EDA

Süleyman Erim, Giacomo Schiavo, Mattia Varagnolo

2023-07-22

Introduction to data

This section introduces the purpose of the exploratory data analysis (EDA) and sets up the necessary libraries and data files.

```
# import libraries
library(tidyverse)
library(corrplot)
library(ggplot2)
library(gridExtra)
library(correlation)
library(reshape)
library(reshape2)
```

```
data_train = read.csv("train.csv")
data_test = read.csv("test.csv")

# merge train and test data
data = rbind(data_train, data_test)
attach(data)
```

Introduction

In this project, we will predict whether a passenger will be satisfied or dissatisfied with the services offered by an airline company. The dataset comprises a survey on airline passenger satisfaction.

The main objectives of this project are to identify the factors that have a strong correlation with passenger satisfaction or dissatisfaction and to develop a predictive model for passenger satisfaction.

The dataset: https://www.kaggle.com/datasets/teejmahal20/airline-passenger-satisfaction

All the variables in the dataset are described below: - Gender: Gender of the passengers (Female, Male) - Customer Type: The customer type (Loyal customer, disloyal customer) - Age: The actual age of the passengers - Type of Travel: Purpose of the flight of the passengers (Personal Travel, Business Travel) - Class: Travel class in the plane of the passengers (Business, Eco, Eco Plus) - Flight distance: The flight distance of this journey - Inflight wifi service: Satisfaction level of the inflight wifi service (0:Not Applicable;1-5) - Departure/Arrival time convenient: Satisfaction level of Departure/Arrival time convenient - Ease of Online booking: Satisfaction level of online booking - Gate location: Satisfaction level of Gate location - Food and drink: Satisfaction level of Food and drink - Online boarding: Satisfaction level of online boarding - Seat comfort: Satisfaction level of Seat comfort - Inflight entertainment: Satisfaction level of inflight entertainment - On-board service: Satisfaction level of On-board service - Leg room service: Satisfaction level of Leg room service - Baggage handling: Satisfaction level of baggage handling - Check-in service: Satisfaction level of Check-in service - Inflight service: Satisfaction level of inflight service - Cleanliness: Satisfaction level of

Cleanliness - Departure Delay in Minutes: Minutes delayed when departure - Arrival Delay in Minutes: Minutes delayed when Arrival - Satisfaction: Airline satisfaction level(Satisfaction, neutral or dissatisfaction)

The objective of our report is to predict passenger satisfaction with airline services based on the provided dataset, which includes various demographic and satisfaction-related variables such as gender, age, travel type, flight class, and satisfaction levels with different aspects of the journey. The dataset represents a survey on airline passenger satisfaction and will be used to develop a predictive model to determine whether passengers will be satisfied or dissatisfied with the airline services.

Now we're going to get a summary of all the features in our dataset:

```r
summary(data)
```

```
##        X                id            Gender          Customer.Type     
##  Min.   :     0   Min.   :     1   Length:129880      Length:129880     
##  1st Qu.: 16235   1st Qu.: 32471   Class :character   Class :character  
##  Median : 38964   Median : 64941   Mode  :character   Mode  :character  
##  Mean   : 44159   Mean   : 64941                                        
##  3rd Qu.: 71433   3rd Qu.: 97410                                        
##  Max.   :103903   Max.   :129880                                        
##                                                                         
##       Age        Type.of.Travel        Class           Flight.Distance
##  Min.   : 7.00   Length:129880      Length:129880      Min.   :  31   
##  1st Qu.:27.00   Class :character   Class :character   1st Qu.: 414   
##  Median :40.00   Mode  :character   Mode  :character   Median : 844   
##  Mean   :39.43                                         Mean   :1190   
##  3rd Qu.:51.00                                         3rd Qu.:1744   
##  Max.   :85.00                                         Max.   :4983   
##                                                                       
##  Inflight.wifi.service Departure.Arrival.time.convenient Ease.of.Online.booking
##  Min.   :0.000         Min.   :0.000                     Min.   :0.000         
##  1st Qu.:2.000         1st Qu.:2.000                     1st Qu.:2.000         
##  Median :3.000         Median :3.000                     Median :3.000         
##  Mean   :2.729         Mean   :3.058                     Mean   :2.757         
##  3rd Qu.:4.000         3rd Qu.:4.000                     3rd Qu.:4.000         
##  Max.   :5.000         Max.   :5.000                     Max.   :5.000         
##                                                                                
##  Gate.location   Food.and.drink  Online.boarding  Seat.comfort  
##  Min.   :0.000   Min.   :0.000   Min.   :0.000   Min.   :0.000  
##  1st Qu.:2.000   1st Qu.:2.000   1st Qu.:2.000   1st Qu.:2.000  
##  Median :3.000   Median :3.000   Median :3.000   Median :4.000  
##  Mean   :2.977   Mean   :3.205   Mean   :3.253   Mean   :3.441  
##  3rd Qu.:4.000   3rd Qu.:4.000   3rd Qu.:4.000   3rd Qu.:5.000  
##  Max.   :5.000   Max.   :5.000   Max.   :5.000   Max.   :5.000  
##                                                                 
##  Inflight.entertainment On.board.service Leg.room.service Baggage.handling
##  Min.   :0.000          Min.   :0.000    Min.   :0.000    Min.   :1.000   
##  1st Qu.:2.000          1st Qu.:2.000    1st Qu.:2.000    1st Qu.:3.000   
##  Median :4.000          Median :4.000    Median :4.000    Median :4.000   
##  Mean   :3.358          Mean   :3.383    Mean   :3.351    Mean   :3.632   
##  3rd Qu.:4.000          3rd Qu.:4.000    3rd Qu.:4.000    3rd Qu.:5.000   
##  Max.   :5.000          Max.   :5.000    Max.   :5.000    Max.   :5.000   
##                                                                           
##  Checkin.service Inflight.service  Cleanliness    Departure.Delay.in.Minutes
##  Min.   :0.000   Min.   :0.000    Min.   :0.000   Min.   :   0.00           
##  1st Qu.:3.000   1st Qu.:3.000    1st Qu.:2.000   1st Qu.:   0.00           
```

```
##  Median :3.000   Median :4.000    Median :3.000   Median :   0.00
##  Mean   :3.306   Mean   :3.642    Mean   :3.286   Mean   :  14.71
##  3rd Qu.:4.000   3rd Qu.:5.000    3rd Qu.:4.000   3rd Qu.:  12.00
##  Max.   :5.000   Max.   :5.000    Max.   :5.000   Max.   :1592.00
##
##  Arrival.Delay.in.Minutes satisfaction
##  Min.   :   0.00          Length:129880
##  1st Qu.:   0.00          Class :character
##  Median :   0.00          Mode  :character
##  Mean   :  15.09
##  3rd Qu.:  13.00
##  Max.   :1584.00
##  NA's   :393
```

From the summary, it is evident that many features represent ratings on the services provided by the airline agency, and these ratings range from 0 to 5. Additionally, we noticed that the "Arrival Delay in Minutes" feature contains some missing values (NA).

Next, we will examine the distribution of all nominal features. Specifically, we have categorical data for Gender, Customer Type, Type of Travel, and Class, while all the rating features are ordinal.

```r
table(data$Gender)
```

```
##
## Female   Male
##  65899  63981
```

The "Gender" feature appears to be well-balanced, meaning that it has an approximately equal number of occurrences for each category, likely male and female. This balance can be beneficial for modeling as it prevents any significant bias towards a particular gender in the analysis and predictions.

```r
table(data$Customer.Type)
```

```
##
## disloyal Customer    Loyal Customer
##             23780            106100
```

The "Customer Type" feature contains only two values, "disloyal customer" and "loyal customer." The distribution of values is imbalanced, with one category potentially having significantly more occurrences than the other.

```r
table(data$Type.of.Travel)
```

```
##
## Business travel Personal Travel
##           89693           40187
```

The "Type of Travel" feature consists of only two values: "personal travel" and "business travel." The distribution of values is imbalanced, with "business travel" occurring twice as much as "personal travel."

```r
table(data$Class)
```

```
##
## Business      Eco Eco Plus
##    62160    58309     9411
```

The "Class" feature contains three values: "business," "eco plus," and "eco." The distribution of values is imbalanced. "Business" and "eco" classes appear to be relatively balanced, while "eco plus" is significantly underrepresented compared to the other two classes.

```
table(data$satisfaction)
```

```
##
## neutral or dissatisfied               satisfied
##                   73452                   56428
```

The "satisfaction" feature, which serves as our target variable, is an important aspect of the analysis. The values for this feature are not perfectly balanced, meaning that there is an unequal distribution of satisfied and dissatisfied passengers in the dataset.

