Practical 2

Working with data

Joan Navarro Bellido

EXERCISE 1

```r
# Load the dataset
titanic <- read.csv("./datasets/titanic.csv", header = TRUE, sep = ",")
```

```r
# Remove the row index column
titanic <- subset(titanic, select = -X)
```

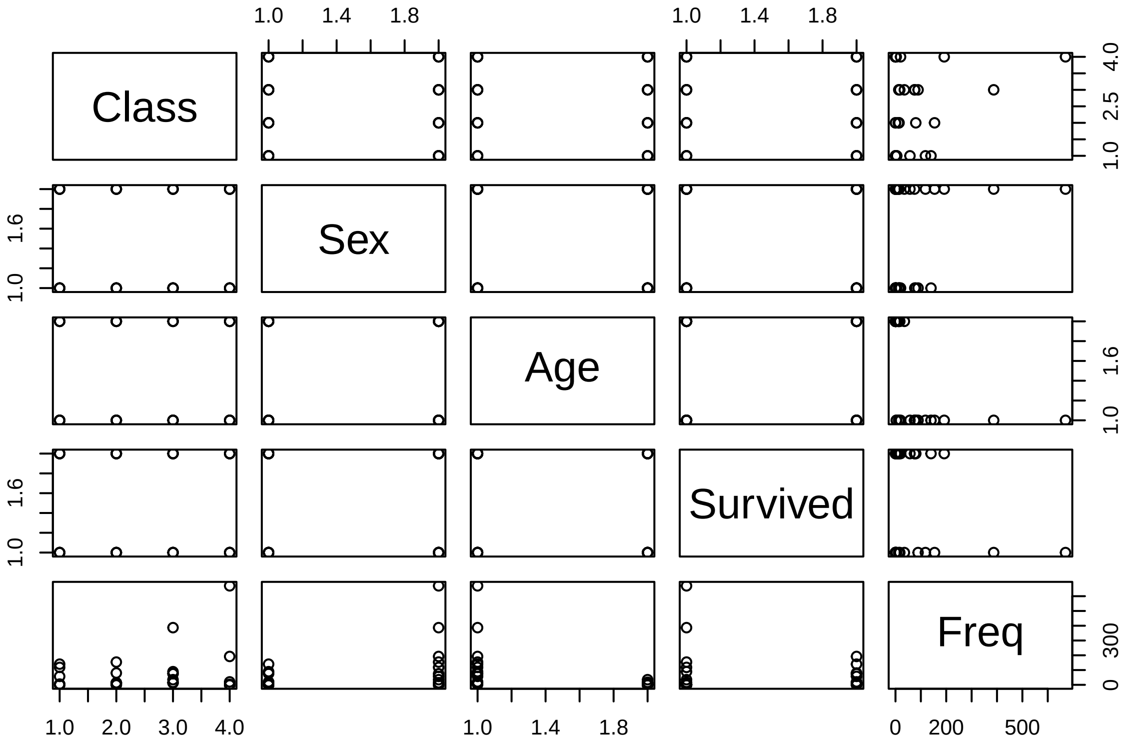
```r
# Display the first few rows of the dataset
head(titanic)
```

```
##   Class    Sex   Age Survived Freq
## 1   1st   Male Child       No    0
## 2   2nd   Male Child       No    0
## 3   3rd   Male Child       No   35
## 4  Crew   Male Child       No    0
## 5   1st Female Child       No    0
## 6   2nd Female Child       No    0
```

```r
# Display the summary statistics of the dataset
summary(titanic)
```

```
##     Class               Sex                Age              Survived
##  Length:32          Length:32          Length:32          Length:32
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##       Freq
##  Min.   :  0.00
##  1st Qu.:  0.75
##  Median : 13.50
##  Mean   : 68.78
##  3rd Qu.: 77.00
##  Max.   :670.00
```

```r
# Create a scatterplot matrix of the dataset
plot(titanic)
```




```r
# Using str() function we can see the structure of the dataset
str(titanic)
```

```
## 'data.frame':    32 obs. of  5 variables:
##  $ Class   : chr  "1st" "2nd" "3rd" "Crew" ...
##  $ Sex     : chr  "Male" "Male" "Male" "Male" ...
##  $ Age     : chr  "Child" "Child" "Child" "Child" ...
##  $ Survived: chr  "No" "No" "No" "No" ...
##  $ Freq    : int  0 0 35 0 0 0 17 0 118 154 ...
```

Quantitative (Numerical) Variables:

- `Freq: Represents the frequency (number of people)`

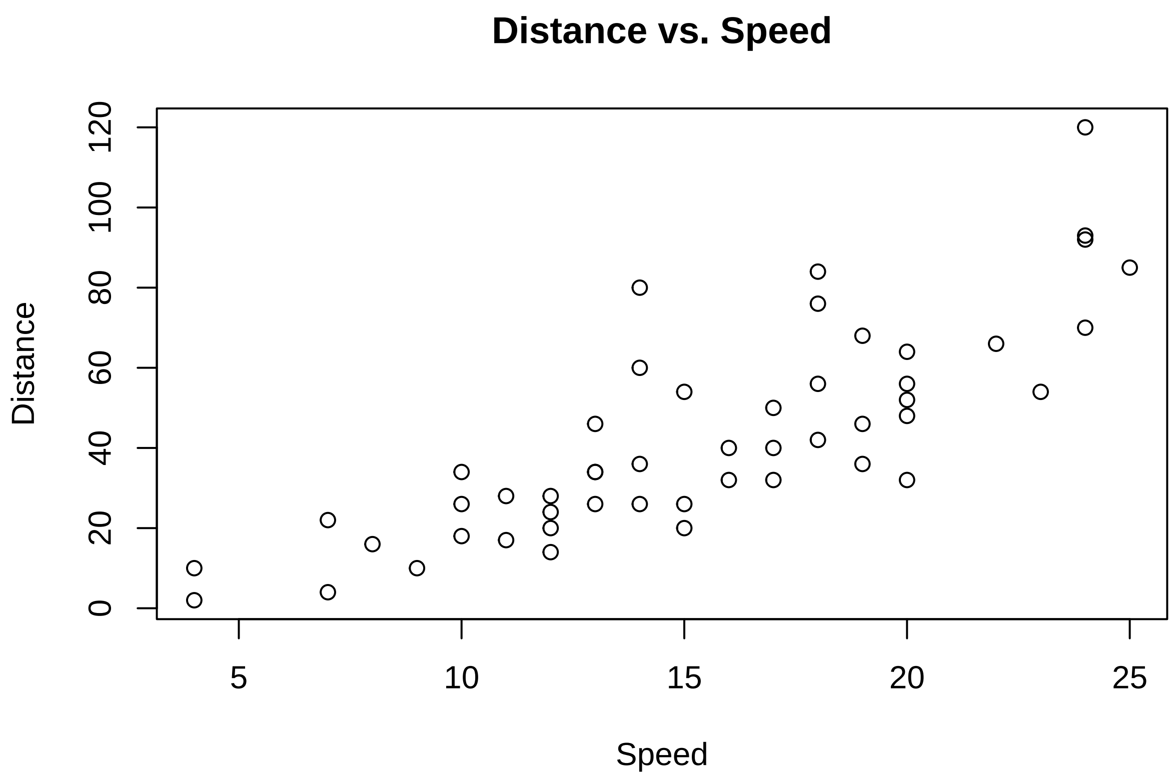
Categorical Variables:

- `X: Row index (treated as categorical, despite being an integer)`
- `Class: Passenger class (e.g., "1st", "2nd", "3rd", "Crew")`
- `Sex: Gender (e.g., "Male", "Female")`
- `Age: Age group (e.g., "Child", "Adult")`
- `Survived: Survival status (e.g., "Yes", "No")`

