

# 2021101113\_Visualisation\_Assignment

Gowlapalli Rohit

23/1/2024

## Contents

<b>Statistical Deception</b>	<b>1</b>
<b>Personality and Motion</b>	<b>5</b>
<b>Data Plotting Adventure</b>	<b>12</b>
Subtask 3.1 (The Last of Us) . . . . .	12
Subtask 3.2: Glass Glimpse . . . . .	14
Subtask 3.3: Night at the Museum . . . . .	23
<b>FAST AND FURIOUS: HEATMAP</b>	<b>28</b>

## Statistical Deception

The following is the code for the question - Statistical Deception

```
library(readxl)
library(ggplot2)
library(tidyr)
library(vioplplot)

deception_data <- read_excel("./2024_Assignment1_BRSM.xlsx", 1)

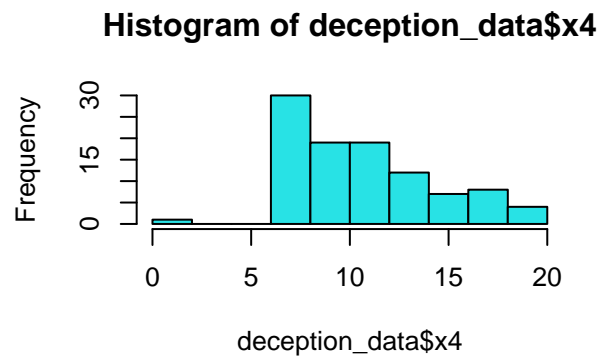
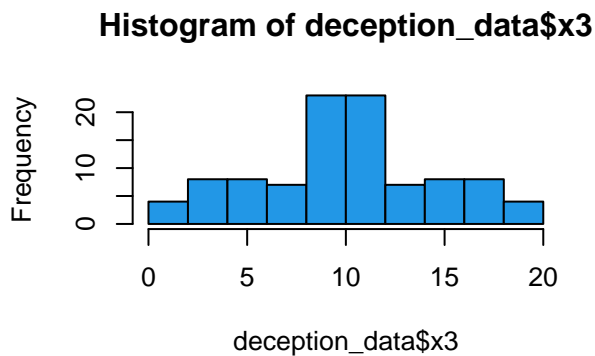
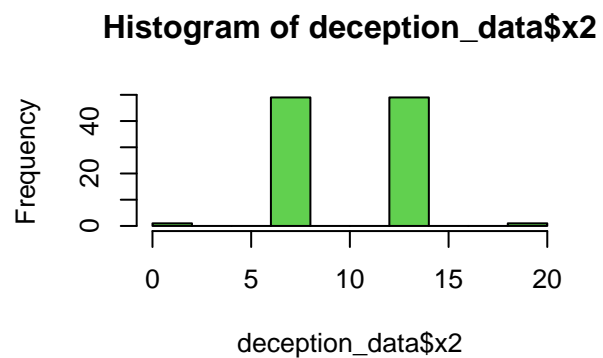
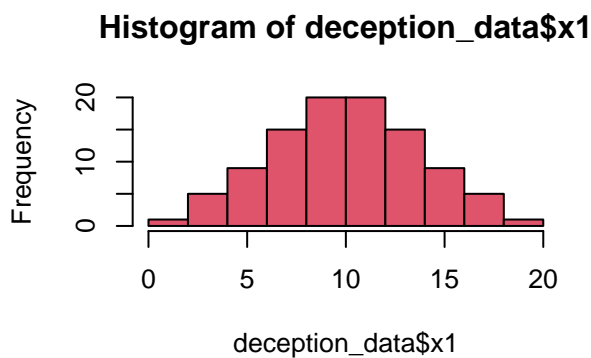
print(deception_data)
```

```
## # A tibble: 100 x 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)
```

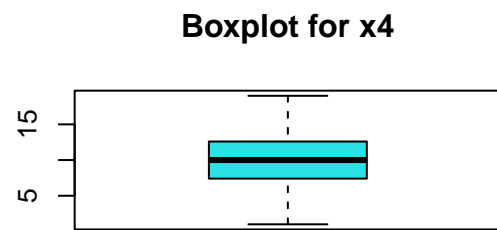
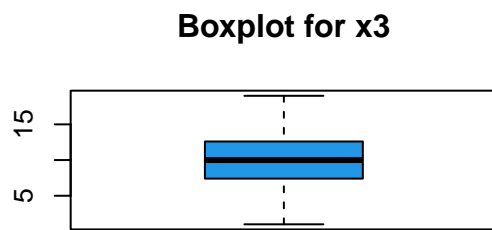
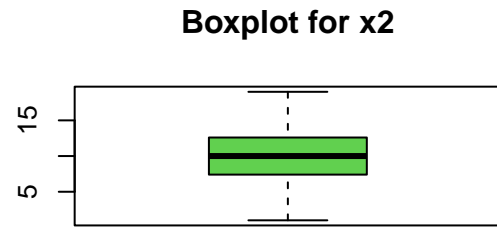
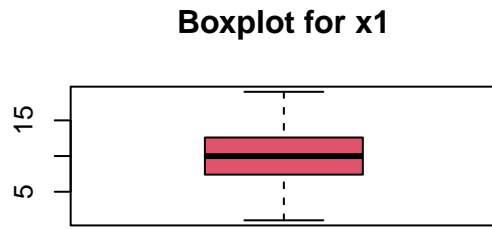


Upon examining the histogram plots, it becomes evident that the data exhibits a diverse distribution. The histograms convey that the mean of the dataset is roughly equivalent, but the distribution itself is notably heterogeneous.

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 x4")
```



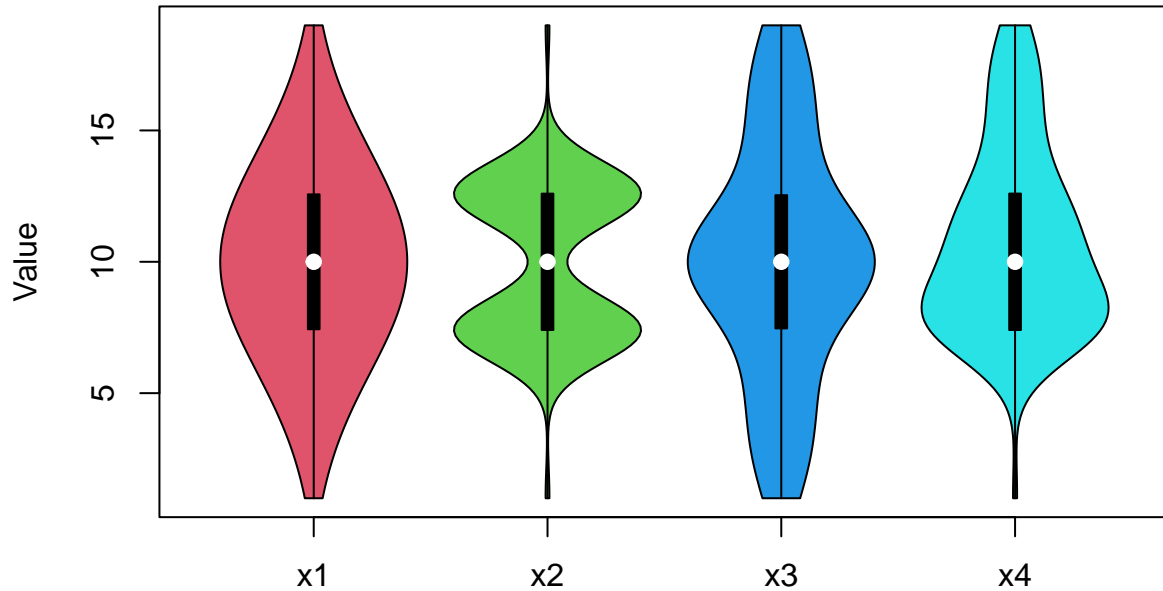
The Boxplot is misleading because it has limitations in representing modes, handling datasets with multiple peaks, revealing data gaps in the interquartile range, and accurately portraying the detailed shape of distributions. This was evident in our visualization, where it failed to highlight differences among the four datasets and did not provide exact values. 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")
```

## 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.

Violin plots prove to be the most effective visualization method for the provided data, offering a powerful tool to distinguish between distributions in x1, x2, x3, and x4. Each category—x1 (Normal distribution), x2 (Bimodal distribution), x3 (Fairly Normally Distributed with more spread), and x4 (Roughly right-skewed data)—is distinctly visualized through the violin plot, providing insights into the shape, spread, and summary statistics of the data.

A violin plot combines features of a box plot and a kernel density plot, offering a nuanced understanding of the density of each variable. It excels in highlighting outliers, particularly evident in x2 and x4, which may not be easily discernible in conventional box plots.

Key observations include:

- x1 (Normal Distribution): Displays a symmetrical and bell-shaped curve, characteristic of a normal distribution.
- x2 (Bimodal Distribution): Shows distinct peaks, indicating the presence of two modes, more apparent than in a box plot.
- x3 (Fairly Normally Distributed with More Spread): Demonstrates a broader distribution with a central peak, capturing the increased spread of the data.
- x4 (Roughly Right-skewed Data): Illustrates a skewed right distribution, emphasizing the asymmetry and tail on the right side.

Violin plots offer a compelling means to visually compare multiple distributions simultaneously, providing a comprehensive view of data spreadness and distribution characteristics. Their ability to combine aspects of box plots and histograms, using a smoothed probability density function (PDF), makes them a versatile tool for exploratory data analysis.

For enhanced visualization, raincloud plots—a hybrid consisting of a halved violin plot, a box plot, and raw data as a scatter plot—can be considered. These plots further enhance the visual representation of data distributions, offering a holistic view of the underlying patterns and variations in 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 x 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
```

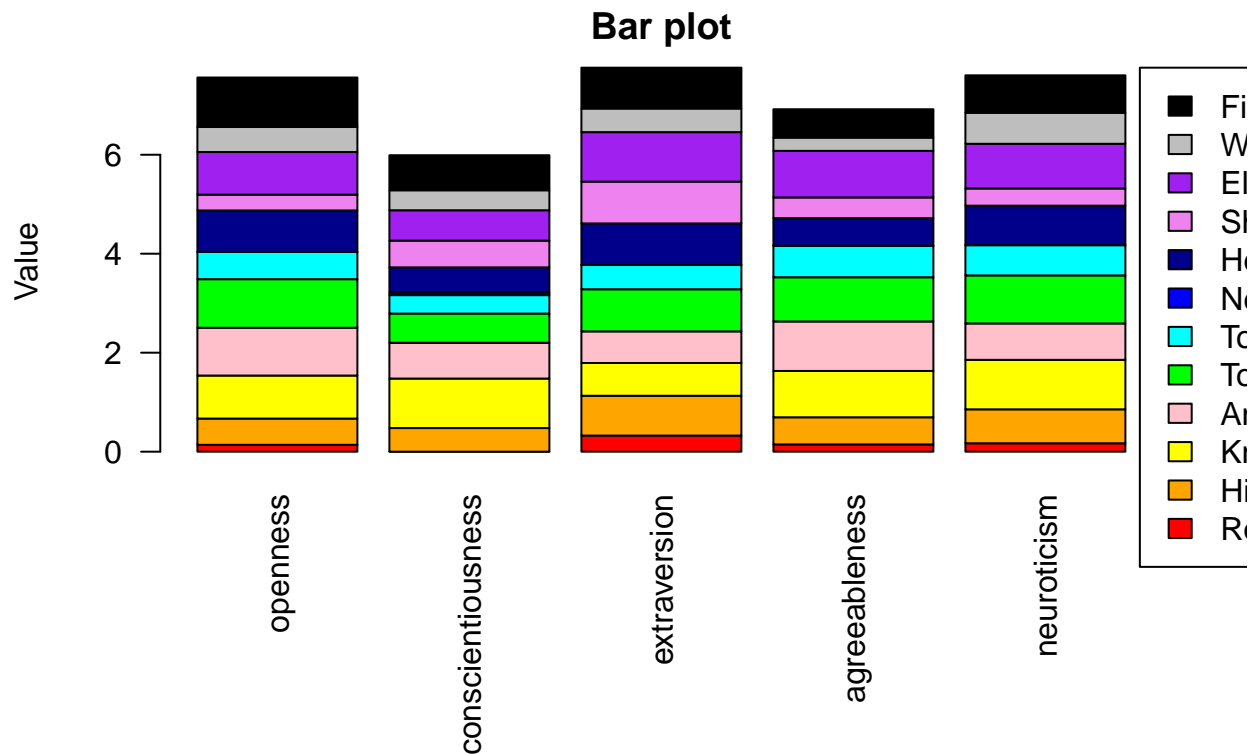
Presented below is a stacked bar plot designed for the analysis of the significance of various joints in evaluating personality traits.