Data preprocessing

In this section of data preprocessing, several steps are performed to prepare the dataset for further analysis and modeling. The specific actions taken include:

1. Renaming columns: the names of the features (columns) are modified to improve their clarity and usability.
2. Dropping unnecessary columns: two columns, "X" and "id," are removed from the dataset. The "X" column likely represents the index of the row, which does not carry any meaningful information for analysis. The "id" column is presumed to be an unknown indexing number, which may not contribute to the predictive modeling process.
3. Converting categorical variables to factors: categorical variables, such as "Gender", "Customer Type", "Type of Travel" and "Class" are converted into factors. Converting categorical variables into factors is a common practice in R to represent these variables as distinct levels, allowing for better handling and analysis in statistical models.

By performing these data preprocessing steps, the dataset is cleaned and transformed into a more suitable format for the subsequent analysis, making it easier to build a predictive model for passenger satisfaction.

```
# replace dots with underscores in column names
names(data) = gsub("\\.", "_", names(data))
# drop X and id column
data = data %>% select(-X, -id)
names(data)
```

```
##  [1] "Gender"                            "Customer_Type"
##  [3] "Age"                               "Type_of_Travel"
##  [5] "Class"                             "Flight_Distance"
##  [7] "Inflight_wifi_service"             "Departure_Arrival_time_convenient"
##  [9] "Ease_of_Online_booking"            "Gate_location"
## [11] "Food_and_drink"                    "Online_boarding"
## [13] "Seat_comfort"                      "Inflight_entertainment"
## [15] "On_board_service"                  "Leg_room_service"
## [17] "Baggage_handling"                  "Checkin_service"
## [19] "Inflight_service"                  "Cleanliness"
## [21] "Departure_Delay_in_Minutes"        "Arrival_Delay_in_Minutes"
## [23] "satisfaction"
```

```
# convert categorical features to factor
data$Gender = factor(data$Gender, levels = c("Male", "Female"))
data$Customer_Type = factor(data$Customer_Type, levels = c("Loyal Customer", "disloyal Customer"))
data$Type_of_Travel = factor(data$Type_of_Travel, levels = c("Personal Travel", "Business travel"))
data$Class = factor(data$Class, levels = c("Business", "Eco Plus", "Eco"))
data$satisfaction = factor(data$satisfaction, levels = c("neutral or dissatisfied", "satisfied"))
```

Handling na values

In this section, we analyze the dataset to identify variables with missing values, particularly focusing on the "Arrival_Delay_in_Minutes" variable. We calculate the proportion of missing values for this variable and subsequently remove the examples or rows with missing values from the dataset.

```
# list features with na values
prop.table(colSums(is.na(data)))
```

```
##                            Gender                     Customer_Type
##                                 0                                 0
##                               Age                    Type_of_Travel
##                                 0                                 0
##                             Class                   Flight_Distance
##                                 0                                 0
##             Inflight_wifi_service Departure_Arrival_time_convenient
##                                 0                                 0
##            Ease_of_Online_booking                     Gate_location
##                                 0                                 0
##                    Food_and_drink                   Online_boarding
##                                 0                                 0
##                      Seat_comfort            Inflight_entertainment
##                                 0                                 0
##                  On_board_service                  Leg_room_service
##                                 0                                 0
##                  Baggage_handling                   Checkin_service
##                                 0                                 0
##                  Inflight_service                       Cleanliness
##                                 0                                 0
##        Departure_Delay_in_Minutes          Arrival_Delay_in_Minutes
##                                 0                                 1
##                      satisfaction
##                                 0
```

To determine the proportion of missing values for the "Arrival_Delay_in_Minutes" variable, we can count the number of instances where this variable has missing values (commonly denoted as "NaN" or "NA") and divide it by the total number of examples in the dataset. This will give us the proportion of missing values for the "Arrival_Delay_in_Minutes" variable.

```
# Arrival_Delay_in_Minutes has na values, proportion of na values
prop.table(table(is.na(data$Arrival_Delay_in_Minutes)))
```

```
##
##      FALSE       TRUE
## 0.99697413 0.00302587
```

Indeed, since the proportion of missing values for the "Arrival_Delay_in_Minutes" variable is very low (less than 3% of the entire dataset), it is reasonable to proceed with dropping these missing values from the dataset.

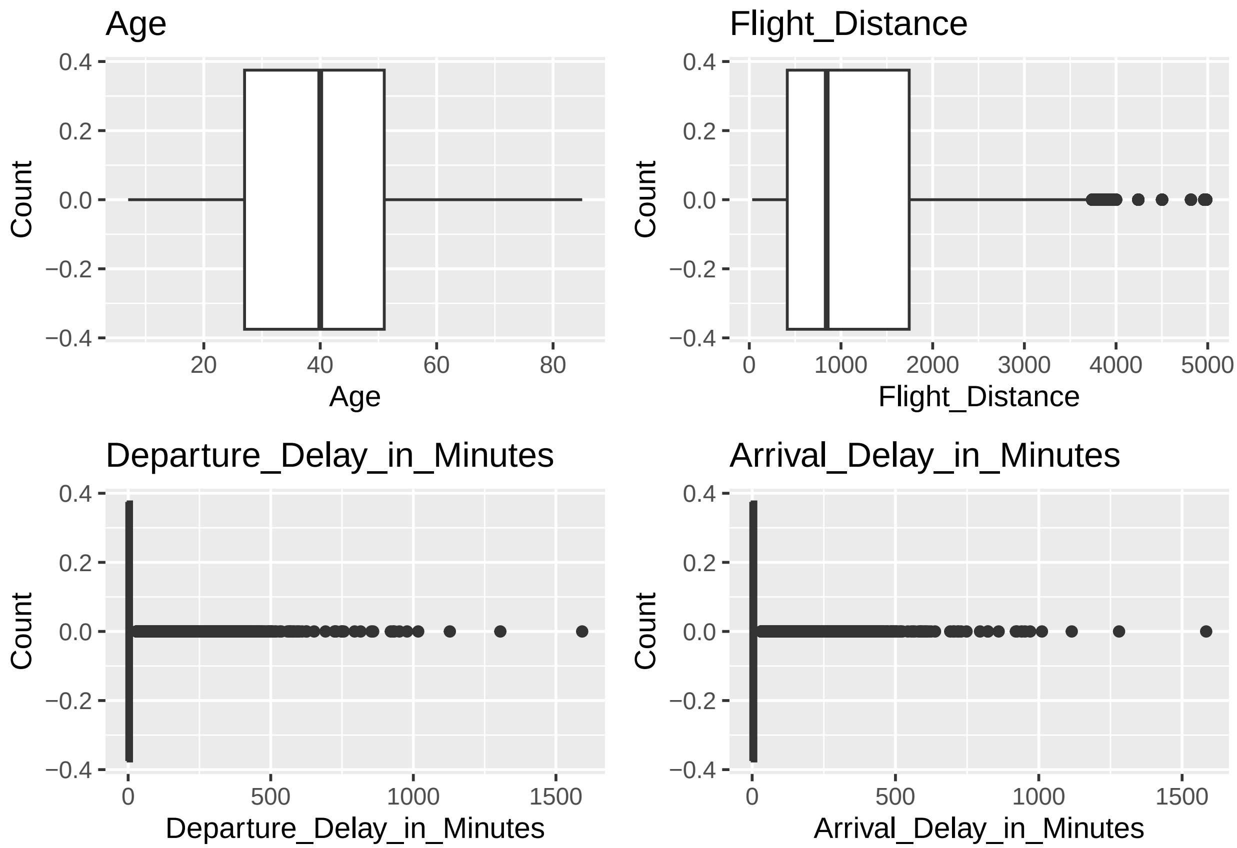
```
# na values are only 0.03% of the data -> drop na values
data = data %>% drop_na(Arrival_Delay_in_Minutes)
```

Outliers

In this section, box plots are created for each numeric variable present in the dataset. Box plots are a powerful visualization tool used to identify the presence of outliers in the data. For each numeric variable, the box plot displays a box that represents the interquartile range (IQR), with the median indicated by a line inside the box. The "whiskers" extending from the box show the range of the data, and any data points beyond the whiskers are considered potential outliers.

By examining the box plots for each numeric variable, we can visually identify any data points that lie far outside the typical range of the data, indicating potential outliers. Outliers can significantly impact statistical analyses, so detecting and handling them appropriately is crucial for ensuring the integrity of the dataset and the accuracy of subsequent analyses and modeling.

```r
ratings_fts_names = c("Inflight_wifi_service", "Departure_Arrival_time_convenient",
  "Ease_of_Online_booking", "Gate_location", "Food_and_drink", "Online_boarding",
  "Seat_comfort", "Inflight_entertainment", "On_board_service", "Leg_room_service",
  "Baggage_handling", "Checkin_service", "Inflight_service", "Cleanliness", "On_board_service")

# plot boxplot of each numeric variable excluding ratings features
plots = list()
for (col in names(data)[sapply(data, is.numeric)]) {
  if (col %in% ratings_fts_names) {
    next
  }
  plot = ggplot(data, aes(x = .data[[col]])) +
  geom_boxplot() +
  labs(title = col, x = col, y = "Count")
  plots[[col]] = plot
}

grid.arrange(grobs = plots, ncol = 2)
```



We can see that there are outliers in Departure_Delay_in_Minutes, Arrival_Delay_in_Minutes and Flight_Distance. Considering the presence of both near-zero and very large values in the dataset, alternative distributions like the log-normal distribution may be more appropriate for modeling the "Departure_Delay_in_Minutes" and "Arrival_Delay_in_Minutes" variables, as they can better capture the variability in delay times.

