

Midprojec

Abhinag Reddy

2024-03-10

```
getwd()

## [1] "C:/Users/abhin/OneDrive/Desktop/STAT/Midpro/midvisual"

setwd("C:/Users/abhin/OneDrive/Desktop/STAT/Midpro/midvisual")

# Read the CSV file into a data frame
data <- read.csv("data1.csv")

# Convert "Value" column to numeric
data$value <- as.numeric(gsub("\\$", "", data$value)) # Remove "$" sign and convert to numeric

# Sort the data by value in descending order
data_sorted <- data[order(-data$value),]

# Now, you can proceed with creating the visualizations

#DATA1_ReDESIGN2

library(ggplot2)

## Warning: package 'ggplot2' was built under R version 4.3.3

library(plotly)

## Warning: package 'plotly' was built under R version 4.3.3

##
## Attaching package: 'plotly'

## The following object is masked from 'package:ggplot2':
##
##   last_plot

## The following object is masked from 'package:stats':
##
##   filter

## The following object is masked from 'package:graphics':
##
##   layout
```

```
# Count the occurrences of each unique industry
industry_counts <- table(data$Industry)

# Print the counts of each unique industry
print(industry_counts)
```

```
##
##      Aerospace & Defence      Automobiles
##      1                      10
##      Banking                Commercial Services
##      11                     6
## Engineering & Construction      Food & Drinks
##      5                        7
##      Insurance                  Logistics
##      4                        2
##      Media                     Mining, Iron & Steel
##      7                        