```
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", "Purple", "Grey", "Black"))
```



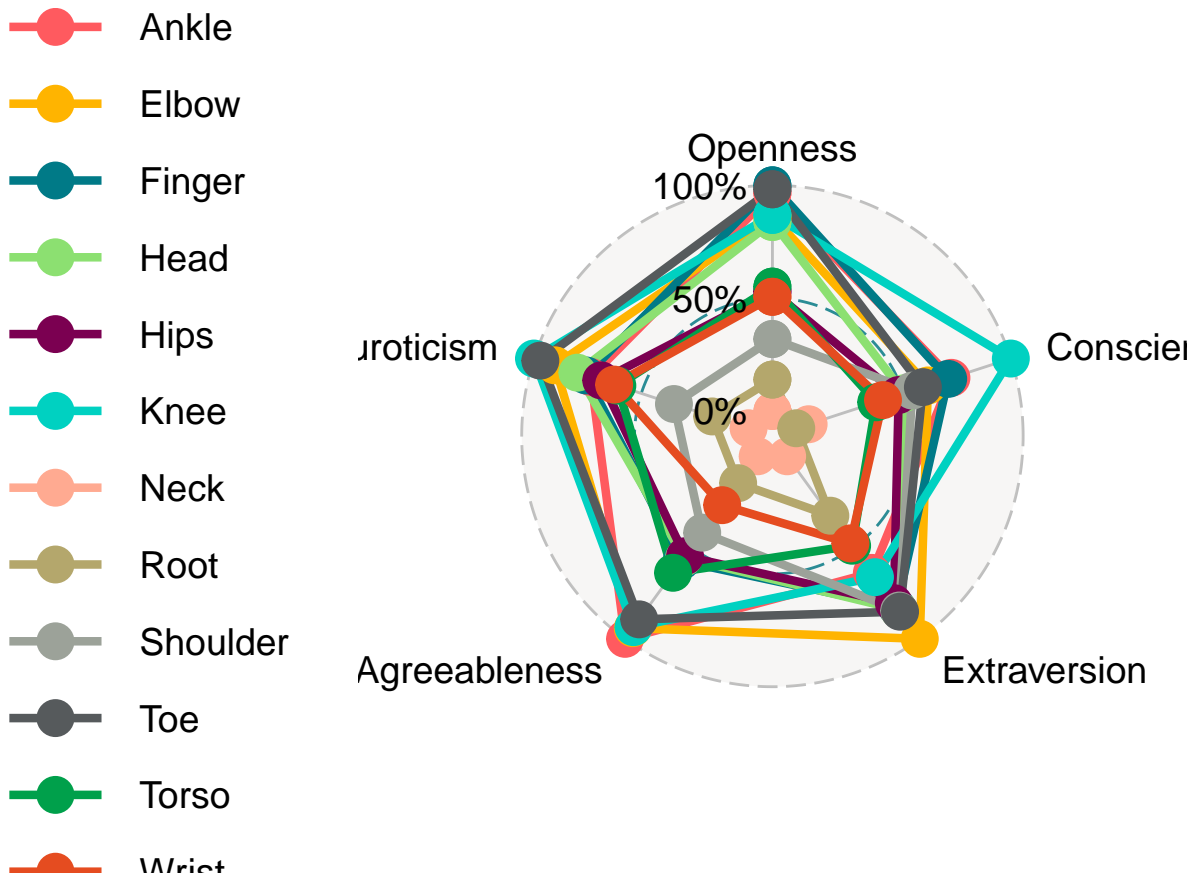
The provided graph illustrates the proportionate importance of each joint in the evaluation of personality traits. However, the visibility of the ratios is suboptimal, rendering this graph unsuitable for effective visualization.

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

The Radar Chart stands out as the ideal visualization to represent the joint importance in predicting personality traits, efficiently condensing multidimensional data into a 2-dimensional chart. This choice is justified by its capacity to unveil patterns of association or dissociation among the 12 joint importance vectors for each personality trait.

In this visualization, each personality dimension (Openness, Conscientiousness, Extraversion, Agreeableness, Neuroticism) is intricately linked to unique patterns of joint movement characteristics (Root, Hips, Knees, Ankle, Toe, Torso, Neck, Head, Shoulder, Elbow, Wrist, Finger). These joint movements strongly associate with each personality trait, evident in the average joint importance vector scores.

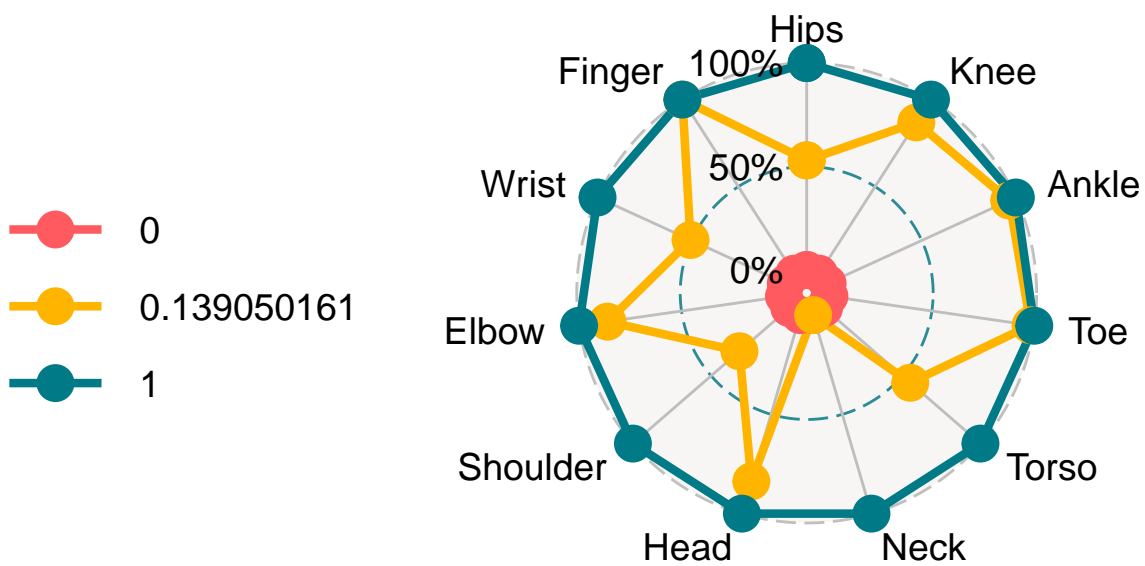
```
library(ggradar)
ggradar(motion_data)
```



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.

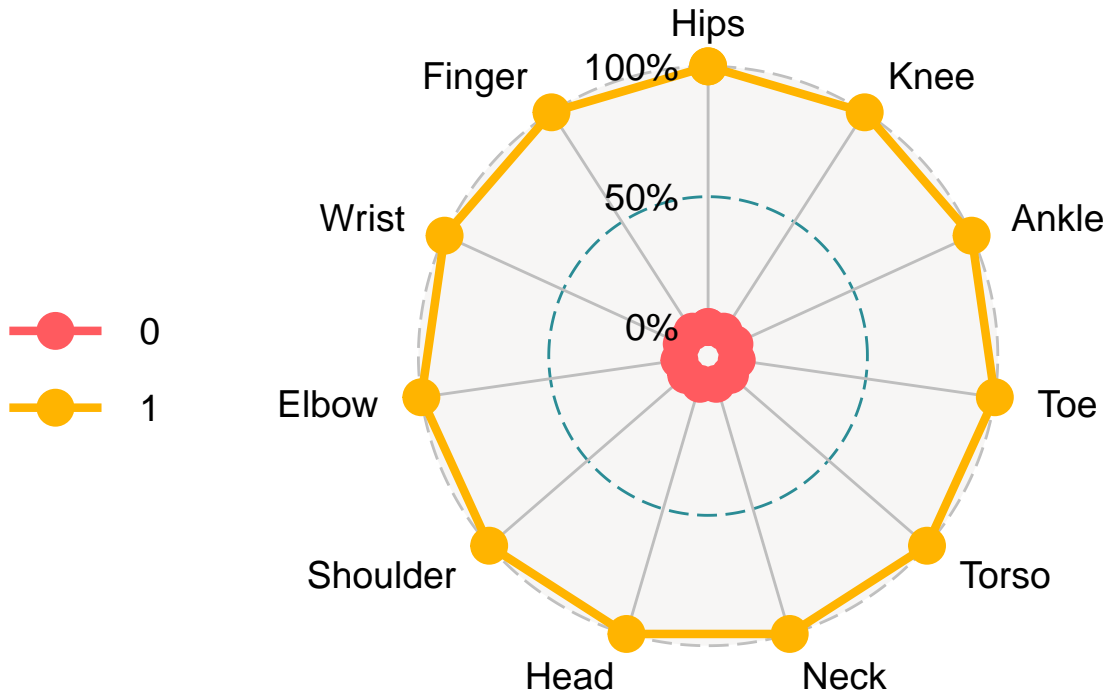
```
data1 <- data.frame(rbind(rep(1, 12), rep(0, 12), t(matrix(motion_data$Openness))))
colnames(data1) <- c("Root", "Hips", "Knee", "Ankle", "Toe", "Torso", "Neck", "Head", "Shoulder", "Elbow", "Finger", "Wrist")
ggradar(data1)
```



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)
```



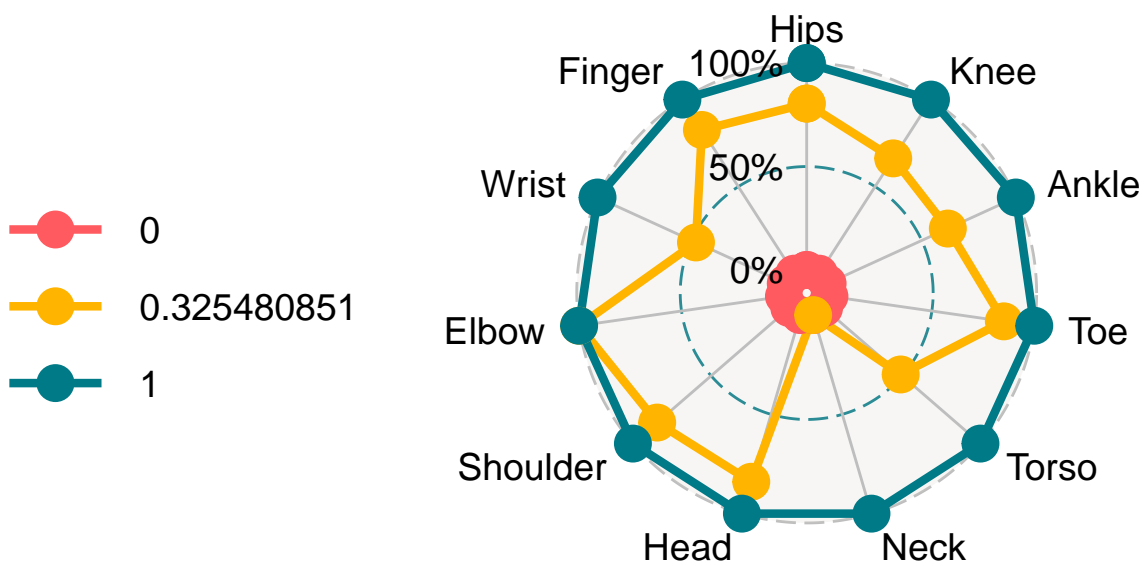


The ensuing plot pertains to the personality trait - Extraversion

```
print(motion_data$Extraversion)
```

