Alberto Fernández sentiment analysis

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

Análisis de sentimiento - AF scrap por M.O.

Ante todo, cargamos las librerías.

Luego, definimos un tema para utilizar en nuestros gráficos.

```
tema_graf <-
  theme_minimal() +
  theme(text = element_text(family = "serif"),
        panel.grid.minor = element_blank(),
        strip.background = element_rect(fill = "#EBEBEB", colour = NA),
        legend.position = "none",
        legend.box.background = element_rect(fill = "#EBEBEB", colour = NA))
```

Cargamos el archivo con los tuits de Alberto Fernández y la librería para cargar archivos de Excel. Separamos el tweet entre el texto y el usuario, y limpiamos las comillas primeras y últimas que marcaban el comienzo y el fin de tweet.

```
library("readxl")
af <- read_excel("af.xlsx") %>%
  tbl_df()# %>%
#  separate("Tweet Text", into = c("usuario", "texto"), sep = ":")

af <- rename(af, texto="Tweet Text")
#af$user <- NULL
#af$usuario = sub('.', '', af$usuario)
af$texto = sub('.$', '', af$texto)
```

```
af <- tibble::rowid_to_column(af, "ID")
```

Haremos una copia de archivo para trabajar sólo con frecuencias. (modelo:https://rpubs.com/HAVB/tangos)

```
af_frec <- data.frame(af)
```

Extraemos tokens, eliminamos stopwords y calculamos frecuencias.

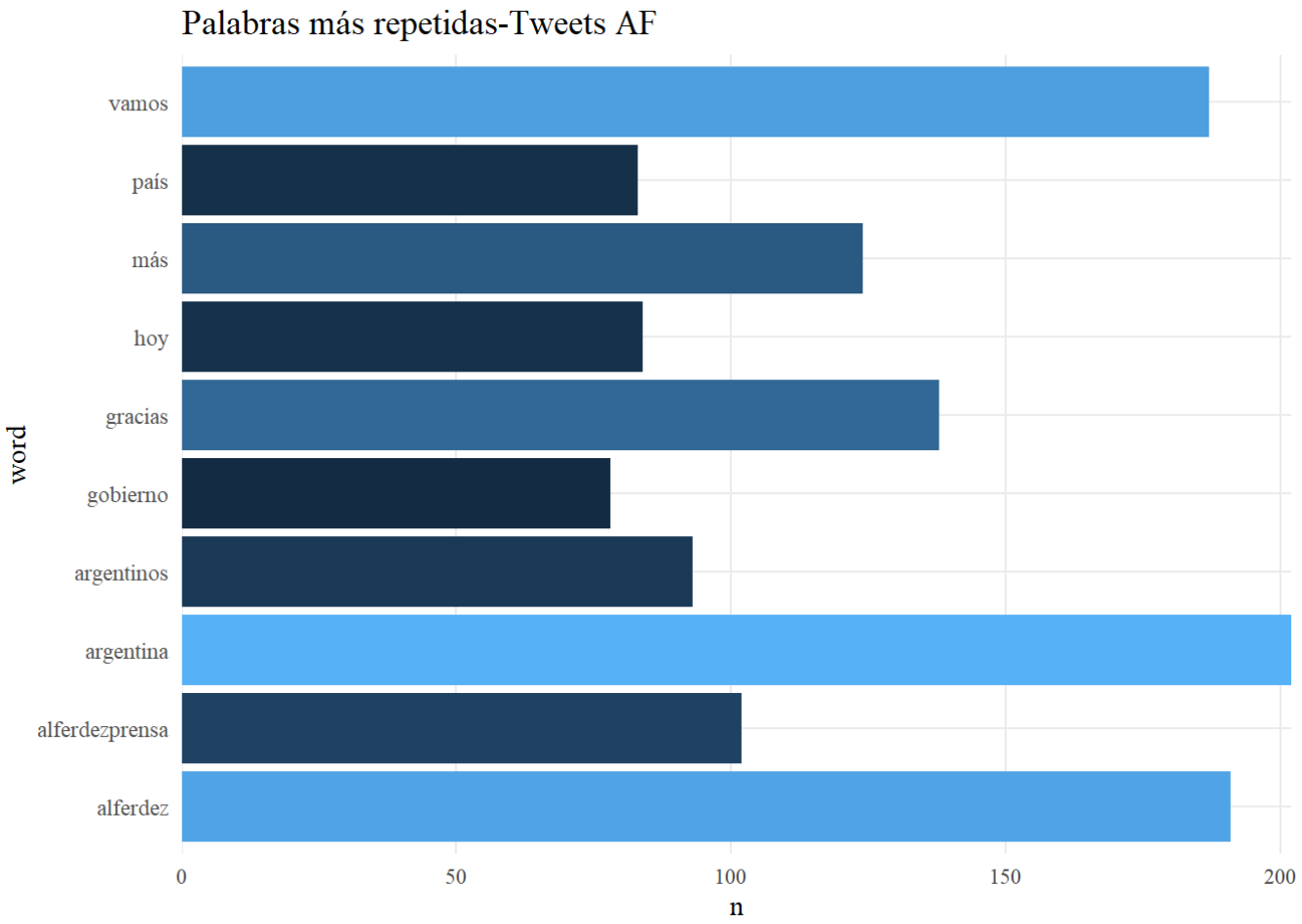
```
af_tokenizado <- af_frec %>% unnest_tokens(word, texto)

stopwords_es <- read.csv("https://bitsandbricks.github.io/data/stopwords_es.csv",stringsAsFactors = FALSE)

stopwords_es <- append(stopwords_es$STOPWORD, c("rt","t.co", "n", "hola", "https","http","q"))%>% tbl_df()

af_tokenizado <- af_tokenizado %>%
   anti_join(stopwords_es, by = c("word" = "value"))

af_tokenizado %>%
  count(word, sort = TRUE)%>%
  top_n(n = 10, wt = n)%>%
  ggplot() +
  aes(word, n, fill=n) +
  geom_col()+
  scale_y_continuous(expand = c(0, 0)) +coord_flip()+
  tema_graf+ggtitle("Palabras más repetidas-Tweets AF")
```



