EDA

Süleyman Erim, Giacomo Schiavo, Mattia Varagnolo

2023-07-08

Introduction to data

```
# 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)
```

VARIABLE DESCRIPTION

```
# 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
```

```r
# 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

Here we create factors from categorical variables

```r
# 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

```
# 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)))
```

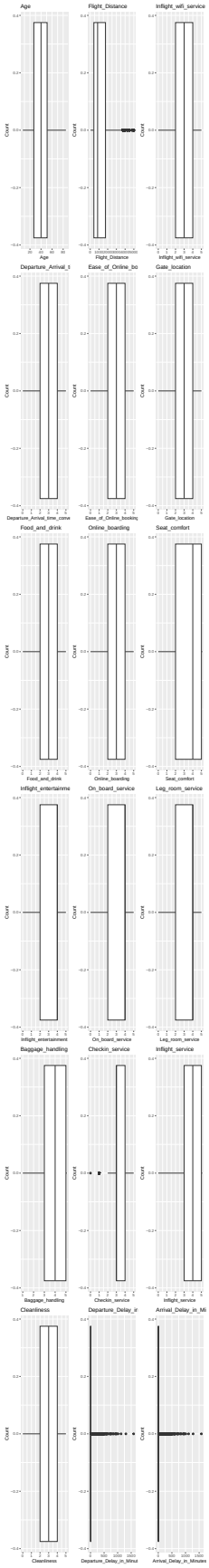
```
##
##      FALSE       TRUE
## 0.99697413 0.00302587
```

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

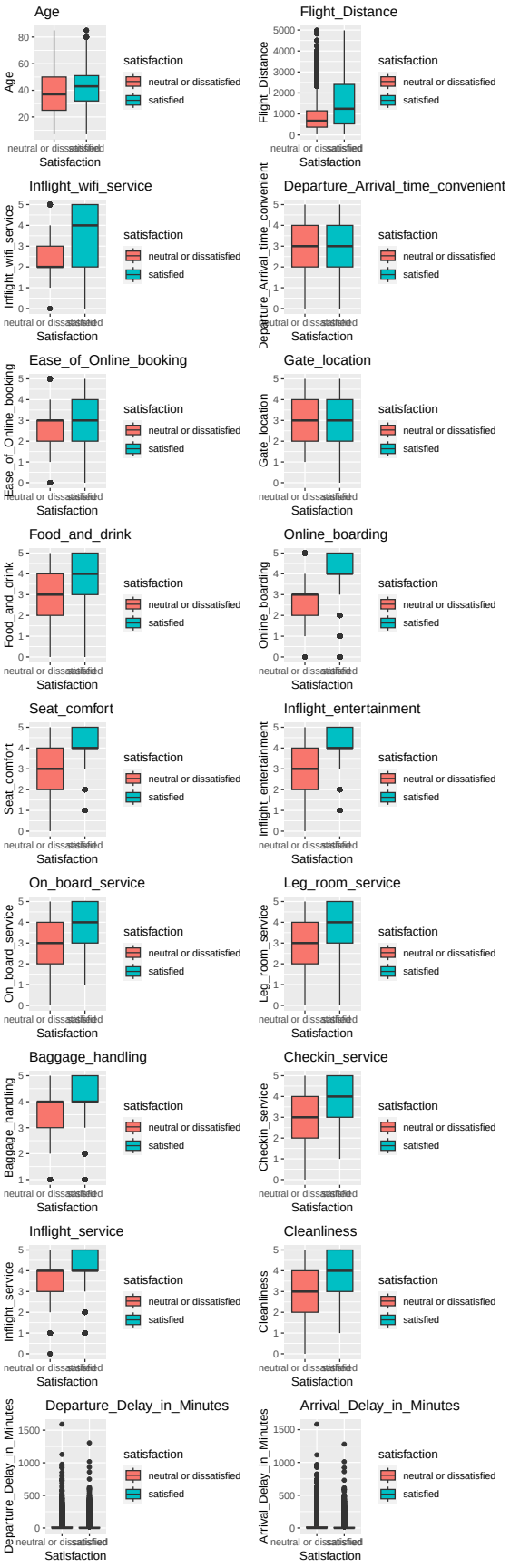
It turns out that Arrival_Delay_in_Minutes has only 3% of na values so we drop this examples.

OUTLIERS

```r
# plot boxplot of numeric variables
plots = list()
for (col in names(data)[sapply(data, is.numeric)]) {
  plot = ggplot(data, aes(x = .data[[col]])) +
  geom_boxplot() +
  labs(title = col, x = col, y = "Count")
  plots[[col]] = plot
}

grid.arrange(grobs = plots, ncol = 3)
```

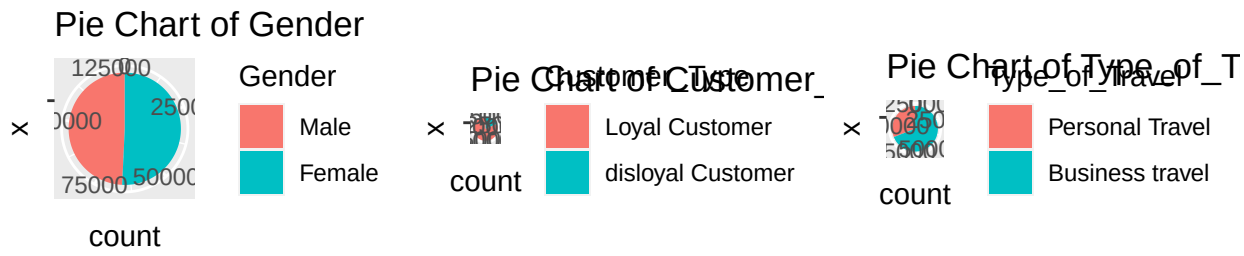

Age
Flight_Distance
Inflight_wifi_service
Departure_Arrival_t
Ease_of_Online_bo
Gate_location
Food_and_drink
Online_boarding
Seat_comfort
Inflight_entertainme
On_board_service
Leg_room_service
Baggage_handling
Checkin_service
Inflight_service
Cleanliness
Departure_Delay_in
Arrival_Delay_in_Mi
Count

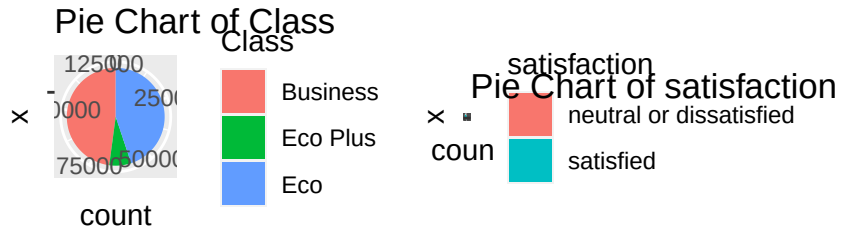
```r
# plot boxplot against satisfaction with colors
plots = list()
for (col in names(data)[sapply(data, is.numeric)]) {
  plot = ggplot(data, aes(x = satisfaction, y = .data[[col]], fill = satisfaction)) +
  geom_boxplot() +
  labs(title = col, x = "Satisfaction", y = col)
  plots[[col]] = plot
}

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


Age
Age
Satisfaction
satisfaction
neutral or dissatisfied
satisfied
Flight_Distance
Flight_Distance
Satisfaction
satisfaction
neutral or dissatisfied
satisfied
Inflight_wifi_service
Inflight_wifi_service
Satisfaction
satisfaction
neutral or dissatisfied
satisfied
Departure_Arrival_time_convenient
Departure_Arrival_time_convenient
Satisfaction
satisfaction
neutral or dissatisfied
satisfied
Ease_of_Online_booking
Ease_of_Online_booking
Satisfaction
satisfaction
neutral or dissatisfied
satisfied
Gate_location
Gate_location
Satisfaction
satisfaction
neutral or dissatisfied
satisfied
Food_and_drink
Food_and_drink
Satisfaction
satisfaction
neutral or dissatisfied
satisfied
Online_boarding
Online_boarding
Satisfaction
satisfaction
neutral or dissatisfied
satisfied
Seat_comfort
Seat_comfort
Satisfaction
satisfaction
neutral or dissatisfied
satisfied
Inflight_entertainment
Inflight_entertainment
Satisfaction
satisfaction
neutral or dissatisfied
satisfied
On_board_service
On_board_service
Satisfaction
satisfaction
neutral or dissatisfied
satisfied
Leg_room_service
Leg_room_service
Satisfaction
satisfaction
neutral or dissatisfied
satisfied
Baggage_handling
Baggage_handling
Satisfaction
satisfaction
neutral or dissatisfied
satisfied
Checkin_service
Checkin_service
Satisfaction
satisfaction
neutral or dissatisfied
satisfied
Inflight_service
Inflight_service
Satisfaction
satisfaction
neutral or dissatisfied
satisfied
Cleanliness
Cleanliness
Satisfaction
satisfaction
neutral or dissatisfied
satisfied
Departure_Delay_in_Minutes
Departure_Delay_in_Minutes
Satisfaction
satisfaction
neutral or dissatisfied
satisfied
Arrival_Delay_in_Minutes
Arrival_Delay_in_Minutes
Satisfaction
satisfaction
neutral or dissatisfied
satisfied

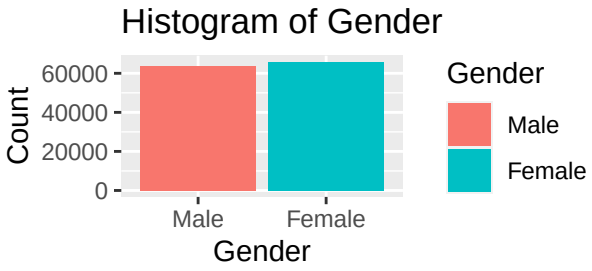
VISUALIZATION

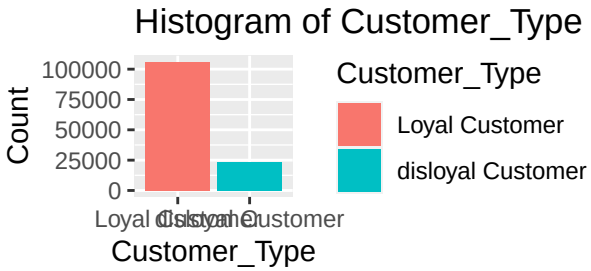
```
# plot pie chart for each variable
plots = list()
for (col in names(data)[sapply(data, is.factor)]) {
  plot = ggplot(data, aes(x = "", fill = .data[[col]])) +
  geom_bar(width = 1) +
  coord_polar("y", start = 0) +
  labs(title = paste("Pie Chart of", col))
  plots[[col]] = plot
}

grid.arrange(grobs = plots, ncol = 3)
```

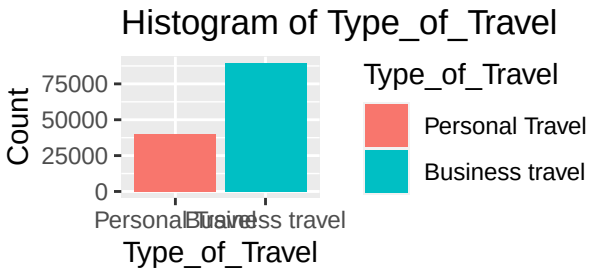


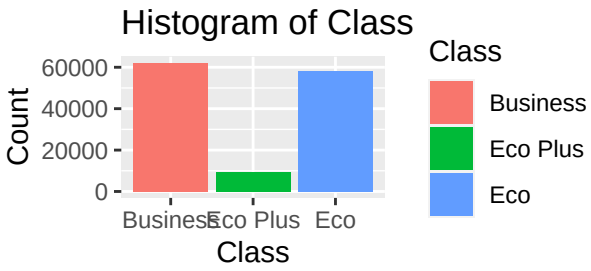


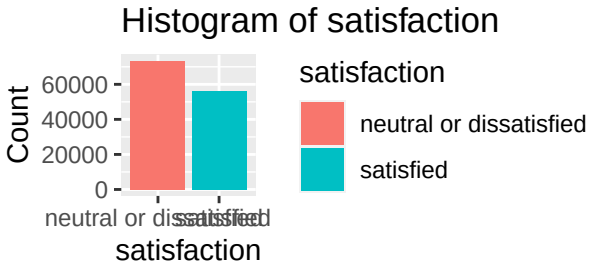

