Analysis of Hotel Bookings Cancellation

Contents

Introduction

This document presents an analysis of a real-life hotel stay dataset using R Markdown to identify factors influencing booking cancellations. By exploring this dataset, we aim to uncover insights that can help inform strategies for better managing hotel bookings and enhancing guest satisfaction.

Step1: Installing and loading necessary packages

```
# install.packages("caret")
# install.packages("corrplot")

library(tidyverse)  # For data manipulation
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ---------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(ggplot2)    # For data visualization

library(corrplot)   # For visualizing correlation matrices
```

```
## corrplot 0.92 loaded
```

```r
# library(RColorBrewer) # For defining a custom color palette

# library(caret)      # For Machine Learning

library(rpart)  # For Machine Learning
```

Step2: Loading the dataset

```r
# Load the hotel bookings data
hotel_data <- read.csv("hotel_bookings.csv")

# path to the dataset is: https://intro-datascience.s3.us-east-2.amazonaws.com/Resort01.csv
```

Step3: Exploring and manipulating the dataset

```r
# Show the structure of the dataset
str(hotel_data)
```

```
## 'data.frame':    40060 obs. of  20 variables:
##  $ IsCanceled                : int  0 0 0 0 0 0 0 0 1 1 ...
##  $ LeadTime                  : int  342 737 7 13 14 14 0 9 85 75 ...
##  $ StaysInWeekendNights      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ StaysInWeekNights         : int  0 0 1 1 2 2 2 2 3 3 ...
##  $ Adults                    : int  2 2 1 1 2 2 2 2 2 2 ...
##  $ Children                  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Babies                    : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Meal                      : chr  "BB       " "BB       " "BB       " "BB       " ...
##  $ Country                   : chr  "PRT" "PRT" "GBR" "GBR" ...
##  $ MarketSegment             : chr  "Direct" "Direct" "Direct" "Corporate" ...
##  $ IsRepeatedGuest           : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ PreviousCancellations     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ PreviousBookingsNotCanceled: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ ReservedRoomType          : chr  "C               " "C               " "A               " "A
##  $ AssignedRoomType          : chr  "C               " "C               " "C               " "A
##  $ BookingChanges            : int  3 4 0 0 0 0 0 0 0 0 ...
##  $ DepositType               : chr  "No Deposit     " "No Deposit     " "No Deposit     " "No Depos
##  $ CustomerType              : chr  "Transient" "Transient" "Transient" "Transient" ...
##  $ RequiredCarParkingSpaces  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ TotalOfSpecialRequests    : int  0 0 0 0 1 1 0 1 1 0 ...
```

```r
# Show summary statistics for numerical variables
summary(hotel_data)
```

```
##    IsCanceled        LeadTime      StaysInWeekendNights StaysInWeekNights
##  Min.   :0.0000   Min.   :  0.00   Min.   : 0.00        Min.   : 0.000
##  1st Qu.:0.0000   1st Qu.: 10.00   1st Qu.: 0.00        1st Qu.: 1.000
##  Median :0.0000   Median : 57.00   Median : 1.00        Median : 3.000
##  Mean   :0.2776   Mean   : 92.68   Mean   : 1.19        Mean   : 3.129
```

```
##  3rd Qu.:1.0000   3rd Qu.:155.00   3rd Qu.: 2.00        3rd Qu.: 5.000
##  Max.   :1.0000   Max.   :737.00   Max.   :19.00        Max.   :50.000
##      Adults          Children          Babies           Meal
##  Min.   : 0.000   Min.   : 0.0000   Min.   :0.0000   Length:40060
##  1st Qu.: 2.000   1st Qu.: 0.0000   1st Qu.:0.0000   Class :character
##  Median : 2.000   Median : 0.0000   Median :0.0000   Mode  :character
##  Mean   : 1.867   Mean   : 0.1287   Mean   :0.0139
##  3rd Qu.: 2.000   3rd Qu.: 0.0000   3rd Qu.:0.0000
##  Max.   :55.000   Max.   :10.0000   Max.   :2.0000
##    Country          MarketSegment      IsRepeatedGuest   PreviousCancellations
##  Length:40060       Length:40060       Min.   :0.00000   Min.   : 0.0000
##  Class :character   Class :character   1st Qu.:0.00000   1st Qu.: 0.0000
##  Mode  :character   Mode  :character   Median :0.00000   Median : 0.0000
##                                        Mean   :0.04438   Mean   : 0.1017
##                                        3rd Qu.:0.00000   3rd Qu.: 0.0000
##                                        Max.   :1.00000   Max.   :26.0000
##  PreviousBookingsNotCanceled ReservedRoomType   AssignedRoomType
##  Min.   : 0.0000             Length:40060       Length:40060
##  1st Qu.: 0.0000             Class :character   Class :character
##  Median : 0.0000             Mode  :character   Mode  :character
##  Mean   : 0.1465
##  3rd Qu.: 0.0000
##  Max.   :30.0000
##  BookingChanges   DepositType        CustomerType
##  Min.   : 0.000   Length:40060       Length:40060
##  1st Qu.: 0.000   Class :character   Class :character
##  Median : 0.000   Mode  :character   Mode  :character
##  Mean   : 0.288
##  3rd Qu.: 0.000
##  Max.   :17.000
##  RequiredCarParkingSpaces TotalOfSpecialRequests
##  Min.   :0.0000           Min.   :0.0000
##  1st Qu.:0.0000           1st Qu.:0.0000
##  Median :0.0000           Median :0.0000
##  Mean   :0.1381           Mean   :0.6198
##  3rd Qu.:0.0000           3rd Qu.:1.0000
##  Max.   :8.0000           Max.   :5.0000
```

```
# View the first few rows of the data
head(hotel_data)
```

```
##   IsCanceled LeadTime StaysInWeekendNights StaysInWeekNights Adults Children
## 1          0      342                    0                 0      2        0
## 2          0      737                    0                 0      2        0
## 3          0        7                    0                 1      1        0
## 4          0       13                    0                 1      1        0
## 5          0       14                    0                 2      2        0
## 6          0       14                    0                 2      2        0
##   Babies      Meal Country MarketSegment IsRepeatedGuest PreviousCancellations
## 1      0 BB            PRT        Direct               0                     0
## 2      0 BB            PRT        Direct               0                     0
## 3      0 BB            GBR        Direct               0                     0
## 4      0 BB            GBR     Corporate               0                     0
## 5      0 BB            GBR     Online TA               0                     0
```

```
## 6      0 BB            GBR     Online TA               0                       0
##   PreviousBookingsNotCanceled ReservedRoomType AssignedRoomType BookingChanges
## 1                           0 C                C                             3
## 2                           0 C                C                             4
## 3                           0 A                C                             0
## 4                           0 A                A                             0
## 5                           0 A                A                             0
## 6                           0 A                A                             0
##       DepositType CustomerType RequiredCarParkingSpaces TotalOfSpecialRequests
## 1 No Deposit         Transient                        0                       0
## 2 No Deposit         Transient                        0                       0
## 3 No Deposit         Transient                        0                       0
## 4 No Deposit         Transient                        0                       0
## 5 No Deposit         Transient                        0                       1
## 6 No Deposit         Transient                        0                       1
```

```
# View the last few rows of the data
tail(hotel_data)
```

```
##       IsCanceled LeadTime StaysInWeekendNights StaysInWeekNights Adults
## 40055          0      169                    2                 9      2
## 40056          0      212                    2                 8      2
## 40057          0      169                    2                 9      2
## 40058          0      204                    4                10      2
## 40059          0      211                    4                10      2
## 40060          0      161                    4                10      2
##       Children Babies      Meal Country MarketSegment IsRepeatedGuest
## 40055        0      0 BB            IRL        Direct               0
## 40056        1      0 BB            GBR Offline TA/TO               0
## 40057        0      0 BB            IRL        Direct               0
## 40058        0      0 BB            IRL        Direct               0
## 40059        0      0 HB            GBR Offline TA/TO               0
## 40060        0      0 HB            DEU Offline TA/TO               0
##       PreviousCancellations PreviousBookingsNotCanceled ReservedRoomType
## 40055                     0                           0 E
## 40056                     0                           0 A
## 40057                     0                           0 E
## 40058                     0                           0 E
## 40059                     0                           0 D
## 40060                     0                           0 A
##       AssignedRoomType BookingChanges     DepositType    CustomerType
## 40055 E                             0 No Deposit      Transient-Party
## 40056 A                             1 No Deposit            Transient
## 40057 E                             0 No Deposit      Transient-Party
## 40058 E                             0 No Deposit            Transient
## 40059 D                             0 No Deposit             Contract
## 40060 A                             0 No Deposit            Transient
##       RequiredCarParkingSpaces TotalOfSpecialRequests
## 40055                        0                      1
## 40056                        0                      0
## 40057                        0                      1
## 40058                        0                      3
## 40059                        0                      1
## 40060                        0                      0
```

```r
# Check for number of unique values in each column
sapply(hotel_data, function(x) length(unique(x)))
```

```
##                  IsCanceled                    LeadTime
##                           2                         412
##        StaysInWeekendNights           StaysInWeekNights
##                          16                          31
##                      Adults                    Children
##                          14                           5
##                      Babies                        Meal
##                           3                           5
##                     Country              MarketSegment
##                         126                           6
##             IsRepeatedGuest       PreviousCancellations
##                           2                          11
## PreviousBookingsNotCanceled            ReservedRoomType
##                          31                          10
##            AssignedRoomType              BookingChanges
##                          11                          15
##                 DepositType                CustomerType
##                           3                           4
##    RequiredCarParkingSpaces      TotalOfSpecialRequests
##                           5                           6
```