Descargamos el léxico Afinn. Este es un conjunto de palabras, puntuadas de acuerdo a qué tan positivamente o negativamente son percibidas. Las palabras que son percibidas de manera positiva tienen puntuaciones de -4 a -1; y las positivas de 1 a 4.

La versión que usaremos es una traducción automática, de inglés a español, de la versión del léxico presente en el conjunto de datos sentiments de tidytext, con algunas correcciones manuales. Por supuesto, esto quiere decir que este léxico tendrá algunos defectos, pero será suficiente para nuestro análisis. (Más en https://rpubs.com/jboscomendoza/analisis_sentimientos_lexico_afinn)

```
download.file("https://raw.githubusercontent.com/jboscomendoza/rpubs/master/sentimientos_afinn/lexico_afinn.en.es.csv",
              "lexico_afinn.en.es.csv")
```

```
afinn <- read.csv("lexico_afinn.en.es.csv", stringsAsFactors = F, fileEncoding = "latin1") %>%
  tbl_df()
```

Tokenizamos tweets, juntamos aquellas palabras del diccionario que estén presente, y luego creamos otra columna de “positiva” o “negativa” según la puntuación que da el diccionario de Afinn.

```
af_afinn <- af %>%
  unnest_tokens(input ='texto', output = "Palabra") %>% #tokenizamos
  inner_join(afinn, ., by = "Palabra") %>%
  mutate(Tipo = ifelse(Puntuacion > 0, "Positiva", "Negativa"))
```

Agrupamos la media de la puntuación por cada tuit.

```
af <-
  af_afinn %>%
  group_by(ID) %>%
  summarise(Puntuacion_tuit = mean(Puntuacion)) %>%
  left_join(af, ., by = "ID") %>%
  mutate(Puntuacion_tuit = ifelse(is.na(Puntuacion_tuit), 0, Puntuacion_tuit))
```

Vemos el total de palabras únicas que quedaron luego de tokenizar y comparar con el diccionario.

