Starbucks and Dunkin' Donuts Analysis

Zane Shango

4/6/2020

Load in and Clean Data

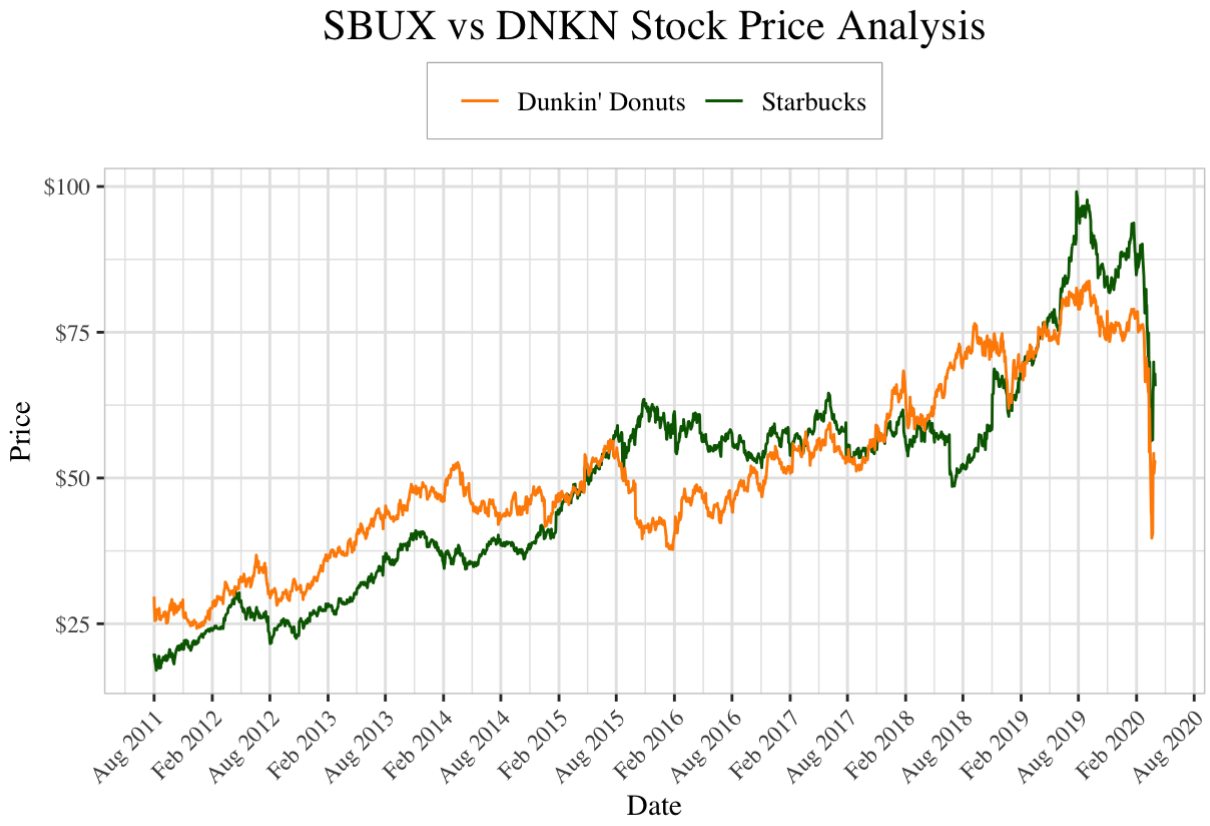
```
starbucks <- read.csv("directory.csv")


# change factor variables to characters
starbucks$City <- as.character(starbucks$City)
starbucks$State.Province <- as.character(starbucks$State.Province)
starbucks$Country <- as.character(starbucks$Country)
starbucks$Postcode <- as.character(starbucks$Postcode)
starbucks$Brand <- as.character(starbucks$Brand)

# filter starbucks to only US locations, filter out alaska and hawaii
# starbucks <- starbucks[(starbucks$Country == 'US') & (starbucks$State.Province != 'A
K') & starbucks$State.Province != 'HI',]
starbucks <- starbucks %>%
  filter(Country == 'US')


# change state abbreviations to lowercase and names to match map data
starbucks$State.Province <- tolower(abbr2state(starbucks$State.Province))

# create state abbreviation column for map
starbucks$State.Abbr <- state2abbr(starbucks$State.Province)


dunkin <- read.csv("dunkin.csv")
```