```
# plot distribution of categorical variables
plots = list()
for (col in names(data)[sapply(data, is.factor)]) {
  plot = ggplot(data, aes(x = .data[[col]], fill = .data[[col]])) +
  geom_bar() +
  labs(title = paste("Histogram of", col), x = col, y = "Count")

  plots[[col]] = plot
}

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



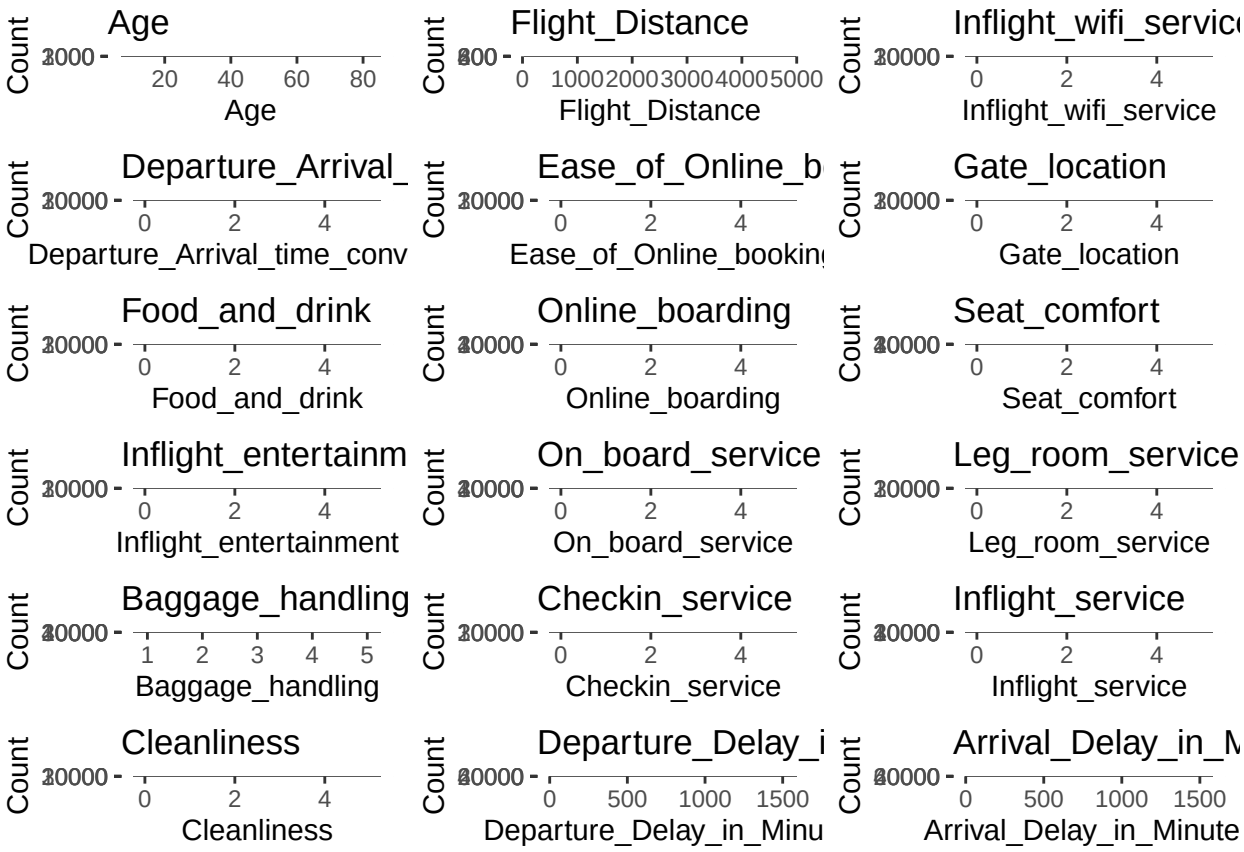




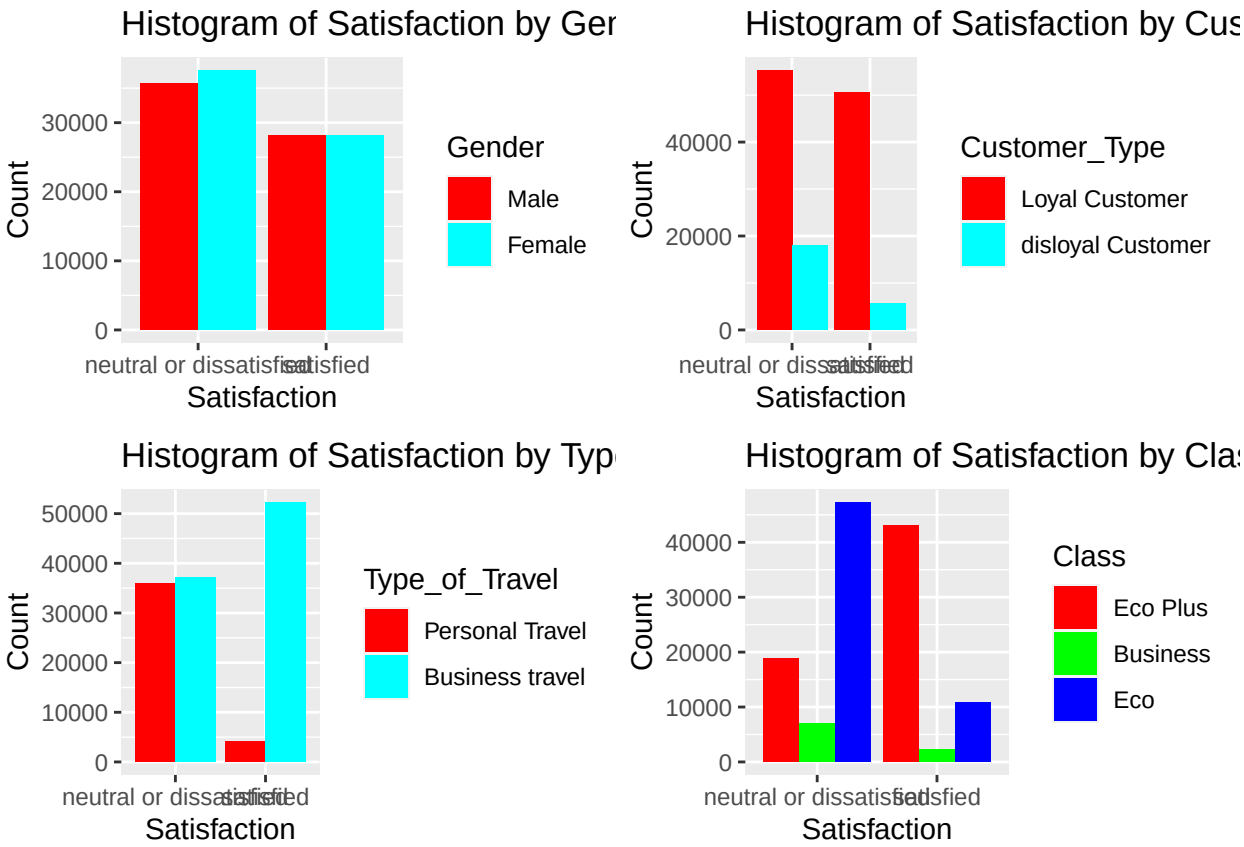




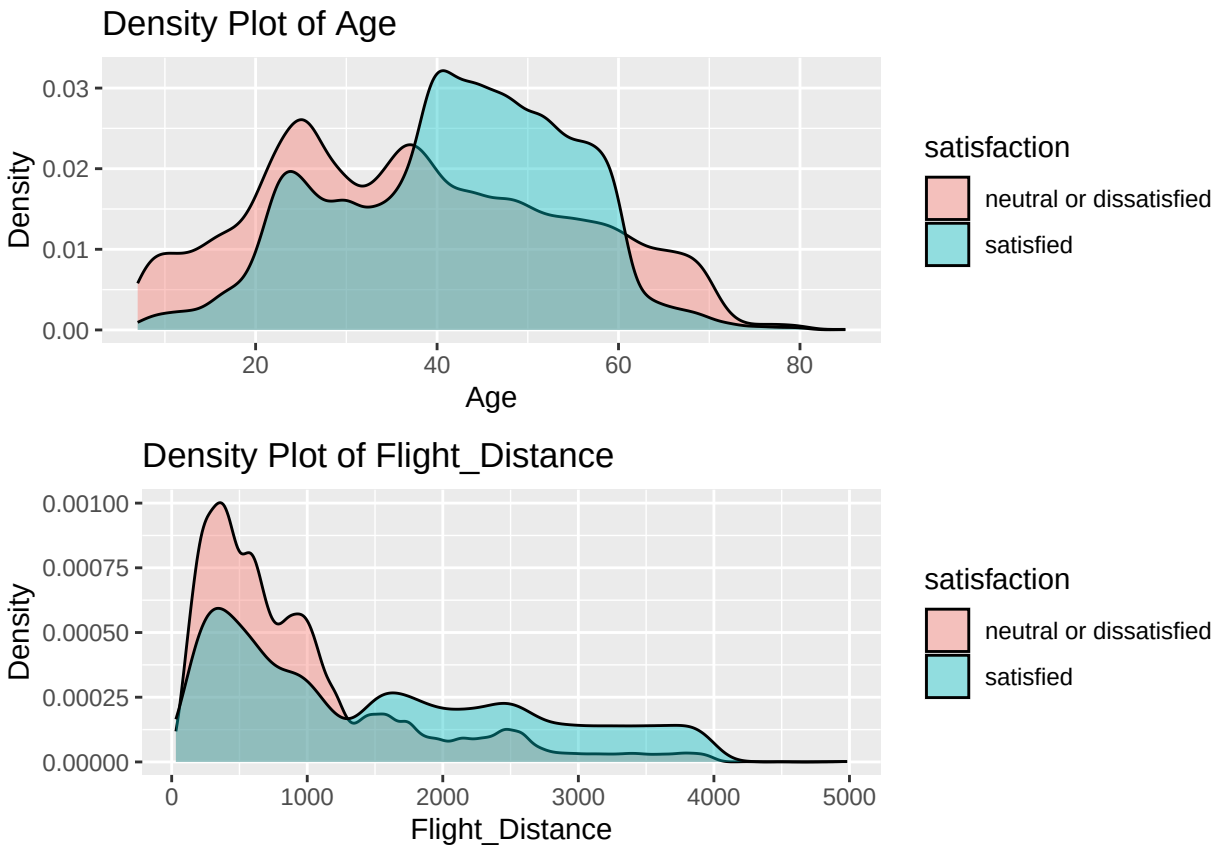

```
# plot distribution of numeric variables
plots = list()
for (col in names(data)[sapply(data, is.numeric)]) {
  plot = ggplot(data, aes(x = .data[[col]])) +
  geom_histogram(binwidth = 0.5) +
  labs(title = col, x = col, y = "Count")
  plots[[col]] = plot
}

grid.arrange(grobs = plots, ncol = 3)
```



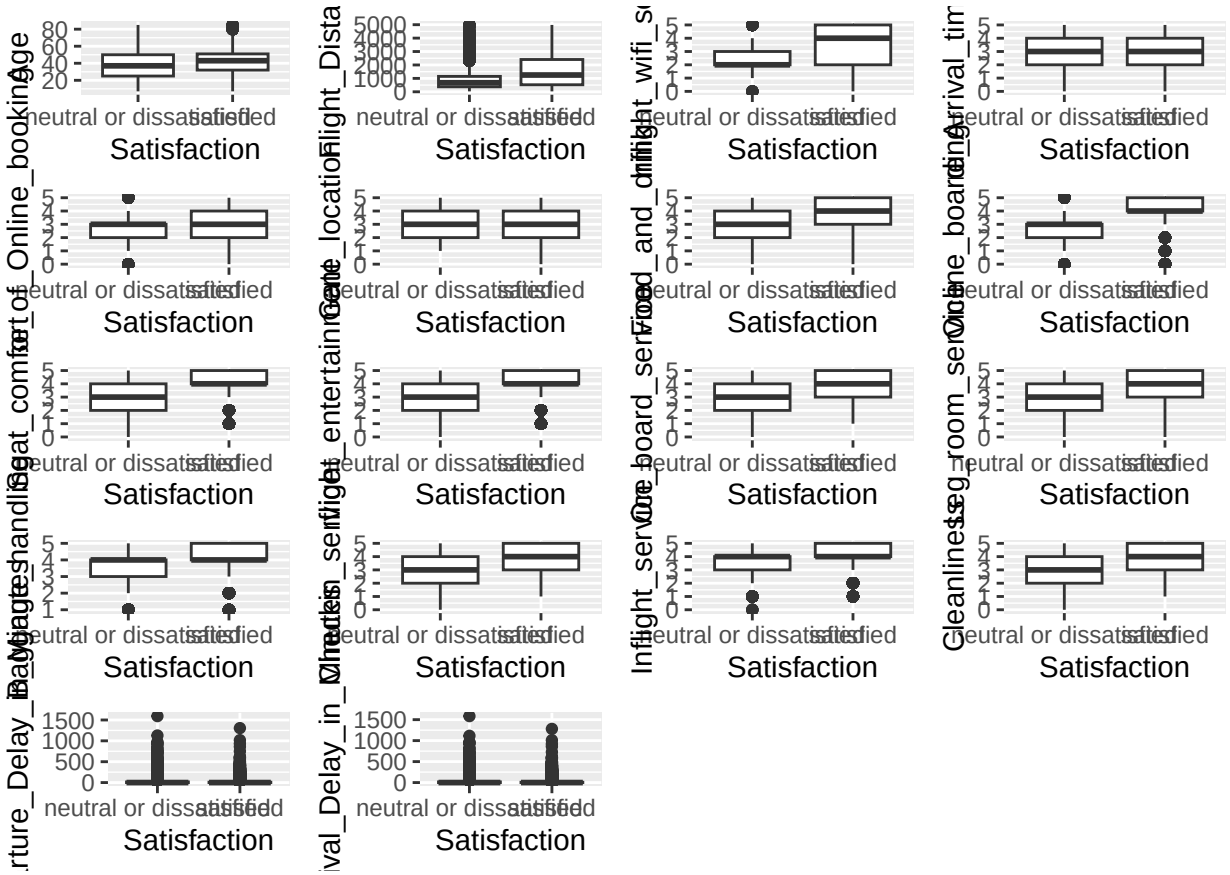

```r
# plots categorical variables vs satisfaction
plots = list()
for (col in names(data)[sapply(data, is.factor)]) {
  if (col == "satisfaction") {
    next
  }
  plot = ggplot(data, aes(x = satisfaction, fill = .data[[col]])) +
  geom_bar(position = "dodge") +
  scale_fill_manual(values = rainbow(length(unique(data[[col]]))),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




```r
# Create density plots for Age and Flight_Distance
plots = list()
for (col in c("Age", "Flight_Distance")) {
  plot = ggplot(data, aes(x = .data[[col]], fill = satisfaction)) +
  geom_density(alpha = 0.4) +
  labs(title = paste("Density Plot of", col), x = col, y = "Density")
  plots[[col]] = plot
}

# Arrange the density plots in a grid
grid.arrange(grobs = plots)
```




```r
# 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 NUMERIC

```
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

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

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

TRAIN TEST SPLIT

