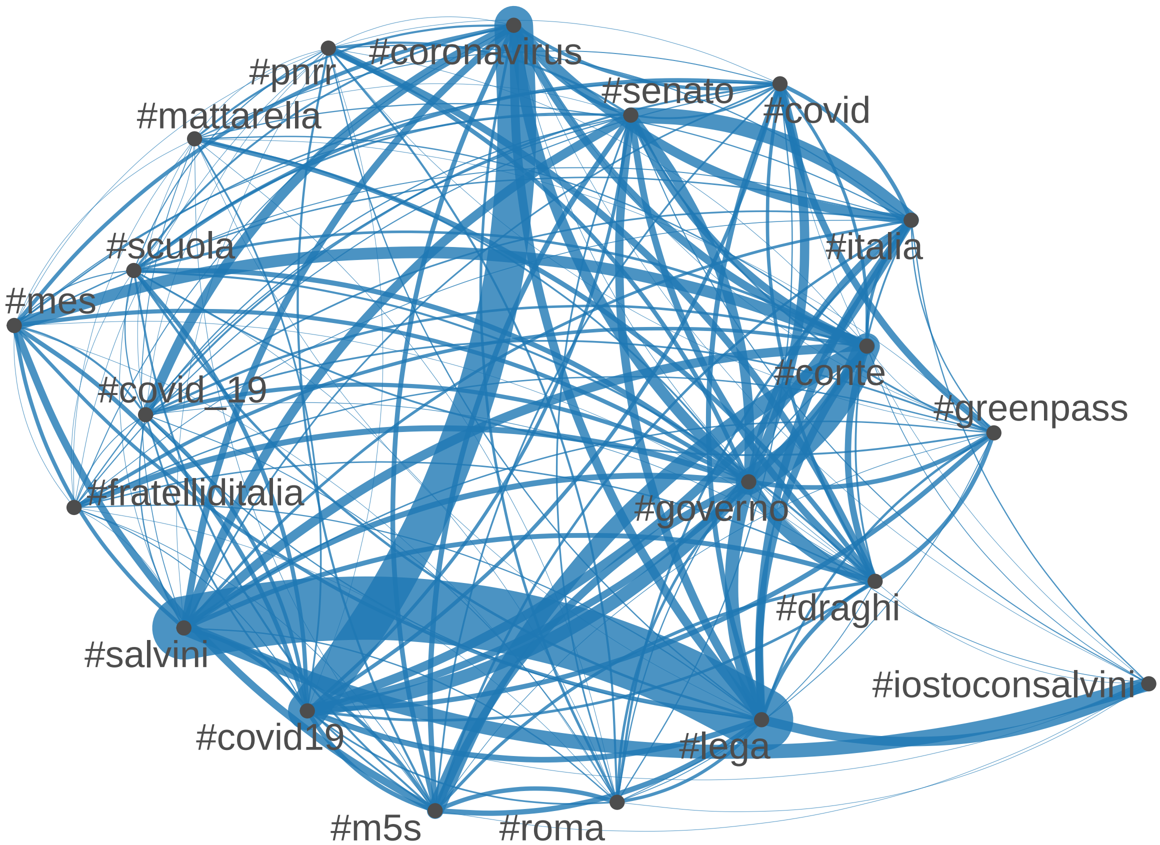
Most common hashtag by Gender

```r
# 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)
```



Co-occurrence Plot of hashtags

```
# NOT WEIGHTED
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)
```