Note: Checking for unique values is particularly useful for understanding the diversity and distribution of categorical variables. As we can see from the result, the 'Meal' variable has 5 distinct values, 'MarketSegment' has 6 distinct values, and 'DepositType' has 3 distinct values.

```r
# Check for missing values in each column (variable) of the dataset.
colSums(is.na(hotel_data))
```

```
##                  IsCanceled                    LeadTime
##                           0                           0
##        StaysInWeekendNights           StaysInWeekNights
##                           0                           0
##                      Adults                    Children
##                           0                           0
##                      Babies                        Meal
##                           0                           0
##                     Country              MarketSegment
##                           0                           0
##             IsRepeatedGuest       PreviousCancellations
##                           0                           0
## PreviousBookingsNotCanceled            ReservedRoomType
##                           0                           0
##            AssignedRoomType              BookingChanges
##                           0                           0
##                 DepositType                CustomerType
##                           0                           0
##    RequiredCarParkingSpaces      TotalOfSpecialRequests
##                           0                           0
```

From the output, we can see there is no missing value in the entire dataset.

Display frequencies of selected categorical variables in the upcoming code batch.

```
# Calculate frequencies of Country categories
country_freq <- table(hotel_data$Country)

# Sort the frequencies in descending order
sorted_country_freq <- sort(country_freq, decreasing = TRUE)

# Display the sorted frequencies and corresponding country names
print(sorted_country_freq)
```

```
##
##   PRT   GBR   ESP   IRL   FRA   DEU    CN   NLD   USA  NULL   ITA   BEL   CHE
## 17630  6814  3957  2166  1611  1203   710   514   479   464   459   448   435
##   BRA   POL   SWE   AUT   RUS   ROU   FIN   CHN   NOR   AUS   LUX   MAR   DNK
##   430   333   304   210   189   177   151   134   123    87    80    75    65
##   ARG   HUN   LTU   IND   EST   LVA   ISR   CZE   AGO   TUR   UKR   ZAF   CHL
##    57    47    46    37    33    33    28    27    24    23    23    18    17
##   COL   PHL   NZL   GIB   DZA   SVK   TWN   ARE   GEO   HRV   OMN   SVN   GRC
##    16    16    14    13    12    12    12    11    11    11    11    11    10
##   MYS   NGA   JPN   KOR   PRI   CYP   URY   BLR   SRB   ISL   LBN   MDV   MEX
##    10    10     9     9     9     8     8     7     7     6     6     6     6
##   MOZ   THA   AND   BGR   CPV   IDN   IRN   JAM   KAZ   CUB   HKG   PAK   SGP
##     6     6     5     5     5     5     5     5     5     4     4     4     4
##   SUR   ALB   AZE   CAF   DOM   JEY   KWT   VEN   ARM   CIV   CMR   CRI   ECU
##     4     3     3     3     3     3     3     3     2     2     2     2     2
##   JOR   MLT   MWI   VNM   ZWE   BDI   BHR   BHS   BIH   BWA   COM   CYM   DJI
##     2     2     2     2     2     1     1     1     1     1     1     1     1
##   EGY   FJI   GGY   LKA   MAC   MDG   MKD   MUS   NPL   PER   PLW   QAT   SAU
##     1     1     1     1     1     1     1     1     1     1     1     1     1
##   SEN   SMR   SYC   SYR   TGO   TUN   UGA   UZB   ZMB
##     1     1     1     1     1     1     1     1     1
```

