Alma Mater Studiorum · Università di Bologna

DIPARTIMENTO DI SCIENZE STATISTICHE
"PAOLO FORTUNATI"
Corso di Laurea Triennale in Scienze statistiche - Economia e Impresa

ELEZIONI AMERICANE 2020: SENTIMENT ANALYSIS SUI PRINCIPALI CANDIDATI

in

"UTILIZZO STATISTICO

DI BANCHE DATI ECONOMICHE ONLINE"

Presentata da:
Alberto Parenti
Matricola 0000880577

Relatore:
Chiar.mo Prof.
Ignazio Drudi

II Appello di Laurea
Anno Accademico 2020/2021

Indice

1 Introduzione

Gli argomenti principali trattati nel seguente saggio sono stati analizzati in occasione del corso "Utilizzo statistico di banche dati online" utilizzando come esempi i tweet riguardanti i principali esponenti dei partiti politici italiani. Questo mi ha spinto ad applicare la medesima metodologia in una realtà nazionale dove Twitter è utilizzato con maggior frequenza.

Negli Stati Uniti il social network dell'uccellino blu è un mezzo comunemente usato per esporre le proprie opinioni quotidianamente e per ricevere notizie. Considerando dati di *Statista*, azienda tedesca di database specializzata nei dati del mercato e dei consumatori, aggiornati a Giugno 2021, 187 milioni di utenti accedono a Twitter ogni giorno, il 20% di loro ha sede negli Stati Uniti. Per comprendere l'importanza della funzione informativa di Twitter e del suo legame con la politica è utile evidenziare che la categoria professionale più rappresentata tra quelle dei profili Twitter verificati sia proprio quella dei giornalisti, 25% del totale (2015). É curioso notare che il profilo più seguito al mondo è quello del Presidente degli U.S.A. numero 44, Barack Obama, con 130 milioni di followers.

Di fatti, l'analisi dell'elaborato nasce dalla volontà di quantificare l'opinione pubblica americana sui candidati principali delle elezioni presidenziali del 4 novembre 2020. Si è deciso di porre come domanda chiave dello studio: "Come è cambiata l'opinione della popolazione americana nei confronti dei principali candidati delle elezioni politiche a cavallo dell'*election day* (3 novembre 2020)?". L'analisi si riferisce esclusivamente al candidato Joe Biden del Partito Democratico e a Donald Trump del Partito Repubblicano. Per ottenere una risposta si é applicata la sentiment analysis su un dataset di tweets ed é stato utilizzato il software R.

La tesi è articolata in 3 capitoli: nel primo capitolo si tratta la metodologia applicata necessaria a svolgere lo studio. Per poter analizzare i dati in maniera chiara ed esaustiva é stato necessario applicare operazioni di pulizia e di lemmatizzazione sul campione scaricato dal social network Twitter. In questo modo é stato possibile realizzare l'analisi descrittiva trattata nel secondo capitolo. Dopo aver definito il tipo di tweet, si è compiuta sia un'analisi quantitativa, che una qualitativa tramite l'analisi degli hashtag e la sentiment analysis. Nel terzo capitolo si procede a discutere i risultati ottenuti dall'analisi dei dati e nelle conclusioni, infine, viene ipotizzato un possibile studio futuro.

2 Metodologia

2.1 Download tweet

Si è iniziato scaricando i tweet che citano Biden e/o Trump. Questa rilevazione è stata eseguita per almeno due volte al giorno per entrambi i candidati dal 13 ottobre al 14 novembre 2020, per un totale di 33 giorni.

Viene applicato il comando `search_tweets` per tale scopo. Si impone il download di massimo 20000 tweet geolocalizzati presso gli Stati Uniti di America, esclusi i retweet.

```
twT <- search_tweets("trump",n=20000,
       geocode = lookup_coords("usa"),include_rts = F)

twB <- search_tweets("biden",n=20000,
       geocode = lookup_coords("usa"),include_rts = F)
```

2.2 Operazioni di recupero e aggregazione dei tweet

Da ogni singola rilevazione attuata si ottengono 134 file *.RData*, per poi inserirli all'interno della sotto-cartella *election2020*. Di consegenza, si è creato la lista lstFl con gli elementi presenti in *election2020*.

```
lstFl <- list.files("./election2020",pattern = "RData")
length(lstFl)
```

Successivamente, si è creato il dataframe complessivo, denominato *dftw*.

```
for(i in 1:length(lstFl)){
    print(i)
    temp.space <- new.env()
    bar <- load(paste("./election2020/",lstFl[i],sep=""),temp.space)
    tmpdat <- get(bar, temp.space)
    tmpdat <- tmpdat %>%
              select(user_id,status_id,created_at,screen_name,text,
              is_quote,is_retweet,favorite_count,retweet_count,
              quote_count,hashtags,mentions_screen_name)
    rm(temp.space)
```

```
    if(i == 1) {
       dftw <- tmpdat
    } else {
    dftw <- rbind(dftw,tmpdat)
  }
}
```

La seconda parte di questa fase di aggregazione dei tweet consiste nell'identificare i tweet univoci, poichè durante il download dei tweet può capitare che venga considerato più volte il medesimo messaggio.

```
dftw$id <- 1:nrow(dftw)
unici <- dftw %>%
         group_by(status_id) %>%
         summarise(n=n(),id=min(id))
```

Quindi, si termina creando il dataframe denominato *dftwr* inserendo all'interno i tweet univoci indivuati sopra.

```
dftwr <- dftw[dftw$id %in% unici$id,]
```

2.3 Creazione dataframe campionario

Data la mole di 2081460 di osservazioni appartenenti a *dftwr*, si è deciso di creare un dataframe campionario composto da 200000 tweet, denominato *dftwrc*.

```
dftwrc <- dftwr[sample(nrow(dftwr), 200000),]
```

Utilizzando il comando `summary`, si è riuscito a determinare il periodo temporale in cui sono stati scritti i tweet.

```
summary(dftwrc$created_at)
```

Min.	1st Qu.	Mean
"13-10-2020 23:55:14"	"22-10-2020 22:24:43"	"30-10-2020 09:08:51"
Median	3rd Qu.	Max.
"31-10-2020 13:52:49"	"06-11-2020 13:18:00"	"14-11-2020 00:25:22"

Tabella 1: periodo di scrittura tweet di *dftwrc*

Per la visione dell'andamento per giorno di *dftwrc* si applica la funzione `ts_plot`.

```
ts_plot(dftwrc)
```

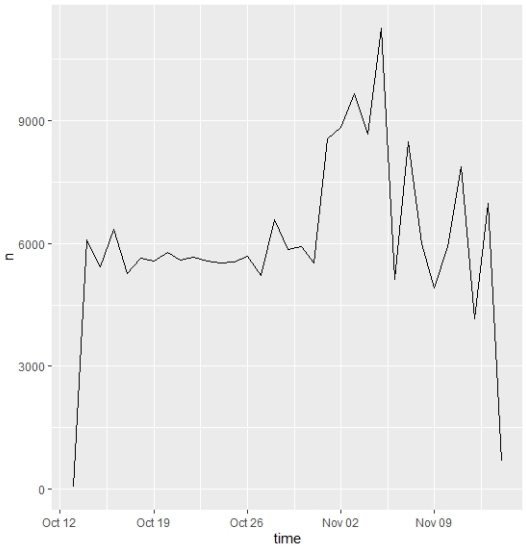


Figura 1: andamento per giorno del campione

Dall'analisi del grafico in (Figura 1) si denota un comportamento pressochè stabile fino al 31 ottobre. A cavallo dell'*election day* (3 novembre 2020) avviene un netto incremento registrando più di 11000 osservazioni il 5 novembre rispetto alle 200000 che compongono il dataframe campionario *dftwrc*. Nei restanti giorni si hanno forti impennate e collassi dovuti probabilmente dalla natura randomica del campione.

