Group 4 Writeup

Reid Jumper, Harsh Nagarkar, and Micheal Davis

12/4/2019

Analysis of Public Coffee Chains

WHO SHOULD YOU INVEST INTO?

Company Description

Quick service restaurant

Starbucks Corporation (formed in 1985) - Seattle Washington - 17 countries. selling coffee, tea and deli items.

DUNKIN' BRANDS GROUP, INC. (formed in 1980) entered in market 2002 Made of Dunkin Donuts (Donuts Breakfast) Baskins and Robins (Icecreams,cakes)

In order to compare these companies strength and fruitfulness we choose to investigate using 3 categories 1. US store competition analysis 2. Stock growth analysis 3. Sec sales and stores analysis

```
knitr::opts_chunk$set(echo = TRUE)
library(webshot)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------------------------------- tidyverse 1.2.1

## v ggplot2 3.2.1     v purrr   0.3.2
## v tibble  2.1.3     v dplyr   0.8.3
## v tidyr   0.8.3     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0

## -- Conflicts ------------------------------------------------------------------ tidyverse_conflicts()
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(socviz)
library(usmap)
library(cowplot)
```

```
##
## ********************************************************

## Note: As of version 1.0.0, cowplot does not change the

##   default ggplot2 theme anymore. To recover the previous

##   behavior, execute:
##   theme_set(theme_cowplot())

## ********************************************************
```

```r
library(maps)
```

```
##
## Attaching package: 'maps'

## The following object is masked from 'package:purrr':
##
##     map
```

```r
library(mapproj)
library(RColorBrewer)
library(lubridate)
```

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

## The following object is masked from 'package:cowplot':
##
##     stamp

## The following object is masked from 'package:base':
##
##     date
```

```r
library(tinytex)
library(knitr)
library(htmltools)
library(leaflet)
library(zoo)
```

```
##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```r
webshot::install_phantomjs
```

```
## function (version = "2.1.1", baseURL = "https://github.com/wch/webshot/releases/download/v0.3.1/",
##     force = FALSE)
## {
##     if (is_phantomjs_version_latest(version) && !force) {
##         message("It seems that the version of `phantomjs` installed is ",
##             "greater than or equal to the requested version.",
##             "To install the requested version or downgrade to another version, ",
##             "use `force = TRUE`.")
##         return(invisible())
##     }
##     if (!grepl("/$", baseURL))
##         baseURL <- paste0(baseURL, "/")
##     owd <- setwd(tempdir())
##     on.exit(setwd(owd), add = TRUE)
##     if (is_windows()) {
##         zipfile <- sprintf("phantomjs-%s-windows.zip", version)
##         download(paste0(baseURL, zipfile), zipfile, mode = "wb")
##         utils::unzip(zipfile)
##         zipdir <- sub(".zip$", "", zipfile)
##         exec <- file.path(zipdir, "bin", "phantomjs.exe")
##     }
##     else if (is_osx()) {
##         zipfile <- sprintf("phantomjs-%s-macosx.zip", version)
##         download(paste0(baseURL, zipfile), zipfile, mode = "wb")
##         utils::unzip(zipfile)
##         zipdir <- sub(".zip$", "", zipfile)
##         exec <- file.path(zipdir, "bin", "phantomjs")
##         Sys.chmod(exec, "0755")
##     }
##     else if (is_linux()) {
##         zipfile <- sprintf("phantomjs-%s-linux-%s.tar.bz2", version,
##             if (grepl("64", Sys.info()[["machine"]]))
##                 "x86_64"
##             else "i686")
##         download(paste0(baseURL, zipfile), zipfile, mode = "wb")
##         utils::untar(zipfile)
##         zipdir <- sub(".tar.bz2$", "", zipfile)
##         exec <- file.path(zipdir, "bin", "phantomjs")
##         Sys.chmod(exec, "0755")
##     }
##     else {
##         message("Sorry, this platform is not supported.")
##         return(invisible())
##     }
##     success <- FALSE
##     dirs <- phantom_paths()
##     for (destdir in dirs) {
##         dir.create(destdir, showWarnings = FALSE)
##         success <- file.copy(exec, destdir, overwrite = TRUE)
##         if (success)
##             break
##     }
##     unlink(c(zipdir, zipfile), recursive = TRUE)
##     if (!success)
##         stop("Unable to install PhantomJS to any of these dirs: ",
##             paste(dirs, collapse = ", "))
##     message("phantomjs has been installed to ", normalizePath(destdir))
##     invisible()
## }
## <bytecode: 0x000000001885d078>
## <environment: namespace:webshot>
```

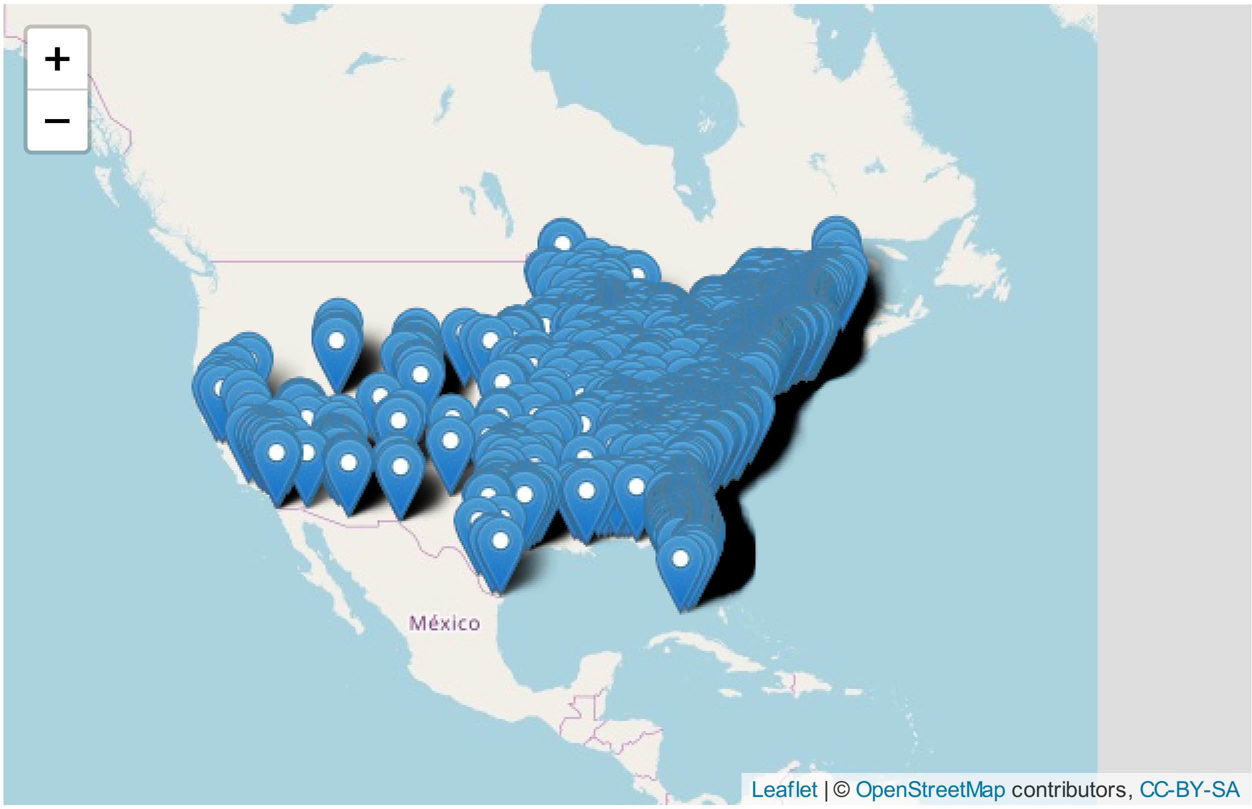
Where are the companies, Dunkin' Donuts and Starbucks, competing?

```r
sbuxloc <- read_csv("data/sbuxlocations.csv")
dnknloc <- read_csv("data/dnknlocations.csv", col_names = FALSE)
```

```r
dnknloc <- dnknloc %>%
  rename(Longitude = X1,
         Latitude = X2,
         Specs = X3,
         Location = X4)
sbuxloc <- sbuxloc %>%
  filter(Country %in% c("US"))

dnknloc <- dnknloc %>%
  mutate(store = "Dunkin")
sbuxloc <- sbuxloc %>%
  mutate(store = "Starbucks")

dnknloc <- dnknloc %>%
  separate(Specs, c("Specs", "State"), sep = ",")

dnknloc <- dnknloc %>%
  mutate(State = dnknloc %>%
           pull(State) %>%
           str_replace_all("\\ ", ""))

