Practice5S24

BilalTariq

2024-03-07

Practice5 - Due Thursday, 3/7 by midnight to Gradescope

Reminder: Practice assignments may be completed working with other individuals.

Reading

The associated reading for the week is Chapter 19 and Section 12.1.

Practicing Academic Integrity

If you worked with others or used resources outside of provided course material (anything besides our textbook, course materials in the repo, labs, R help menu) to complete this assignment, please acknowledge them below using a bulleted list.

I acknowledge the following individuals with whom I worked on this assignment:

Name(s) and corresponding problem(s)

- N/A

I used the following sources to help complete this assignment:

Source(s) and corresponding problem(s)

- StackOverflow

1 - MDSR 12.6 (modified)

"Baseball players are voted into the Hall of Fame by the members of the Baseball Writers of America Association. Quantitative criteria are used by the voters, but they are also allowed wide discretion. The following code identifies the position players (not pitchers) who have been elected to the Hall of Fame and tabulates a few basic statistics, include their number of career hits (`tH`), home runs (`tHR`), runs batted in (`tRBI`), and stolen bases (`tSB`)." Only players with more than 1000 total hits are included as a way to obtain the position players only (not pitchers).

```
hof <- Batting %>%
  group_by(playerID) %>%
  inner_join(HallOfFame, by = "playerID") %>%
  filter(inducted == "Y" & votedBy == "BBWAA") %>%
  summarize(tH = sum(H), tHR = sum(HR), tRBI = sum(RBI), tSB = sum(SB)) %>%
  filter(tH > 1000)
```

```
Warning in inner_join(., HallOfFame, by = "playerID"): Detected an unexpected many-to-many re
i Row 5 of `x` matches multiple rows in `y`.
i Row 72 of `y` matches multiple rows in `x`.
i If a many-to-many relationship is expected, set `relationship =
  "many-to-many"` to silence this warning.
```

- Use the `kmeans()` function to perform a cluster analysis on these players.
- Explain your choice of k, the number of clusters.
- Describe the properties that seem common to each cluster in your solution.
- Include at least one visual that helps explore the clusters found.
- Your solution should include some discussion of whether or not you chose to scale the variables and why. (You should determine whether or not you need to scale before clustering.)
- Remember that your solution must be reproducible. (Hint: this means you need to do something in your code.)

Solution:

```
set.seed(100)
clustering_prep_hof <- hof %>%
  select("tH","tHR") %>%
  drop_na()

glimpse(clustering_prep_hof)
```

```
Rows: 86
Columns: 2
$ tH  <int> 3771, 2724, 2677, 2314, 2583, 2048, 2150, 3060, 3010, 1779, 3154, ~
$ tHR <int> 755, 210, 83, 449, 512, 389, 358, 291, 118, 68, 317, 149, 242, 92,~
```

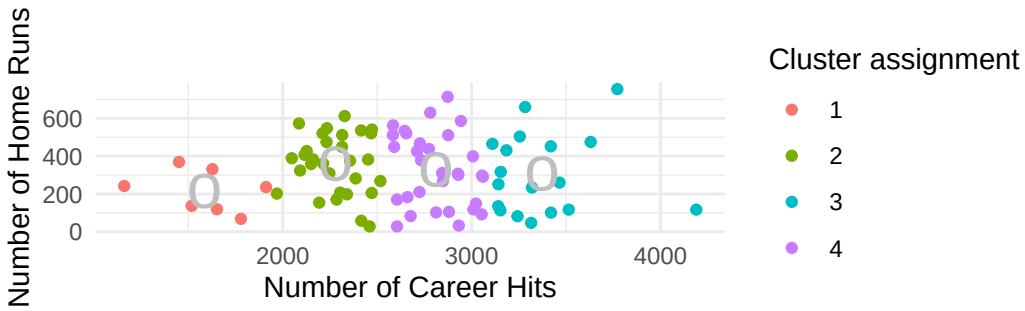
```r
set.seed(231) #Reproducibility

clustering_hof <- clustering_prep_hof %>%
  select("tH", "tHR") %>%
  kmeans(centers = 4, nstart = 20)

clustering_hof$cluster
```

```
 [1] 3 4 4 2 4 2 2 4 4 1 3 4 1 4 2 3 1 3 2 4 2 2 2 4 4 1 4 4 3 1 4 4 4 4 3 4 4 4
[39] 2 1 3 2 2 4 2 2 3 2 2 3 2 3 3 2 4 4 2 2 4 2 3 4 4 1 4 4 2 2 4 4 2 2 3 2 2 2
[77] 2 2 3 2 3 4 4 3 3 3
```

```r
clustering_hof$centers
```

```
        tH      tHR
1 1585.857 214.5714
2 2279.267 359.2000
3 3372.556 306.4444
4 2810.484 328.3226
```

```r
hof <- hof %>%
  mutate(clusters2 = factor(clustering_hof$cluster))

ggplot(data = hof, aes(x = tH, y = tHR)) +
geom_point(aes(color = clusters2)) +
coord_fixed() +
geom_point(data = data.frame(clustering_hof$centers),
aes(x = tH, y = tHR),
pch = "o", size = 8, color = "gray") +
labs(x = "Number of Career Hits",
y = "Number of Home Runs",
color = "Cluster assignment") + theme_minimal()
```