```
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,]
```

```r
# 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))
```

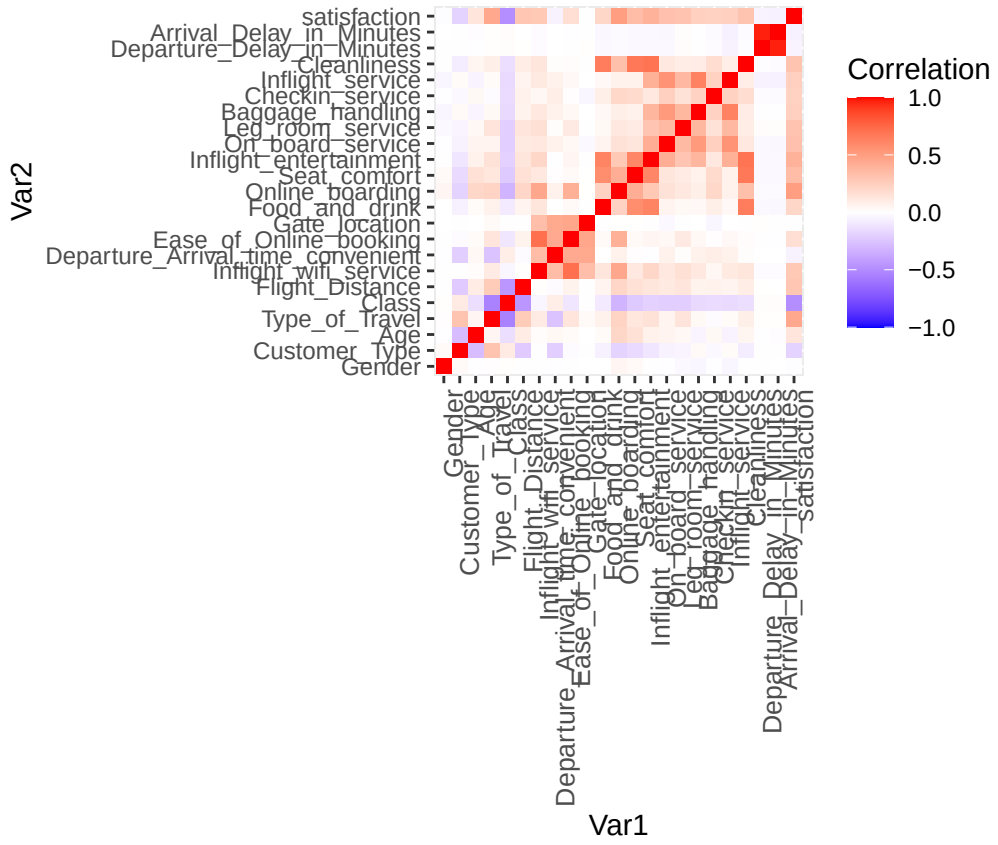
```
##
##         0         1
## 0.5668845 0.4331155
```

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

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

CORRELATION MATRIX

```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 = 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
```

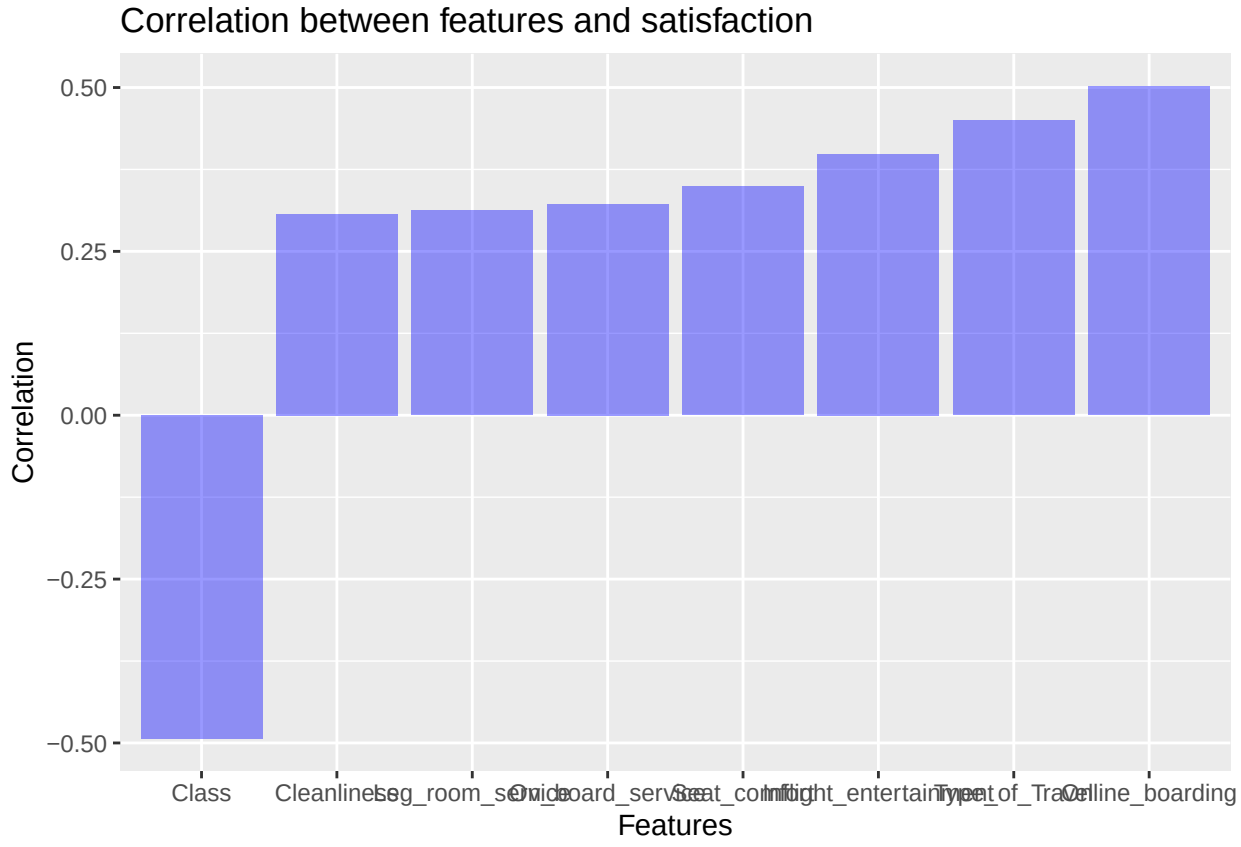
```
## [1] "Type_of_Travel"         "Class"                  "Online_boarding"
## [4] "Seat_comfort"           "Inflight_entertainment" "On_board_service"
## [7] "Leg_room_service"       "Cleanliness"
```

```
# 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
```

```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')
```



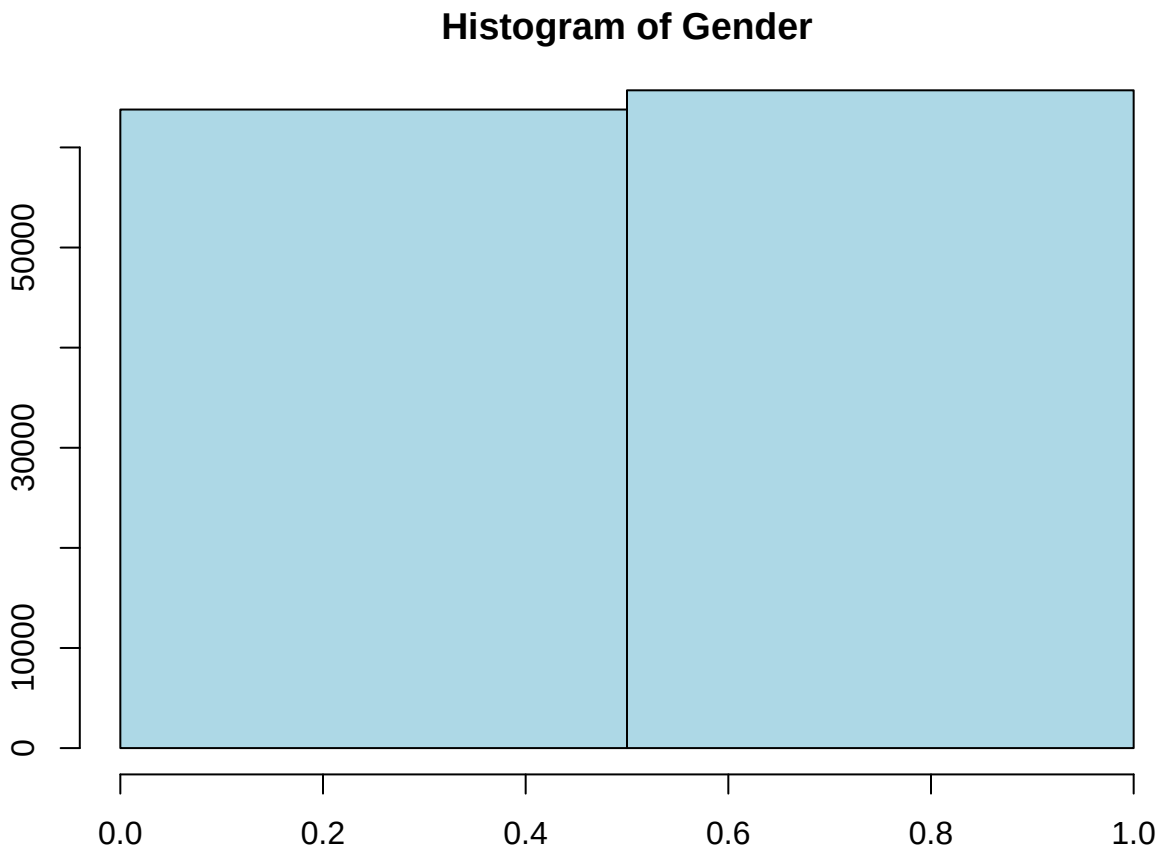

```r
par(mfrow = c(1, 1))
```

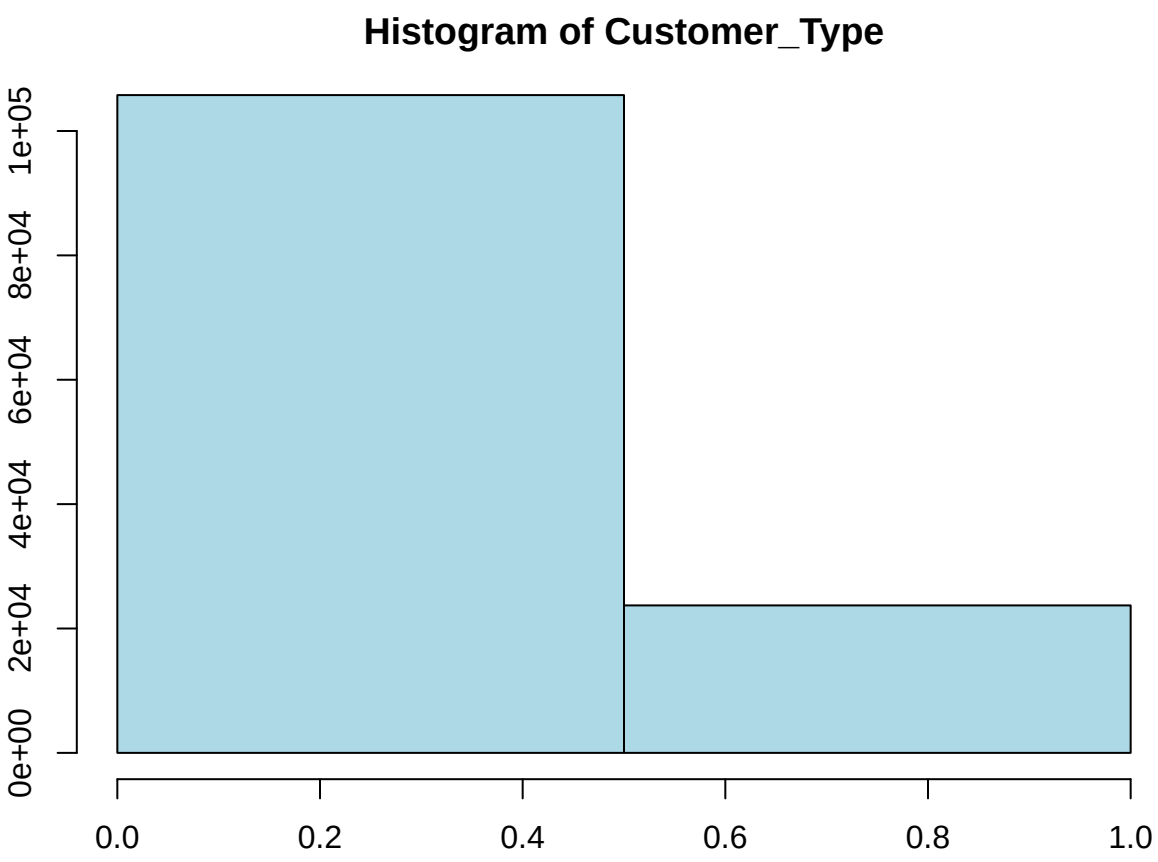
```r
#save on cvs
# write.csv(correlations, file = "correlations.csv")
```

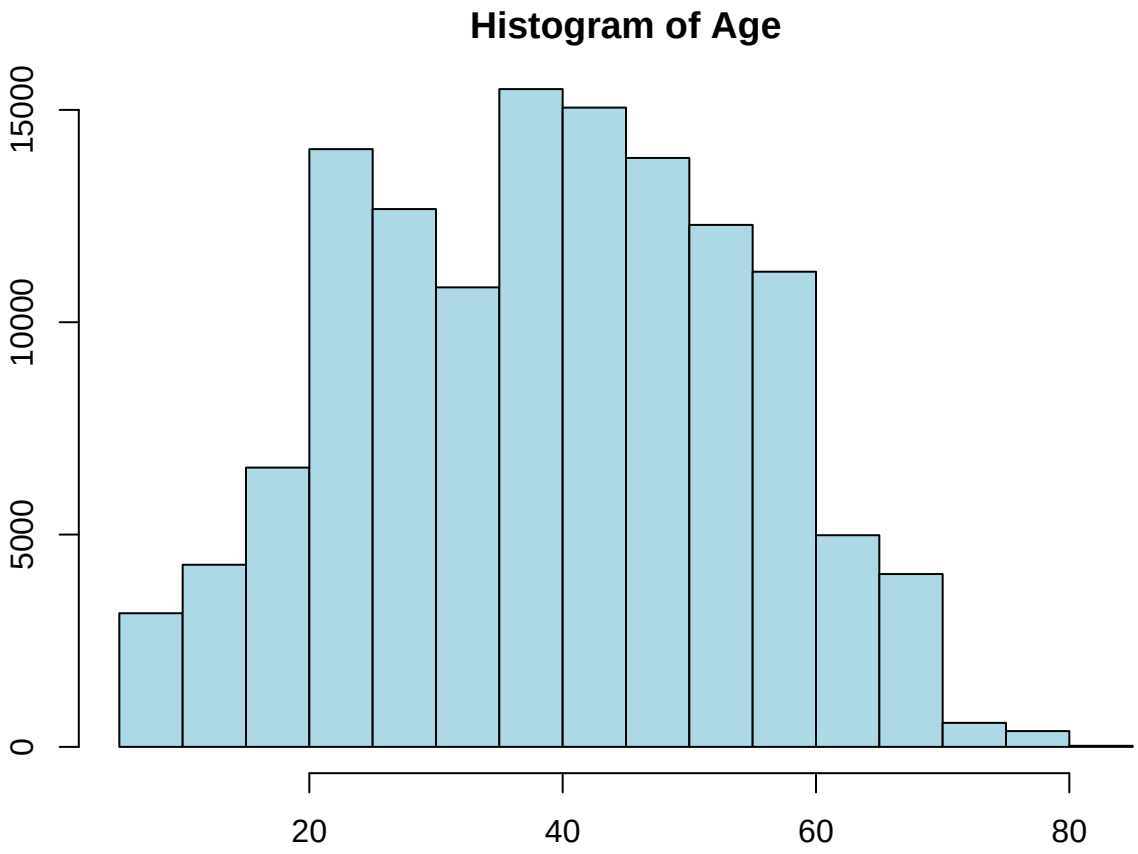
PLOT ALL DISTRIBUTIONS (with numerical categories)

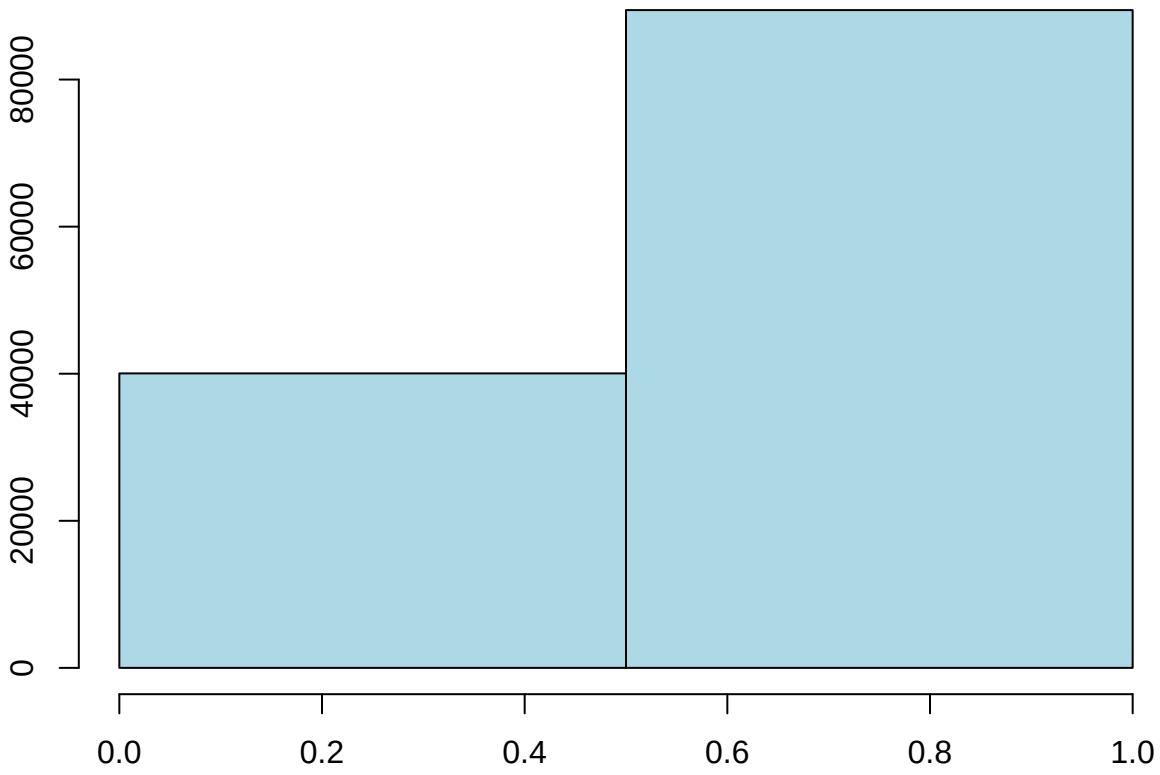
```r
sat <- data$satisfaction
features_names <- names(data)