From the result of frequencies of Country, we observe: - The top 6 countries from which most guests come are Portugal (PRT), United Kingdom (GBR), Spain (ESP), Ireland (IRL), France (FRA), and Germany (DEU). - There are 464 entries labeled as NULL. To enhance clarity, in the next code block, I will replace NULL entries with 'Unknown'.

```
# Replace NULL values with "Unknown" in the Country column
hotel_data$Country <- gsub("NULL", "Unknown", hotel_data$Country)

# Check the number of entries with "Unknown" in the Country column
print(sum(hotel_data$Country == "Unknown"))
```

```
## [1] 464
```

```
# Calculate frequencies of MarketSegment categories
market_segment_freq <- table(hotel_data$MarketSegment)

print(market_segment_freq)
```

```
##
```

```
## Complementary     Corporate        Direct        Groups Offline TA/TO
##           201          2309          6513          5836          7472
##     Online TA
##         17729
```

The analysis indicates that the most prevalent market segment is Online Travel Agents (Online TA), followed by Offline TA and Direct.

Histogram of Canceled Bookings by Market Segments

```r
# Histogram of Canceled booking

# Calculate MarketSegment frequencies for canceled bookings
canceled_market_segment_freq <- table(hotel_data$MarketSegment[hotel_data$IsCanceled == 1])

# Sort MarketSegment frequencies in descending order
sorted_canceled_market_segment_freq <- sort(canceled_market_segment_freq, decreasing = TRUE)

# Generate a smooth color palette
color_palette_canceled <- colorRampPalette(c("Skyblue", "Purple"))(length(sorted_canceled_market_segmen

# Set up the plotting environment to display multiple plots in one device
par(mfrow = c(1, 2))  # 1 row and 2 columns for side-by-side plots

# Plotting a bar chart of MarketSegment frequencies (sorted)
barplot(
  sorted_canceled_market_segment_freq,
  main = "Canceled Bookings \n by Market Segment",
  xlab = "",
  ylab = "Frequency",
  ylim = c(0, max(sorted_canceled_market_segment_freq) * 1.2),
  col = color_palette_canceled,
  names.arg = names(sorted_canceled_market_segment_freq),
  cex.names = 0.8,
  las = 2,  # Rotate x-axis labels vertically
  cex.main = 1,  # Adjust font size of main title
  cex.lab = 0.8  # Adjust font size of y-axis label
)


# Histogram of Not Canceled data

# Calculate MarketSegment frequencies for non-canceled bookings
not_canceled_market_segment_freq <- table(hotel_data$MarketSegment[hotel_data$IsCanceled == 0])

# Sort MarketSegment frequencies in descending order
sorted_not_canceled_market_segment_freq <- sort(not_canceled_market_segment_freq, decreasing = TRUE)

# Generate a smooth color palette
color_palette_not_canceled <- colorRampPalette(c("Skyblue", "Purple"))(length(sorted_not_canceled_marke

# Plotting a bar chart of MarketSegment frequencies (sorted)
barplot(
  sorted_not_canceled_market_segment_freq,
```

```
  main = "Not Canceled Bookings\n by Market Segment",
  xlab = "",
  ylab = "Frequency",
  ylim = c(0, max(sorted_not_canceled_market_segment_freq) * 1.2),
  col = color_palette_not_canceled,
  names.arg = names(sorted_not_canceled_market_segment_freq),
  cex.names = 0.8,
  las = 2,  # Rotate x-axis labels vertically
  cex.main = 1,  # Adjust font size of main title
  cex.lab = 0.8  # Adjust font size of y-axis label
)
```

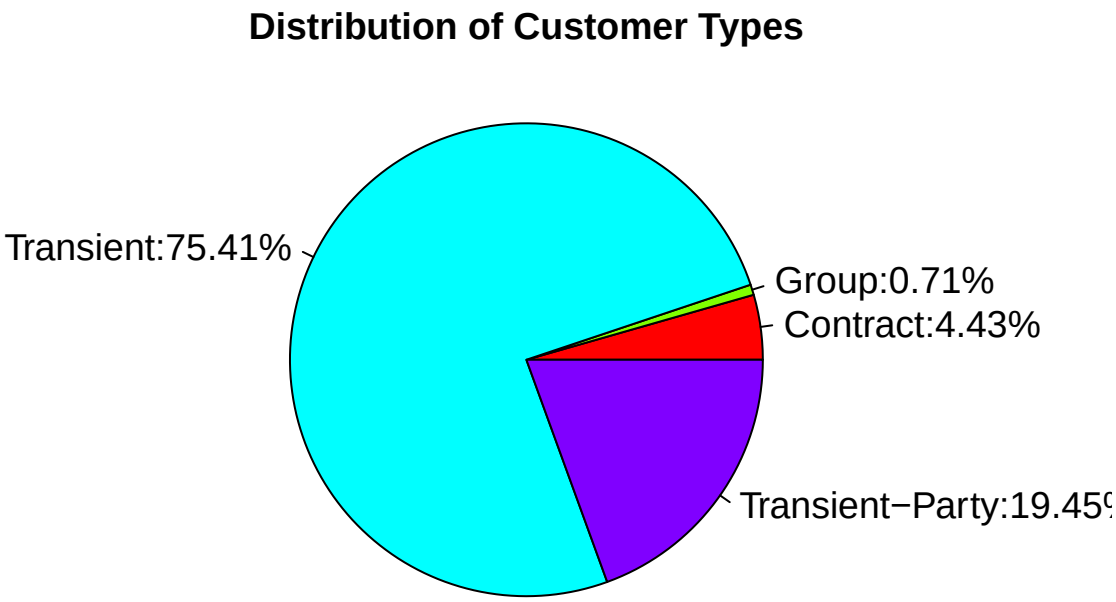

```
# Reset plot parameters to default after plotting
par(mfrow = c(1, 1))
```

The graph illustrates that the Online TA segment exhibits the highest cancellation rate, followed by the Group segment.

