Coding Challenge 2 (Undergraduate version)

2024-09-22

Instructions

This version of Coding Challenge 2 is specifically for students enrolled in the COMM section (undergraduate) of the class. If you are a DACSS graduate student, please use the DACSS version instead.

The topic for this coding challenge is *Trend Analysis*. Please review the lecture slides before proceeding to complete the coding challenge.

What you need to do?

Edit the Code: You will see code presented in R code chunks. Follow the prompts to edit the code as needed. *Run the Code*: After making your edits, click *Run* to check if the code produces the correct output. *Focus on Changes*: Pay special attention to the part of the code marked as xxxx; this is where you need to make your changes.

Once you have completed all required tasks, click the **Knit** button, a HTML document will be generated that includes both content as well as the output of any embedded R code chunks within the document. The .html file is saved to your working directory by default.

We will start this activity in class. You can collaborate with your group; however, each student is expected to complete their own work. Please submit the final .html file by *11:59 PM on October 4, 2024 (Friday)*, to receive participation credit.

Step 1: Load the data

know which dataset you are assigned to

Group 1: New England Patriots: https://curiositybits.cc/files/Patriots.csv (https://curiositybits.cc/files/Patriots.csv)

Group 2: Philadelphia Eagles: https://curiositybits.cc/files/Eagles.csv (https://curiositybits.cc/files/Eagles.csv)

Group 3: UMass football: https://curiositybits.cc/files/UMass-

Football.csv (https://curiositybits.cc/files/UMass-Football.csv)

Group 4: UMass hockey: https://curiositybits.cc/files/UMass-Hockey.csv (https://curiositybits.cc/files/UMass-Hockey.csv)

Revise the code below to import the CSV file nfl.csv as df1, and your group’s data as a dataframe called df2.

```
library(readr) # load the library readr
library(ggplot2)
library(ggthemes)
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(reshape2)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(stringr)
library(tidyr)
```

```
##
## Attaching package: 'tidyr'
```

```
## The following object is masked from 'package:reshape2':
##
##     smiths
```

```
df1 <- read_csv("https://curiositybits.cc/files/nfl.csv")
```

```
## Warning: One or more parsing issues, call `problems()` on your data frame for detail
s,
## e.g.:
##   dat <- vroom(...)
##   problems(dat)
```

```
## Rows: 24905 Columns: 21
```

```
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (10): Account, User Name, Post Created, Type, URL, Link, Photo, Descrip...
## dbl   (6): Followers at Posting, Likes, Comments, Views, Sponsor Id, Overper...
## num   (1): Total Interactions
## lgl   (2): Like and View Counts Disabled, Title
## date  (1): Post Created Date
## time  (1): Post Created Time
##
## 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.
```

```
df2 <- read_csv("https://curiositybits.cc/files/Eagles.csv")
```

```
## Rows: 4383 Columns: 21
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (10): Account, User Name, Post Created, Type, URL, Link, Photo, Descrip...
## dbl   (6): Followers at Posting, Likes, Comments, Views, Sponsor Id, Overper...
## num   (1): Total Interactions
## lgl   (2): Like and View Counts Disabled, Title
## date  (1): Post Created Date
## time  (1): Post Created Time
##
## 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.
```

Step 2:

Run the code below. No need to change anything. The rbind() function combine df1 and df2 into a new dataframe called df. The new dataframe contains post the NFL posts and the posts from the data your group is assigned to.

```
df <- rbind(df1, df2)
```

Step 3

Suppose the analytical goal is to compare the daily average like counts between the NFL and the team represented in your group’s data.

Revise the code below to calculate the average daily likes.

Hint: You need to group the data by date and account name (the column named ‘Account’).

```
df$created_date <- ymd_hms(df$`Post Created`)  #something needs to be changed here.
df$created_date <- with_tz(df$created_date,"America/New_York")
df$created_date <- as.Date(df$created_date)



df$date_label <- as.factor(df$created_date)
df$account_name <- as.factor(df$Account)

daily_count <- df %>%
  group_by(date_label, account_name) %>%
  summarise(avg_likes = mean(Likes))
```

```
## `summarise()` has grouped output by 'date_label'. You can override using the
## `.groups` argument.
```

```
daily_count <- daily_count %>% pivot_longer(cols = -c(date_label, account_name), names_t
o = "variable", values_to = "value")
```

