Applied Statistical Programming - Relational Databases

Peter Bachman, Patrick Edwards, Amaan Charaniya, Alex Avery.

3/7/2022

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 3/9 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.

Relational Databases

Data rarely come in nicely combined CSV files. This exercise gives you practice combining data sources. You are given three sets of Twitter data that need to be combined to answer a set of questions below. The data can be found at the following URLs.

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

Once you have imported the data, use relational database commands join data as necessary in order to answer the following questions.

1. For each mayor, calculate the number of times they were mentioned
2. Add to the mentions datset the number of times each mayor tweeted.
3. Create a combined dataset of all tweets from the tweets and mentions data. Subset down to overlapping columns (and rename where needed) to make this easy.
4. Are there any tweets in the mentions dataset from mayors?

Libraries; Load Data:

```
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()
```

```
library(dplyr)

# Load datasets
mayors <- read_csv(file = "https://raw.githubusercontent.com/jmontgomery/jmontgomery.github.io/master/P
```

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

## Rows: 1473 Columns: 51
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (15): FullName, LastName, FirstName, LastElectionDate, Title, CityName, ...
## dbl (36): ...1, MayorID, GenderMale, GenderFemale, RaceWhite, RaceBlack, Rac...
##
## 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.
```

```
mentions <- read_csv(file = "https://raw.githubusercontent.com/jmontgomery/jmontgomery.github.io/master/
```

```
## New names:
## * `` -> ...1
## Rows: 61570 Columns: 18-- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (5): ScreenName, Text, MayorHandle, 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.
```

```
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.
```

Preliminary Data Analysis:

```
# (i). Rename first column of 'tweets' dataset to 'tweetsdata_id':
tweets <- rename(tweets, tweetsdata_id = ...1)
# (ii). Rename first column of 'mentions' dataset to 'mentionsdata_id'
mentions <- rename(mentions, mentionsdata_id = ...1)

# (iii). Ensure that each dataset has a matching identifier (or 'key'):
colnames(mayors)  #TwitterHandle
```

```
##  [1] "...1"                "MayorID"             "FullName"
##  [4] "LastName"            "FirstName"           "GenderMale"
##  [7] "GenderFemale"        "RaceWhite"           "RaceBlack"
## [10] "RaceHispanic"        "RaceOther"           "PartyRepublican"
## [13] "PartyDemocrat"       "PartyNonPartisan"    "PartyOther"
## [16] "Ideology"            "IdeologySD"          "LastElectionDate"
## [19] "PercentVote"         "YearsCurrentPosition" "CityManager"
## [22] "CouncilManager"      "MayorCouncil"        "Title"
## [25] "CityID"              "CityName"            "CityNameFull"
## [28] "CensusID"            "CensusID2"           "State"
## [31] "StateAB"             "Region"              "Division"
## [34] "StateFIPS"           "Population"          "Latitude"
## [37] "Longitude"           "CityAge"             "CityMale"
## [40] "CityFemale"          "CityWhite"           "CityBlack"
## [43] "CityHispanic"        "CityOwner"           "CityRenter"
## [46] "GovWebsite"          "FacebookPageName"    "FacebookPageID"
## [49] "FacebookLink"        "TwitterHandle"       "TwitterLink"
```

```
colnames(mentions)  #MayorHandle
```

```
##  [1] "mentionsdata_id" "TweetID"         "ScreenName"      "Text"
##  [5] "MayorHandle"     "CreatedTime"     "Favorited"       "FavoritesCount"
##  [9] "IsRetweet"       "RetweetCount"    "Retweeted"       "ReplyToSN"
## [13] "ReplyToSID"      "ReplyToUID"      "Truncated"       "StatusSource"
## [17] "Longitude"       "Latitude"
```

```
mentions <- rename(mentions, TwitterHandle = MayorHandle)
colnames(tweets)  #ScreenName
```

```
##  [1] "tweetsdata_id"  "TweetID"        "ScreenName"     "Text"
##  [5] "CreatedTime"    "Favorited"      "FavoritesCount" "IsRetweet"
##  [9] "RetweetCount"   "Retweeted"      "ReplyToSN"      "ReplyToSID"
## [13] "ReplyToUID"     "Truncated"      "StatusSource"   "Longitude"
## [17] "Latitude"
```

```
tweets <- rename(tweets, TwitterHandle = ScreenName)
# FINISHED: all datasets should have the same matching identifier.

