assessment

Daniel DeWaters

10/25/2020

This is a writeup document for the assessment given by Efortels Inc. I chose to implement my solution to the coding challege with R as it is the easiest language to document and share an analysis through the use of R Markdown.

Finding the Busiest Station in December 2016 (Code for Previous Question)

Getting the data

I downloaded the data and used the `read_delim` function from the `readr` package to read it in. This package is nice for selecting which columns to read in and setting their type easily.

```
# download file
download.file("http://web.mta.info/developers/data/nyct/turnstile/turnstile_161224.txt",
              destfile = "data/turnstile_161224.txt")

# Read in file
station_data <- read_delim("data/turnstile_161224.txt", delim=",", trim_ws=TRUE,
                           # Read in only the station, entries, and exits columns,
                           # and set their type
                           col_types=cols_only("STATION"=col_character(),
                                               "ENTRIES"=col_number(),
                                               "EXITS"=col_number()))
```

Calculating the Busiest Station

After reading in the data I used the `group_by` and `summarise` functions from the `dplyr` package to calculate the sum of entries and exits for each station and sort them in descending order to get the top sums of entries and exits.

```
busiest_stations <-
  station_data %>%
  # Get the sum of each station's entries and exits
  group_by(STATION) %>%
  summarise(sum_entries=sum(ENTRIES), sum_exits=sum(EXITS), .groups="drop") %>%
  # Sort by descending order of entries, then by descending order of exits
  arrange(desc(sum_entries), desc(sum_exits))

head(busiest_stations)
```

```
## # A tibble: 6 x 3
##   STATION          sum_entries    sum_exits
##   <chr>                  <dbl>        <dbl>
## 1 42 ST-PORT AUTH 316902103019 260555918711
## 2 125 ST          291276298390 166439786900
## 3 23 ST           273635974527 254184602851
## 4 CANAL ST        253470306707 252678618530
## 5 TIMES SQ-42 ST  240432487850 189359465518
## 6 104 ST          209216934068 213102891164
```

Finding the 5 Stations with the Least Number of Entries and Exits

Web Scraping for download links

I used the `rvest` package to read in the html source and grab the links for downloading the data. I created a data frame that had one column for the data file links and a column for the corresponding date.

```r
# Read in html source
my_url = "http://web.mta.info/developers/turnstile.html"
page <- read_html(my_url)

# Get dates and corresponding download links for the data
page_text <-
  page %>%
  html_nodes("a") %>%
  html_text()

page_links <-
  page %>%
  html_nodes("a") %>%
  html_attr("href")

# combine links and dates into single data frame
data_links <- as.data.frame(cbind(page_text, page_links))
colnames(data_links) <- c("date", "links")

# Display a few rows of the data frame
data_links[38:42,]
```

```
##                             date                                    links
## 38   Saturday, October 24, 2020 data/nyct/turnstile/turnstile_201024.txt
## 39   Saturday, October 17, 2020 data/nyct/turnstile/turnstile_201017.txt
## 40   Saturday, October 10, 2020 data/nyct/turnstile/turnstile_201010.txt
## 41   Saturday, October 03, 2020 data/nyct/turnstile/turnstile_201003.txt
## 42 Saturday, September 26, 2020 data/nyct/turnstile/turnstile_200926.txt
```

The data frame created in the previous code chunk needed to be tidied a bit before I could download any of the data files. I removed the URLs that didn't lead to the data files, added a column for the file names of the files to be downloaded, and completed the URLs in the "links" column. Finally, I formatted the dates column and removed any URLs for files that weren't from 2012 to 2014.

```
data_links_clean <-
  data_links %>%
  # Remove any links that aren't for the data
  filter(grepl("data", links)) %>%
  # fix file paths
  mutate(filenames = sub("data/nyct/turnstile/", "data/", links)) %>%
  # Complete url
  mutate(links = paste0("http://web.mta.info/developers/", links)) %>%
  # Clean dates column and filter out files that aren't for 2012-2014
  mutate(date = sub("^Saturday, ", "", date)) %>%
  mutate(date = mdy(date)) %>%
  filter(year(date) >= 2012 & year(date) <= 2014)

# Grab urls and file paths to facilitate download code
download_links <- data_links_clean$links
filenames <- data_links_clean$filenames
```

Downloading and Reading in the Data

I used a loop to download each file.

```
# Download each file from 2012-2014
for(i in 1:dim(data_links_clean)[1]){
  download.file(download_links[i], destfile = filenames[i])
}
```

I read in each file into a data frame using a loop, and appended each new file to a data frame as they were read in.

```
# Initialize emtpy data frame
station_data <- data.frame(matrix(ncol=3, nrow=0))

for(i in 1:length(filenames)){
  # Read in new file
  new_df <- read_delim(filenames[i], delim=",", trim_ws=TRUE,
                       col_types=cols_only("STATION"=col_character(),
                                           "ENTRIES"=col_number(),
                                           "EXITS"=col_number()))
  # Append new file to data frame
  station_data <- rbind(station_data, new_df)
}
```

