Project_1.3

Ruturaj Solanki, Kunal Vaghela, Satya Akhil Govvala

2023-04-23

R and RStudio Versions

- R version 4.2.2 (2022-10-31 ucrt)
- RStudio 2022.12.0+353 "Elsbeth Geranium" Release (7d165dcfc1b6d300eb247738db2c7076234f6ef0, 2022-12-03) for Windows, Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) RStudio/2022.12.0+353 Chrome/102.0.5005.167 Electron/19.1.3 Safari/537.36

R Package Used

- ggplot2 (3.4.1)
- ggthemes (4.2.4)
- caret (6.0.94)
- caTools (1.18.2)
- car (3.1.2)
- DMwR (0.4.1)
- glmnet (4.1.7)
- ResourceSelection (0.3.5)
- naniar (1.0.o)

```
library("ggplot2")
library("ggthemes")
library("naniar")
library("caret")
```

```
## Loading required package: lattice
```

```
library("caTools")
library("car")
```

```
## Loading required package: carData
```

```
library("DMwR")
```

```
## Loading required package: grid
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```r
library("nnet")
library("glmnet")
```

```
## Loading required package: Matrix

## Loaded glmnet 4.1-7
```

```r
library("ResourceSelection")
```

```
## ResourceSelection 0.3-5   2019-07-22
```

Data defination and visiualisation

```r
df <- read.csv("HRD.csv", header=TRUE, stringsAsFactors=FALSE)
head(df)
```

```
##   Booking_ID no_of_adults no_of_children no_of_weekend_nights no_of_week_nights
## 1   INN00002            2              0                    2                 3
## 2   INN00003            1              0                    2                 1
## 3   INN00005            2              0                    1                 1
## 4   INN00007            2              0                    1                 3
## 5   INN00008            2              0                    1                 3
## 6   INN00009            3              0                    0                 4
##   type_of_meal_plan required_car_parking_space room_type_reserved lead_time
## 1      Not Selected                          0        Room_Type 1         5
## 2       Meal Plan 1                          0        Room_Type 1         1
## 3      Not Selected                          0        Room_Type 1        48
## 4       Meal Plan 1                          0        Room_Type 1        34
## 5       Meal Plan 1                          0        Room_Type 4        83
## 6       Meal Plan 1                          0        Room_Type 1       121
##   arrival_year arrival_month arrival_date market_segment_type repeated_guest
## 1         2018            11            6              Online              0
## 2         2018             2           28              Online              0
## 3         2018             4           11              Online              0
## 4         2017            10           15              Online              0
## 5         2018            12           26              Online              0
## 6         2018             7            6             Offline              0
##   no_of_previous_cancellations no_of_previous_bookings_not_canceled
## 1                            0                                    0
## 2                            0                                    0
## 3                            0                                    0
## 4                            0                                    0
## 5                            0                                    0
## 6                            0                                    0
##   avg_price_per_room no_of_special_requests booking_status sqrt_lead_time
## 1             106.68                      1   Not_Canceled       2.236068
## 2              60.00                      0       Canceled       1.000000
## 3              94.50                          0       Canceled       6.928203
## 4             107.55                          1   Not_Canceled       5.830952
## 5             105.61                          1   Not_Canceled       9.110434
## 6              96.90                          1   Not_Canceled      11.000000
```

```
n_miss(df)
```

```
## [1] 0
```

