

PSY6422 - Project

ID: 220251464

2023-05-16



STARBUCKS & COSTA COFFEE

#enable github pages (a single clickable link which displays the markdown pages) #code rationale clearly and succinctly explained

Background

Data Origins

Answering questions of where it comes from, how it was collected, what the variables mean, etc show the first few rows of the raw data, if possible
data source fully acknowledged, including details on data collection which are relevant to interpretation full permanent citation

Project Organization

The /data folder contains data required for the project, and /images contains images required for the project, as well as visualization outputs.

A codebook describing all labels and abbreviations used in this project for data, variables, functions etc. within this project is located at /codebook.xlsx.

Research Questions

in plain English, a simple statement of what question(s) your visualisation will attempt to address

Data Preparation

Loading packages

The project utilized the renv package to retain package versions, safeguarding it from potential updates to packages in the future.

Package versions used in this project are listed within the file /renv.lock

```
#Load packages with renv
install.packages("renv")
library(renv)
renv::restore()

#Import packages
library(tidyverse)
library(gapminder)
library(ggplot2)
library(png)
library(RCurl)
library(grid)
library(dplyr)
library(scales)
library(showtext)
library(here)
library(readxl)
library(knitr)
library(kableExtra)
```

Cleaning the data

steps taken to clean the data, exclude outliers, create summary statistics, grouped variables, etc show the first few rows of the processed data, if possible showing the code which does this where relevant

```
#Specify image relative paths for logos and legend icons
Starbucks_image <- readPNG((here::here("images", "starbucks_logo.png")), native = TRUE)
Costa_image <- readPNG((here::here("images", "costa_logo.png")), native = TRUE)

starbucks_cup <- readPNG((here::here("images", "starbucks_cup.png")), native = TRUE)
costa_cup <- readPNG((here::here("images", "costa_cup.png")), native = TRUE)
```

```
#Read raw data
rawdata <- read_excel(here::here("data","DAAVDATA.xlsx"))
kable(rawdata, format = "markdown")
```

Type of coffee	Volume (mL) Starbucks	Volume (mL) Costa	Caffeine (mg) Starbucks	Caffeine (mg) Costa
Decaf Coffee	455	382	2	2
Single-shot Espresso	25	30	33	100
Frappuccino / Frappé	455	499	33	100
Cold Brew	455	345	50	210
Cortado	170	180	66	141
Double-shot Espresso	50	60	66	200
Latte	455	364	66	200
Iced Latte	455	473	66	200
Flat White	227	300	66	241
Filter / Brewed Coffee	455	382	136	256
Triple-shot Espresso	75	90	99	325
Americano	455	340	99	325
Mocha	455	332	66	325
Cappucino	455	362	66	325

```
#Create clean dataframe
df <- data.frame(
  Type.of.Coffee = c(rep(DAAVDATA$'Type of coffee', 2)),
  Brand = rep(c("Starbucks" , "Costa"), 14),
  Caffeine = c(DAAVDATA$`Caffeine (mg) Starbucks`, DAAVDATA$`Caffeine (mg) Costa`),
  Volume = c(DAAVDATA$`Volume (mL) Starbucks`, DAAVDATA$`Volume (mL) Costa`))
```

```
#Define the desired orders
order_list <- c(1, 1, 2, 2, 3, 3, 4, 4, 5, 5, 6, 6, 7, 7, 8, 8, 9, 9, 10, 10, 11, 11, 12,
order_list2 <- c(1, 15, 2, 16, 3, 17, 4, 18, 5, 19, 6, 20, 7, 21, 8, 22, 9, 23, 10, 24, 11
```

```
#Define a function to reorder a vector based on an order list
reorder_vec <- function(vec, order_list) {
  vec[order_list]}

```

```
#Reorder the columns using the reorder_vec function
df <- df %>%
  mutate(Type.of.Coffee = reorder_vec(Type.of.Coffee, order_list),
         Caffeine = reorder_vec(Caffeine, order_list2),
         Volume = reorder_vec(Volume, order_list2))
```

```
#Specify order of Coffee
df$Type.of.Coffee <- factor(df$Type.of.Coffee, levels = unique(df$Type.of.Coffee), ordered = TRUE)
```

```
#Print Clean data frame
head(df)
```

```
##
## 1 function (x, df1, df2, ncp, log = FALSE)
## 2 {
## 3     if (missing(ncp))
## 4         .Call(C_df, x, df1, df2, log)
## 5     else .Call(C_dnf, x, df1, df2, ncp, log)
## 6 }
```

Visualisation

graph or graphs documentation explaining any motivation (although good graph labelling is better than explanation in the accompanying text) code for producing them
alignment between visualisation question(s) and visualisation(s) such that the question is addressed and new insights provided (explains and justifies choice of chart (type,style, etc))

Colours

Colours of bars in the visualization correspond to iconic colours of the brands, for the Caffeine content, increasing glance value for those familiar with brands. This is important as those most likely to be interested in the data are those familiar with the brands. Additionally, for better clarity, the bars corresponding to the drink Volume, are coloured coffee-like colours, a light contrast with the Caffeine bars is ensured by slightly fading them and brands are differentiated by the two different hues of brown.