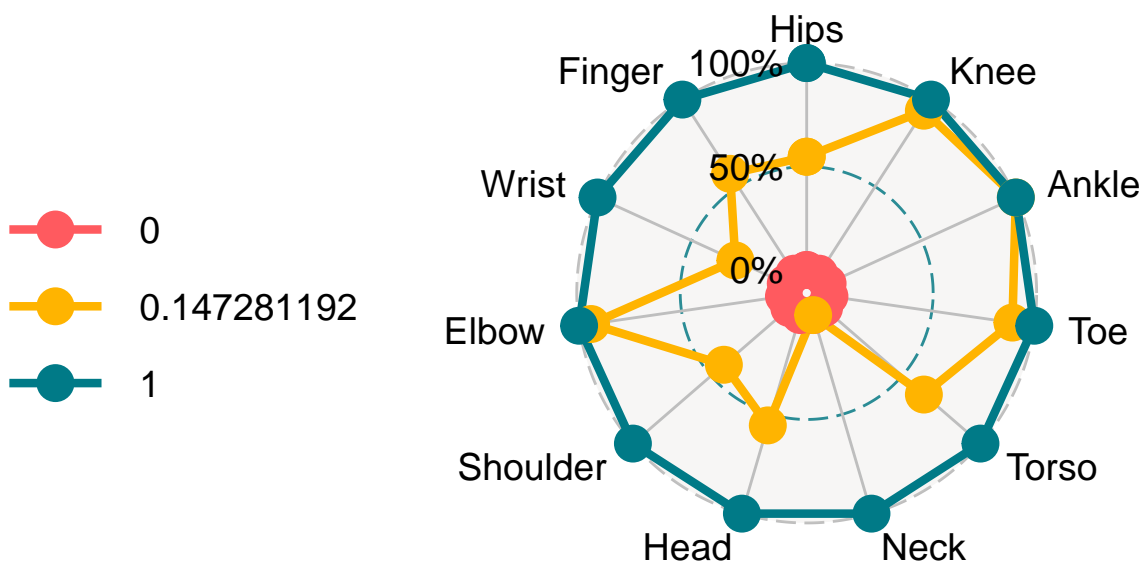
```
## [1] 0.3254809 0.8042766 0.6621905 0.6387358 0.8512893 0.4899196 0.0000000
## [8] 0.8395976 0.8449116 1.0000000 0.4770724 0.8255081
```

```
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)
```



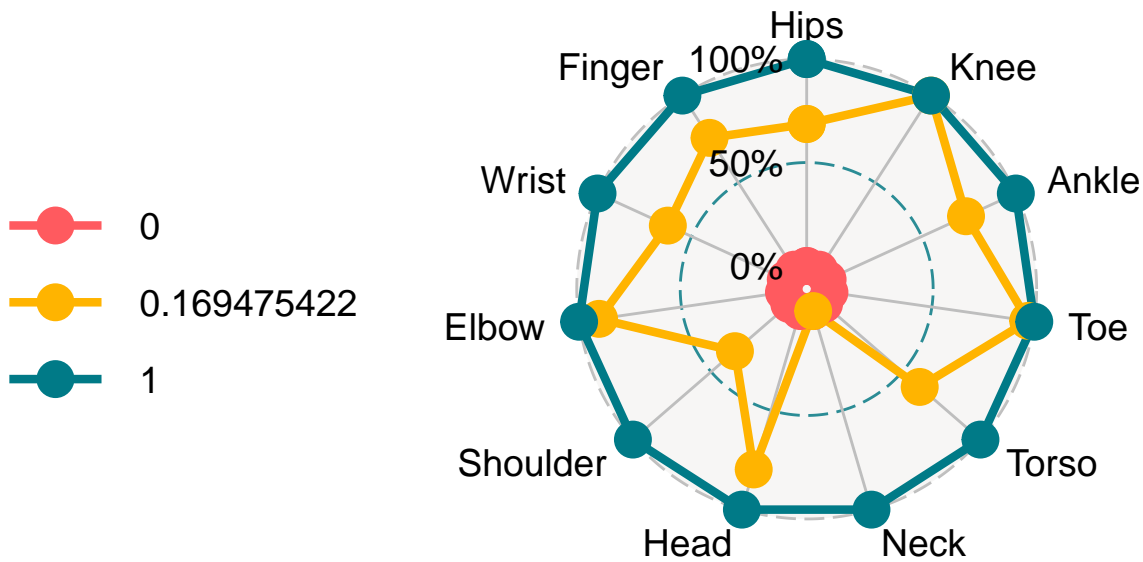
The ensuing plot pertains to the personality trait - Agreeableness

```
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)
```



In summary, Openness, Extraversion, and Neuroticism traits reveal connections with a higher number of movement components. Extraversion is notably linked to both upper and lower body movements, emphasizing upper joint movements. Neuroticism places significant weight on knee, ankle, toe, and joints in comparison to upper extremity movements. Agreeableness strongly associates with lower body joint movements, while Openness demonstrates strong connections in both upper and lower body joint movements.

Furthermore, joint movements such as Root, Neck, and Wrist don't strongly correlate with personality traits. Meanwhile, Hip movement notably associates with Extraversion and Neuroticism.

The individual filled Radar plots provide a detailed view of joint movements for each personality trait, from Openness to Neuroticism, delivering a comprehensive visual representation of the intricate relationships between joint characteristics and personality dimensions.

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

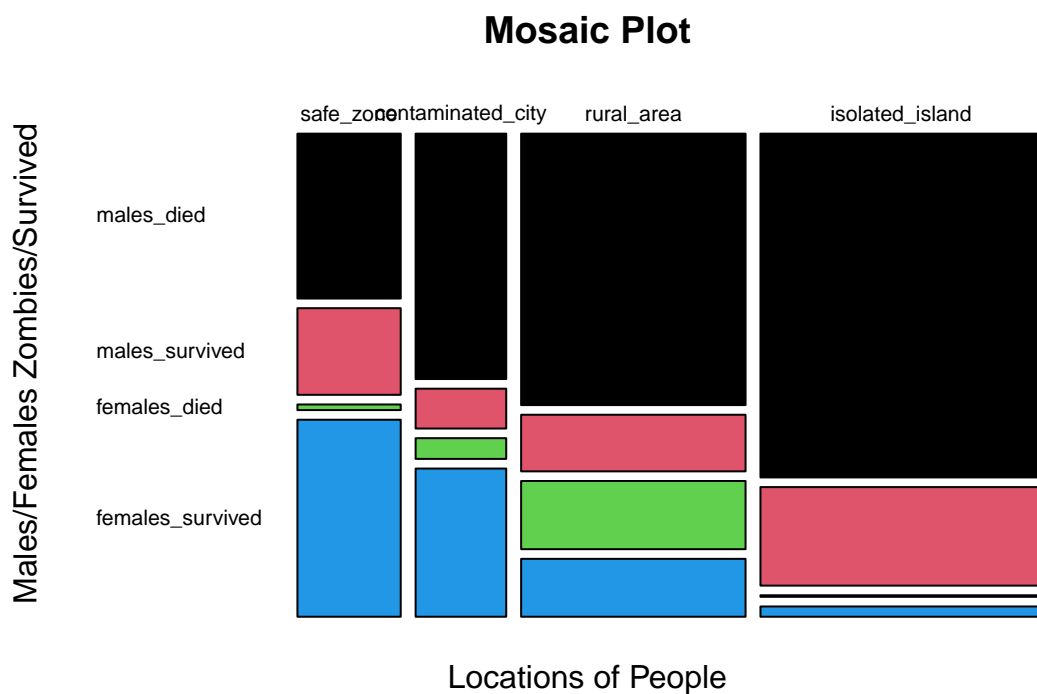
### Subtask 3.1 (The Last of Us)

```
data = matrix(c(118, 62, 4, 141, 154, 25, 13, 93, 422, 88, 106, 90, 670, 192, 3, 20), ncol = 4, byrow =
rownames(data) <- c('safe_zone', 'contaminated_city', 'rural_area', 'isolated_island')
```

```
colnames(data) <- c('males_died', 'males_survived', 'females_died', 'females_survived')
print(data)
```

```
##           males_died males_survived females_died females_survived
## safe_zone           118           62           4           141
## contaminated_city    154           25           13           93
## rural_area           422           88          106           90
## isolated_island      670          192           3           20
```

```
mosaicplot(data, color=1:4, las=1, legend=TRUE, xlab="Locations of People", ylab="Males/Females Zombies/Survived",
```



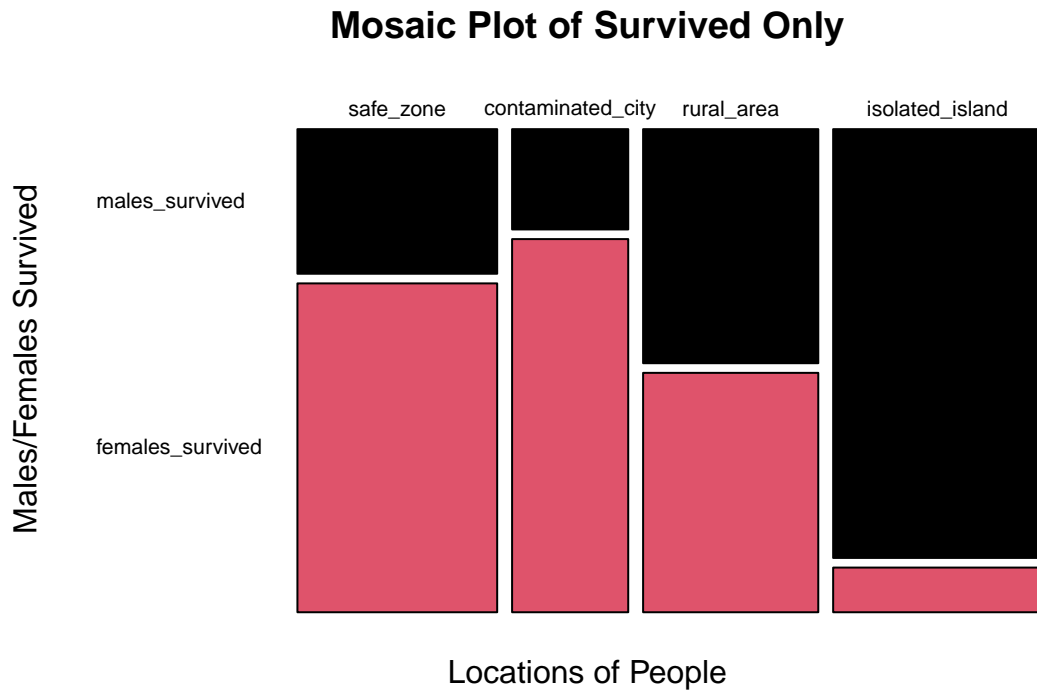
```
data = matrix(c(62, 141, 25, 93, 88, 90, 192, 20), ncol = 2, byrow = TRUE)

rownames(data) <- c('safe_zone', 'contaminated_city', 'rural_area', 'isolated_island')
colnames(data) <- c('males_survived', 'females_survived')

print(data)
```

```
##           males_survived females_survived
## safe_zone              62             141
## contaminated_city      25             93
## rural_area             88             90
## isolated_island       192             20
```

```
mosaicplot(data, color=1:4, las=1, legend=TRUE, xlab="Locations of People", ylab="Males/Females Survived")
```



The Mosaic Plot emerges as the optimal choice for the provided data, particularly suited for categorical and hierarchical data. It effectively visualizes relationships between different categories, showcasing symmetries within tables, especially in scenarios with a hierarchy of labeled elements and numerous categories.