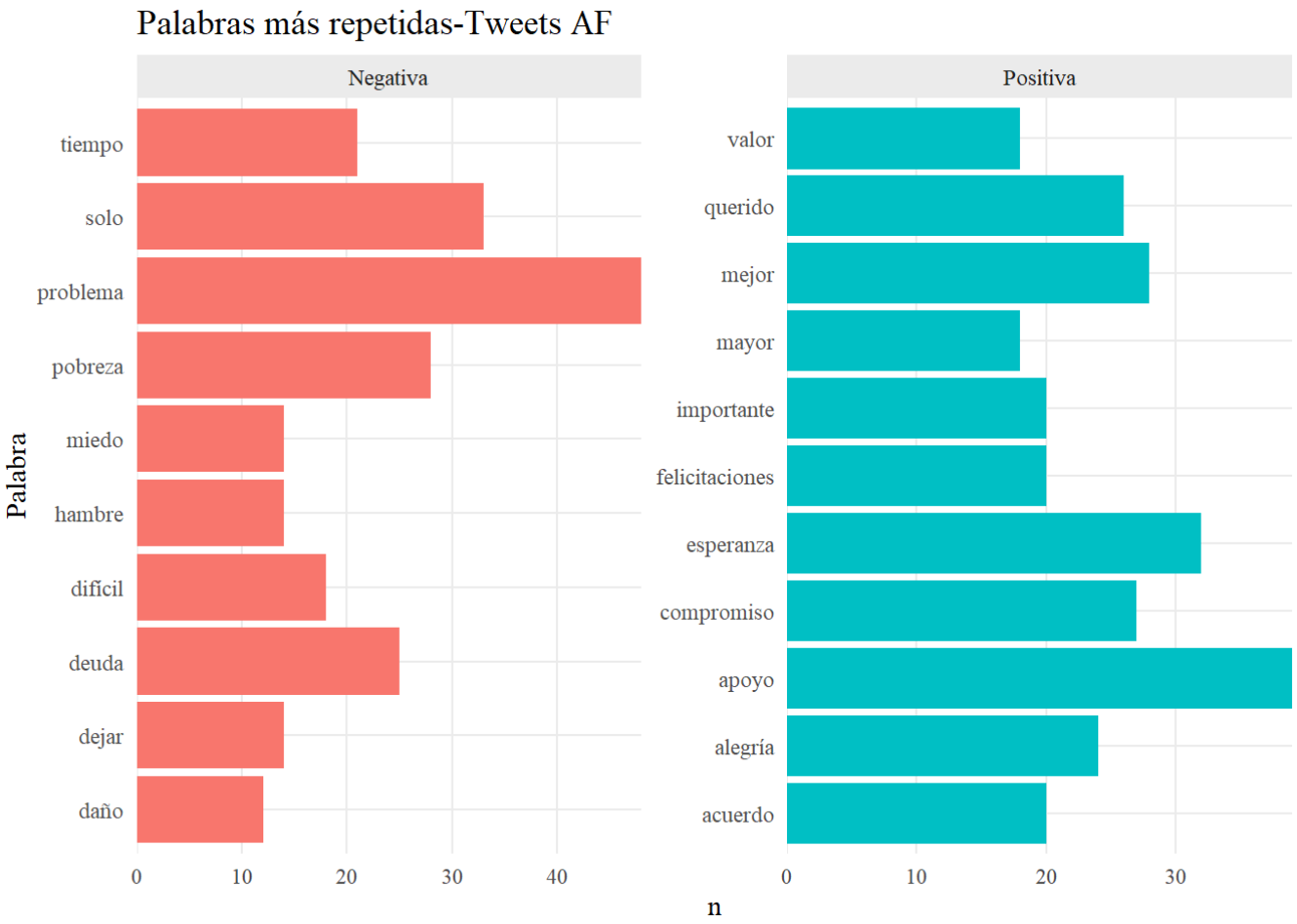
```
af_afinn %>%
  group_by(Tipo) %>%
  distinct(Palabra) %>%
  count()
```

```
## # A tibble: 2 x 2
## # Groups:   Tipo [2]
##   Tipo         n
##   <chr>    <int>
## 1 Negativa   143
## 2 Positiva   157
```

Ahora, graficamos las primeras diez palabras positivas y negativas más repetidas.Filtramos la palabra “no” y "gracias.

```
af_afinn <-
  af_afinn %>%
  filter(Palabra != "no") %>%
  filter(Palabra != "gracias")

af_afinn %>%
  group_by(Tipo) %>%
  count(Palabra, sort = T) %>%
  top_n(n = 10, wt = n) %>%
  ggplot() +
  aes(Palabra, n, fill=Tipo) +
  geom_col() +
  facet_wrap("Tipo", scales = "free") +
  scale_y_continuous(expand = c(0, 0)) +
  coord_flip()+
  tema_graf+ggtitle("Palabras más repetidas-Tweets AF")
```