2.4 Correzione forme composte

Dopo aver diminuito la quantità di osservazioni, si procede con la ricerca di forme composte applicando il comando `visNGram`. Nella personalizzazione del comando, *ngrF* indica il valore massimo della lunghezza di n-grammi e *nn* il numero di n-grammi da visualizzare per ciascuna lunghezza. Un n-gramma è una sottosequenza di n elementi di una data sequenza. Secondo l'applicazione, gli elementi in questione possono essere fonemi, sillabe, lettere e parole. Quindi, si è cercato i 50 unigrammi, digrammi, trigrammi e 4-grammi più ricorrenti.

```
visNGram(dftwrc$txt, ngrF = 4, nn = 50)
```

Successivamente, si crea il vettore *correz* con le correzioni delle forme composte.

```
correz <- c("Joe␣Biden","Biden",
            "Biden␣and␣Kamala␣Harris","Biden_Harris",
            "Joe␣Biden␣and␣Kamala","Biden_Harris",
            "Hunter␣Biden",NA,
            "Biden␣Harris",NA,
            "U␣S","United_States",
            "White␣House",NA,
            "U␣S␣President","Potus",
            "United␣States",NA,
            "key␣contacts",NA,
            "Vice␣President",NA,
            "Biden␣and␣Kamala","Biden_Harris",
            "Biden␣and␣Harris","Biden_Harris",
            "Trump␣and␣Biden","Trump_Biden",
            "American␣people",NA,
            "Kamala␣Harris","Harris",
            "Donald␣Trump","Trump",
            "BIDEN␣HARRIS","Biden_Harris",
            "BIDEN","Biden",
            "HARRIS","Harris",
            "Democrat␣party",NA,
            "Chinese␣firm",NA,
            "President␣of␣the␣United␣States␣of␣America","Potus",
            "President␣of␣the␣United␣States","Potus",
            "President␣Elect",NA,
            "President␣elect",NA,
            'United␣States␣of␣America',"U.S.A.",
            "Biden␣and␣Vice␣President","Biden_Harris",
            "Amy␣Coney␣Barrett",NA,
            "President␣Trump","Trump",
            "President␣Donald␣Trump","Trump",
            "President␣Donald␣J␣Trump","Trump",
            "Donald␣J␣Trump","Trump",
            "supreme␣Court",NA,
```

```
                "New York Times",NA,
                "Make America Great Again",NA,
                "GOP","Republican_Party",
                "Republican Party",NA,
                "Democrat Party",NA,
                "America Great Again",NA,
                "trump","Trump",
                "TRUMP","Trump",
                "Department of Homeland Security",NA,
                "Hunter Biden's","Hunter_Biden",
                "Joe Biden's","Biden",
                "Biden's","Biden",
                "Donald Trump's","Trump",
                "Trump's","Trump",
                "Second Amendment",NA,
                "First Lady",NA
)
```

Per terminare questa fase, è necessario applicare al vettore testo di *dftwrc* le correzioni sopra elencate.

```
dftwrc$txt <- corFrmComp(vText = dftwrc$txt, correzioni=correz)
```

2.5 Pulizia

Dopo aver corretto le forme composte in *dftwrc*, si calcola quante parole vi sono in ogni testo.

```
dftwrc$id <- 1:nrow(dftwrc)
dftwrc$nparole <- sapply(dftwrc$txt,wordcount)
```

Successivamente, per restringere il campo di osservazione ad opinioni minimamente articolate, vengono presi ad esame solo testi composti da almeno 6 parole; oltre a far attenzione di considerare solo i tweet in cui è citato almeno uno dei due candidati. Si costruisce il dataframe *dftwrrc* contemplando tali requisiti.

```
dftwrrc <- dftwrc[dftwrc$nparole>5 & !is.na(dftwrc$tipo),]
```

Nella tabella sottostante si può notare la suddivisione dei 182868 tweet in base a chi viene citato all'interno del testo.

```
table(dftwrrc$tipo)
```

BT	DT	JB
44268	81108	57492

Tabella 2: spartizione di *dftwrrc*

In 44268 occasioni (24.2 %) vengono citati sia Biden che Trump; in 81108 (44.4 %) il candidato dei repubblicani Donald Trump e i restanti 57492 tweet (31.4 %) al democratico Joe Biden.

Una volta che si è creato il *dftwrrc*, si aggiunge un identificato progressivo ai testi selezionati.

```
dftwrrc$id <- 1:nrow(dftwrrc)
dftwrrc %>%
  group_by(tipo) %>%
  summarise(paroletot=sum(nparole),sum(nJB),sum(nDT))
summary(dftwrrc$nparole)
```

Il procedimento indicato permette di calcolare quante parole sono presenti all'interno dei testi di *dftwrrc*, come riportato nella seguente tabella.

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
6.00	14.00	23.00	26.41	38.00	118.00

Tabella 3: quantità di parole all'interno di *dftwrrc*

Come imposto precedentemente, il minimo di parole nei testi di *dftwrrc* è 6. In media i tweet considerati hanno 26 parole e il tweet più lungo ne contiene 118.

Si termina questa fase, ripulendo i testi da tutte le mention, ad esempio *@JoeBiden* e *@realDonaldTrump*, e gli hashtag utilizzando il comando `cleanTesto`.

```
dftwrrc$txtp <- cleanTesto(dftwrrc$txt, punteggiatura = T,
                numeri = F, minuscolo = F,hashtag = T,mention = T)
```

2.6 Lemmatizzazione

Nella fase di *lemmatizzazione* ci si concentra sul valore delle singole parole, di conseguenza, si trascura la sintassi, ossia il modo in cui le parole si combinano. Da evidenziare il fatto che la lemmatizzazione è un procedimento che può esser sintetizzato in quattro passi. Per prima cosa, si eliminano tutti i segni non alfabetici, come ad esempio punti e virgole. Poi, si applica la cosiddetta *tokenizzazione*, la quale consiste nell'individuare le singole parole. Si prosegue con l'eliminazione di tutte le parole che appartengono a parti del discorso non significative come articoli, preposizioni e congiunzioni. Si conclude con le parole lemmatizzate.

Per comprendere al meglio la lemmatizzazione, è necessario concentrarsi sulla differenza fra un *lessema* e un *lemma*. I primi sono entità linguistiche astratte che includono tutte le forme flesse di una parola. Ad esempio *vota*, *ha votato*, *elegge*, *ha eletto*, *voti* e *vittorioso*. I lemmi, invece, sono la forma di citazione dei lessemi nei dizionari o lessico di frequenza. Coincidono con l'infinito per i verbi (*eleggere*, *votare*) e con il maschile singolare per gli aggettivi e sostantivi (*vittorioso*).

Di fatto, la lemmatizzazione consiste nel ridurre le forme fisse di uno stesso lessema a una forma di citazione, per l'appunto il lemma. Inoltre, la lista di frequenza conta solo le diverse forme di citazione come lemmi. Quindi, prendendo ad esempio una frase come "*Il popolo americano elegge il presidente ogni 4 anni.*", lemmatizzata diventa: "*popolo americano eleggere presidente ogni 4 anni.*"

Il processo di lemmatizzazione inizia col comando `lemmaUDP` utilizzando la libreria di R `udpipe` in lingua inglese.

```
outUV <- lemmaUDP(dftwrrc$txtp,model = ud_model_EN,
         doc_id = dftwrrc$id, stopw = tm::stopwords("en"))
```

Tramite la funzione `head` si ottiene il contenuto dell'output della lemmatizzazione.

```
head(outUV[,c(1:3,5:9,15)],40)
```

Si ottiene il file *outUV*, che contiene per ogni termine il suo lemma e il tipo di forma grammaticale. Vengono eliminate le forme ausiliarie, come congiunzioni e numeri.

```
outUV %>%
  filter(upos=="AUX") %>%
  group_by(lemma,STOP) %>%
  summarise(n=n()) %>%
  spread(STOP,n)
xpos_del <- c("PRP","DT")
upos_del <- c("SCONJ","CCONJ","NUM","AUX")
```

Successivamente, si compone il dataframe `txtL_UV2` ricostruendo i testi dei messaggi lemmatizzati.

```
txtL_UV2 <- outUV %>%
            filter(!is.na(outUV$lemma) & outUV$STOP==FALSE &
            !outUV$xpos %in% xpos_del & !outUV$upos %in% upos_del) %>%
            select(doc_id,lemma) %>%
            group_by(doc_id=as.numeric(doc_id)) %>%
            summarise(txtL=paste(lemma,collapse = "␣"))
```

Tramite l'utilizzo della funzione `gsub`, si sistemano i simboli # e @ isolati.

```
txtL_UV2$txtL <- gsub("␣#␣","␣#",txtL_UV2$txtL)
txtL_UV2$txtL <- gsub("␣@␣","␣@",txtL_UV2$txtL)
```

Le parole possono essere distinte in parole funzionali e parole piene. Quest'ultime hanno un proprio contenuto semantico autonomo; sono portatrici di parti *sostantive* del contenuto di un discorso (nomi e aggettivi), delle sue modalità di enunciazione (avverbi) o di azione (verbi). Le prime, invece, sono parole generalmente con un'alta frequenza di occorrenza nel testo, che hanno funzione grammaticale (articoli, pronomi, preposizioni, congiunzioni) e che non sono portatrici di significato autonomo.