In the specific context of this analysis, the mosaic plot illustrates the distribution of survival across four zones (Safe Zone, Contaminated City, Rural Area, and Isolated Island) based on gender. Rectangles represent the components, forming a table with rectangles of varying sizes and colors. Key observations include:

Males generally exhibited higher survival rates than females across all zones. For females, the highest survival rate was in the Safe Zone, while for males, it was in the Isolated Island. A higher number of females in the Rural Area did not survive, whereas more males died in the Isolated Island. More male casualties occurred in the Safe Zone compared to female casualties, indicating a higher survival rate for females. The lowest survival rates for females were observed in the Isolated Island, while for males, it was in the Contaminated City.

### Subtask 3.2: Glass Glimpse

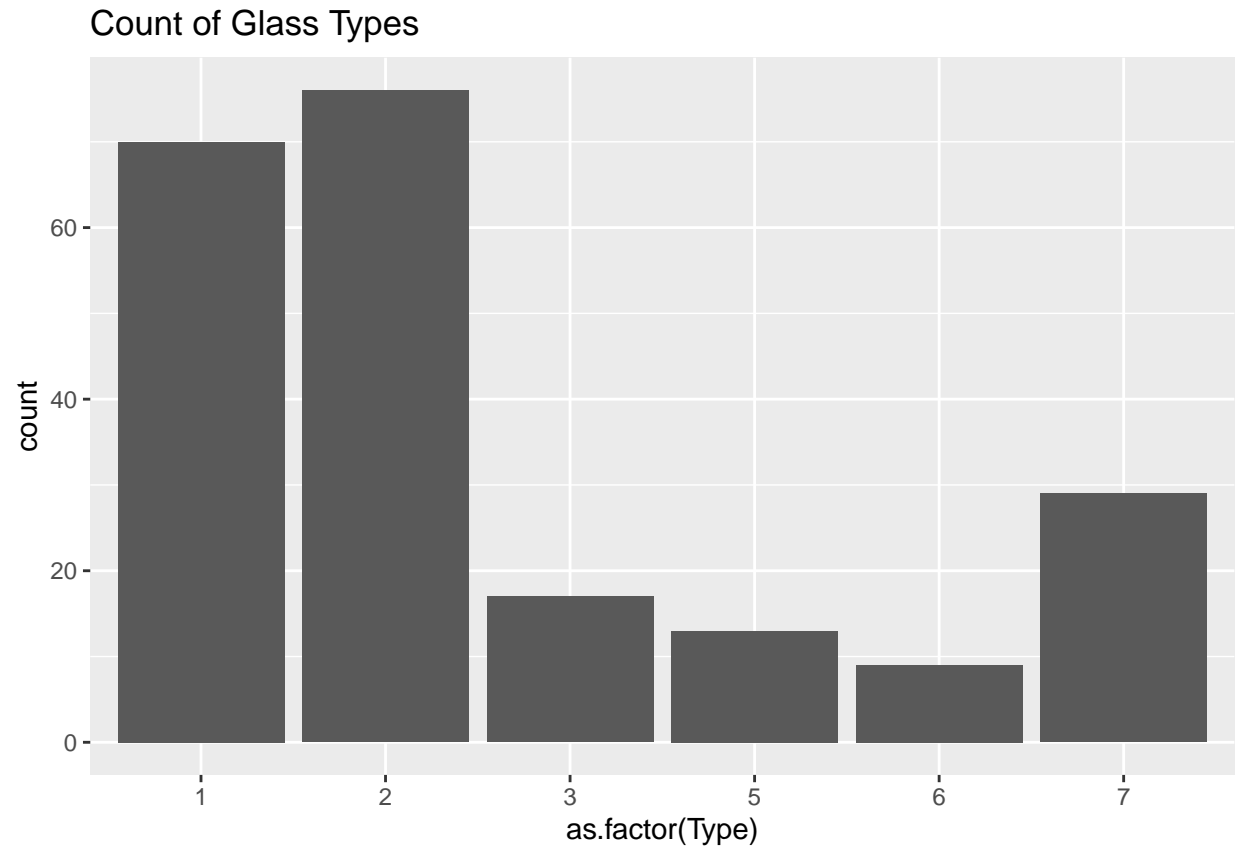
```
library(tidyverse)
library(readxl)
library(corrplot)
glass_data <- read_excel("./2024_Assignment1_BRSM.xlsx", sheet = 3)

# 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")
```

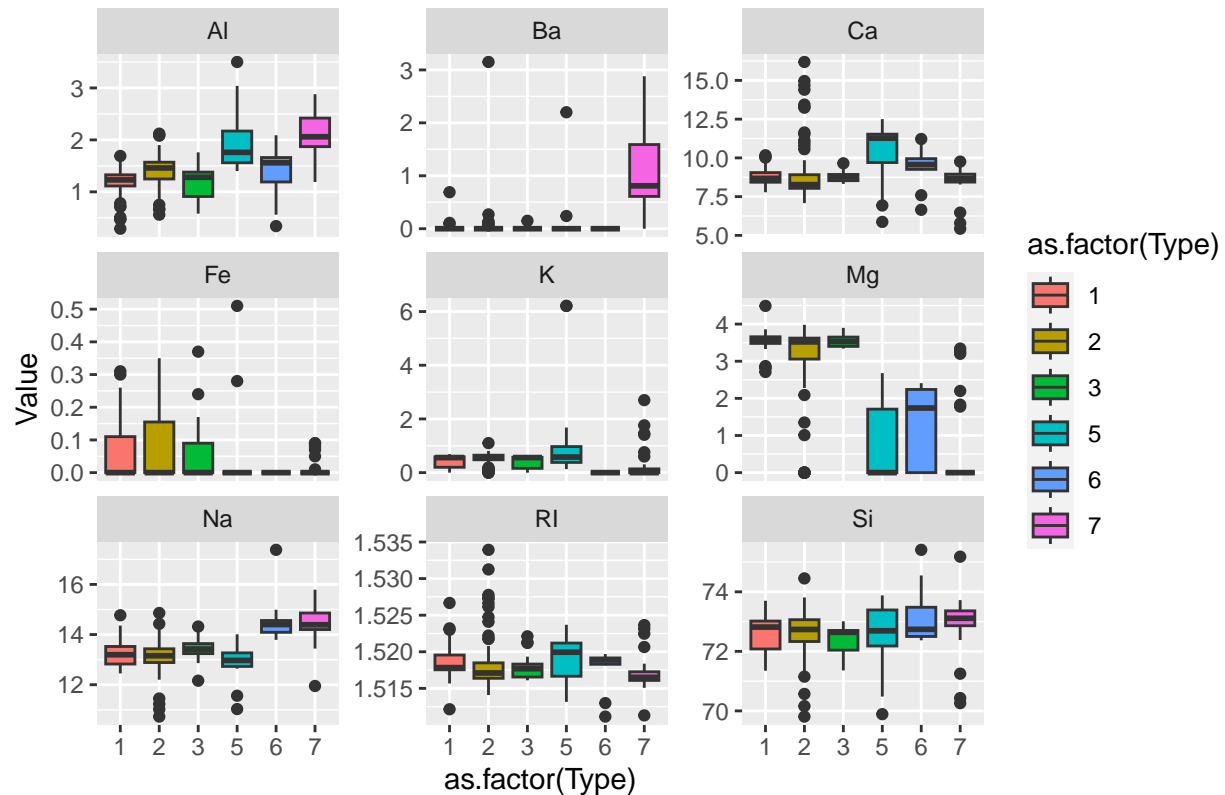


Type 1 and Type 2 are the most prevalent categories in the dataset, characterized as building windows subjected to a float processing method (Type 1) and building windows subjected to a non-float processing method (Type 2).

```
# 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")
```



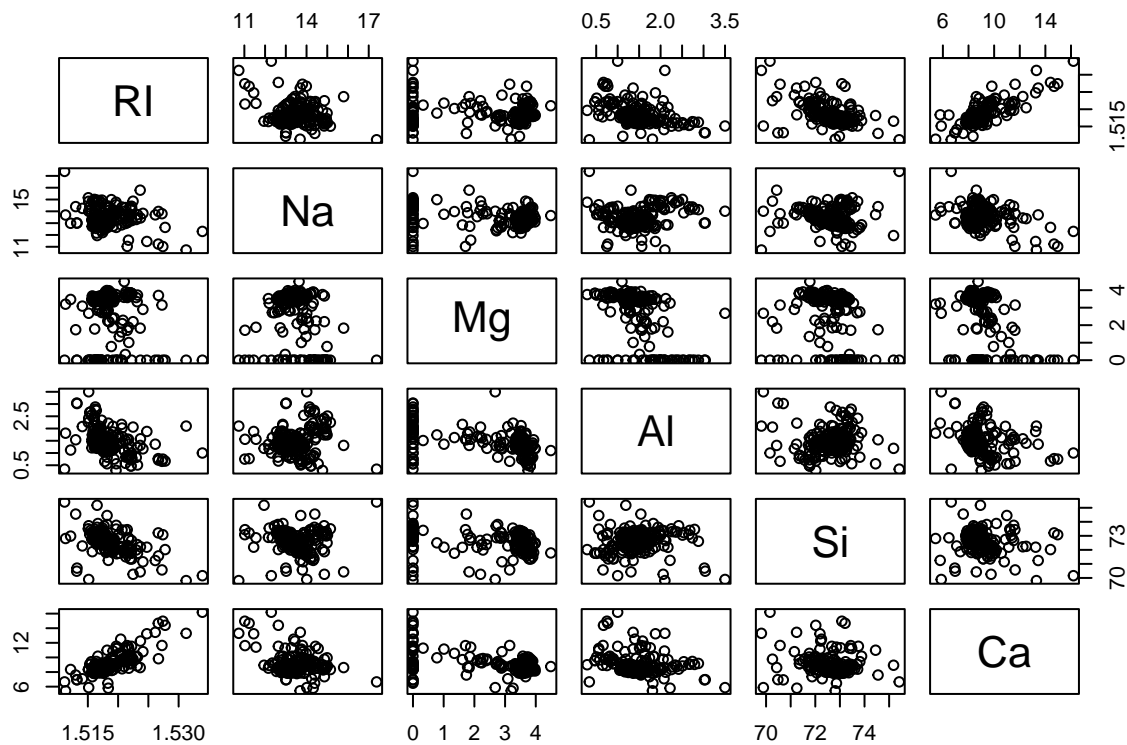
## Box Plots for Different Variables Grouped by Glass Type



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.