Ahora probamos con otro diccionario: SDAL

Volvemos el archivo con los tuits de AF y hacemos la limpieza necesaria.

```
library("readxl")
af_v2 <- read_excel("af.xlsx") %>%
  tbl_df()

af_v2 <- rename(af_v2, texto="Tweet Text")
```

```
af_v2$texto = sub('.$', '', af$texto)

af_v2 <- tibble::rowid_to_column(af_v2, "ID")
```

léxico SDAL

Para asignar un “sentimiento” a cada palabra utilizaremos SDAL, un léxico de 2880 palabras. El SDAL fue producido por el Grupo del Procesamiento del Habla de la Facultad de Ciencias Exactas y Naturales (FCEyN), parte de la Universidad de Buenos Aires.

Las palabras contenidas en el léxico han sido “puntuadas” manualmente asignando su valor según tres dimensiones afectivas:

-agrado (agradable / neutra / desagradable) -activación (activa / neutra / pasiva) -imaginabilidad (fácil de imaginar / neutra / difícil de imaginar)

```
sdal <- read.csv("https://bitsandbricks.github.io/data/sdal.csv",
                 stringsAsFactors = F, fileEncoding = "utf-8")
sdal <- rename(sdal, "Palabra" = palabra)
```

Mismo procesdimiento que antes. Tokenizamos y concatenamos las palabras y puntuaciones.

```
af_sdal <- af_v2 %>%
  unnest_tokens(input ='texto', output = "Palabra") %>% #tokenizamos
  inner_join(sdal, ., by = "Palabra") %>%
  mutate(Tipo_agrado = ifelse(media_agrado > 0, "Positiva", "Negativa"),
         Tipo_activacion = ifelse(media_activacion > 0, "Positiva", "Negativa"),
         Tipo_imaginabilidad = ifelse(media_imaginabilidad>0,"Positiva","Negativa"))
```

Agrupamos la media de la puntuación por cada tuit.

```
af_v2 <-
  af_sdal %>%
  group_by(ID) %>%
  summarise(agrado_tuit = mean(media_agrado),
            activacion_tuit = mean(media_activacion),
            imaginabilidad_tuit = mean(media_imaginabilidad)) %>%
  left_join(af_v2, ., by = "ID") %>%
  mutate(agrado_tuit = ifelse(is.na(agrado_tuit), 0, agrado_tuit),
```

```
        activacion_tuit = ifelse(is.na(activacion_tuit), 0, activacion_tuit),
        imaginabilidad_tuit=ifelse(is.na(imaginabilidad_tuit),0,imaginabilidad_tuit))
```

Vemos el total de palabras únicas que quedaron luego de tokenizar y comparar con el diccionario.