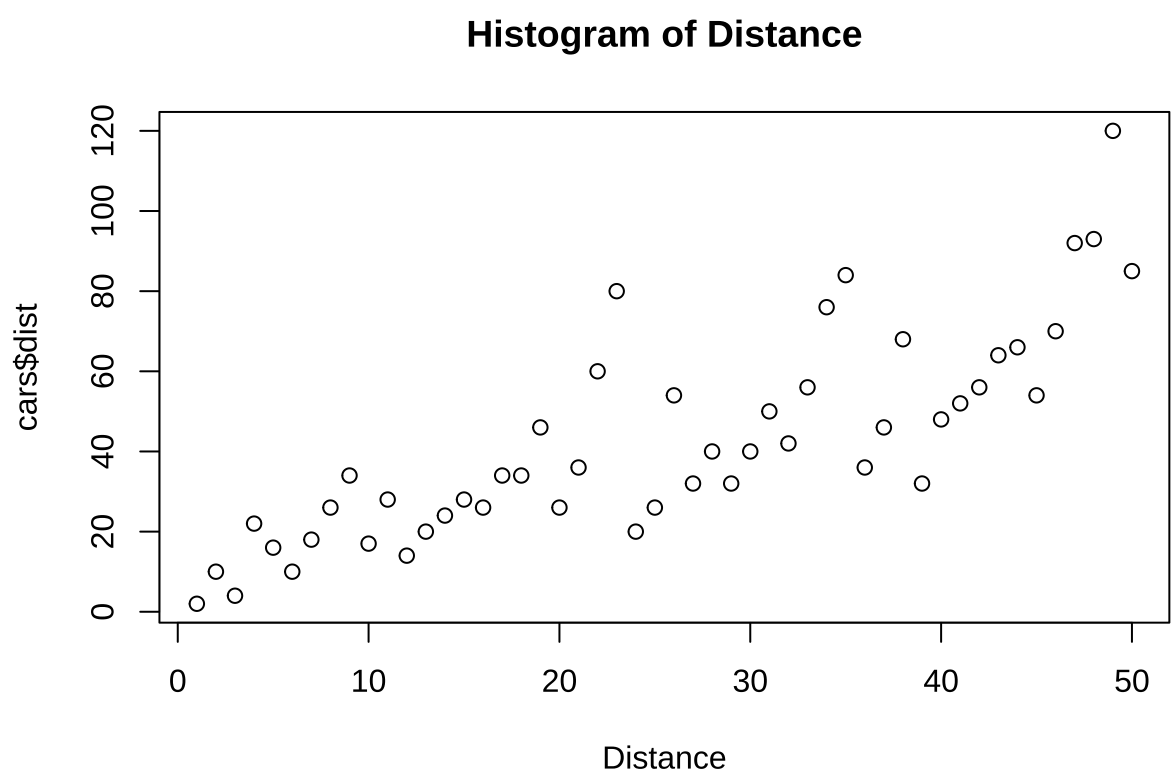
EXERCISE 2

```r
# Load the dataset
cars <- read.csv("./datasets/cars.csv", header = TRUE, sep = ",")
```

```r
# Make a plot of the distance field in terms of the speed field
plot(cars$speed, cars$dist, xlab = "Speed", ylab = "Distance",
     main = "Distance vs. Speed")
```

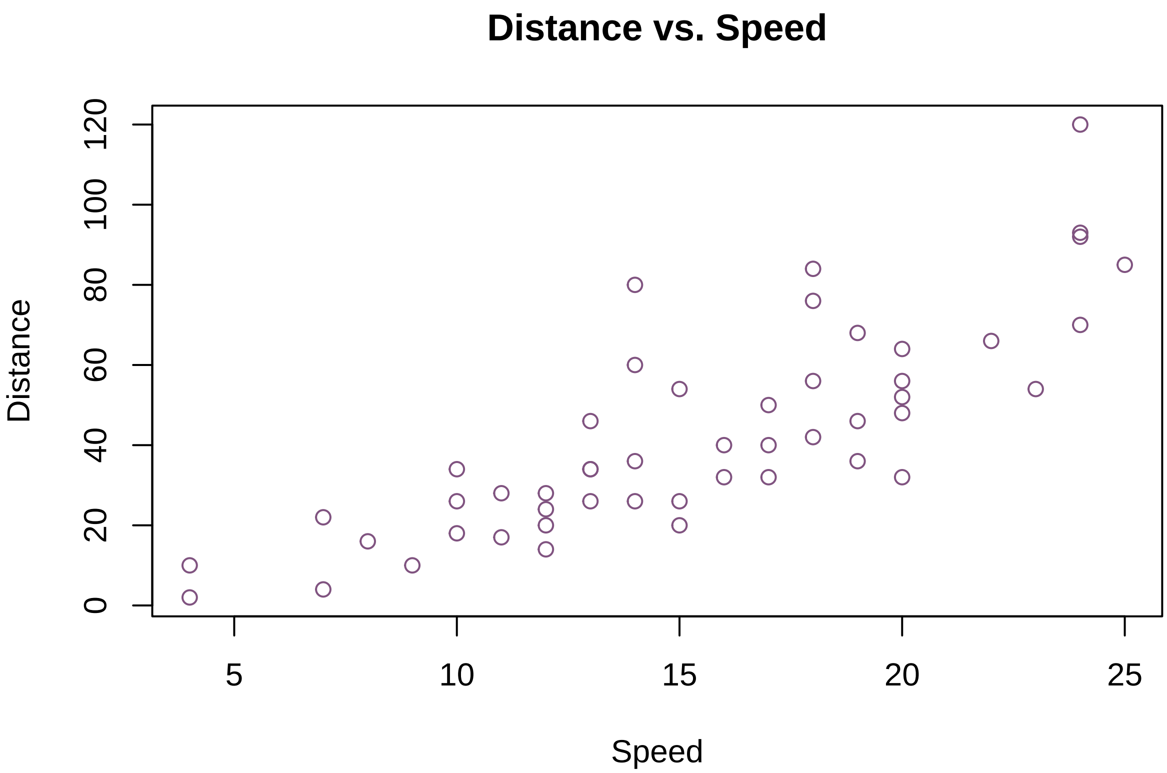



```r
# Create a histogram of the distance field
plot(cars$dist, xlab = "Distance", main = "Histogram of Distance")
```




```r
pdf("./documents/exercice2/distance_vs_speed_plot.pdf")
```

```r
# Create a scatterplot of the speed field
plot(cars$speed, cars$dist,
     xlab = "Speed",
     ylab = "Distance",
     main = "Distance vs. Speed",
     col = "#805380")
```

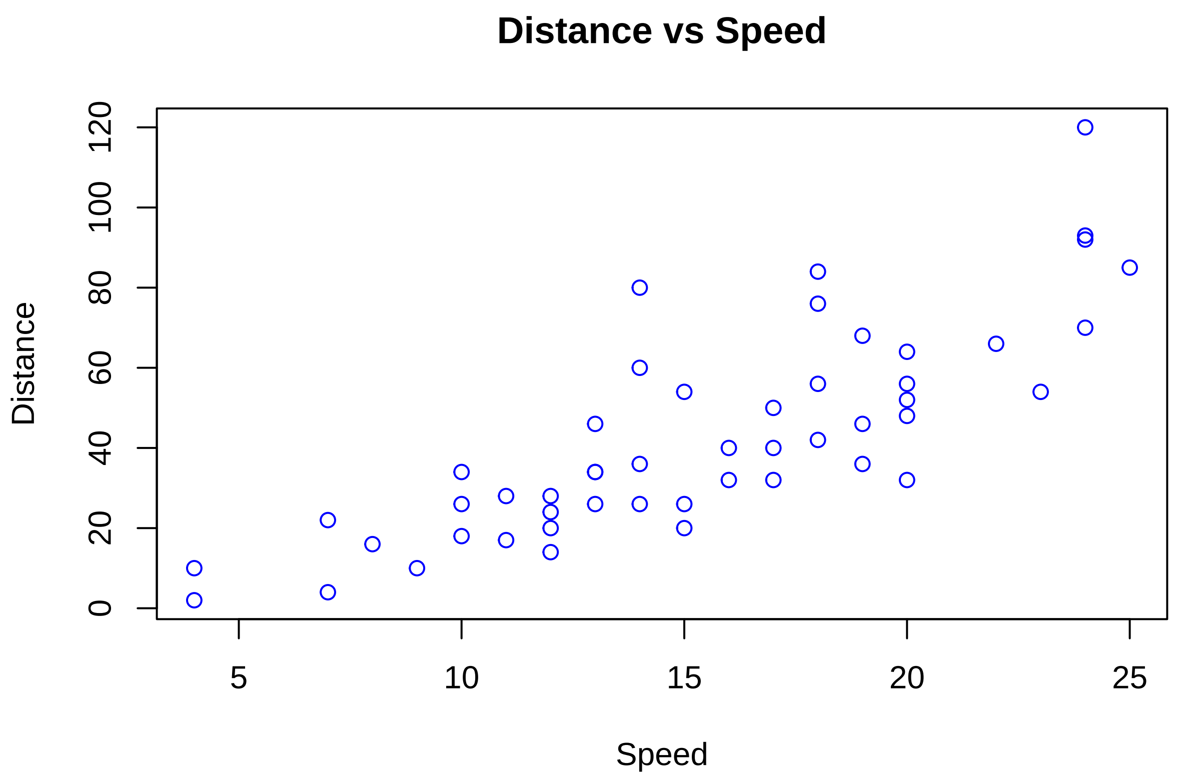



```r
dev.off()
```

```
## pdf
##   3
```

```r
# Save the scatterplot to a PDF file
pdf("./documents/exercice2/modified_distance_vs_speed_plot.pdf")
```

```r
# Scatterplot of Distance vs Speed with modified title, axis labels, and color
plot(cars$speed, cars$dist,
     xlab = "Speed",
     ylab = "Distance",
     main = "Distance vs Speed",
     col = "blue")
```