2
##      Oil & Gas                  Others
##      7                        6
##      Real Estate               Retail
##      1                        10
##      Tech                     Telecoms
##      10                      11
```

```
# Count the total number of unique industries
num_unique_industries <- length(industry_counts)

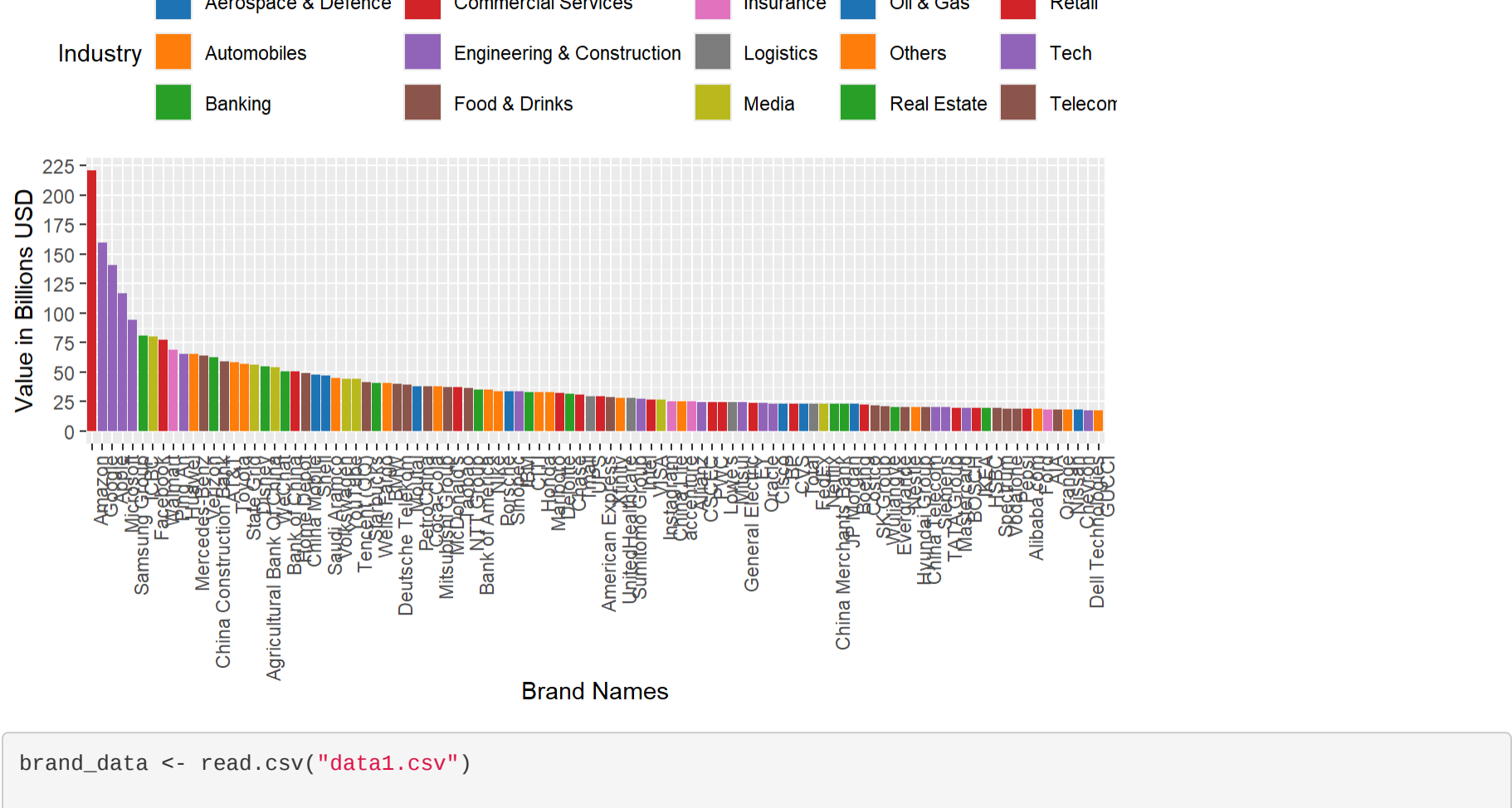
# Print the total number of unique industries
print(num_unique_industries)
```

```
## [1] 16
```

```
# Define updated colors for each industry
industry_colors <- c("Aerospace & Defence" = "#1f77b4", # blue
                    "Automobiles" = "#ff7f0e", # orange
                    "Banking" = "#2ca02c", # green
                    "Commercial Services" = "#d62728", # red
                    "Engineering & Construction" = "#9467bd", # purple
                    "Food & Drinks" = "#8c564b", # brown
                    "Insurance" = "#e377c2", # pink
                    "Logistics" = "#7f7f7f", # gray
                    "Media" = "#bcbd22", # yellow
                    "Mining, Iron & Steel" = "#17becf", # cyan
                    "Oil & Gas" = "#1f77b4", # blue
                    "Others" = "#ff7f0e", # orange
                    "Real Estate" = "#2ca02c", # green
                    "Retail" = "#d62728", # red
                    "Tech" = "#9467bd", # purple
                    "Telecoms" = "#8c564b") # brown

# Create a bar chart with updated color mapping and legend at the top
bar_chart <- ggplot(data_sorted, aes(x = reorder(Brand, -Value), y = Value, fill = Industry)) +