Our variables vs satisfaction

Now we can visualize how the features are distributed by satisfaction. This section provides a summary of each variable in the dataset, grouped by our target variable.

```r
# Print summary for each variable grouped by satisfaction, including the name of the variable
for (col in names(data)) {
  print(col)
  print(by(data[[col]], data$satisfaction, summary))
}
```

```
## [1] "Gender"
## data$satisfaction: neutral or dissatisfied
##   Male Female
##  35701  37524
## ------------------------------------------------------------
## data$satisfaction: satisfied
##   Male Female
##  28083  28179
## [1] "Customer_Type"
## data$satisfaction: neutral or dissatisfied
```

```
##    Loyal Customer disloyal Customer
##             55199            18026
## ------------------------------------------------------------
## data$satisfaction: satisfied
##    Loyal Customer disloyal Customer
##             50574             5688
## [1] "Age"
## data$satisfaction: neutral or dissatisfied
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    7.00   25.00   37.00   37.65   50.00   85.00
## ------------------------------------------------------------
## data$satisfaction: satisfied
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    7.00   32.00   43.00   41.74   51.00   85.00
## [1] "Type_of_Travel"
## data$satisfaction: neutral or dissatisfied
## Personal Travel Business travel
##           35987           37238
## ------------------------------------------------------------
## data$satisfaction: satisfied
## Personal Travel Business travel
##            4055           52207
## [1] "Class"
## data$satisfaction: neutral or dissatisfied
## Business Eco Plus      Eco
##    18940     7070    47215
## ------------------------------------------------------------
## data$satisfaction: satisfied
## Business Eco Plus      Eco
##    43050     2310    10902
## [1] "Flight_Distance"
## data$satisfaction: neutral or dissatisfied
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    31.0   372.0   674.0   929.5  1149.0  4983.0
## ------------------------------------------------------------
## data$satisfaction: satisfied
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      31     525    1249    1530    2407    4983
## [1] "Inflight_wifi_service"
## data$satisfaction: neutral or dissatisfied
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   2.000   2.000   2.398   3.000   5.000
## ------------------------------------------------------------
## data$satisfaction: satisfied
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   2.000   4.000   3.158   5.000   5.000
## [1] "Departure_Arrival_time_convenient"
## data$satisfaction: neutral or dissatisfied
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00    2.00    3.00    3.13    4.00    5.00
## ------------------------------------------------------------
## data$satisfaction: satisfied
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   2.000   3.000   2.962   4.000   5.000
```

```
## [1] "Ease_of_Online_booking"
## data$satisfaction: neutral or dissatisfied
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00    2.00    3.00    2.55    3.00    5.00
## ------------------------------------------------------------
## data$satisfaction: satisfied
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   2.000   3.000   3.027   4.000   5.000
## [1] "Gate_location"
## data$satisfaction: neutral or dissatisfied
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1.00    2.00    3.00    2.98    4.00    5.00
## ------------------------------------------------------------
## data$satisfaction: satisfied
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   2.000   3.000   2.973   4.000   5.000
## [1] "Food_and_drink"
## data$satisfaction: neutral or dissatisfied
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   2.000   3.000   2.959   4.000   5.000
## ------------------------------------------------------------
## data$satisfaction: satisfied
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   3.000   4.000   3.525   5.000   5.000
## [1] "Online_boarding"
## data$satisfaction: neutral or dissatisfied
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   2.000   3.000   2.659   3.000   5.000
## ------------------------------------------------------------
## data$satisfaction: satisfied
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   4.000   4.000   4.026   5.000   5.000
## [1] "Seat_comfort"
## data$satisfaction: neutral or dissatisfied
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   2.000   3.000   3.039   4.000   5.000
## ------------------------------------------------------------
## data$satisfaction: satisfied
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   4.000   4.000   3.966   5.000   5.000
## [1] "Inflight_entertainment"
## data$satisfaction: neutral or dissatisfied
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   2.000   3.000   2.892   4.000   5.000
## ------------------------------------------------------------
## data$satisfaction: satisfied
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   4.000   4.000   3.964   5.000   5.000
## [1] "On_board_service"
## data$satisfaction: neutral or dissatisfied
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00    2.00    3.00    3.02    4.00    5.00
## ------------------------------------------------------------
## data$satisfaction: satisfied
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   3.000   4.000   3.856   5.000   5.000
## [1] "Leg_room_service"
## data$satisfaction: neutral or dissatisfied
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00    2.00    3.00    2.99    4.00    5.00
## ------------------------------------------------------------
## data$satisfaction: satisfied
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00    3.00    4.00    3.82    5.00    5.00
## [1] "Baggage_handling"
## data$satisfaction: neutral or dissatisfied
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   3.000   4.000   3.375   4.000   5.000
## ------------------------------------------------------------
## data$satisfaction: satisfied
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   4.000   4.000   3.967   5.000   5.000
## [1] "Checkin_service"
## data$satisfaction: neutral or dissatisfied
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   2.000   3.000   3.043   4.000   5.000
## ------------------------------------------------------------
## data$satisfaction: satisfied
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   3.000   4.000   3.649   5.000   5.000
## [1] "Inflight_service"
## data$satisfaction: neutral or dissatisfied
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00    3.00    4.00    3.39    4.00    5.00
## ------------------------------------------------------------
## data$satisfaction: satisfied
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   4.000   4.000   3.971   5.000   5.000
## [1] "Cleanliness"
## data$satisfaction: neutral or dissatisfied
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   2.000   3.000   2.933   4.000   5.000
## ------------------------------------------------------------
## data$satisfaction: satisfied
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   3.000   4.000   3.746   5.000   5.000
## [1] "Departure_Delay_in_Minutes"
## data$satisfaction: neutral or dissatisfied
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00    0.00    0.00   16.34   15.00 1592.00
## ------------------------------------------------------------
## data$satisfaction: satisfied
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00    0.00    0.00   12.44    9.00 1305.00
## [1] "Arrival_Delay_in_Minutes"
## data$satisfaction: neutral or dissatisfied
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00    0.00    0.00   17.06   16.00 1584.00
```

```
## ------------------------------------------------------------
## data$satisfaction: satisfied
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00    0.00    0.00   12.53    8.00 1280.00
## [1] "satisfaction"
## data$satisfaction: neutral or dissatisfied
## neutral or dissatisfied               satisfied
##                   73225                       0
## ------------------------------------------------------------
## data$satisfaction: satisfied
## neutral or dissatisfied               satisfied
##                       0                   56262
```

From the analysis of the dataset, we observe the following insights:

1. Gender does not appear to significantly influence satisfaction levels, as both men and women show similar satisfaction and dissatisfaction rates.
2. There is a trend suggesting that younger passengers are more likely to be dissatisfied compared to older passengers, as evidenced by the lower values at the 1st quartile and median of the age distribution for dissatisfied customers.
3. Passengers traveling for personal reasons are more likely to be dissatisfied than those on business trips.
4. Conversely, passengers on business trips have a slightly higher likelihood of being satisfied with their travel experience.
5. Customers in the business class are more likely to be satisfied with the airline services compared to those in the economy and economy plus classes.
6. Longer distance flights tend to have higher satisfaction levels among passengers.
7. Departure and arrival delays are associated with a higher likelihood of passenger dissatisfaction.

These insights provide valuable information for the airline company to understand customer preferences and pain points, allowing them to improve services, prioritize customer satisfaction, and address specific areas that may lead to dissatisfaction among passengers. Visualizing each distribution can indeed provide a clearer understanding of the results.

Visualization

In this section, histograms are used to visualize the distribution of the variables in the dataset, starting with the nominal features. By creating histograms for the nominal features, we can gain insights into the distribution of categories within each feature.