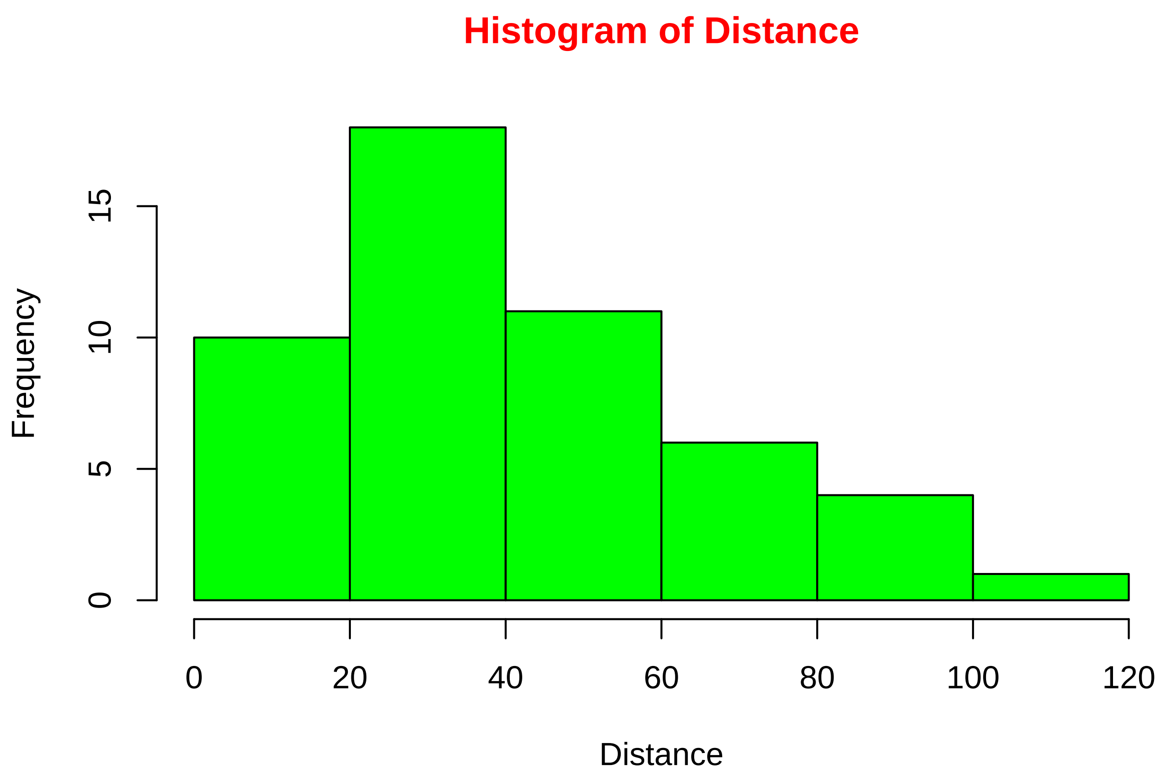



```
dev.off()
```

```
## pdf
##   3
```

```
# Save the histogram of the distance field to a PDF file
pdf("./documents/exercice2/ex2_histogram_distance.pdf")
```

```
# Histogram of Distance with modified title, axis labels, and color
hist(cars$dist,
     xlab = "Distance",
     main = "Histogram of Distance",
     col = "green",
     col.main = "red")
```

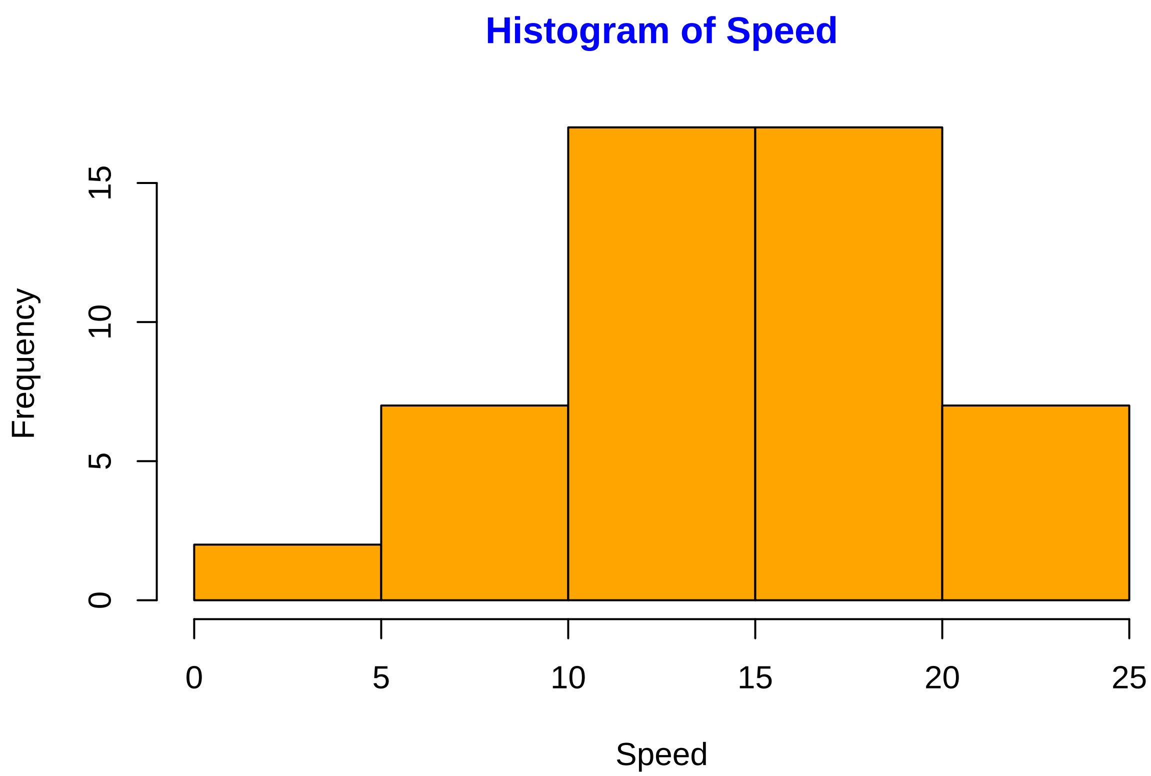



```r
dev.off()
```

```
## pdf
##   3
```

```r
# Save the histogram of the speed field to a PDF file
pdf("./documents/exercice2/ex2_histogram_speed.pdf")
```

```r
# Histogram of Speed with modified title, axis labels, and color
hist(cars$speed,
     xlab = "Speed",
     main = "Histogram of Speed",
     col = "orange",
     col.main = "blue")
```




```r
dev.off()
```

```
## pdf
##   3
```

EXERCISE 3

```r
# Load the dataset
cars <- read.csv("./datasets/cars.csv", header = TRUE, sep = ",")
```

```r
# Remove the first column of the cars data frame
cars <- cars[, -1]
```

```r
# Construct a new data frame
new_cars <- data.frame(speed = c(21, 34), dist = c(47, 87))
```

```r
# Add the constructed data frame to the cars data frame
cars <- rbind(cars, new_cars)
```

```r
# Sort the data in the resulting dataset by column speed (ascending)
cars <- cars[order(cars$speed), ]
```

```r
# Write the resulting dataset to a CSV file
write.csv(cars, file = "./datasets/cars_sorted.csv", row.names = FALSE)
```

EXERCISE 4

```r
# Load the dataset
airquality <- read.csv("./datasets/airquality.csv", header = TRUE, sep = ",")
```

```r
# Display the first two rows of the dataset
print(airquality[1:2, ])
```

```
##   Ozone Solar.R Wind Temp Month Day
## 1    41     190  7.4   67     5   1
## 2    36     118  8.0   72     5   2
```

```r
# How many rows are in the dataset?
nrow(airquality)
```

```
## [1] 153
```

```r
# What is the value of Ozone in the 40th row?
airquality[40, "Ozone"]
```

```
## [1] 71
```

```r
# How many missing values are there in the Ozone column?
sum(is.na(airquality$Ozone))
```

```
## [1] 37
```

```r
# What is the mean of the Ozone column in this dataset? Exclude NA values
airquality <- read.csv("./datasets/airquality.csv", header = TRUE, sep = ",")
ozone_clean <- na.omit(airquality$Ozone)
print(mean(ozone_clean))
```

```
## [1] 42.12931
```

```r
# Extract the rows where the Ozone value is greater than 31
# and Temp value is greater than 90
airquality <- read.csv("./datasets/airquality.csv", header = TRUE, sep = ",")
airquality <- na.omit(airquality)
airquality_subset <- airquality[airquality$Ozone > 31 & airquality$Temp > 90, ]

# What is the mean of Solar.R in this subset?
print(mean(airquality_subset$Solar.R))
```

```
## [1] 212.8
```