  geom_bar(stat = "identity") +
  scale_fill_manual(values = industry_colors) + # Set updated colors according to industry
  labs(title = "Brand Value in Billions USD",
       x = "Brand Names",
       y = "Value in Billions USD") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1),
        legend.position = "top") + # Move legend to the top
  guides(fill = guide_legend(title = "Industry")) + # Add legend for industry
  scale_y_continuous(breaks = seq(0, 300, by = 25)) # Set y-axis breaks to 0, 25, 50, 75, 100

# Display the bar chart
print(bar_chart)
```



```
brand_data <- read.csv("data1.csv")

library(dplyr)

##
## Attaching package: 'dplyr'

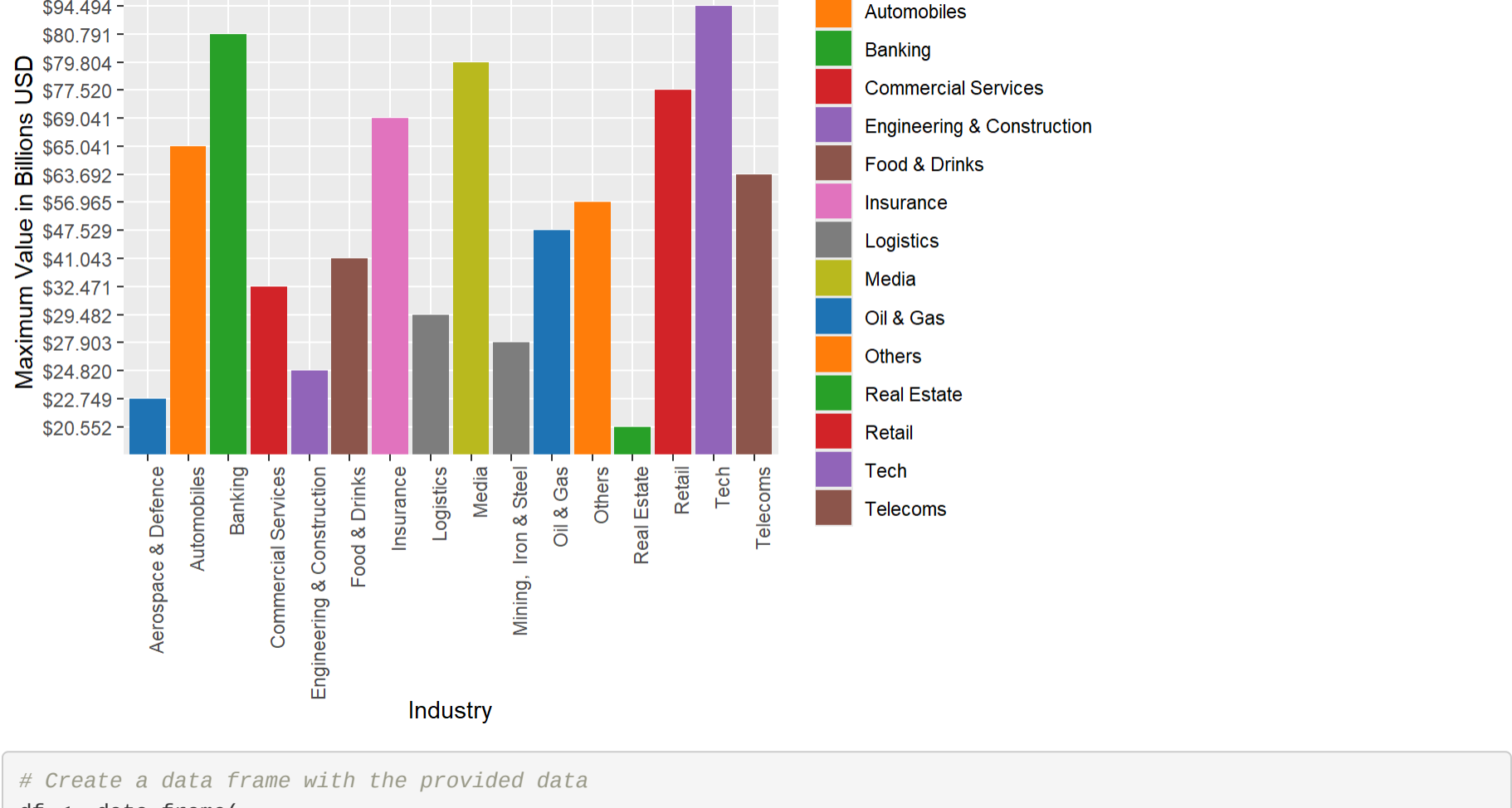
## The following objects are masked from 'package:stats':
##
##   filter, lag

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

# Calculate maximum value for each industry
max_values <- brand_data %>%
  group_by(Industry) %>%
  summarize(Max_Value = max(Value, na.rm = TRUE))

# Create the bar plot
max_value_bar_plot <- ggplot(max_values, aes(x = Industry, y = Max_Value, fill = Industry)) +
  geom_bar(stat = "identity") +
  scale_fill_manual(values = industry_colors) +
  labs(title = "Maximum Brand Value by Industry",
       x = "Industry",
       y = "Maximum Value in Billions USD") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))

# Display the bar plot
print(max_value_bar_plot)
```



```
# Create a data frame with the provided data
df <- data.frame(
  Rank = 1:20,