dnknLabel <- sprintf("<b>Dunkin'</b><br>%s", dnknloc$State) %>%
  lapply(htmltools::HTML)
sbuxLabel <- sprintf("<b>Starbucks</b><br>%s", sbuxloc$`State/Province`) %>%
  lapply(htmltools::HTML)
```

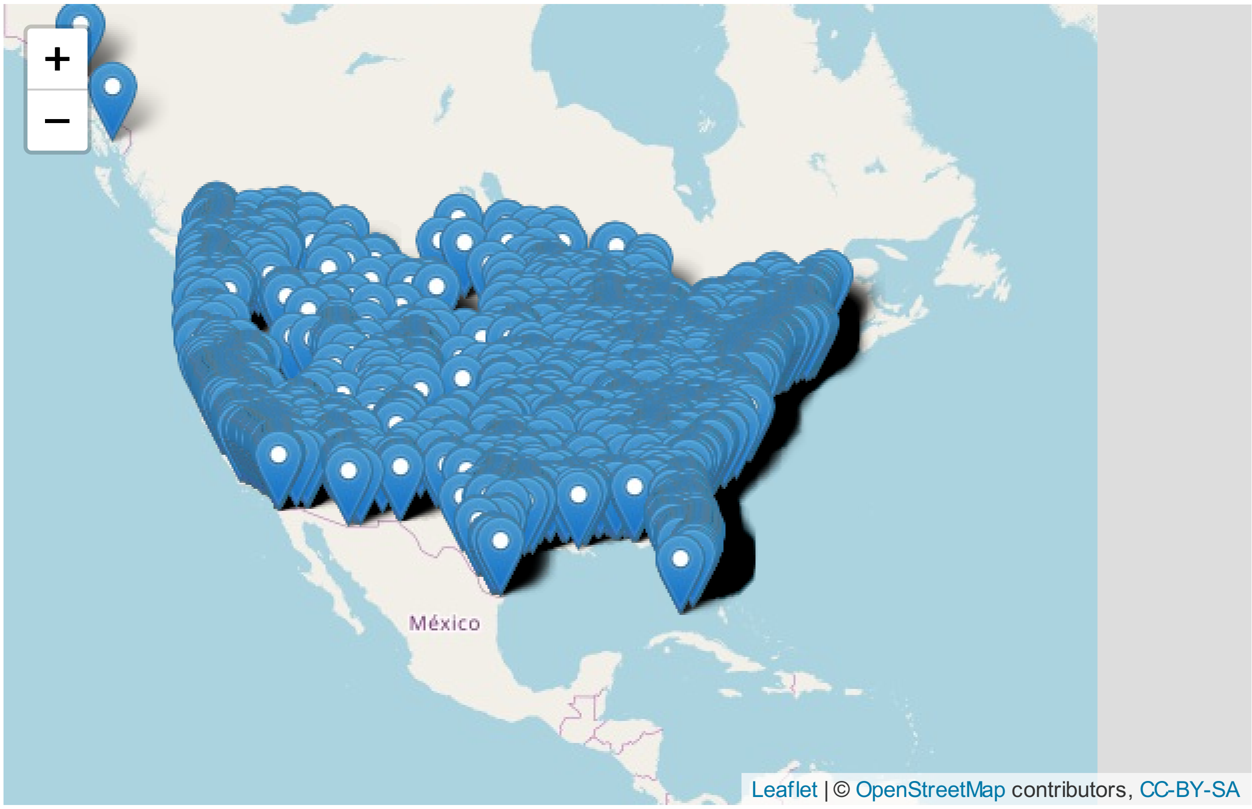
First, it is important to look at where these restaurants are located in the US. # Dunkin' Leaflet

```r
dnknloc %>%
  leaflet(width = "100%", options = leafletOptions(zoomSnap = 0.1)) %>%
  setView(lng = -100, lat = 40, zoom = 3) %>%
  addTiles() %>%
  addMarkers(~Longitude, ~Latitude, popup = dnknLabel, label = dnknLabel)
```



Starbucks Leaflet

```r
sbuxloc %>%
  leaflet(width = "100%", options = leafletOptions(zoomSnap = 0.1)) %>%
  setView(lng = -100, lat = 40, zoom = 3) %>%
  addTiles() %>%
  addMarkers(~Longitude, ~Latitude, popup = sbuxLabel, label = sbuxLabel)
```



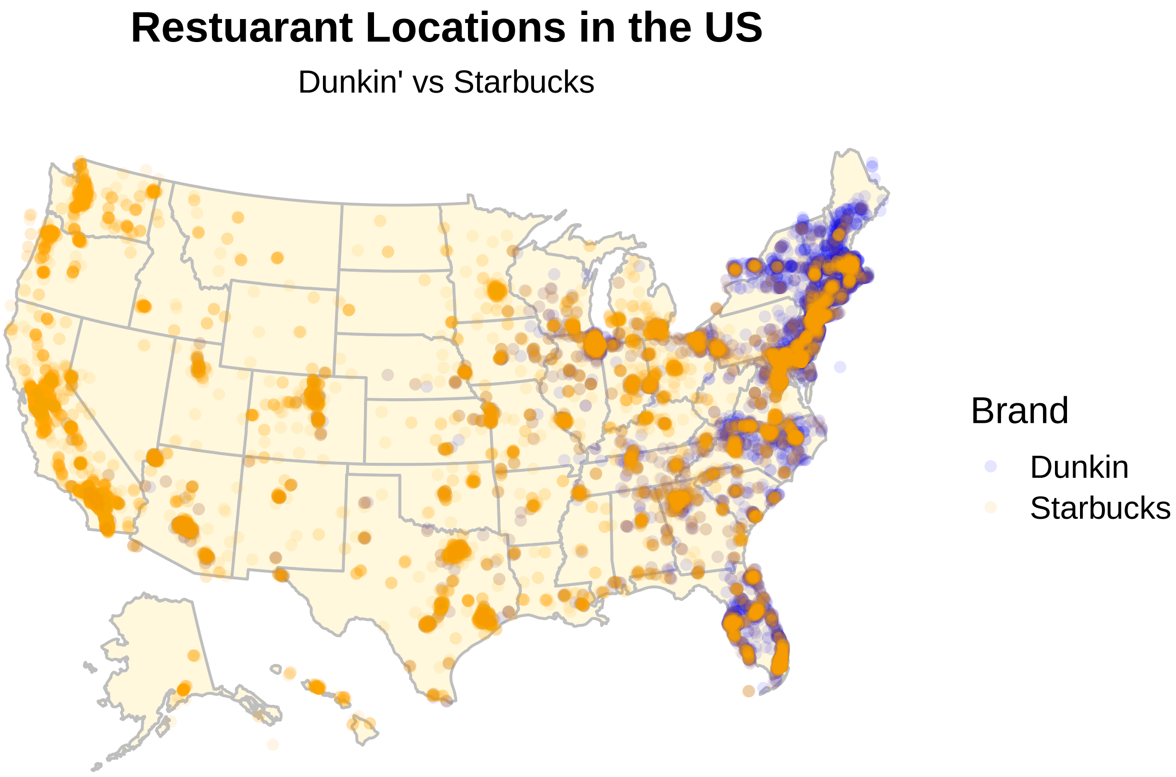
Leaflet | © OpenStreetMap contributors, CC-BY-SA

```
dnkn_sbux <- tibble(
  longitude = c(dnknloc %>%
                 pull(Longitude), sbuxloc %>%
                 pull(Longitude)),
  latitude = c(dnknloc %>%
                pull(Latitude), sbuxloc %>%
                pull(Latitude)),
  store = c(dnknloc %>%
              pull(store), sbuxloc %>%
              pull(store)),
  state = c(dnknloc %>%
              pull(State), sbuxloc %>%
              pull(`State/Province`))
)

scale_coord <- function(abbreviation, x, y) {
  library(usmap)
  min_max <- read_csv("data/min_max_coord.csv")
  values <- tibble(
```

```r
    abbreviation = abbreviation,
    x = x,
    y = y
  )
  usMap <- us_map(regions = "states")
  scaler <- usMap %>%
    group_by(abbr) %>%
    summarize(max.x = max(x),
              max.y = max(y),
              min.x = min(x),
              min.y = min(y))
  scaler <- scaler %>%
    mutate(variation.x = max.x - min.x,
           variation.y = max.y - min.y)
  min_max <- min_max %>%
    mutate(variation.long = max_long - min_long,
           variation.lat = max_lat - min_lat) %>%
    arrange(abbr)
  longitude <- c()
  latitude <- c()
  indexes <- c()
  count <- 1
  for(state in values$abbreviation) {
    for(abbrs in scaler$abbr) {
      if(state == abbrs) {
        indexes <- c(indexes, count)
      }
      count <- count + 1
    }
    count <- 1
  }
  count <- 1
  for(index in indexes) {
    longitude <- c(longitude, ((values$x[count]-min_max$min_long[index])/min_max$variation.long[index])
    latitude <- c(latitude, ((values$y[count]-min_max$min_lat[index])/min_max$variation.lat[index]) * s
    count <- count + 1
  }
  return_tibble <- tibble(
    state = values$abbreviation,
    long = longitude,
    lat = latitude
  )
  return(return_tibble)
}

modified_values <- scale_coord(dnkn_sbux$state, dnkn_sbux$longitude, dnkn_sbux$latitude) %>%
  mutate(Store = dnkn_sbux$store)
```

Locations Graphs

```
ggplot() +
  geom_polygon(data = us_map(regions = "states"), mapping = aes(x = x, y = y, group = group), color = "g
  geom_point(data = modified_values, mapping = aes(x = long, y = lat, color = Store), alpha = 0.1) +
  coord_equal() +
  theme_map() +
  scale_color_manual(values = c("blue", "orange")) +
  labs(title = "Restuarant Locations in the US",
       subtitle = "Dunkin' vs Starbucks") +
  guides(color = guide_legend(title = "Brand")) +
  theme(plot.title = element_text(hjust = 0.5),
        plot.subtitle = element_text(hjust = 0.5))
```

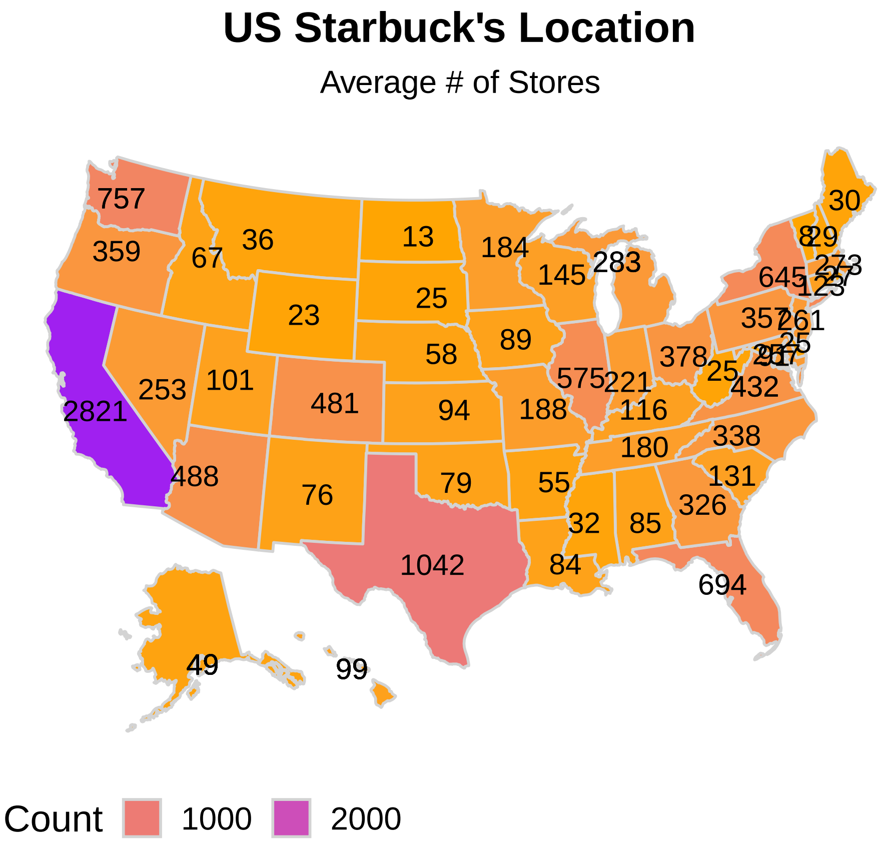


Which states have the highest number of stores for Dunkin Donuts and Starbucks?