EXERCISE 5

```r
# Discretize the Ozone column into 5 bins
aux <- cut(airquality$Ozone,
           breaks = 5,
           labels = c("bin1", "bin2", "bin3", "bin4", "bin5"))
# Add NA to the levels
aux <- addNA(aux)
print(aux)
```

```
##   [1] bin2 bin2 bin1 bin1 bin1 bin1 bin1 bin1 bin1 bin1 bin1 bin1 bin1 bin1 bin1
##  [16] bin1 bin1 bin1 bin1 bin1 bin1 bin2 bin4 bin2 bin1 bin3 bin2 bin1 bin1 bin2
##  [31] bin1 bin1 bin1 bin5 bin2 bin1 bin2 bin2 bin3 bin3 bin3 bin3 bin1 bin1 bin1
##  [46] bin2 bin2 bin2 bin3 bin2 bin1 bin3 bin4 bin1 bin2 bin3 bin2 bin2 bin2 bin2
##  [61] bin1 bin1 bin4 bin3 bin4 bin2 bin1 bin2 bin1 bin2 bin1 bin1 bin2 bin1 bin1
##  [76] bin2 bin5 bin3 bin3 bin4 bin3 bin3 bin3 bin3 bin3 bin3 bin2 bin1 bin1 bin1
##  [91] bin1 bin1 bin2 bin1 bin1 bin1 bin1 bin2 bin1 bin1 bin1 bin1 bin1 bin1 bin2
```

```
## [106] bin1 bin1 bin1 bin1 bin1 bin1
## Levels: bin1 bin2 bin3 bin4 bin5 <NA>
```

```r
# Discretize the Solar.R column into 4 bins
aux <- cut(airquality$Solar.R,
           breaks = 4,
           labels = c("bin1", "bin2", "bin3", "bin4"))

# Add NA to the levels
aux <- addNA(aux)
print(aux)
```

```
##   [1] bin3 bin2 bin2 bin4 bin4 bin2 bin1 bin4 bin4 bin4 bin1 bin4 bin4 bin1 bin4
##  [16] bin1 bin1 bin4 bin1 bin2 bin1 bin3 bin3 bin4 bin2 bin4 bin4 bin2 bin3 bin4
##  [31] bin1 bin2 bin2 bin4 bin3 bin3 bin3 bin4 bin4 bin4 bin4 bin3 bin4 bin3 bin1
##  [46] bin4 bin4 bin4 bin3 bin3 bin1 bin4 bin3 bin1 bin1 bin3 bin4 bin4 bin4 bin1
##  [61] bin1 bin1 bin4 bin3 bin3 bin3 bin4 bin2 bin1 bin1 bin2 bin3 bin3 bin4 bin1
##  [76] bin3 bin3 bin3 bin3 bin3 bin3 bin3 bin2 bin3 bin3 bin3 bin2 bin2 bin3 bin3
##  [91] bin3 bin4 bin3 bin4 bin3 bin1 bin2 bin3 bin3 bin1 bin3 bin3 bin3 bin1 bin2
## [106] bin1 bin1 bin3 bin3 bin2 bin3
## Levels: bin1 bin2 bin3 bin4 <NA>
```

```r
# Load the dataset
airquality <- read.csv("./datasets/airquality.csv", header = TRUE, sep = ",")

# Create a new column called "cumulative_days"
cumulative_days <- c(0, 31, 61, 92, 123)

# Add the new column to the dataset and adjust the index (May is month 5)
airquality$AbsDay <- airquality$Day + cumulative_days[airquality$Month - 4]

# Display the updated dataset
head(airquality)
```

```
##   Ozone Solar.R Wind Temp Month Day AbsDay
## 1    41     190  7.4   67     5   1      1
## 2    36     118  8.0   72     5   2      2
## 3    12     149 12.6   74     5   3      3
## 4    18     313 11.5   62     5   4      4
## 5    NA      NA 14.3   56     5   5      5
## 6    28      NA 14.9   66     5   6      6
```

EXERCISE 6

```r
# Load the dataset
titanic <- read.csv("./datasets/titanic.csv", header = TRUE, sep = ",")
titanic$Class <- as.numeric(factor(titanic$Class))

# Numerize the Class column
print(titanic$Class <- as.numeric(titanic$Class))
```

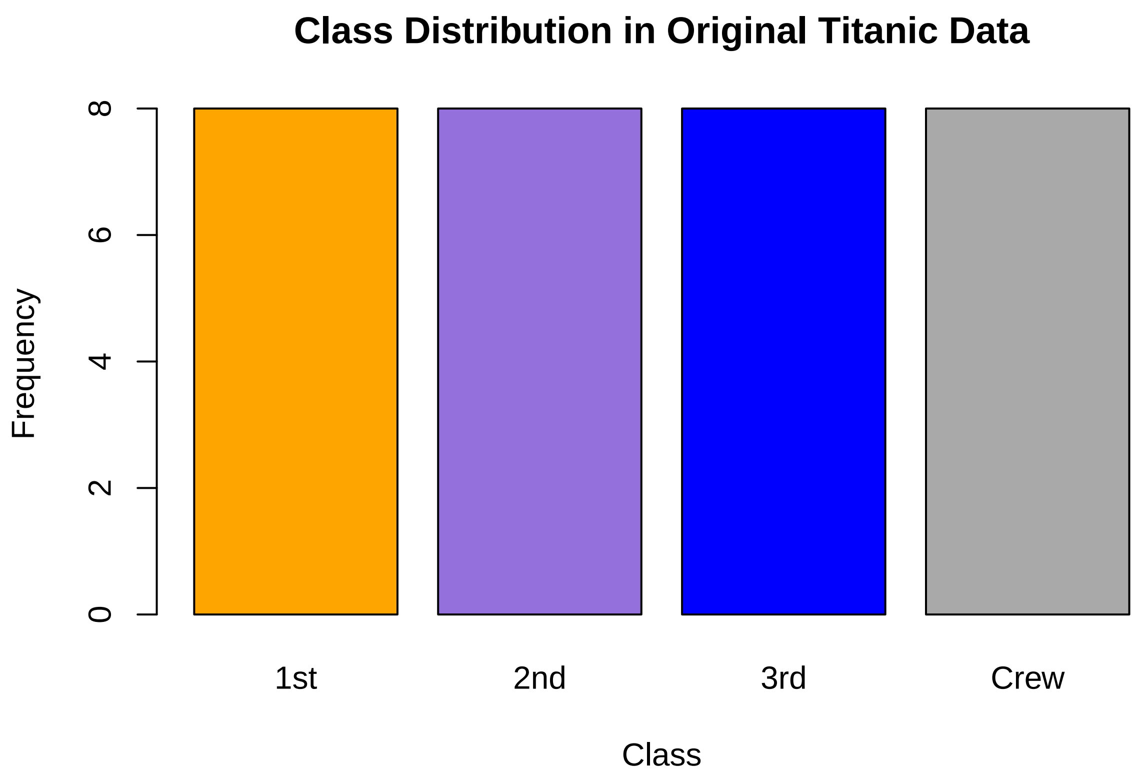
```
##  [1] 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4
```

```r
# Load the dataset
titanic <- read.csv("./datasets/titanic.csv", header = TRUE, sep = ",")
```

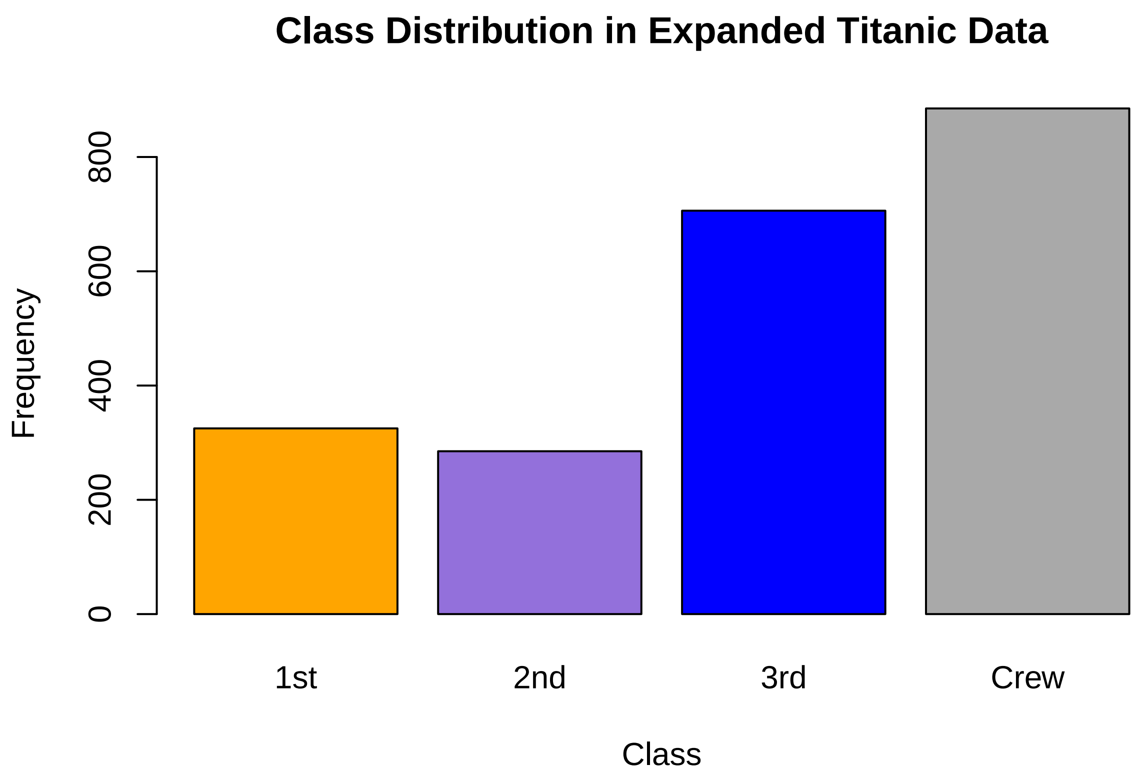
```r
# Create a new data frame (titanic2) by expanding rows based on the Freq column
titanic2 <- titanic[rep(seq_len(nrow(titanic)), titanic$Freq), ]
head(titanic2)
```

```
##     X Class  Sex   Age Survived Freq
## 3   3   3rd Male Child       No   35
## 3.1 3   3rd Male Child       No   35
## 3.2 3   3rd Male Child       No   35
## 3.3 3   3rd Male Child       No   35
## 3.4 3   3rd Male Child       No   35
## 3.5 3   3rd Male Child       No   35
```