# (iv). Ensure that each dataset has a UNIQUE matching identifier:
mayors %>%
    count(TwitterHandle) %>%
    filter(n > 1)
```

```
## # A tibble: 3 x 2
##   TwitterHandle      n
##   <chr>          <int>
## 1 robertgarcialb     2
## 2 rodhiggins2017     2
## 3 <NA>             743
```

```
# Two duplicates: 'robertgarcialb' and 'rodhiggins2017' have two observations
# each.  Addressed in Drill #3.

mentions %>%
    count(TweetID) %>%
    filter(n > 1)
```

```
## # A tibble: 0 x 2
## # ... with 2 variables: TweetID <dbl>, n <int>
```

```
# No duplicates.

tweets %>%
    count(TweetID) %>%
    filter(n > 1)
```

```
## # A tibble: 2 x 2
##   TweetID     n
##     <dbl> <int>
## 1 4.13e17     2
## 2 7.78e17     2
```

```
# Two duplicates.
vec <- tweets %>%
    count(TweetID) %>%
    filter(n > 1) %>%
    pull(TweetID)
vec
```

```
## [1] 4.134505e+17 7.777528e+17
```

```
format(vec, scientific = FALSE)
```

```
## [1] "413450477196816384" "777752770510290944"
```

```
rm(vec)
# Two duplicates: '413450477196816384' for ScreenName 'PATownhall'.
# '777752770510290944' for ScreenName '\tderek_armstead'.

# Investigate tweet '413450477196816384' for ScreenName 'PATownhall'.
tweets %>%
    mutate(TweetID = format(TweetID, scientific = FALSE)) %>%
    filter(TweetID == "413450477196816384")
```

```
## # A tibble: 2 x 17
##   tweetsdata_id TweetID        TwitterHandle Text  CreatedTime         Favorited
##           <dbl> <chr>          <chr>         <chr> <dttm>                  <dbl>
## 1        472216 4134504771968~ patownhall    No L~ 2013-12-18 23:27:29         0
## 2        472217 4134504771968~ patownhall    From~ 2013-12-18 23:27:29         0
## # ... with 11 more variables: FavoritesCount <dbl>, IsRetweet <dbl>,
## #   RetweetCount <dbl>, Retweeted <dbl>, ReplyToSN <chr>, ReplyToSID <dbl>,
## #   ReplyToUID <dbl>, Truncated <dbl>, StatusSource <chr>, Longitude <dbl>,
## #   Latitude <dbl>
```

```
# Represents these two tweets:
# https://twitter.com/PATownhall/status/413450477196816386
# https://twitter.com/PATownhall/status/413450477196816385 Notice that neither
# tweet has TweetID = 413450477196816384. Correct the TweetIDs:

tweets %>%
    filter(tweetsdata_id == 472216)  # First tweet
```

```
## # A tibble: 1 x 17
##   tweetsdata_id TweetID TwitterHandle Text         CreatedTime         Favorited
##           <dbl>   <dbl> <chr>         <chr>        <dttm>                  <dbl>
## 1        472216 4.13e17 patownhall    No Limits o~ 2013-12-18 23:27:29         0
## # ... with 11 more variables: FavoritesCount <dbl>, IsRetweet <dbl>,
## #   RetweetCount <dbl>, Retweeted <dbl>, ReplyToSN <chr>, ReplyToSID <dbl>,
## #   ReplyToUID <dbl>, Truncated <dbl>, StatusSource <chr>, Longitude <dbl>,
## #   Latitude <dbl>
```

