Lab 1 R Basics

Gowlapalli Rohit

11/1/2024

Table of Contents

# Statistical Deception

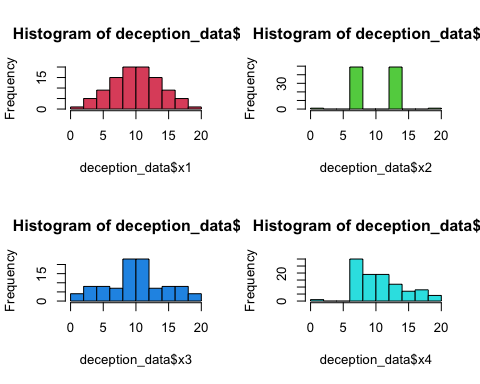
The following is the code for the question - Statistical Deception

library(readxl)  
library(ggplot2)  
library(tidyr)  
library(vioplot)  
  
deception\_data <- read\_excel("./2024\_Assignment1\_BRSM.xlsx", 1)  
  
  
print(deception\_data)

## # A tibble: 100 × 4  
## x1 x2 x3 x4  
## <dbl> <dbl> <dbl> <dbl>  
## 1 1 1 1 1   
## 2 2.02 7.10 1.26 7.40  
## 3 2.68 7.16 1.52 7.40  
## 4 3.18 7.19 1.78 7.40  
## 5 3.59 7.21 2.04 7.40  
## 6 3.93 7.23 2.31 7.40  
## 7 4.24 7.25 2.57 7.40  
## 8 4.51 7.26 2.83 7.40  
## 9 4.76 7.27 3.09 7.40  
## 10 4.99 7.28 3.35 7.40  
## # ℹ 90 more rows

The following are the histogram plots

par(mfrow =c(2,2))  
hist(deception\_data$x1, col=2)  
hist(deception\_data$x2, col=3)  
hist(deception\_data$x3, col=4)  
hist(deception\_data$x4, col=5)

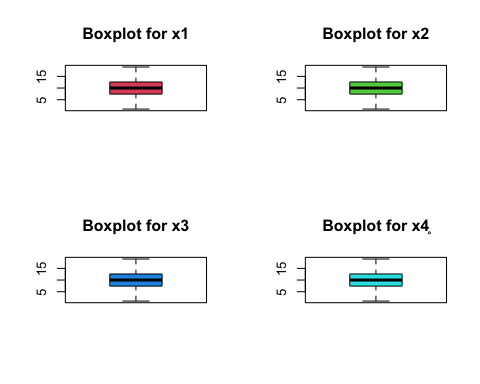


Upon examining the histogram plots, it becomes evident that the data exhibits a diverse distribution. The histograms convey that the mean of the dataset is roughly equivalent, but the distribution itself is notably heterogeneous.

Next, we turn our attention to the box plots.

The ensuing visualizations depict the box plots.

par(mfrow =c(2,2))  
boxplot(deception\_data$x1, col=2, main="Boxplot for x1")  
boxplot(deception\_data$x2, col=3, main="Boxplot for x2")  
boxplot(deception\_data$x3, col=4, main="Boxplot for x3")  
boxplot(deception\_data$x4, col=5, main="Boxplot for x4̥")

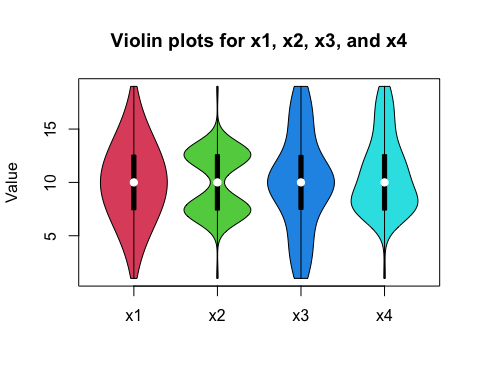


By scrutinizing the box plots, it is evident that no informative insights about the data can be gleaned, except for the observation that the box plots are identical. This uniformity may suggest an apparent similarity in data distribution, potentially leading to a misleading interpretation.

Subsequently, we will generate a violin plot, posited as the optimal visualization for this dataset.

The ensuing visualization represents the violin plot for the data.

vioplot(deception\_data, ylab="Value", col=2:5, main="Violin plots for x1, x2, x3, and x4")



The presented violin plot imparts substantial insights into the dataset. Notably, it reveals distinct data distribution characteristics across each column, with consistent median values. The dissimilarity among the columns lies in the manner in which the data is distributed. Specifically, the distribution of x1 conforms to a normal-like pattern, while the remaining columns exhibit divergent distribution patterns. It is noteworthy that the distribution patterns in the 1st and 3rd quartiles are analogous for the first three columns of the dataset.

Given the richness of information provided by the violin plot for this dataset, it stands out as one of the most effective visualizations.