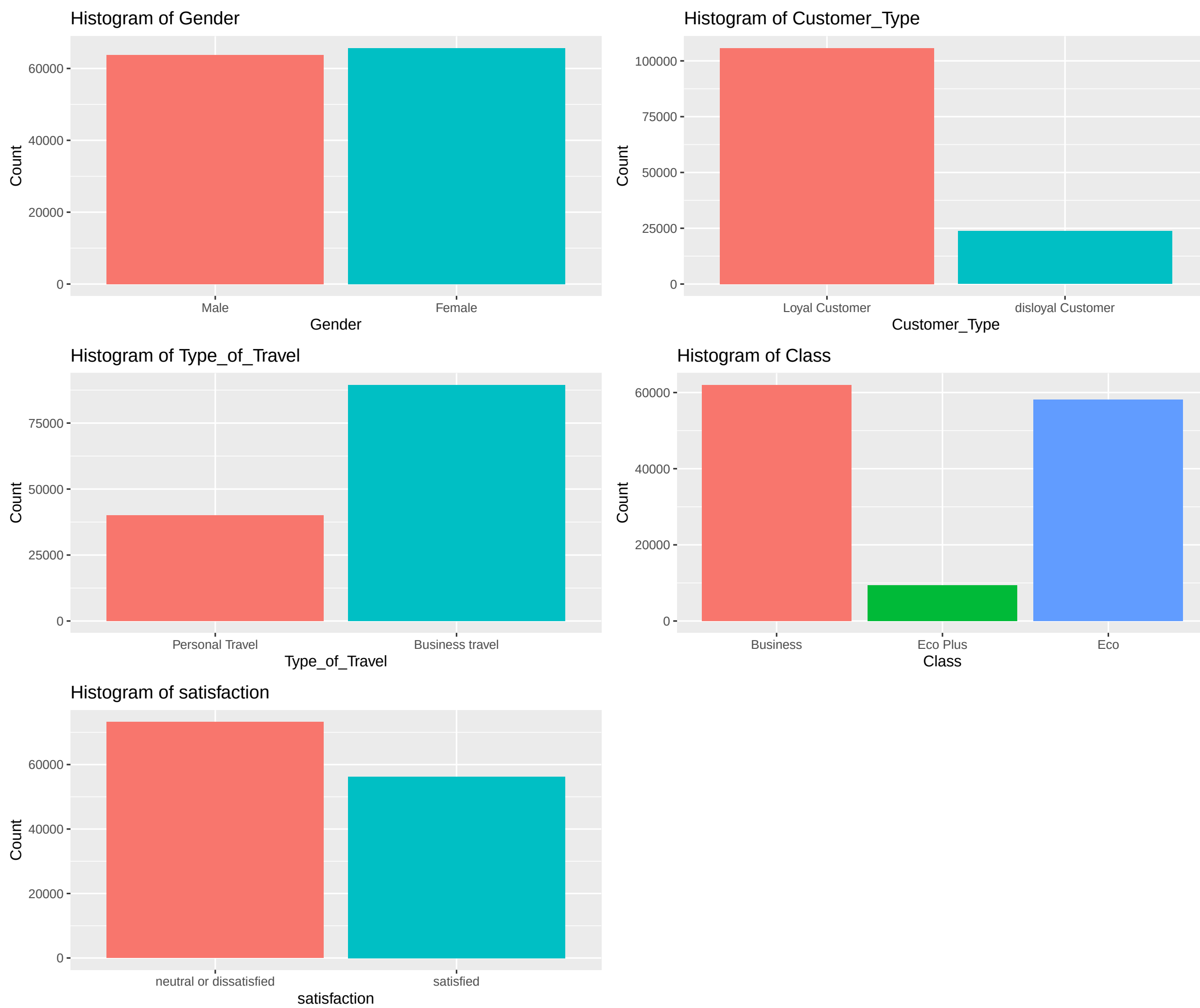
Upon visualizing the nominal features, it becomes apparent that some features exhibit heavily unbalanced distributions. This means that certain categories within these features have significantly higher frequencies compared to others. The presence of such imbalanced distributions could have implications for analysis and modeling, as it may lead to biased results or difficulties in predicting less frequent categories accurately.

```r
# plot distribution of categorical variables
plots = list()
for (col in names(data)[sapply(data, is.factor)]) {
  plot = ggplot(data, aes(x = .data[[col]], fill = .data[[col]])) +
  geom_bar() +
  labs(title = paste("Histogram of", col), x = col, y = "Count") +
  guides(fill = FALSE)
```

```
  plots[[col]] = plot
}
```

```
## Warning: The `<scale>` argument of `guides()` cannot be `FALSE`. Use "none" instead as
## of ggplot2 3.3.4.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

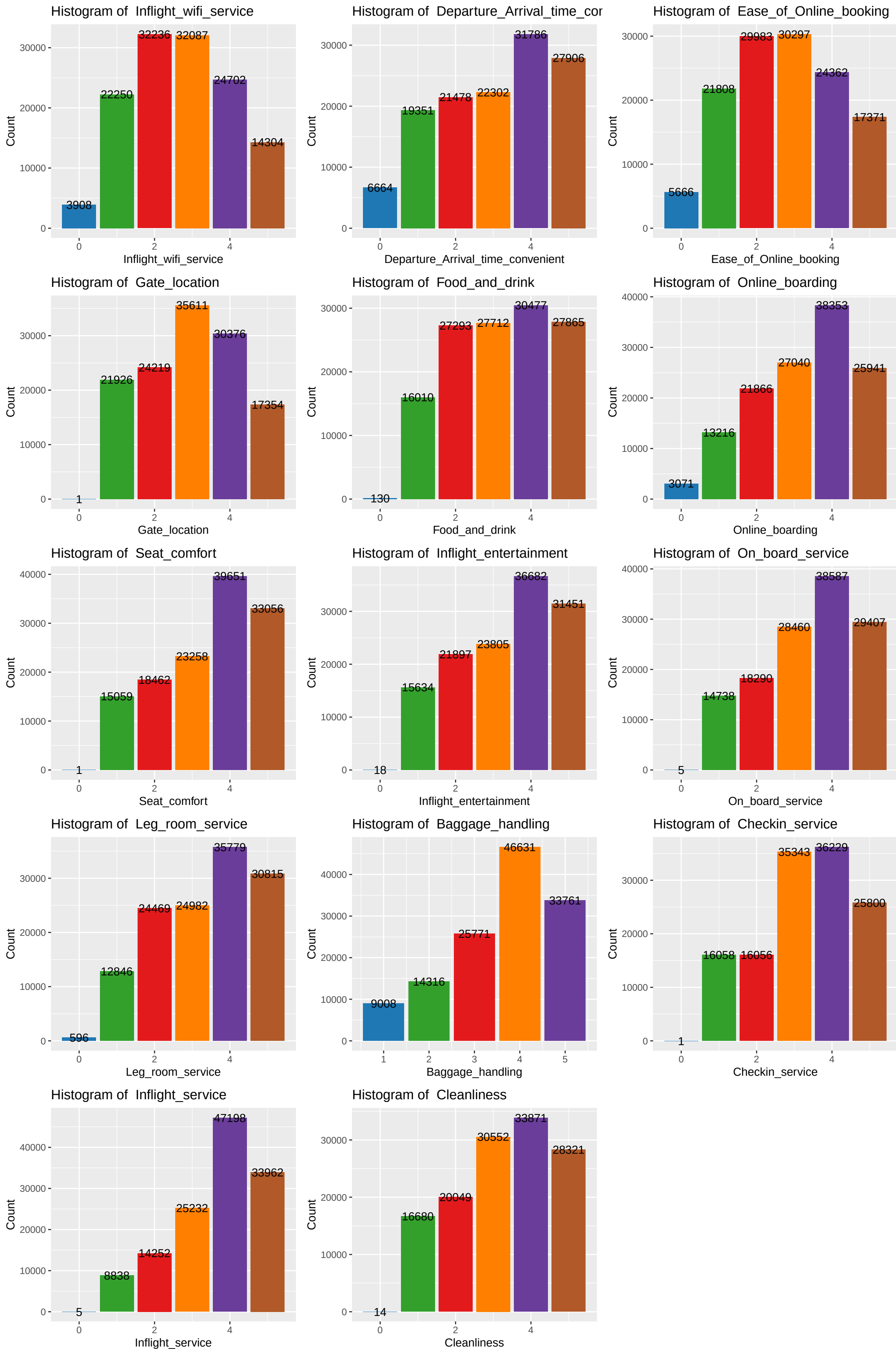
```
grid.arrange(grobs = plots, ncol = 2)
```



Then we plot the distribution of ratings features.

