

EDA

Süleyman Erim, Giacomo Schiavo, Mattia Varagnolo

2023-07-21

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>

```
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

```

(lot of categorical data, why we can remove X and id is explained later, lots of ordinal features, NA's only on Arrival.Delay.in.Minutes)

From the summary we can see that a lot of features are ordinal because they represent ratings for different services offered during a flight. We can observe that some NA values are present in "Arrival.Delay.in.Minutes" feature, it will be dealt later. Now we will look into all the nominal features to see their primal distribution.

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

```
##
```

```

## Female    Male
## 65899   63981





```

We can see that the “Gender” feature looks pretty balanced while our target feature “satisfaction” has more data on neutral/dissatisfied values.

This section provides a summary of each variable in the dataset, grouped by the target variable (satisfaction).

```

# 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] "X"
## data$satisfaction: neutral or dissatisfied
##   Min. 1st Qu. Median   Mean 3rd Qu.   Max.
##     0    16334   39327   44318   71685  103903
## -----
## data$satisfaction: satisfied
##   Min. 1st Qu. Median   Mean 3rd Qu.   Max.
##     0    16101   38477   43951   71136  103900
## [1] "id"
## data$satisfaction: neutral or dissatisfied
##   Min. 1st Qu. Median   Mean 3rd Qu.   Max.
##     1    31813   64082   64507   97173  129880
## -----
## data$satisfaction: satisfied
##   Min. 1st Qu. Median   Mean 3rd Qu.   Max.
##     2    33201   65948   65504   97758  129879
## [1] "Gender"
## data$satisfaction: neutral or dissatisfied
##   Length   Class    Mode
##     73452 character character
## -----

```

```

## data$satisfaction: satisfied
##      Length     Class     Mode
##      56428 character character
## [1] "Customer.Type"
## data$satisfaction: neutral or dissatisfied
##      Length     Class     Mode
##      73452 character character
## -----
## data$satisfaction: satisfied
##      Length     Class     Mode
##      56428 character character
## [1] "Age"
## data$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
##      Length     Class     Mode
##      73452 character character
## -----
## data$satisfaction: satisfied
##      Length     Class     Mode
##      56428 character character
## [1] "Class"
## data$satisfaction: neutral or dissatisfied
##      Length     Class     Mode
##      73452 character character
## -----
## data$satisfaction: satisfied
##      Length     Class     Mode
##      56428 character character
## [1] "Flight.Distance"
## data$satisfaction: neutral or dissatisfied
##      Min. 1st Qu. Median   Mean 3rd Qu.   Max.
##      31.0   372.0   674.0   929.7  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$satisfaction: satisfied
##      Min. 1st Qu. Median   Mean 3rd Qu.   Max.
##      0.000   2.000   4.000   3.159   5.000   5.000
## [1] "Departure.Arrival.time.convenient"
## data$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$satisfaction: satisfied
##   Min. 1st Qu. Median   Mean 3rd Qu.   Max.
##   0.000  2.000  3.000  2.963  4.000  5.000
## [1] "Ease.of.Online.booking"
## data$satisfaction: neutral or dissatisfied
##   Min. 1st Qu. Median   Mean 3rd Qu.   Max.
##   0.000  2.000  3.000  2.549  3.000  5.000
## -----
## 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  2.958  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  3.038  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   3.747   5.000   5.000
## [1] "Departure.Delay.in.Minutes"
## data$satisfaction: neutral or dissatisfied
##      Min. 1st Qu. Median   Mean 3rd Qu.   Max.
##      0.00    0.00  0.00    16.41   15.00  1592.00
## -----
## data$satisfaction: satisfied
##      Min. 1st Qu. Median   Mean 3rd Qu.   Max.

```

```

##      0.00    0.00    0.00   12.51    9.00 1305.00
## [1] "Arrival.Delay.in.Minutes"
## data$satisfaction: neutral or dissatisfied
##      Min. 1st Qu. Median    Mean 3rd Qu.    Max.    NA's
##      0.00    0.00    0.00   17.06   16.00 1584.00     227
## -----
## data$satisfaction: satisfied
##      Min. 1st Qu. Median    Mean 3rd Qu.    Max.    NA's
##      0.00    0.00    0.00   12.53    8.00 1280.00     166
## [1] "satisfaction"
## data$satisfaction: neutral or dissatisfied
##      Length    Class     Mode
##      73452 character character
## -----
## data$satisfaction: satisfied
##      Length    Class     Mode
##      56428 character character

# print names of the columns
names(data)

## [1] "X"                               "id"
## [3] "Gender"                          "Customer.Type"
## [5] "Age"                            "Type.of.Travel"
## [7] "Class"                           "Flight.Distance"
## [9] "Inflight.wifi.service"          "Departure.Arrival.time.convenient"
## [11] "Ease.of.Online.booking"         "Gate.location"
## [13] "Food.and.drink"                "Online.boarding"
## [15] "Seat.comfort"                  "Inflight.entertainment"
## [17] "On.board.service"              "Leg.room.service"
## [19] "Baggage.handling"              "Checkin.service"
## [21] "Inflight.service"               "Cleanliness"
## [23] "Departure.Delay.in.Minutes"    "Arrival.Delay.in.Minutes"
## [25] "satisfaction"

```

Data preprocessing

This section performs data preprocessing steps such as renaming columns, dropping unnecessary columns, and converting categorical variables to factors.

```

# 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

This section identifies the variables with missing values and calculates the proportion of missing values for the “Arrival_Delay_in_Minutes” variable. It then drops the examples with missing values.