Conversely, as previously discussed, the box plot emerges as the least suitable visualization for this dataset, offering limited and potentially misleading information.

# Personality and Motion

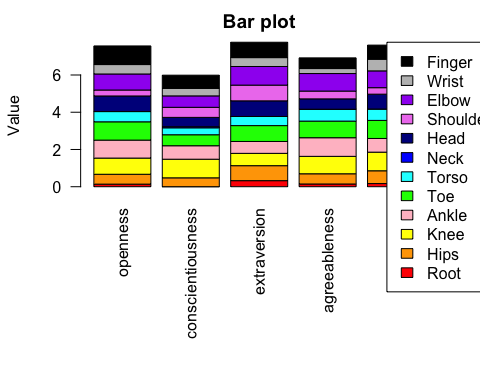
The following is the code for the question - Personality and Motion

library(fmsb)  
  
motion\_data <- read\_excel("./2024\_Assignment1\_BRSM.xlsx", 2)  
  
print(motion\_data)

## # A tibble: 12 × 6  
## Movements Openness Conscientiousness Extraversion Agreeableness Neuroticism  
## <chr> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 Root 0.139 0 0.325 0.147 0.169  
## 2 Hips 0.530 0.477 0.804 0.548 0.686  
## 3 Knee 0.869 1 0.662 0.936 1   
## 4 Ankle 0.965 0.723 0.639 1 0.735  
## 5 Toe 0.982 0.590 0.851 0.893 0.970  
## 6 Torso 0.551 0.373 0.490 0.638 0.612  
## 7 Neck 0 0.0576 0 0 0   
## 8 Head 0.838 0.503 0.840 0.556 0.798  
## 9 Shoulder 0.319 0.541 0.845 0.418 0.348  
## 10 Elbow 0.861 0.614 1 0.941 0.902  
## 11 Wrist 0.506 0.404 0.477 0.268 0.627  
## 12 Finger 1 0.708 0.826 0.574 0.757

Presented below is a stacked bar plot designed for the analysis of the significance of various joints in evaluating personality traits.

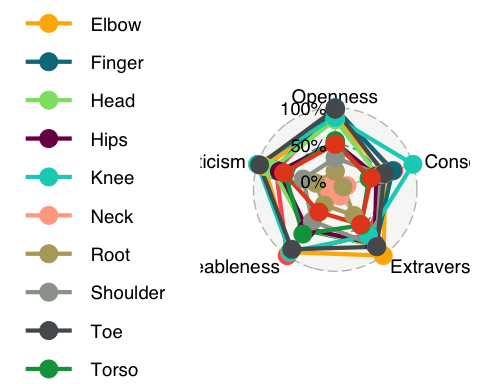
openness <- matrix(motion\_data$Openness)  
conscientiousness <- matrix(motion\_data$Conscientiousness)  
extraversion <- matrix(motion\_data$Extraversion)  
agreeableness <- matrix(motion\_data$Agreeableness)  
neuroticism <- matrix(motion\_data$Neuroticism)  
  
here <- data.frame(openness, conscientiousness, extraversion, agreeableness, neuroticism)  
  
par(mar = c(10, 4, 2, 2) + 0.2)  
  
barplot(as.matrix(here) , main = "Bar plot", ylab="Value", col=c("Red", "Orange", "Yellow", "Pink", "Green", "Cyan", "Blue", "Darkblue", "Violet", "Purple", "Grey", "Black"), xpd = TRUE, legend = c("Root", "Hips", "Knee", "Ankle", "Toe", "Torso", "Neck", "Head", "Shoulder", "Elbow", "Wrist", "Finger"), args.legend = list(x = "topright", inset = c(- 0.17, 0)), pch = 15, beside=FALSE, las=2)



The provided graph illustrates the proportionate importance of each joint in the evaluation of personality traits. However, the visibility of the ratios is suboptimal, rendering this graph unsuitable for effective visualization.

Subsequently, we will examine a radar plot for the same analysis.

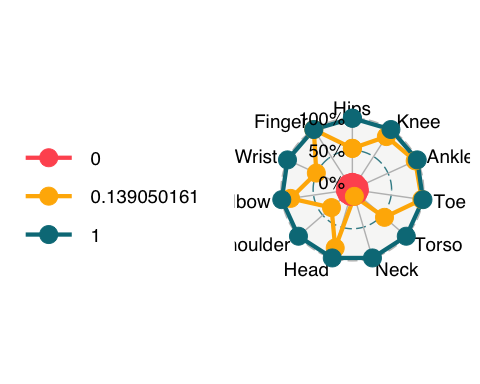
library(ggradar)  
ggradar(motion\_data)



The aforementioned chart is a radar plot that ostensibly represents the impact of Joint Importance values on personality traits. Nonetheless, the current presentation is somewhat unwieldy. Therefore, an attempt will be made to refine the plot by segmenting it according to each specific personality trait.