Distribution of Customer Types

```
# Calculate frequencies of CustomerType categories
customer_type_freq <- table(hotel_data$CustomerType)

# Calculate percentages for each category
customer_type_percent <- prop.table(customer_type_freq) * 100
```

```r
# Round percentages to two decimal places
customer_type_percent <- round(customer_type_percent, 2)

# Determine the number of categories
num_categories <- length(customer_type_freq)

# Plotting a pie chart of DepositType percentages with a larger size
pie(customer_type_freq,
    main = "Distribution of Customer Types",
    labels = paste(names(customer_type_freq), ":", customer_type_percent, "%", sep = ""),
    col = rainbow(num_categories),  # Using a rainbow color palette
    cex = 1.2,  # Adjust label size
    radius = 1  # Increase the size of the pie chart
   )
```




```r
# Calculate frequencies of DepositType categories
deposit_type_freq <- table(hotel_data$DepositType)

print(deposit_type_freq)
```

```
##
## No Deposit      Non Refund      Refundable
##          38199            1719             142
```

```r
# Calculate DepositType frequencies for canceled bookings
canceled_deposit_type_freq <- table(hotel_data$DepositType[hotel_data$IsCanceled == 1])

# Calculate DepositType frequencies for non-canceled bookings
not_canceled_deposit_type_freq <- table(hotel_data$DepositType[hotel_data$IsCanceled == 0])

# Combine deposit type frequencies with labels
combined_data <- data.frame(DepositType = names(canceled_deposit_type_freq),
                            Canceled = as.numeric(canceled_deposit_type_freq),
                            NotCanceled = as.numeric(not_canceled_deposit_type_freq))

# Reshape data for plotting
combined_data_long <- tidyr::pivot_longer(combined_data, cols = c("Canceled", "NotCanceled"),
                                          names_to = "BookingStatus", values_to = "Frequency")

# Plotting using ggplot2
ggplot(combined_data_long, aes(x = DepositType, y = Frequency, fill = BookingStatus)) +
  geom_bar(stat = "identity", position = "dodge", width = 0.7) +
  labs(title = "Deposit Types: Canceled vs. Not Canceled Bookings",
       x = "Deposit Type", y = "Frequency") +
  theme_minimal()
```


```r
# Calculate frequencies of Meal
meal_type_freq <- table(hotel_data$Meal)
```

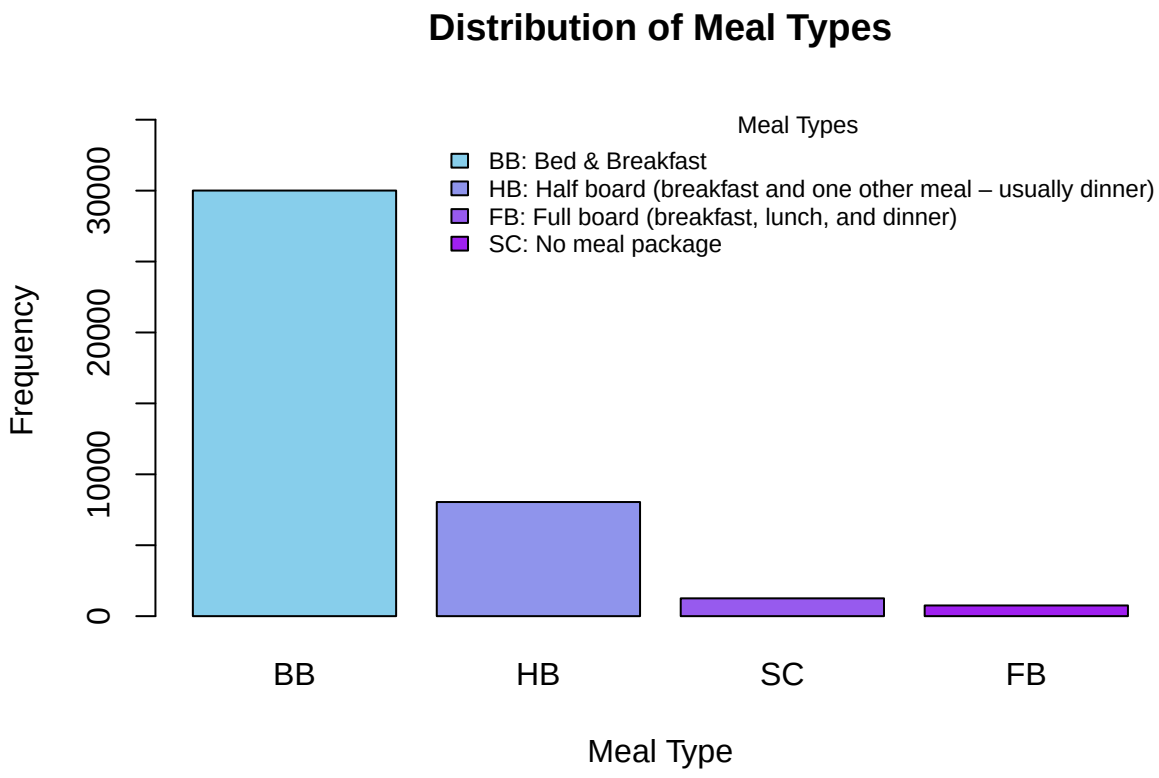
```r
# Display Meal types with their frequencies
print(meal_type_freq)
```

```
##
## BB        FB        HB        SC        Undefined
##     30005       754      8046        86      1169
```

Note: - According to the data dictionary, 'SC' and 'Undefined' both translate to "no meal package". - To standardize meal types, in the next code block, I will change 'Undefined' to 'SC'.