Stock Price Analysis

```
# extract stock price data over time period (August 2011- January 2020)
#sbux <- getSymbols("SBUX", src = "yahoo", from = "2011-08-01", to = "2020-04-01", auto.
assign = FALSE)
#dnkn <- getSymbols("DNKN", src = "yahoo", from = "2011-08-01", to = "2020-04-01", auto.
assign = FALSE)

# timeout issues, just load in variables used for coffee_app (includes stock price data
 scraping using the above code)
load(file = "coffee_vars.rda")
```

```
sbux <- as.data.frame(sbux)
dnkn <- as.data.frame(dnkn)

# create date column and column with only year for shiny app
sbux$date <- as.Date(format(rownames(sbux)))
sbux$year <- as.numeric(format(sbux$date, "%Y"))

dnkn$date <- as.Date(format(rownames(dnkn)))
dnkn$year <- as.numeric(format(dnkn$date, "%Y"))
```

```
# create plots showing the price changes
stocks_plot <- ggplot() +
  geom_line(sbux, mapping = aes(x = date, y = SBUX.Close, color = "Starbucks")) +
  geom_line(dnkn, mapping = aes(x = date, y = DNKN.Close, color = "Dunkin' Donuts")) +
  scale_color_manual(name = "", values = c("Starbucks" = "darkgreen", "Dunkin' Donuts" =
"darkorange")) +
  labs(
    x = "Date",
    y = "Price",
    title = "SBUX vs DNKN Stock Price Analysis") +
  theme_bw() +
  theme(
    text = element_text(family = "Times"),
    axis.text.x = element_text(angle = 45, hjust = 1),
    plot.title = element_text(hjust = 0.5, size = 16),
    plot.margin = unit(c(1,1,1,1), "cm"),
    legend.position = "top",
    legend.text = element_text(size = 10),
    legend.box.background = element_rect(color = "grey70"),
    panel.border = element_rect(linetype = 'solid', fill = NA, color = "grey80"),
    panel.grid = element_line(color = "grey90", linetype = "solid", lineend = "butt")) +
  scale_x_date(date_labels = "%b %Y", date_breaks = "6 months") +
  scale_y_continuous(labels=scales::dollar_format())


stocks_plot
```