```

# 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	

From here we can see that Arrival_Delay_in_Minutes has missing values, let's the proportion of na values

```

# 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

```

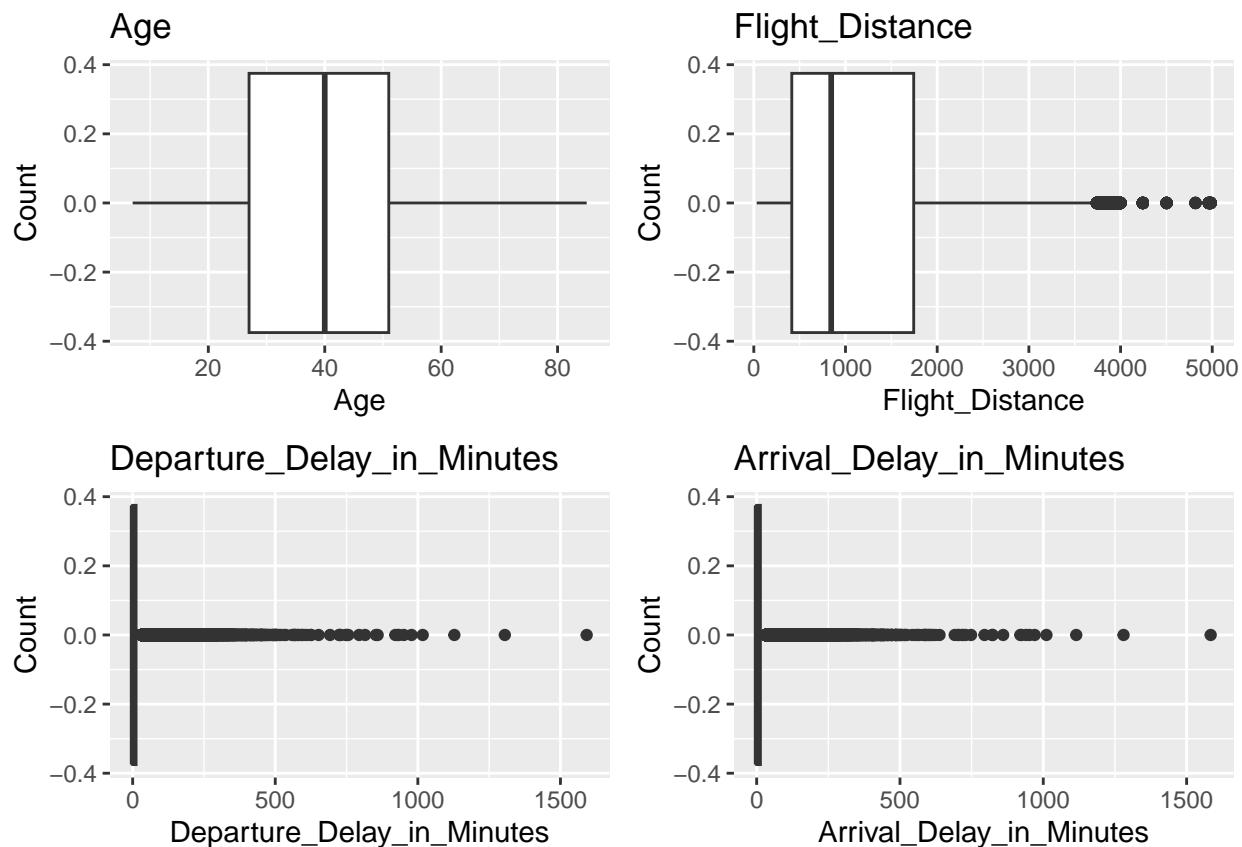
```
# na values are only 0.03% of the data -> drop na values
data = data %>% drop_na(Arrival_Delay_in_Minutes)
```

Outliers

This section creates box plots for each numeric variable in the dataset to visualize the presence of outliers. It also compares the box plots against the target variable (satisfaction).

```
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)
```