Nei processi di text mining le parole funzionali, insieme ad altre molto frequenti, come verbi ausiliari e modali, e moderatamente informative, vengono inserite in apposite liste di parole da trascurare, le cosiddette *stop words*. Quindi, si crea il vettore con le stop words, quali *elections2020*, *election2020*, *election*, *2020*, *vote*.

```
mystpw <- c("elections2020","election2020","election","2020","vote")
```

In seguito, con la funzione `creaTDM`, si crea una *Document Term Matrix* dopo aver processato e rimosso le stop words. La *DTM* è una matrice che descrive la frequenza dei termini che ricorrono in una collezione di documenti.

```
risIDF2 <- creaTDM(tolower(txtL_UV2$txtL), lemmatiz = F,
           mystopwords = mystpw, output = "all",
           weight = "tfidf", lang = "english")
names(risIDF2)
head(risIDF2$freq.Frm)
par(mar=c(0,0,0,0))
```

Si conclude la lemmatizzazione con la composizione della *wordcloud* per determinare le parole più ricorrenti nelle osservazioni campionarie. L'intento di base è quello di visualizzare i pesi delle parole utilizzate. Si tratta di forme grafiche nelle quali i termini chiave sono rappresentati con un corpo che aumenta in funzione della frequenza con le quali si sono ripetuti. L'effetto finale è una *nuvola* composta, in questo caso, da massimo 125 parole con grandezze diverse e che ruota intorno ad alcune chiavi principali in corpo maggiore. Quest'immagine in qualche modo dovrebbe rappresentare l'essenza del testo considerato.

```
wordcloud(risIDF2$freq.Frm$forme,risIDF2$freq.Frm$freq,
          max.words = 125,random.order = F, scale = c(2, 0.3),
          colors = rev(pal_lancet("lanonc")(4)))
```

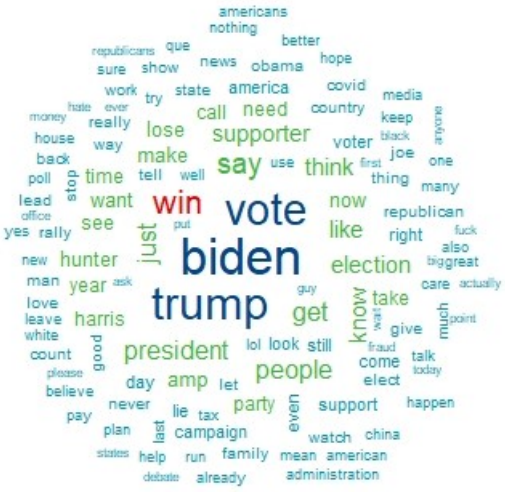


Figura 2: wordcloud lemmatizzazione

Quindi, le parole più ricorrenti sono: "vote", "biden", "trump" e "win".

3 Analisi descrittiva

3.1 Definizione del tipo di tweet

Durante il processo di pulizia e lemmatizzazione si è considerato il dataframe campionario *dftwrc*, trasformato poi in *dftwrrc*; per l'analisi quantitiva si considera il dataframe originario *dftwr* per aumentare le osservazioni al fine di ottenere un risultato più realistico.

Per prima cosa si raggruppano i tweet in base alla presenza di Biden e/o Trump nel testo tramite la funzione `grepl`.

```
xJB <- grepl("biden",dftwr$text,ignore.case = T)
xDT <- grepl("trump",dftwr$text,ignore.case = T)
xPO <- grepl("potus",dftwr$text,ignore.case = T)
xJBDT <- ifelse(xJB==T & (xDT==T  |xPO==T),"BT",
          ifelse(xJB==T,"JB", ifelse(xDT==T | xPO==T,"DT",NA)))
dftwr$tipo <- xJBDT
```

Il conteggio viene fatto tenendo conto anche delle mention (@realDonaldTrump e @JoeBiden). Inoltre, vengono considerati per Trump anche i tweet in cui è presente la parola *potus*.

3.2 Creazione grafici quantitativi

Al fine di ottenere la somma dei tweet classificati nei tre gruppi (Biden Trump, Trump, Biden) si utilizza il comando `addmargins`. Questo permette, data una tabella, di specificare quale fattore di classificazione si vuole espandere di un livello per trattenere i margini da calcolare.

```
addmargins(table(dftwr$tipo))
```

BT	DT	JB	Sum
470665	890445	632625	1993735

Tabella 4: spartizione di *dftwr*

Pertanto, si è ottenuto un totale di 1993735 tweet: 470665 dei quali contengono all'interno sia il nominativo di Trump che quello di Biden (23.6 %); in 890445 tweet

(44.7 %) vi è presente un riferimento nominativo di Donald Trump, mentre i restanti 6326625 (31.7 %) contengono quelli che citano Joe Biden.

Partendo dal dataframe originario *dftwr*, si prosegue con la costruzione del dataframe *tab1*, in cui vengono classificati i tweet in base alla data di creazione. Inoltre, viene calcolato l'engagement generato per tweet sommando il conteggio dei retweet e dei like per poi dividerlo col totale dei tweet.

```
tab1 <- dftwr %>%
        filter(!is.na(tipo)) %>%
        group_by(tipo,giorno=as.Date(created_at)),summarise(n=n() %>%
        engagement=sum(favorite_count+retweet_count)) %>%
        mutate(engag_tweet=engagement/n)
```

Di seguito, utilizzando la funzione `ggplot`, si costruisce il grafico dell'andamento giornaliero di *tab1*.

```
ggplot(tab1,aes(x=giorno,y=n,color=tipo))+
       geom_line()+
       scale_color_manual(values = c("darkgreen","red","blue"))+
       theme_light()
```

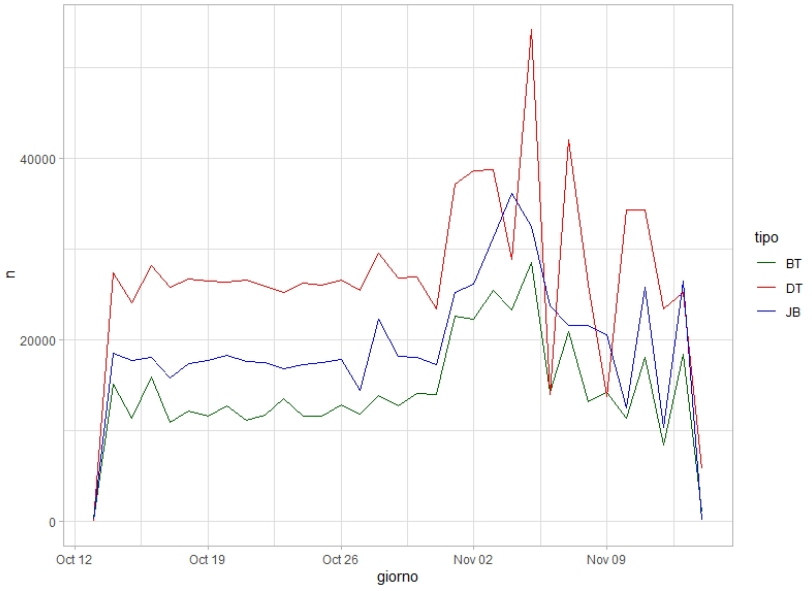


Figura 3: andamento giornaliero di *tab1*

Come si può notare, i tweet citanti Trump sono quantitativamente maggiori per tutto il periodo analizzato, con un picco di oltre 50000 tweet verificatosi il 5 novembre 2020. Da sottolineare un aumento del dato riferito a Biden in prossimità delle votazioni avvenute il 3 novembre 2020.

