Applied Statistical Programming - Text

3/9/2022

Write R code to answer the following questions. Write the code, and then show what the computer returns when that code is run. Thoroughly comment your solutions.

You have until the beginning of class 3/21 at 10:00am to complete the assignment below. You may use R, but not any online R documentation. Submit the Rmarkdown and the knitted PDF to Canvas. Have one group member submit the activity with all group members listed at the top.

```
library(tidyverse)
```

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

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.8
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
# In-Class drill
tweets <- read_csv("Tweets.csv")
```

```
## New names:
## * `` -> ...1
```

```
## Rows: 604818 Columns: 17
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (4): ScreenName, Text, ReplyToSN, StatusSource
## dbl  (12): ...1, TweetID, Favorited, FavoritesCount, IsRetweet, RetweetCount...
## dttm  (1): CreatedTime
##
## 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.
```

```
## Filter down to only her tweets
tweetsKrewson <- tweets %>%
   filter(ScreenName == "lydakrewson")

## Find mean number of words in tweets
TextSplit <- str_split(tweetsKrewson$Text, pattern = " ")

mean(sapply(TextSplit, length))
```

```
## [1] 16.58602
```

```
## Get unique words
rawText <- tweetsKrewson$Text %>%
   str_split(pattern = " ") %>%
```

```
   unlist()

# Remove web links
rawText <- rawText[!str_detect(rawText, "https")]

## Reduce all words to their first 5 characters and compare to previous two
## things
TextShorten <- lapply(TextSplit, str_sub, start = 1, end = 5)
TextShorten %>%
   unlist() %>%
   unique() %>%
   length()
```

```
## [1] 9831
```

```
mean(sapply(TextShorten, length))
```

```
## [1] 16.58602
```

`regex` practice

Write a `regex` expression that would be able to identify phone numbers with the following patterns.

- 1234567890
- 123 456 7890
- 123-456-7890
- +1 1234567890

```
phone_numbers <- c("1234567890", "123 456 7890", "123-456-7890", "+1 1234567890")

str_detect(phone_numbers, "\\+?\\d\\s?-?\\d\\s?-?\\d")
```

```
## [1] TRUE TRUE TRUE TRUE
```

Text-as-Data

For this exercise, import the mayors data that we've been using.

- https://raw.githubusercontent.com/jmontgomery/jmontgomery.github.io/master/PDS/Datasets/Mayors.csv

Using string and/or `regex` commands, find how many mayors tweeted about (1) police and (2) Black Lives Matter.

You will need to define a list of synonymous terms related to law enforcement as well as Black Lives Matter. Your code comments should describe why you think the root you chose is sufficient to identify the aforementioned tweets. For example, "police", "policing", and "polices" are all relevant terms, but you would only subset based on the root "polic" and include a condition to exclude "policy" and "policies".

Once you have found the counts, plot the frequency distribution for each of (1) and (2). Be sure to properly label your figures.