Visualization

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

```
# 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")

  plots[[col]] = plot
}

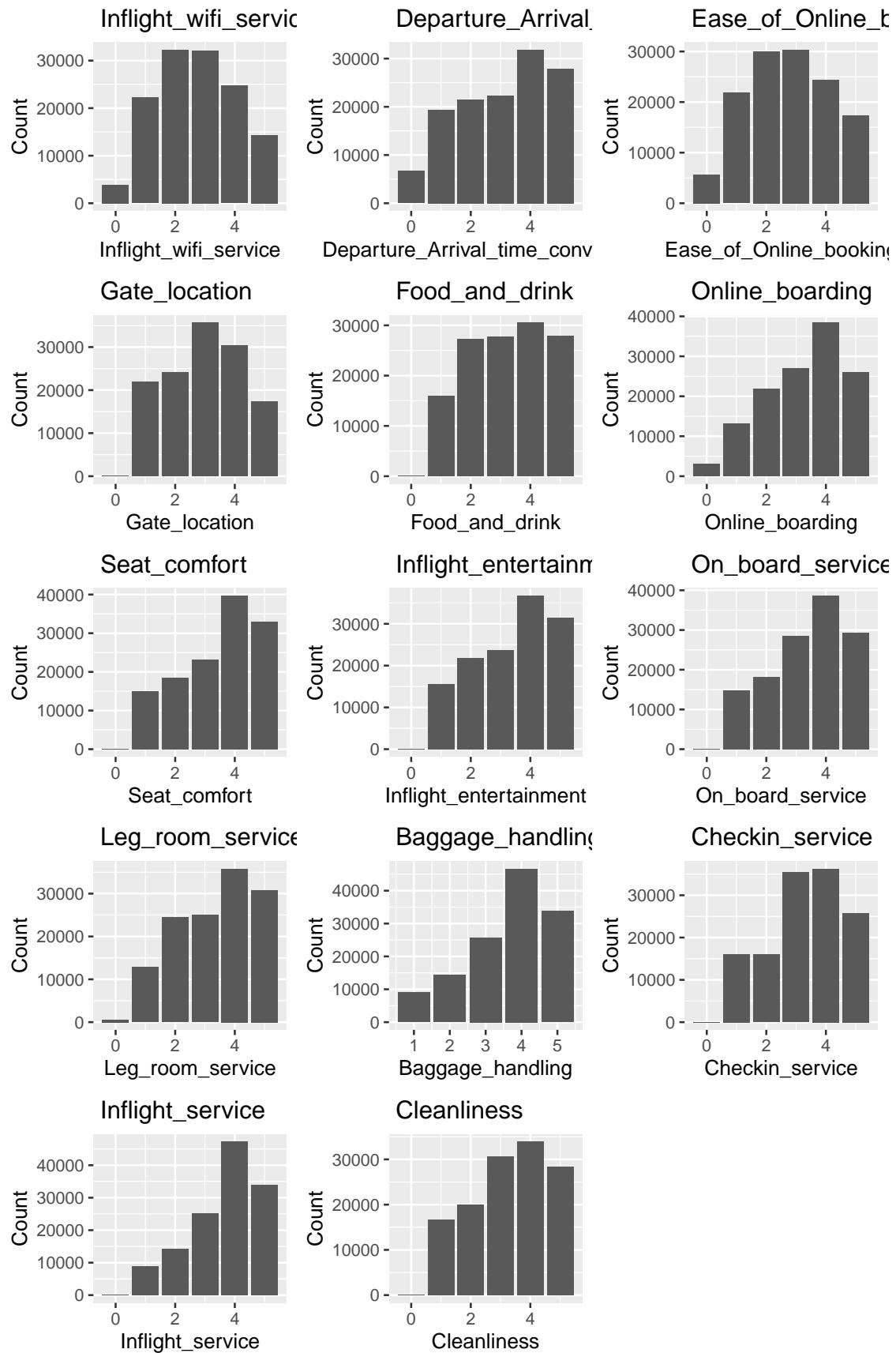
grid.arrange(grobs = plots, ncol = 2)
```



Here we plot the distribution of ratings features.

```
# plot distribution of 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_bar() +
    labs(title = paste("", col), x = col, y = "Count", bins=5)
```