Calculate Sums of Entries and Exits

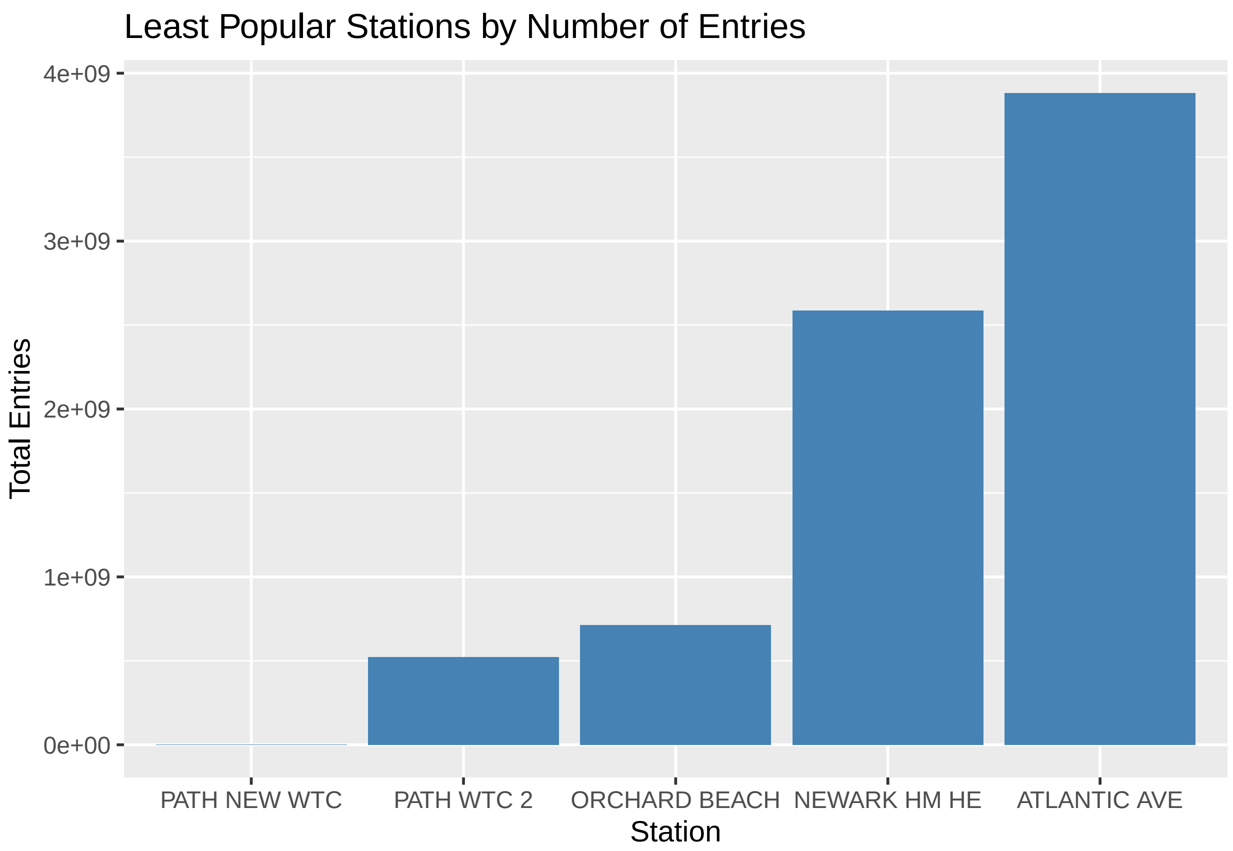
Using the same code from the previous question, I calculated the sums of entries and exits for each station, and sorted by the total number of entries and exits. I decided to find the stations with the least numbers of entries separate from the stations with the least numbers of exits because the number of exits are much higher than the number of entries, which makes it hard to see all of the bars when they're plotted onto a single graph.

```r
# Calculate sums of entries and exits for each station
station_data_totals <-
  station_data %>%
  group_by(STATION) %>%
  summarize(sum_entries=sum(ENTRIES), sum_exits=sum(EXITS), .groups="drop")

# Get the 5 stations with the least entries
least_entries <-
  station_data_totals %>%
  select(STATION, sum_entries) %>%
  top_n(sum_entries, n=-5) %>%
  arrange(sum_entries) %>%
  mutate(STATION=as.factor(STATION))

# Get the 5 stations with the least exits
least_exits <-
  station_data_totals %>%
  select(STATION, sum_exits) %>%
  top_n(sum_exits, n=-5) %>%
  arrange(sum_exits) %>%
  mutate(STATION=as.factor(STATION))
```