```
## Regex explanation:
## - polic(?!.*y|.*ies): Looks for things that start with polic- but does not
##   count policy or policies
## - \\bcop(s?)(?!a|ied|iar): Looks for cop or cops, but excludes other words
##   that start with cop-
```

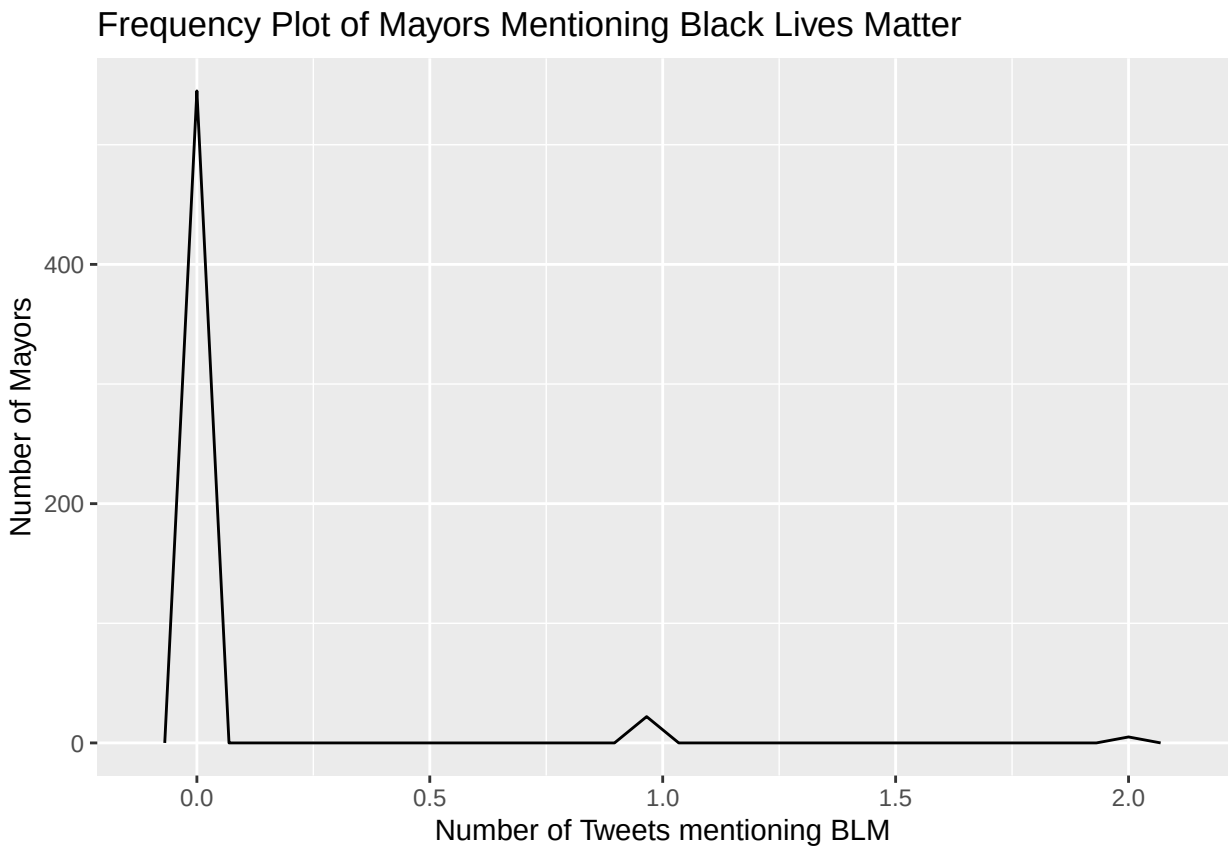
```r
tweets$police_tweet <- str_detect(tweets$Text, pattern = "polic(?!.*y|.*ies)|\\bcop(s?)(?!a|ied|iar)")

## Regex explanation:
## - Black Lives Matter|BLM: Captures either "Black Lives Matter" or "BLM"
tweets$blm_tweet <- str_detect(tweets$Text, pattern = "Black Lives Matter|BLM")

## Count tweets by mayor
mayor_tweets <- tweets %>%
  group_by(ScreenName) %>%
  mutate(
    blm_count = sum(blm_tweet),
    police_count = sum(police_tweet)
  ) %>%
  select(c("ScreenName", "blm_count", "police_count")) %>%
  unique()

ggplot(mayor_tweets, aes(x = blm_count)) +
  geom_freqpoly() +
  labs(
    title = "Frequency Plot of Mayors Mentioning Black Lives Matter",
    x = "Number of Tweets mentioning BLM",
    y = "Number of Mayors"
  )
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
ggplot(mayor_tweets, aes(x = police_count)) +
  geom_freqpoly() +
  labs(
    title = "Frequency Plot of Mayors Mentioning Policing",
    x = "Number of Tweets mentioning Policing",
    y = "Number of Mayors"
  )
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```