604 Data Driven Story Telling - Citizen Connect Word Cloud

true

2022-08-15

Load Libraries:

```
#all necessary libraries here

library(rtweet)
library(twitteR)
```

```
##
## Attaching package: 'twitteR'

## The following object is masked from 'package:rtweet':
##
##     lookup_statuses
```

```
library(leaflet)
library(quanteda)
```

```
## Package version: 3.2.1
## Unicode version: 13.0
## ICU version: 69.1

## Parallel computing: 4 of 4 threads used.

## See https://quanteda.io for tutorials and examples.
```

```
library(readr)
library(httr)
library(tidytext)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v dplyr   1.0.8
## v tibble  3.1.6     v stringr 1.4.0
## v tidyr   1.2.0     v forcats 0.5.1
## v purrr   0.3.4
```

```
## -- Conflicts ---------------------------------------- tidyverse_conflicts() --
## x dplyr::filter()   masks stats::filter()
## x purrr::flatten()  masks rtweet::flatten()
## x dplyr::id()       masks twitteR::id()
## x dplyr::lag()      masks stats::lag()
## x dplyr::location() masks twitteR::location()
```

```
library(quanteda.textmodels)
library(tm)
```

```
## Loading required package: NLP
```

```
##
## Attaching package: 'NLP'
```

```
## The following object is masked from 'package:ggplot2':
##
##     annotate
```

```
## The following object is masked from 'package:httr':
##
##     content
```

```
## The following objects are masked from 'package:quanteda':
##
##     meta, meta<-
```

```
##
## Attaching package: 'tm'
```

```
## The following object is masked from 'package:quanteda':
##
##     stopwords
```

```
library(wordcloud)
```

```
## Loading required package: RColorBrewer
```

```
library(RColorBrewer)
 library(paletteer)
library(wordcloud2)
library(dplyr)

#Copied over the whole doc from 5b-12
```

Please note: this code is almost entirely from my project 911 5b-12 TAKE 4. New code will be cited as necessary.

Citizen Connect Word Cloud

Created csv from Google Analytics keywords

```
## Rows: 43 Columns: 2
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): Phrase
## dbl (1): n
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

No cleaning needed - it's already done

```
#color = paletteer_d("ggsci::default_gsea")
#paletteer_d("ggsci::default_igv")

library(colorspace)
library(viridis)
```

```
## Loading required package: viridisLite
```

```
library(RColorBrewer)
set.seed(131)
wordcloud2(data=keywords, color = paletteer_d("ggsci::default_igv"), size = 3)
```

And now to export it as a png!

Note: taking this out of code, my save isn't working

{r} # install webshot library(webshot) #webshot::install_phantomjs()

Make the graph

set.seed(131) my_graph <- wordcloud2(data=keywords, color = paletteer_d("ggsci::default_igv"), size = 3)

my_graph

save it in html

library("htmlwidgets") saveWidget(my_graph,"tmp.html",selfcontained = F)

and in png or pdf

webshot("tmp.html","wordcloud1.png", delay =5, vwidth = 1000, vheight=800)

‘‘‘