```
tweets <- tweets %>%
    mutate(TweetID = format(TweetID, scientific = FALSE)) %>%
    mutate(TweetID = replace(TweetID, tweetsdata_id == 472216, "413450477196816386")) %>%
    mutate(TweetID = format(TweetID, scientific = TRUE))

tweets %>%
    filter(TweetID == 413450477196816384)
```

```
## # A tibble: 1 x 17
##   tweetsdata_id TweetID        TwitterHandle Text  CreatedTime         Favorited
##           <dbl> <chr>          <chr>         <chr> <dttm>                  <dbl>
## 1        472217 4134504771968~ patownhall    From~ 2013-12-18 23:27:29         0
## # ... with 11 more variables: FavoritesCount <dbl>, IsRetweet <dbl>,
## #   RetweetCount <dbl>, Retweeted <dbl>, ReplyToSN <chr>, ReplyToSID <dbl>,
## #   ReplyToUID <dbl>, Truncated <dbl>, StatusSource <chr>, Longitude <dbl>,
## #   Latitude <dbl>
```

```
# Second tweet's ID: 472217.
tweets %>%
    filter(tweetsdata_id == 472217)  # Second tweet
```

```
## # A tibble: 1 x 17
##   tweetsdata_id TweetID        TwitterHandle Text  CreatedTime         Favorited
##           <dbl> <chr>          <chr>         <chr> <dttm>                  <dbl>
## 1        472217 4134504771968~ patownhall    From~ 2013-12-18 23:27:29         0
```

```
## # ... with 11 more variables: FavoritesCount <dbl>, IsRetweet <dbl>,
## #   RetweetCount <dbl>, Retweeted <dbl>, ReplyToSN <chr>, ReplyToSID <dbl>,
## #   ReplyToUID <dbl>, Truncated <dbl>, StatusSource <chr>, Longitude <dbl>,
## #   Latitude <dbl>
```

```
tweets <- tweets %>%
    mutate(TweetID = format(TweetID, scientific = FALSE)) %>%
    mutate(TweetID = replace(TweetID, tweetsdata_id == 472216, "413450477196816385")) %>%
    mutate(TweetID = format(TweetID, scientific = TRUE))

tweets %>%
    count(TweetID) %>%
    filter(n > 1)
```

```
## # A tibble: 1 x 2
##   TweetID                n
##   <chr>              <int>
## 1 777752770510290944     2
```

```
# Check last duplicate: '777752770510290944' for ScreenName 'derek_armstead'

tweets %>%
    mutate(count = n()) %>%
    filter(TweetID == 777752770510290944)
```

```
## # A tibble: 2 x 18
##   tweetsdata_id TweetID        TwitterHandle Text  CreatedTime         Favorited
##           <dbl> <chr>          <chr>         <chr> <dttm>                  <dbl>
## 1         83587 7777527705102~ derek_armste~ I po~ 2016-09-19 06:14:45         0
## 2         83588 7777527705102~ derek_armste~ I po~ 2016-09-19 06:14:45         0
## # ... with 12 more variables: FavoritesCount <dbl>, IsRetweet <dbl>,
## #   RetweetCount <dbl>, Retweeted <dbl>, ReplyToSN <chr>, ReplyToSID <dbl>,
## #   ReplyToUID <dbl>, Truncated <dbl>, StatusSource <chr>, Longitude <dbl>,
## #   Latitude <dbl>, count <int>
```

```
# These tweets do not exist on Twitter. I'm going to remove them.
dim(tweets)
```

```
## [1] 604818     17
```

```
tweets <- tweets %>%
    filter(TweetID != 777752770510290944)
dim(tweets)
```

```
## [1] 604816     17
```

```
tweets %>%
    count(TweetID) %>%
    filter(n > 1)
```

```
## # A tibble: 0 x 2
## # ... with 2 variables: TweetID <chr>, n <int>
```

```
# No remaining duplicates in tweets dataset.
```

 # In-class drill 1 & 2:

```
# Left join 3 variables (FacebookLink, TwitterLink, GenderFemale) into `tweets`
# dataset by `TwitterHandle` unique identifier:
left <- tweets %>%
    left_join(select(mayors, TwitterHandle, FacebookLink, TwitterLink, GenderFemale),
        by = "TwitterHandle")

# Right join:
right <- tweets %>%
    right_join(select(mayors, TwitterHandle, FacebookLink, TwitterLink, GenderFemale),
        by = "TwitterHandle")