```
# plot distribution of ratings features
plots = list()
my_palette <- c("#1f78b4", "#33a02c", "#e31a1c", "#ff7f00", "#6a3d9a", "#b15928")

for (col in names(data)[sapply(data, is.numeric)]) {
  if (!col %in% ratings_fts_names) {
    next
  }
  plot <- ggplot(data, aes(x = .data[[col]], fill = factor(.data[[col]]))) +
```

```r
    geom_bar() +
    geom_text(stat = 'count', aes(label = after_stat(count))) +
    labs(title = paste("Histogram of ", col), x = col, y = "Count") +
    scale_fill_manual(values = my_palette) +
    guides(fill = FALSE)

  plots[[col]] <- plot
}
grid.arrange(grobs = plots, ncol = 3)
```

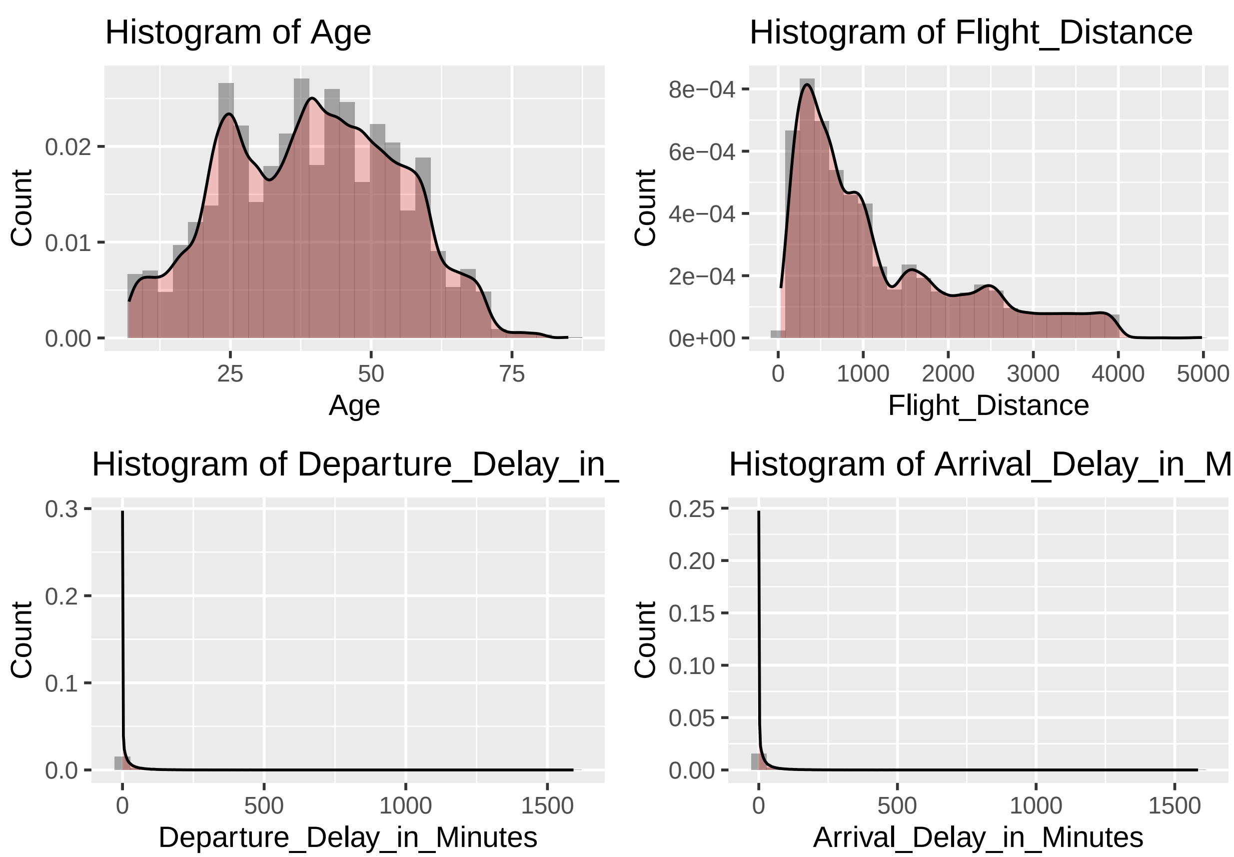


Histogram of Inflight_wifi_service
3908 22250 32236 32087 24702 14304
Inflight_wifi_service
Histogram of Departure_Arrival_time_cor
6664 19351 21478 22302 31786 27906
Departure_Arrival_time_convenient
Histogram of Ease_of_Online_booking
5666 21808 29983 30297 24362 17371
Ease_of_Online_booking
Histogram of Gate_location
1 21926 24219 35611 30376 17354
Gate_location
Histogram of Food_and_drink
130 16010 27293 27712 30477 27865
Food_and_drink
Histogram of Online_boarding
3071 13216 21866 27040 38353 25941
Online_boarding
Histogram of Seat_comfort
1 15059 18462 23258 39651 33056
Seat_comfort
Histogram of Inflight_entertainment
18 15634 21897 23805 36682 31451
Inflight_entertainment
Histogram of On_board_service
5 14738 18290 28460 38587 29407
On_board_service
Histogram of Leg_room_service
596 12846 24469 24982 35779 30815
Leg_room_service
Histogram of Baggage_handling
9008 14316 25771 46631 33761
Baggage_handling
Histogram of Checkin_service
1 16058 16056 35343 36229 25800
Checkin_service
Histogram of Inflight_service
5 8838 14252 25232 47198 33962
Inflight_service
Histogram of Cleanliness
14 16680 20049 30552 33871 28321
Cleanliness
Count

This section includes histograms to visualize the distribution of numeric variables in the dataset.

needs interpretation

```r
# plot distribution and density of numeric variables excluding ratings features
plots = list()
for (col in names(data)[sapply(data, is.numeric)]) {
  if (col %in% ratings_fts_names) {
    next
  }
  plot = ggplot(data, aes(x = .data[[col]])) +
  geom_histogram(aes(y = after_stat(density)), bins = 30, alpha = 0.5) +
  geom_density(alpha = 0.2, fill = "red") +
  labs(title = paste("Histogram of", col), x = col, y = "Count")

  plots[[col]] = plot
}

grid.arrange(grobs = plots, ncol = 2)
```

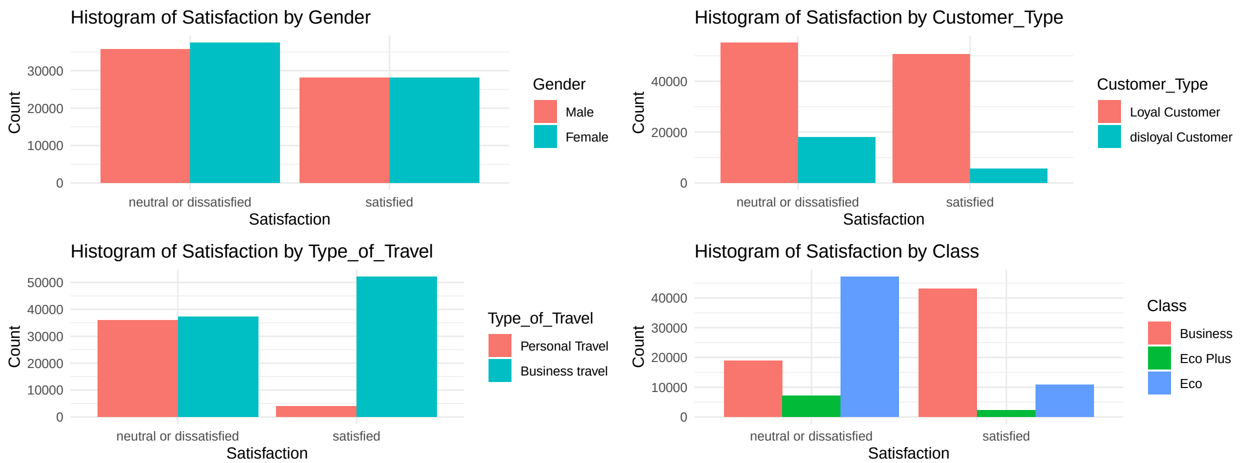


Visualization vs satisfaction

** TODO: write interpretations of the graphs **