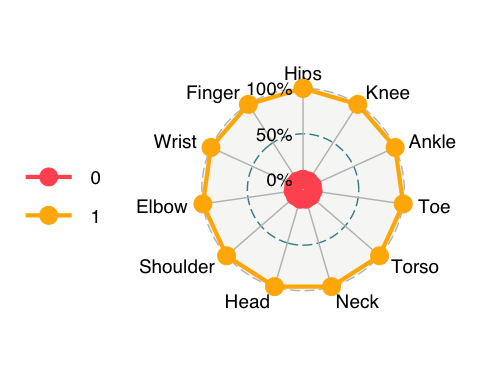
The ensuing plot pertains to the personality trait - Openness.

data1 <- data.frame(rbind(rep(1, 12), rep(0, 12), t(matrix(motion\_data$Openness))))  
colnames(data1) <- c("Root", "Hips", "Knee", "Ankle", "Toe", "Torso", "Neck", "Head", "Shoulder", "Elbow", "Wrist", "Finger")  
  
ggradar(data1)



The ensuing plot pertains to the personality trait - Conscientiousness

data2 <- data.frame(rbind(rep(1, 12), rep(0, 12), t(matrix(motion\_data$Conscientiousness))))  
colnames(data2) <- c("Root", "Hips", "Knee", "Ankle", "Toe", "Torso", "Neck", "Head", "Shoulder", "Elbow", "Wrist", "Finger")  
  
ggradar(data2)

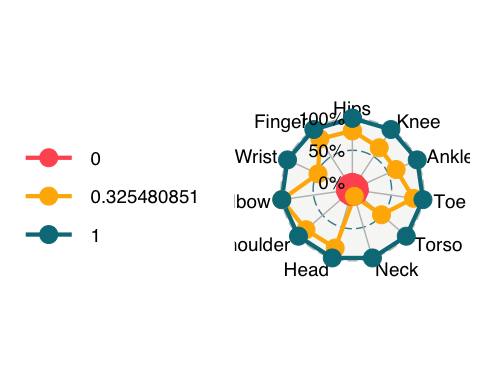


The ensuing plot pertains to the personality trait - Extraversion

print(motion\_data$Extraversion)

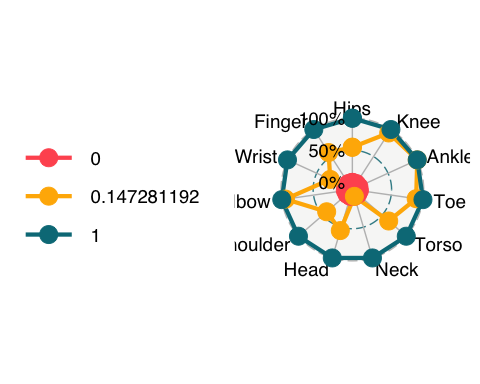
## [1] 0.3254809 0.8042766 0.6621905 0.6387358 0.8512893 0.4899196 0.0000000  
## [8] 0.8395976 0.8449116 1.0000000 0.4770724 0.8255081

data3 <- data.frame(rbind(rep(1, 12), rep(0, 12), t(matrix(motion\_data$Extraversion))))  
colnames(data3) <- c("Root", "Hips", "Knee", "Ankle", "Toe", "Torso", "Neck", "Head", "Shoulder", "Elbow", "Wrist", "Finger")  
  
ggradar(data3)



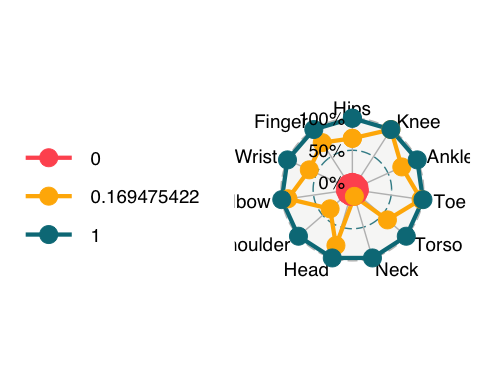
The ensuing plot pertains to the personality trait - Agreeableness

data4 <- data.frame(rbind(rep(1, 12), rep(0, 12), t(matrix(motion\_data$Agreeableness))))  
colnames(data4) <- c("Root", "Hips", "Knee", "Ankle", "Toe", "Torso", "Neck", "Head", "Shoulder", "Elbow", "Wrist", "Finger")  
  
ggradar(data4)



The ensuing plot pertains to the personality trait - Neuroticism

data5 <- data.frame(rbind(rep(1, 12), rep(0, 12), t(matrix(motion\_data$Neuroticism))))  
colnames(data5) <- c("Root", "Hips", "Knee", "Ankle", "Toe", "Torso", "Neck", "Head", "Shoulder", "Elbow", "Wrist", "Finger")  
  
ggradar(data5)



The radar plots provided above offer a lucid depiction of the proportional significance of each joint in predicting personality traits. Consequently, this visualization technique proves to be more effective for addressing this inquiry.