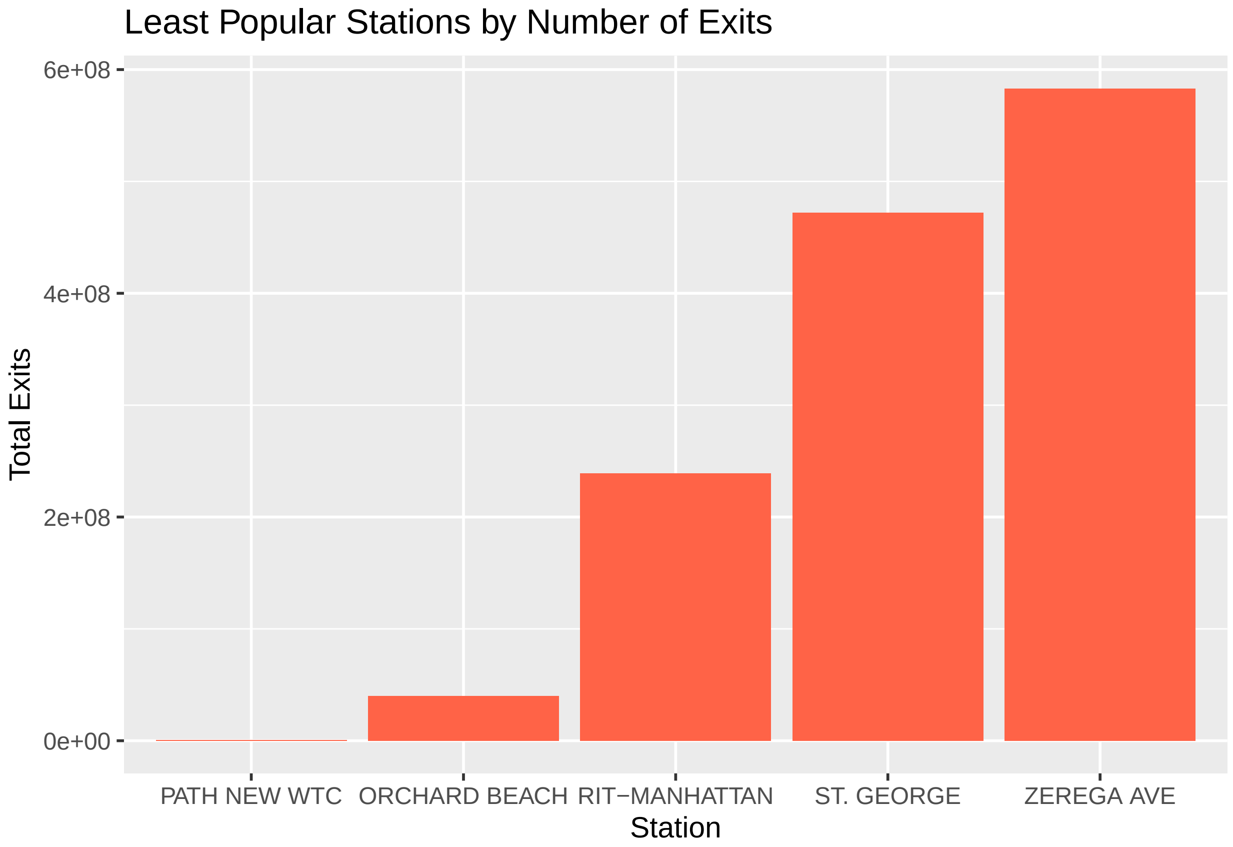
Making a Barplot

I used the ggplot2 package to create the bar plots.

```r
least_entries %>%
  ggplot(aes(x=reorder(STATION, sum_entries), y=sum_entries)) +
  geom_bar(stat="identity", fill="steelblue") +
  xlab("Station") + ylab("Total Entries") +
  ggtitle("Least Popular Stations by Number of Entries")
```



```
least_exits %>%
  ggplot(aes(x=reorder(STATION, sum_exits), y=sum_exits)) +
  geom_bar(stat="identity", fill="tomato") +
  xlab("Station") + ylab("Total Exits") +
  ggtitle("Least Popular Stations by Number of Exits")
```



Conclusions

The resulting bar graphs show that the 5 stations with the least number of entries are “Path New WTC”, “Path WTC 2”, “Orchard Beach”, “Newark HM HE”, and “Atlantic AVE”. The 5 stations with the least number of exits are “Path New WTC”, “Orchard Beach”, “Rit-Manhattan”, “St. George”, and “Zerega Ave”.

Contact Information

Thank you for reading my assessment. I had a lot of fun completing it. Please feel free to call me at 434-362-0601 or email me to set up a call at dandewaters@gmail.com. If I am not a fit for this specific opportunity, I am flexible and willing to take other positions that may better fit my qualifications. I hope to hear from you soon!