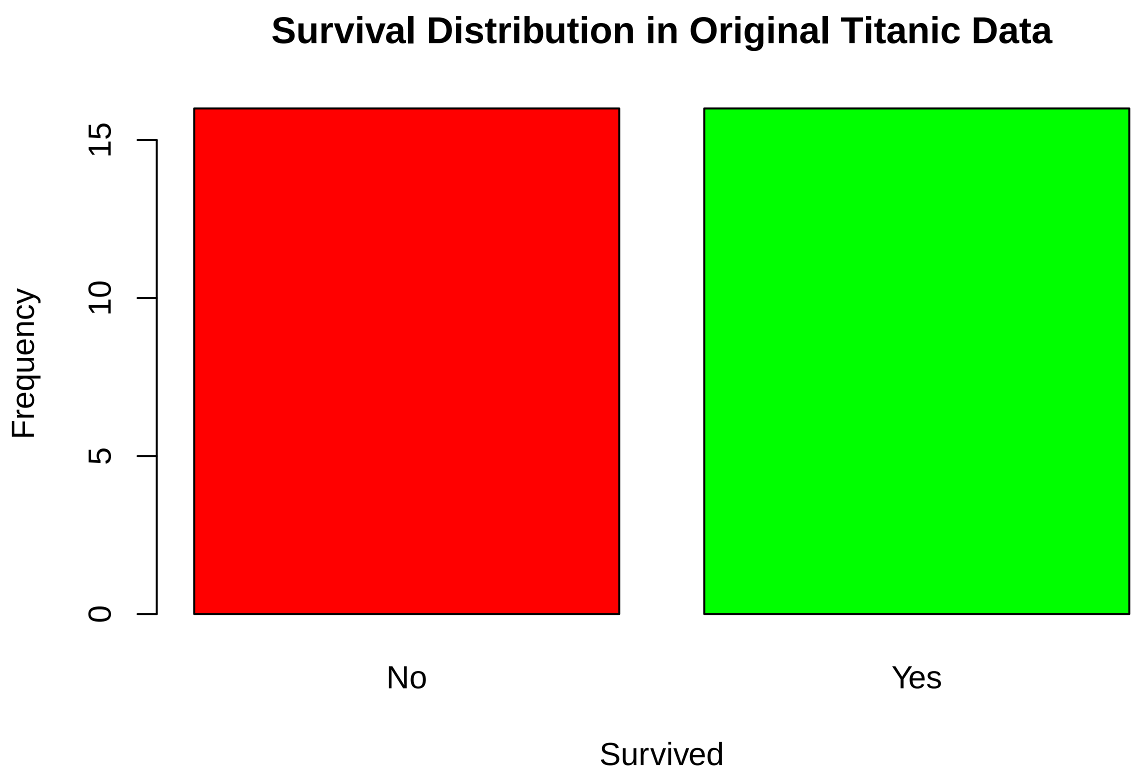
```r
# Load the dataset
titanic <- read.csv("./datasets/titanic.csv", header = TRUE, sep = ",")

# Define the colors for the bar plots
colors <- c("orange", "#9370DB", "blue", "darkgrey")

# Plot distribution of Class in the original dataset with multiple colors
barplot(table(titanic$Class),
        main = "Class Distribution in Original Titanic Data",
        xlab = "Class",
        ylab = "Frequency",
        col = colors)

# Plot distribution of Class in the original dataset
barplot(table(titanic$Class),
        main = "Class Distribution in Original Titanic Data",
        xlab = "Class",
        ylab = "Frequency",
        col = colors)
```



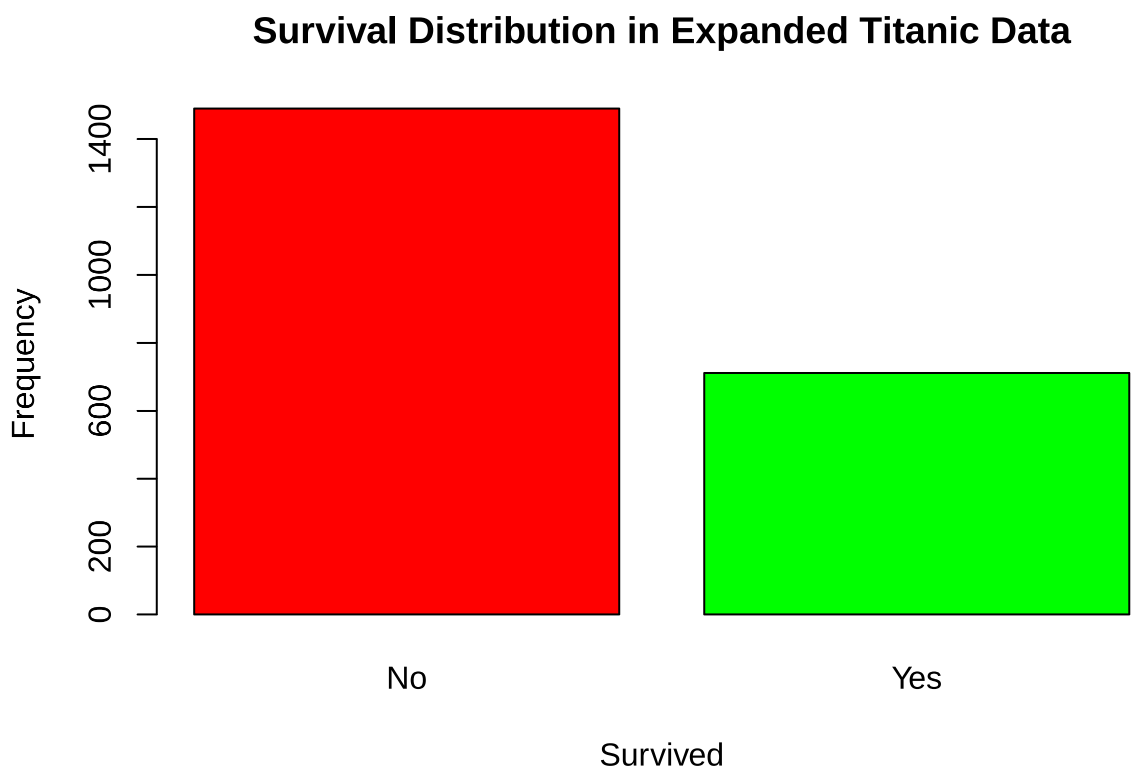

```
# Plot distribution of Class in the new dataset (titanic2)
barplot(table(titanic2$Class),
        main = "Class Distribution in Expanded Titanic Data",
        xlab = "Class",
        ylab = "Frequency",
        col = colors)
```




```r
# Plot distribution of Survival in the original dataset
barplot(table(titanic$Survived),
        main = "Survival Distribution in Original Titanic Data",
        xlab = "Survived",
        ylab = "Frequency",
        col = c("red", "green"))
```




```r
# Plot distribution of Survival in the new dataset (titanic2)
barplot(table(titanic2$Survived),
        main = "Survival Distribution in Expanded Titanic Data",
        xlab = "Survived",
        ylab = "Frequency",
        col = c("red", "green"))
```




```r
# Correlation between Class and Survival in the original dataset
# Create a contingency table
class_survived_table <- table(titanic$Class, titanic$Survived)

# Calculate survival rates by class
survival_rates <- prop.table(class_survived_table, margin = 1)

# Print the contingency table and survival rates
print(class_survived_table)
```