Custom Text

Using the showtext package, a personalized text was imported and incorporated into the visualization. The chosen font closely resembles the fonts employed by the two brands, thereby enhancing the professional appearance and overall attractiveness of the visualization.

```
font_add_google(name = "Source Sans Pro", family = "Source Sans Pro")
#Load new custom font for showtext package

showtext_auto()
#Automatically use showtext for plot
```

Design

```
#Plot graph and customize various plot aesthetics
ggp <- ggplot(df, aes(fill=Brand, y=Caffeine, x=Type.of.Coffee)) + #creating baseplot
  geom_bar(aes(y = Volume / 1, #adding volume barplot
    fill = ifelse(Brand == "Starbucks", "C", "D")), #specifying seperate bar colours
    position = "dodge", stat="identity", #specifying a grouped barplot
    width = 0.8, alpha=.5, #altering bar width and transparency
    show.legend = FALSE) + #hiding default legend
  geom_bar(aes(fill = ifelse(Brand == "Starbucks", "A", "B")), #adding caffeine barplot, specifying
    position = "dodge", stat = "identity", #making it a grouped barplot
    width = 0.5, #narrowing bar width to better see
    show.legend = FALSE) + #hiding default legend
  scale_y_continuous(name = "Caffeine (mg)", #naming left y-axis
    breaks = seq(0, 500, by = 50), #changing breaks on axis (range, f
```

```

limits = c(0, 670), expand = c(0,0),

sec.axis = sec_axis(~.*1, name="Volume (mL)",
  breaks = seq(0, 500, by = 50))) +
ggtitle("Costa VS Starbucks: Caffeine & Size") +
labs(x = "Type of Coffee*",
  subtitle = "Europe & UK, 2023",
  caption = "*Takeaway, Standard/Medium Size") +
scale_fill_manual(values = c("A" = "#1b703f", "B" = "#B91345",
  "C" = "#63330b", "D" = "#996633")) +
theme(aspect.ratio = 3/5,
  panel.grid.major = element_blank(),
  panel.grid.minor = element_blank(),
  panel.background = element_blank(),
  plot.margin = unit(c(1,1,2,1), "cm"),
  axis.line = element_line(colour = "black", linewidth = 1),
  axis.text.x = element_text(angle = 50, vjust = 0.5,
  hjust = 1, margin = margin(t = -30)),
  axis.title.x = element_text(margin = margin(t = 70),
  size = 12),
  axis.title.y = element_text(margin = margin(r = 15)),
  axis.title.y.right = element_text(margin = unit
  (c(0, 0, 0, 5), 'mm')),
  plot.title = element_text(margin = margin(b = 6),
  hjust = 0.5, size = 14, face = "bold"),
  plot.subtitle = element_text(margin = margin(b = 20),
  size = 11, hjust = 0.5),
  plot.caption = element_text(margin = margin(t = 10),
  color = "#6e6e6e", face = "italic",
  hjust = 0.5)) +