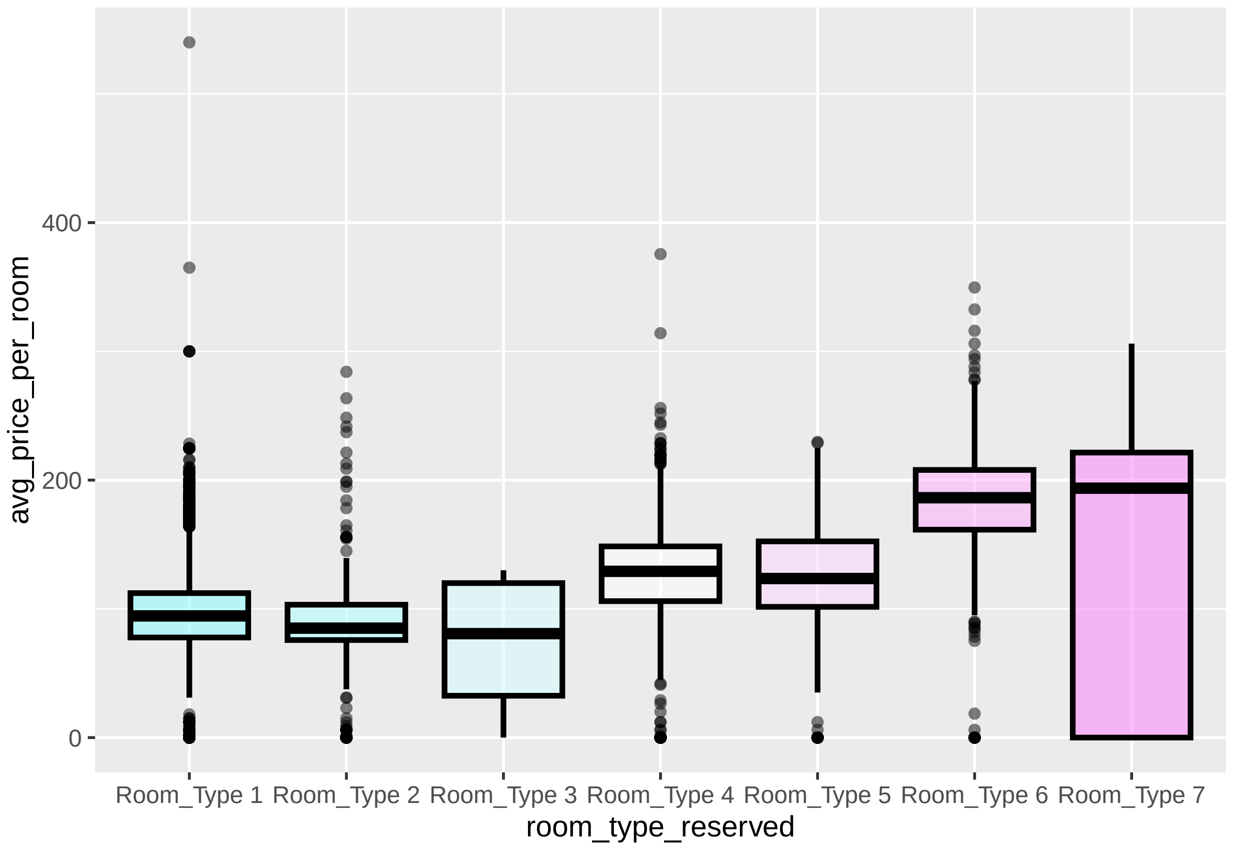
```
any_na(df)
```

```
## [1] FALSE
```