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

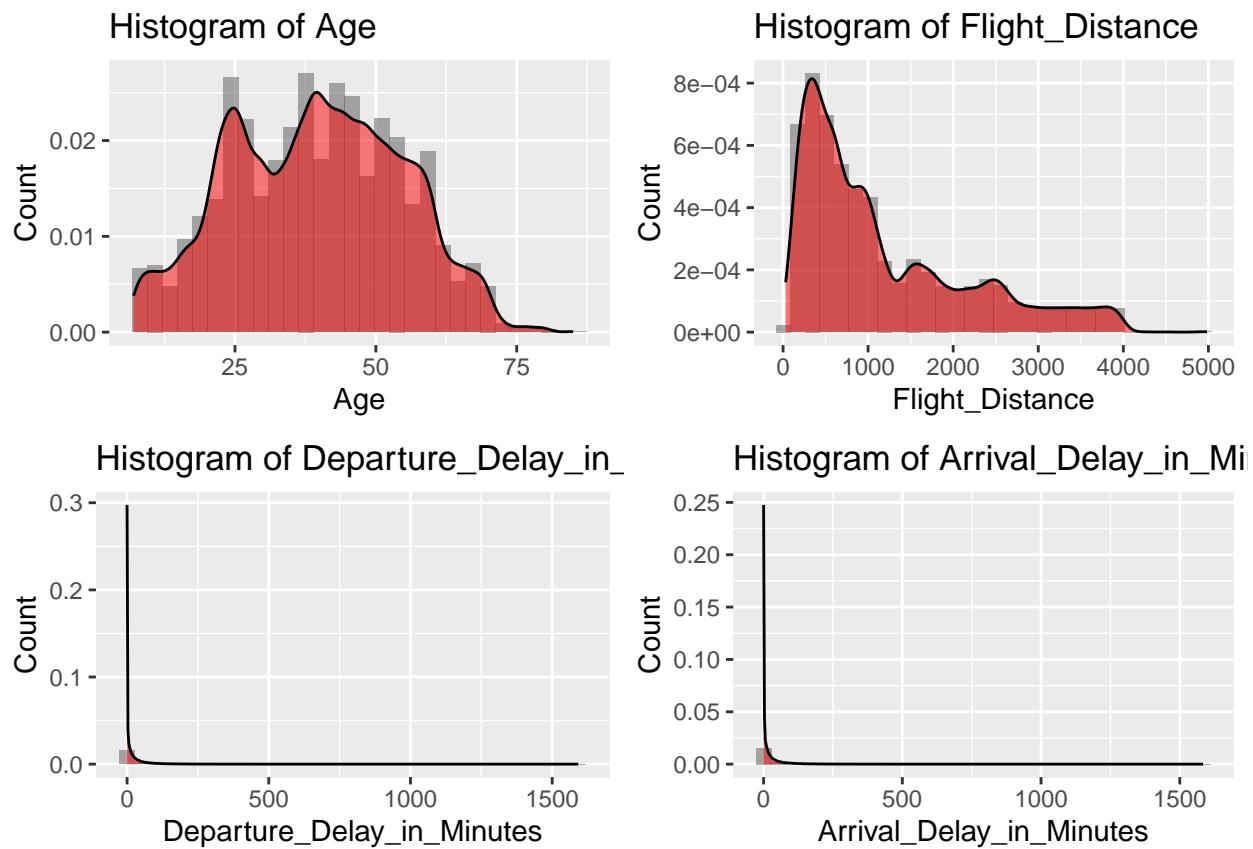


This section includes histograms to visualize the distribution of numeric variables in the dataset.

```
# 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 = 0.5, fill = "red") +
    labs(title = paste("Histogram of", col), x = col, y = "Count")

  plots[[col]] = plot
}

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



```
# plots categorical variables vs satisfaction
plots = list()
for (col in names(data)[sapply(data, is.factor)]) {
  if (col == "satisfaction") {
    next
  }
  plot = ggplot(data, aes(x = satisfaction, fill = .data[[col]])) +
    geom_bar(position = "dodge") +
    scale_fill_manual(values = rainbow(length(unique(data[[col]])))),
```

```

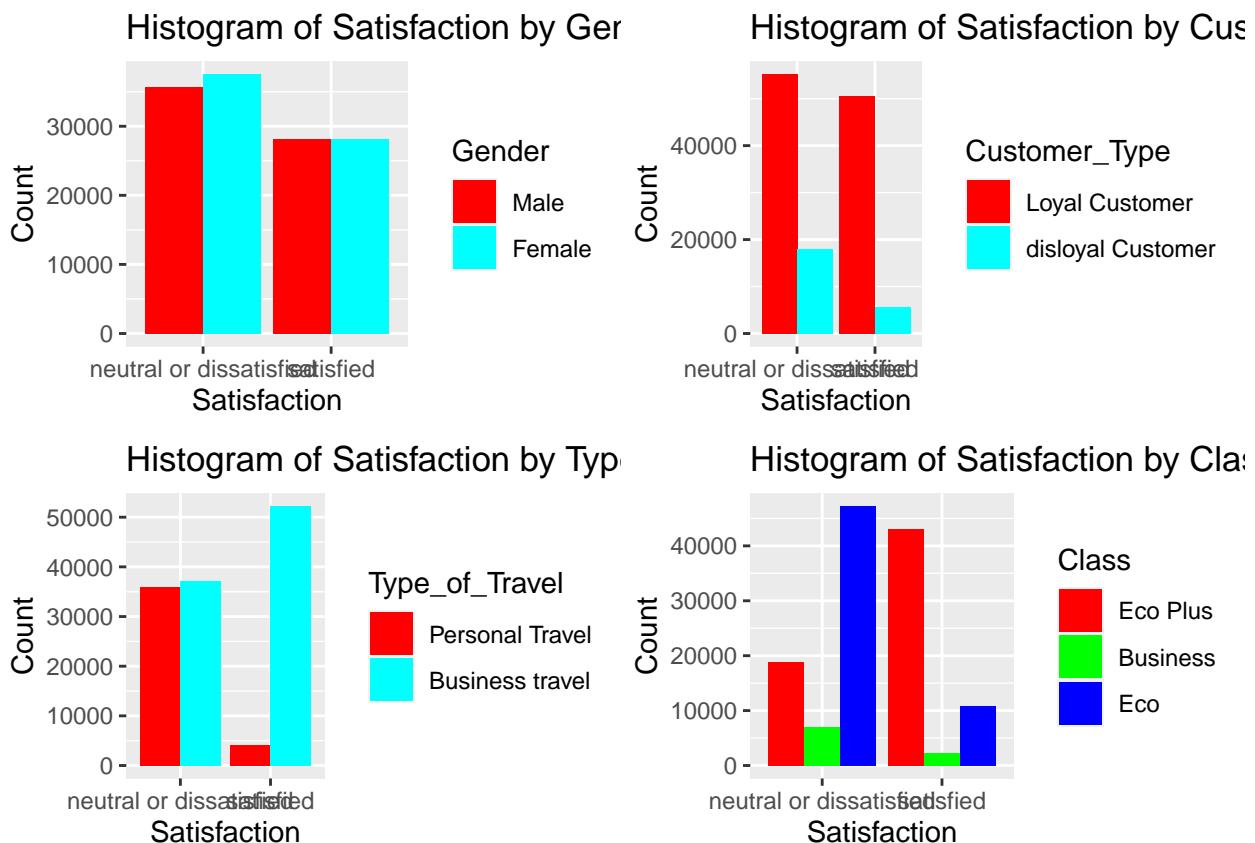
        labels = unique(data[[col]]),
        name = col) +
    labs(title = paste("Histogram of Satisfaction by", col), x = "Satisfaction", y = "Count")