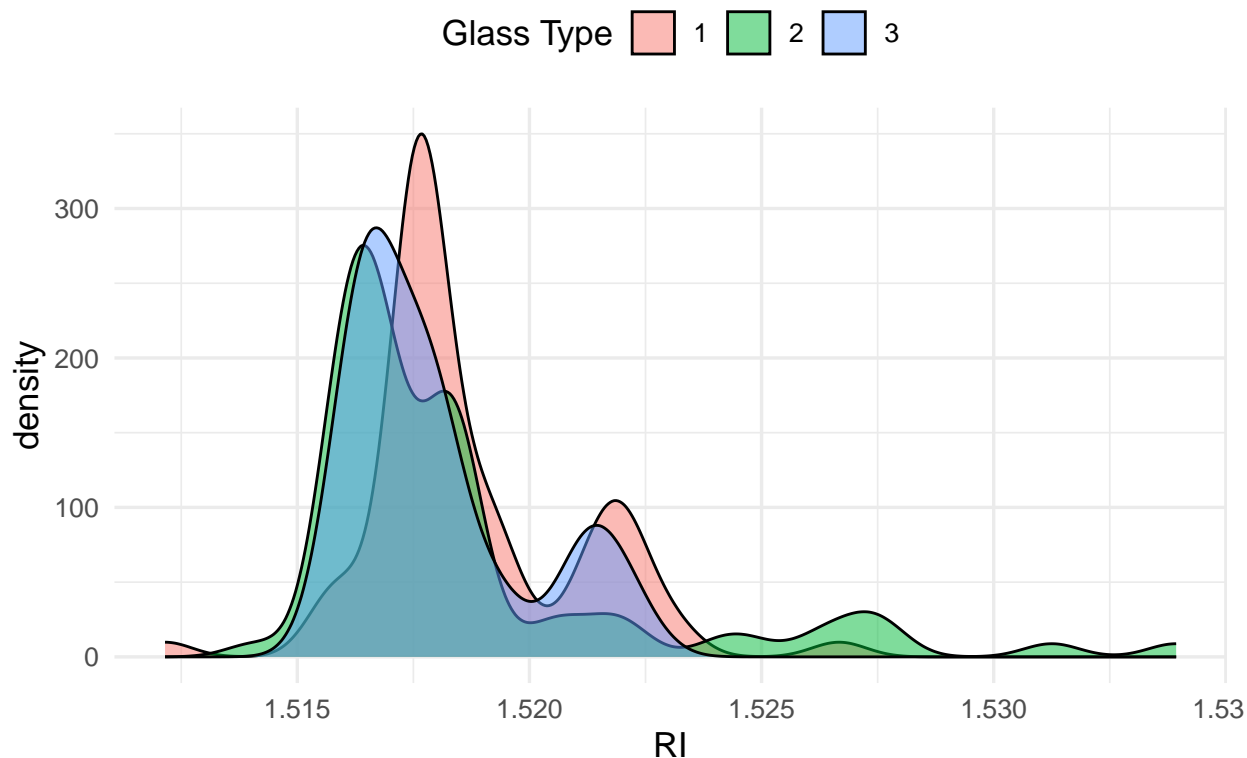
```
# Set the style and font scale
theme_set(theme_minimal(base_size = 13))

# 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")
```

## Density Distribution for RI (Type 1, 2, 3)



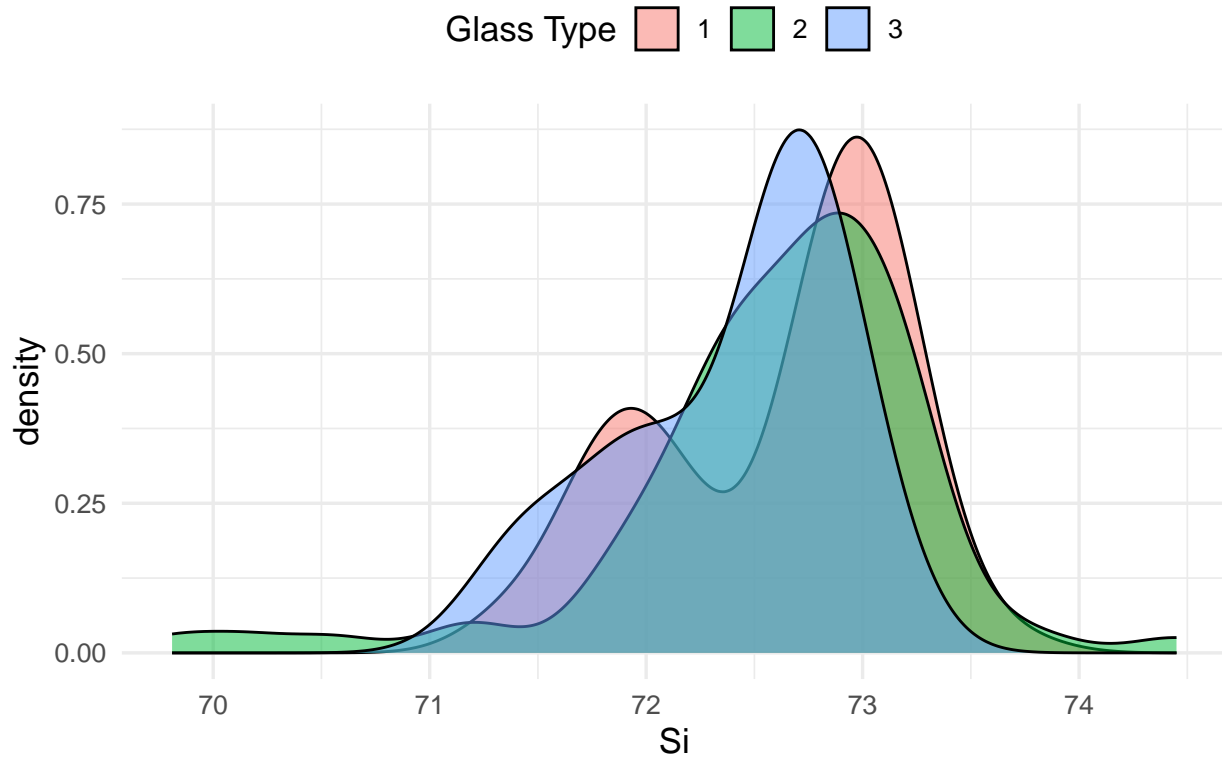
```
# Set the style and font scale
theme_set(theme_minimal(base_size = 13))

# 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")
```

## Density Distribution for Si (Type 1, 2, 3)



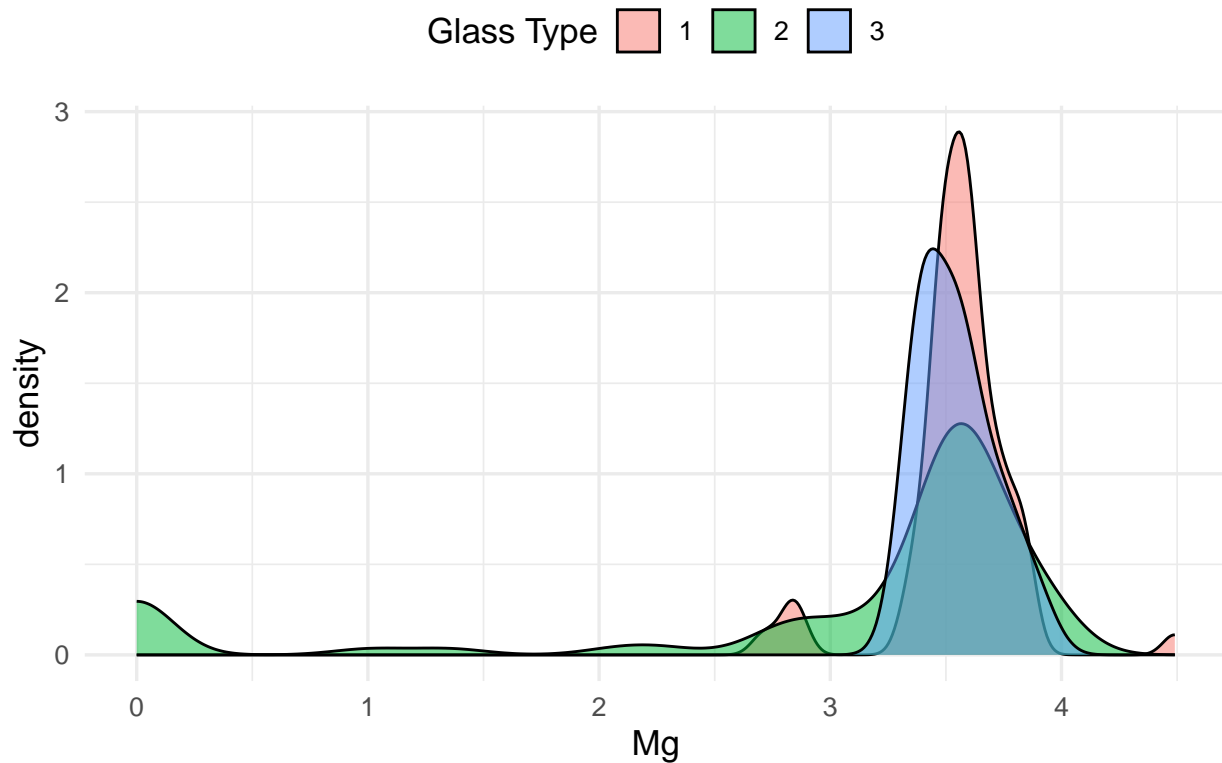
```
# Set the style and font scale
theme_set(theme_minimal(base_size = 13))

# 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")
```