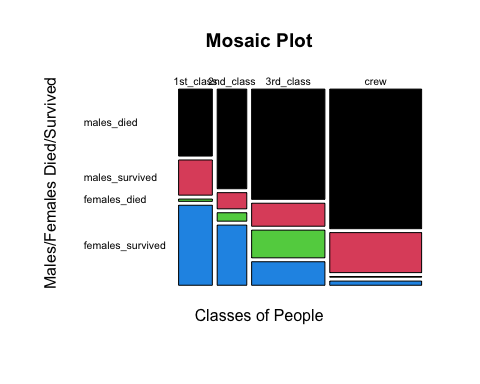
# Data Plotting Adventure

The following is the plot for the subtask 3.1 (The Last of Us)

data = matrix(c(118, 62, 4, 141, 154, 25, 13, 93, 422, 88, 106, 90, 670, 192, 3, 20), ncol = 4, byrow = TRUE)  
  
rownames(data) <- c('1st\_class', '2nd\_class', '3rd\_class', 'crew')  
colnames(data) <- c('males\_died', 'males\_survived', 'females\_died', 'females\_survived')  
  
print(data)

## males\_died males\_survived females\_died females\_survived  
## 1st\_class 118 62 4 141  
## 2nd\_class 154 25 13 93  
## 3rd\_class 422 88 106 90  
## crew 670 192 3 20

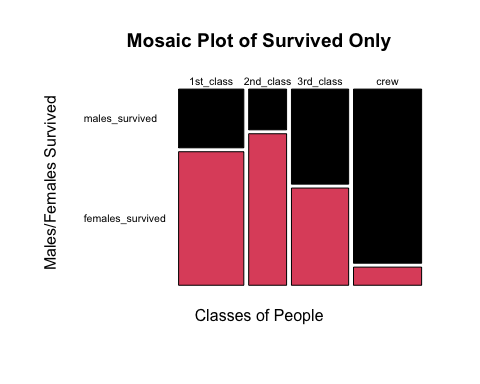
mosaicplot(data, color=1:4, las=1, legend=TRUE, xlab="Classes of People", ylab="Males/Females Died/Survived", main = "Mosaic Plot")



data = matrix(c(62, 141, 25, 93, 88, 90,192, 20), ncol = 2, byrow = TRUE)  
  
rownames(data) <- c('1st\_class', '2nd\_class', '3rd\_class', 'crew')  
colnames(data) <- c('males\_survived', 'females\_survived')  
  
print(data)

## males\_survived females\_survived  
## 1st\_class 62 141  
## 2nd\_class 25 93  
## 3rd\_class 88 90  
## crew 192 20

mosaicplot(data, color=1:4, las=1, legend=TRUE, xlab="Classes of People", ylab="Males/Females Survived", main = "Mosaic Plot of Survived Only")



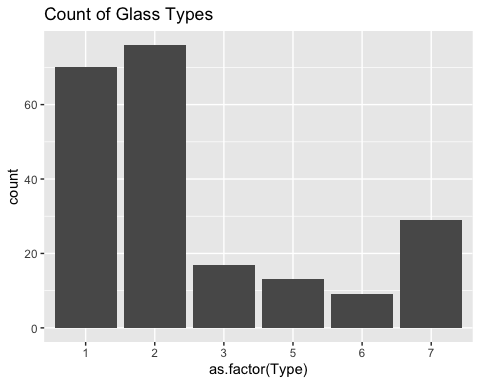
The following is the subtask 3.2: Glass Glimpse

library(tidyverse)  
library(readxl)  
library(corrplot)  
glass\_data <- read\_excel("./2024\_Assignment1\_BRSM.xlsx", sheet = 3)  
  
# Display the first few rows, information, and summary statistics of the dataset  
summary(glass\_data)

## RI Na Mg Al   
## Min. :1.511 Min. :10.73 Min. :0.000 Min. :0.290   
## 1st Qu.:1.517 1st Qu.:12.91 1st Qu.:2.115 1st Qu.:1.190   
## Median :1.518 Median :13.30 Median :3.480 Median :1.360   
## Mean :1.518 Mean :13.41 Mean :2.685 Mean :1.445   
## 3rd Qu.:1.519 3rd Qu.:13.82 3rd Qu.:3.600 3rd Qu.:1.630   
## Max. :1.534 Max. :17.38 Max. :4.490 Max. :3.500   
## Si K Ca Ba   
## Min. :69.81 Min. :0.0000 Min. : 5.430 Min. :0.000   
## 1st Qu.:72.28 1st Qu.:0.1225 1st Qu.: 8.240 1st Qu.:0.000   
## Median :72.79 Median :0.5550 Median : 8.600 Median :0.000   
## Mean :72.65 Mean :0.4971 Mean : 8.957 Mean :0.175   
## 3rd Qu.:73.09 3rd Qu.:0.6100 3rd Qu.: 9.172 3rd Qu.:0.000   
## Max. :75.41 Max. :6.2100 Max. :16.190 Max. :3.150   
## Fe Type   
## Min. :0.00000 Min. :1.00   
## 1st Qu.:0.00000 1st Qu.:1.00   
## Median :0.00000 Median :2.00   
## Mean :0.05701 Mean :2.78   
## 3rd Qu.:0.10000 3rd Qu.:3.00   
## Max. :0.51000 Max. :7.00