plots[[col]] = plot

}

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

```



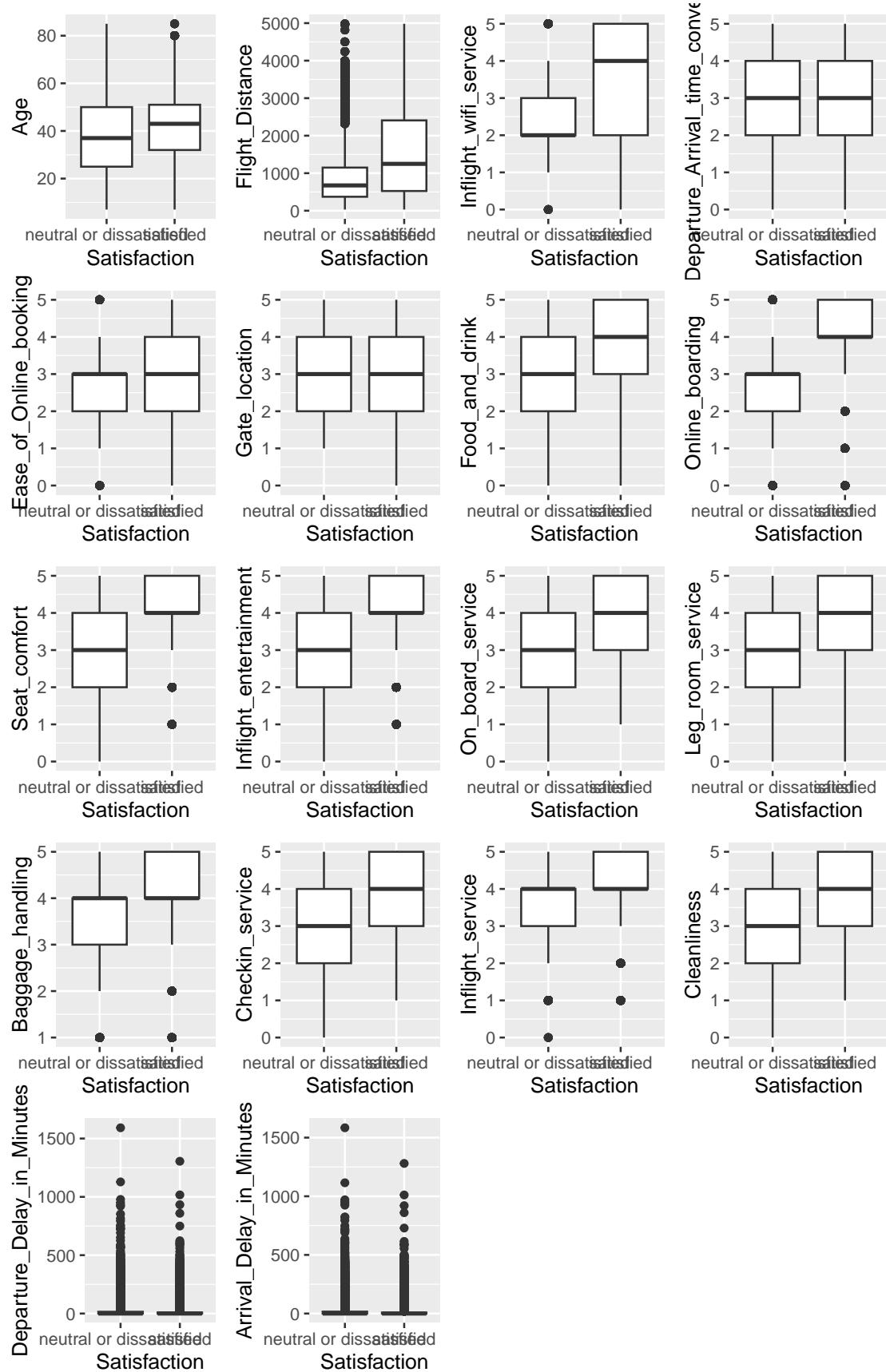
```