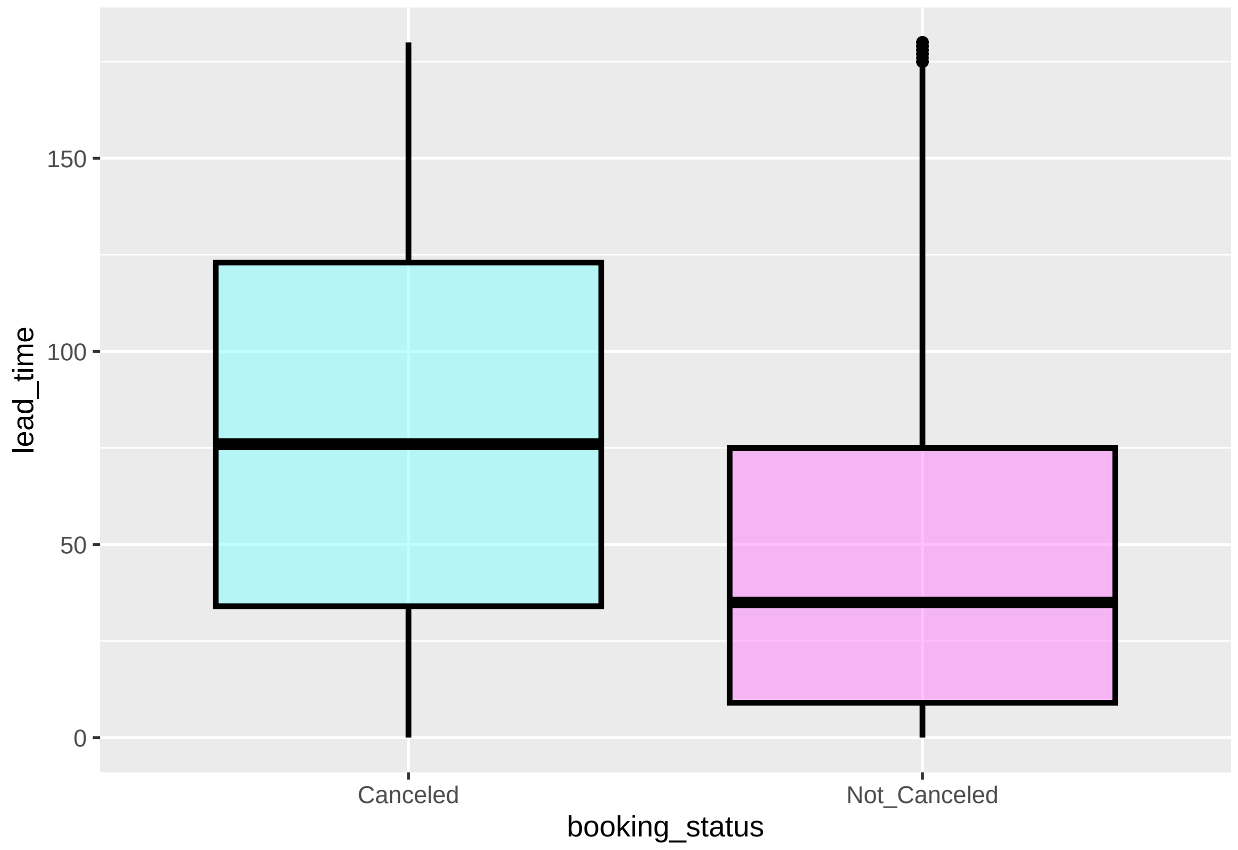
```
miss_var_summary(df)
```

```
## # A tibble: 20 x 3
##    variable                             n_miss pct_miss
##    <chr>                                 <int>    <dbl>
##  1 Booking_ID                                0        0
##  2 no_of_adults                              0        0
##  3 no_of_children                            0        0
##  4 no_of_weekend_nights                      0        0
##  5 no_of_week_nights                         0        0
##  6 type_of_meal_plan                         0        0
##  7 required_car_parking_space                0        0
##  8 room_type_reserved                        0        0
##  9 lead_time                                 0        0
## 10 arrival_year                              0        0
## 11 arrival_month                             0        0
## 12 arrival_date                              0        0
## 13 market_segment_type                       0        0
## 14 repeated_guest                            0        0
## 15 no_of_previous_cancellations              0        0
## 16 no_of_previous_bookings_not_canceled      0        0
## 17 avg_price_per_room                        0        0
## 18 no_of_special_requests                    0        0
## 19 booking_status                            0        0
## 20 sqrt_lead_time                            0        0
```

```
ggplot(df)+
  geom_boxplot(aes(room_type_reserved, avg_price_per_room), color = "black", fill = cm.colors(7), size
```



```
ggplot(df)+
  geom_boxplot(aes(booking_status, lead_time), color = "black", fill = cm.colors(2), size = 1, alpha =
```



Q1 - Identify the explanatory variable.

- Answer: In earlier stages of these projects we have identified two exploratory variables.
- avg__price__per__room
- lead__time

Q2 - Identify the response variable.

- Answer: Above two exploratory variables have these two response variables.
- room__type__reserved (dependent on avg__price__per__room)
- booking__status (dependent on lead__time)

  Balancing the Data.

room__type__reserved and booking__status our response variables.