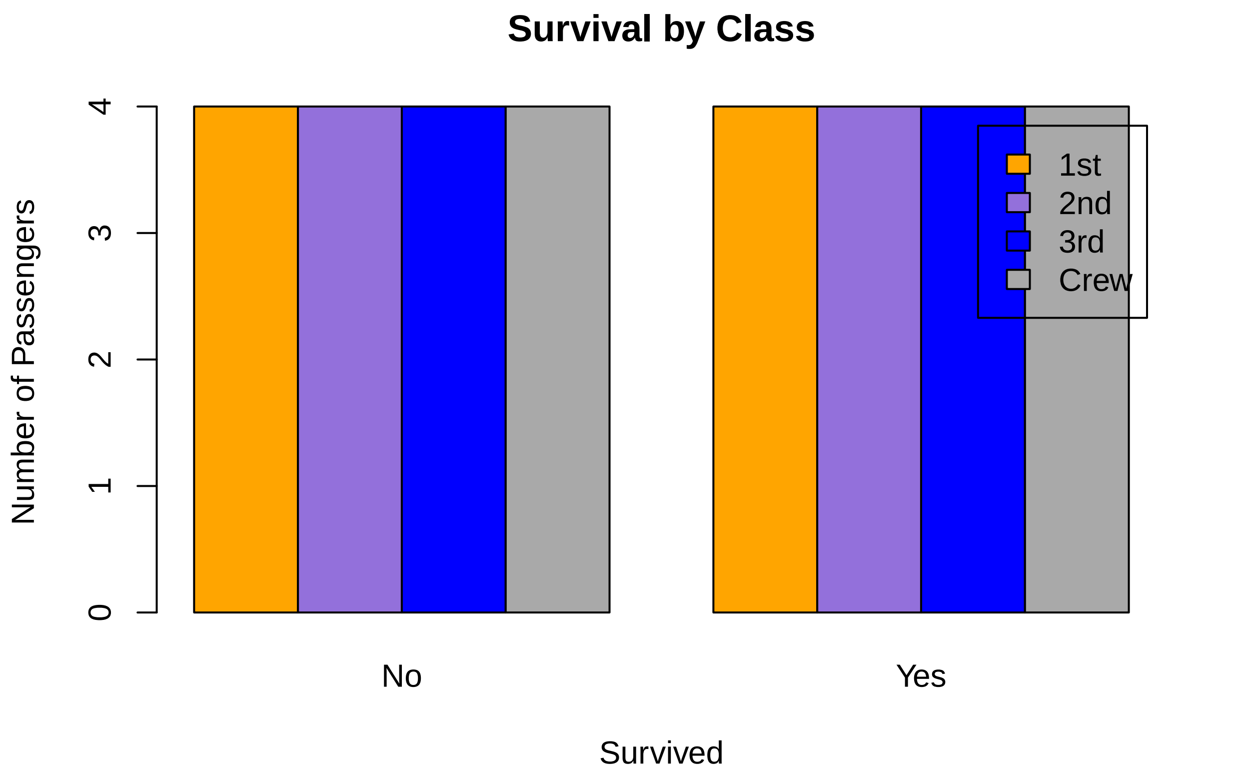
```
##
##         No Yes
##    1st   4   4
##    2nd   4   4
##    3rd   4   4
##    Crew  4   4
```

```r
print(survival_rates)
```

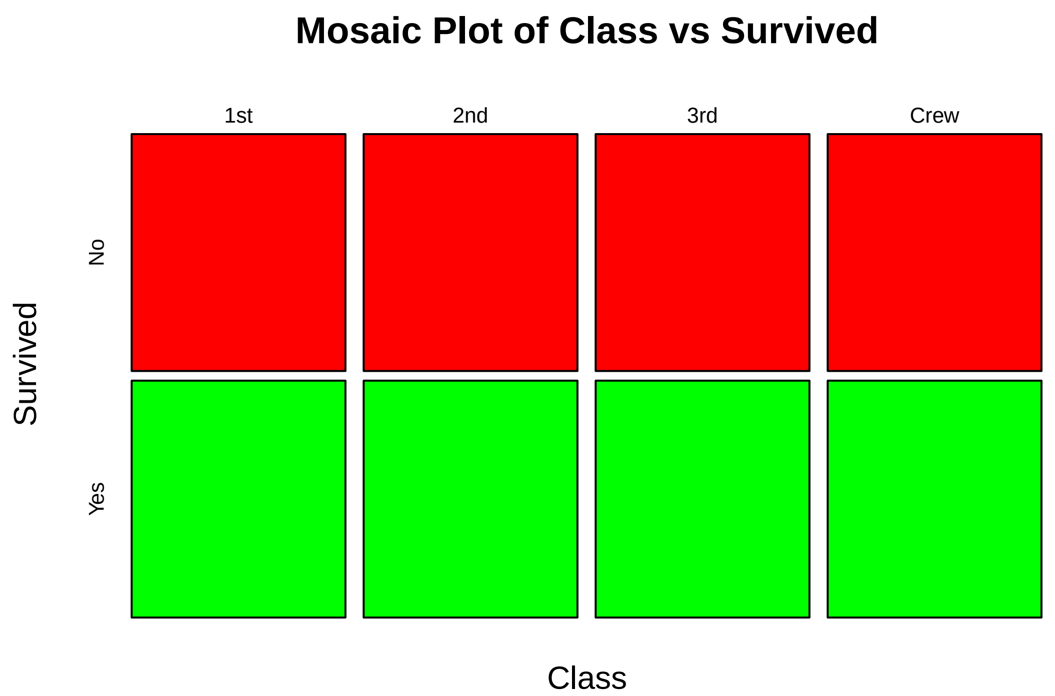
```
##
##          No Yes
##    1st  0.5 0.5
##    2nd  0.5 0.5
##    3rd  0.5 0.5
##    Crew 0.5 0.5
```

```r
# Visualizing the relationship

# Bar plot of survival by class
barplot(class_survived_table,
        beside = TRUE,
```

```
        col = colors,
        legend = rownames(class_survived_table),
        main = "Survival by Class",
        xlab = "Survived",
        ylab = "Number of Passengers")
```




```
# Mosaic plot of Class vs Survived
mosaicplot(~ Class + Survived,
           data = titanic,
           col = c("red", "green"),
           main = "Mosaic Plot of Class vs Survived")
```




```r
# Correlation between Class and Survival in the new dataset (titanic2)

# Create a contingency table for titanic2
class_survived_table2 <- table(titanic2$Class, titanic2$Survived)

# Calculate survival rates by class for titanic2
survival_rates2 <- prop.table(class_survived_table2, margin = 1)

# Print the contingency table and survival rates for titanic2
print(class_survived_table2)
```

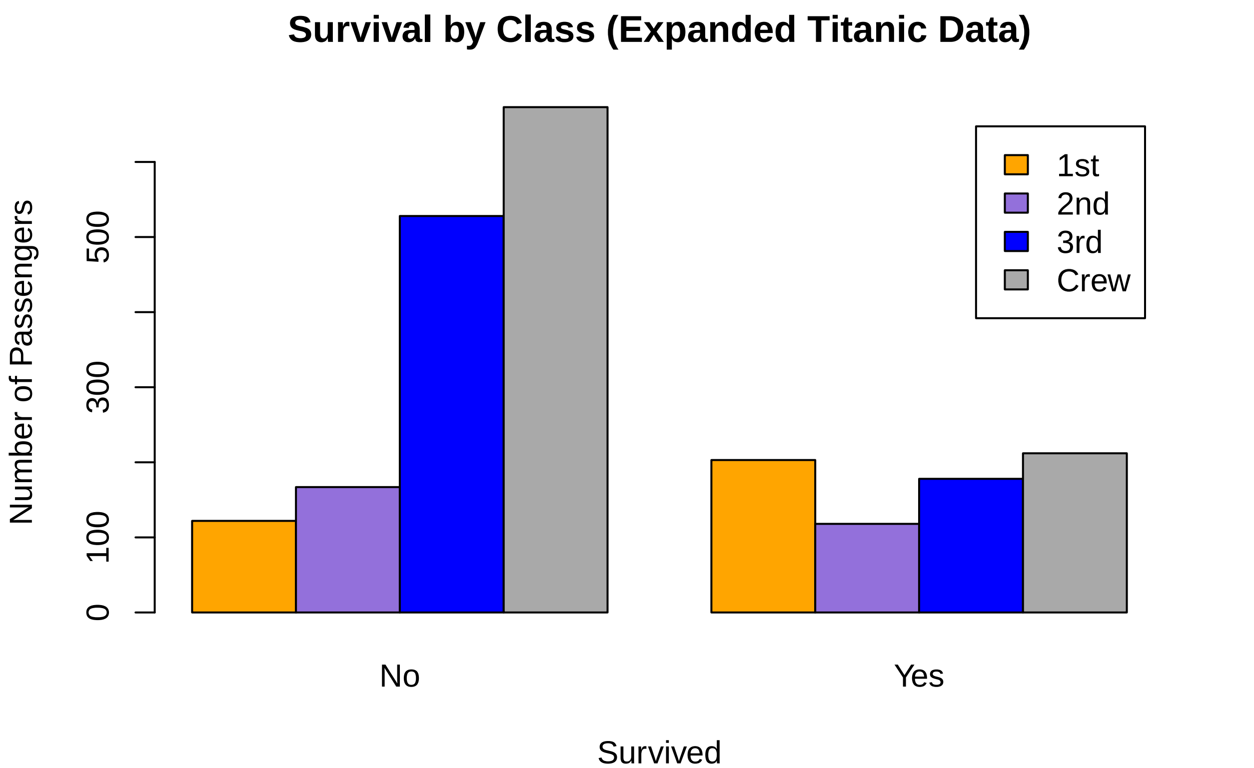
```
##
##         No Yes
##   1st  122 203
##   2nd  167 118
##   3rd  528 178
##   Crew 673 212
```

```r
print(survival_rates2)
```