# Inner join (matches only those observations for which the linking variable
# appears on both tables):
inner <- tweets %>%
    inner_join(select(mayors, TwitterHandle, FacebookLink, TwitterLink, GenderFemale),
        by = "TwitterHandle")

# Full join (keeps all data):
full <- tweets %>%
    full_join(select(mayors, TwitterHandle, FacebookLink, TwitterLink, GenderFemale),
        by = "TwitterHandle")
```

Drill 3:

```
# (i). dataframes to only the duplicate mayors:
subsetMayors <- mayors %>%
    filter(TwitterHandle %in% c("robertgarcialb", "rodhiggins2017"))

subsetTweets <- tweets %>%
    filter(TwitterHandle %in% c("robertgarcialb", "rodhiggins2017"))


# (ii). Repeat each join from Drill #2. Explain what happens due to the
# duplication:

# Left join:
leftsubset <- subsetTweets %>%
    left_join(subsetMayors, by = "TwitterHandle")
# EXPLANATION: there are two observations for each tweet. In other words, each
# of these mayor's tweets are duplicated.

# Right join:
rightsubset <- subsetTweets %>%
    right_join(subsetMayors, by = "TwitterHandle")
# EXPLANATION: same as for the left_join of each subset.

# Inner join:
innersubset <- subsetTweets %>%
    inner_join(subsetMayors, by = "TwitterHandle")
```

```
innersubset <- arrange(innersubset, TweetID)
# EXPLANATION: same as for left_join and right_join above.

# Full join:
fullsubset <- subsetTweets %>%
    full_join(subsetMayors, by = "TwitterHandle")
# EXPLANATION: same as for the three joins above.

# OVERALL EXPLANATION: all joins result in duplicate copies of each tweet by
# these mayors. This is because we lack a unique matching 'key' for these
# mayors for which we can match across datasets.
```

Housingkeeping: remove irrelevant datasets.

```
ls()
```

```
##  [1] "full"         "fullsubset"   "inner"        "innersubset"  "left"
##  [6] "leftsubset"   "mayors"       "mentions"     "right"        "rightsubset"
## [11] "subsetMayors" "subsetTweets" "tweets"
```

```
rm(full, inner, left, rightsubset, subsetTweets, fullsubset, innersubset, leftsubset, right, subsetMayo
```

In-Class Assignment:

```
# Item 1
# ------------------------------------------------------------------------
# Create count for times mayors were mentioned
mentions_count <- mentions %>%
    group_by(TwitterHandle) %>%
    mutate(TimesMentioned = n()) %>%
    select(TwitterHandle, TimesMentioned) %>%
    unique()

## Add mentions_count to dataset of mayors by TwitterHandle
mayors_mentions_count <- mayors %>%
    left_join(select(mentions_count, TwitterHandle, TimesMentioned), by = "TwitterHandle")

# Item 2
# ------------------------------------------------------------------------

# Count the number of times each mayor tweeted.
mayors_tweets <- tweets %>%
    group_by(TwitterHandle) %>%
    mutate(times_tweeted = n()) %>%
    select(TwitterHandle, times_tweeted) %>%
    unique()

# Add 'mayors_tweets' to 'mentions' dataset.

mentions <- mentions %>%
    left_join(mayors_tweets, by = "TwitterHandle")
```

```
## NOTE: this is a one-to-many join. The dataset 'mayors_tweets' -- with one
## observation per mayor -- is joined to the dataset 'mentions' -- with
## multiple observations per mayor.

# Item 3
# ------------------------------------------------------------------------ STEP
# (i). Fixed class of 'TweetID' in datasets.  I convert the 'TweetID'column in
# BOTH datasets to character for ease of joining.
class(tweets$TweetID)  # 'character' class.
```

```
## [1] "character"
```

```
class(mentions$TweetID)  # 'numeric' instead of 'character' class.
```

```
## [1] "numeric"
```

```
mentions <- mentions %>%
    mutate(TweetID = format(TweetID, scientific = FALSE))  # Convert to 'character' class and remove sc
class(tweets$TweetID)
```