# plots numeric variables vs satisfaction
plots = list()
for (col in names(data)[sapply(data, is.numeric)]) {
  if (col == "satisfaction") {
    next
  }
  plot = ggplot(data, aes(x = satisfaction, y = .data[[col]])) +
    geom_boxplot() +
    labs(x = "Satisfaction", y = col)

  plots[[col]] = plot
}

grid.arrange(grobs = plots, ncol = 4)

```



Convert categorical to numerical

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

```
data$Gender = as.numeric(data$Gender) - 1  
data$Customer_Type = as.numeric(data$Customer_Type) - 1  
data>Type_of_Travel = as.numeric(data>Type_of_Travel) - 1  
data$Class = as.numeric(data$Class) - 1  
data$satisfaction = as.numeric(data$satisfaction) - 1
```

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.

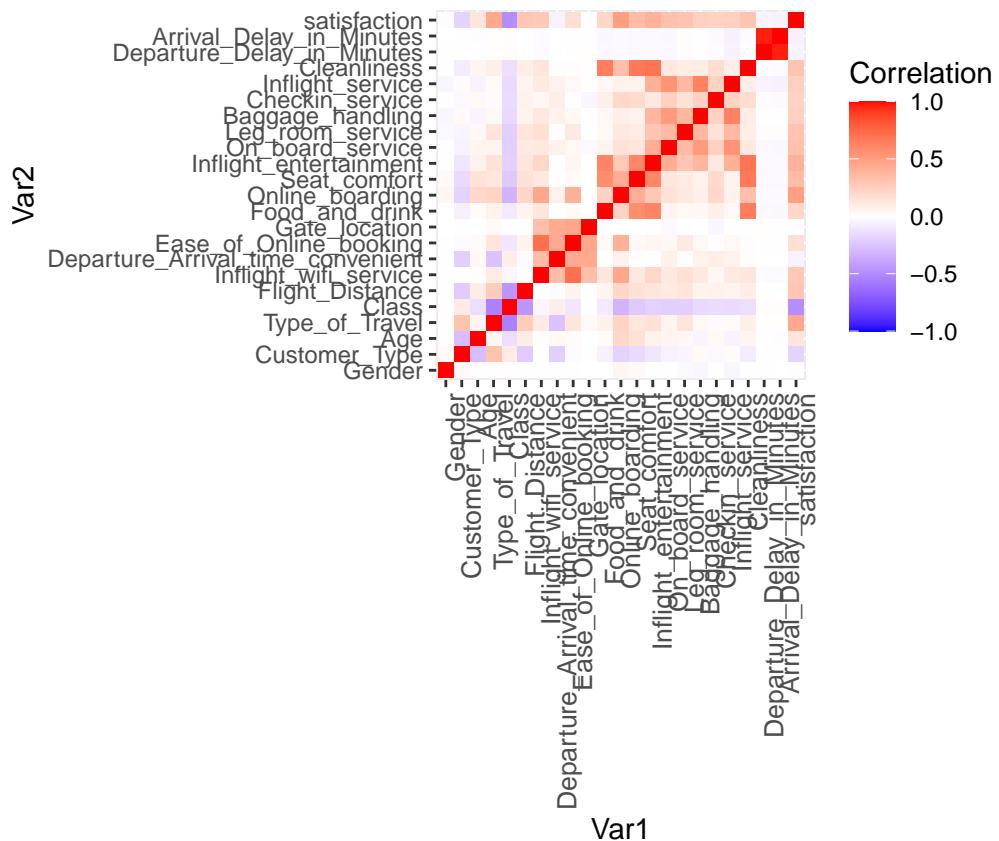
```
set.seed(123)  
train_index = sample(1:nrow(data), 0.8*nrow(data))  
# 80% of data is used for training  
train = data[train_index,]  
# 20% of data is used for testing  
test = data[-train_index,]  
  
# merge train and test data  
data = rbind(train, test)  
# save on cvs  
# write.csv(data, "data.csv")  
  
# 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  
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.

```
# 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 = 90, vjust = 1,
                                    size = 10, hjust = 1)) +
  coord_fixed()
```



```
# 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"
```