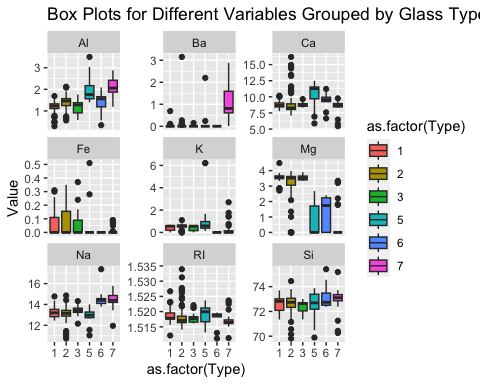
Having taken an initial look at the dataset, it becomes evident that there are no missing values, and all attributes are numeric. Even the categorical variable “Type” is encoded as integers, with no inherent order among the classes.

# Count plot of glass types  
ggplot(glass\_data, aes(x = as.factor(Type))) +  
 geom\_bar() +  
 ggtitle("Count of Glass Types")



Type 1 and Type 2 are the most prevalent categories in the dataset, characterized as building windows subjected to a float processing method (Type 1) and building windows subjected to a non-float processing method (Type 2).

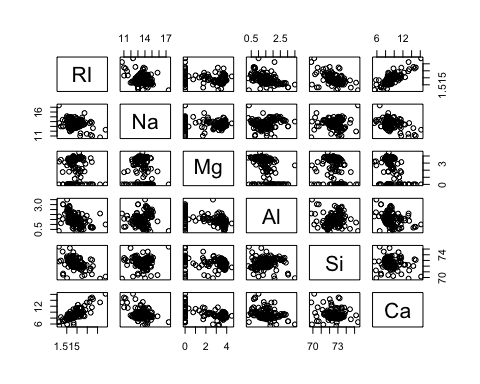
# Box plots for different variables grouped by glass type  
glass\_data %>%  
 gather(key = "Variable", value = "Value", -Type) %>%  
 ggplot(aes(x = as.factor(Type), y = Value, fill = as.factor(Type))) +  
 geom\_boxplot() +  
 facet\_wrap(~Variable, scales = "free\_y", ncol = 3) +  
 ggtitle("Box Plots for Different Variables Grouped by Glass Type")



The box plot indicates that the average refractive index is comparable across all types, although Type 5 exhibits a broader range and a slightly higher mean. Sodium content is elevated in Type 6 and 7, while magnesium content is notably high in Type 1, 2, and 3. Aluminum is more abundant in Type 5 and 7. Silica, despite having the highest concentration among all minerals, does not offer significant differentiation across types, displaying a similar range for all. Potassium does not provide substantial insights, except for its higher presence in Type 5 (containers), where it is utilized in toughened glass production, notably in items like Pyrex. Calcium predominates in Type 5, and Barium is most abundant in Type 7. Although iron is present in extremely low concentrations overall, Type 1, 2, and 3 exhibit higher values compared to the other types, attributed to the deliberate addition of iron in the production of colored glasses.

Having gained insights into which features contribute more to the differentiation of one type from another, we now aim to explore the relationships between these features through a pairplot, excluding potassium (K), barium (Ba), and iron (Fe) due to their sparse concentrations.

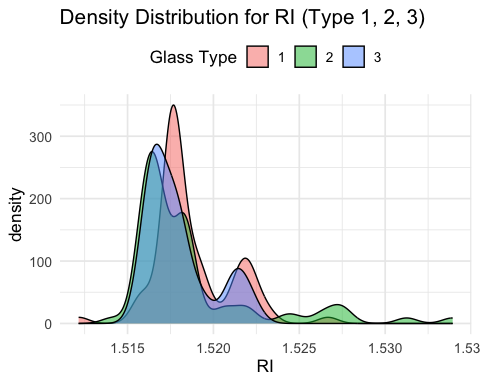
# Pair plot for selected variables  
glass\_data\_selected <- select(glass\_data, RI, Na, Mg, Al, Si, Ca)  
pairs(glass\_data\_selected)



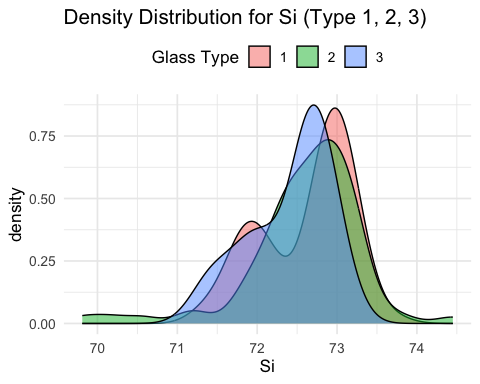
Across most dimensions, the data points exhibit scattered patterns. Nevertheless, a discernible strong relationship emerges between calcium and refractive index, while silica and refractive index appear to demonstrate an inverse correlation.