  Industry = c("Finance and insurance", "Other services, except public administration", "Professional, scientific, and technical services",
    "Agriculture, forestry, fishing and hunting", "Management of companies and enterprises", "Retail trade",
    "Health care and social assistance", "Information", "Transportation and warehousing", "Manufacturing",
    "Arts, entertainment, and recreation", "Public administration", "Accommodation and food services",
    "Utilities", "Wholesale trade", "Administrative and support and waste management services", "Educational services",
    "Real estate and rental and leasing", "Mining, quarrying, and oil and gas extraction", "Construction"),
  Median_earnings_Men = c(83660, 35778, 84749, 32021, 85219, 30592, 51233, 65475, 44984, 52026,
    28313, 66032, 20953, 76804, 51407, 31531, 47489, 46799, 73037, 42098),
  Median_earnings_Women = c(50456, 22083, 53152, 20689, 58718, 21415, 35916, 46552, 32345, 37694,
    21066, 50132, 16256, 60481, 40630, 25056, 39704, 40293, 65364, 39222),
  Women_earnings_percent_of_mens = c(60.3, 61.7, 62.7, 64.6, 68.9, 70.0, 70.1, 71.1, 71.9, 72.5,
    74.4, 75.9, 77.6, 78.7, 79.0, 79.5, 83.6, 86.1, 89.5, 93.2)
)

library(ggplot2)

# Melt the dataframe to long format for easier plotting
melted_df <- melt(df, id.vars = "Industry")

# Plot the median earnings for men and women across industries using bars
print(ggplot(melted_df, aes(x = value, y = Industry, fill = variable)) +

  # Use geom_bar to create bars for men's and women's median earnings
  geom_bar(stat = "identity", position = position_dodge(width = 0.9), width = 0.7) + # Adjust width of the bars