Loading [MathJax]/extensions/MathZoom.js

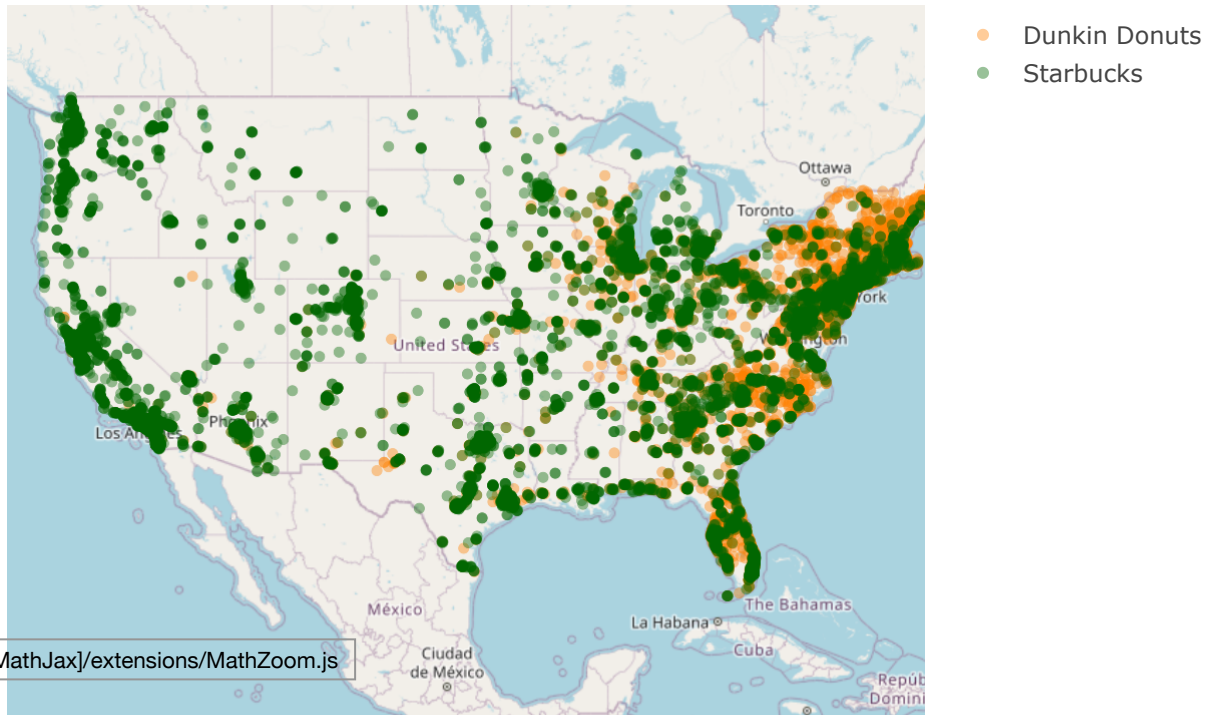
Map

```
# combine dataframes for mapping with plotly

# start by masking starbucks and dunkin dataframes, to have only the columns needed for
 mapping
# and rename the columns so they match
starbucks_map <- starbucks %>%
  select(Brand, City, State.Abbr, Country, Longitude, Latitude)

dunkin_map <- dunkin %>%
  select(name, city, state, country, lng, lat) %>%
  rename(Brand = name, City = city, State.Abbr = state, Country = country, Longitude = l
ng, Latitude = lat)


# rbind them into a new dataframe
map_data <- rbind(starbucks_map, dunkin_map)

# filter for only starbucks and dunkin donuts brands (remove Teavana, Coffee House Holdi
ngs, and Evolution Fresh)
# and add color column for mapping
# and add  city.state column for hovertext
map_data <- map_data %>%
  filter(Brand %in% c("Starbucks", "Dunkin Donuts")) %>%
  mutate(Color = ifelse(Brand == "Starbucks", "darkgreen",
                        ifelse(Brand == "Dunkin Donuts", "orange", NA))) %>%
  mutate(City.State = paste(City, State.Abbr, sep = ', '))


head(map_data)
```

```
##       Brand      City State.Abbr Country Longitude Latitude     Color
## 1 Starbucks Anchorage         AK      US   -149.78    61.21 darkgreen
## 2 Starbucks Anchorage         AK      US   -149.84    61.14 darkgreen
## 3 Starbucks Anchorage         AK      US   -149.85    61.11 darkgreen
## 4 Starbucks Anchorage         AK      US   -149.89    61.13 darkgreen
## 5 Starbucks Anchorage         AK      US   -149.86    61.14 darkgreen
## 6 Starbucks Anchorage         AK      US   -149.87    61.19 darkgreen
##      City.State
## 1 Anchorage, AK
## 2 Anchorage, AK
## 3 Anchorage, AK
## 4 Anchorage, AK
## 5 Anchorage, AK
## 6 Anchorage, AK
```

```
#plotly map with streetmap style

# list with colors orange and dark green to assign below
map_colors <- c("#FF8000", "#006600")


spatial <- map_data
spatial <- spatial %>%
  plot_ly(
    lat = ~Latitude,
    lon = ~Longitude,
    color = ~Brand, # color by Brand
    colors = map_colors, # assign map_colors list as colors to use
    marker = list(opacity = .4), # set opacity (alpha) to .4 to help show density better
    type = 'scattermapbox',
    hovertext = map_data[,"City.State"]) # use new column with City and State for hovert
ext

# assign layout and style
spatial <- spatial %>%
  layout(
    mapbox = list(
      style = 'open-street-map',
      zoom =2.5,
      center = list(lon = -98, lat = 34)))

spatial
```

```
## No scattermapbox mode specifed:
##   Setting the mode to markers
##   Read more about this attribute -> https://plot.ly/r/reference/#scatter-mode
```