num_cols <- 3
num_rows <- ceiling(length(features_names) / num_cols)

# Set smaller figure margins
par(mar = c(2, 2, 2, 2))  # Adjust the margin values as per your preference

# Loop through each feature and plot the density of satisfied and dissatisfied customers
for (col in features_names) {
```

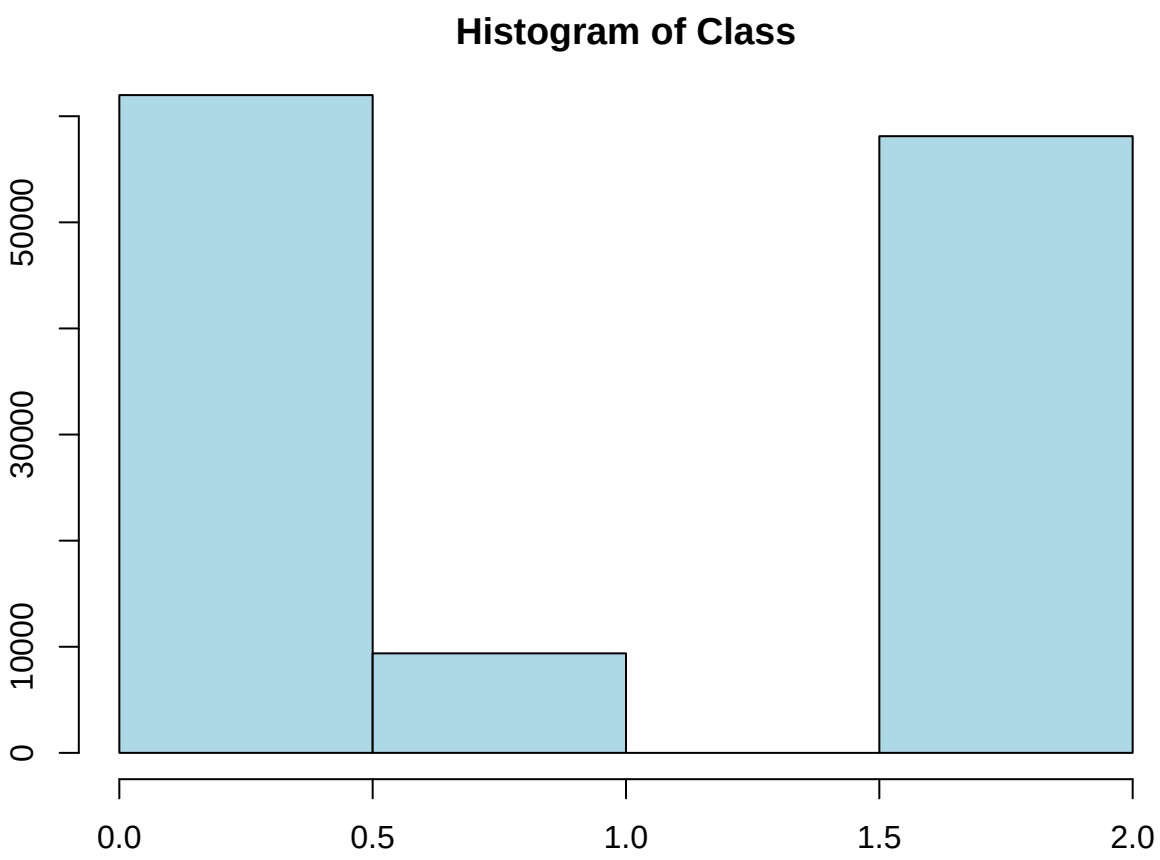
```r
  # Calculate the number of breaks
  num_breaks <- length(unique(data[[col]]))
  num_breaks <- min(num_breaks, 20)