Types 1, 2, and 3 exhibit strikingly similar properties. However, a distinguishing factor is their flatness or non-flatness, where the float process is employed to flatten the glass into sheets. Let’s explore the density plot comparing flat and non-flat glass across various properties.

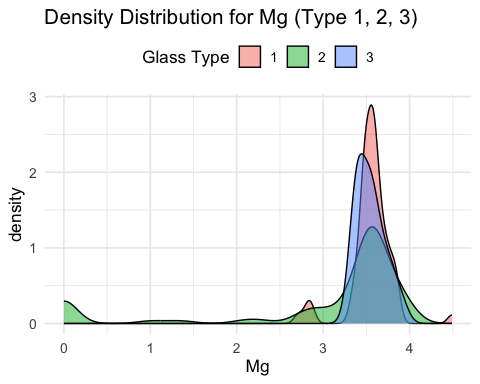
# Set the style and font scale  
theme\_set(theme\_minimal(base\_size = 13))  
  
# Filter data for Type 1, 2, and 3  
filtered\_glass\_df <- glass\_data %>%  
 filter(Type %in% c(1, 2, 3))  
  
# Create a data frame with selected variables for flat vs. non-flat glass (RI)  
glass\_flat\_df\_ri <- filtered\_glass\_df %>%  
 select(Type, RI)  
  
# Plot density distribution separately for RI in a separate row  
ggplot(glass\_flat\_df\_ri, aes(x = RI, fill = as.factor(Type))) +  
 geom\_density(alpha = 0.5) +  
 ggtitle("Density Distribution for RI (Type 1, 2, 3)") +  
 labs(fill = "Glass Type") +  
 theme\_minimal(base\_size = 13) +  
 theme(legend.position = "top")



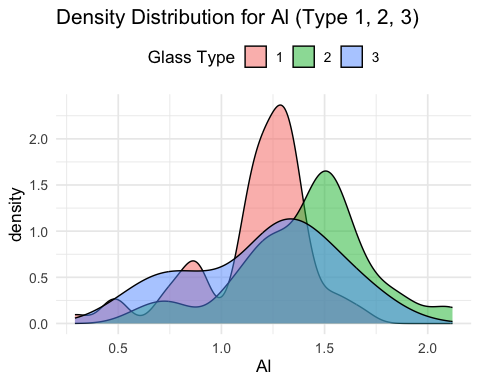
# Set the style and font scale  
theme\_set(theme\_minimal(base\_size = 13))  
  
# Filter data for Type 1, 2, and 3  
filtered\_glass\_df <- glass\_data %>%  
 filter(Type %in% c(1, 2, 3))  
  
# Create a data frame with selected variables for flat vs. non-flat glass (Si)  
glass\_flat\_df\_si <- filtered\_glass\_df %>%  
 select(Type, Si)  
  
# Plot density distribution separately for Si in a separate row  
ggplot(glass\_flat\_df\_si, aes(x = Si, fill = as.factor(Type))) +  
 geom\_density(alpha = 0.5) +  
 ggtitle("Density Distribution for Si (Type 1, 2, 3)") +  
 labs(fill = "Glass Type") +  
 theme\_minimal(base\_size = 13) +  
 theme(legend.position = "top")



# Set the style and font scale  
theme\_set(theme\_minimal(base\_size = 13))  
  
# Filter data for Type 1, 2, and 3  
filtered\_glass\_df <- glass\_data %>%  
 filter(Type %in% c(1, 2, 3))  
  
# Create a data frame with selected variables for flat vs. non-flat glass (Mg)  
glass\_flat\_df\_mg <- filtered\_glass\_df %>%  
 select(Type, Mg)  
  
# Plot density distribution separately for Mg in a separate row  
ggplot(glass\_flat\_df\_mg, aes(x = Mg, fill = as.factor(Type))) +  
 geom\_density(alpha = 0.5) +  
 ggtitle("Density Distribution for Mg (Type 1, 2, 3)") +  
 labs(fill = "Glass Type") +  
 theme\_minimal(base\_size = 13) +  
 theme(legend.position = "top")