Loading [MathJax]/extensions/MathZoom.js

Summary Table

```
# create summary that gets the number of stores for each brand in each state
summary <- map_data %>%
  select(Brand, State.Abbr) %>%
  group_by(Brand, State.Abbr) %>%
  summarize(
    stores = length(State.Abbr)
  ) %>%
  rename(State = State.Abbr)

# use dcast to "flip" the summary and have the states and brands as columns, while store
s is the values
# use adorn_totals to create a row at the bottom with total number of stores in the USA
 for each brand
summary_table <- dcast(summary, State~Brand, value.var = "stores")
summary_table <- summary_table %>%
  adorn_totals("row")

summary_table
```

```
##  State Dunkin Donuts Starbucks
##     AK              2         48
##     AL             41         84
##     AR              8         54
##     AZ             96        479
##     CA            116       2782
##     CO             45        477
##     CT            524        118
##     DC             26         91
##     DE             67         24
##     FL           1002        671
##     GA            228        318
##     HI             12         98
##     IA             33         87
##     ID             NA         66
##     IL            695        562
##     IN             87        215
##     KS             29         93
##     KY             38        114
##     LA             14         82
##     MA           1156        262
##     MD            282        252
##     ME            161         29
##     MI             85        272
##     MN             23        175
##     MO             48        181
##     MS              8         32
##     MT             NA         36
##     NC            344        330
##     ND             NA         13
##     NE             19         56
##     NH            220         26
##     NJ            892        249
##     NM             16         75
##     NV             26        249
##     NY           1501        627
##     OH            211        366
##     OK             11         77
##     OR             NA        356
##     PA            647        348
##     RI            175         26
##     SC            119        129
##     SD             NA         24
##     TN            106        175
##     TX            175       1024
##     UT              7         97
##     VA            263        421
##     VT             52          7
##     WA             NA        747
##     WI             64        140
##     WV             23         24
##     WY              1         23
##  Total           9678      13311
```

Google Trends Analysis

Hits is a range from 0 to 100 in relation to the search term's most popular time in the range provided. So, a hits level of 40 means the term is being searched at 40% of the highest its ever been searched.