In the above, what I did was perform a kmeans clustering using 4 centres. The reason why I chose 4 centres was because a lot of data points seem to be concentrated around 4 ranges in the number of career hits: less than 2000 hits, between 2000-2500 hits, between 2500-3000 hits and greater than 3000 hits. This made clustering based on these 4 points optimal. We can see that there are roughly the same number of data points for each region except for $<$2000 hits due to it having a greater spread. We also see that the clustering seems to be on the basis of career hits rather than the number of home runs, due to them having lesser variability. The reason why I didn't scale this was because I only considered two variables in this situation which were not vastly different from each other so I deemed it uneccessary.

2 - Trump Tweets

David Robinson, Chief Data Scientist at DataCamp, wrote a blog post "Text analysis of Trump's tweets confirms he writes only the (angrier) Android half". He provides a dataset with over 1,500 tweets from the account realDonaldTrump between 12/14/2015 and 8/8/2016. We'll use this dataset to explore the tweeting behavior of @realDonaldTrump during this time period.

First, read in the file. Note that there is a `TwitteR` package which provides an interface to the Twitter web API. We'll use this R dataset David Robinson created using that package so that you don't have to set up Twitter authentication.

```
# the .rda file is also provided if this website ever breaks
load(url("http://varianceexplained.org/files/trump_tweets_df.rda"))
```

> part a - Wrangling! There are a number of variables in the dataset we won't need. First, confirm that all the observations in the dataset are from the screen-name *realDonaldTrump*. Then, create a new dataset called `tweets` that only includes the variables `text`, `created` and `statusSource`.

Solution:

```
trump_tweets_df2 <- trump_tweets_df %>%
  filter(screenName == "realDonaldTrump")

tweets <- trump_tweets_df2 %>%
  select(text,created,statusSource)
```

> part b - Using the `statusSource` variable, compute the number of tweets from each source. How many different sources are there? How often are each used?

Hint: You could answer the questions with a nice table printed to the screen.

Solution:

```
sourcesTweets <- tweets %>%
  group_by(statusSource) %>%
  summarise(Number_Tweets = n())

kable(sourcesTweets, booktabs = TRUE)
```

statusSource	Number_Tweets
Instagram	1
Twitter Web Client	120
Twitter for iPad	1
Twitter for Android	762
Twitter for iPhone	628

> part c - We're going to compare the language used between the Android and iPhone sources, so we only want to keep tweets coming from those sources. Explain what the `extract()` function (from the **tidyverse** package) is doing below. Include in your own words what each argument is doing.

head(tweets)

```
tweets <- tweets %>%
  extract(col = statusSource, into = "source",
          regex = "Twitter for (.*)<",
          remove = FALSE) %>%
  filter(source %in% c("Android", "iPhone"))
```

Solution: It seems as though we're extracting entries from one column and making them into another column called "source". Additionally, the regex expression tells us which parts to extract from after the "Twitter for" expression. The remove expression tells us if we want to remove the input column from the data frame.

> part d - How does the language of the tweets differ by source? Create a word cloud for the top 50 words used in tweets sent from the Android. Create a second word cloud for the top 50 words used in tweets sent from the iPhone. How do these word clouds compare? (Are there some common words frequently used from both sources? Are the most common words different between the sources?)