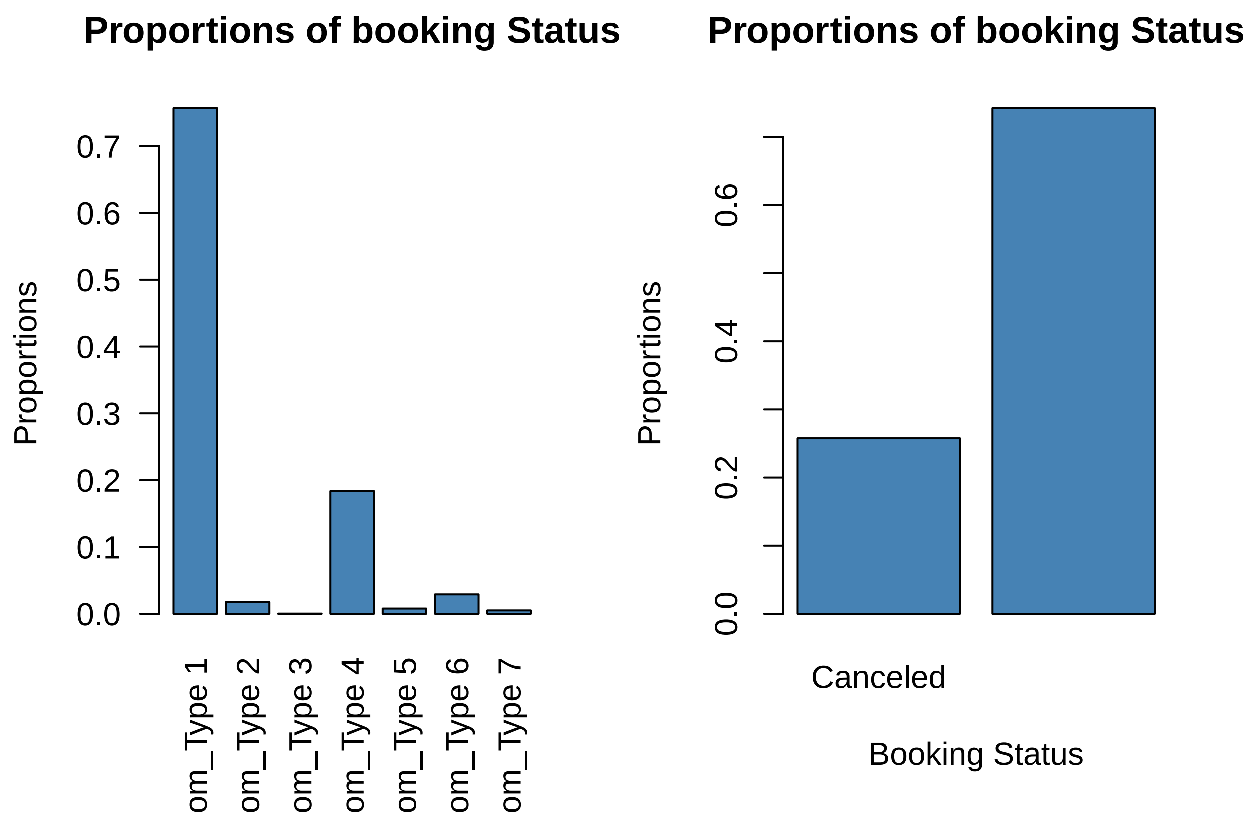
```
# Create a table of proportions for room types
prop.table(table(df$room_type_reserved))
```

```
##
## Room_Type 1 Room_Type 2 Room_Type 3 Room_Type 4 Room_Type 5 Room_Type 6
## 0.756855281 0.017388219 0.000225821 0.183657010 0.007806955 0.029034131
## Room_Type 7
## 0.005032583
```

```
# Create a table of proportions for booking status
prop.table(table(df$booking_status))
```

```
##
##     Canceled Not_Canceled
##    0.2576295    0.7423705
```

```
# Set up the plot area
par(mfrow = c(1, 2))

# Plot the first barplot
barplot(prop.table(table(df$room_type_reserved)), main = "Proportions of booking Status",
        ylab = "Proportions", col = "steelblue", las = 2)

# Plot the second barplot
barplot(prop.table(table(df$booking_status)), main = "Proportions of booking Status", xlab = "Booking St
        ylab = "Proportions", col = "steelblue")
```



```
table(df$room_type_reserved)
```

As we can see both of these variables are imbalanced. So we will need to balance both of them before applying logistic regression.

```
##
## Room_Type 1 Room_Type 2 Room_Type 3 Room_Type 4 Room_Type 5 Room_Type 6
##       23461         539           7        5693         242         900
## Room_Type 7
##         156
```

```
table(df$booking_status)
```

```
##
##     Canceled Not_Canceled
##         7986        23012
```