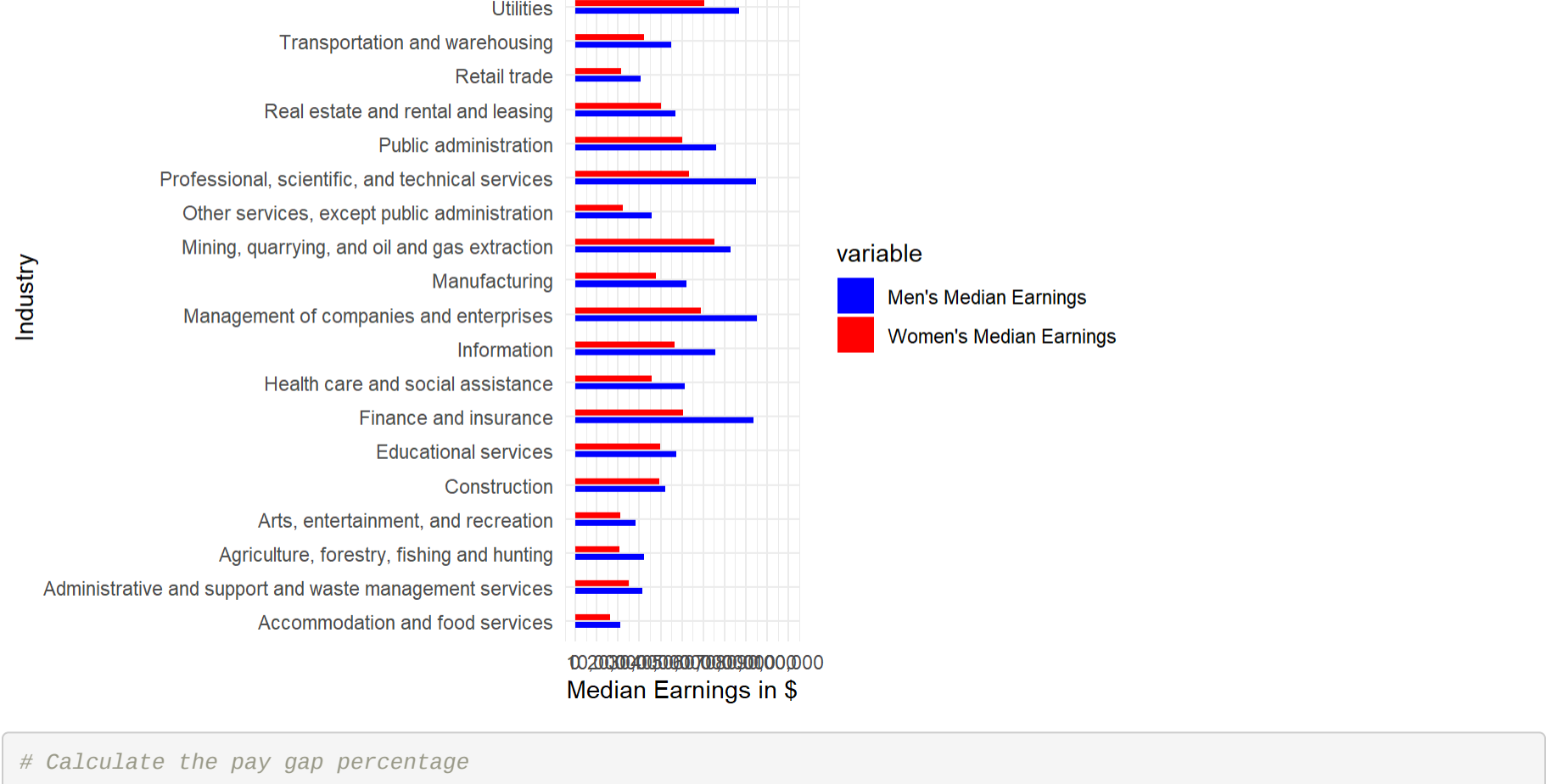
  # Set the y-axis limits from 0 to 100,000 with a range of 10,000
  scale_x_continuous(labels = scales::comma, limits = c(0, 100000), breaks = seq(0, 100000, by = 10000)) +
  # Adjust x-axis labels

  # Add title and labels
  labs(title = "Comparison of Median Earnings by Gender Across Industries",
       x = "Median Earnings in $",
       y = "Industry") +

  # Customize legend labels and colors
  scale_fill_manual(values = c("Median_earnings_Men" = "blue", "Median_earnings_Women" = "red"),
    labels = c("Men's Median Earnings", "Women's Median Earnings")) +

  # Rotate x-axis labels for better readability
  theme(axis.text.y = element_text(angle = 0, hjust = 1, vjust = 0.5)) +

  # Customize the theme
  theme_minimal())
```



```
# Calculate the pay gap percentage
df$Pay_Gap_Percentage <- 100 - df$Women_earnings_percent_of_mens

library(plotly)

# Specify the order of industries
industry_order <- df$Industry

# Create the line graph with red line and points using plotly
pay_gap_plot <- plot_ly(df, x = ~Pay_Gap_Percentage, y = ~Pay_Gap_Percentage, type = "scatter", mode = "lines+markers",
  line = list(color = "red", width = 2),
  marker = list(color = "red", size = 8)) %>%
  layout(title = "Gender Pay Gap Percentage Across Industries",
    xaxis = list(title = "Industry", categoryorder = "array", categoryarray = industry_order),
    yaxis = list(title = "Pay Gap Percentage"))

# Print the interactive line graph
pay_gap_plot
```