Note: Don't forget to remove stop words before creating the word cloud. Also remove the terms "https" and "t.co".

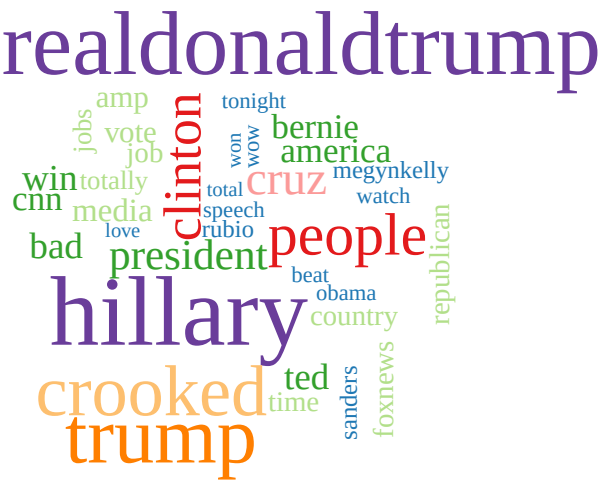
Solution:

```
data(stop_words)

tweets_iphone <- tweets %>%
  filter(source == "iPhone")

tweets_android <- tweets %>%
  filter(source == "Android")
```

```r
frequencyTweets_iphone <- tweets_iphone %>%
  unnest_tokens(output = word, input = text) %>%
  anti_join(stop_words, by = "word") %>%
  count(word, sort = TRUE)

frequencyTweets_android <- tweets_android %>%
  unnest_tokens(output = word, input = text) %>%
  anti_join(stop_words, by = "word") %>%
  count(word, sort = TRUE)

#We also don't want words like https and .co so let's filter them out



set.seed(231)
mypal <- brewer.pal(10, "Paired")

frequencyTweets_iphone %>%
  slice(1:50) %>%
  filter(word != "https") %>%
  filter(word != "t.co") %>%

with(wordcloud(words = word,
               freq = n,
               min.freq = 20,
               random.order = TRUE,
               scale = c(4,0.3),
               rot.per = 0.15,
               colors = mypal,
               family = "serif"))
```


```
set.seed(231)
mypal <- brewer.pal(10, "Paired")

frequencyTweets_android %>%
  slice(1:50) %>%
  filter(word != "https") %>%
  filter(word != "t.co") %>%
```

```r
with(wordcloud(words = word,
               freq = n,
               min.freq = 20,
               random.order = TRUE,
               scale = c(4,0.3),
               rot.per = 0.15,
               colors = mypal,
               family = "serif"))
```



part e - Consider the sentiment. Compute the proportion of words among the tweets within each source classified as "angry" and the proportion of words classified as "joy" based on the NRC lexicon. How does the proportion of "angry" and "joy" words compare between the two sources? What about "positive" and "negative" words?

Solution:

```r
nrc_lexicon <- get_sentiments("nrc")

nrc_tweetsiPhone <- frequencyTweets_iphone %>%
  inner_join(nrc_lexicon, by = "word") %>%
  filter(sentiment %in% c("anger","joy" )) %>%
  arrange(desc(n)) %>%
  group_by(sentiment) %>%
  reframe(
    Number_of_words = sum(n)
  )

kable(nrc_tweetsiPhone, booktabs = TRUE)
```

sentiment	Number_of_words
anger	170
joy	161

```r
nrc_tweetsAndroid <- frequencyTweets_android %>%
  inner_join(nrc_lexicon, by = "word") %>%
  filter(sentiment %in% c("anger","joy" )) %>%
  arrange(desc(n)) %>%
  group_by(sentiment) %>%
  reframe(
    Number_of_words = sum(n)
  )

kable(nrc_tweetsAndroid, Booktabs = TRUE)
```

sentiment	Number_of_words
anger	363
joy	265

part f - Lastly, based on your responses above, do you think there is evidence to support Robinson's claim that Trump only writes the Android half of the tweets from realDonaldTrump? In 2-4 sentences, please explain.

Solution: Yes there seems to be evidence that suggests that the angrier half of trumps tweets come from Android. As we can see from above, the ratio of angry tweets to joyous tweets on android is 363/265 which is significantly higher than 170/161 that's for the joyous tweets on iPhone.