```
#gather data from csv files
starbuck_local<- read_csv("data/directory.csv")
starbuck_zip<-read_csv("data/ZIP-COUNTY-FIPS-2018.csv")

#getting states dataset
```

```r
us_states<-map_data("state")

#chaging dataset zip to numeric
starbuck_zip<-starbuck_zip %>%
  mutate(ZIP=as.numeric(ZIP)) %>%
  distinct(CITY, .keep_all = TRUE)

#filtering by country and renaming
starbuck_local <- starbuck_local %>%
  filter(Country == "US") %>%
  rename(
    CITY = City,
    STATE = `State/Province`
  )

#joining datasets by city
data<-left_join(starbuck_local,starbuck_zip,by=c("CITY","STATE"))

#grouping by state and removing na values
data<- data %>%
  group_by(STATE) %>%
  count() %>%
  na.omit()

#renaming states to match join function
us_states<-us_map("states") %>%
  rename(STATE=abbr)

#left joining and renaming variables
data<-left_join(us_states,data,by='STATE') %>%
  rename(count = n)



temp_data <- data %>%
  select(STATE, count) %>%
  group_by(STATE) %>%
  summarise(count = mean(count))

coord <- us_states %>%
  select(x, y, STATE) %>%
  group_by(STATE) %>%
  summarise(
    x_avg = mean(x),
    y_avg = mean(y))
```

```r
new_data <- merge(temp_data, coord, by = "STATE" )

final_data <- merge(new_data, us_states, by = "STATE" )

#plot data for locations
centroid <- aggregate(data=final_data, cbind(x_avg, y_avg) ~ count + group, FUN=mean)
data %>%
ggplot() +
  geom_polygon(mapping = aes(x=x,y=y,group=group,fill=count), color = "lightgray", size = 0.5) +
  coord_equal()+
  theme_map()+
  theme(panel.background = element_blank()) + ggtitle("US Starbuck's Location", subtitle = "Average # o
  theme(legend.position="bottom", plot.title = element_text(hjust = 0.5),plot.subtitle =  element_text(
```



```
data %>%
  distinct(STATE, count) %>%
  arrange(desc(count)) %>%
  head(5)
```

```
##   STATE count
## 1    CA  2821
## 2    TX  1042
## 3    WA   757
## 4    FL   694
## 5    NY   645
```

```
df<- read_csv("data/DD-US.csv",col_names = FALSE)
zip<-read_csv("data/ZIP-COUNTY-FIPS-2018.csv")
county_map <- county_map
us_states<-map_data("state")
```

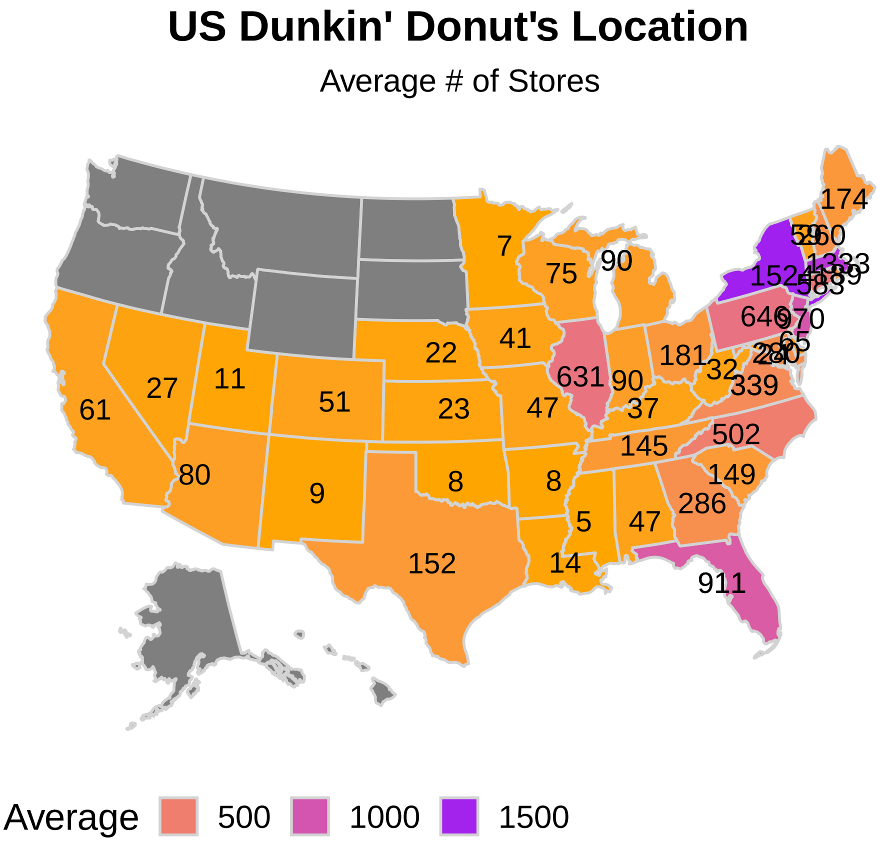
```r
zip<-zip %>%
  mutate(ZIP=as.numeric(ZIP))



df <- df %>%
  mutate(ZIP = sapply(strsplit(df$X4, split='|', fixed=TRUE), function(x) (x[2])))

df <- df %>%
  mutate(DATA = sapply(strsplit(df$ZIP, split=c(','), fixed=TRUE), function(x) (x[2])))

df <- df %>%
  mutate(ZIP = sapply(strsplit(df$DATA, split=c(' '), fixed=TRUE), function(x) (x[2])))
```

```r
df<-df %>%
  mutate(ZIP=as.numeric(ZIP))
```

```
## Warning: NAs introduced by coercion
```

```r
data<-left_join(df,zip, by=c("ZIP")) %>%
  group_by(STATE) %>%
  count() %>%
  na.omit()

us_states<-us_map("states") %>%
  rename(STATE=abbr)

data<-left_join(us_states,data,by='STATE')


temp_data <- data %>%
  select(STATE, n) %>%
  group_by(STATE) %>%
  summarise(n = mean(n))

coord <- us_states %>%
  select(x, y, STATE) %>%
  group_by(STATE) %>%
  summarise(
    x_avg = mean(x),
    y_avg = mean(y))

new_data <- merge(temp_data, coord, by = "STATE" )

final_data <- merge(new_data, us_states, by = "STATE" )
```

```r
#plot data for locations
centroid <- aggregate(data=final_data, cbind(x_avg, y_avg) ~ n + group, FUN=mean)

data%>%
  ggplot() +
  geom_polygon(mapping = aes(x=x,y=y,group=group,fill=n),color = "lightgray", size = 0.5) +
  coord_equal()+
  theme_map()+
  theme(panel.background = element_blank()) + ggtitle("US Dunkin' Donut's Location", subtitle = "Averag
  scale_fill_gradient(low = "orange", high = "purple", guide = guide_legend()) + geom_text(data = centr
```



```r
data %>%
  arrange(desc(n)) %>%
  distinct(STATE, count) %>%
  head(5)
```

```
## Warning: Trying to compute distinct() for variables not found in the data:
## - `count`
## This is an error, but only a warning is raised for compatibility reasons.
## The following variables will be used:
## - STATE
```

```
##   STATE
## 1    NY
## 2    MA
## 3    NJ
## 4    FL
## 5    PA
```

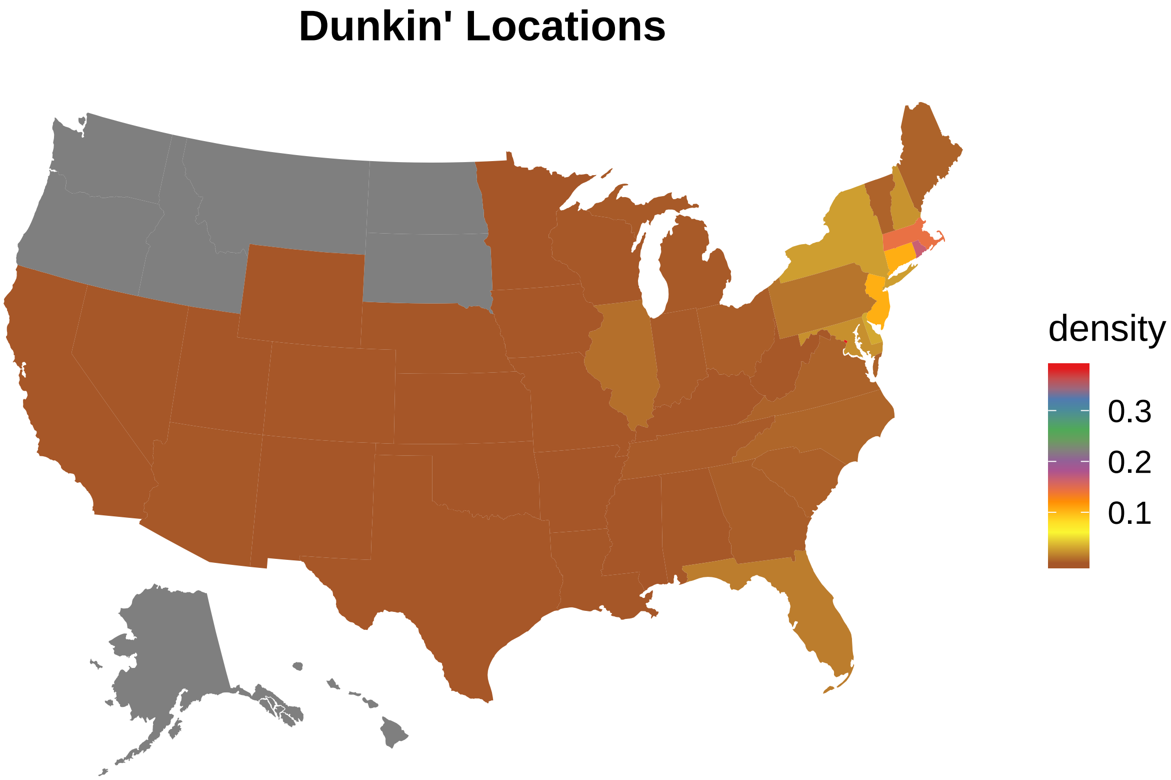
```
#------------------------------------------------------------------------------------------------------
# by county