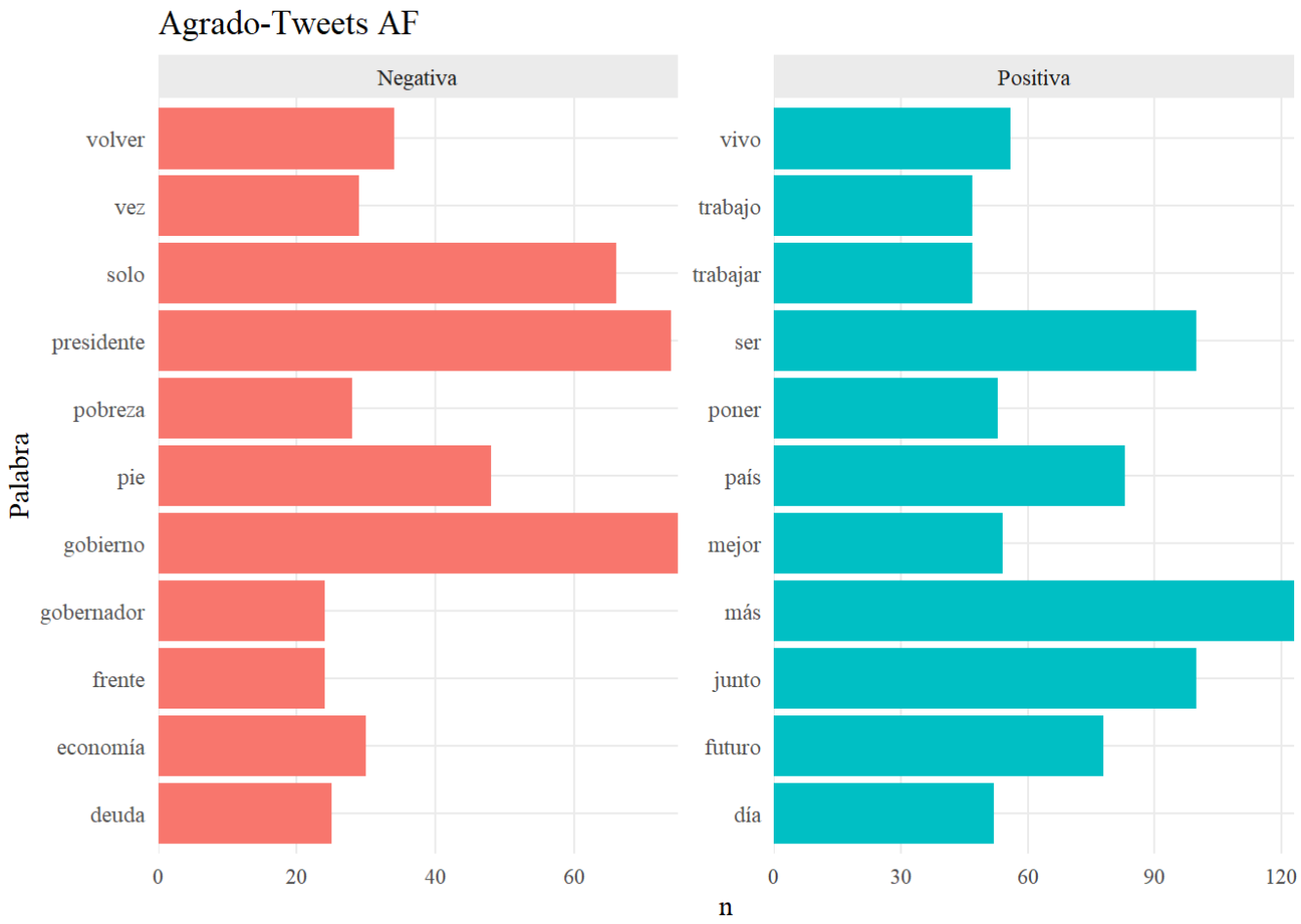
```
af_sdal %>%
  group_by(Tipo_agrado) %>%
  distinct(Palabra) %>%
  count()
```

```
## # A tibble: 2 x 2
## # Groups:   Tipo_agrado [2]
##   Tipo_agrado     n
##   <chr>       <int>
## 1 Negativa      327
## 2 Positiva      609
```

Ahora, graficamos las primeras diez palabras positivas y negativas más repetidas.Filtramos la palabra “no” y "gracias.