```
## [1] "character"
```

```
class(mentions$TweetID)  # Both columns have matching classes now.
```

```
## [1] "character"
```

```
# STEP (ii). Determine overlapping & unique columns:
colnames(tweets)
```

```
##  [1] "tweetsdata_id"  "TweetID"        "TwitterHandle"  "Text"
##  [5] "CreatedTime"    "Favorited"      "FavoritesCount" "IsRetweet"
##  [9] "RetweetCount"   "Retweeted"      "ReplyToSN"      "ReplyToSID"
## [13] "ReplyToUID"     "Truncated"      "StatusSource"   "Longitude"
## [17] "Latitude"
```

```
colnames(mentions)
```

```
##  [1] "mentionsdata_id" "TweetID"         "ScreenName"      "Text"
##  [5] "TwitterHandle"   "CreatedTime"     "Favorited"       "FavoritesCount"
##  [9] "IsRetweet"       "RetweetCount"    "Retweeted"       "ReplyToSN"
## [13] "ReplyToSID"      "ReplyToUID"      "Truncated"       "StatusSource"
## [17] "Longitude"       "Latitude"        "times_tweeted"
```

```
# Overlapping columns: TweetID, TwitterHandle, Text, CreatedTime, Favorited,
# FavoritesCount, IsRetweet, RetweetCount, Retweeted, ReplyToSN, ReplyToSID,
# ReplyToUID, Truncated, StatusSource, Longitude, Latitude.

# Columns unique to 'tweets' dataset: tweetsdata_id
```

```
# Columns unique to 'mentions' dataset: ScreenName, mentionsdata_id


# STEP (iii). Create dummy variables.
tweets <- tweets %>%
    mutate(tweetsdata_origin = 1)
mentions <- mentions %>%
    mutate(mentionsdata_origin = 1)
## NOTE: these dummy variables will be useful in part #4 of this problem set -
## Patrick.  Tweets present in both datasets will have both columns
## 'tweetsdata_origin == 1' & 'mentionsdata_origin = 1' - Patrick.


# STEP (iv). Remove irrelevant variables

## Columns to remove: - 'ScreenName' from 'mentions' dataset.  -
## 'times_tweeted' from 'mentions' dataset.  - Column 1 from each one? - Peter.
## Peter is referring to the column I renamed 'mentionsdata_id' and
## 'tweetsdata_id' in the 'mentions' and 'tweets' dataset.  I think we should
## remove these and replace them with a dummy variable signifying each dataset
## these tweets came from. - Patrick.

## Remove 'tweetsdata_id' column from 'tweets' dataset:
tweets_subset <- tweets %>%
    select(-tweetsdata_id)

## Remove 'mentionsdata_id', 'ScreenName', and 'times_tweeted' columns from
## 'mentions' dataset:
colnames(mentions)
```

```
##  [1] "mentionsdata_id"     "TweetID"             "ScreenName"
##  [4] "Text"                "TwitterHandle"       "CreatedTime"
##  [7] "Favorited"           "FavoritesCount"      "IsRetweet"
## [10] "RetweetCount"        "Retweeted"           "ReplyToSN"
## [13] "ReplyToSID"          "ReplyToUID"          "Truncated"
## [16] "StatusSource"        "Longitude"           "Latitude"
## [19] "times_tweeted"       "mentionsdata_origin"
```

```
mentions_subset <- mentions %>%
    select(-mentionsdata_id) %>%
    select(-ScreenName) %>%
    select(-times_tweeted)



# STEP (v). GAMEPLAN: we should merge by 'TweetID' column.  Recall that, in in
# step (iv) of the **Preliminary Analysis** section above, we already checked
# for the uniqueness of the 'TweetID' variable in each dataset.


# STEP (vi). Merge the sub-datasets 'tweets_subset' and 'mentions_subset'
# together:
```

```r
# Make sure both datasets are arranged/grouped by 'TweetID':
tweets_subset <- tweets_subset %>%
    arrange(TweetID)

mentions_subset <- mentions_subset %>%
    arrange(TweetID)

## DON'T RUN. It's currently not working - Peter.  tweets_mentions <-
## tweets_subset %>% left_join(mentions_subset, by = 'TwitterHandle')

# What if we join by 'TweetID' instead of 'TwitterHandle'? - Patrick.  I
# believe some tweets are included in both datasets because mayors often
# mention themselves in their tweets. - Patrick.  Joining by 'TweetID',
# therefore, should allow identical observations to merge. - Patrick.  Include
# all overlapping columns common to both datasets mentioned in STEP (ii) -
# Patrick.