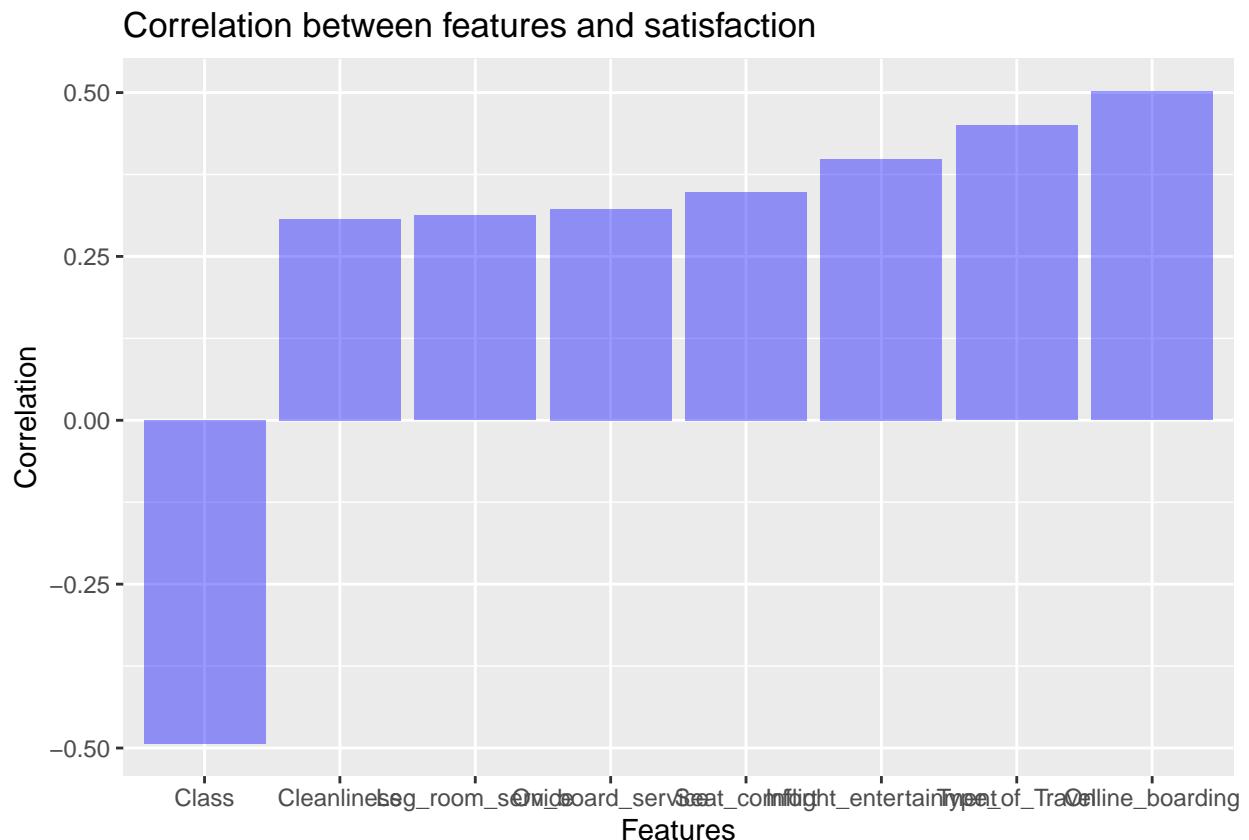
```

## [7] "Leg_room_service"           "Cleanliness"
# 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

# 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')

```



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

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

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.

```
#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 threshold 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 threshold 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
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),]

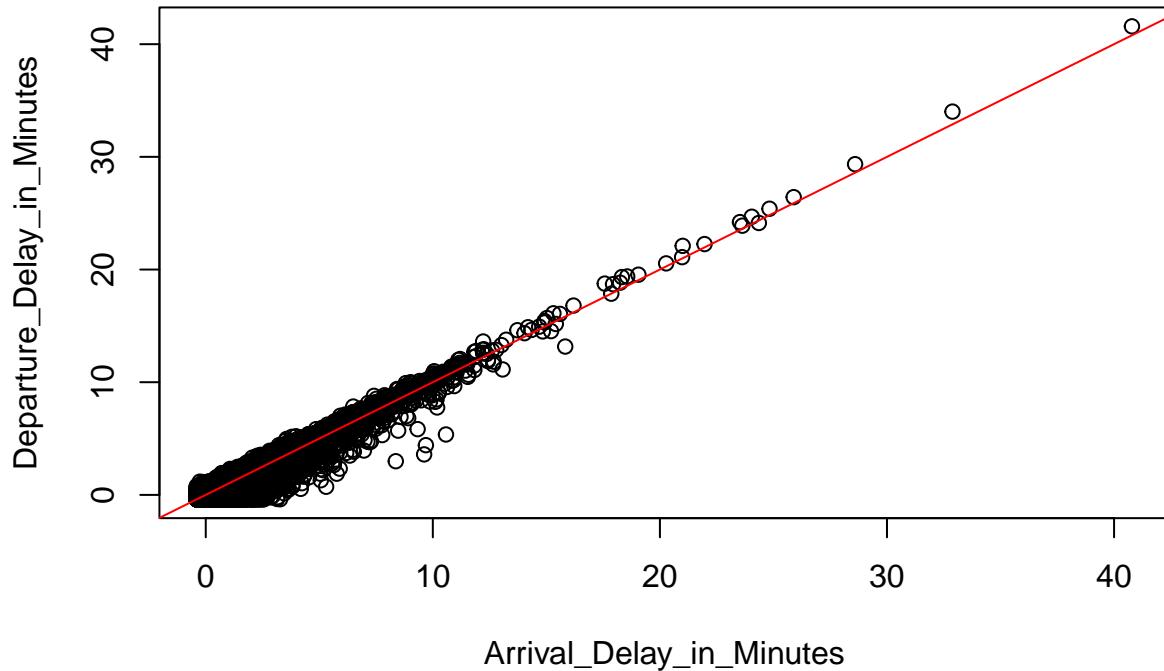
print(df_top3, n = nrow(df_top3))

## # 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

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

# 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")