```r
# plots categorical variables vs satisfaction
plots = list()
for (col in names(data)[sapply(data, is.factor)]) {
```

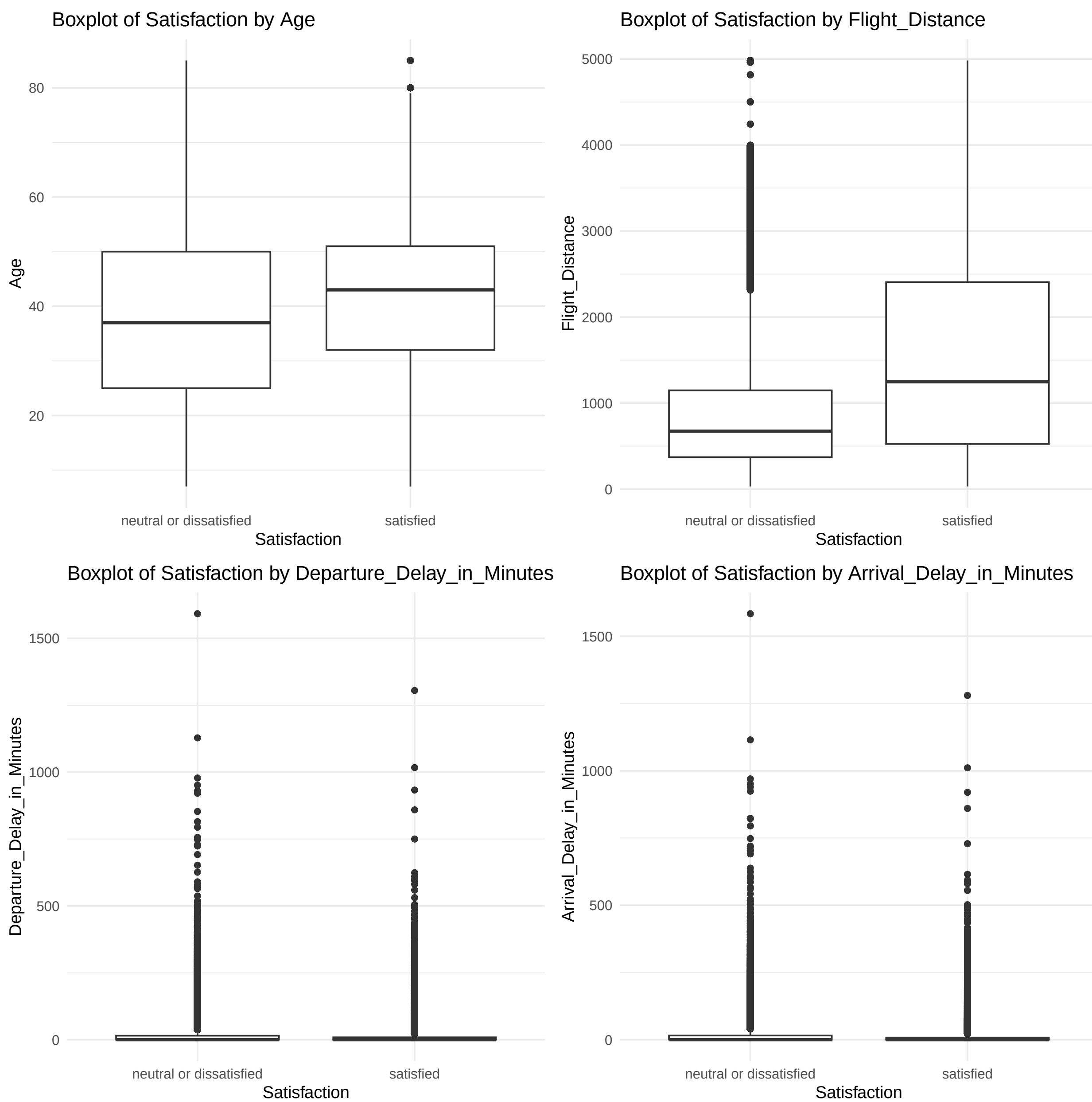
```r
  if (col == "satisfaction") {
    next
  }
  plot = ggplot(data, aes(x = satisfaction, fill = .data[[col]])) +
  theme_minimal() +
  geom_bar(position = "dodge") +
  labs(title = paste("Histogram of Satisfaction by", col), x = "Satisfaction", y = "Count")

  plots[[col]] = plot

}

grid.arrange(grobs = plots, ncol = 2)
```




```r
# plots ratings features vs satisfaction
plots = list()
for (col in names(data)[sapply(data, is.numeric)]) {
  if (!col %in% ratings_fts_names) {
    next
  }
  plot = ggplot(data, aes(x = .data[[col]], fill = satisfaction)) +
  theme_minimal() +
  geom_bar(position = "dodge") +
  labs(title = paste("Histogram of Satisfaction by", col), x = "Satisfaction", y = "Count")

  plots[[col]] = plot
}

grid.arrange(grobs = plots, ncol = 2)
```


```r
# plots numeric variables vs satisfaction excluding ratings features
plots = list()
```

```r
for (col in names(data)[sapply(data, is.numeric)]) {
  if (col %in% ratings_fts_names) {
    next
  }
  plot = ggplot(data, aes(x = satisfaction, y = .data[[col]])) +
  theme_minimal() +
  geom_boxplot() +
  labs(title = paste("Boxplot of Satisfaction by", col), x = "Satisfaction", y = col)

  plots[[col]] = plot
}

grid.arrange(grobs = plots, ncol = 2)
```



Convert categorical to numerical

This section converts the categorical variables to numeric representation for further analysis.

```
gender_map = c("Male" = 0, "Female" = 1)
data$Gender = gender_map[as.numeric(data$Gender)]

customer_type_map = c("Loyal Customer" = 0, "disloyal Customer" = 1)
data$Customer_Type = customer_type_map[as.numeric(data$Customer_Type)]

type_of_travel_map = c("Personal Travel" = 0, "Business travel" = 1)
data$Type_of_Travel = type_of_travel_map[as.numeric(data$Type_of_Travel)]

class_map = c("Business" = 0, "Eco" = 1, "Eco Plus" = 2)
data$Class = class_map[as.numeric(data$Class)]
```

```
satisfaction_map = c("neutral or dissatisfied" = 0, "satisfied" = 1)
data$satisfaction = satisfaction_map[as.numeric(data$satisfaction)]
```

Data balance

This section calculates the proportion of satisfied and dissatisfied customers in the dataset.

```
prop.table(table(data$satisfaction))
```

```
##
##         0         1
## 0.5655008 0.4344992
```

Train test split

This section splits the data into training and testing sets, prints the proportion of satisfied and dissatisfied customers in each set, and saves the true values of the target variable for the test set.

```
set.seed(123)
train_index = sample(1:nrow(data), 0.8*nrow(data))
# 80% of data is used for training
train = data[train_index,]
# 20% of data is used for testing
test = data[-train_index,]

# merge train and test data
data = rbind(train, test)
# save on cvs
# write.csv(data, "data.csv")

# save true values of test satisfaction column
test_true = test$satisfaction

# drop satisfaction column from test data
test = test %>% select(-satisfaction)

# print proportion of satisfied and dissatisfied customers in train and test data
prop.table(table(train$satisfaction))
```

```
##
##         0         1
## 0.5668845 0.4331155
```

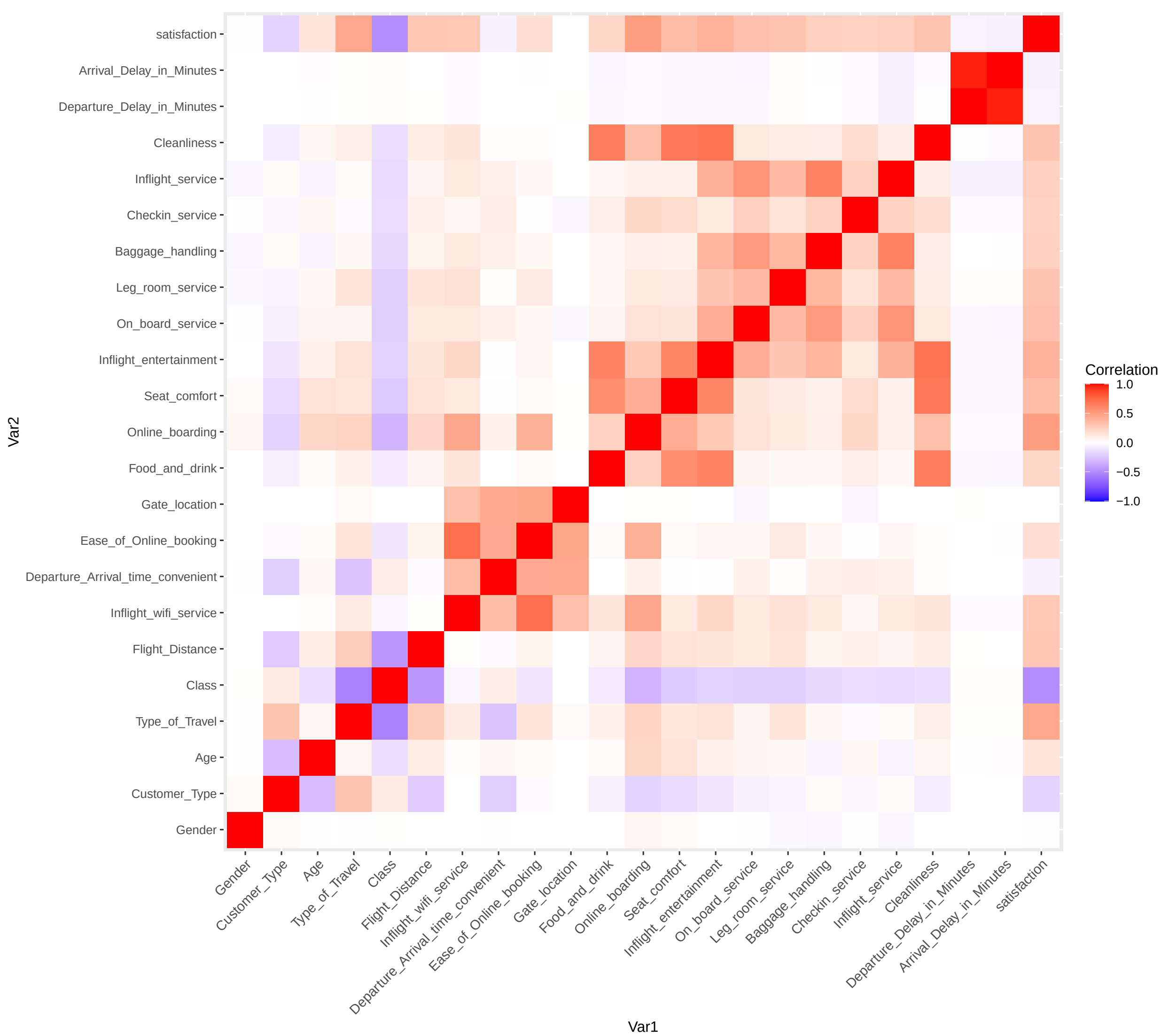
```r
prop.table(table(test_true))
```

```
## test_true
##        0        1
## 0.559966 0.440034
```

Correlation matrix

This section calculates the correlation matrix for numeric variables and plots a heatmap to visualize the correlations between variables.