La funzione `ggplot` viene utilizzata anche per la raffigurazione grafica della variazione dell'engagement per tweet durante il periodo di analisi:

```
ggplot(tab1,aes(x=giorno,y=engag_tweet,color=tipo))+
        geom_line()+
        scale_color_manual(values = c("darkgreen","red","blue"))+
        theme_light()
```

Differentemente, l'engagement per tweet citanti Biden è maggiore per tutto il periodo analizzato, fatta eccezione di 3 giornate dove predominano le interazioni nei tweet con presenti entrambi i candidati; in particolar modo il primo giorno vengono superate le 40 interazioni medie per tweet.

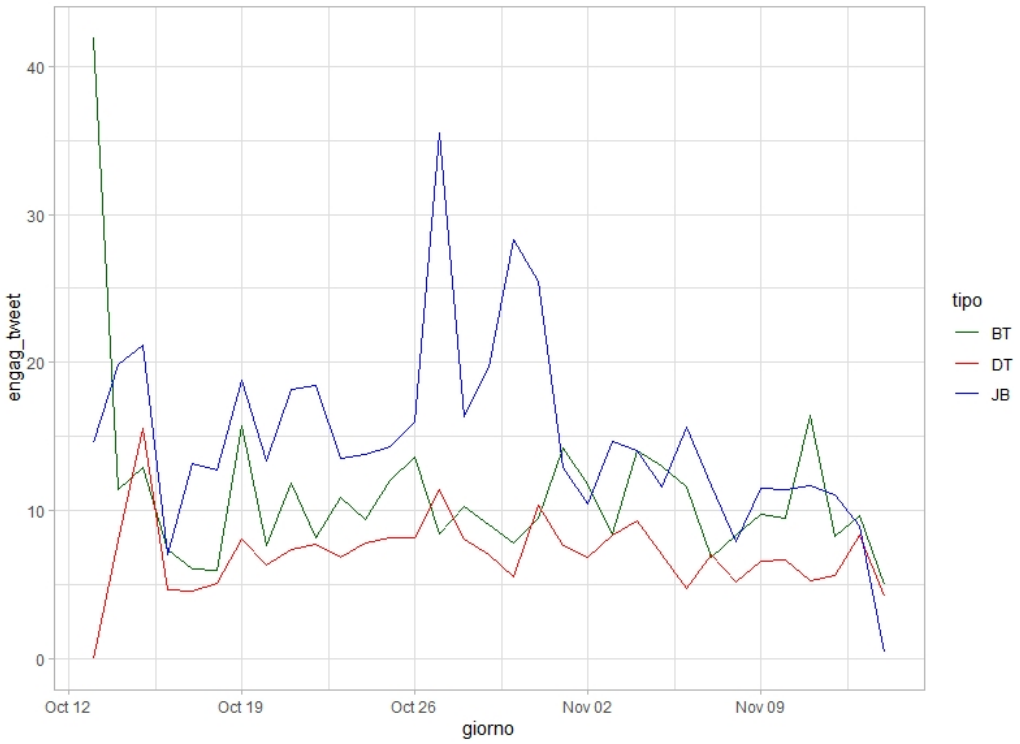


Figura 4: engagement per tweet giornaliero di *tab1*

3.3 Analisi degli hashtag

Per l'*analisi degli hashtag* si torna a considerare le osservazioni campionarie. Lo studio inizia unendo gli hashtag per `status_id`, ossia per ogni tweet si associano tutti gli hashtag all'interno del testo stesso.

```
dfHash <- dftwrrc %>%
          select(status_id,tipo,hashtags) %>%
          group_by(status_id) %>%
          mutate(allHash=paste(unlist(hashtags),collapse ="␣")) %>%
          select(status_id,tipo,allHash)
```

Si prosegue trasformando il dataframe *dfHash* ottenuto precedentemente raggruppando gli hashtag associati ai tweet per gruppo, componendo così il dataframe *dfHashGR*.

```
dfHashGR <- dfHash %>%
            select(tipo,allHash) %>%
            filter(allHash!="NA") %>%
            group_by(tipo) %>%
            summarise(hash=paste(allHash,collapse = "␣"))
```

A questo punto si crea il vettore contenente le parole, le *stop words*, che bisogna eliminare per poter diversificare i tweet. In questo caso sono *biden*, *trump*, *Biden* e *Trump*.

```
mystop <- c("biden","trump","Biden","Trump")
```

Una delle domande principali che ci si pone nel *text mining* è quella di quantificare il contenuto qualitativo del testo. Una possibile misura che determina l'importanza di una parola è la *term frequency* (tf), quanto frequentemente una parola ricorre all'interno di un testo. Attraverso la funzione *creaTDM* avviene l'eliminazione delle *stop words* e l'analisi degli hashtag con la ponderazione term frequency.

```
tdmHS <- creaTDM(dfHashGR$hash,lemmatiz = F, mystopwords = mystop)
```

Determinato ciò, si prosegue con la raffigurazione grafica tramite la già definita *nuvola di parole*.

```
wordcloud(words = tdmHS$freq.Frm$forme,freq = tdmHS$freq.Frm$freq,
          max.words = 100, random.order = F,
          colors = brewer.pal(4,"Dark2"))
```

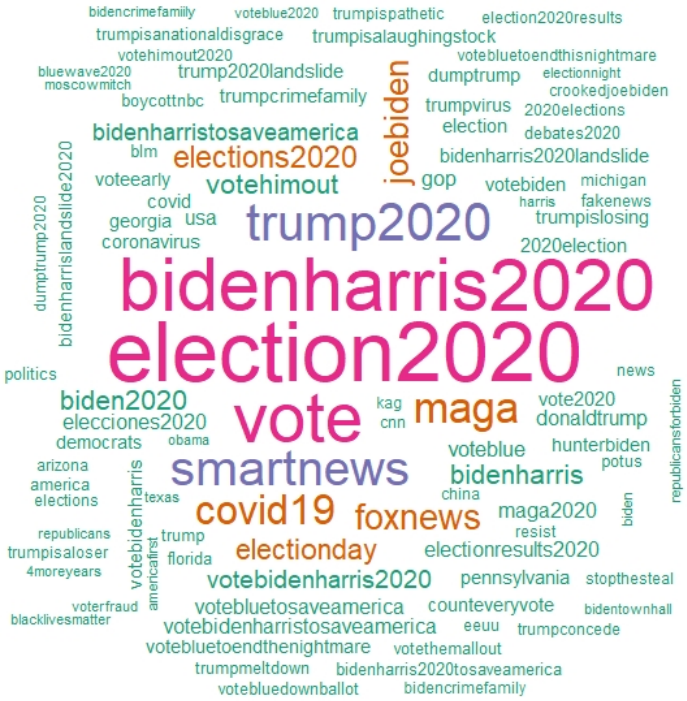


Figura 5: wordcloud ponderazione tf

Ci sono parole in un testo, comunque, che ricorrono molto spesso anche se non sono importanti, ad esempio *è*, *il*, *la* e *di*. Si può optare per aggiungere questo tipo di parole alla lista vettoriale di stop words e rimuoverle prima dell'analisi, ma potrebbe essere che alcune di queste parole siano più importanti in certi testi rispetto ad altri. Una lista di stop words non è un approccio molto sofisticato per aggiustare la term frequency per le parole più comunemente usate.