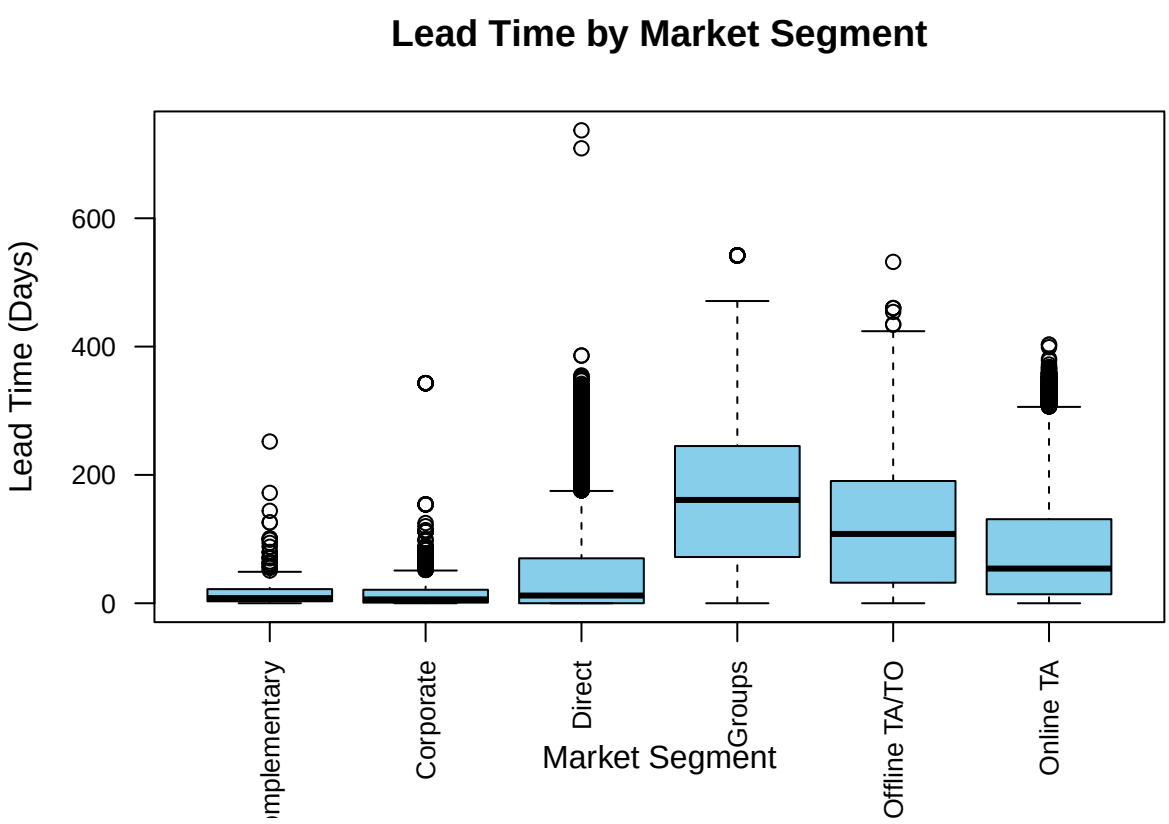
```r
# Clean up leading and trailing spaces in the Meal column
hotel_data$Meal <- trimws(hotel_data$Meal)

# Replace "Undefined" with "SC" in the Meal column
hotel_data$Meal[hotel_data$Meal == "Undefined"] <- "SC"
```

```r
# Calculate frequencies of Meal
meal_type_freq <- table(hotel_data$Meal)

# Sort MealType frequencies in descending order
sorted_meal_type_freq <- sort(meal_type_freq, decreasing = TRUE)

# Generate a smooth color palette by defining the start_color to end_color and the number of colors nee
col_palette <- colorRampPalette(c("Skyblue", "Purple"))(4)

# Plotting a bar chart of MealTypes frequencies (sorted)
barplot(sorted_meal_type_freq, main = "Distribution of Meal Types",
        xlab = "Meal Type",
        ylab = "Frequency",
        ylim = c(0, max(sorted_meal_type_freq) * 1.2),  # Adjust ylim for better visualization
        col = col_palette,
        names.arg = names(sorted_meal_type_freq), cex.names = 1)


# Define meal types and their descriptions based on the provided information
meal_types <- c("BB", "HB", "FB", "SC")
meal_descriptions <- c("Bed & Breakfast", "Half board (breakfast and one other meal - usually dinner)",
                       "Full board (breakfast, lunch, and dinner)", "No meal package")

# Create a custom legend with meal type descriptions
legend("topright",
       legend = paste(meal_types, ": ", meal_descriptions, sep = ""),
       fill = col_palette,
       bty = "n",  # No box around legend
       title = "Meal Types",  # Legend title
       cex = 0.72)  # Adjust legend label size
```



Step4: Uncovering Patterns and Relationships between variables

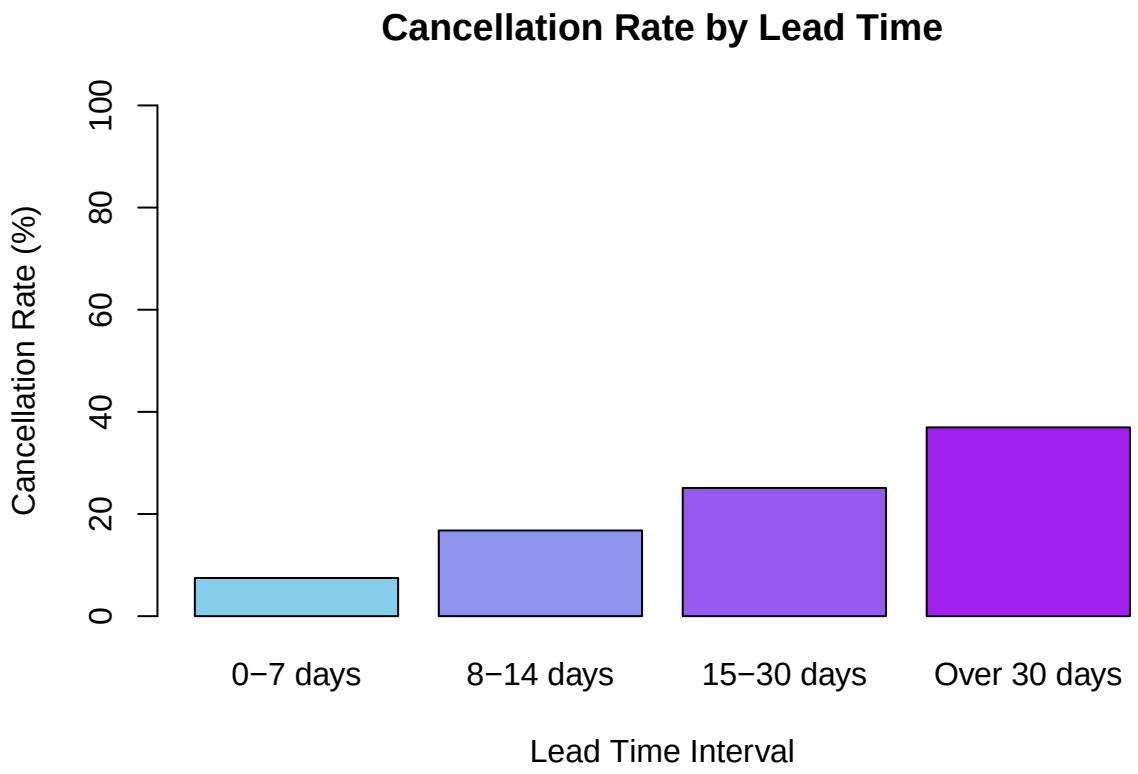
```
# Plotting a box plot of LeadTime by MarketSegment
boxplot(LeadTime ~ MarketSegment, data = hotel_data,
        main = "Lead Time by Market Segment",
        xlab = "Market Segment",
        ylab = "Lead Time (Days)",
        col = "Skyblue",
        las = 2,  # Rotate x-axis labels vertically
        cex.axis = 0.8)  # Adjust axis label size
```



Analysis - The "Groups" market segment shows the longest lead time, indicating that bookings in this segment typically require a longer advance notice before the stay date. - In contrast, the "Complementary" and "Corporate" market segments exhibit the shortest lead times, suggesting a more immediate or last-minute nature of bookings in these segments.