  # Create a new plot for each feature
  hist(data[[col]], breaks = num_breaks,
       main = paste("Histogram of", col), xlab = col, ylab = "Frequency",
       col = "lightblue"
  )
}
```

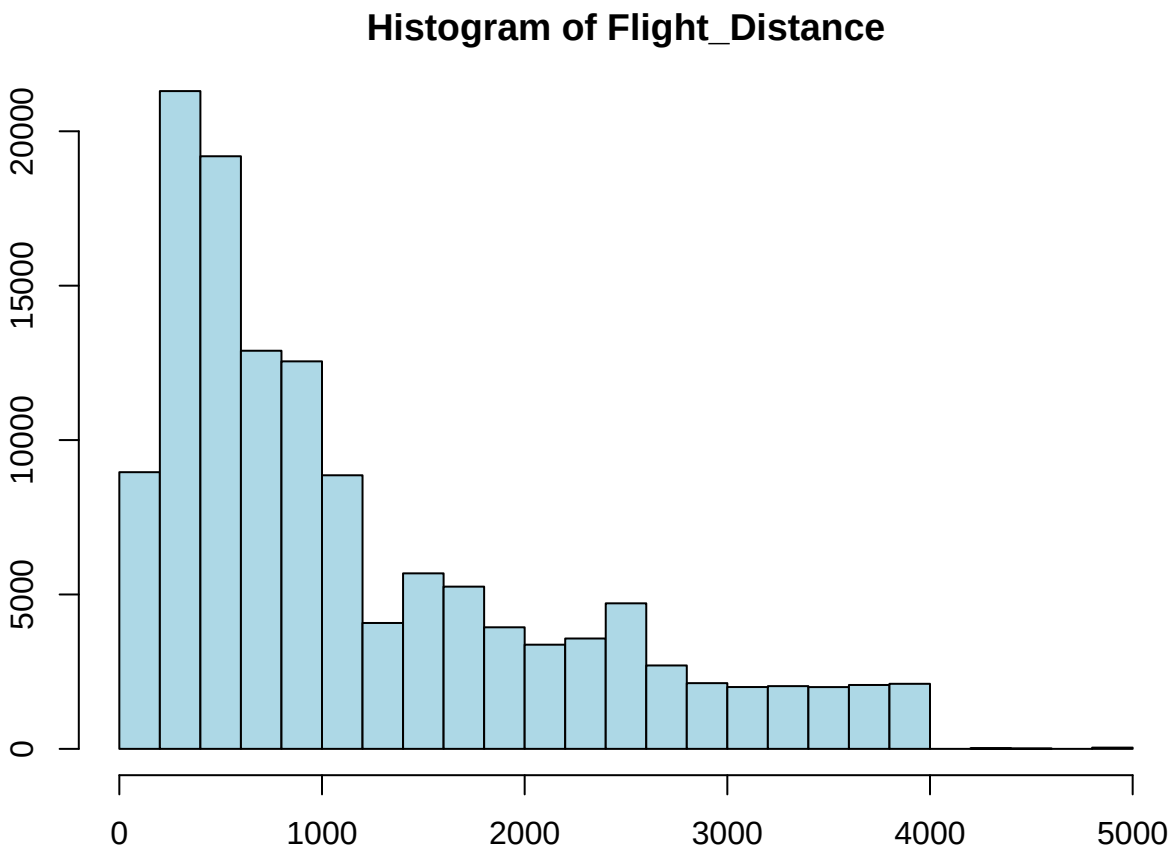


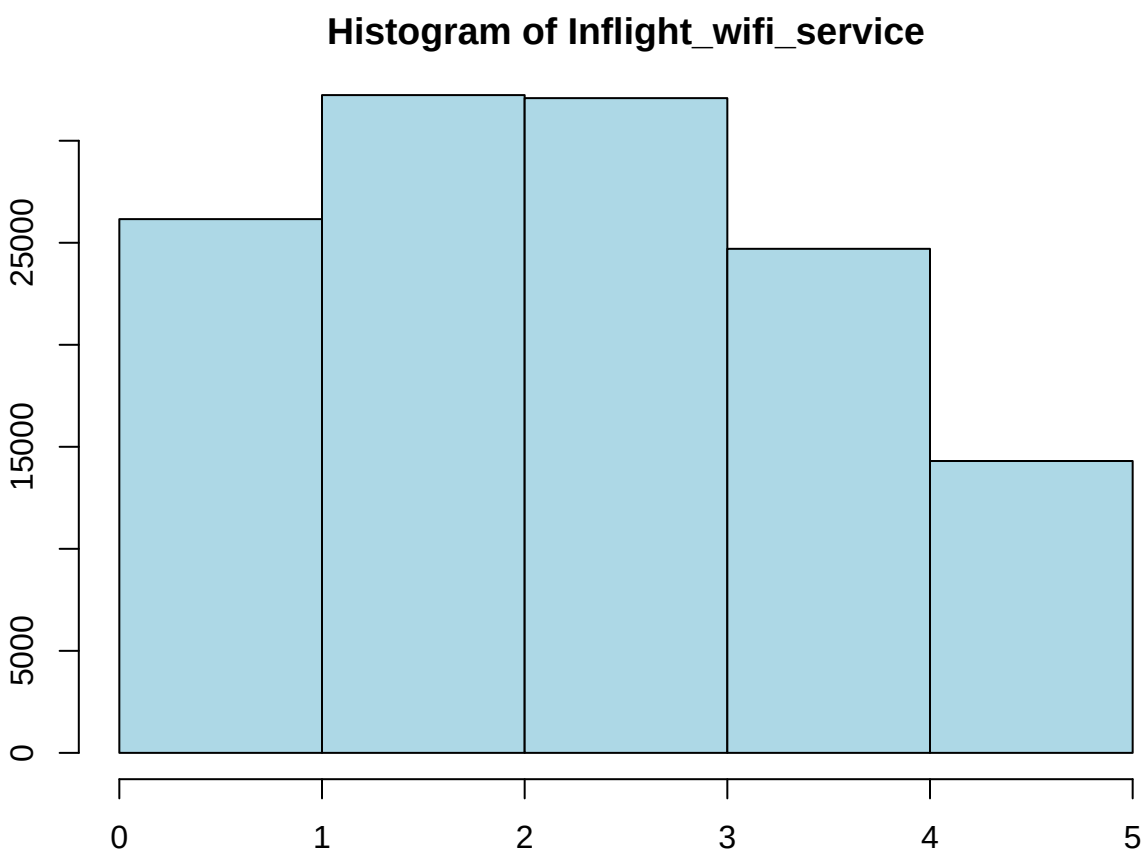

Histogram of Customer_Type
0e+00
2e+04
4e+04
6e+04
8e+04
1e+05
0.0
0.2
0.4
0.6
0.8
1.0

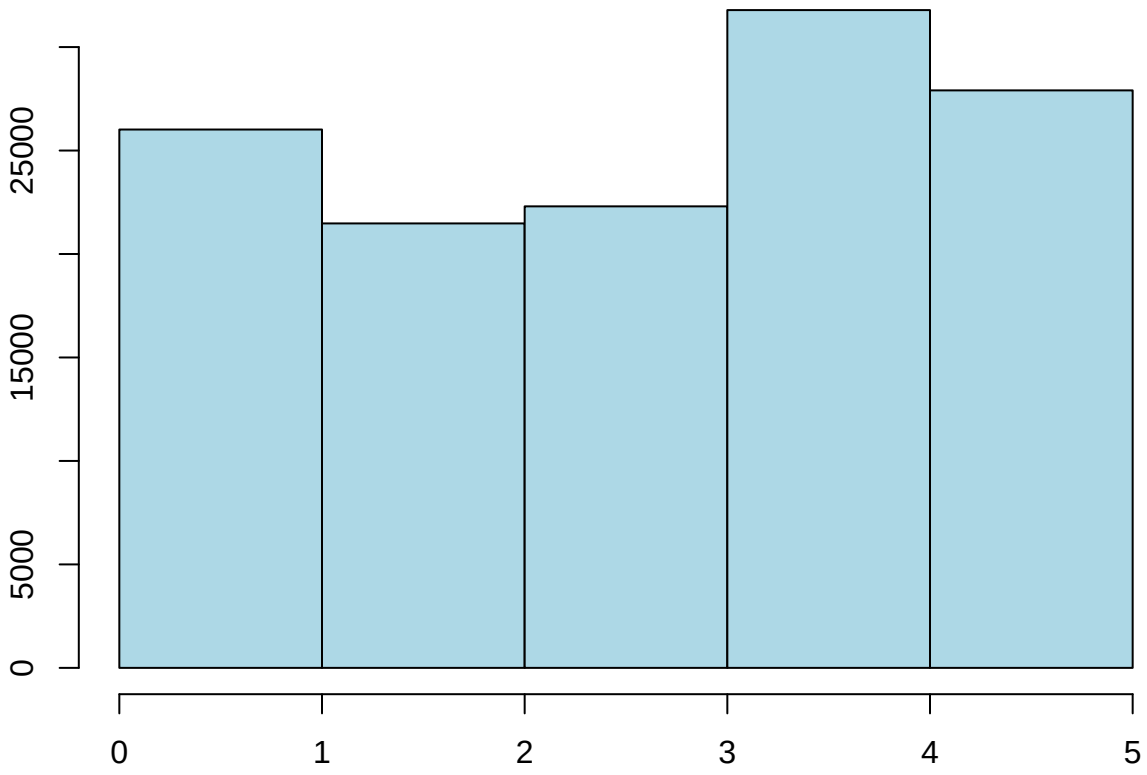

Histogram of Age
15000
10000
5000
0
20
40
60
80

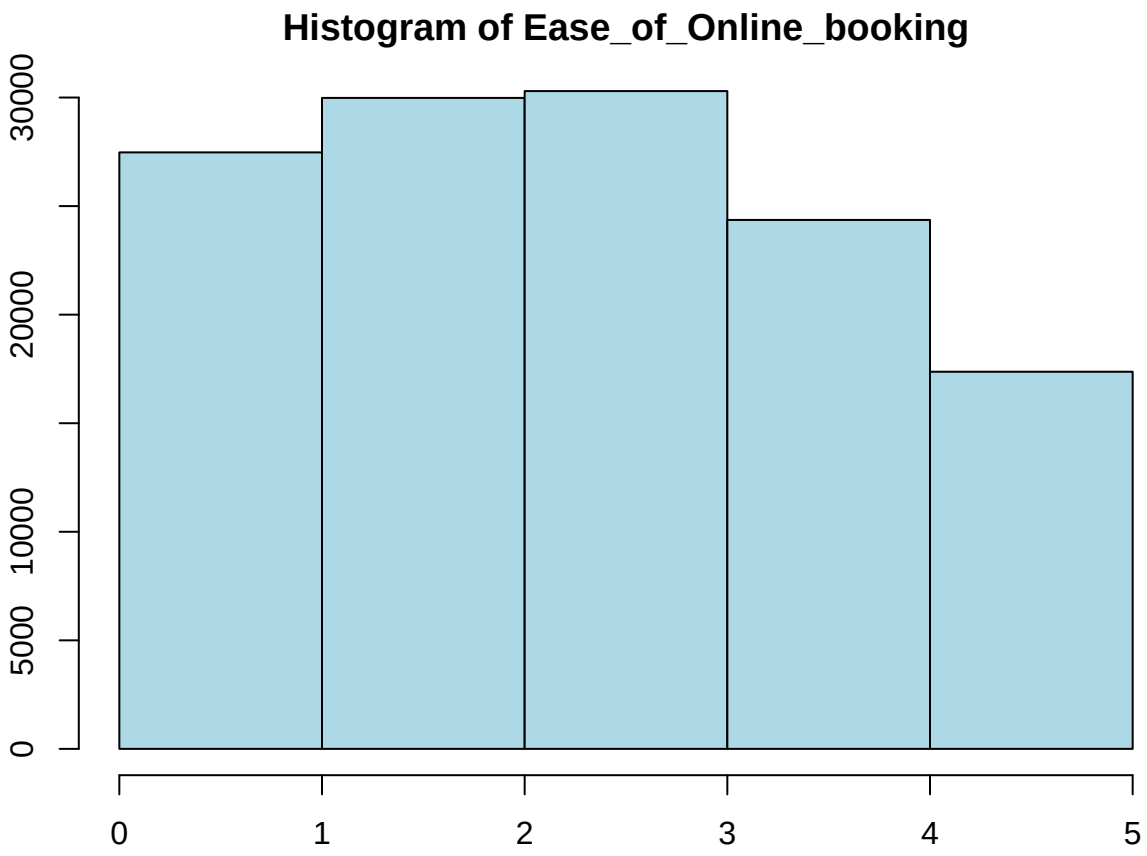

Histogram of Type_of_Travel
0
20000
40000
60000
80000
0.0
0.2
0.4
0.6
0.8
1.0

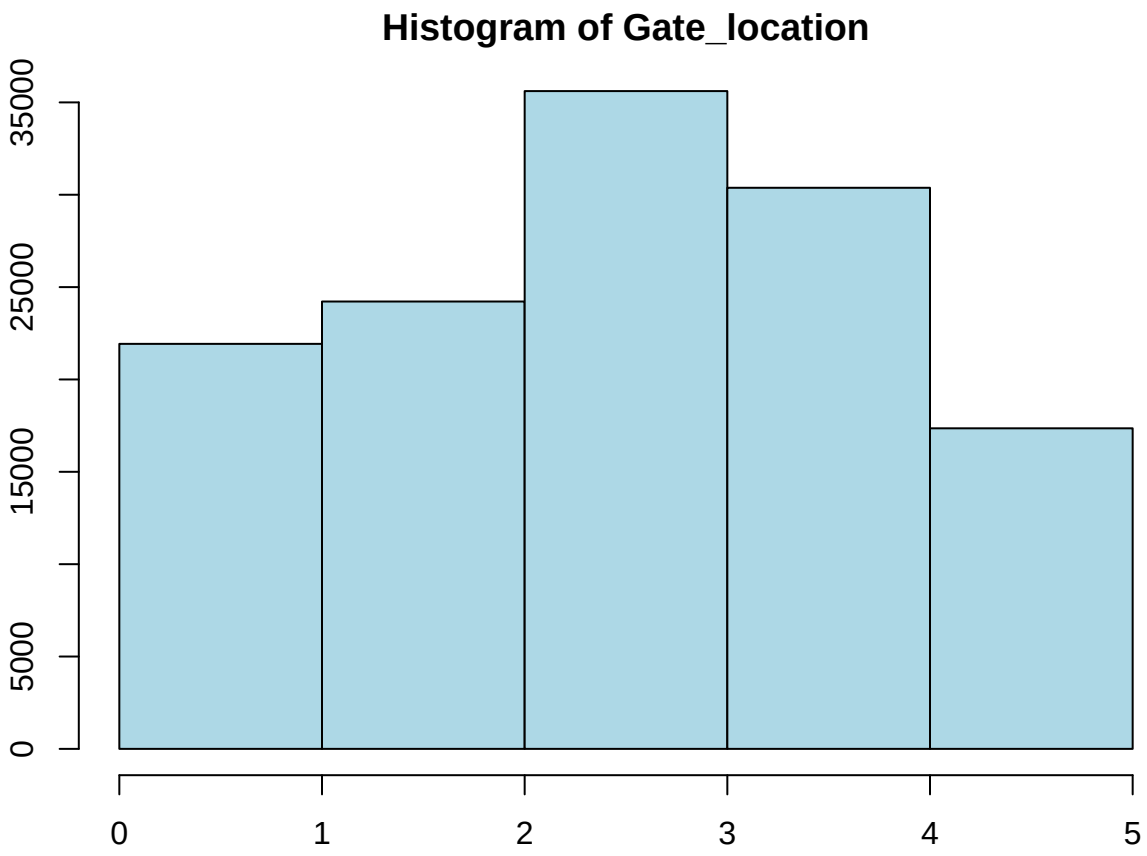

Histogram of Class
0
10000
30000
50000
0.0
0.5
1.0
1.5
2.0

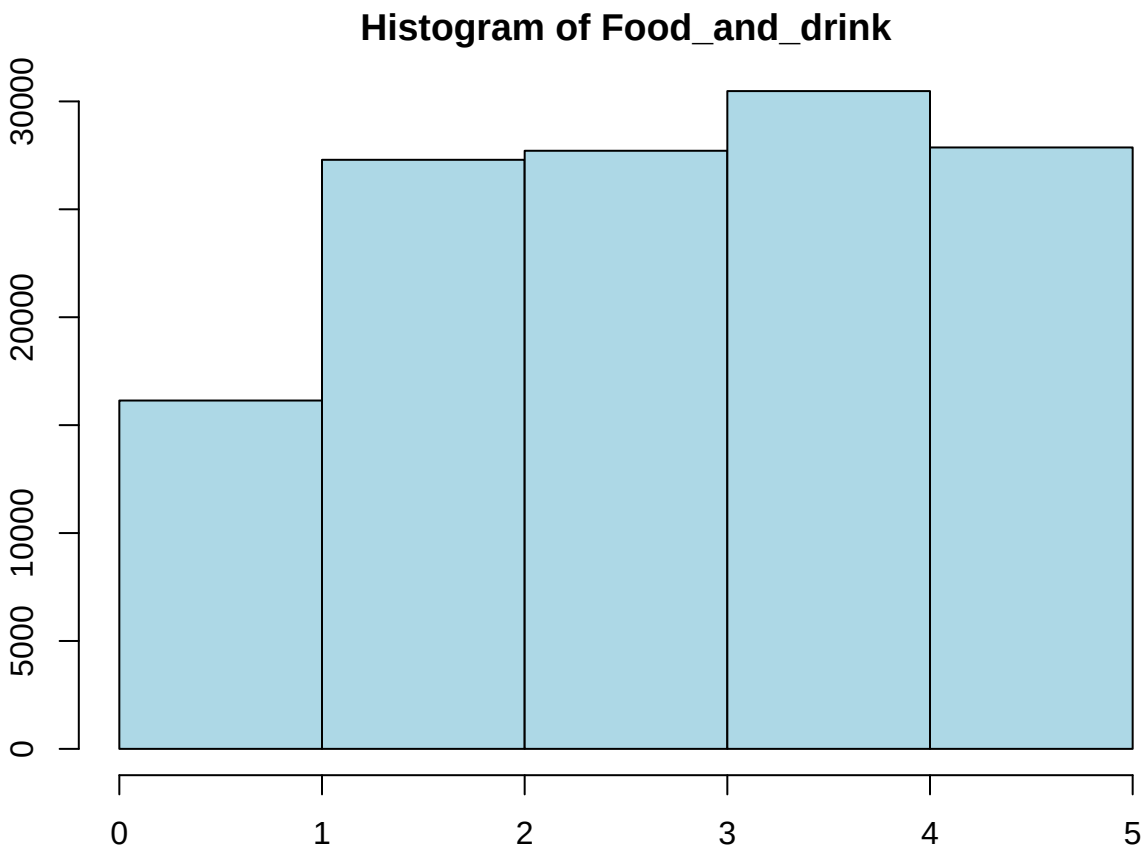

Histogram of Flight_Distance
0
5000
10000
15000
20000
0
1000
2000
3000
4000
5000