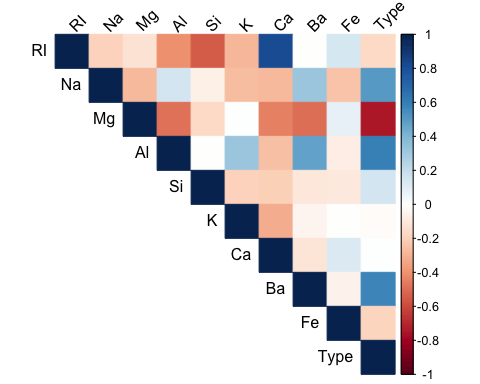


# Set the style and font scale  
theme\_set(theme\_minimal(base\_size = 13))  
  
# Filter data for Type 1, 2, and 3  
filtered\_glass\_df <- glass\_data %>%  
 filter(Type %in% c(1, 2, 3))  
  
# Create a data frame with selected variables for flat vs. non-flat glass (Al)  
glass\_flat\_df\_al <- filtered\_glass\_df %>%  
 select(Type, Al)  
  
# Plot density distribution separately for Al in a separate row  
ggplot(glass\_flat\_df\_al, aes(x = Al, fill = as.factor(Type))) +  
 geom\_density(alpha = 0.5) +  
 ggtitle("Density Distribution for Al (Type 1, 2, 3)") +  
 labs(fill = "Glass Type") +  
 theme\_minimal(base\_size = 13) +  
 theme(legend.position = "top")



Primarily, the flat glass types (1 and 3) exhibit overlap across all properties, except for aluminum. Type 3 displays a broader range of aluminum values compared to Type 1. In contrast, the non-float glass Type 2 features numerous extreme points, contributing to a notably extensive range.

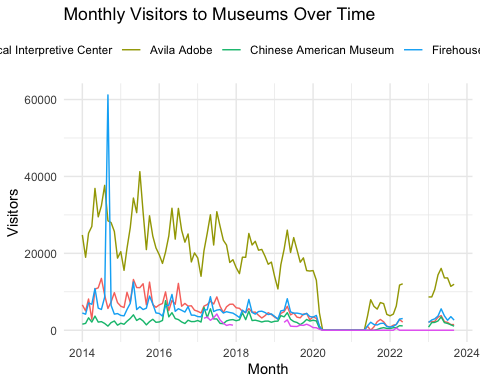
cor\_matrix <- cor(glass\_data)  
corrplot(cor\_matrix, method = "color", type = "upper", tl.col = "black", tl.srt = 45)



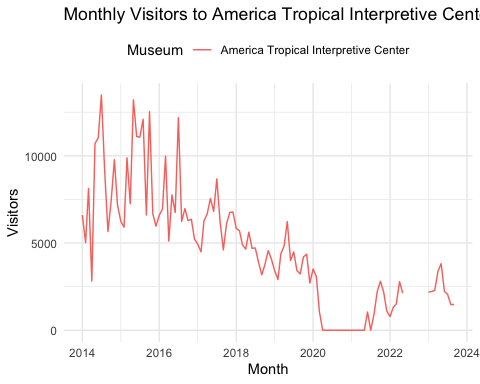
As evident in the pairplot, there exists a strong positive correlation between refractive index (RI) and calcium (Ca), while silica (Si) and RI exhibit a negative correlation. However, given that this analysis pertains to classification rather than regression, it is not advisable to straightforwardly eliminate collinear variables from the model.

The following is the code for the subtask 3.3: Night at the Museum

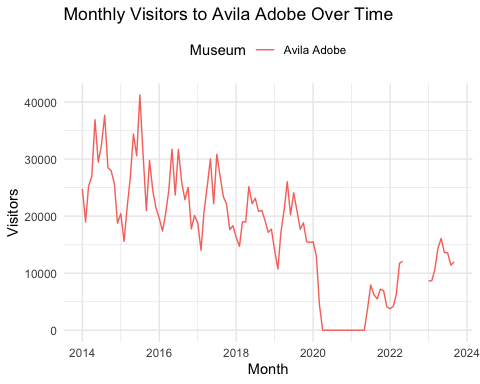
# Load necessary libraries  
library(readxl)  
library(ggplot2)  
library(dplyr)  
library(tidyr)  
  
# Set the file path  
museum\_filepath <- "./2024\_Assignment1\_BRSM.xlsx"  
  
# Read the Excel file into a variable museum\_data  
museum\_data <- read\_excel(museum\_filepath,sheet=4)  
  
# Convert 'Month' column to Date format  
museum\_data$Month <- as.Date(paste("01", museum\_data$Month, sep = " "), format = "%d %b %Y")  
  
# Melt the data for easier plotting  
melted\_data <- museum\_data %>%  
 pivot\_longer(cols = -Month, names\_to = "Museum", values\_to = "Visitors")  
  