Un'altra metodologia è di considerare l' *inverse document frequency* di un termine (idf), che cala il peso delle parole più comunemente usate e incrementa per quelle meno presenti nel testo considerato. Quest'ultimo approccio può essere combinato con il term frequency per calcolare il *tf-idf* del termine (le due quantità moltiplicate fra loro), la frequenza di un termine in base a quanto raramente è usato. Come si nota in corrispondenza del peso nel comando sottostante si calcola la ponderazione tf-idf.

```
tdmHS <- creaTDM(dfHashGR$hash,lemmatiz = F,
         weight = "tfidf",mystopwords = mystop)
```

Come avvenuto nel caso precedente, si ricava la *wordcloud*.

```
wordcloud(words = tdmHS$freq.Frm$forme,
          freq = tdmHS$freq.Frm$freq, max.words = 100,
          random.order = F,colors = brewer.pal(4,"Dark2"))
```

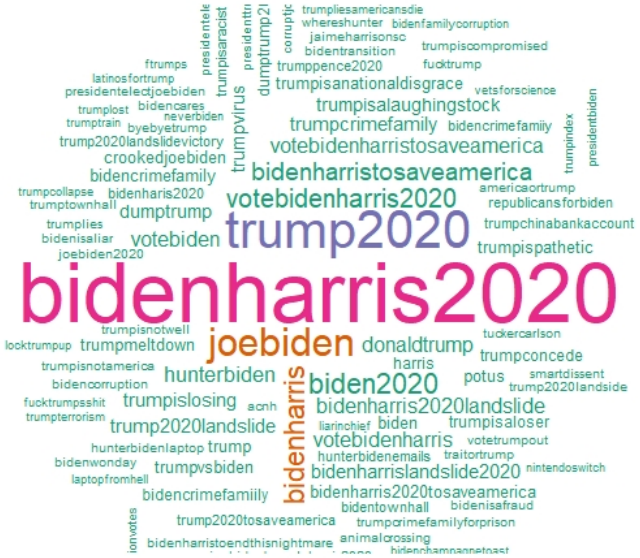


Figura 6: wordcloud ponderazione tfidf

Si termina l'analisi degli hashtag, creando una *wordcloud* comparativa fra i tre gruppi considerati (Biden Trump, Biden, Trump).

```
tdmHS <- creaTDM(dfHashGR$hash,lemmatiz = F,mystopwords = mystop)
colnames(tdmHS$tdm.Frm) <- c("Biden-Trump","Trump","Biden")
par(mar=c(0,0,0,0))
comparison.cloud(
  term.matrix = tdmHS$tdm.Frm, scale = c(3,0.3),
  max.words = 100, random.order = F,
  rot.per = 0.2, colors = c("gray20","darkred","blue"),
  match.colors = T, title.size = 2.5,  title.bg.colors = "white")
```

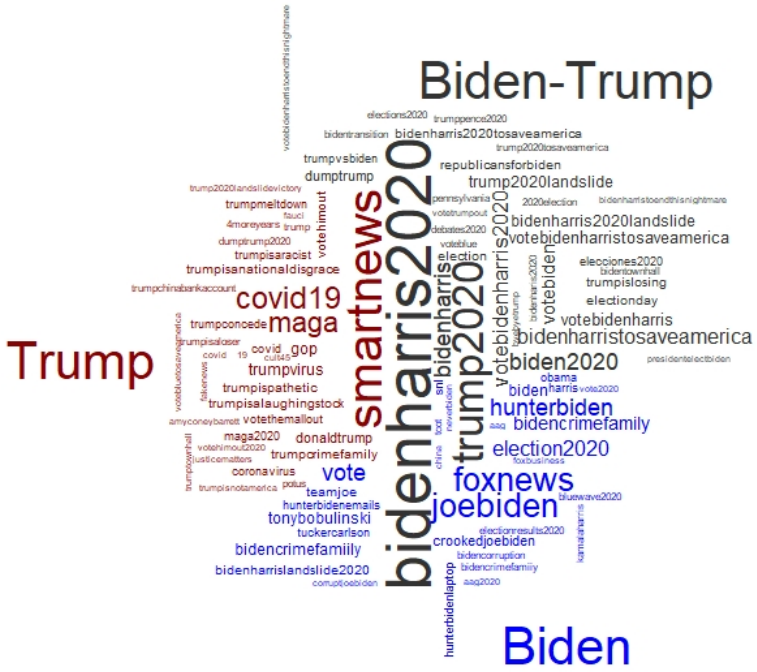


Figura 7: wordcloud comparazione candidati

É importante sottolineare gli hashtag più ricorrenti per il candidato repubblicano sono "smartnews","covid19" e "maga"(acronimo di "Make America Great Again"); per quello democratico sono "joebiden", "foxnews", "hunterbiden" e "vote", mentre nel gruppo citante entrambi si trova l'hastag più usato nel corso delle elezioni: "bidenharris2020", decisamente più ricorrente di "trump2020".

3.4 Sentiment Analysis

Con il termine *sentiment analysis*, o *open mining*, si intende uno studio che analizza le opinioni, i sentimenti, le valutazioni e le emozioni del linguaggio utilizzato in un testo. In una serie di frasi che esprimono un'opinione si possono trovare due diverse componenti: il soggetto o entità g su cui verte l'opinione e quest'ultima espressa, s. Questa tipologia di frase può essere rappresentata come una coppia di questi due valori $(g;s)$. Il valore s può esprimere sia parere positivo, negativo o neutro (definito *polarity*) oppure un valore che esprime l'intensità di tale sentimento.

Prima di tutto, si applica la sentiment analysis tramite la funzione `get_nrc_sentiment`, la quale permette utilizzando il dizionario sentiment NRC di calcolare la presenza di otto emozioni (rabbia, anticipazione, disgusto, paura, gioia, tristezza, sorpresa e fiducia) e la loro corrispettiva valenza. Ciò che si ottiene è un dataframe `syuz_sent` dove ogni riga rappresenta una frase del testo originale e ogni colonna include un valore per ciascuna emozione sia positiva che negativa.

```
syuz_sent <- get_nrc_sentiment(txtL_UV$txtL)
```

Pertanto, le ultime due colonne rappresentano il valore della *positività* e della *negatività*. Il risultato di *valence* è dato proprio dalla somma dei risultati delle ultime due colonne.

```
valence <- (syuz_sent[, 9]*-1) + syuz_sent[, 10]
```

L'istogramma evidenzia che il valore più frequente è attorno allo 0 e la media è uguale a 0.0475.

```
hist(valence)
```

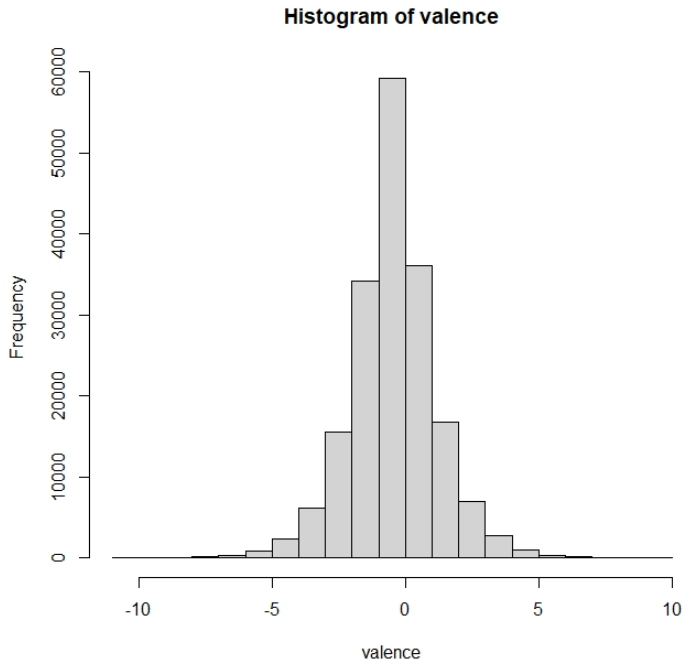


Figura 8: Istogramma di valenza

Si può prendere ad esempio il seguente tweet per comprendere il procedimento che viene eseguito: "Kamala Harris is an embarrassment. On the odd chance Biden loses it will be largely because of the ineptitude of Harris." (Kamala Harris è imbarazzante. Nella difficile possibilità che Biden perda, sarà in gran parte a causa dell'inettitudine di Harris). A questo viene attribuito un valore di negatività uguale a 4 e di positività di 1, portando il valore di *valence* a -3 contro Joe Biden.