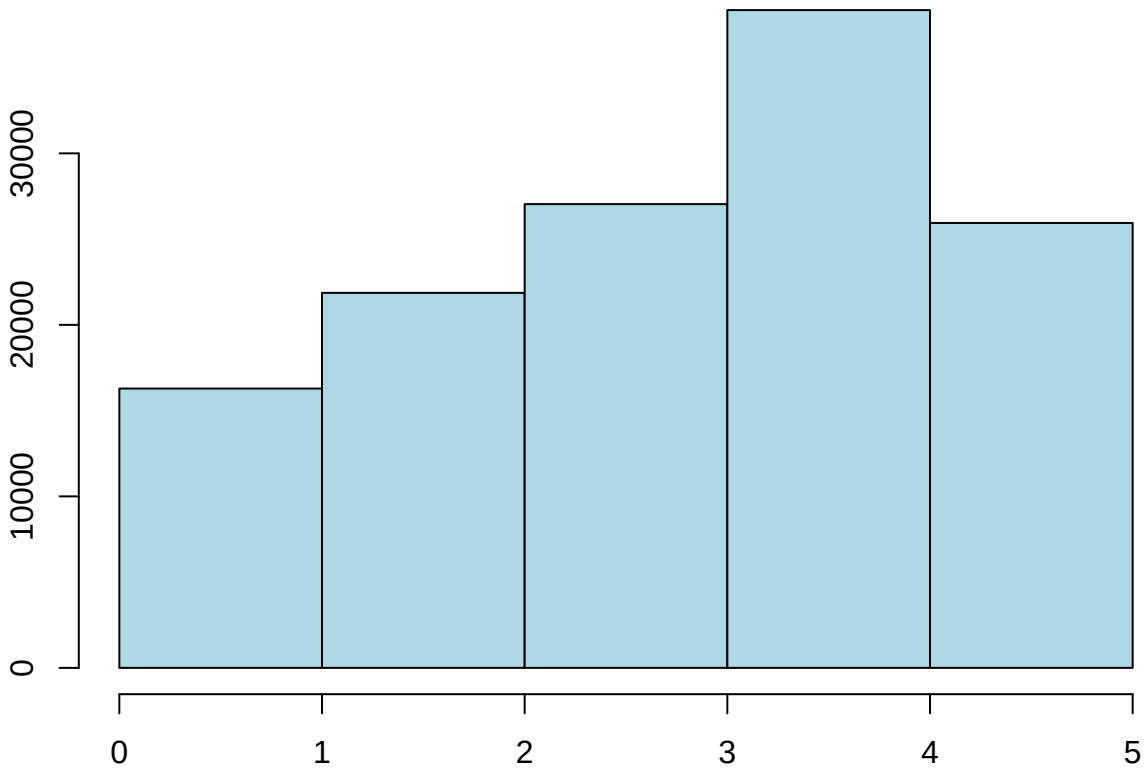

Histogram of Inflight_wifi_service
0
5000
15000
25000
0
1
2
3
4
5

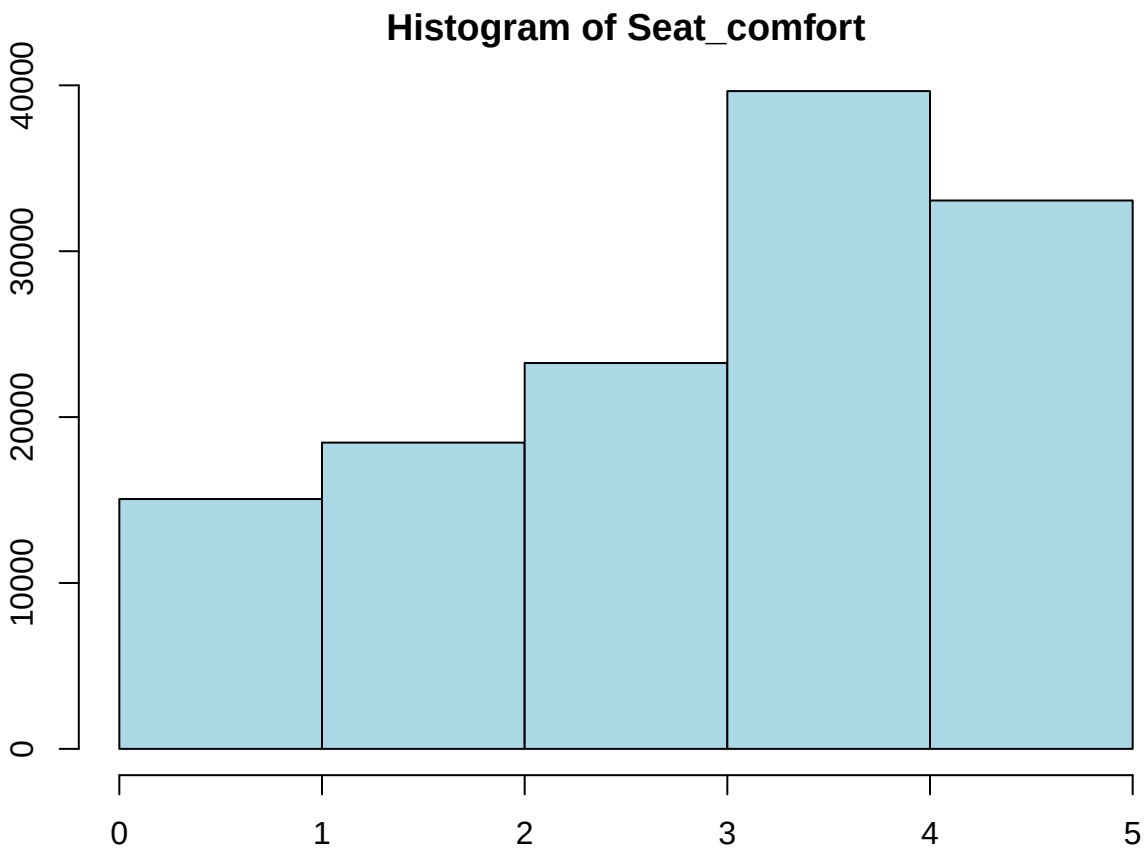

Histogram of Departure_Arrival_time_convenient
0
5000
15000
25000
0
1
2
3
4
5

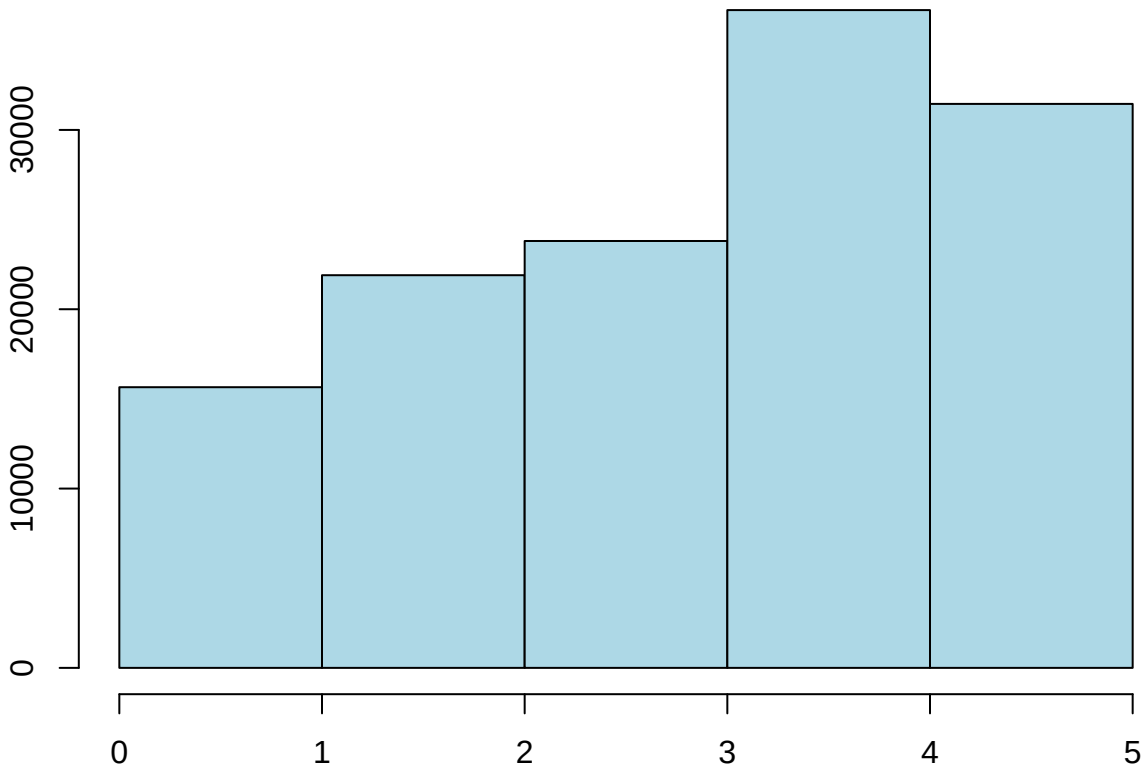

Histogram of Ease_of_Online_booking
0
5000
10000
20000
30000
0
1
2
3
4
5

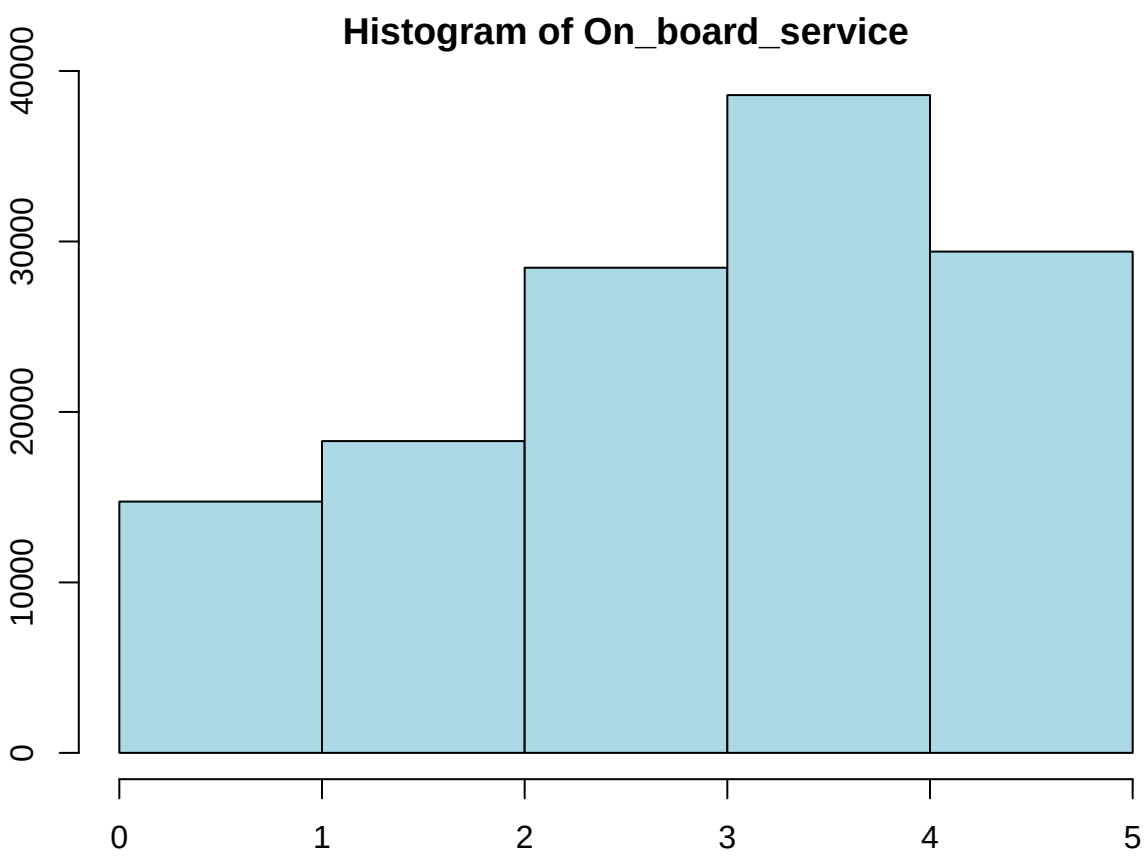

Histogram of Gate_location
35000
25000
15000
5000
0
0
1
2
3
4
5

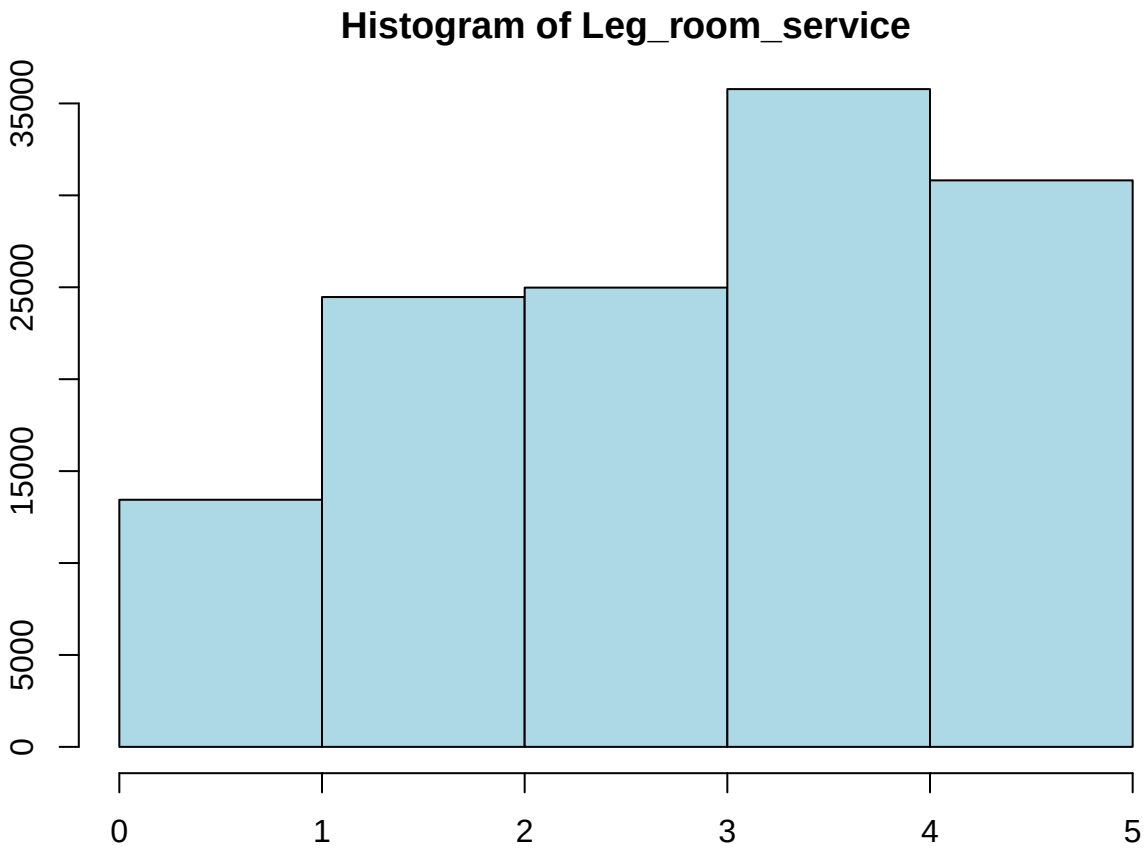

Histogram of Food_and_drink
0
5000
10000
20000
30000
0
1
2
3
4
5

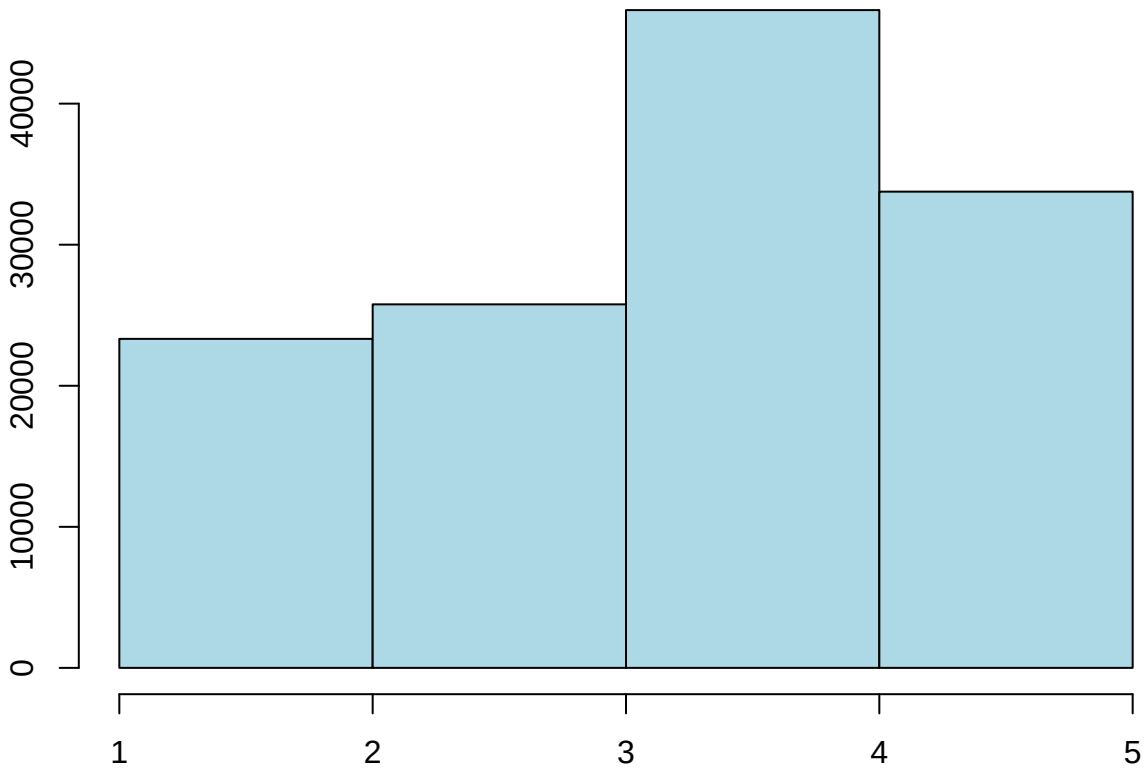

Histogram of Online_boarding
0
10000
20000
30000
0
1
2
3
4
5

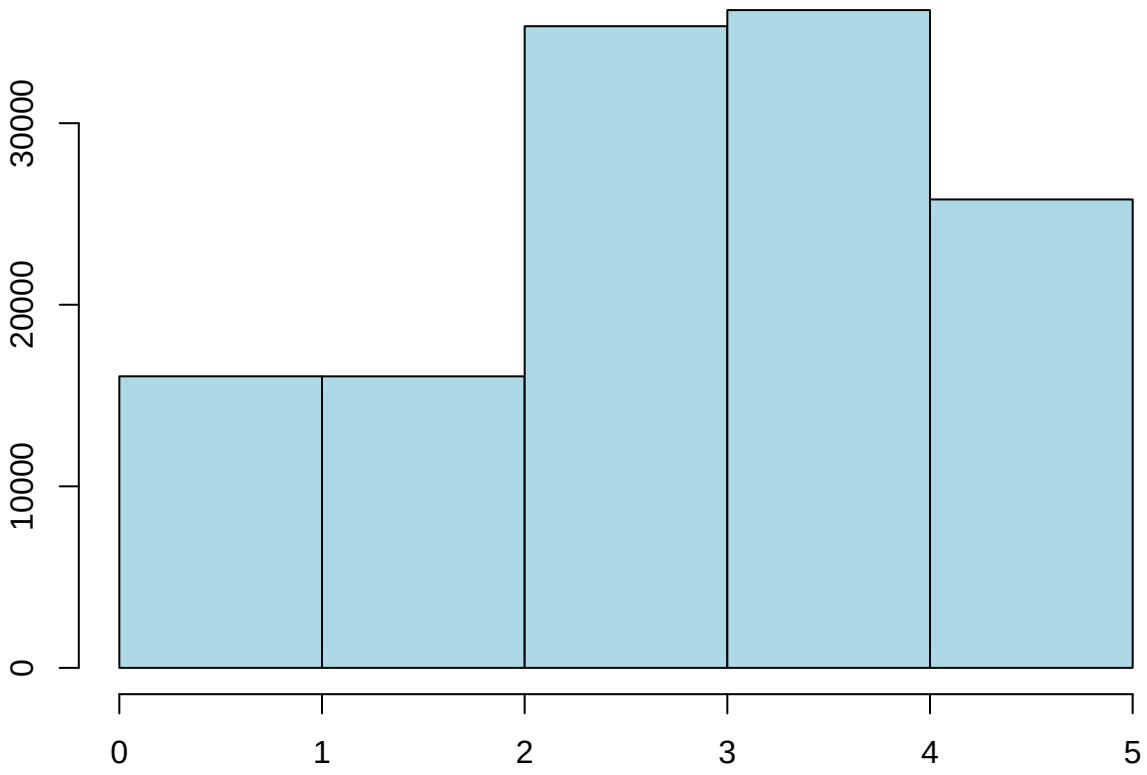

Histogram of Seat_comfort
0
10000
20000
30000
40000
0
1
2
3
4
5

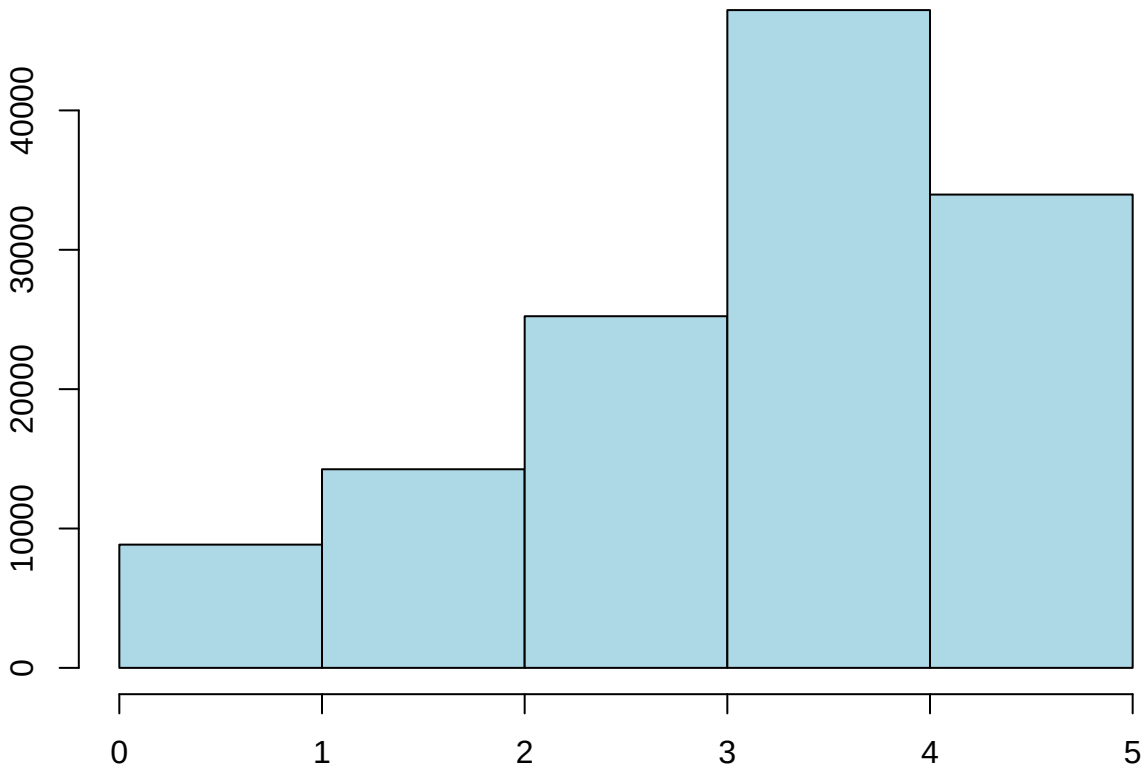

Histogram of Inflight_entertainment