df<- read_csv("data/DD-US.csv",col_names = FALSE)
zip<-read_csv("data/ZIP-COUNTY-FIPS-2018.csv")

cmap<-county_map
us_states<-map_data("state")

zip<-zip %>%
  mutate(ZIP=as.numeric(ZIP))

df<-df %>%
  mutate(ZIP=as.numeric(str_match(X4,"(\\d{5})")[1]))

data<-left_join(zip,df,by="ZIP") %>%
  mutate(COUNTYNAME=str_remove(COUNTYNAME," County")) %>%
  rename(county=COUNTYNAME) %>%
  mutate(county=tolower(county)) %>%
  group_by(county, STATE) %>%
  count() %>%
  na.omit()

data <- data %>%
  summarise(s= max(n)) %>%
  arrange(desc(s)) %>%
  na.omit()

r <- us_map(regions = "counties")

r <- r %>%
  rename(
    STATE = abbr
  )
r <-  r %>%
  mutate(county = str_remove(county, " County"))

r <- r %>%
  mutate(county = tolower(county))

r <- r %>%
  group_by(STATE, county)

data<-left_join(r,data, by = c("STATE","county"))

new_data <- data %>%
  select(STATE, county, s) %>%
  group_by(STATE,county) %>%
  summarise(count= max(s)) %>%
  arrange(desc(count)) %>%
  na.omit() %>%
  head(20)
```

```r
sq_mileage <- read_csv("data/mileage.csv")
DNKN_loc <- read_csv("data/DD-US.csv", col_names = FALSE)
SBUX_loc <- read_csv("data/directory.csv")

SBUX_loc <- SBUX_loc %>%
  filter(Country %in% c("US")) %>%
  rename(state = `State/Province`) %>%
  count(state)
DNKN_loc <- DNKN_loc %>%
  rename(specs = X3) %>%
  separate(specs, c("specs", "state"), sep = ",")
DNKN_loc <- DNKN_loc %>%
  mutate(state = DNKN_loc %>%
           pull(state) %>%
           str_replace_all("// ", "")) %>%
  count(state)

DNKN_loc <- left_join(us_map(region = "states") %>%
                        rename(state = abbr), DNKN_loc, by = "state")
SBUX_loc <- left_join(us_map(region = "states") %>%
                        rename(state = abbr), SBUX_loc, by = "state")

sq_mileage <- sq_mileage %>%
  rename(full = state,
         state_area = mileage)

DNKN_loc <- left_join(DNKN_loc, sq_mileage, by = "full")
SBUX_loc <- left_join(SBUX_loc, sq_mileage, by = "full")

DNKN_loc <- DNKN_loc %>%
  mutate(density = n/state_area)
SBUX_loc <- SBUX_loc %>%
  mutate(density = n/state_area)
```

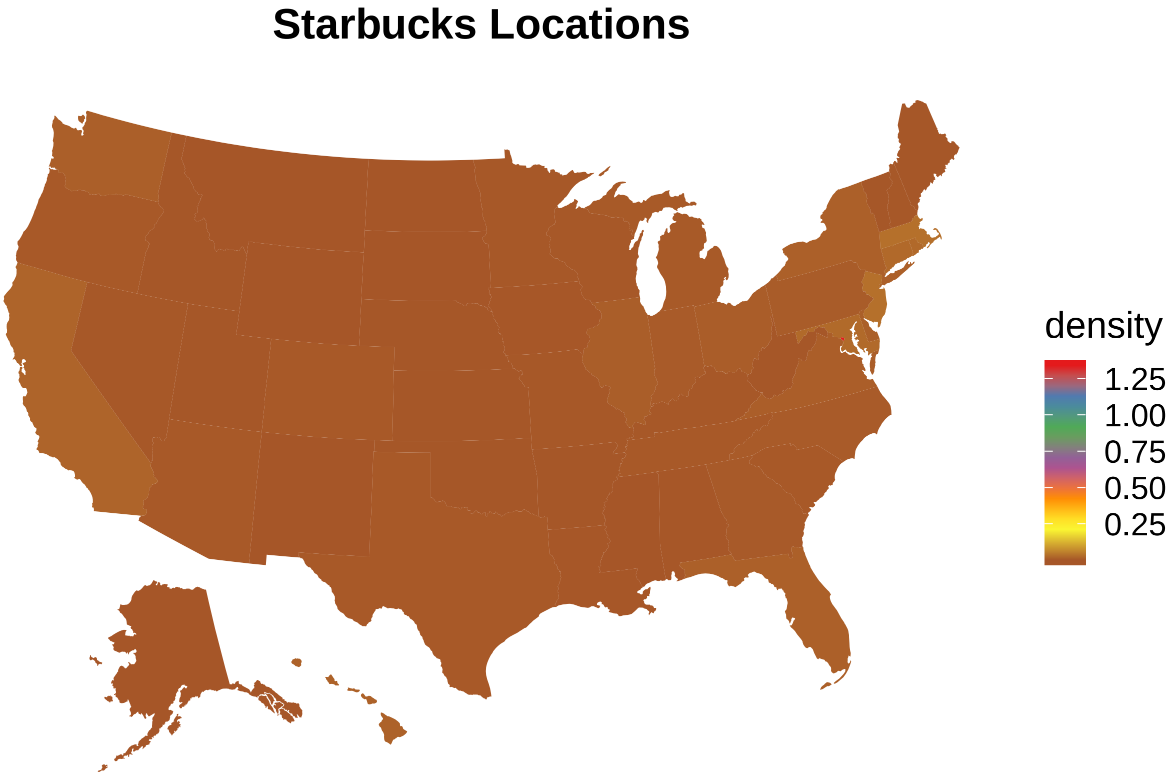
What states have the highest densities of Dunkin’ and Starbucks?

When Washington D.C. is included in the density plots, we noticed that the data is heavily skewed, so in order to get a proper look at the densities of these companies, Washington D.C. must be filtered out.

```r
DNKN_loc %>%
  ggplot(mapping = aes(x = x, y = y, group = group, fill = density)) +
  geom_polygon() +
  coord_equal() +
  theme_map() +
  scale_fill_distiller(palette = "Set1") +
  labs(title = "Dunkin' Locations") +
  theme(plot.title = element_text(hjust = 0.5))
```

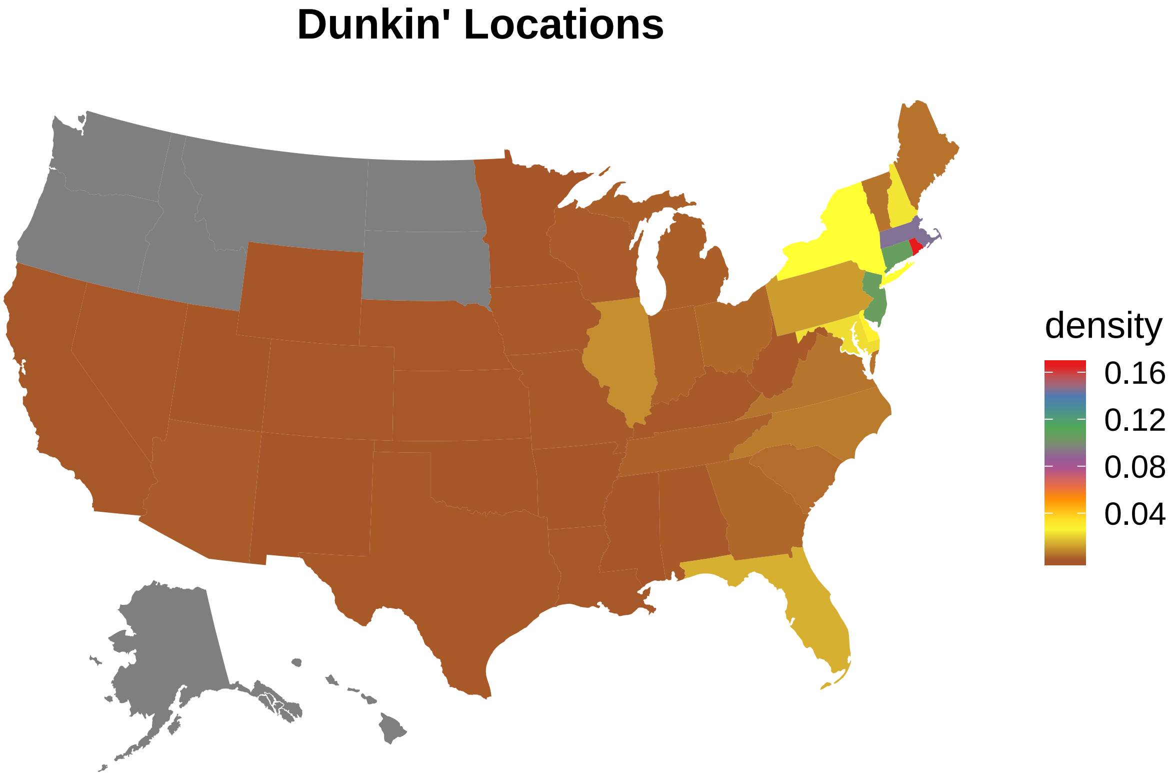