A questo punto, si applica la sentiment analysis nel tempo sfruttando il pacchetto di linguaggio su R `syuzhet`. Si inizia con la costruzione del dataframe *dfvalence*.

```
dfvalence <- data.frame(id=txtL_UV2$doc_id,valence)
```

Il sucessivo dataframe *tmp2* si fonda sull'unione fra il dataframe campionario *dftwrrc* e *dfvalence*.

```
tmp2 <- inner_join(dftwrrc %>%
                     select(id,created_at,tipo) %>%
                     mutate(giorno=as.Date(created_at)) %>%
                     rename(tempo=created_at),
                     dfvalence %>%
                     rename(sentiment=valence),by="id")
```

Il dataframe *tmp2* è necessario per creare un grafico dove si è posto nell'asse x il tempo e nell'y il sentiment.

```
tmp2 %>%
  group_by(giorno) %>%
  summarise(sentim=mean(sentiment)) %>%
  ggplot(aes(x=giorno,y=sentim))+
  geom_line(color="blue")
```

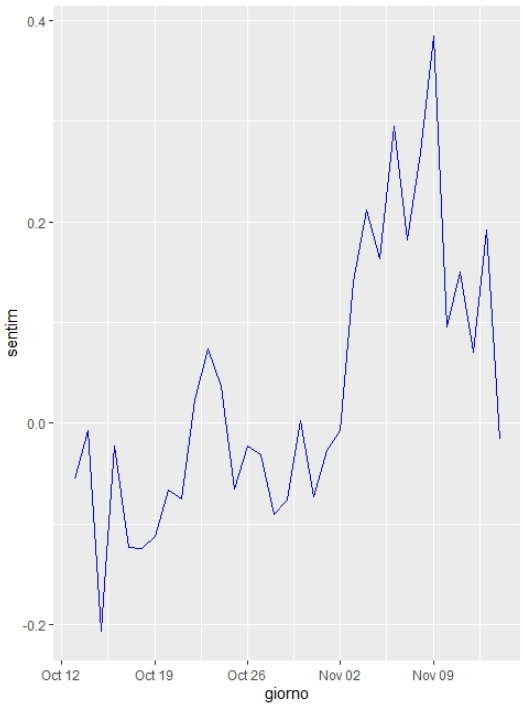


Figura 9: Sentiment analysis per giorno

Come si può notare nella figura 9, il grafico si caratterizza per una iniziale depressione che supera il valore -0.2, per poi con l'avvicinarsi alla neutralità nei giorni imminenti all'*election day*. Pertanto, dal 3 novembre, avviene, sostanzialmente, una crescita fino a quasi toccare il picco di 0.4 il 9 novembre 2020; gli ultimi giorni dello studio invece hanno evidenziato un calo tanto da portare l'andamento per giorno in prossimità della neutralità.

Ora si decide di separare *tmp2* in gruppi (Biden Trump, Trump, Biden) per verificare la sentiment analysis per giorno per gruppi con tanto di grafico.

```
tmp2 %>%
  group_by(giorno,tipo) %>%
  summarise(sentim=mean(sentiment)) %>%
  ggplot(aes(x=tempo,y=sentim))+
  geom_line(color="blue")+
  facet_wrap(~tipo)
```

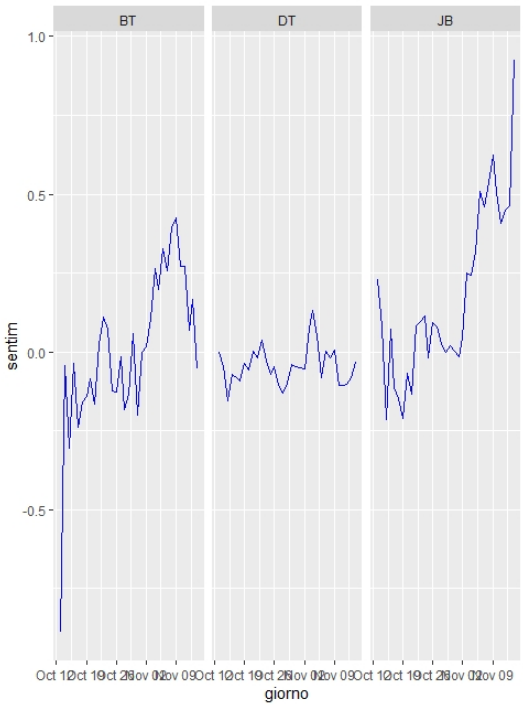


Figura 10: Sentiment analysis per giorno per gruppi

Nel primo grafico, inerente al gruppo Biden Trump, si nota un iniziale valore minimo equivalente a circa -0.8, per poi, tranne che per due occasioni, assumere valori comunque negativi all'interno dell'intervallo -0.25 e 0. Il valore minimo avviene sempre il 16 ottobre misurando -0.16 e quello massimo, il 2 novembre, che sfiora lo 0.3. Dal 3 novembre ha un comportamento settimanale molto simile a quello precedente (Figura 10).

Nel secondo grafico su Donald Trump, si evidenziano valori esclusivamente neutrali o negativi, con eccezion fatta a cavallo delle elezioni in cui viene raggiunto un valore massimo circa di 0.1.

A proposito del grafico di Joe Biden, si ha una situazione frastagliata in coincidenza della prima settimana in cui assume valori tra il -0.25 e il 0.25. Come evidenziato in precedenza, dalla data delle elezioni, si ha una costante crescita, che in questo specifico caso persiste fino al termine dello studio avvicinandosi al picco di 1.0.

Dopo aver completato ciascuna forma di sentiment analysis per giorno, si prosegue calcolando quella per settimana.

```
tmp2 %>%
  group_by(settimana=round_time(tmp2$tempo,"week")) %>%
  summarise(sentim=mean(sentiment)) %>%
  ggplot(aes(x=settimana,y=sentim))+
  geom_line(color="blue")
```

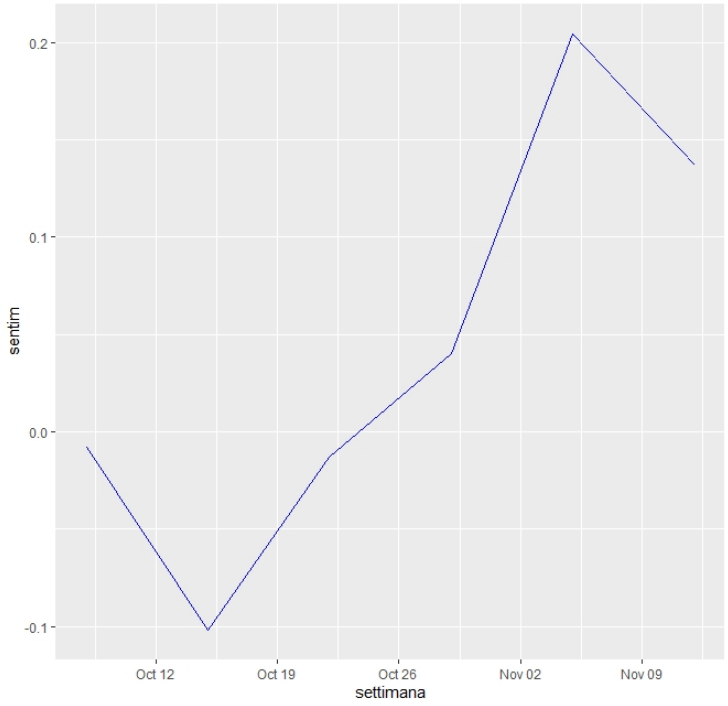


Figura 11: Sentiment per settimana

Nel periodo corrispondente alla prima settimana il grafico evidenzia una depressione, tanto da misurare il risultato più basso il 16 ottobre con -0.1. Da questo punto non smette di crescere, ottenendo un valore positivo a fine ottobre e la settimana successiva arriva a misurare 0.2. Nell'ultima settimana avviene un calo che fa terminare l'analisi poco sotto il valore di 0.15.