Cancellation Rate by Lead Time

```
# Define lead time intervals (in days)
lead_time_intervals <- cut(hotel_data$LeadTime, breaks = c(0, 7, 14, 30, Inf),
                           labels = c("0-7 days", "8-14 days", "15-30 days", "Over 30 days"))

# Calculate cancellation rate by lead time interval
cancellation_rate_by_lead_time <- tapply(hotel_data$IsCanceled, lead_time_intervals, function(x) mean(x

# Plotting a bar chart of cancellation rates by lead time
barplot(cancellation_rate_by_lead_time,
        main = "Cancellation Rate by Lead Time",
        xlab = "Lead Time Interval",
        ylab = "Cancellation Rate (%)",
        ylim = c(0, 100),  # Set y-axis limit from 0 to 100%
        col = col_palette,
        cex.names = 1,  # Adjust label size
        )
```



The graph illustrates that longer lead times correlate with higher cancellation rates.

Correlation Matrix - A correlation matrix helps identify which variables are positively, negatively, or not significantly correlated with each other.

```
# Select relevant numeric variables and calculate correlation matrix
correlation_matrix <- hotel_data %>%
  select(LeadTime,StaysInWeekendNights, StaysInWeekNights, Adults, Children, Babies, IsCanceled) %>%
  cor()

# Print the correlation matrix
print(correlation_matrix)
```

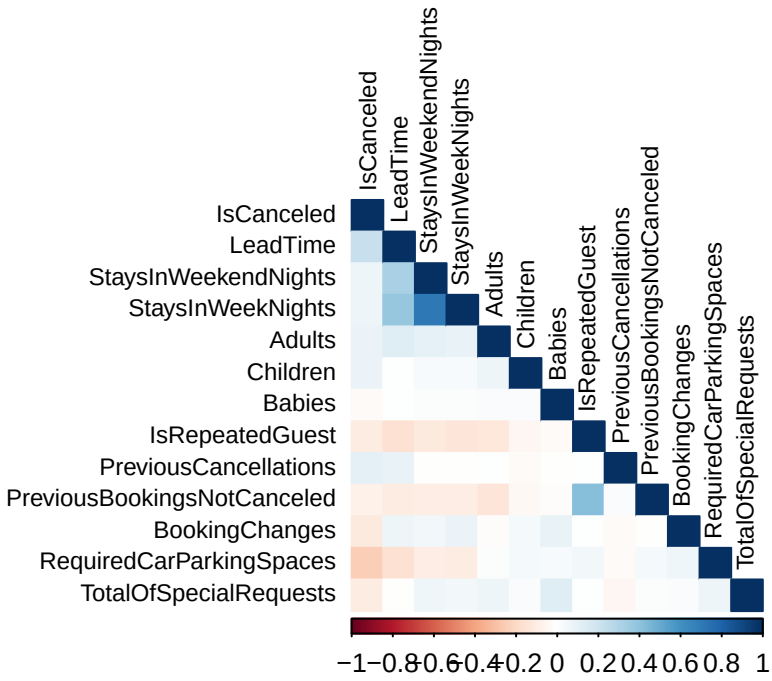
```
##                          LeadTime StaysInWeekendNights StaysInWeekNights
## LeadTime             1.0000000000           0.32571232        0.38760793
## StaysInWeekendNights 0.3257123240           1.00000000        0.71688940
## StaysInWeekNights    0.3876079325           0.71688940        1.00000000
## Adults               0.1367443859           0.10100007        0.09701806
## Children             0.0006396774           0.03925235        0.03367967
## Babies               0.0012563643           0.01503649        0.01442907
## IsCanceled           0.2294438411           0.07856945        0.07847725
##                           Adults     Children       Babies  IsCanceled
## LeadTime             0.13674439 0.0006396774  0.001256364  0.22944384
## StaysInWeekendNights 0.10100007 0.0392523517  0.015036493  0.07856945
## StaysInWeekNights    0.09701806 0.0336796701  0.014429073  0.07847725
## Adults               1.00000000 0.0732459307  0.023164803  0.08054572
```

```
## Children             0.07324593 1.0000000000  0.020414563  0.08123430
## Babies               0.02316480 0.0204145634  1.000000000 -0.02325352
## IsCanceled           0.08054572 0.0812343016 -0.023253522  1.00000000
```

Based on the correlation matrix:

- There is a positive correlation between LeadTime and StaysInWeekendNights (0.33) as well as StaysInWeekNights (0.39). This suggests that longer lead times tend to be associated with longer weekend and weekday stays.
- StaysInWeekendNights and StaysInWeekNights exhibit a strong positive correlation (0.72), indicating that guests who stay longer on weekends also tend to stay longer on weekdays.
- There is a mild positive correlation between Adults and IsCanceled (0.08), suggesting a slight association between the number of adults in bookings and the likelihood of cancellation.
- Similarly, Children and IsCanceled (0.08) show a mild positive correlation, indicating a modest relationship between the presence of children in bookings and cancellation likelihood.
- The correlations involving Babies and IsCanceled (-0.02) are negligible, suggesting minimal impact of infants (babies) on booking cancellations.

```r
# Select numeric variables for correlation analysis
numeric_vars <- hotel_data[, sapply(hotel_data, is.numeric)]

# Calculate correlation matrix
correlation_matrix <- cor(numeric_vars)

# Visualize correlation matrix with customized labels
corrplot(correlation_matrix, method = "color",  type = "lower",
         tl.col = "black",   # Set label color to black
         tl.cex = 0.72,       # Set label size (smaller than default)
         title = "Correlation Matrix of Numeric Variables",
         mar = c(0, 0,1, 0)  # Adjust margin (bottom margin increased to move title closer)
)
```



Contingency table for Meal type

A contingency table is used to display the frequency distribution of two categorical variables and examine the relationship between them. In this case, the contingency table explores the relationship between meal type (BB, FB, HB, SC) and booking cancellation status (0 for not cancelled, 1 for cancelled).

```
# Create contingency table for Meal type vs. IsCanceled
meal_cancel_table <- table(hotel_data$Meal, hotel_data$IsCanceled)

# Display the contingency table
print(meal_cancel_table)
```

```
##
##          0     1
##   BB 22162  7843
##   FB   311   443
##   HB  5499  2547
##   SC   966   289
```

```
# Calculate percentage of cancellations within each category
meal_cancel_table_prop <- prop.table(meal_cancel_table, margin = 1) * 100

# Round percentages to two decimal places
meal_cancel_table_prop_rounded <- round(meal_cancel_table_prop, 2)

# Print the rounded contingency table percentages
print(meal_cancel_table_prop_rounded)
```

```
##
##          0     1
##   BB 73.86 26.14
##   FB 41.25 58.75
##   HB 68.34 31.66
##   SC 76.97 23.03
```

Analysis

Based on the contingency table, it's evident that Full Board (FB) exhibits a high cancellation proportion of 58.75%, indicating that a significant portion of bookings for this meal type are cancelled. Given the widespread popularity of FB bookings, this high cancellation rate translates into a substantial number of cancellations due to the large volume of bookings.