```r
SBUX_loc %>%
  ggplot(mapping = aes(x = x, y = y, group = group, fill = density)) +
  geom_polygon() +
  coord_equal() +
  theme_map() +
  scale_fill_distiller(palette = "Set1") +
  labs(title = "Starbucks Locations") +
  theme(plot.title = element_text(hjust = 0.5))
```

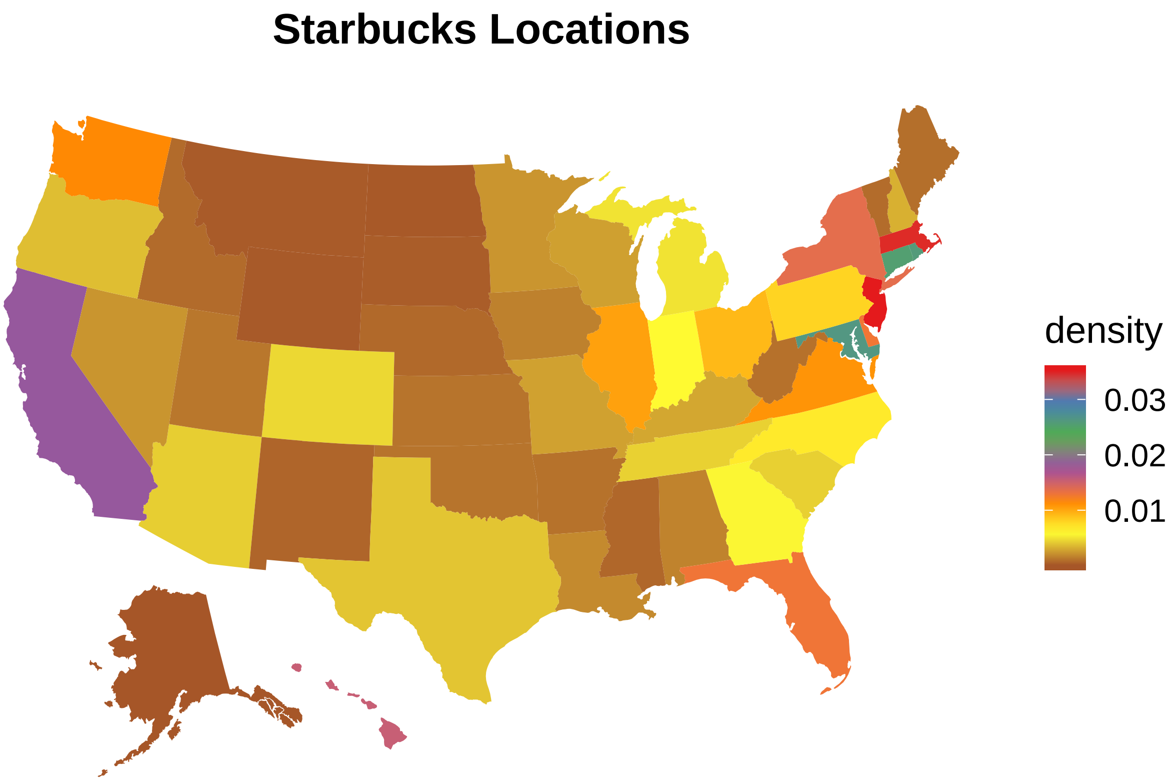


```
DNKN_loc <- DNKN_loc %>%
  filter(state %nin% c("DC"))
SBUX_loc <- SBUX_loc %>%
  filter(state %nin% c("DC"))
```

```
DNKN_loc %>%
  ggplot(mapping = aes(x = x, y = y, group = group, fill = density)) +
  geom_polygon() +
  coord_equal() +
  theme_map() +
  scale_fill_distiller(palette = "Set1") +
  labs(title = "Dunkin' Locations") +
  theme(plot.title = element_text(hjust = 0.5))
```



```r
SBUX_loc %>%
  ggplot(mapping = aes(x = x, y = y, group = group, fill = density)) +
  geom_polygon() +
  coord_equal() +
  theme_map() +
  scale_fill_distiller(palette = "Set1") +
  labs(title = "Starbucks Locations") +
  theme(plot.title = element_text(hjust = 0.5))
```



```
DNKN_loc %>%
  distinct(state, density) %>%
  arrange(desc(density)) %>%
  top_n(5)
```

```
##   state    density
## 1    RI 0.16555024
## 2    MA 0.14413265
## 3    CT 0.10505676
## 4    NJ 0.10475934
## 5    DE 0.03121801
```

```
SBUX_loc %>%
  distinct(state, density) %>%
  arrange(desc(density)) %>%
  top_n(5)
```

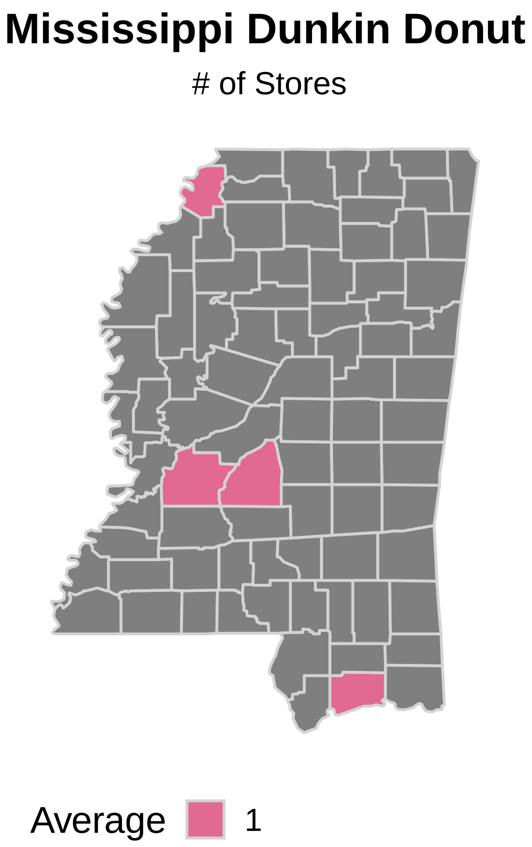
```
##   state    density
## 1    NJ 0.03518943
## 2    MA 0.03482143
## 3    MD 0.02629425
## 4    RI 0.02583732
## 5    CT 0.02538700
```

The most competition between Dunkin' and Starbucks is located primarily in the North and on the East Coast. The least competition is in the north-west where Dunkin' does not have any stores.

```
df<- read_csv("data/DD-US.csv",col_names = FALSE)
zip<-read_csv("data/ZIP-COUNTY-FIPS-2018.csv")
county_map <- map_data("county") %>%
  filter(region=="mississippi")
us_states<-map_data("state")

zip<-zip %>%
  mutate(ZIP=as.numeric(ZIP))

df <- df %>%
  mutate(ZIP = sapply(strsplit(df$X4, split='|', fixed=TRUE), function(x) (x[2])))

df <- df %>%
  mutate(DATA = sapply(strsplit(df$ZIP, split=c(','), fixed=TRUE), function(x) (x[2])))

df <- df %>%
  mutate(ZIP = sapply(strsplit(df$DATA, split=c(' '), fixed=TRUE), function(x) (x[2])))


df<-df %>%
  mutate(ZIP=as.numeric(ZIP))
```

```
## Warning: NAs introduced by coercion
```

```
data<-left_join(df,zip, by=c("ZIP")) %>%
  mutate(COUNTYNAME=tolower(str_remove(COUNTYNAME," County"))) %>%
  group_by(STATE) %>%
  na.omit() %>%
  filter(STATE=='MS') %>%
  group_by(COUNTYNAME) %>%
  count()

data<-data%>%
  rename(subregion=COUNTYNAME)


data<-left_join(county_map,data,by='subregion')


data %>%
  ggplot()+
  geom_polygon(mapping = aes(x=long,y=lat,group=group,fill=n),color = "lightgray", size = 0.5) +
  coord_equal()+
  theme_map()+
  theme(panel.background = element_blank()) + ggtitle("Mississippi Dunkin Donut", subtitle = " # of Sto
  scale_fill_gradient(low = "orange", high = "purple", guide = guide_legend())
```



```
data %>%
  na.omit() %>%
  distinct(subregion, n) %>%
  select(subregion, n) %>%
  arrange(desc(n))
```

```
##   subregion n
## 1  harrison 1
## 2     hinds 1
## 3    rankin 1
## 4    tunica 1
```

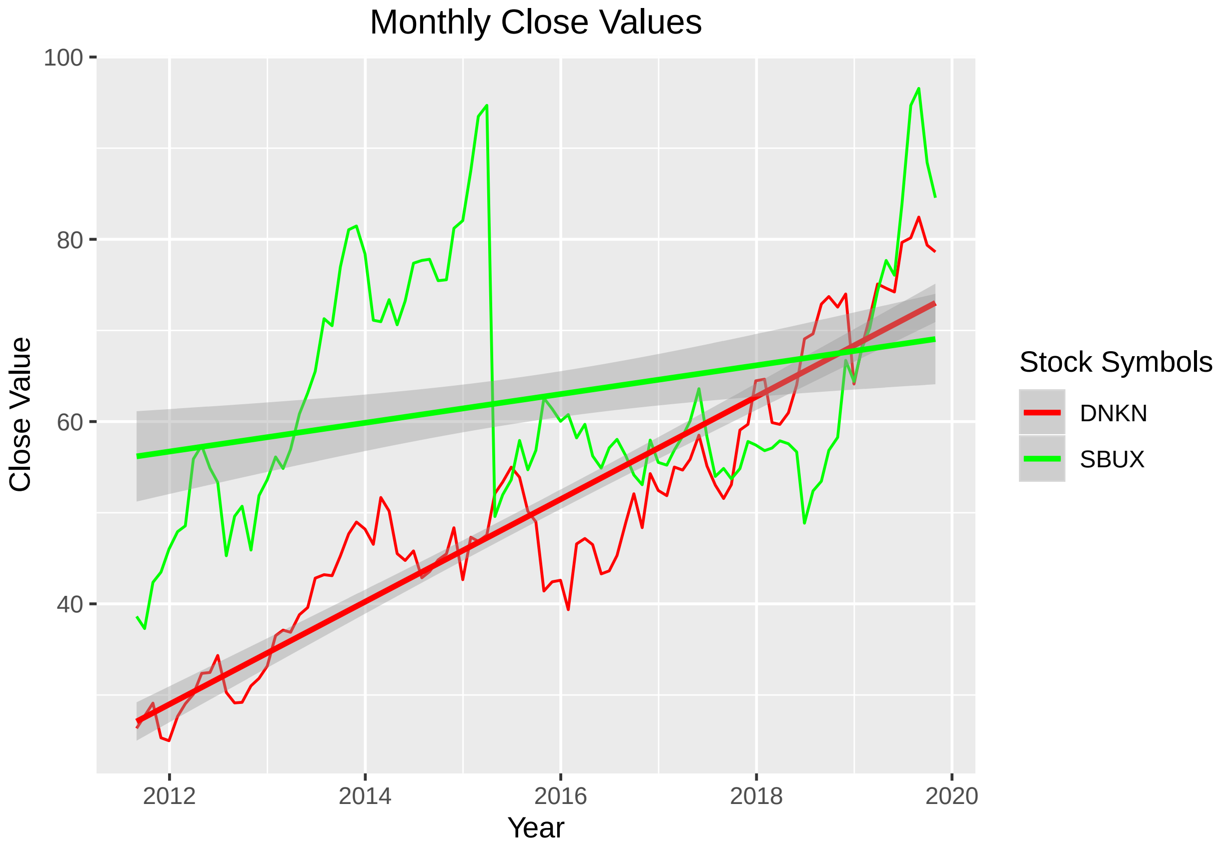
###Do the stocks of Starbucks and Dunkin’ Donuts comply with the Warren Buffet Rule?

Warren claimed that over the period of time the average yearly yeild of a company is in 6-7%.