0
10000
20000
30000
0
1
2
3
4
5

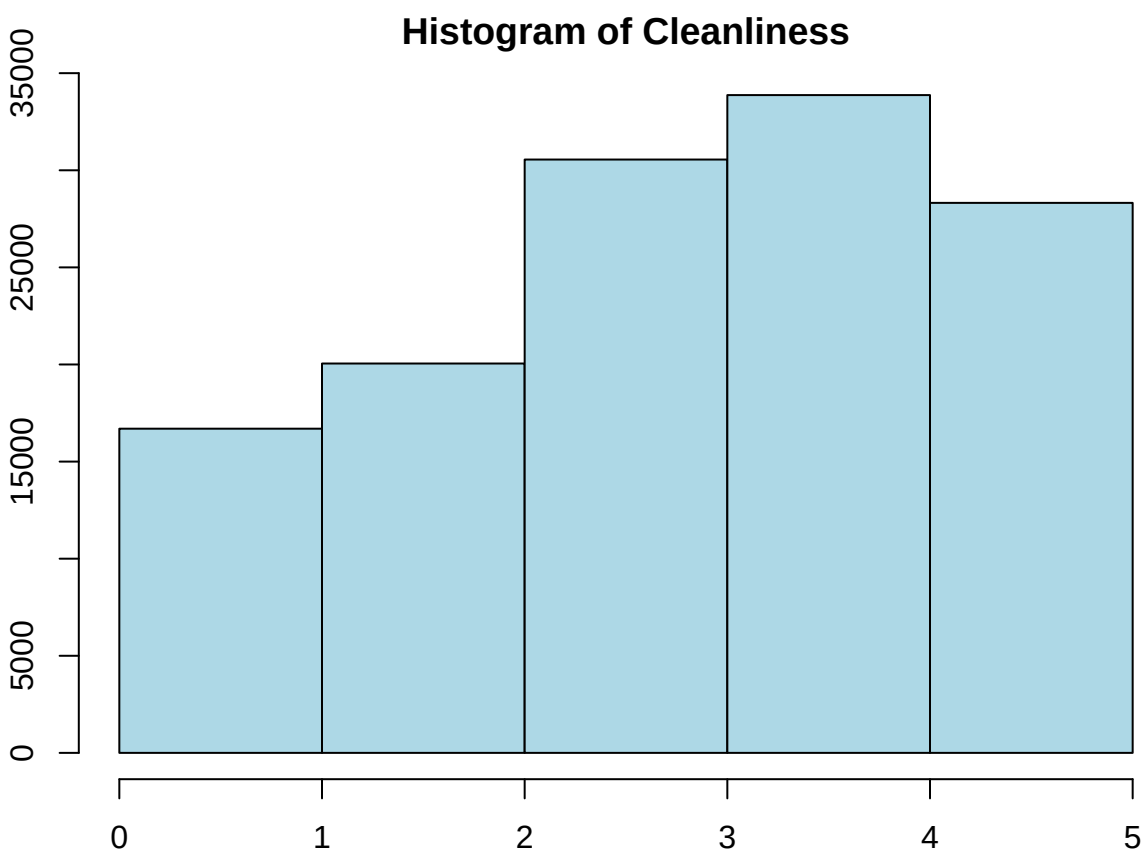

Histogram of On_board_service
0
10000
20000
30000
40000
0
1
2
3
4
5

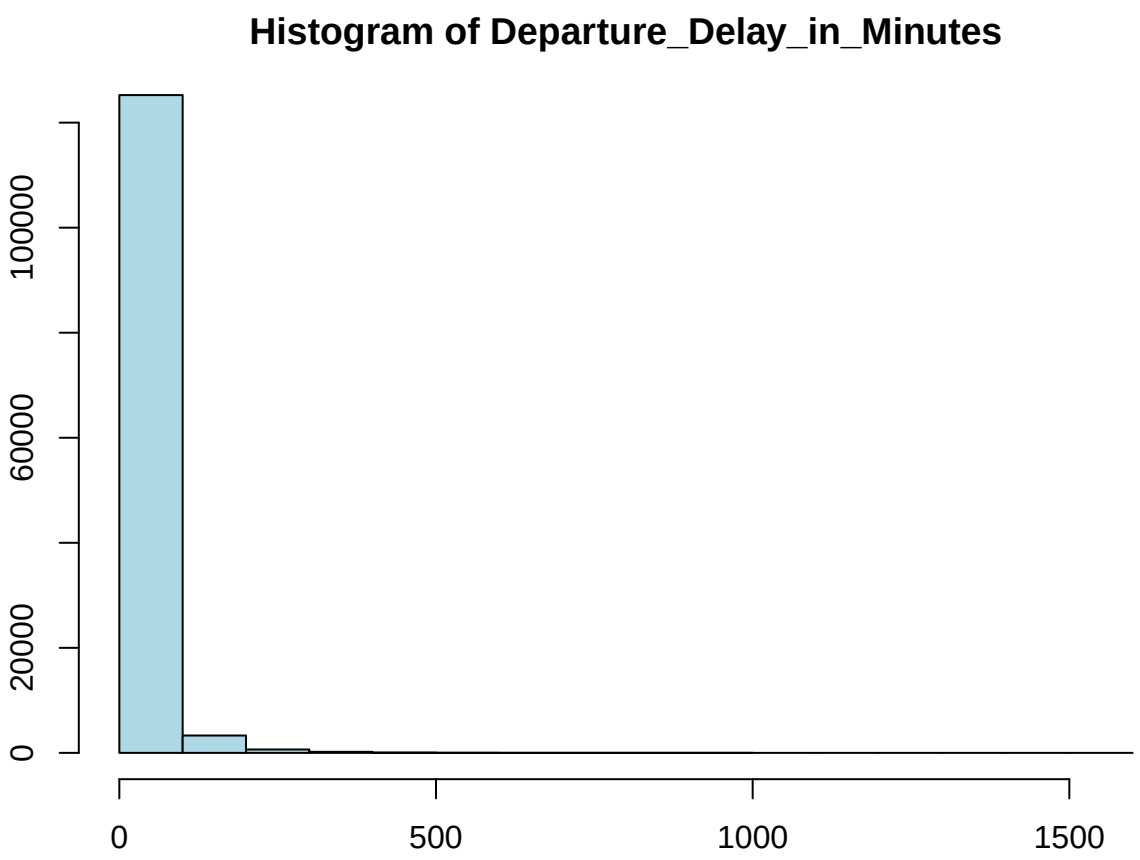

Histogram of Leg_room_service
0
5000
15000
25000
35000
0
1
2
3
4
5

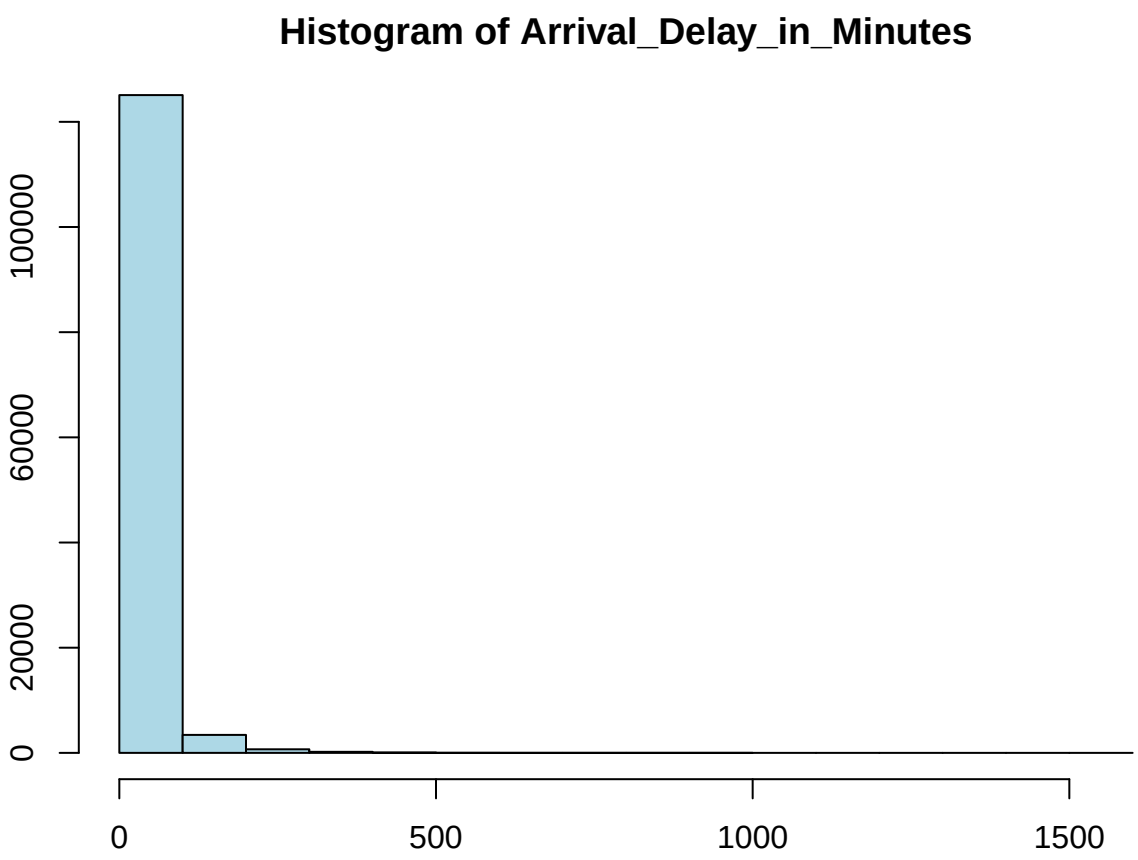

Histogram of Baggage_handling
0
10000
20000
30000
40000
1
2
3
4
5

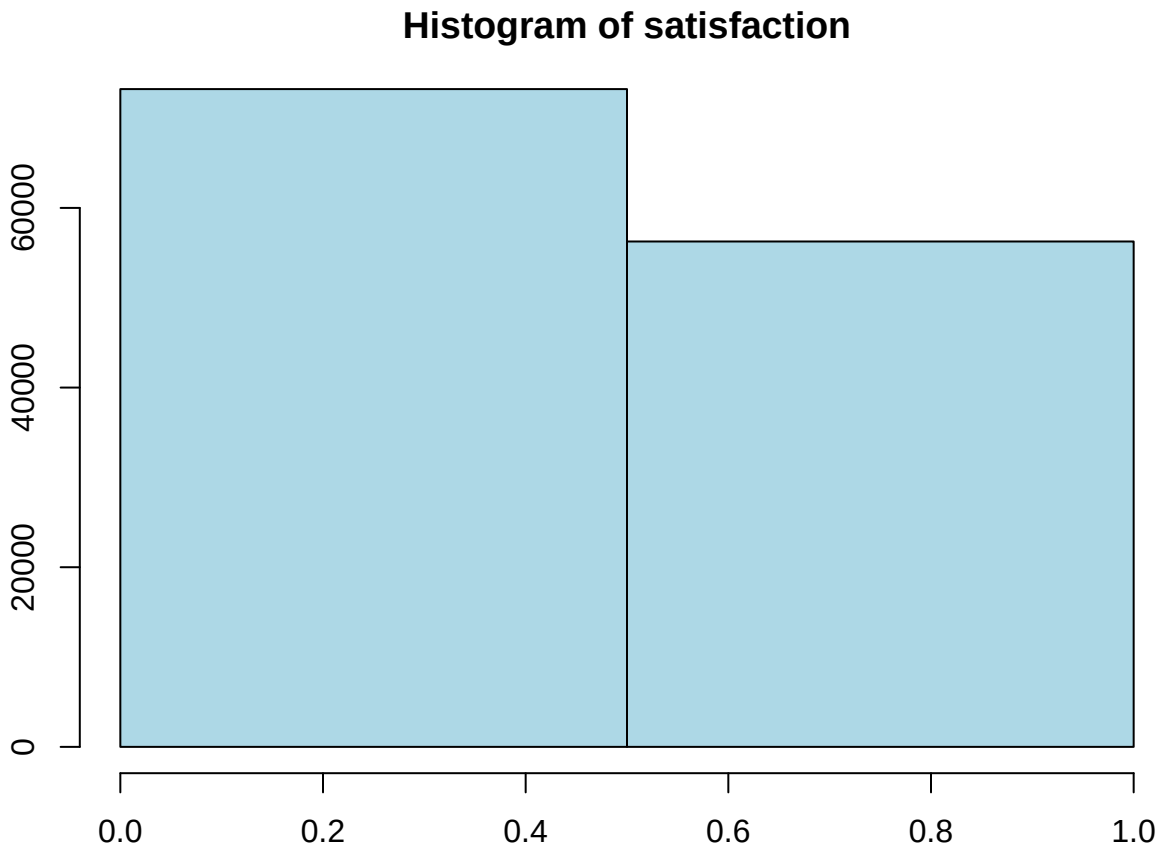

Histogram of Checkin_service
0
10000
20000
30000
0
1
2
3
4
5


Histogram of Inflight_service
0
10000
20000
30000
40000
0
1
2
3
4
5


Histogram of Cleanliness
0
5000
15000
25000
35000
0
1
2
3
4
5


Histogram of Departure_Delay_in_Minutes
0
20000
60000
100000
0
500
1000
1500

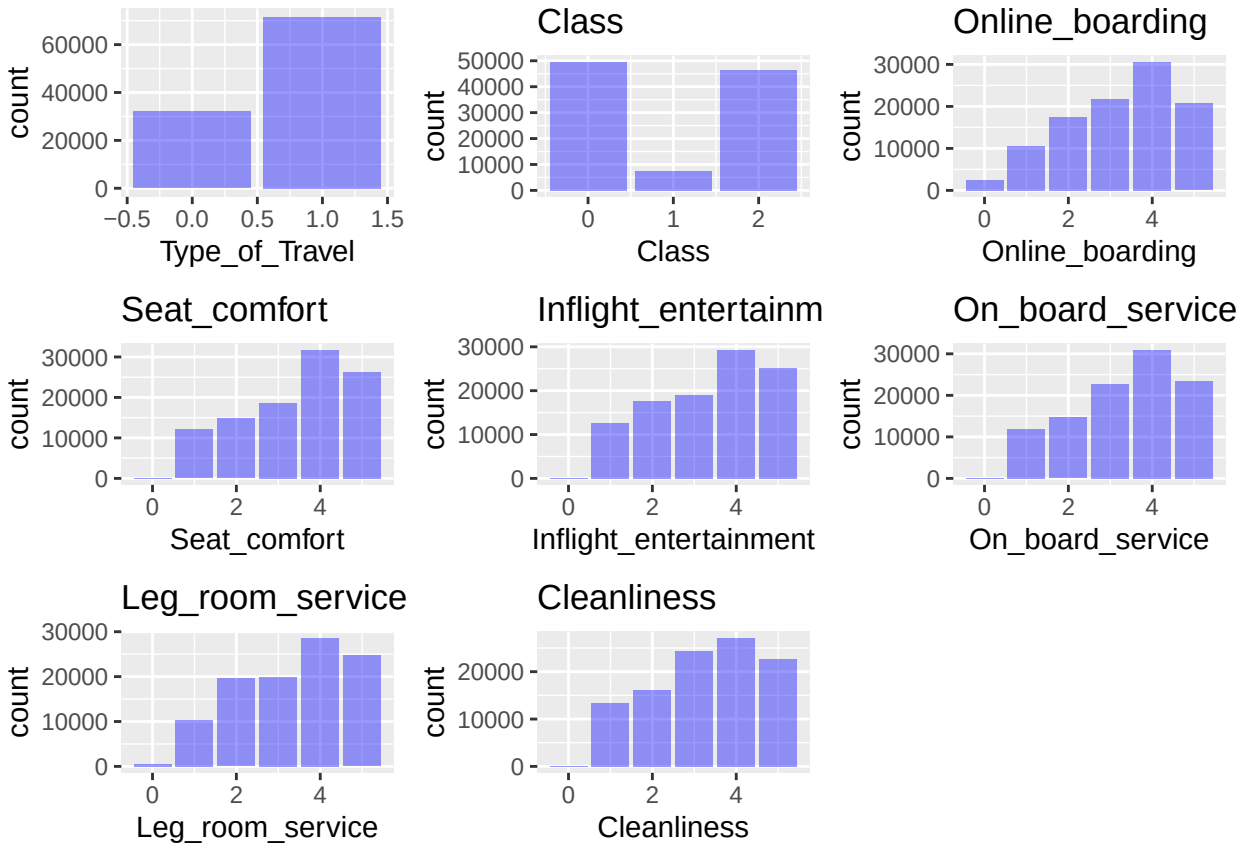

Histogram of Arrival_Delay_in_Minutes
0
20000
60000
100000
0
500
1000
1500