## Density Distribution for Mg (Type 1, 2, 3)



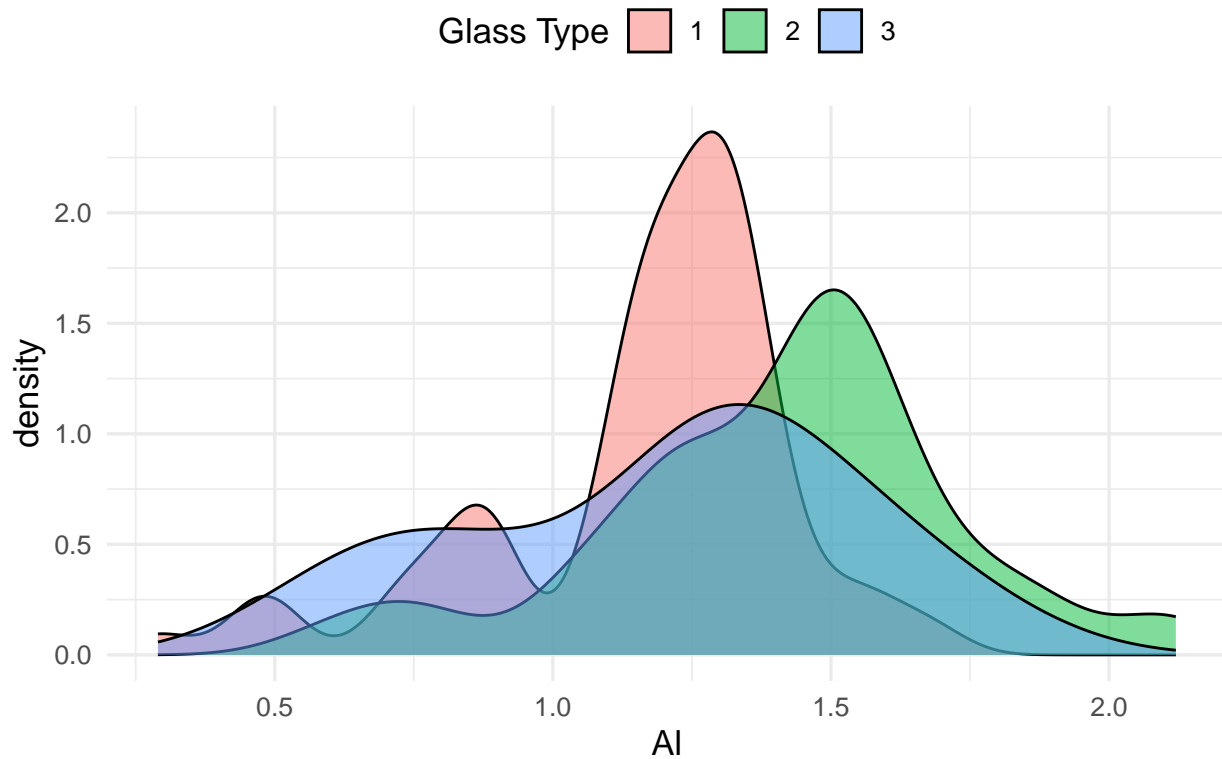
```
# Set the style and font scale
theme_set(theme_minimal(base_size = 13))

# 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)

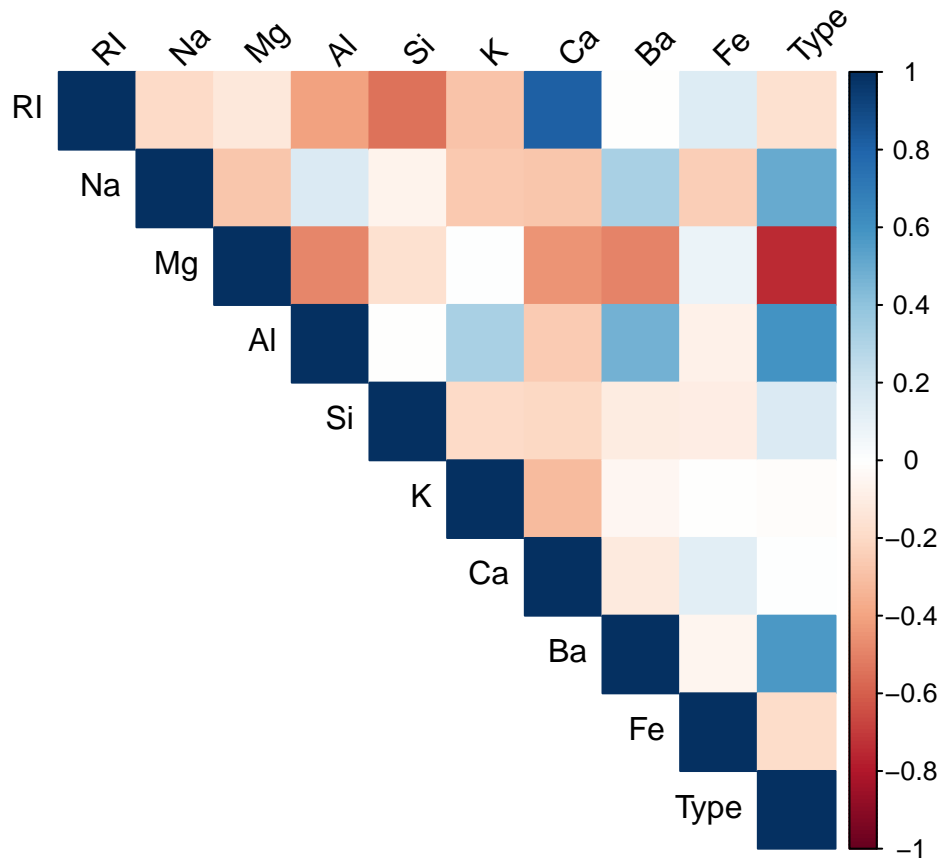
# Plot density distribution separately for Al in a separate row
ggplot(glass_flat_df_al, aes(x = Al, fill = as.factor(Type))) +
  geom_density(alpha = 0.5) +
  ggtitle("Density Distribution for Al (Type 1, 2, 3)") +
  labs(fill = "Glass Type") +
  theme_minimal(base_size = 13) +
  theme(legend.position = "top")
```

## Density Distribution for Al (Type 1, 2, 3)



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.

### 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 %>%
```

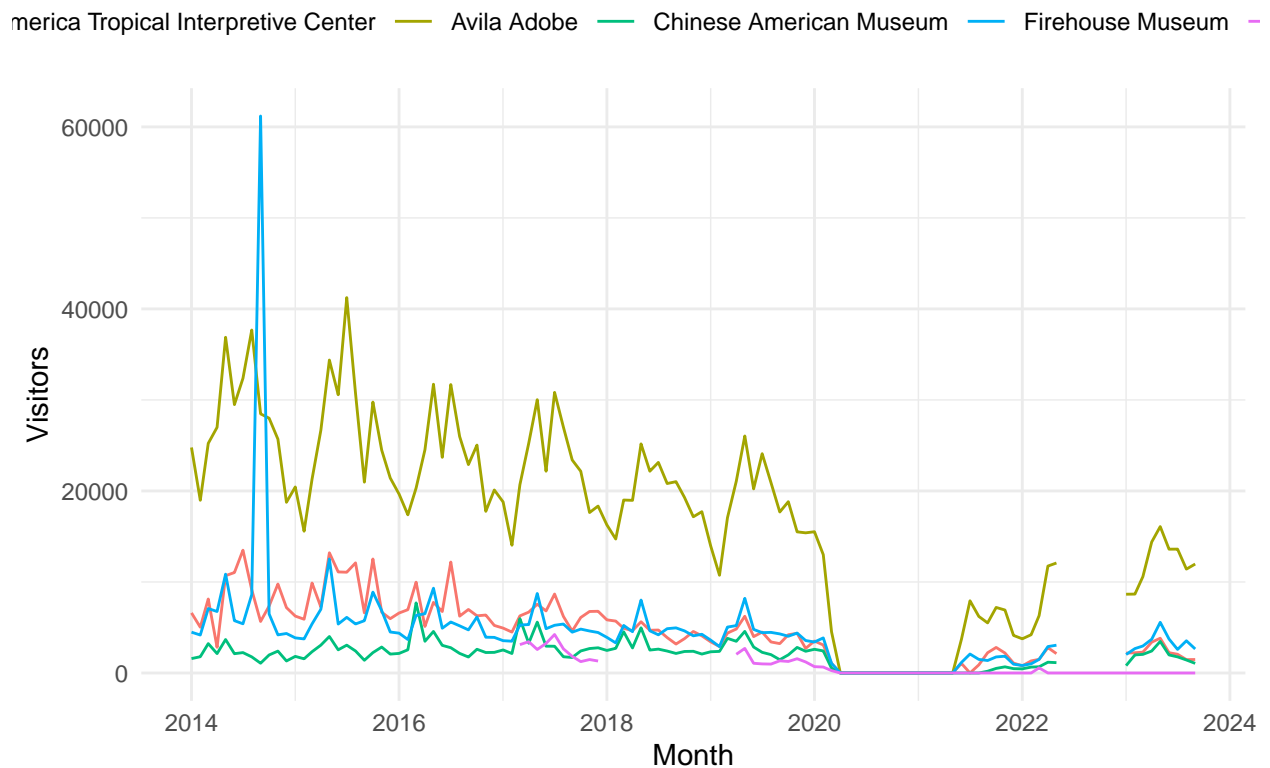
```

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")

```

## Monthly Visitors to Museums Over Time



The analysis through line plots offers valuable insights into visitor counts for various museums during both pandemic and regular days. Key observations include:

- Pandemic Period (Lowest Visitor Count):

All museums experienced a decline in visitor counts during the pandemic, with Avila Adobe registering the lowest count.

- Regular Days (Rising Visitor Count):

During regular days, visitor counts increased uniformly across all museums. The consistent upward trend over time suggests a proportional rise in attendance for each museum on regular days.

- Decreasing Trend Over Time:



The decreasing trend over time is consistently observed across museums in the following order: Avila Adobe America Tropical Interpretive Center–Firehouse Museum Chinese American Museum Gateway to Nature Center

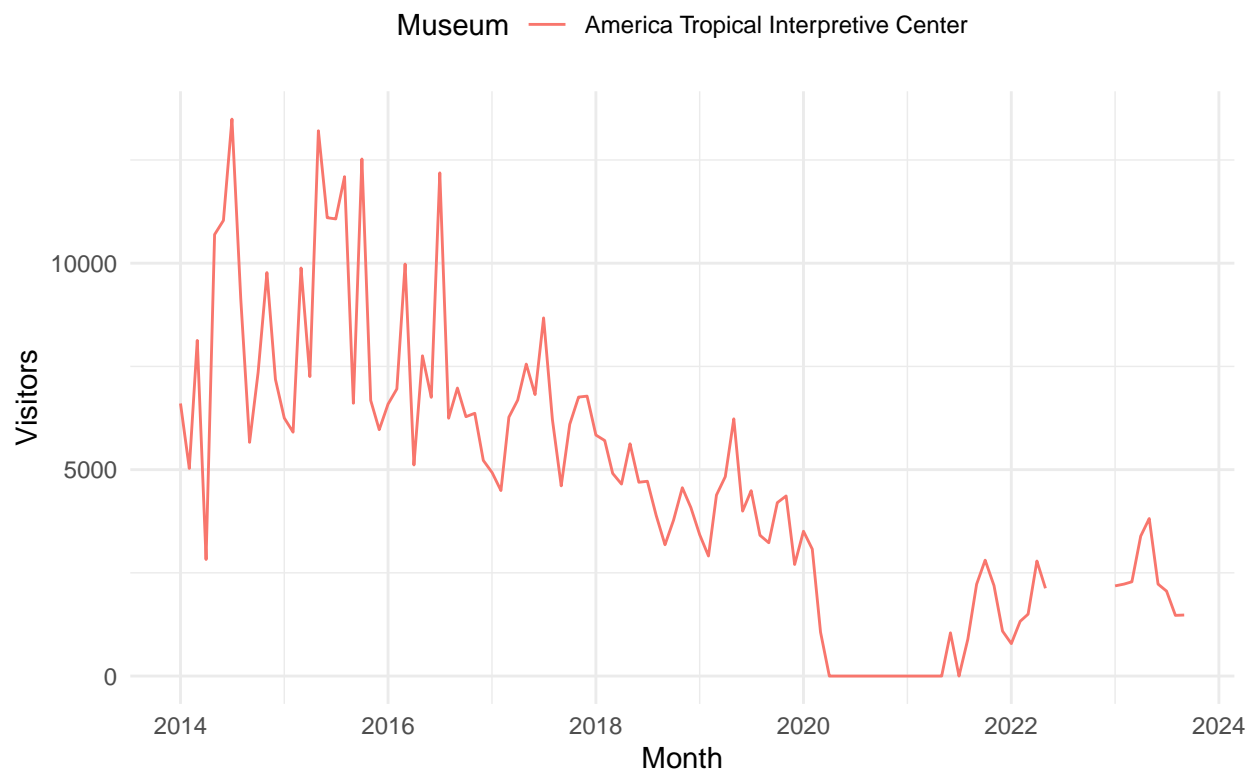
- Line Plot Advantages:

The use of lines enhances trend and pattern identification, providing visual continuity for an intuitive understanding of changing values over time. Line plots effectively highlight upward or downward trends, aiding in the discernment of the overall trajectory of visitor counts for each museum. In summary, the line plot analysis offers valuable insights into museum dynamics during pandemic and regular periods. The visual continuity of lines improves trend interpretability, facilitating an intuitive understanding of changing patterns over time.

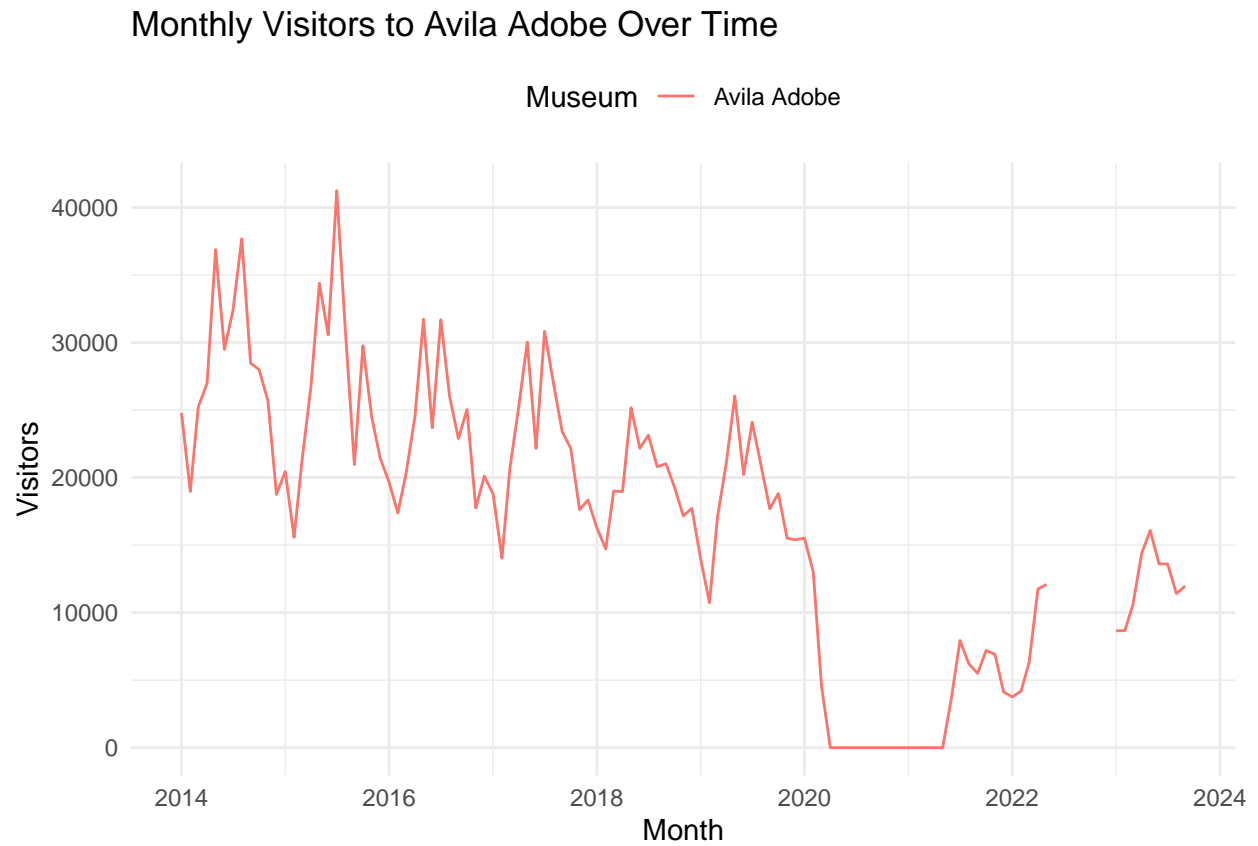
```
plot_seasonal_line_chart <- function(museum_name) {
  ggplot(subset(melted_data, Museum == museum_name),
    aes(x = Month, y = Visitors, color = Museum)) +
  geom_line() +
  labs(title = paste("Monthly Visitors to", museum_name, "Over Time"),
    x = "Month",
    y = "Visitors") +
  theme_minimal() +
  theme(legend.position = "top")
}