#
annotation_raster(Starbucks_image, xmin = 0.73, xmax = 1.39,
  ymin = 521, ymax = 573) +
annotation_raster(Costa_image, xmin = 0.491, xmax = 1.63,
  ymin = 578, ymax = 630) +
annotation_raster(starbucks_cup, xmin = 13.5, xmax = 14.5,
  ymin = 515, ymax = 570) +
annotation_raster(costa_cup, xmin = 13.5, xmax = 14.5,
  ymin = 575, ymax = 630) +
annotate("text", x = 1.7:1.7, y = c(550, 605),
  label = c("Starbucks", "Costa"), hjust = 0) +
annotate("text", x = 13.5:13.5, y = c(543, 603),
  label = c("Starbucks", "Costa"), hjust = 1) +
annotate("text", x = 0.8, y = 659, label = "Brand",
  hjust = 0, fontface = "bold") +
annotate("text", x = 14.2, y = 658, label = "Brand Volume",
  hjust = 1, fontface = "bold")

#making space for legends
#and removing space between graph
#adding secondary y-axis
#and specifying axis breaks
#creating the title
#labelling the x-axis
#creating the subtitle
#and caption for the x-axis title
#specifying colours for the types

#setting the proportions of the plot
#removing the plot grid

#removing the plot background
#altering the plot margin around the plot
#changing colour + thickness of axes
#adjusting x-axis break labels: distance from the axis
#alignment and distance from the axis
#adjusting x-axis label title: distance from the axis and text size
#adjusting left y-axis label: distance from the axis
#adjusting right y-axis label: distance from the axis
#(from the axis)
#altering plot title aesthetics: distance from the axis, alignment, size and making it bold
#changing subtitle aesthetics: distance from the axis, size and alignment
#altering the captions margin, color, italicizing
#and shifting the alignment

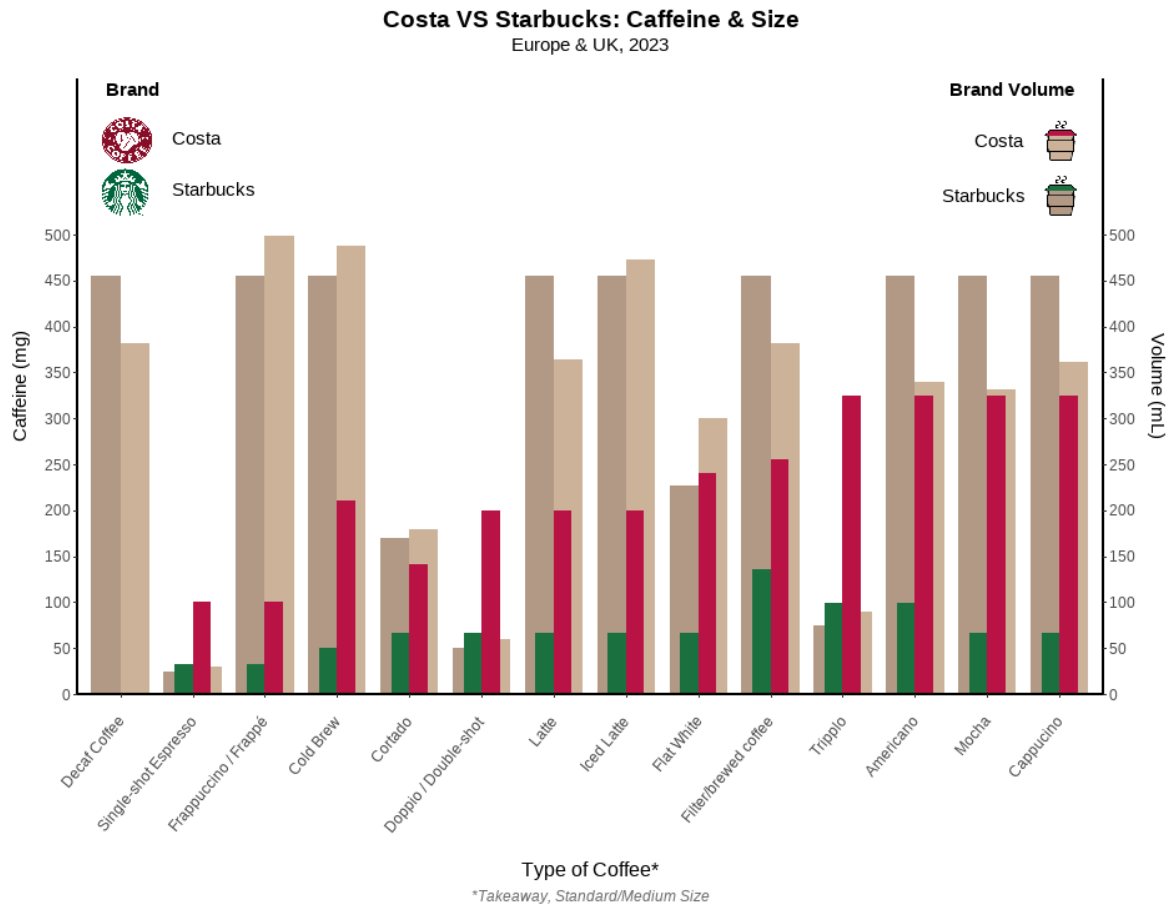
#creating my own legend:
#adding Starbucks logo + coordinates
#adding Costa logo + coordinates
#adding cup icon for Starbucks Volume
#plus coordinates
#adding cup icon for Costa Volume
#plus coordinates
#adding legend text on the left
#adding legend text on the right
#adding legend title on the left
#(including making it bold)
#adding legend title on the right
#(including making it bold)

```

Saving Visualisation

```
#Save plot to images folder
windows(width = 1000, height = 800) #Open windows graphics device
print(ggp)
dev.print(file = here("images", "StarbucksVSCosta_Graph.png"), device = png, width = 1000, height = 800)
```

Result



Interpretation & Future Direction

Notes on interpretation are provided Notes on possible extensions or follow-ups are provided

Summary

Brief thoughts on what you have learnt, what you might do next if you had more time / more data