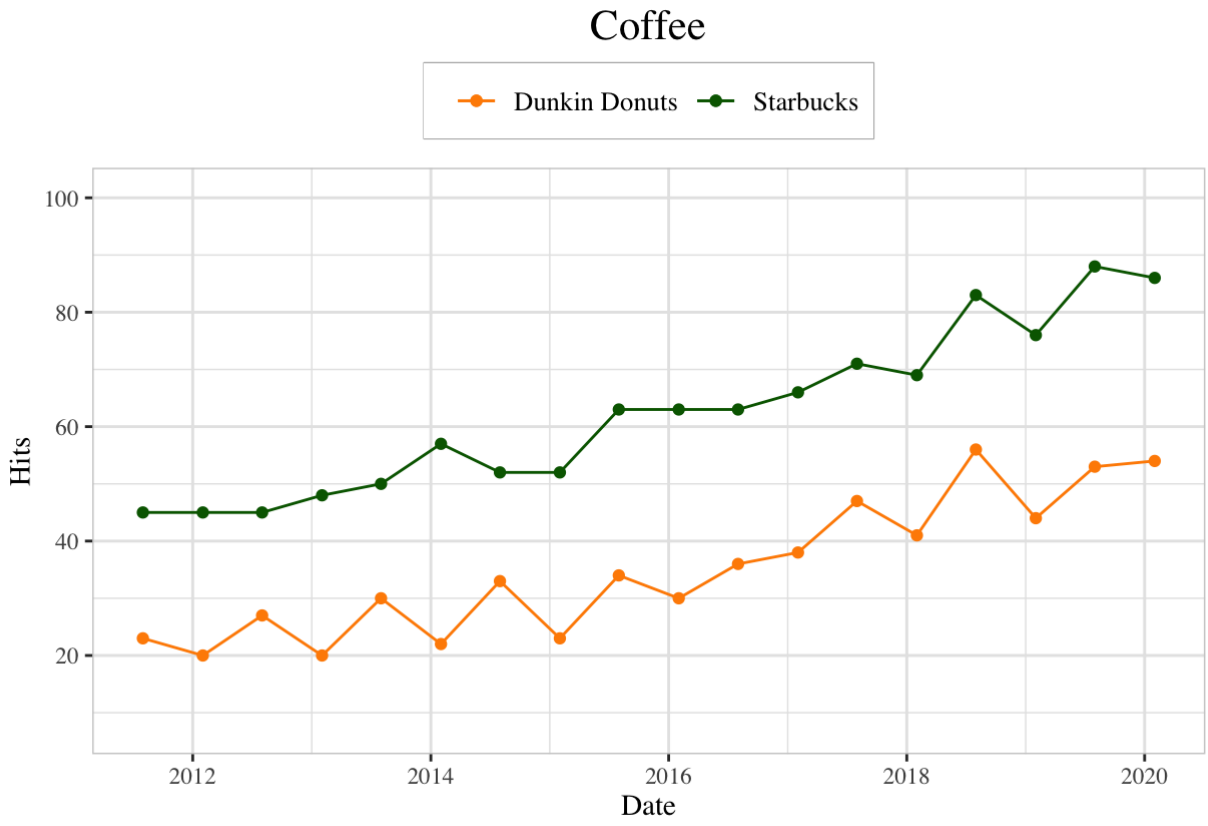
```
# starbucks gtrendR scraping
sbux_food1 <- c("starbucks breakfast", "starbucks lunch", "starbucks sandwich", "starbuc
ks muffin")

sbux_food2 <- c("starbucks bagel", "starbucks bakery", "starbucks beyond meat", "starbuc
ks snacks",
              "starbucks food")

sbux_drinks1 <- c("starbucks latte", "starbucks americano", "starbucks cappucino", "star
bucks macchiato",
                  "starbucks espresso")

sbux_drinks2 <- c("starbucks cold brew", "starbucks iced coffee", "starbucks iced latte"
, "starbucks coffee",
                  "starbucks drinks")

sbux_drinks3 <- c("starbucks hot chocolate", "starbucks tea", "starbucks chai", "starbuc
ks dark roast",
                   "starbucks medium roast")


# run gtrends, use same time frame as stock analysis
# sbux_food_trends1 <- gtrends(keyword = sbux_food1, gprop = "web", geo = "US", time =
 "2011-08-01 2020-04-01")
# sbux_food_trends2 <- gtrends(keyword = sbux_food2, gprop = "web", geo = "US", time =
 "2011-08-01 2020-04-01")
# sbux_drinks_trends1 <- gtrends(keyword = sbux_drinks1, gprop = "web", geo = "US", time
= "2011-08-01 2020-04-01")
# sbux_drinks_trends2 <- gtrends(keyword = sbux_drinks2, gprop = "web", geo = "US", time
= "2011-08-01 2020-04-01")
# sbux_drinks_trends3 <- gtrends(keyword = sbux_drinks3, gprop = "web", geo = "US", time
= "2011-08-01 2020-04-01")

# timeout issues running this as well, variables are stored in loaded rda file "coffee_v
ars", data was scraped using the code above
```

```
# dunkin gtrendR scraping
dnkn_food1 <- c("dunkin breakfast", "dunkin lunch", "dunkin sandwich", "dunkin muffin")

dnkn_food2 <- c("dunkin bagel", "dunkin bakery", "dunkin beyond meat", "dunkin snacks",
              "dunkin food")

dnkn_drinks1 <- c("dunkin latte", "dunkin americano", "dunkin cappucino", "dunkin macchi
ato",
                  "dunkin espresso")

dnkn_drinks2 <- c("dunkin cold brew", "dunkin iced coffee", "dunkin iced latte", "dunkin
coffee",
                  "dunkin drinks")

dnkn_drinks3 <- c("dunkin hot chocolate", "dunkin tea", "dunkin chai", "dunkin dark roas
t",
                   "dunkin medium roast")


# run gtrends, use same time frame as stock analysis
# dnkn_food_trends1 <- gtrends(keyword = dnkn_food1, gprop = "web", geo = "US", time =
 "2011-08-01 2020-04-01")
# dnkn_food_trends2 <- gtrends(keyword = dnkn_food2, gprop = "web", geo = "US", time =
 "2011-08-01 2020-04-01")
# dnkn_drinks_trends1 <- gtrends(keyword = dnkn_drinks1, gprop = "web", geo = "US", time
= "2011-08-01 2020-04-01")
# dnkn_drinks_trends2 <- gtrends(keyword = dnkn_drinks2, gprop = "web", geo = "US", time
= "2011-08-01 2020-04-01")
# dnkn_drinks_trends3 <- gtrends(keyword = dnkn_drinks3, gprop = "web", geo = "US", time
= "2011-08-01 2020-04-01")

# same as starbucks google data, timeout issues running this as well, variables are stor
ed in loaded rda file "coffee_vars", data was scraped using the code above