To assess whether there is a significant association between meal types and booking cancellations (both categorical variables), I will conduct a Chi-Squared Test in the next code.

Chi-Squared Test

```
# Perform chi-squared test
chi_squared_test <- chisq.test(meal_cancel_table)
print(chi_squared_test)
```

```
##
##  Pearson's Chi-squared test
##
## data:  meal_cancel_table
## X-squared = 475.35, df = 3, p-value < 2.2e-16
```

Analysis

The chi-squared test results reveal a strong and statistically significant relationship between meal type and booking cancellation:

- The very low p-value (< 2.2e-16) indicates a highly significant association between meal type and booking cancellation status. This suggests that the observed differences in cancellation rates across meal types are unlikely to be random and are likely influenced by the meal type chosen by guests.
- The large value of the Chi-squared statistic (475.35) further supports this evidence of association. Higher values of the test statistic indicate stronger evidence against the null hypothesis of independence, reinforcing the conclusion that meal type plays a significant role in predicting booking cancellations.

Step5: Predictive Modeling

To predict cancellation, a binary outcome, I will utilize logistic regression and decision trees. This approach will allow me to compare their respective accuracy rates. Before proceeding with modeling, it's essential to split the dataset into training and testing sets.

```
# Split data into training and test sets (70% train, 30% test)

set.seed(111) # Set seed for reproducibility

train_indices <- sample(1:nrow(hotel_data), 0.7 * nrow(hotel_data))
train_data <- hotel_data[train_indices, ]
test_data <- hotel_data[-train_indices, ]
```

Logistic Regression

```
# Define and train the logistic regression model
log_model <- glm(IsCanceled ~ LeadTime + MarketSegment + CustomerType + DepositType + PreviousBookingsN
                 data = train_data,
                 family = "binomial")

# Evaluate the model using the test set
predicted_probs <- predict(log_model, newdata = test_data, type = "response")
predicted_labels <- ifelse(predicted_probs > 0.5, 1, 0)

# Evaluate model performance
confusion_matrix <- table(test_data$IsCanceled, predicted_labels)
accuracy <- sum(diag(confusion_matrix)) / sum(confusion_matrix)

# Display model performance metrics
print(confusion_matrix)
```

```
##    predicted_labels
##        0    1
##   0 8451  347
##   1 2343  877
```

```
print(paste("Accuracy:", round(accuracy, 4)))
```

```
## [1] "Accuracy: 0.7762"
```

Logistic Regression Model Result

Accuracy: 77.62%

The logistic regression model achieved an accuracy of 77.62%, indicating its effectiveness in predicting booking cancellations. Despite strong performance in identifying non-canceled bookings (True Negatives = 8451), there were misclassifications, including false positives (347) and false negatives (2343). Experimentation with feature modifications did not yield accuracy improvements. Next, I will explore a decision tree model to capture nonlinear relationships and feature importance.