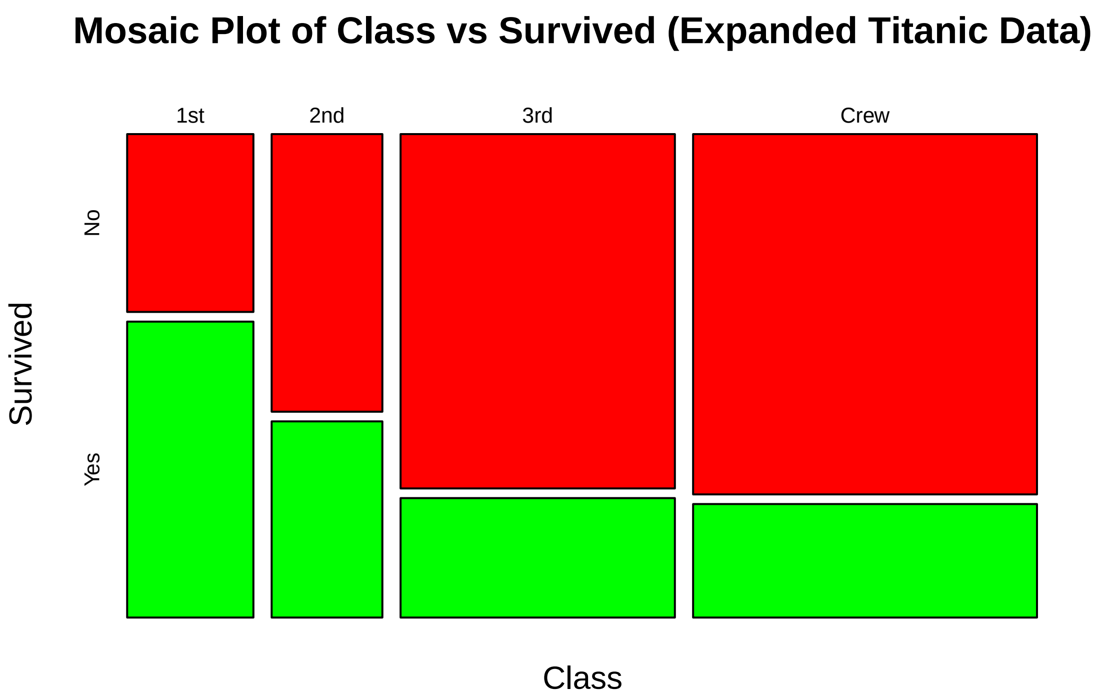
```
##
##               No       Yes
##   1st  0.3753846 0.6246154
##   2nd  0.5859649 0.4140351
##   3rd  0.7478754 0.2521246
##   Crew 0.7604520 0.2395480
```

```r
# Visualizing the relationship in titanic2

# Bar plot of survival by class in titanic2
barplot(class_survived_table2,
        beside = TRUE,
        col = colors,
        legend = rownames(class_survived_table2),
```

```
        main = "Survival by Class (Expanded Titanic Data)",
        xlab = "Survived",
        ylab = "Number of Passengers")
```




```
# Mosaic plot of Class vs Survived in titanic2
mosaicplot(~ Class + Survived,
           data = titanic2,
           col = c("red", "green"),
           main = "Mosaic Plot of Class vs Survived (Expanded Titanic Data)")
```



EXERCISE 7

```
# Load the Airquality dataset
airquality <- read.csv("./datasets/airquality.csv", header = TRUE, sep = ",")

# Clean the dataset
airquality <- na.omit(airquality)

# Calculate the correlation matrix
print(correlation_matrix <- cor(airquality))
```

```
##                Ozone     Solar.R        Wind       Temp        Month
## Ozone    1.000000000  0.34834169 -0.61249658  0.6985414  0.142885168
## Solar.R  0.348341693  1.00000000 -0.12718345  0.2940876 -0.074066683
## Wind    -0.612496576 -0.12718345  1.00000000 -0.4971897 -0.194495804
## Temp     0.698541410  0.29408764 -0.49718972  1.0000000  0.403971709
## Month    0.142885168 -0.07406668 -0.19449580  0.4039717  1.000000000
## Day     -0.005189769 -0.05775380  0.04987102 -0.0965458 -0.009001079
##                  Day
## Ozone   -0.005189769
## Solar.R -0.057753801
## Wind     0.049871017
## Temp    -0.096545800
## Month   -0.009001079
## Day      1.000000000
```

```
# High correlation between Ozone and Temp (0.6985414)
# Low correlation between Wind and Temp (-0.4579883)
```

```
# Load the Cars dataset
cars <- read.csv("./datasets/cars.csv", header = TRUE, sep = ",")
```

```r
# Clean the dataset
cars <- na.omit(cars)

# Calculate the correlation matrix
print(correlation_matrix <- cor(cars))
```

```
##               X     speed      dist
## X     1.0000000 0.9854590 0.8176576
## speed 0.9854590 1.0000000 0.8068949
## dist  0.8176576 0.8068949 1.0000000
```

```r
# High correlation between all variables
```

```r
# Perform a simple random sampling of 50 examples.

# Load the Airquality dataset
airquality <- read.csv("./datasets/airquality.csv", header = TRUE, sep = ",")

# Clean the dataset
airquality <- na.omit(airquality)

# Perform simple random sampling
airquality <- airquality[sample(nrow(airquality), 50), ]
head(airquality)
```

```
##     Ozone Solar.R Wind Temp Month Day
## 134    44     236 14.9   81     9  11
## 101   110     207  8.0   90     8   9
## 94      9      24 13.8   81     8   2
## 152    18     131  8.0   76     9  29
## 118    73     215  8.0   86     8  26
## 13     11     290  9.2   66     5  13
```

```r
# Perform a stratified random sampling of 5 examples each month.
library(dplyr)
```

```
##
## Adjuntando el paquete: 'dplyr'

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

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

```r
# Load the Airquality dataset
airquality <- read.csv("./datasets/airquality.csv", header = TRUE, sep = ",")

# Clean the dataset
airquality <- na.omit(airquality)

# Perform stratified random sampling
airquality <- airquality %>%
  group_by(Month) %>%
  sample_n(5)
```

```r
head(airquality)
```

```
## # A tibble: 6 x 6
## # Groups:   Month [2]
##   Ozone Solar.R  Wind  Temp Month   Day
##   <int>   <int> <dbl> <int> <int> <int>
## 1    11     290   9.2    66     5    13
## 2    23     299   8.6    65     5     7
## 3   115     223   5.7    79     5    30
## 4    45     252  14.9    81     5    29
## 5    14     334  11.5    64     5    16
## 6    39     323  11.5    87     6    10
```

EXERCISE 8

```r
# Load the dataset
sales <- read.table("./datasets/sales.txt", header = TRUE, sep = "")

# Calculate the total sales per store using Aggregate()
total_sales_per_store <- aggregate(sales_amount ~ store, data = sales, sum)

total_sales_per_store
```

```
##     store sales_amount
## 1 Store_A         2430
## 2 Store_B         2980
## 3 Store_C         1820
```

```r
# Load the dataset
sales <- read.table("./datasets/sales.txt", header = TRUE, sep = "")

# Find the avg sales amount for each product category
avg_sales_per_category <- aggregate(sales_amount ~ product_category,
                                    data = sales,
                                    mean)

avg_sales_per_category
```

```
##   product_category sales_amount
## 1         Clothing      660.000
## 2      Electronics     1463.333
## 3        Groceries      430.000
```