```
DNKN <- read_csv("data/modified_DNKN.csv")
SBUX <- read_csv("data/modified_SBUX.csv")

DNKN <- DNKN %>%
  mutate(stock = "DNKN")
SBUX <- SBUX %>%
  mutate(stock = "SBUX")

D_S <- tibble(
  timestamp = c(DNKN %>%
                  pull(timestamp),
                SBUX %>%
                  pull(timestamp)),
  stock = c(DNKN %>%
              pull(stock),
            SBUX %>%
              pull(stock)),
  close = c(DNKN %>%
              pull(close),
            SBUX %>%
              pull(close))
)
```

Monthly Stock Close

```
D_S %>%
  group_by(stock) %>%
  ggplot() +
  geom_line(mapping = aes(x = timestamp, y = close, color = stock)) +
  scale_color_manual(values = c("red", "green")) +
  geom_smooth(mapping = aes(timestamp,close,color=stock),method = "lm") +
  labs(title = "Monthly Close Values",
       x = "Year",
       y = "Close Value") +
  guides(color = guide_legend(title = "Stock Symbols")) +
  theme(plot.title = element_text(hjust = 0.5))
```



#Prediction Values

```
Starbucks_intercepts<-lm(SBUX$close~SBUX$timestamp)[1]$coefficient
Starbucks_monthly_average_growth_rate<-(-atan(Starbucks_intercepts[1]/Starbucks_intercepts[2]))

Dunkin_intercepts<-lm(DNKN$close~DNKN$timestamp)[1]$coefficient
Dunkin_monthly_average_growth_rate<-(-atan(Dunkin_intercepts[1]/Dunkin_intercepts[2]))


Starbucks_average_growth_rate<-(tail(SBUX,1)$close-head(SBUX,1)$close)/tail(SBUX,1)$close*100/7

Dunkin_average_growth_rate<-(tail(DNKN,1)$close-head(DNKN,1)$close)/tail(DNKN,1)$close*100/7
```

Yearly Growth rate for Starbucks and Dunkin respectively

```
Starbucks_average_growth_rate
```

```
## [1] -16.99342
```

```
Dunkin_average_growth_rate
```

```
## [1] -28.35449
```

Dunkin exceed a bit where as starbucks is almost in range. Starbuck stock yeild is coming down because of the stock split happening in 2015.

What would the monthly growth rate be for Starbucks and Dunkin be?

```
Starbucks_monthly_average_growth_rate
```

```
## (Intercept)
##    1.570342
```

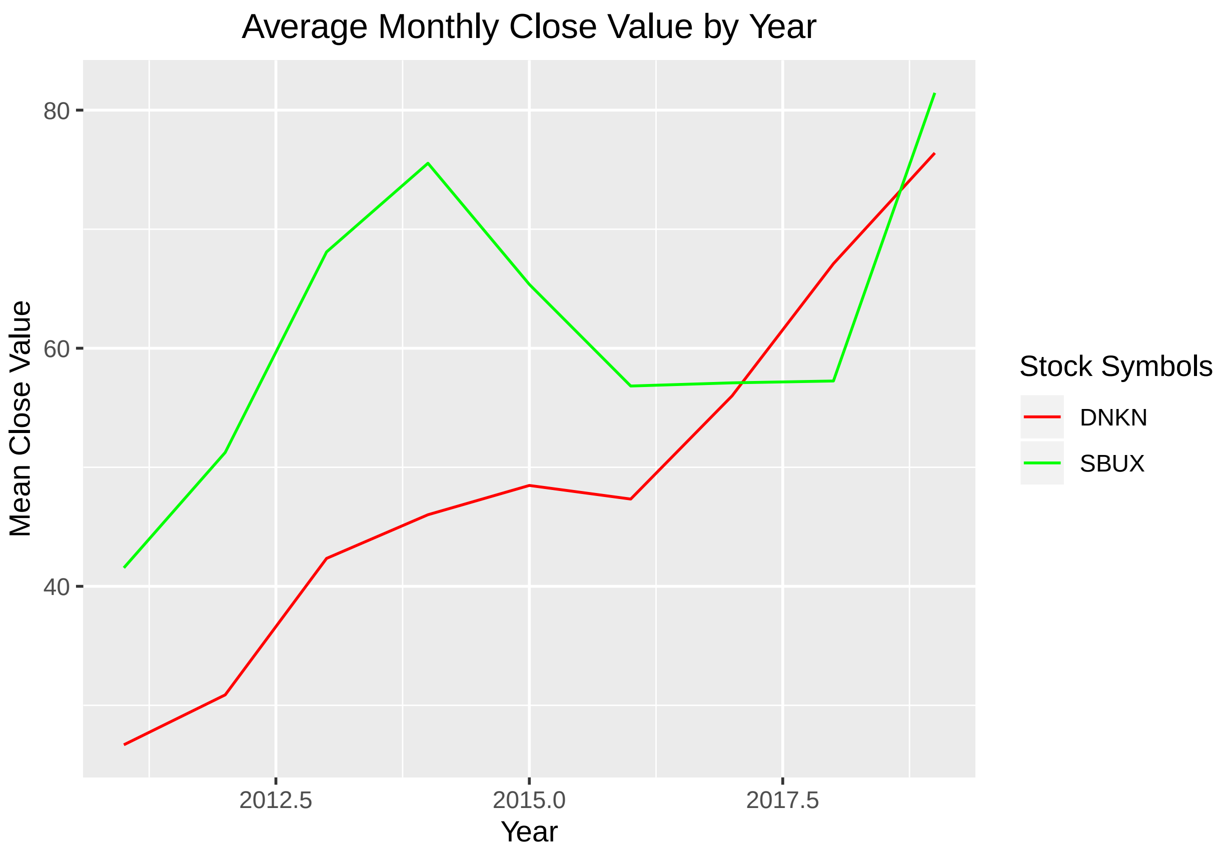
```
Dunkin_monthly_average_growth_rate
```

```
## (Intercept)
##    1.570722
```

```
yrly <- D_S %>%
  mutate(year = year(timestamp)) %>%
  group_by(year, stock) %>%
  summarize(mean_close = mean(close))
```

Yearly Mean Stock Close

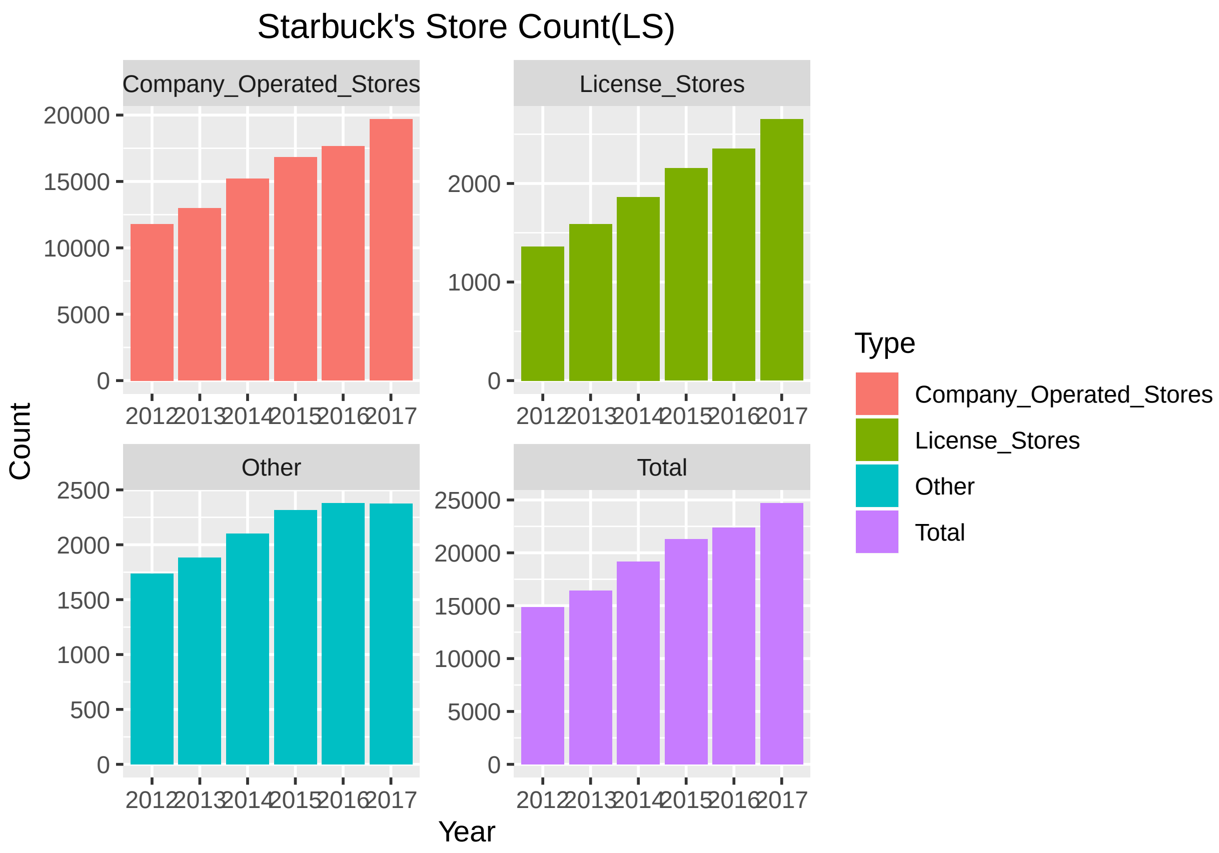
```
yrly %>%
  group_by(stock) %>%
  ggplot() +
  geom_line(mapping = aes(x = year, y = mean_close, color = stock)) +
  scale_color_manual(values = c("red", "green")) +
  labs(title = "Average Monthly Close Value by Year",
       x = "Year",
       y = "Mean Close Value") +
  guides(color = guide_legend(title = "Stock Symbols")) +
  theme(plot.title = element_text(hjust = 0.5))
```



What are some patterns that show in the data of either Dunkin' or Starbucks?

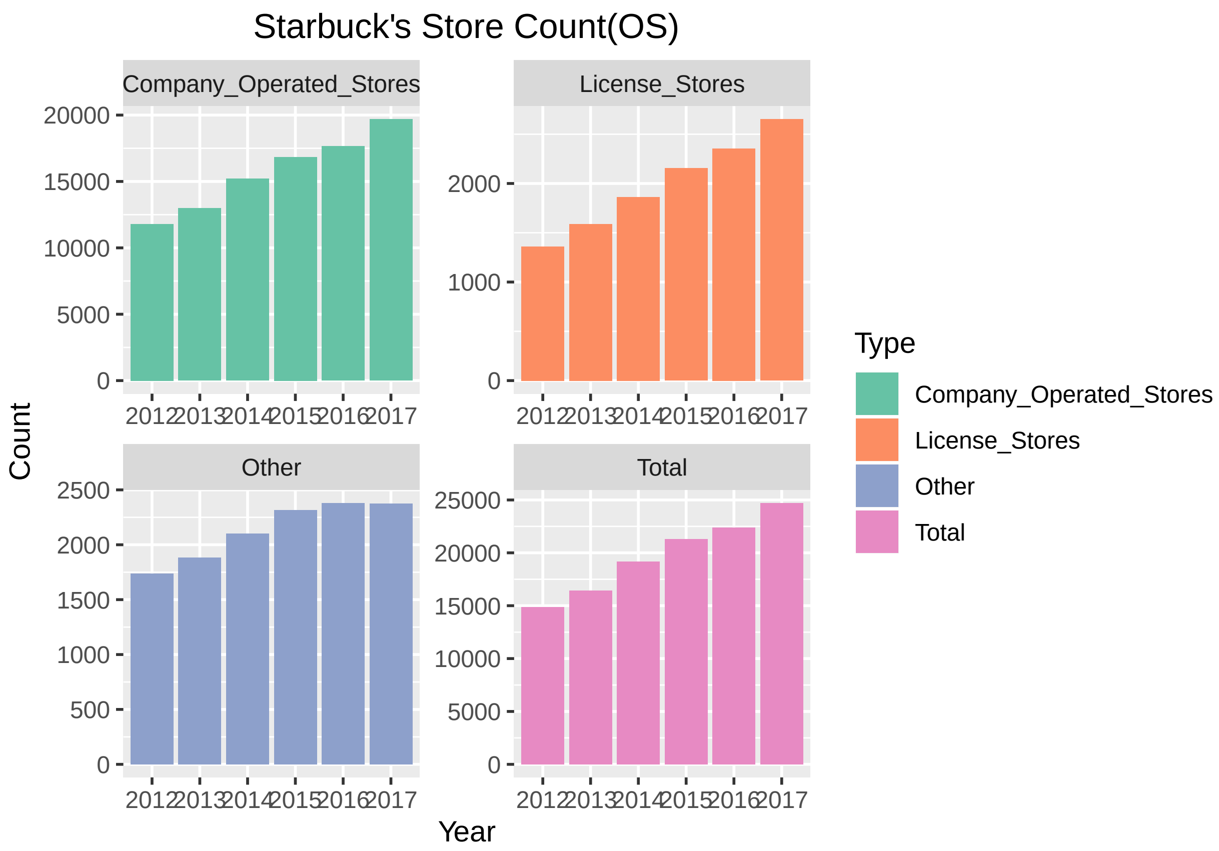
```
#Revenue of company stores
starbuck_rev<- read_csv("data/starbucks.csv")