Decision Tree Model

```
# Train the decision tree model using the rpart function
tree_model <- rpart(IsCanceled ~ LeadTime + MarketSegment + CustomerType + DepositType + Adults +Babies
                    + PreviousCancellations + BookingChanges + RequiredCarParkingSpaces +
                    PreviousBookingsNotCanceled + TotalOfSpecialRequests
                    , data = train_data, method = "class")

# Print a summary of the decision tree model
summary(tree_model)
```

```
## Call:
## rpart(formula = IsCanceled ~ LeadTime + MarketSegment + CustomerType +
##     DepositType + Adults + Babies + PreviousCancellations + BookingChanges +
##     RequiredCarParkingSpaces + PreviousBookingsNotCanceled +
##     TotalOfSpecialRequests, data = train_data, method = "class")
```

```
##   n= 28042
##
##           CP nsplit rel error    xerror        xstd
## 1 0.14249557      0 1.0000000 1.0000000 0.009533596
## 2 0.03483295      1 0.8575044 0.8575044 0.009071688
## 3 0.02708175      5 0.7181726 0.7184257 0.008515348
## 4 0.01000000      6 0.6910909 0.6913440 0.008393178
##
## Variable importance
##                 DepositType                    LeadTime
##                          27                          16
##               MarketSegment      TotalOfSpecialRequests
##                          15                          12
##    RequiredCarParkingSpaces       PreviousCancellations
##                          12                           8
##                CustomerType                      Adults
##                           6                           2
## PreviousBookingsNotCanceled              BookingChanges
##                           1                           1
##
## Node number 1: 28042 observations,    complexity param=0.1424956
##   predicted class=0  expected loss=0.2817916  P(node) =1
##     class counts: 20140  7902
##    probabilities: 0.718 0.282
##   left son=2 (26818 obs) right son=3 (1224 obs)
##   Primary splits:
##       DepositType              splits as  LRL,       improve=1177.2770, (0 missing)
##       LeadTime                 < 14.5  to the left,  improve= 849.1881, (0 missing)
##       MarketSegment            splits as  LLLRLR,    improve= 718.6676, (0 missing)
##       RequiredCarParkingSpaces < 0.5   to the right, improve= 705.3233, (0 missing)
##       PreviousCancellations    < 0.5   to the left,  improve= 502.5858, (0 missing)
##   Surrogate splits:
##       PreviousCancellations < 9     to the left,  agree=0.959, adj=0.051, (0 split)
##
## Node number 2: 26818 observations,    complexity param=0.03483295
##   predicted class=0  expected loss=0.250839  P(node) =0.9563512
##     class counts: 20091  6727
##    probabilities: 0.749 0.251
##   left son=4 (7960 obs) right son=5 (18858 obs)
##   Primary splits:
##       LeadTime                 < 14.5  to the left,  improve=631.7436, (0 missing)
##       RequiredCarParkingSpaces < 0.5   to the right, improve=562.9541, (0 missing)
##       MarketSegment            splits as  LLLLLR,    improve=519.8274, (0 missing)
##       PreviousCancellations    < 0.5   to the left,  improve=268.3153, (0 missing)
##       BookingChanges           < 0.5   to the right, improve=136.1191, (0 missing)
##   Surrogate splits:
##       MarketSegment               splits as  LLLRRR,    agree=0.735, adj=0.107, (0 split)
##       PreviousBookingsNotCanceled < 0.5   to the right, agree=0.731, adj=0.094, (0 split)
##       Adults                      < 1.5   to the left,  agree=0.728, adj=0.083, (0 split)
##       CustomerType                splits as  RLRR,      agree=0.705, adj=0.007, (0 split)
##
## Node number 3: 1224 observations
##   predicted class=1  expected loss=0.04003268  P(node) =0.04364881
##     class counts:    49  1175
```

```
##    probabilities: 0.040 0.960
##
## Node number 4: 7960 observations
##   predicted class=0  expected loss=0.08379397  P(node) =0.2838599
##     class counts:  7293   667
##    probabilities: 0.916 0.084
##
## Node number 5: 18858 observations,    complexity param=0.03483295
##   predicted class=0  expected loss=0.321349  P(node) =0.6724913
##     class counts: 12798  6060
##    probabilities: 0.679 0.321
##   left son=10 (9624 obs) right son=11 (9234 obs)
##   Primary splits:
##       MarketSegment            splits as  LLLLLR,    improve=595.6208, (0 missing)
##       RequiredCarParkingSpaces < 0.5   to the right, improve=474.2463, (0 missing)
##       PreviousCancellations    < 0.5   to the left,  improve=272.7436, (0 missing)
##       CustomerType             splits as  LLRL,      improve=260.7302, (0 missing)
##       BookingChanges           < 0.5   to the right, improve=207.6932, (0 missing)
##   Surrogate splits:
##       CustomerType           splits as  LLRL,      agree=0.710, adj=0.409, (0 split)
##       TotalOfSpecialRequests < 0.5   to the left,  agree=0.682, adj=0.352, (0 split)
##       LeadTime               < 151.5 to the right, agree=0.564, adj=0.110, (0 split)
##       BookingChanges         < 0.5   to the right, agree=0.543, adj=0.067, (0 split)
##       Adults                 < 1.5   to the left,  agree=0.537, adj=0.054, (0 split)
##
## Node number 10: 9624 observations,    complexity param=0.02708175
##   predicted class=0  expected loss=0.1982544  P(node) =0.3431995
##     class counts:  7716  1908
##    probabilities: 0.802 0.198
##   left son=20 (9362 obs) right son=21 (262 obs)
##   Primary splits:
##       PreviousCancellations    < 0.5   to the left,  improve=271.64980, (0 missing)
##       RequiredCarParkingSpaces < 0.5   to the right, improve= 82.95495, (0 missing)
##       BookingChanges           < 0.5   to the right, improve= 61.91997, (0 missing)
##       TotalOfSpecialRequests   < 0.5   to the right, improve= 45.11077, (0 missing)
##       MarketSegment            splits as  RRRRL-,    improve= 33.16870, (0 missing)
##   Surrogate splits:
##       PreviousBookingsNotCanceled < 6.5   to the left,  agree=0.973, adj=0.011, (0 split)
##
## Node number 11: 9234 observations,    complexity param=0.03483295
##   predicted class=0  expected loss=0.4496426  P(node) =0.3292918
##     class counts:  5082  4152
##    probabilities: 0.550 0.450
##   left son=22 (1096 obs) right son=23 (8138 obs)
##   Primary splits:
##       RequiredCarParkingSpaces < 0.5   to the right, improve=502.86070, (0 missing)
##       TotalOfSpecialRequests   < 0.5   to the right, improve=318.16040, (0 missing)
##       CustomerType             splits as  -LRL,      improve=150.50860, (0 missing)
##       LeadTime                 < 97.5  to the left,  improve= 93.18764, (0 missing)
##       BookingChanges           < 0.5   to the right, improve= 75.54929, (0 missing)
##
## Node number 20: 9362 observations
##   predicted class=0  expected loss=0.1783807  P(node) =0.3338564
##     class counts:  7692  1670
```

```
##    probabilities: 0.822 0.178
##
## Node number 21: 262 observations
##   predicted class=1  expected loss=0.09160305  P(node) =0.009343128
##     class counts:    24   238
##    probabilities: 0.092 0.908
##
## Node number 22: 1096 observations
##   predicted class=0  expected loss=0  P(node) =0.03908423
##     class counts:  1096     0
##    probabilities: 1.000 0.000
##
## Node number 23: 8138 observations,    complexity param=0.03483295
##   predicted class=1  expected loss=0.4898009  P(node) =0.2902075
##     class counts:  3986  4152
##    probabilities: 0.490 0.510
##   left son=46 (5379 obs) right son=47 (2759 obs)
##   Primary splits:
##       TotalOfSpecialRequests < 0.5   to the right, improve=299.25100, (0 missing)
##       CustomerType           splits as  -LRL,      improve=166.50980, (0 missing)
##       LeadTime               < 134.5 to the left,  improve= 72.22473, (0 missing)
##       BookingChanges         < 0.5   to the right, improve= 59.25952, (0 missing)
##       PreviousCancellations  < 0.5   to the left,  improve= 42.43804, (0 missing)
##   Surrogate splits:
##       Adults   < 2.5   to the left,  agree=0.667, adj=0.017, (0 split)
##       LeadTime < 357.5 to the left,  agree=0.661, adj=0.001, (0 split)
##
## Node number 46: 5379 observations
##   predicted class=0  expected loss=0.4130879  P(node) =0.1918194
##     class counts:  3157  2222
##    probabilities: 0.587 0.413
##
## Node number 47: 2759 observations
##   predicted class=1  expected loss=0.3004712  P(node) =0.09838813
##     class counts:   829  1930
##    probabilities: 0.300 0.700
```

Decision Tree Model Result

- The decision tree model highlights several key predictors for booking cancellations.
- The most influential features include DepositType, LeadTime, MarketSegment, TotalOfSpecialRequests, RequiredCarParkingSpaces, PreviousCancellations, and CustomerType, emphasizing their significant impact on predicting cancellations.
- In contrast, features such as Adults, PreviousBookingsNotCanceled, and BookingChanges show relatively lower importance in this predictive model.