```

```
# merge starbucks and dunkin into one dataframe each
# select interest_over_time gtrendsR result
# create column with month number, only plot every six months or twice a year


## THIS DATA CLEANING CAN BE SEEN IN FINAL GOOGLE TRENDS DATAFRAMES
## IT IS COMMENTED OUT DUE TO TIMEOUT ISSUES IN ABOVE CHUNKS, BUT THE CHANGES WERE SAVED
AND LOADED IN WITH "coffee_vars.rda"
# sbux_food_trends <- rbind(sbux_food_trends1$interest_over_time, sbux_food_trends2$inte
rest_over_time)
# sbux_drinks_trends <- rbind(sbux_drinks_trends1$interest_over_time, sbux_drinks_trends
2$interest_over_time,
#                             sbux_drinks_trends3$interest_over_time)
#
# dnkn_food_trends <- rbind(dnkn_food_trends1$interest_over_time, dnkn_food_trends2$inte
rest_over_time)
# dnkn_drinks_trends <- rbind(dnkn_drinks_trends1$interest_over_time, dnkn_drinks_trends
2$interest_over_time,
#                             dnkn_drinks_trends3$interest_over_time)
#
# # add month column to dataframes
# sbux_food_trends <- sbux_food_trends %>%
#   mutate(
#     month = as.numeric(format(date, "%m"))
#   )
#
# sbux_drinks_trends <- sbux_drinks_trends %>%
#   mutate(
#     month = as.numeric(format(date, "%m"))
#   )
#
# dnkn_food_trends <- dnkn_food_trends %>%
#   mutate(
#     month = as.numeric(format(date, "%m"))
#   )
#
# dnkn_drinks_trends <- dnkn_drinks_trends %>%
#   mutate(
#     month = as.numeric(format(date, "%m"))
#   )
#
# # change "hits" to numeric rather than factor
# sbux_drinks_trends$hits <- as.numeric(sbux_drinks_trends$hits)
# sbux_food_trends$hits <- as.numeric(sbux_food_trends$hits)
# dnkn_drinks_trends$hits <- as.numeric(dnkn_drinks_trends$hits)
# dnkn_food_trends$hits <- as.numeric(dnkn_food_trends$hits)
#
#
# # set NAs to zero for hits column
# sbux_drinks_trends$hits[is.na(sbux_drinks_trends$hits)] <- 0
# sbux_food_trends$hits[is.na(sbux_food_trends$hits)] <- 0
# dnkn_drinks_trends$hits[is.na(dnkn_drinks_trends$hits)] <- 0
# dnkn_food_trends$hits[is.na(dnkn_food_trends$hits)] <- 0
```

```
#
#
# # create original column to preserve original keyword column
# # use separate split keyword column into brand and item
# # use unite to concatonate items with two words back together
#
# sbux_drinks_trends <- sbux_drinks_trends %>%
#   mutate(
#     original = keyword
#   ) %>%
#   separate(keyword, c("brand", "item1", "item2"), sep = " ") %>%
#   unite("item_total", item1, item2,  sep = " ", na.rm = TRUE)
#
# sbux_food_trends <- sbux_food_trends %>%
#   mutate(
#     original = keyword
#   ) %>%
#   separate(keyword, c("brand", "item1", "item2"), sep = " ") %>%
#   unite("item_total", item1, item2,  sep = " ", na.rm = TRUE)
#
# dnkn_drinks_trends <- dnkn_drinks_trends %>%
#   mutate(
#     original = keyword
#   ) %>%
#   separate(keyword, c("brand", "item1", "item2"), sep = " ") %>%
#   unite("item_total", item1, item2,  sep = " ", na.rm = TRUE)
#
# dnkn_food_trends <- dnkn_food_trends %>%
#   mutate(
#     original = keyword
#   ) %>%
#   separate(keyword, c("brand", "item1", "item2"), sep = " ") %>%
#   unite("item_total", item1, item2,  sep = " ", na.rm = TRUE)
#
#
# # use stringr to make all item_total columns capital first letters
#
# sbux_drinks_trends$item_total <- str_to_title(sbux_drinks_trends$item_total)
# sbux_food_trends$item_total <- str_to_title(sbux_food_trends$item_total)
# dnkn_drinks_trends$item_total <- str_to_title(dnkn_drinks_trends$item_total)
# dnkn_food_trends$item_total <- str_to_title(dnkn_food_trends$item_total)
#
```

```
# example of plot
sbux_coffee <- sbux_drinks_trends %>%
  filter(item_total == "Coffee" & month %in% c(2,8))

dnkn_coffee <- dnkn_drinks_trends %>%
  filter(item_total == "Coffee" & month %in% c(2,8))

ggplot() +
  geom_point(sbux_coffee, mapping = aes(x = date, y = hits, color = "Starbucks")) +
  geom_line(sbux_coffee, mapping = aes(x = date, y = hits, color = "Starbucks", group =
1)) +
  geom_point(dnkn_coffee, mapping = aes(x = date, y = hits, color = "Dunkin Donuts")) +
  geom_line(dnkn_coffee, mapping = aes(x = date, y = hits, color = "Dunkin Donuts", grou
p = 1)) +
  scale_color_manual(name = "", values = c("Starbucks" = "darkgreen", "Dunkin Donuts" =
"darkorange")) +
  labs(
    x = "Date",
    y = "Hits",
    title = "Coffee"
  ) +
  theme_bw() +
  theme(
    text = element_text(family = "Times"),
    plot.title = element_text(hjust = 0.5, size = 16),
    plot.margin = unit(c(1,1,1,1), "cm"),
    legend.position = "top",
    legend.text = element_text(size = 10),
    legend.box.background = element_rect(color = "grey70"),
    panel.border = element_rect(linetype = 'solid', fill = NA, color = "grey80"),
    panel.grid = element_line(color = "grey90", linetype = "solid", lineend = "butt"),
  ) +
  scale_y_continuous(breaks = seq(0,100, by = 20), expand = c(.25,.25))
```



Save dataframes and other variables needed for shiny app as an rda file that can be loaded in

```
# save(sbux, dnkn, map_data, summary_table, sbux_drinks_trends, sbux_food_trends, dnkn_drinks_trends,
#      dnkn_food_trends, file = "coffee_vars.rda" )
```

References

For Starbucks and Dunkin Donuts Data

https://www.kaggle.com/starbucks/store-locations (https://www.kaggle.com/starbucks/store-locations)

https://www.kaggle.com/jpbulman/usa-dunkin-donuts-stores (https://www.kaggle.com/jpbulman/usa-dunkin-donuts-stores)

For Flat json File Conversion to csv Table

https://konklone.io/json/ (https://konklone.io/json/)

For Stock Price Info (scraped with quantmod package)

https://finance.yahoo.com/ (https://finance.yahoo.com/)

For Google Trends Data (scraped with gtrendsR package

https://trends.google.com/trends/?geo=US (https://trends.google.com/trends/?geo=US)