Come avvenuto anche per la precedente analisi nel tempo, anche per la sentiment analysis per settimana si applica la suddivione per gruppi.

```
tmp2 %>%
  group_by(settimana=round_time(tempo,"week"),tipo) %>%
  summarise(sentim=mean(sentiment)) %>%
  ggplot(aes(x=settimana,y=sentim))+
  geom_line(color="blue")+
  facet_wrap(~tipo)
```

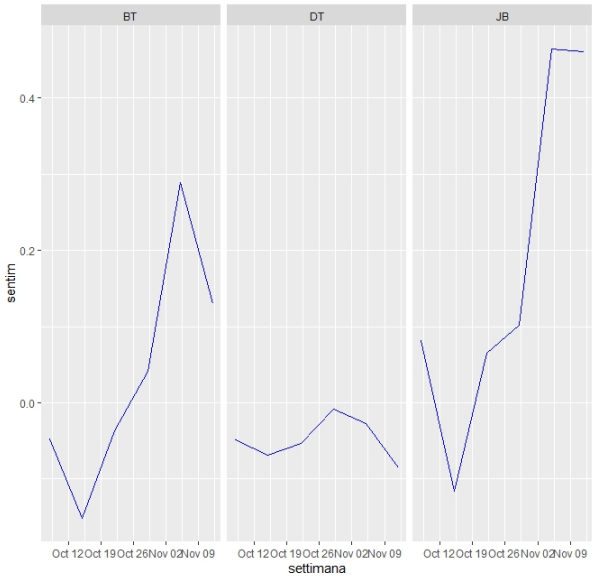


Figura 12: Sentiment per settimana per gruppi

Il primo grafico, sul gruppo Biden Trump, ha un comportamento settimanale molto simile a quello precedente (Figura 11). Il valore minimo avviene sempre il 16 ottobre misurando -0.16 e quello massimo, il 2 novembre, che sfiora lo 0.3.

Proseguendo col secondo grafico a proposito di Donald Trump, si evidenziano valori esclusivamente negativi di cui quello più basso risale all'ultima settimana poco sopra il -0.1.

Parlando del grafico di Joe Biden, si ha una depressione negativa in coincidenza della prima settimana fino ad arrivare a -0.1. In seguito, cresce e raggiunge il picco il 6 novembre con circa lo 0.46.

Il passo successivo consiste nell'interpolazione dell'andamento della sentiment analysis. Con interpolazione, in questo caso, si intende un metodo per individuare nuovi punti del piano cartesiano a partire da un insieme finito di punti conosciuti, nell'ipotesi che tutti i punti si possano riferire ad una funzione f(x), *valence*, di una data famiglia di funzioni di una variabile reale.

```
simple_plot(valence,
            title="Interpolazione␣dell'andamento␣della␣sentiment")
```

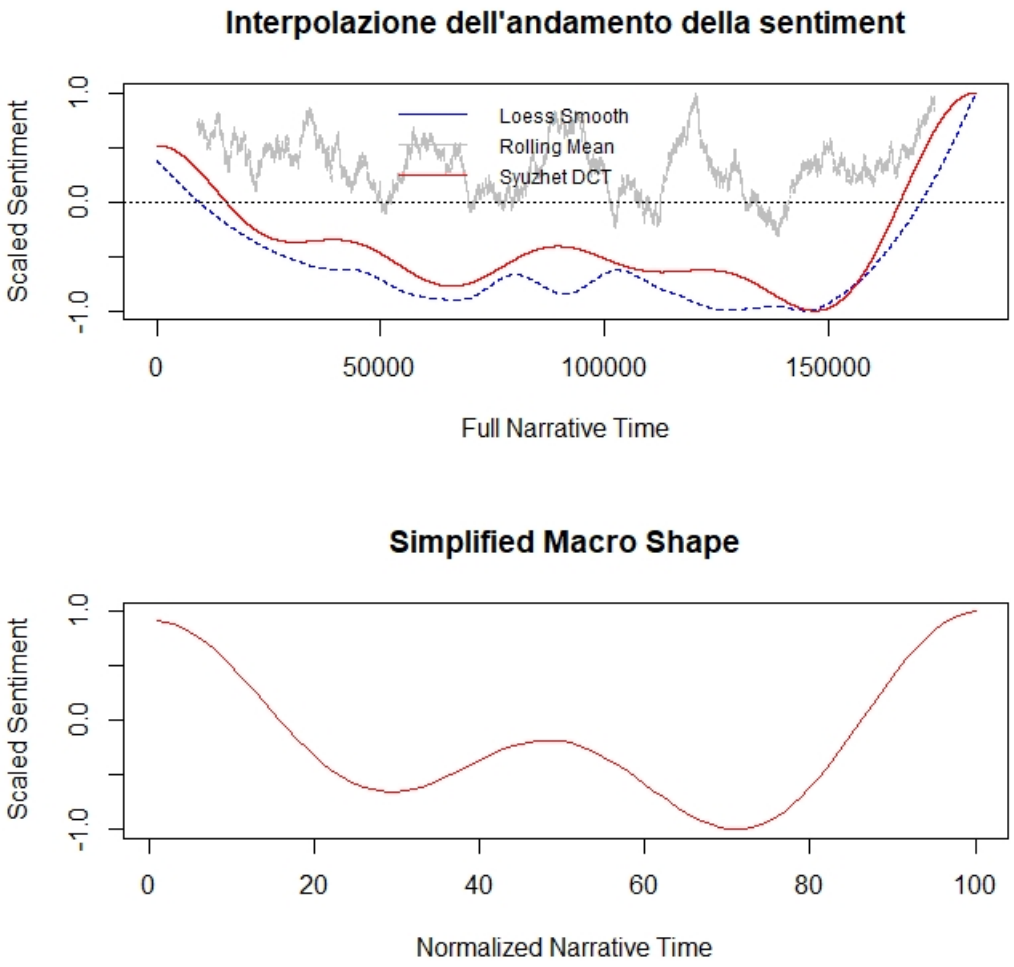


Figura 13: Interpolazione dell'andamento della sentiment analysis

Si conclude la sentiment analysis calcolandola per gruppi di 20 tweet.

```
percent_vals <- get_percentage_values(valence, bins = 50)
ggplot(mapping=aes(x=seq_along(percent_vals),y=percent_vals))+
        geom_line(col="blue")+
        theme_light()+
        ylab("Emotional Valence") +
        xlab("Narrative time") +
        ggtitle("Andamento del sentiment per gruppi di 20 tweet")
```

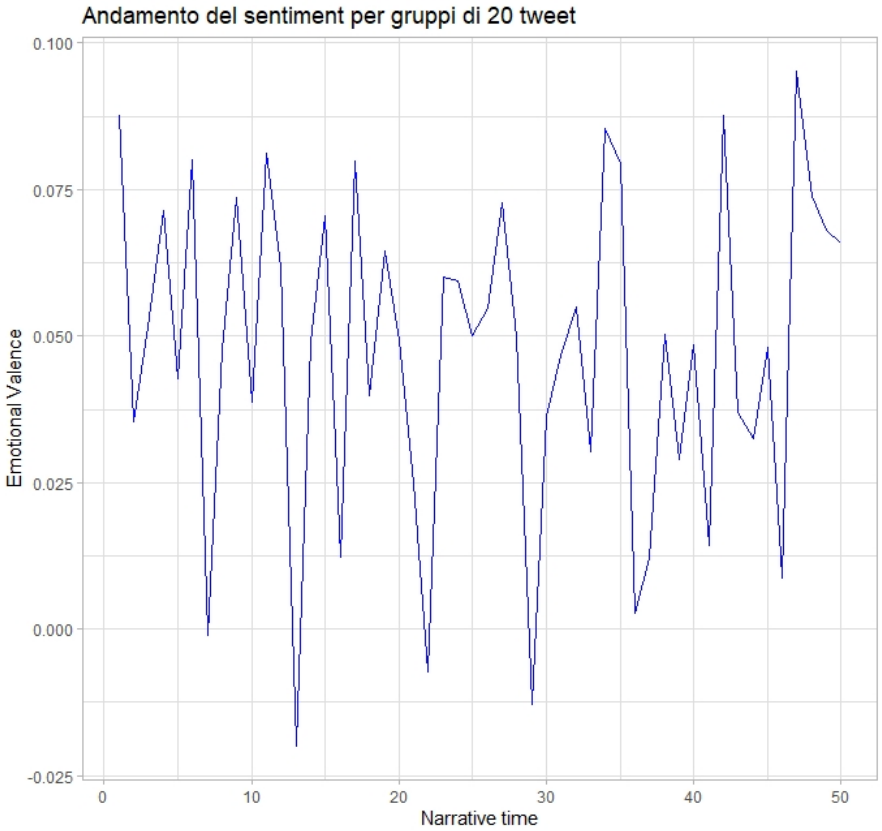


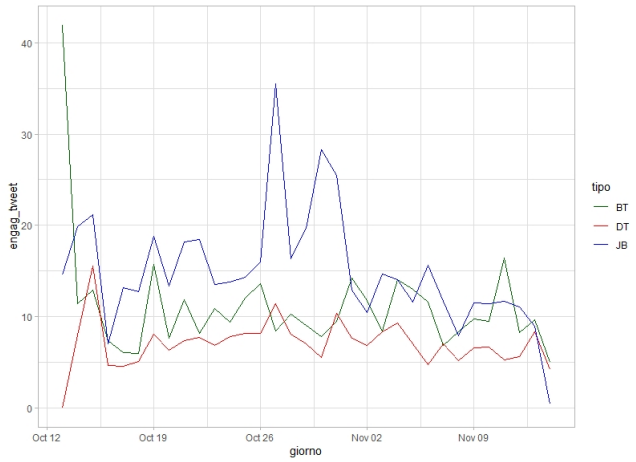
Figura 14: andamento sentiment per gruppi di 20 tweet

Il grafico si caratterizza in balzi che variano da un minimo di -0.02 ad un picco di circa 0.093.