```r
# Reset the plotting settings to default
par(mfrow = c(1, 1))
```

```r
a = ggplot(train, aes(x = Type_of_Travel, fill = sat)) +
  geom_bar(fill = 'Blue', alpha = 0.4)


b = ggplot(train, aes(x = Class, fill = sat)) +
  geom_bar(fill = 'Blue', alpha = 0.4) +
  ggtitle("Class") +
  xlab('Class')

c = ggplot(train, aes(x = Online_boarding, fill = sat)) +
  geom_bar(fill = 'Blue', alpha = 0.4) +
  ggtitle("Online_boarding") +
  xlab('Online_boarding')

d = ggplot(train, aes(x = Seat_comfort, fill = sat)) +
  geom_bar(fill = 'Blue', alpha = 0.4) +
  ggtitle("Seat_comfort") +
  xlab('Seat_comfort')

e = ggplot(train, aes(x = Inflight_entertainment, fill = sat)) +
  geom_bar(fill = 'Blue', alpha = 0.4) +
  ggtitle("Inflight_entertainment") +
  xlab('Inflight_entertainment')
```

```
f = ggplot(train, aes(x = On_board_service, fill = sat)) +
  geom_bar(fill = 'Blue', alpha = 0.4) +
  ggtitle("On_board_service") +
  xlab('On_board_service')


g = ggplot(train, aes(x = Leg_room_service, fill = sat)) +
  geom_bar(fill = 'Blue', alpha = 0.4) +
  ggtitle("Leg_room_service") +
  xlab('Leg_room_service')

h = ggplot(train, aes(x = Cleanliness, fill = sat)) +
  geom_bar(fill = 'Blue', alpha = 0.4) +
  ggtitle("Cleanliness") +
  xlab('Cleanliness')


grid.arrange(a, b, c, d, e, f, g, h, ncol = 3)
```



Relation between Arrival_Delay_in_Minutes and Departure_Delay_in_Minutes (linear)

```
#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
```

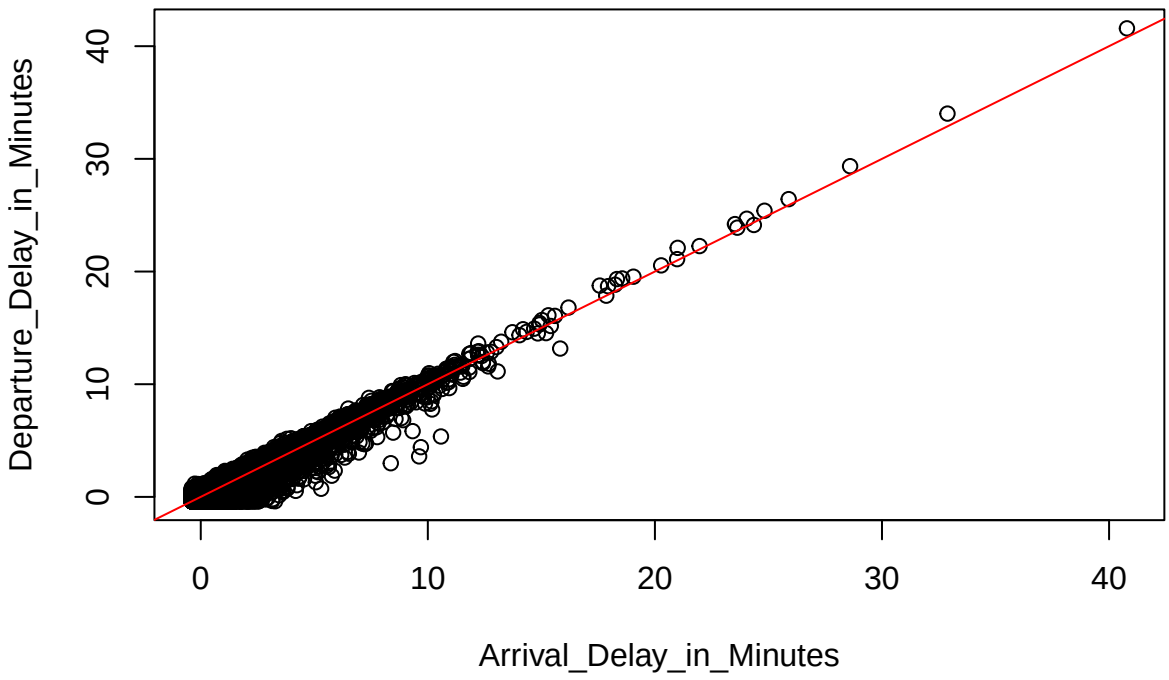
```r
#build a dataframe where for each variable we look the partial correlation with all the others
#we pick the highest and we save it in a dataframe
#we set a treshold of 0


#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
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),]
```

```r
#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")
```




```
# print table of type of travel by satisfacton
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
```

```r
sum(data$Arrival_Delay_in_Minutes <= 0)
```

```
## [1] 72753
```

```r
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
```