# Line chart showing the number of visitors to each museum over time  
ggplot(melted\_data, aes(x = Month, y = Visitors, color = Museum)) +  
 geom\_line() +  
 labs(title = "Monthly Visitors to Museums Over Time") +  
 theme\_minimal() +  
 theme(legend.position = "top")



plot\_seasonal\_line\_chart <- function(museum\_name) {  
 ggplot(subset(melted\_data, Museum == museum\_name),   
 aes(x = Month, y = Visitors, color = Museum)) +  
 geom\_line() +  
 labs(title = paste("Monthly Visitors to", museum\_name, "Over Time"),  
 x = "Month",  
 y = "Visitors") +  
 theme\_minimal() +  
 theme(legend.position = "top")  
}  
  
plot\_seasonal\_line\_chart("America Tropical Interpretive Center")



plot\_seasonal\_line\_chart("Avila Adobe")



plot\_seasonal\_line\_chart("Chinese American Museum")



plot\_seasonal\_line\_chart("Chinese American Museum")



# FAST AND FURIOUS: HEATMAP

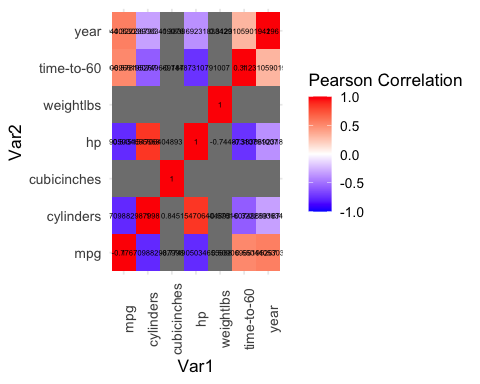
The following is the task 4 (Fast and Furious: Heatmaps)

Given below are the steps to perform before plotting the heatmaps

library(reshape2)  
library(dplyr)  
  
fast\_furious\_data <- read\_excel("./2024\_Assignment1\_BRSM.xlsx", 5)  
fast\_furious\_data\_here <- select\_if(fast\_furious\_data, is.numeric)  
  
mat1 <- melt(cor(fast\_furious\_data\_here, method="pearson"))  
mat2 <- melt(cor(fast\_furious\_data\_here, method="spearman"))  
mat3 <- melt(cor(fast\_furious\_data\_here, method="kendall"))

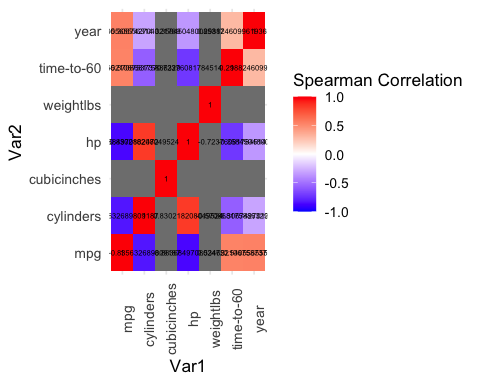
Now, the following is the heat map when the correlation matrix is chosen to be Pearson’s.

ggplot(data=mat1, aes(x=Var1, y=Var2, fill=value)) +  
 geom\_tile() +  
 geom\_text(aes(label = value), size = 2) +  
 scale\_fill\_gradient2(low = "blue", high="red", limit = c(-1, 1), name="Pearson Correlation") +   
 theme(axis.text.x = element\_text(angle = 90))



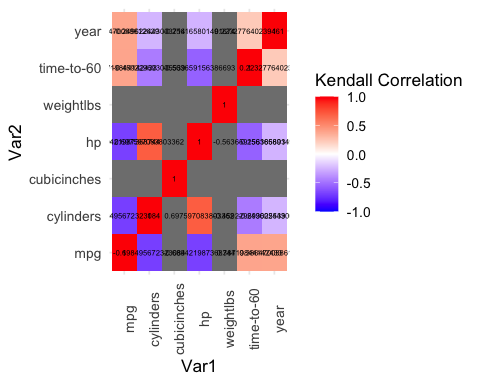
Now, the following is the heat map when the correlation matrix is chosen to be Spearman’s.

ggplot(data=mat2, aes(x=Var1, y=Var2, fill=value)) +  
 geom\_tile() +  
 geom\_text(aes(label = value), size = 2) +  
 scale\_fill\_gradient2(low = "blue", high="red", limit = c(-1, 1), name="Spearman Correlation") +   
 theme(axis.text.x = element\_text(angle = 90))



Now, the following is the heat map when the correlation matrix is chosen to be Kendall’s.

ggplot(data=mat3, aes(x=Var1, y=Var2, fill=value)) +  
 geom\_tile() +  
 geom\_text(aes(label = value), size = 2) +  
 scale\_fill\_gradient2(low = "blue", high="red", limit = c(-1, 1), name="Kendall Correlation") +   
 theme(axis.text.x = element\_text(angle = 90))



By examining the heat maps presented earlier, it becomes evident that the Spearman correlation matrix exhibits cells with elevated correlation values. Consequently, among the three matrices, the Spearman Matrix is selected.

Upon analyzing the heat map, we observe a significant correlation between variables like curb weight and engine size. Utilizing the heat map, we can identify additional highly correlated variables, enhancing our understanding of the dataset.