4 Discussione dei risultati

Al fine di rispondere alla domanda posta all'inizio di questo studio, si è prima condotto un'analisi quantitativa che ha dimostrato, tramite grafici (Figura 3), che Trump è il candidato più citato, mentre Biden quello con maggiori interazioni (Figura 4).

(a)



(b)

Figura 15: andamento ed engagement giornaliero di *tab1*

Successivamente, è stata eseguita l'analisi qualitativa, la quale si divide in analisi degli hashtag e in sentiment analysis. La prima ha portato alla composizione di una nuvola di parole fondata sulla comparazione tra i candidati.

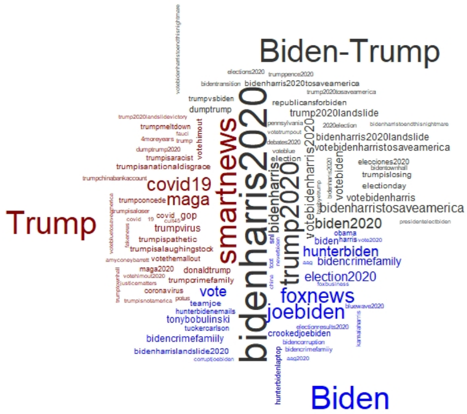


Figura 16: wordcloud comparazione candidati

La sentiment analysis, invece ha definito come l'opinione degli elettori fosse decisamente positiva nei confronti di Joe Biden e leggermente negativa in quelli di Donald Trump (Figura 12).

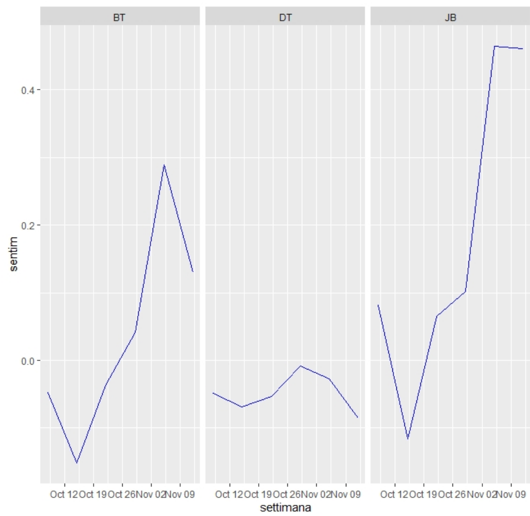


Figura 17: Sentiment per settimana per gruppi

5 Conclusioni

In una società sempre più alla ricerca del comfort, e impegnata ad evitare il peso delle difficoltà, i sentimenti hanno preso il sopravvento sulle ideologie. Sempre più spesso le scelte politiche sembrano dettate da emozioni momentanee e i social sono vettori formidabili di questi sentimenti. Grazie alle loro potenzialità possiamo utilizzare il software statistico R e la sentiment analysis come un termometro per misurare il variare dell'opinione pubblica nel susseguirsi degli eventi.

Per quanto riguarda una possibile futura analisi, potrebbe trarre interesse applicare le tecniche usate in questo studio alle prossime elezioni giapponesi, visto l'interesse mondiale che riscuote il paese nipponico per la sua importanza economica, la terza a livello mondiale, ma anche perchè spesso è stato pioniere di dinamiche poi verificatesi altrove: dallo sgonfiamento di bolle immobiliari e borsistiche all'introduzione di politiche monetarie di allentamento quantitativo (otto anni prima che negli Usa o in Gran Bretagna).

Il 3 settembre Yoshihide Suga, Primo Ministro giapponese, ha annunciato le proprie dimissioni dichiarando che non sarebbe stato in grado di seguire con la necessaria energia sia le misure anti Covid-19 che la campagna elettorale in vista delle imminenti elezioni. La decisione è avvenuta in seguito al calo del sostegno pubblico nei suoi confronti a causa della gestione della pandemia di coronavirus e delle Olimpiadi di Tokyo. L'addio di Suga è arrivato dopo appena un anno dal suo insediamento alla guida del partito Ldp, dopo le dimissioni per motivi di salute di Abe, che aveva ridato stabilità politica al Giappone con un mandato durato otto anni.

Il Giappone è il secondo mercato più grande di Twitter a livello globale, dietro agli Stati Uniti e seguito dall'India. Con oltre 50 milioni di utenti, il 45% della popolazione totale, Twitter è la seconda piattaforma di social media dominante in Giappone, dopo YouTube; ci si può aspettare, quindi, una forte partecipazione dei cittadini con tweet di natura opinionistica circa i principali candidati.

6 Bibliografia

E-book

- Silge J., Robinson D., *Text Mining with R. A Tidy Approach*, O'Reilly Media, 2017, capitoli II,III, consultato il 08/07/2021

Articoli da Internet

- Affde, *Quante persone usano Twitter nel 2021?*,
  data articolo 2/07/2021, consultato il 28/08/2021,
  https://www.affde.com/it/twitter-users.html
- Kamps H.J., *Who Are Twitter's Verified Users?*,
  data articolo 25/05/2015, consultato il 28/08/2021,
  https://haje.medium.com/who-are-twitter-s-verified-users-af976fc1b032
- rdrr.io, *visNGram: Funzione per visualizzare n-grammi*, consultato il 10/07/2021
  https://rdrr.io/github/PaoloDalena/banchedatiz/man/visNGram.html
- RDocumentation, *get-nrc-sentiment: Get Emotions and Valence from NRC Dictionary*, consultato il 10/07/2021, urly.it/3fpax
- Carrer S., Il Sole 24 Ore, *Elezioni in Giappone, ecco perché sono importanti*, data articolo 14/12/2012, consultato il 12/09/2021, urly.it/3fgt2
- Il Sole 24 Ore, *Giappone, si dimette il premier Suga e la Borsa di Tokyo festeggia*, data articolo 02/09/2021, consultato il 12/09/2021, urly.it/3fgt3
- Miranda R, *Tsunami politico in Giappone. L'addio di Suga (per colpa del Covid)*, data articolo 03/09/2021, consultato il 13/09/2021
  https://formiche.net/2021/09/giappone-dimissioni-primo-ministro-suga/
- BigBeat, *Japan's Top Social Media Platforms in 2021: how Covid-19 pandemic has changed Japan Social Media landscape*,
  data articolo 24/02/2021, consultatio il 14/09/2021, urly.it/3fg_y

Documenti accademici

- Aria M., Università degli Studi di Napoli Federico II, *Il concetto di interpolazione*, consultato il 06/07/2021, http://www.federica.unina.it/economia/metodi-quantitativi-modulo-statistica/interpolazione-statistica-retta-regressione/
- Drudi I., Università di Bologna, *Introduzione sentiment analysis*, 2020, consultato il 05/07/2021
- Drudi I., Università di Bologna, *Sentiment analysis*, 2020, consultato il 05/07/2021

7 Ringraziamenti

Vorrei ringraziare calorosamente il mio relatore, prof. Ignazio Drudi, per avermi guidato nella stesura di questa tesi e il prof. Fabrizio Alboni, il quale mi è stato di grande aiuto nell'elaborazione pratica dello script utilizzato per lo studio effettuato.