plot_seasonal_line_chart("America Tropical Interpretive Center")
```

## Monthly Visitors to America Tropical Interpretive Center Over Time

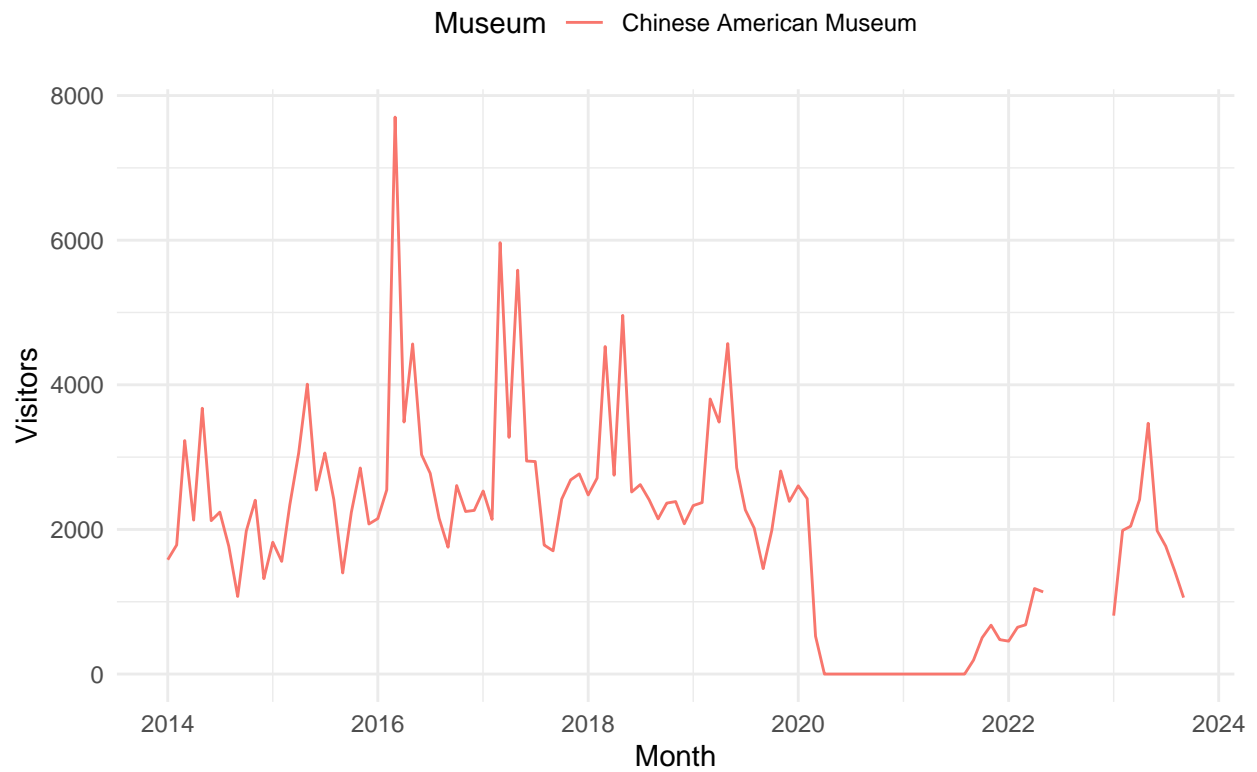


```
plot_seasonal_line_chart("Avila Adobe")
```



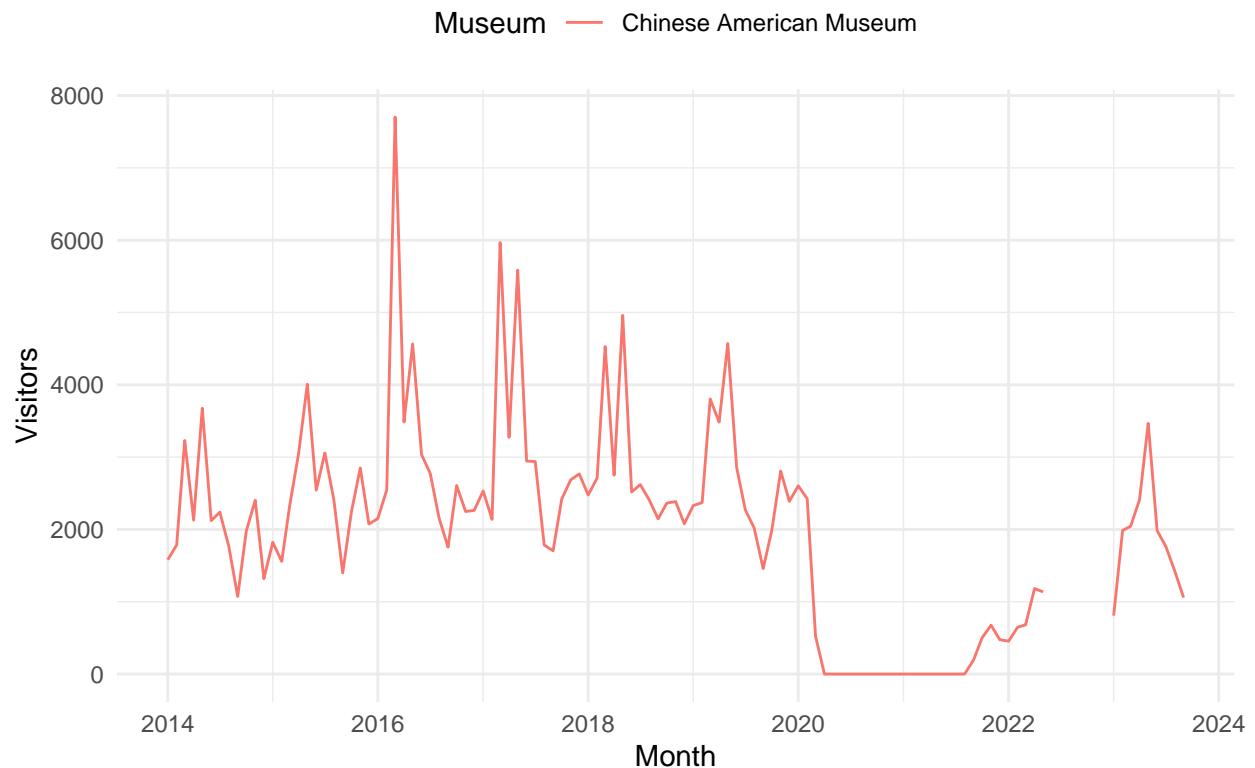
```
plot_seasonal_line_chart("Chinese American Museum")
```

## Monthly Visitors to Chinese American Museum Over Time



```
plot_seasonal_line_chart("Chinese American Museum")
```

## Monthly Visitors to Chinese American Museum Over Time



A line graph effectively illustrates trends compared to other types of plots. From the depicted graph, it is evident that the Avila Adobe museum consistently attracted more visitors over time. The Firehouse museum experienced a sudden spike at a certain point. Notably, all museums witnessed a significant drop in visitors during the 2020-21 period, coinciding with the COVID pandemic. Additionally, the plot indicates an overall decreasing trend over time, revealing several other insightful patterns.

## 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$brand <- as.numeric(factor(fast_furious_data$brand))
View(is.na(fast_furious_data))
fast_furious_data <- na.omit(fast_furious_data)
summary(fast_furious_data)
```

##	mpg	cylinders	cubicinches	hp	weightlbs
## Min.	:10.00	Min. :3.00	Min. : 70.0	Min. : 46.0	Min. :1613
## 1st Qu.	:16.80	1st Qu.:4.00	1st Qu.:100.2	1st Qu.: 75.0	1st Qu.:2246
## Median	:22.00	Median :5.00	Median :156.0	Median : 95.0	Median :2832