```



```

# print table of type of travel by satisfaction
table(data$Type_of_Travel, data$satisfaction)

##
##          0      1
## 0 35987 4055
## 1 37238 52207

# select examples of departure delay greater than 500
examples=data[data$Departure_Delay_in_Minutes > 800,]
# and print table of satisfaction by departure delay
table(examples$satisfaction)

##
## 0 1
## 8 4

# count the number of examples with departure delay = 0
sum(data$Departure_Delay_in_Minutes > 0)

## [1] 56278
sum(data$Departure_Delay_in_Minutes <= 0)

## [1] 73209
sum(data$Arrival_Delay_in_Minutes > 0)

## [1] 56734

```

```

sum(data$Arrival_Delay_in_Minutes <= 0)

## [1] 72753

summary(data)

##      Gender      Customer_Type        Age      Type_of_Travel
## Min.   :0.0000  Min.   :0.0000  Min.   : 7.00  Min.   :0.0000
## 1st Qu.:0.0000  1st Qu.:0.0000  1st Qu.:27.00  1st Qu.:0.0000
## Median :1.0000  Median :0.0000  Median :40.00  Median :1.0000
## Mean   :0.5074  Mean   :0.1831  Mean   :39.43  Mean   :0.6908
## 3rd Qu.:1.0000  3rd Qu.:0.0000  3rd Qu.:51.00  3rd Qu.:1.0000
## Max.   :1.0000  Max.   :1.0000  Max.   :85.00  Max.   :1.0000
##      Class      Flight_Distance Inflight_wifi_service
## Min.   :0.0000  Min.   : 31    Min.   :0.000
## 1st Qu.:0.0000  1st Qu.: 414   1st Qu.:2.000
## Median :1.0000  Median : 844   Median :3.000
## Mean   :0.9701  Mean   :1190   Mean   :2.729
## 3rd Qu.:2.0000  3rd Qu.:1744   3rd Qu.:4.000
## Max.   :2.0000  Max.   :4983   Max.   :5.000
##      Departure_Arrival_time_convenient Ease_of_Online_booking Gate_location
## 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   :3.057          Mean   :2.757          Mean   :2.977
## 3rd Qu.:4.000          3rd Qu.:4.000          3rd Qu.:4.000
## Max.   :5.000          Max.   :5.000          Max.   :5.000
##      Food_and_drink  Online_boarding  Seat_comfort  Inflight_entertainment
## 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 :4.000  Median :4.000
## Mean   :3.205  Mean   :3.253  Mean   :3.442  Mean   :3.358
## 3rd Qu.:4.000  3rd Qu.:4.000  3rd Qu.:5.000  3rd Qu.:4.000
## Max.   :5.000  Max.   :5.000  Max.   :5.000  Max.   :5.000
##      On_board_service Leg_room_service Baggage_handling Checkin_service
## Min.   :0.000  Min.   :0.000  Min.   :1.000  Min.   :0.000
## 1st Qu.:2.000  1st Qu.:2.000  1st Qu.:3.000  1st Qu.:3.000
## Median :4.000  Median :4.000  Median :4.000  Median :3.000
## Mean   :3.383  Mean   :3.351  Mean   :3.632  Mean   :3.306
## 3rd Qu.:4.000  3rd Qu.:4.000  3rd Qu.:5.000  3rd Qu.:4.000
## Max.   :5.000  Max.   :5.000  Max.   :5.000  Max.   :5.000
##      Inflight_service Cleanliness  Departure_Delay_in_Minutes
## Min.   :0.000  Min.   :0.000  Min.   : 0.00
## 1st Qu.:3.000  1st Qu.:2.000  1st Qu.: 0.00
## Median :4.000  Median :3.000  Median : 0.00
## Mean   :3.642  Mean   :3.286  Mean   : 14.64
## 3rd Qu.:5.000  3rd Qu.:4.000  3rd Qu.: 12.00
## Max.   :5.000  Max.   :5.000  Max.   :1592.00
##      Arrival_Delay_in_Minutes satisfaction
## Min.   : 0.00      Min.   :0.0000
## 1st Qu.: 0.00      1st Qu.:0.0000
## Median : 0.00      Median :0.0000
## Mean   : 15.09     Mean   :0.4345
## 3rd Qu.: 13.00     3rd Qu.:1.0000

```

Max. :1584.00 Max. :1.0000