```r
# correlation matrix only for numeric variables
correlation_matrix = cor(data[, sapply(data, is.numeric)])

# Plot a heatmap of the correlation matrix
ggplot(data = reshape2::melt(correlation_matrix)) +
  geom_tile(aes(x = Var1, y = Var2, fill = value)) +
  scale_fill_gradient2(low = "blue", mid = "white", high = "red",
                       midpoint = 0, limit = c(-1,1), space = "Lab",
                       name="Correlation") +
  theme(axis.text.x = element_text(angle = 45, vjust = 1,
                                   size = 10, hjust = 1)) +
  coord_fixed()
```




```r
par(mfrow = c(1, 1))
```

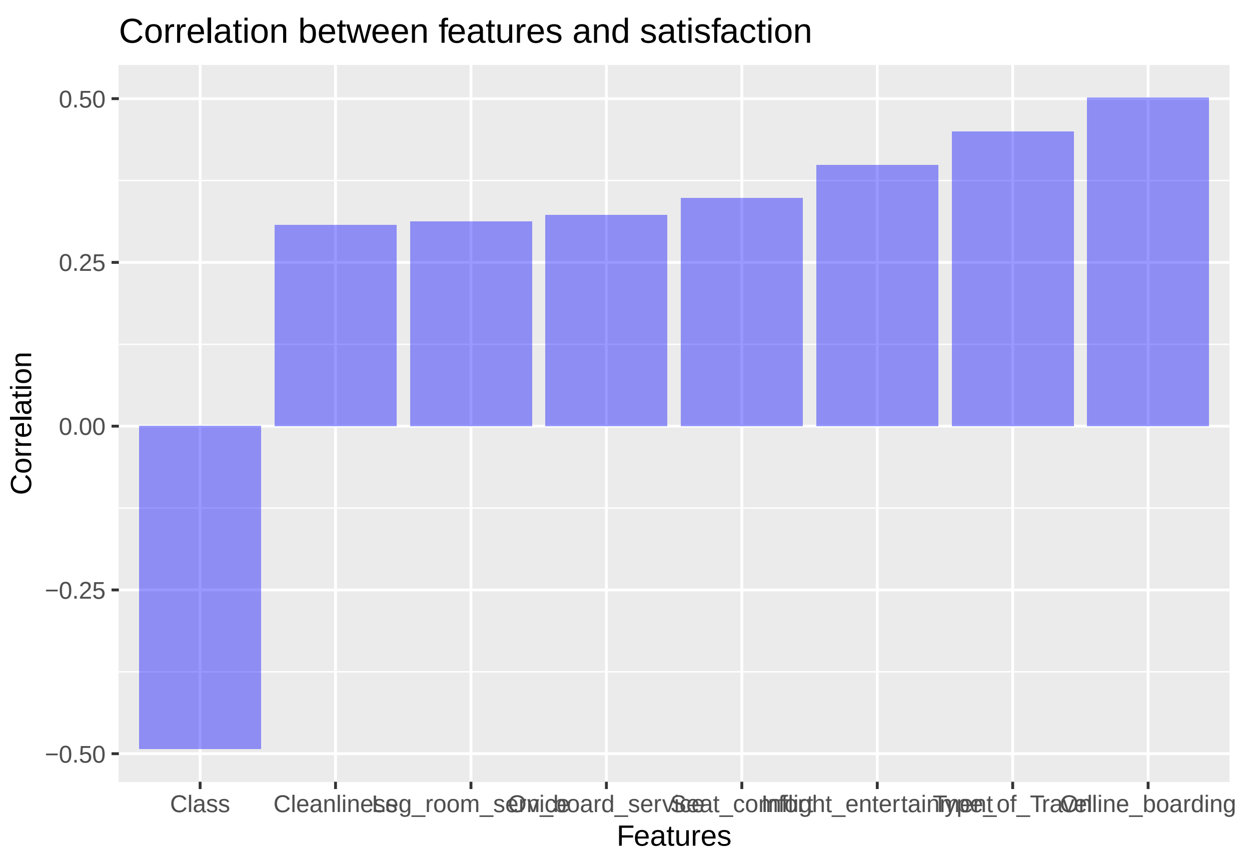
```r
# Find high correlated features with satisfaction
# TODO: do the same with different threshold to find differences
# NOTE: i decided to use 0.3 as threshold
satisfaction_corr <- correlation_matrix['satisfaction',]
high_corr_satis <- names(satisfaction_corr[abs(satisfaction_corr) > 0.3 | abs(satisfaction_corr) < -0.3
high_corr_satis <- high_corr_satis[high_corr_satis != "satisfaction"]
high_corr_satis
```

```
## [1] "Type_of_Travel"         "Class"                  "Online_boarding"
## [4] "Seat_comfort"           "Inflight_entertainment" "On_board_service"
## [7] "Leg_room_service"       "Cleanliness"
```

```r
# Compute the correlations between the high correlation features and satisfaction
correlations <- data.frame(
  feature = high_corr_satis,
  correlation = sapply(high_corr_satis, function(x) cor(data[,x], data$satisfaction))
)
correlations
```

```
##                                       feature correlation
## Type_of_Travel                 Type_of_Travel   0.4497939
## Class                                   Class  -0.4930659
## Online_boarding               Online_boarding   0.5016203
## Seat_comfort                     Seat_comfort   0.3485759
## Inflight_entertainment Inflight_entertainment   0.3983339
## On_board_service             On_board_service   0.3223292
## Leg_room_service             Leg_room_service   0.3125570
## Cleanliness                       Cleanliness   0.3068906
```

```r
# plot the correlations
ggplot(correlations, aes(x = reorder(feature, correlation), y = correlation)) +
  geom_bar(stat = "identity", fill = "blue", alpha = 0.4) +
  ggtitle("Correlation between features and satisfaction") +
  xlab('Features') +
  ylab('Correlation')
```

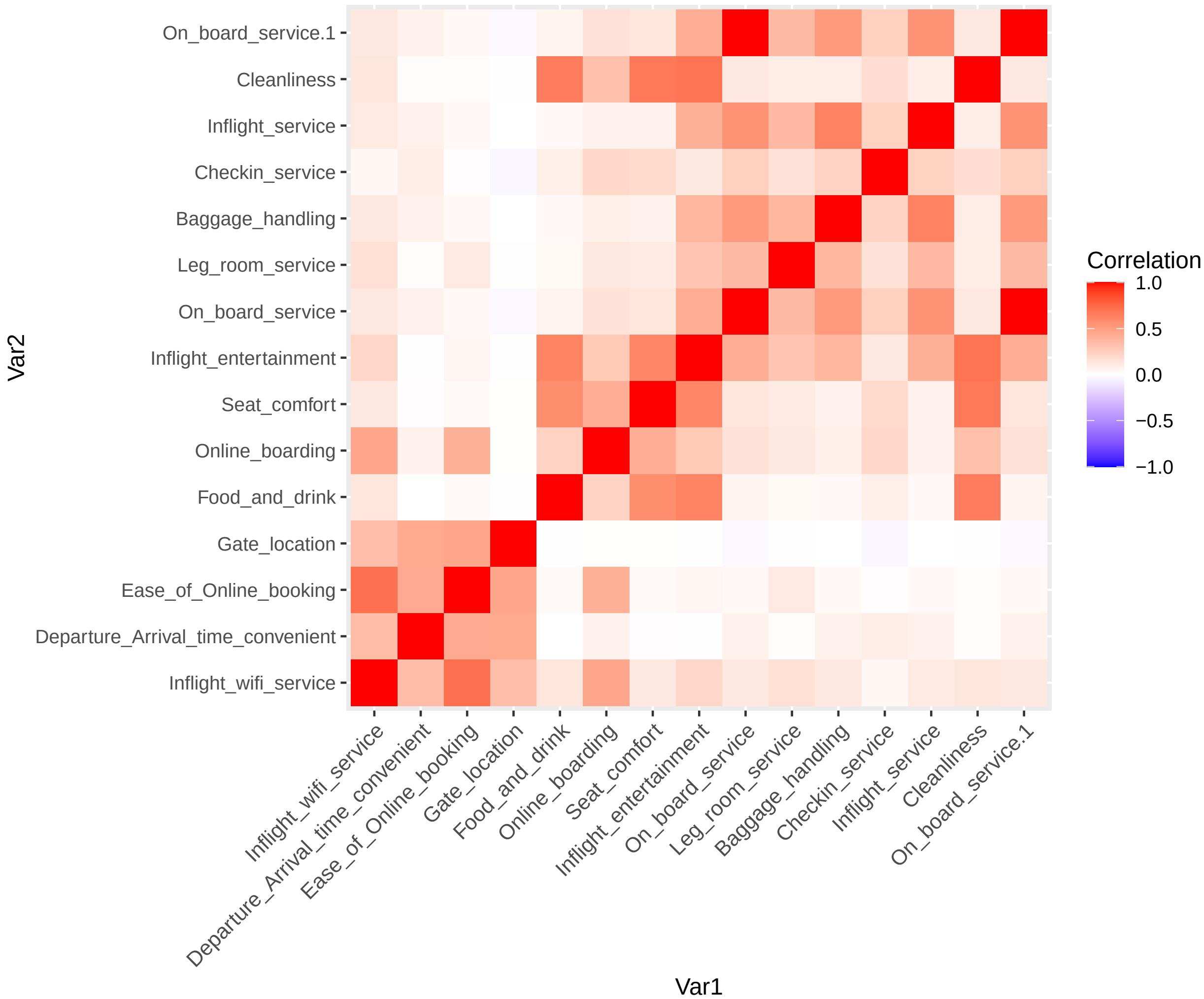