# NOTE: we should do a 'full_join'. This many-to-many joining will keep
# observations from all datasets while also merging matching tweets.

tweets_mentions <- full_join(tweets_subset, mentions_subset, by = c("TweetID", "TwitterHandle",
    "Text", "CreatedTime", "Favorited", "FavoritesCount", "IsRetweet", "RetweetCount",
    "Retweeted", "ReplyToSN", "ReplyToSID", "ReplyToUID", "Truncated", "StatusSource",
    "Longitude", "Latitude"))


# Item
# 4---------------------------------------------------------------------- Is
# this an inner join? - Peter.  inner_tweets <- tweets_subset %>%
# inner_join(mentions_subset)

## nrow(inner_tweets)

# This is where the dummy variables I created in part #3, STEP (iii) come in
# handy - Patrick.

# STEP (i): Replace NAs in each dummy variable with zero.
tweets_mentions <- tweets_mentions %>%
    mutate(mentionsdata_origin = replace(mentionsdata_origin, is.na(mentionsdata_origin),
        0)) %>%
    mutate(tweetsdata_origin = replace(tweetsdata_origin, is.na(tweetsdata_origin),
        0))

# STEP (ii): Create dummy variable for observations where 'tweetsdata_origin ==
# 1' & 'mentionsdata_origin = 1':
tweets_mentions <- tweets_mentions %>%
    mutate(matched_tweets = NA) %>%
    mutate(matched_tweets = replace(matched_tweets, tweetsdata_origin == 1 & mentionsdata_origin ==
        1, 1)) %>%
    mutate(matched_tweets = replace(matched_tweets, tweetsdata_origin == 0 | mentionsdata_origin ==
        0, 0))

# STEP (iii): Check that (a) the 'matched_tweets' column takes on a value for
```

```r
# EVERY observation. tweets_mentions <- ungroup(tweets_mentions, TweetID)

tweets_mentions %>%
    summarise(matched_tweets, tot_NAs = sum(is.na(matched_tweets)))
```

```
## # A tibble: 660,562 x 2
##    matched_tweets tot_NAs
##             <dbl>   <int>
##  1              0       0
##  2              0       0
##  3              0       0
##  4              0       0
##  5              0       0
##  6              0       0
##  7              0       0
##  8              0       0
##  9              0       0
## 10              0       0
## # ... with 660,552 more rows
```

```r
# No NAs.

total_tweets_num <- tweets_mentions %>%
    summarize(total_tweets = dim(tweets_mentions)[[1]])
total_tweets_num
```

```
## # A tibble: 1 x 1
##   total_tweets
##          <int>
## 1       660562
```

```r
# 660,562 total tweets in the combined dataset.

matched_tweets_num <- tweets_mentions %>%
    summarize(total_matched_tweets = sum(matched_tweets))
matched_tweets_num
```

```
## # A tibble: 1 x 1
##   total_matched_tweets
##                  <dbl>
## 1                 5824
```

```r
# 5,824 tweets that appears in both datasets (i.e., where 'matched_tweets == 1)
# out of 660,562 total tweets.

unmatched_tweets_num <- tweets_mentions %>%
    summarize(total_unmatched_tweets = sum(matched_tweets == 0))
unmatched_tweets_num
```

```
## # A tibble: 1 x 1
##   total_unmatched_tweets
##                    <int>
## 1                 654738
```

```r
# 654,738 tweets that appear in one of the 'tweets' or 'mentions' datasets but
# not both.

matched_tweets_num[1, 1] + unmatched_tweets_num[1, 1] - total_tweets_num[1, 1]
```

```
##   total_matched_tweets
## 1                    0
```

```r
# Therefore, no tweets fall outside of the three categories.

# ANSWER: there are 5,824 tweets in the 'mentions' dataset from mayors.
```