starbuck_rev <- starbuck_rev %>%
  select(year, Company_Operated_Stores:Total_OS)

starbuck_rev <- starbuck_rev %>%
  select(Company_Operated_Stores, License_Stores, Other, Total,year) %>%
  gather(Type, count, Company_Operated_Stores, License_Stores, Other, Total) %>%
  na.omit()

starbuck_rev <- starbuck_rev %>%
  group_by(Type, year) %>%
  summarise(mean = mean(count))
```

```
starbuck_rev %>%
  ggplot() +
  geom_col(mapping = aes(reorder(year, mean),
                         y = mean, fill=Type)) + labs(title = "Starbuck's Store Count(LS)")+
  theme(plot.title = element_text(hjust = 0.5)) + xlab("Year") + ylab("Count")  +  facet_wrap(~Type, nr
```



Starbucks Store Count

```
starbuck_rev %>%
  ggplot() +
  geom_col(mapping = aes(reorder(year, mean),
                       y = mean, fill=Type)) + labs(title = "Starbuck's Store Count(OS)")+
  theme(plot.title = element_text(hjust = 0.5)) + xlab("Year") + ylab("Count") + scale_fill_brewer(pale
```



At the visual inspection from 2015 we saw the CAP segment license dropping and comapany owned increasing. On further analysis we concluded that the business model of the Starbucks store is license where company creates company owned stores and then licenses in bulk. 2015 was the only time they happened in reverse and we could track this detail down.

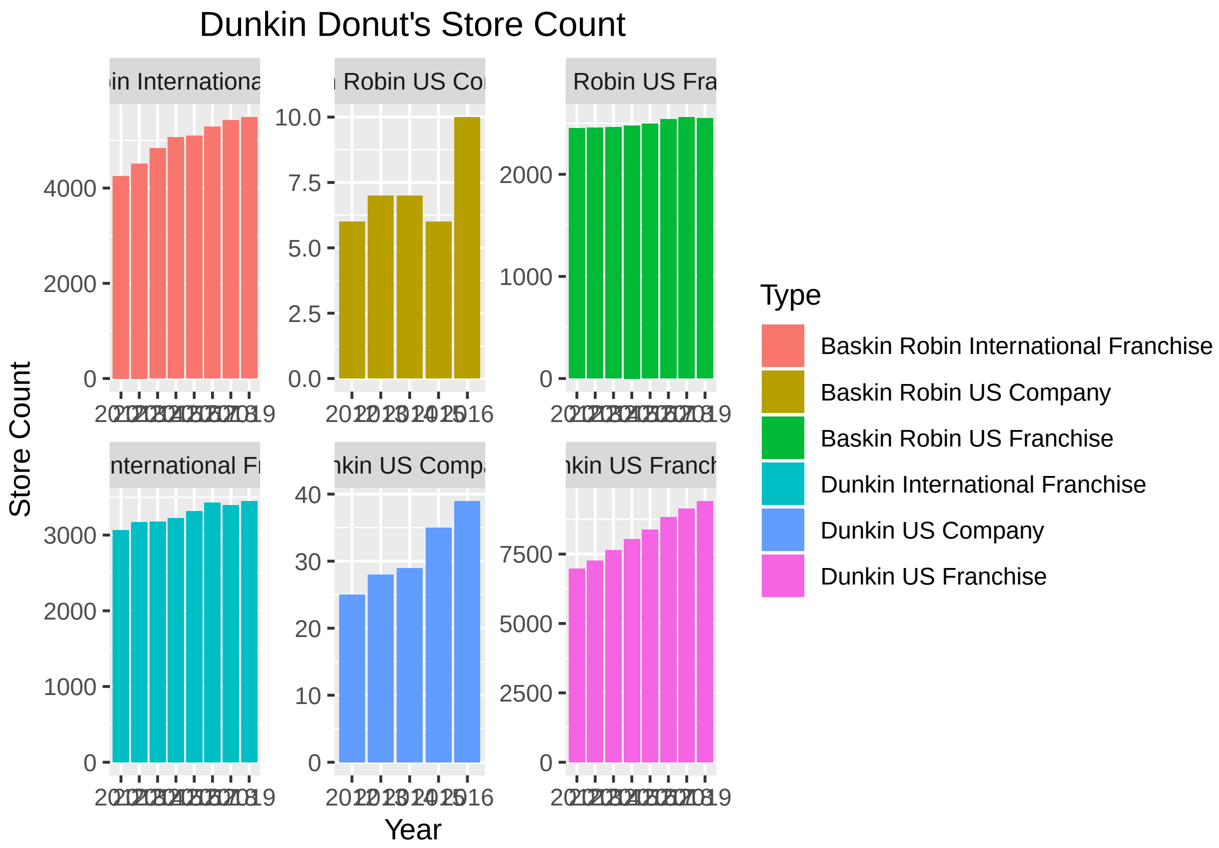
```
dunk_rev<- read_csv("data/dunkin.csv")

dunk_rev <- dunk_rev %>%
  select(year_of_publish, DD_US_F:BR_US_C, Franchise_income, Rental_income:Total_reveneu)

dunk_rev <- dunk_rev %>%
  rename(
    `Dunkin US Franchise` = DD_US_F,
    `Dunkin International Franchise` = DD_IN_F,
    `Baskin Robin US Franchise` = BR_US_F,
    `Baskin Robin International Franchise` = BR_IN_F,
    `Baskin Robin US Company` = BR_US_C,
    `Dunkin US Company` = DD_US_C
  )
```

```
dunk_rev <- dunk_rev %>%
  select(`Dunkin US Franchise`, `Dunkin International Franchise`, `Baskin Robin US Franchise`,  `Baskin
  gather(Type, count,`Dunkin US Franchise`, `Dunkin International Franchise` , `Baskin Robin US Franchi
  na.omit()
```

```
dunk_rev <- dunk_rev%>%
  group_by(Type, year_of_publish) %>%
  summarise(mean = mean(count))
```

Dunkin Donuts Store Count