```r
pdf("./documents/exercice8/ex8_total_sales_per_store_category_matrix.pdf")
```

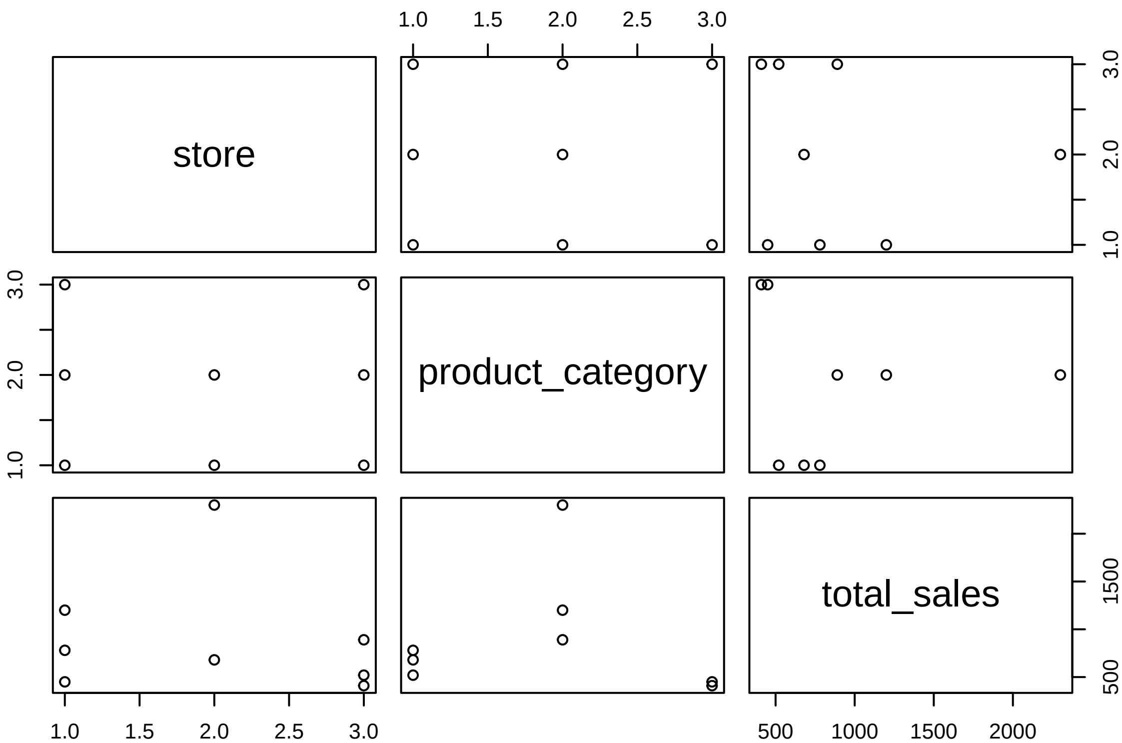
```r
library(dplyr)
library(gridExtra)
```

```
##
## Adjuntando el paquete: 'gridExtra'

## The following object is masked from 'package:dplyr':
##
##     combine
```

```r
# Load the dataset
sales <- read.table("./datasets/sales.txt", header = TRUE, sep = "")
str(sales)
```

```
## 'data.frame':    8 obs. of  3 variables:
##  $ store           : chr  "Store_A" "Store_A" "Store_A" "Store_B" ...
##  $ product_category: chr  "Electronics" "Groceries" "Clothing" "Electronics" ...
##  $ sales_amount    : int  1200 450 780 2300 680 410 520 890
```

```r
# Group the data by store and product category, calculate the total sales
total_sales_per_store_category <- sales %>%
  group_by(store, product_category) %>%
  summarise(total_sales = sum(sales_amount), .groups = "drop")

plot(total_sales_per_store_category)
```




```r
dev.off()
```

```
## pdf
##   3
```

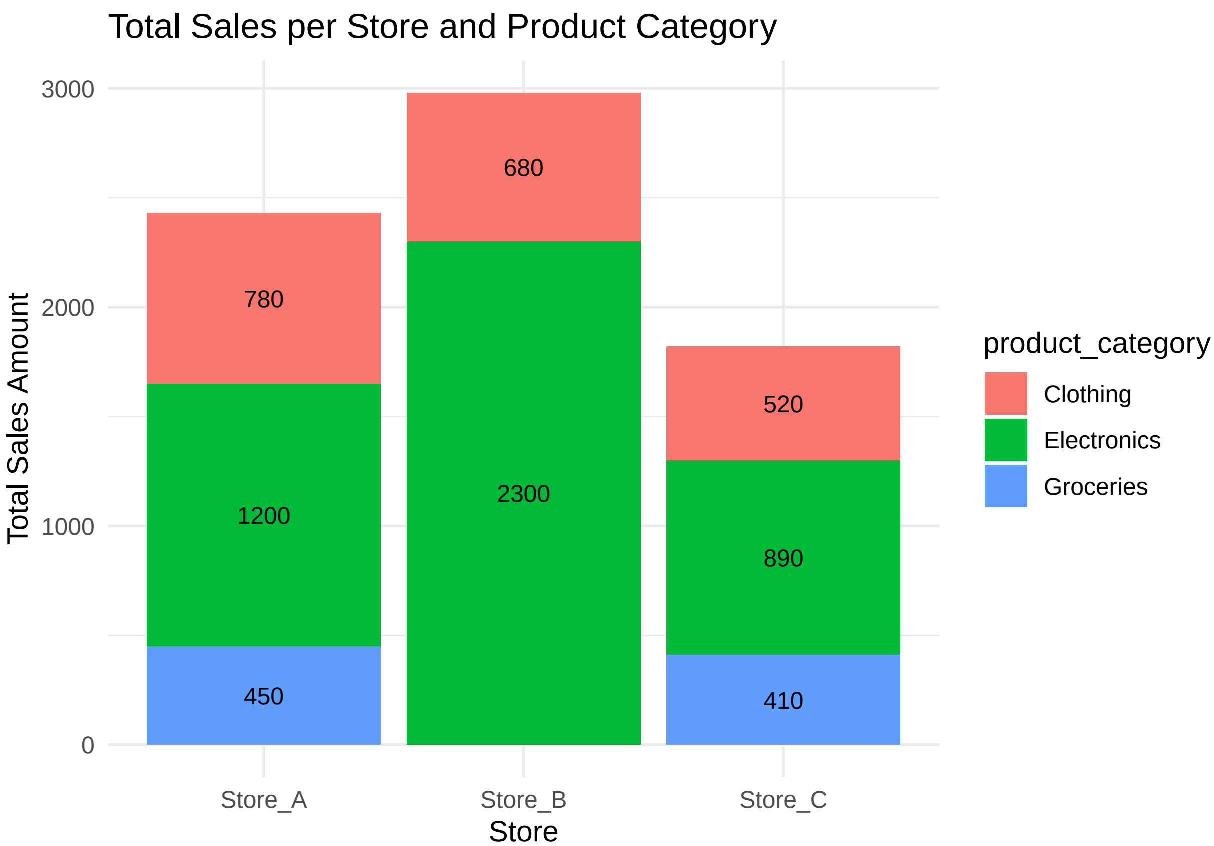
```r
pdf("./documents/exercice8/ex8_total_sales_per_store_category_stacked.pdf")
```

```r
library(dplyr)
library(ggplot2)

# Load the dataset
sales <- read.table("./datasets/sales.txt", header = TRUE, sep = "")
str(sales)
```

```
## 'data.frame':    8 obs. of  3 variables:
##  $ store           : chr  "Store_A" "Store_A" "Store_A" "Store_B" ...
##  $ product_category: chr  "Electronics" "Groceries" "Clothing" "Electronics" ...
```

```
##  $ sales_amount    : int  1200 450 780 2300 680 410 520 890
```

```r
# Group the data by store and product category, calculate the total sales
total_sales_per_store_category <- sales %>%
  group_by(store, product_category) %>%
  summarise(total_sales = sum(sales_amount), .groups = "drop")

# Create the stacked bar plot
ggplot(total_sales_per_store_category, aes(x = store, y = total_sales, fill = product_category)) +
  geom_bar(stat = "identity") +
  geom_text(aes(label = total_sales),
            position = position_stack(vjust = 0.5),
            size = 3) +
  labs(title = "Total Sales per Store and Product Category",
       x = "Store",
       y = "Total Sales Amount") +
  theme_minimal()
```




```r
dev.off()
```

```
## pdf
##   3
```

EXERCISE 9

```r
library(dplyr)
```

```
# Load the datasets
customers <- read.table("./datasets/customers.txt", header = TRUE, sep = "")
orders <- read.table("./datasets/orders.txt", header = TRUE, sep = "")

# Merge the two datasets by customer_id
merged_data <- inner_join(customers, orders, by = "customer_id")

# Count the number of unique customers in the merged dataset
num_unique_customers <- merged_data %>%
  distinct(customer_id) %>%
  n_distinct()

# Print the number of unique customers
print(num_unique_customers)
```

```
## [1] 4
```

```
# Count the number of orders placed by each customer
order_counts <- table(merged_data$customer_id)

# Print the number of orders placed by each customer
print(order_counts)
```

```
##
## 101 102 103 104
##   2   1   1   1
```

```
# Save the merged dataset as a new CSV file called "customer_orders.csv."
write.csv(merged_data, file = "./datasets/customer_orders.csv", row.names = FALSE)
```