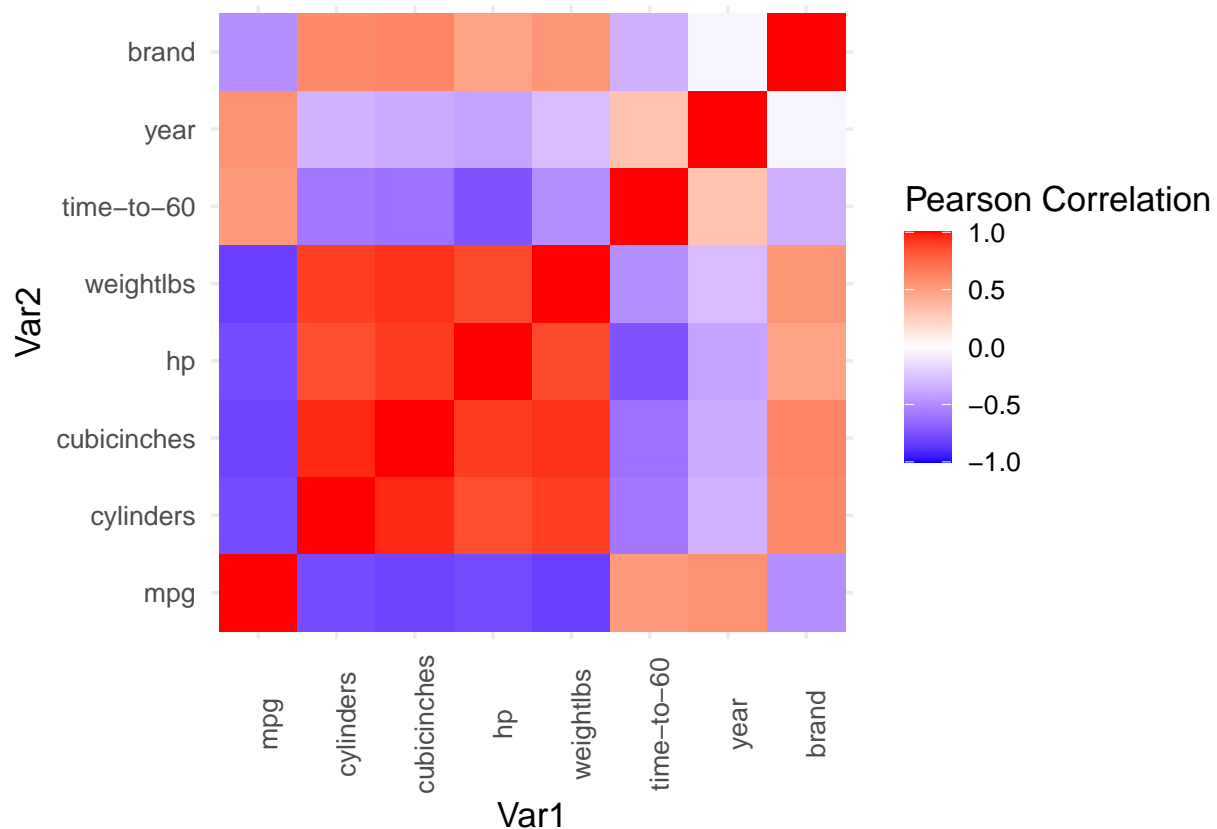
```
## Mean :23.19 Mean :5.59 Mean :201.4 Mean :106.8 Mean :3006
## 3rd Qu.:28.85 3rd Qu.:8.00 3rd Qu.:304.0 3rd Qu.:139.0 3rd Qu.:3666
## Max. :46.60 Max. :8.00 Max. :455.0 Max. :230.0 Max. :4997
## time-to-60 year brand
## Min. : 8.0 Min. :1971 Min. :1.000
## 1st Qu.:14.0 1st Qu.:1974 1st Qu.:2.000
## Median :16.0 Median :1977 Median :3.000
## Mean :15.5 Mean :1977 Mean :2.434
## 3rd Qu.:17.0 3rd Qu.:1980 3rd Qu.:3.000
## Max. :25.0 Max. :1983 Max. :3.000
```

```
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"))
```

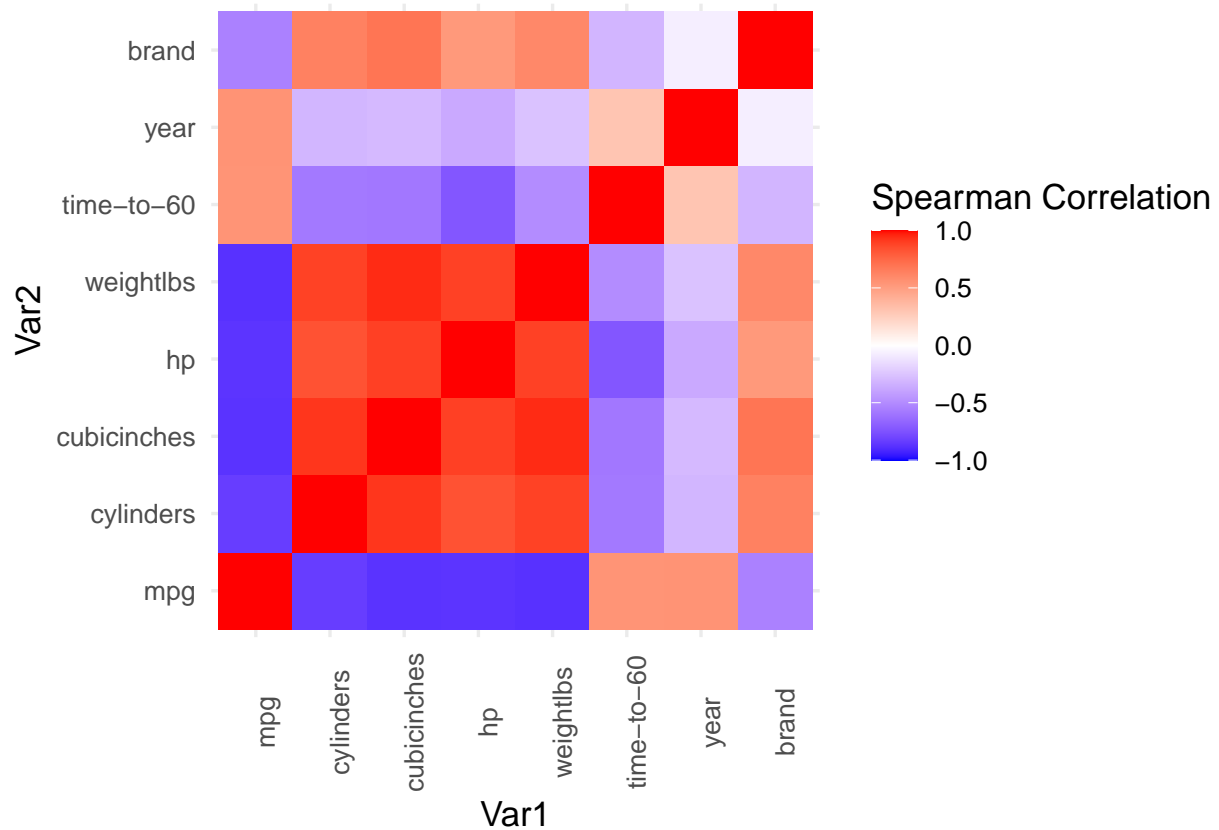
Now, the following is the heat map when the correlation matrix is chosen to be Pearson's.

```
ggplot(data=mat1, aes(x=Var1, y=Var2, fill=value)) +
  geom_tile() +
  scale_fill_gradient2(low = "blue", high="red", limit = c(-1, 1), name="Pearson Correlation") +
  theme(axis.text.x = element_text(angle = 90))
```



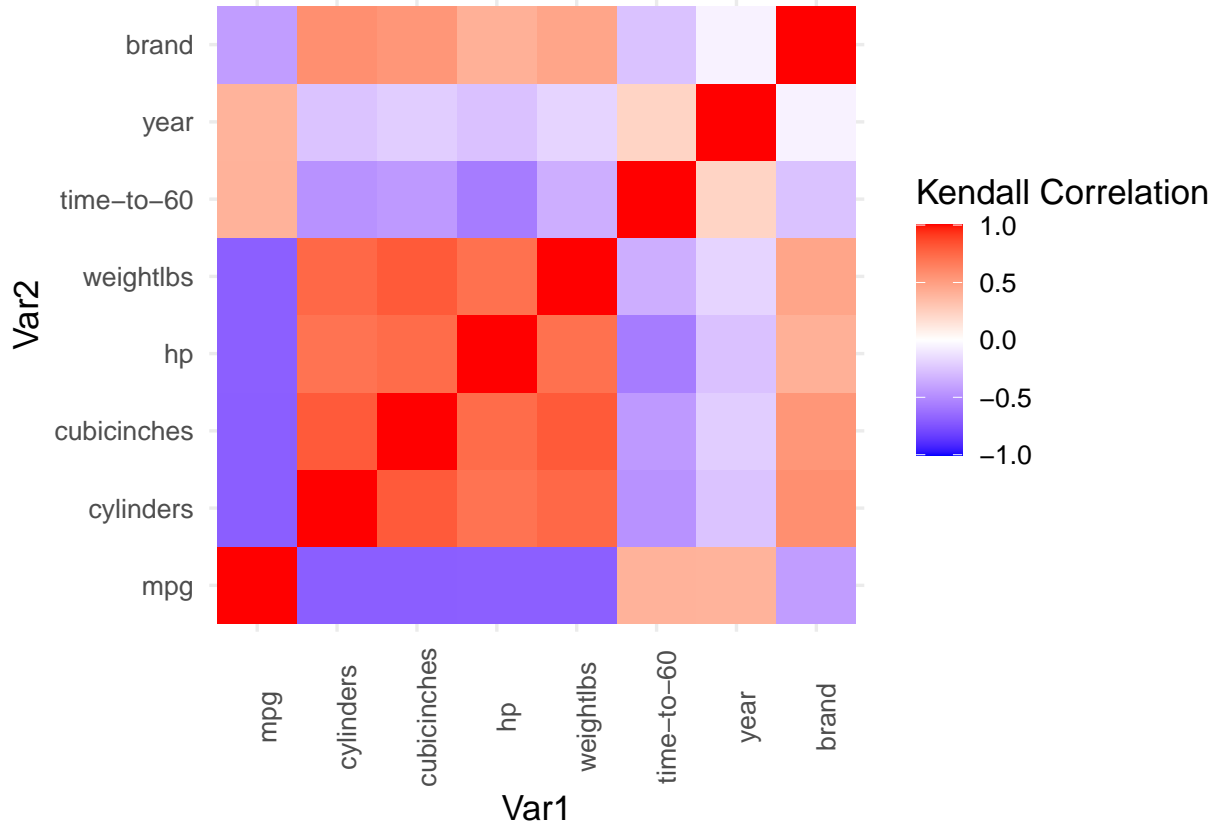
Now, the following is the heat map when the correlation matrix is chosen to be Spearman's.

```
ggplot(data=mat2, aes(x=Var1, y=Var2, fill=value)) +
  geom_tile() +
  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() +
  scale_fill_gradient2(low = "blue", high="red", limit = c(-1, 1), name="Kendall Correlation") +
  theme(axis.text.x = element_text(angle = 90))
```



The automobile dataset, encompassing characteristics like car make, horsepower, mileage, and price, is visually represented through a heatmap. The analysis reveals that cars with smaller engine sizes tend to be more affordable, given their lower fuel consumption and vehicle prices. Specifically, the price variable exhibits positive correlations with wheelbase, length, width, curb weight, engine size, and horsepower. Conversely, price shows negative correlations with peak RPM, city mileage, and highway mileage per gallon.

Variables such as bore, engine size, and curb weight demonstrate strong positive correlations with wheelbase, length, and width. Larger engine sizes are linked to higher horsepower. Both highway and city mileage per gallon display negative correlations with various factors, including wheelbase, length, width, and horsepower. The peak RPM of the car exhibits a negative relationship with the compression ratio.

Additionally, wheelbase, length, width, and height exhibit positive correlations with each other. This analysis provides valuable insights into the relationships among different features in the dataset, presenting a comprehensive understanding of automobile characteristics and their interconnections.

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.