Step 4

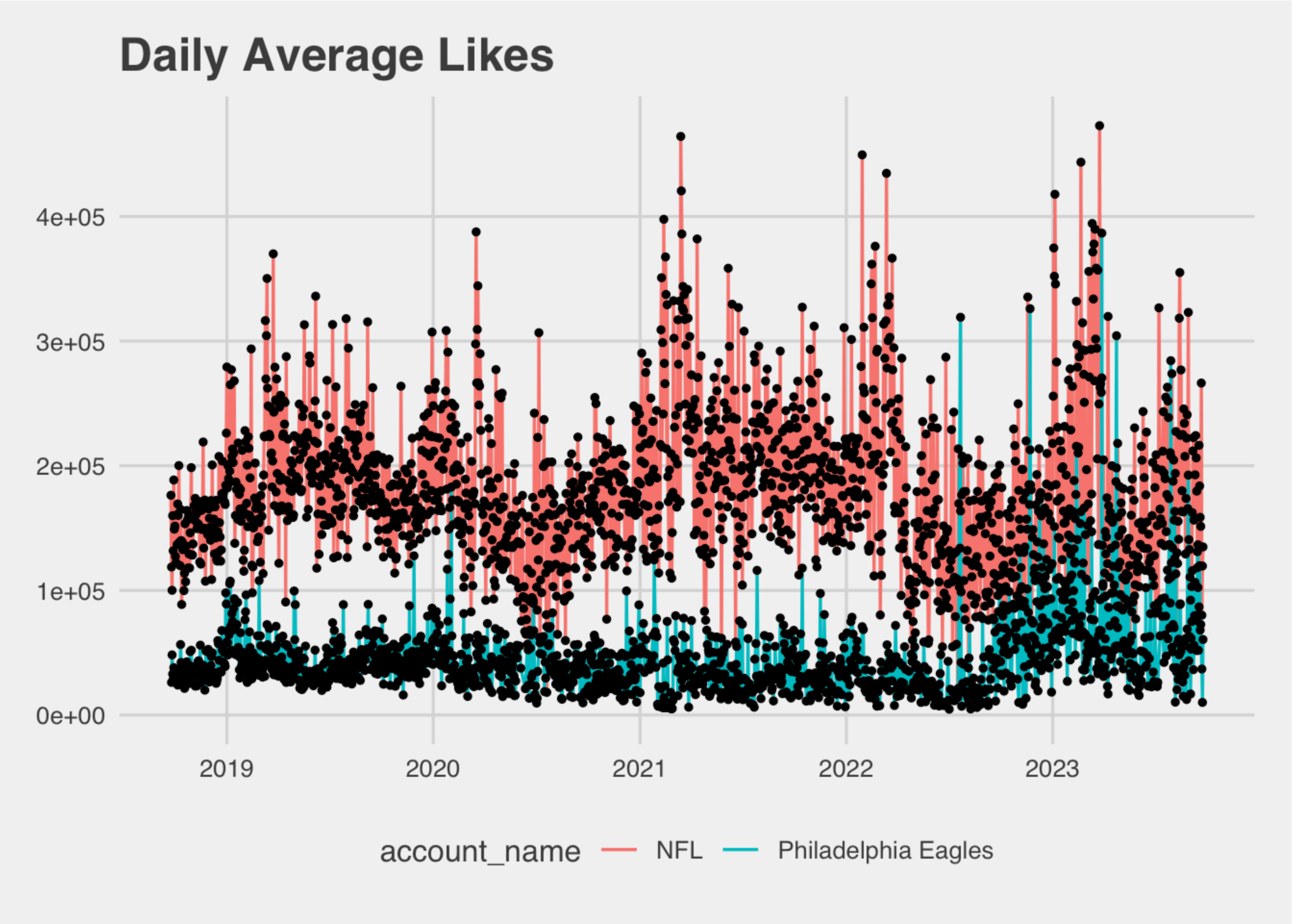
Based on the daily_count created from the previous step, revise the code below to create a ggplot to show and compare the daily average likes of the two teams.

Change the plot title to ‘Daily Average Likes’ and change the y axis title to “average likes”

```
library(ggplot2)
library(ggthemes)

daily_count$date_label <- as.Date(daily_count$date_label)

ggplot(data = daily_count[daily_count$variable== "avg_likes",], aes(x = date_label, y =
value, group = account_name)) +   #something needs to be changed here.
  geom_line(linewidth = 0.6, alpha = 1.7, aes(color = account_name)) + #something needs
to be changed here.
  geom_point(size = 1) +
  ylim(0, NA) +
  #scale_fill_manual(values=moma.colors("Smith", 3))+
  theme(legend.title=element_blank(), axis.title.x = element_blank()) +
  ylab("average likes") +
  ggtitle("Daily Average Likes") +
  theme_fivethirtyeight(base_size = 12, base_family = "sans")
```


Daily Average Likes
4e+05
3e+05
2e+05
1e+05
0e+00
2019
2020
2021
2022
2023
account_name
NFL
Philadelphia Eagles