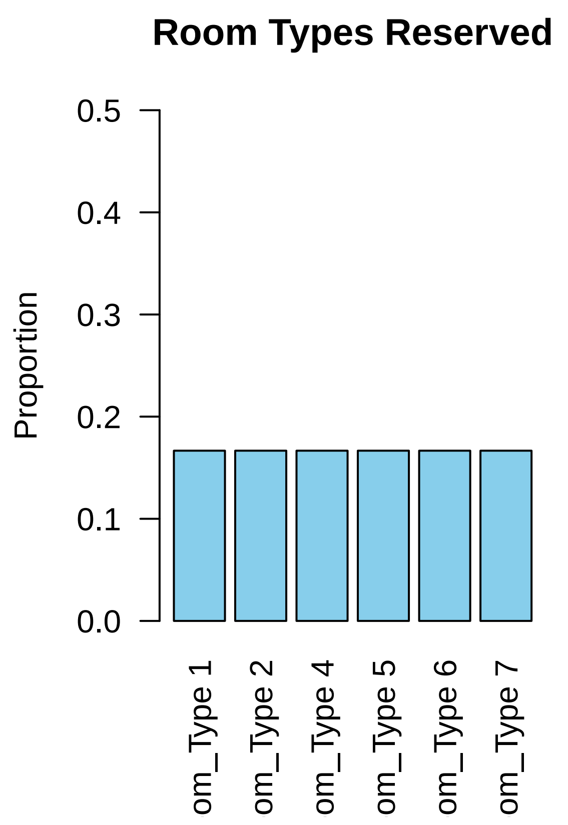
```
df <- df[df$room_type_reserved != "Room_Type 3", ]
```

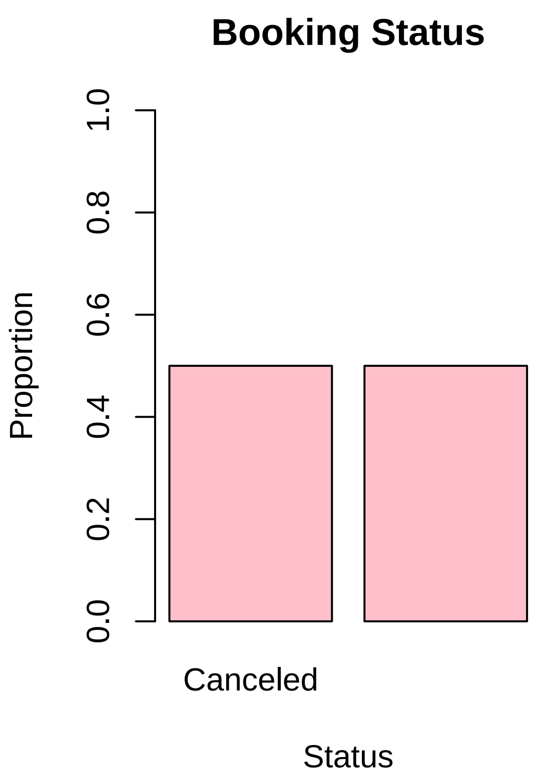
We will remove the room_type 3 as It has only 7 values. Also, there is huge imbalance inbetween catagories. Thus, Many complex balancing techniques will fail such as SMOTE.

```
#Balancing room type reserved using upsample as most of them have very few values.
balance_data <- upSample(x = df[, -ncol(df)], y=factor(df$room_type_reserved))

#Balancing booking status using downsample
df_balanced <- downSample(x = df[, -ncol(df)], y = factor(df$booking_status))
```

```
# Combine the two bar graphs side by side
par(mfrow = c(1, 2))

# Create the first bar graph for room type
barplot(prop.table(table(balance_data$room_type_reserved)),
        main = "Room Types Reserved",
        ylab = "Proportion",
        col = "skyblue",
        ylim = c(0, 0.5),
        las = 2)

# Create the second bar graph for booking status
barplot(prop.table(table(df_balanced$booking_status)),
        main = "Booking Status",
        xlab = "Status",
        ylab = "Proportion",
        col = "pink",
        ylim = c(0, 1))
```

We will use upsampling for room__type__reserved as we would be left with very few value if we did downsampling. Whereas, for booking__status we will do upsampling.