```r
par(mfrow = c(1, 1))
```

```r
#save on cvs
# write.csv(correlations, file = "correlations.csv")
```

Correlation with different ratings

```r
# compute correlation matrix with only ratings features
ratings_data = data[, c(ratings_fts_names)]

# correlation matrix only for ratings features
ratings_correlation_matrix = cor(ratings_data)

# Plot a heatmap of the correlation matrix
ggplot(data = reshape2::melt(ratings_correlation_matrix)) +
  geom_tile(aes(x = Var1, y = Var2, fill = value)) +
  scale_fill_gradient2(low = "blue", mid = "white", high = "red",
                       midpoint = 0, limit = c(-1,1), space = "Lab",
                       name="Correlation") +
  theme(axis.text.x = element_text(angle = 45, vjust = 1,
                                   size = 10, hjust = 1)) +
  coord_fixed()
```




```
par(mfrow = c(1, 1))
```

```
# Assuming you have already calculated the 'ratings_correlation_matrix' using the code you provided.

# Convert the correlation matrix to a data frame to work with it easily
ratings_correlation_df <- as.data.frame(as.table(ratings_correlation_matrix))

# Rename the columns in the data frame
colnames(ratings_correlation_df) <- c("Var1", "Var2", "Correlation")

# Sort the data frame by the absolute correlation values in descending order
sorted_correlation_df <- ratings_correlation_df[order(-abs(ratings_correlation_df$Correlation)), ]

# Filter out the self-correlations (correlation of a variable with itself)
```

```r
sorted_correlation_df <- sorted_correlation_df[sorted_correlation_df$Var1 != sorted_correlation_df$Var2

# Print the top N most correlated features
N <- 15  # Change N to get more or fewer correlated features
top_correlated_features <- head(sorted_correlation_df, N)

print(top_correlated_features)
```

```
##                       Var1                   Var2 Correlation
## 135     On_board_service.1       On_board_service   1.0000000
## 219       On_board_service     On_board_service.1   1.0000000
## 3   Ease_of_Online_booking  Inflight_wifi_service   0.7148885
## 31   Inflight_wifi_service Ease_of_Online_booking   0.7148885
## 119            Cleanliness Inflight_entertainment   0.6924911
## 203 Inflight_entertainment            Cleanliness   0.6924911
## 104            Cleanliness           Seat_comfort   0.6796570
## 202           Seat_comfort            Cleanliness   0.6796570
## 74             Cleanliness          Food_and_drink   0.6580260
## 200         Food_and_drink            Cleanliness   0.6580260
## 163       Inflight_service       Baggage_handling   0.6294924
## 191       Baggage_handling       Inflight_service   0.6294924
## 68  Inflight_entertainment          Food_and_drink   0.6233659
## 110         Food_and_drink Inflight_entertainment   0.6233659
## 98  Inflight_entertainment           Seat_comfort   0.6119491
```

Relation between Arrival_Delay_in_Minutes and Departure_Delay_in_Minutes (linear)

This section explores the partial correlation matrix and identifies variables with high correlations with the target variable (satisfaction). It also creates a bar plot to show the correlations.

```r
#CORRELATION MATRIX again but now we are interested in partial correlation
#So we look for all the correlations between variables
#We pick the highest, setting a treshold of our choice

#build a dataframe where for each variable we look the partial correlation with all the others
#we pick the highest and we save it in a dataframe
#we set a treshold of 0


#correlation(train, partial=TRUE, method='pearson')
#save the partial correlation matrix result in a dataframe and output a file for further analysis


#partial_corr <- correlation(train, partial=TRUE, method='pearson')
#write.csv(partial_corr, file = "partial_corr.csv")

partial_correlations = read.csv("partial_corr.csv", header = TRUE, sep = ",")

#make the first column the row names
rownames(partial_correlations) = partial_correlations[,1]

#drop the first  (X) column
```

```r
partial_correlations = partial_correlations[,-1]

# Create a new matrix with rounded partial correlations
partial_correlations_rounded <- round(partial_correlations, digits = 3)


# Initialize empty data frame with 0 rows
# We need it to create a data frame with the results and
# so to show better the correlations.
df <- data.frame(variable1 = character(),
                 variable2 = character(),
                 value = numeric(),
                 stringsAsFactors = FALSE)

# Loop over rows and columns of matrix
for (i in 1:nrow(partial_correlations_rounded)) {
  for (j in 1:ncol(partial_correlations_rounded)) {
    # Check if value meets criterion
    if ((partial_correlations_rounded[i,j] > 0.300 | partial_correlations_rounded[i,j] < -0.300)& i != j
      # Add row to data frame
      df <- rbind(df, data.frame(variable1 = rownames(partial_correlations_rounded)[i],
                                 variable2 = colnames(partial_correlations_rounded)[j],
                                 value = partial_correlations_rounded[i,j],
                                 stringsAsFactors = FALSE))
    }
  }
}


# Group the data frame by variable1 and extract top 3 values for each group
df_top3 <- df %>% group_by(variable1) %>% top_n(4, value) %>% ungroup()

#order by variable1
df_top3 <- df_top3[order(df_top3$variable1),]


#delete duplicates in the dataframe if variable1 is equal to variable2
df_top3 <- df_top3[!(df_top3$variable1 == df_top3$variable2),]

print(df_top3, n = nrow(df_top3))
```

```
## # A tibble: 16 x 3
##    variable1                  variable2                   value
##    <chr>                      <chr>                       <dbl>
##  1 Arrival_Delay_in_Minutes   Departure_Delay_in_Minutes  0.964
##  2 Baggage_handling           Inflight_service            0.366
##  3 Class                      Type_of_Travel             -0.423
##  4 Cleanliness                Inflight_entertainment      0.411
##  5 Customer_Type              Type_of_Travel              0.497
##  6 Departure_Delay_in_Minutes Arrival_Delay_in_Minutes    0.964
##  7 Ease_of_Online_booking     Inflight_wifi_service       0.539
##  8 Food_and_drink             Inflight_entertainment      0.353
##  9 Inflight_entertainment     Food_and_drink              0.353
## 10 Inflight_entertainment     Cleanliness                 0.411
```

```
## 11 Inflight_service          Baggage_handling          0.366
## 12 Inflight_wifi_service     Ease_of_Online_booking    0.539
## 13 satisfaction              Type_of_Travel            0.351
## 14 Type_of_Travel            Customer_Type             0.497
## 15 Type_of_Travel            Class                    -0.423
## 16 Type_of_Travel            satisfaction              0.351
```

```r
#save on cvs
# write.csv(df_top3, file = "df_top3.csv")
```

```r
# standardize Arrival_Delay_in_Minutes and Departure_Delay_in_Minutes
arrival_std = scale(data$Arrival_Delay_in_Minutes)
departure_std = scale(data$Departure_Delay_in_Minutes)
# scatter plot of Arrival_Delay_in_Minutes and Departure_Delay_in_Minutes
plot(arrival_std, departure_std, xlab = "Arrival_Delay_in_Minutes", ylab = "Departure_Delay_in_Minutes")
# plot line y = x
abline(0, 1, col = "red")
```