```
dunk_rev %>%
    ggplot() +
    geom_col(mapping = aes(reorder(year_of_publish, mean),
                             y = mean, fill=Type)) +
    theme(plot.title = element_text(hjust = 0.5)) + ylab("Store Count") + xlab("Year")+ labs(title = "Du
```



At the visual inspection of the Dunkin company stores went down to zero in 2017. Upon further inspection we figured out that the Dunkin model is franchise based and it changed to 100% franchise in 2016 dropping all it's ompany operated stores.

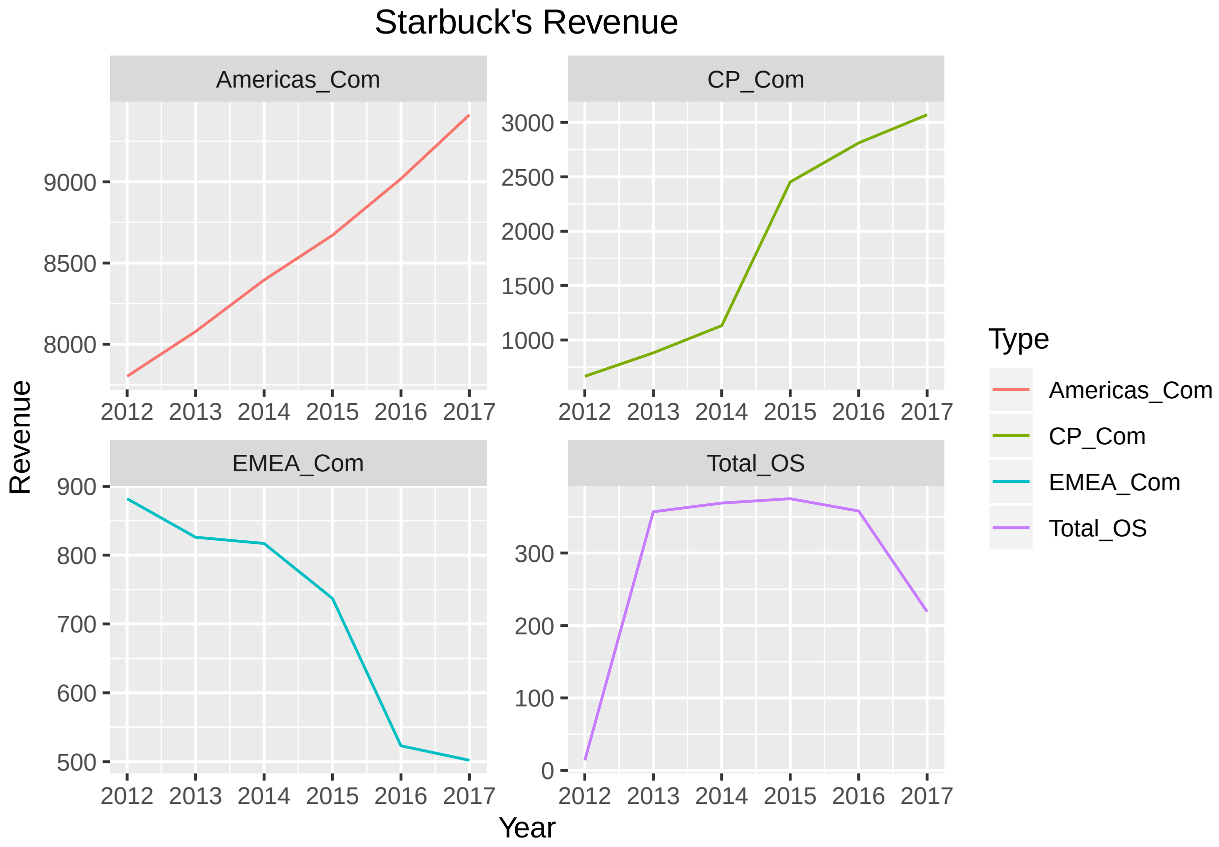
```
starbuck_rev<- read_csv("data/starbucks.csv")

starbuck_rev <- starbuck_rev %>%
  select(year, Company_Operated_Stores:Total_OS)

starbuck_rev <- starbuck_rev %>%
  select(Americas_Licensed, EMEA_Licensed, CP_Licensed, Total_LS, year) %>%
  gather(Type, count, Americas_Licensed, EMEA_Licensed, CP_Licensed, Total_LS) %>%
  na.omit()

starbuck_rev <- starbuck_rev %>%
  group_by(Type, year) %>%
  summarise(mean = mean(count))
```

```
#revenue by country(com)

starbuck_rev<- read_csv("data/starbucks.csv")

starbuck_rev <- starbuck_rev %>%
  select(year, Company_Operated_Stores:Total_OS)

starbuck_rev <- starbuck_rev %>%
  select(Americas_Com, EMEA_Com, CP_Com, Total_OS, year) %>%
  gather(Type, count, Americas_Com, EMEA_Com, CP_Com, Total_OS) %>%
  na.omit()

starbuck_rev <- starbuck_rev %>%
  group_by(Type, year) %>%
  summarise(mean = mean(count))

# starbuck_rev %>%
#   ggplot() +
#   geom_col(mapping = aes(reorder(year, mean),
#                          y = mean, fill=Type)) + labs(title = "Starbuck's Reveue/Country(OS)")+
#   theme(plot.title = element_text(hjust = 0.5)) + xlab("Year") + ylab("Revenue") + scale_fill_brewer(
```

Starbucks Revenue

We found these patterns by analzing visually. By inspection, these resulted from analyzing all 10-K files data.

```
starbuck_rev %>%
group_by(Type) %>%
  ggplot(mapping = aes(x = year, y = mean)) +
  geom_line(mapping = aes(color = Type)) + ylab("Revenue") + xlab("Year")+ facet_wrap(~Type, nrow = 3,s
```



2014- Opens first Starbucks Reserve® Roastery and Tasting Room in Seattle. Launches Starbucks Mobile Order & Pay.

2015- Evolution Fresh Stores and Teavana tea massive close-downs

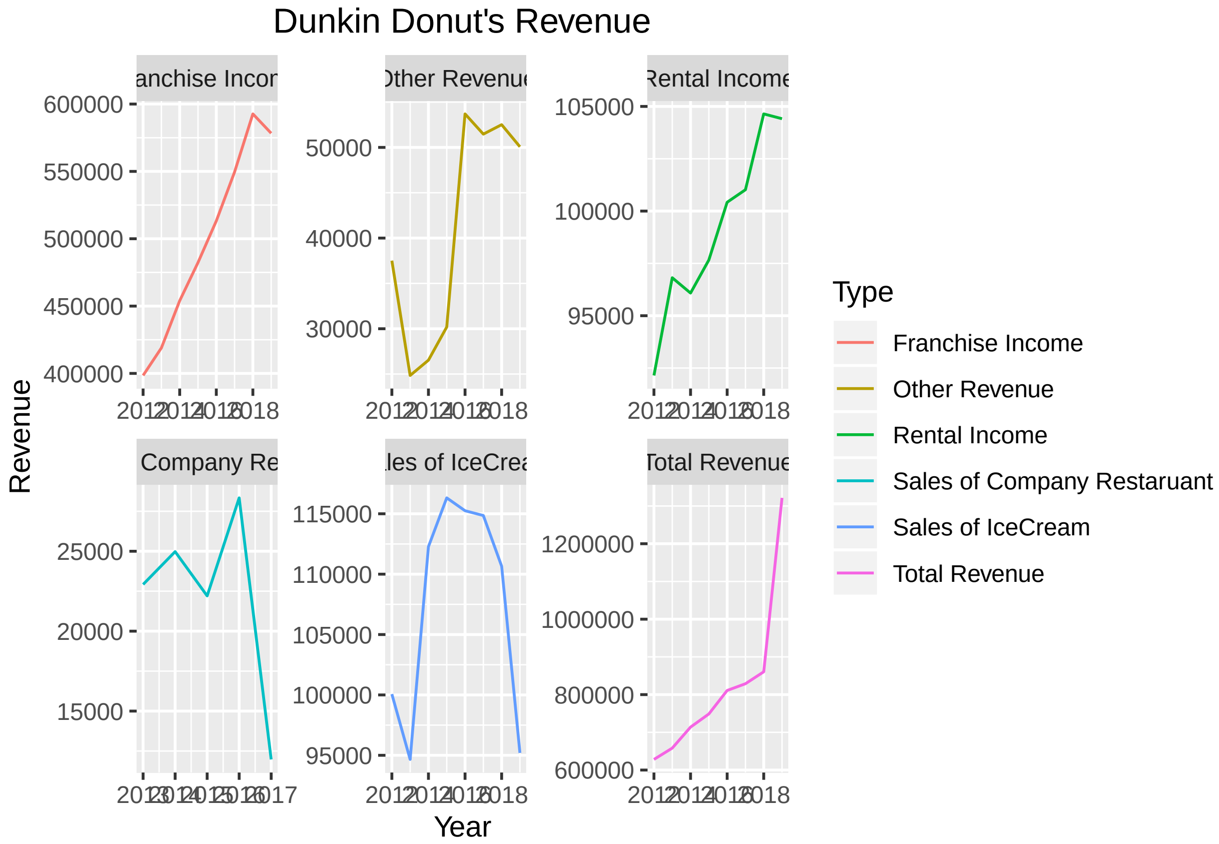
2016- Siren Retail Group,Teavana tea dump,tazo tea,evolution fresh(germany switzerland bad euro rate)

2017- Purchased all of CAP segments. Opened many stores in Americas,emas, Food, consumer packaged goods production like coffee, Keurig cups was hurting so contract to nestle. Improved Starbucks siren(Reserve store)

```
dunk_rev<- read_csv("data/dunkin.csv")

dunk_rev <- dunk_rev %>%
  select(year_of_publish, DD_US_F:BR_IN_F, Franchise_income, Rental_income:Total_reveneu)

dunk_rev <- dunk_rev %>%
  rename(
    `Franchise Income` = Franchise_income,
    `Rental Income` = Rental_income,
    `Sales of Company Restaruant` = `Sales_of_company restaurant`,
    `Sales of IceCream` = Sales_of_Icecream,
    `Total Revenue` = Total_reveneu,
    `Other Revenue` = Other_reveneue

  )
dunk_rev <- dunk_rev %>%
  select( `Franchise Income` , `Rental Income`, `Sales of Company Restaruant`, `Sales of IceCream`,`Tota
  gather(Type, count, `Franchise Income`, `Rental Income`, `Sales of Company Restaurant`, `Sales of IceC
  na.omit()


dunk_rev <- dunk_rev%>%
  group_by(Type, year_of_publish) %>%
  summarise(mean = mean(count))
```

Dunkin Donuts Revenue

```
dunk_rev %>%
  group_by(Type) %>%
  ggplot(mapping = aes(x = year_of_publish, y = mean)) +
  geom_line(mapping = aes(color = Type)) + ylab("Revenue") + xlab("Year")+ facet_wrap(~Type, scales = "
```



2014- Ice Cream cakes were launched and sold online with the order, and across all supermarkets. High ads were carried out. This leads to high sales for k cups and coffee also.

2016- In this year they change to 100%

2018-Other related revenue increased by the addition of the gift card program

But the main increase was due to recognizing advertising revenue(5% of each store sales) separately from this balance sheet. The growth actually declined Other the real growth is 828027