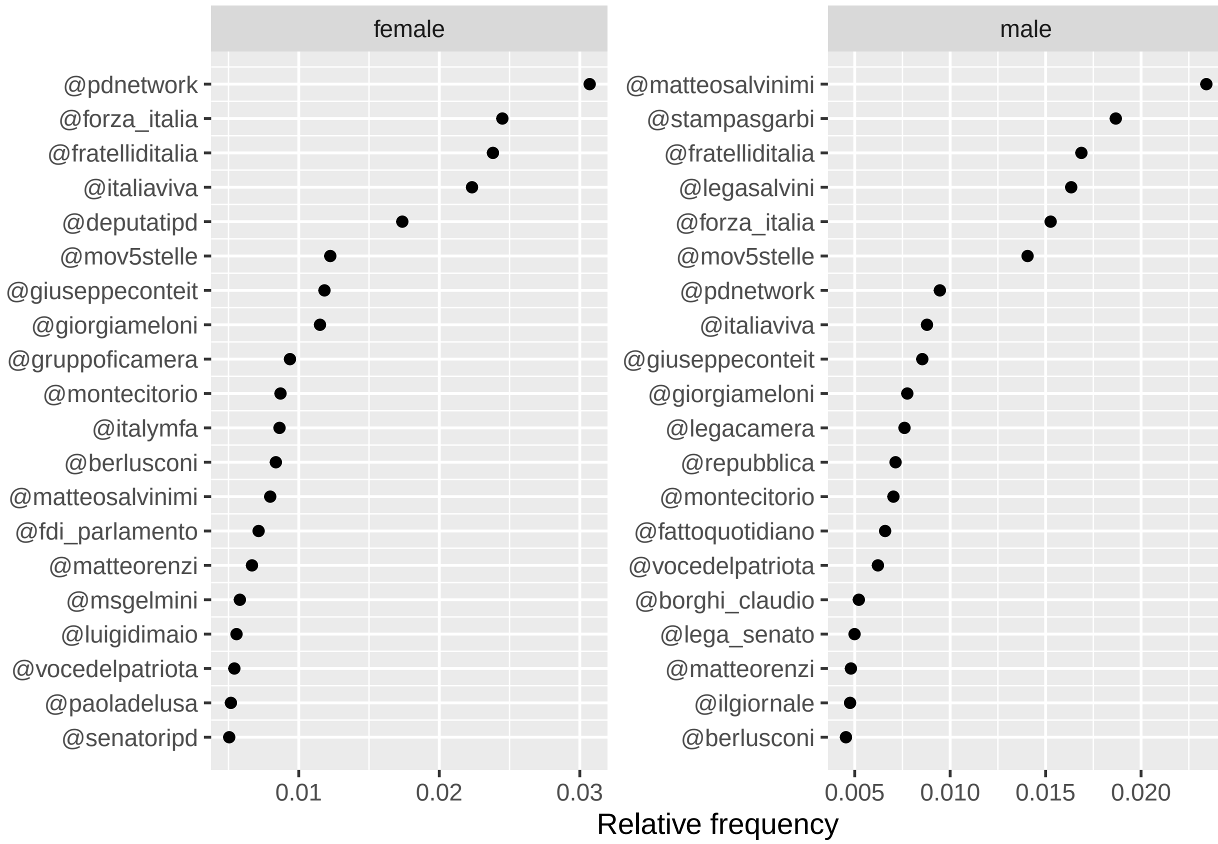
Most frequently mentioned usernames

```
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

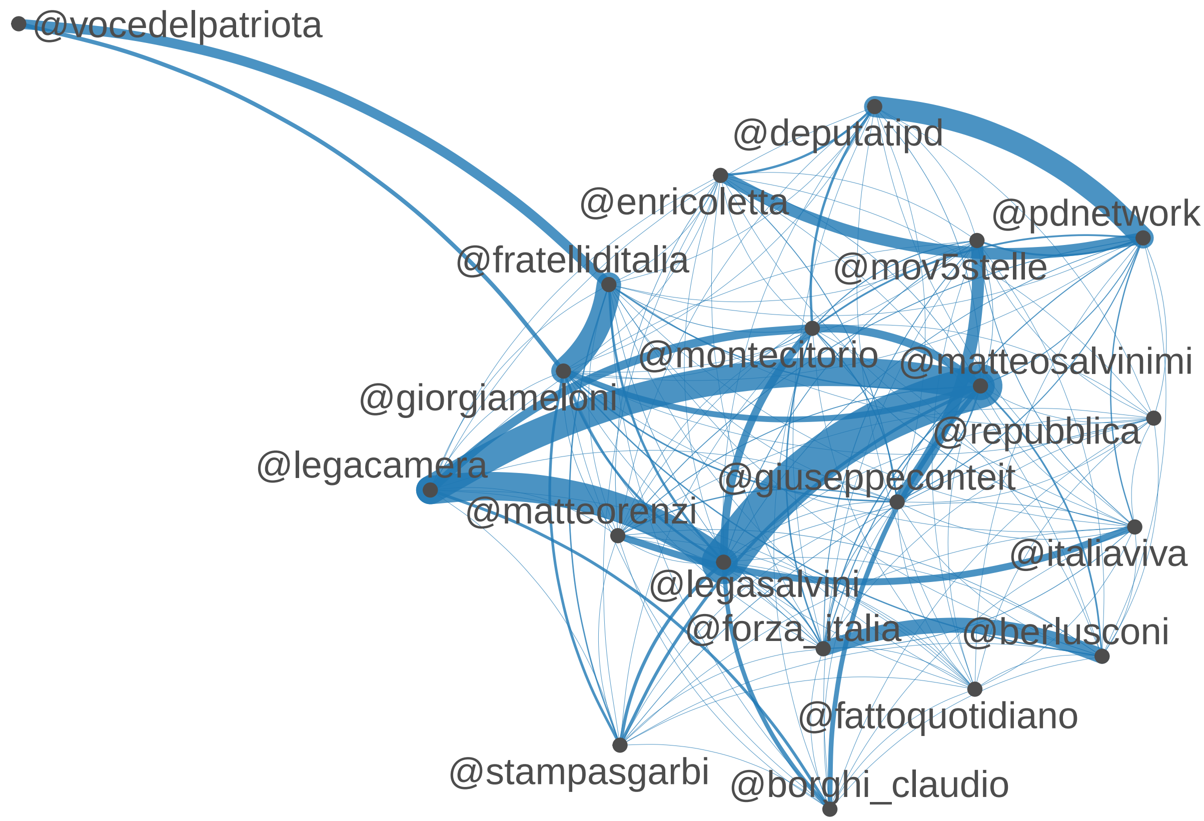
Most frequently mentioned usernames by gender

```
# 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)
```



female
male
@pdnetwork
@forza_italia
@fratelliditalia
@italiaviva
@deputatipd
@mov5stelle
@giuseppeconteit
@giorgiameloni
@gruppoficamera
@montecitorio
@italymfa
@berlusconi
@matteosalvinimi
@fdi_parlamento
@matteorenzi
@msgelmini
@luigidimaio
@vocedelpatriota
@paoladelusa
@senatoripd
@matteosalvinimi
@stampasgarbi
@fratelliditalia
@legasalvini
@forza_italia
@mov5stelle
@pdnetwork
@italiaviva
@giuseppeconteit
@giorgiameloni
@legacamera
@repubblica
@montecitorio
@fattoquotidiano
@vocedelpatriota
@borghi_claudio
@lega_senato
@matteorenzi
@ilgiornale
@berlusconi
0.01
0.02
0.03
0.005
0.010
0.015
0.020
Relative frequency

Co-occurrence plot of usernames



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))

}
)

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

```
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)

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(pa
            if_else(party_id == "CI", "@coraggio_italia",
              if_else(party_id == "LEU", "@liberi_uguali",
                                                      "NA")))))))))
```

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

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

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

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

```
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

How many times a politician cite itself in the tweet

```r
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