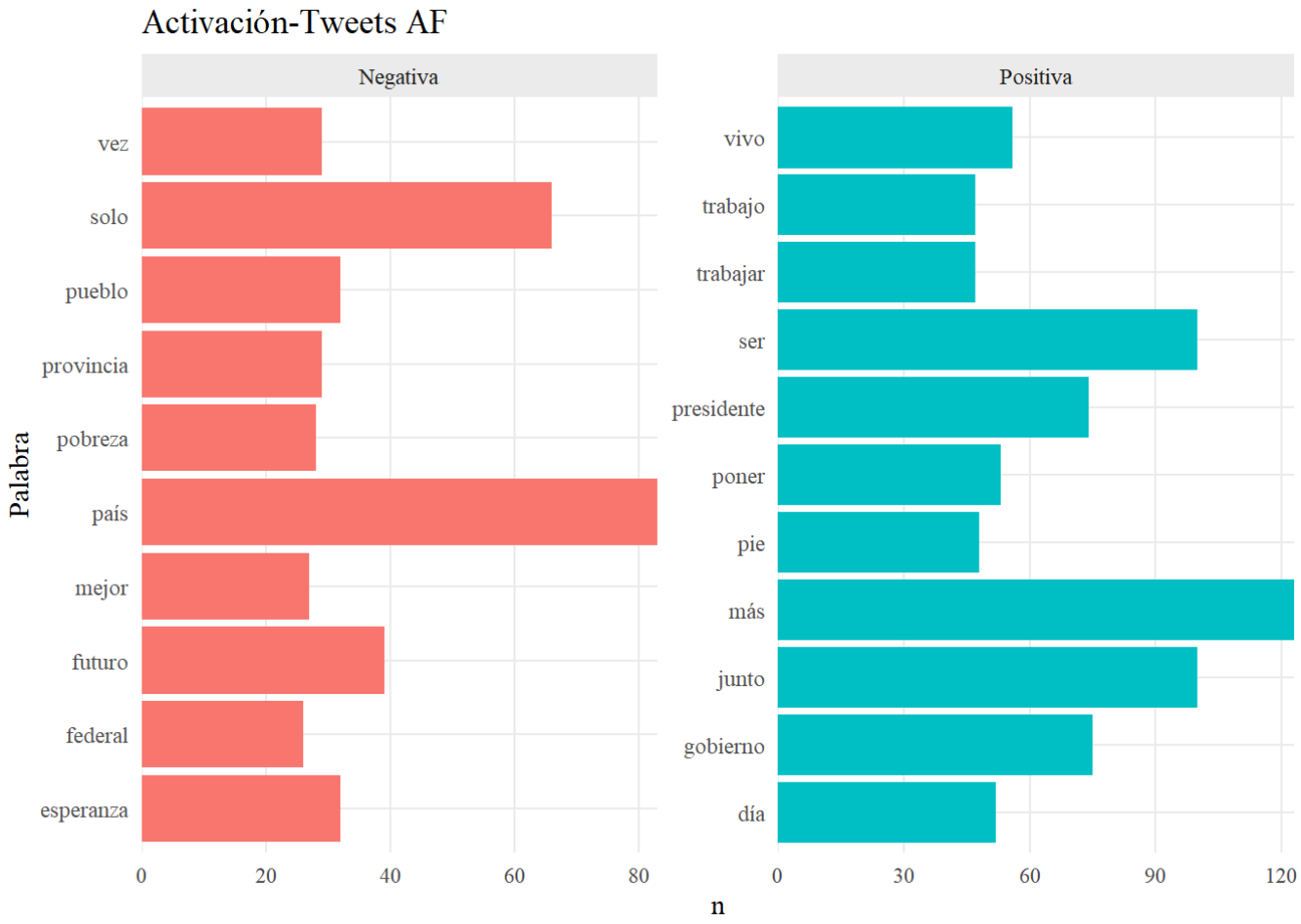
```
af_sdal_filtrado <-
  af_sdal# %>%
  #filter((Palabra != "corte")&(Palabra != "luz")&(Palabra != "ser"))

af_sdal_filtrado %>%
  group_by(Tipo_agrado) %>%
  count(Palabra, sort = T) %>%
  top_n(n = 10, wt = n) %>%
  ggplot() +
  aes(Palabra, n, fill=Tipo_agrado) +
  geom_col() +
  scale_y_continuous(expand = c(0, 0)) +
  facet_wrap("Tipo_agrado", scales = "free") +
  coord_flip()+
  tema_graf+ggtitle("Agrado-Tweets AF")
```




```
af_sdal_filtrado %>%
  group_by(Tipo_activacion) %>%
  count(Palabra, sort = T) %>%
  top_n(n = 10, wt = n) %>%
  ggplot() +
  aes(Palabra, n, fill=Tipo_activacion) +
  geom_col() +
  scale_y_continuous(expand = c(0, 0)) +
  facet_wrap("Tipo_activacion", scales = "free") +
```

```
  coord_flip()+
  tema_graf+ggtitle("Activación-Tweets AF")
```




```
af_sdal_filtrado %>%
  group_by(Tipo_imaginabilidad) %>%
  count(Palabra, sort = T) %>%
  top_n(n = 10, wt = n) %>%
  ggplot() +
  aes(Palabra, n, fill=Tipo_imaginabilidad) +
```

```
  geom_col() +
  scale_y_continuous(expand = c(0, 0)) +
  facet_wrap("Tipo_imaginabilidad", scales = "free") +
  coord_flip()+
  tema_graf+ggtitle("Imaginabilidad-Tweets AF")
```