- Outliers in avg_price_per_room

```
summary(balance_data$avg_price_per_room)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     0.0    87.2   122.5   128.6   174.0   540.0
```

- Let's remove the outliers

```
P_q1 = quantile(balance_data$avg_price_per_room, 0.25)
P_q3 = quantile(balance_data$avg_price_per_room, 0.75)

P_iqr = IQR(balance_data$avg_price_per_room)

lowest_price = P_q1 - (1.5 * P_iqr)
Highest_price = P_q3 + (1.5 * P_iqr)
```

Q3 - Create a logistic regression model and display the full output of the model. (As we do not have any pair of continuous varible which can be identified as explanatory and response variables. Performing a linear regression is not an option.)

- Answer: First performing the logistic regression on avg_price_per_room and room_type_reserved.

```
# split the data into training and testing sets
set.seed(100)
train_indices <- sample(nrow(balance_data), 0.7 * nrow(balance_data))
train <- balance_data[train_indices, ]
test <- balance_data[-train_indices, ]

# convert room_type_reserved to factor
train$room_type_reserved <- factor(train$room_type_reserved)

# train the model
multinom_model <- multinom(room_type_reserved ~ avg_price_per_room, data = train)
```

```
## # weights:  18 (10 variable)
## initial  value 176552.811060
## iter  10 value 162563.196495
## final  value 161047.279506
## converged
```

```
summary(multinom_model)
```

```
## Call:
## multinom(formula = room_type_reserved ~ avg_price_per_room, data = train)
##
## Coefficients:
##             (Intercept) avg_price_per_room
## Room_Type 2   0.2464259       -0.002766743
## Room_Type 4  -1.1523014        0.010346983
## Room_Type 5  -1.0314876        0.009387873
## Room_Type 6  -4.1656460        0.029844099
## Room_Type 7  -2.6069937        0.020654229
##
## Std. Errors:
##             (Intercept) avg_price_per_room
## Room_Type 2  0.02178632       0.0002040819
## Room_Type 4  0.02618942       0.0002120920
## Room_Type 5  0.02577298       0.0002108840
## Room_Type 6  0.03725258       0.0002461942
## Room_Type 7  0.03114843       0.0002266155
##
## Residual Deviance: 322094.6
## AIC: 322114.6
```

```
# get the intercepts for all categories
coef_mat <- coef(multinom_model)
intercepts <- coef_mat[,1]
intercepts
```

```
## Room_Type 2 Room_Type 4 Room_Type 5 Room_Type 6 Room_Type 7
##   0.2464259  -1.1523014  -1.0314876  -4.1656460  -2.6069937
```

- Now we will perform logistic regression on booking_status and lead_time.

```r
# Create a new binary variable for booking status
df_balanced$binary_booking_status <- ifelse(df_balanced$booking_status == "Canceled", 1, 0)

# Train logistic regression model
model <- glm(binary_booking_status ~ lead_time, data = df_balanced, family = "binomial")
```

```r
summary(model)
```

```
##
## Call:
## glm(formula = binary_booking_status ~ lead_time, family = "binomial",
##     data = df_balanced)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.878  -1.023  -0.113   1.107   1.558
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.8614904  0.0269686  -31.94   <2e-16 ***
## lead_time    0.0135339  0.0003429   39.46   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 22136  on 15967  degrees of freedom
## Residual deviance: 20373  on 15966  degrees of freedom
## AIC: 20377
##
## Number of Fisher Scoring iterations: 4
```

Q4 - Using the variables noted in #1 and #2 above and the results of #3, write the equation for your model.

- Answer 4.1 : log(p(Room_Type = i) / p(Room_Type = 1)) = x0 + x1 * avg_price_per_room ; Where room_type 1 is refrence catagory and i is other catagories. x0 is the Intercept, and x1 is the estimated coefficient (slope) of avg_price_per_room where negative value suggests that with the increase in avg_price_per_room the likelyhood of being in room_type i decreases.
- Answer 4.2 : log(p/(1-p)) = -0.854157 + 0.013425 * lead_time; Where P is the probability of canceled booking. (-0.854157) is the entercept and 0.013425 is the slope.

Q5) What the Intercept means in context of the data.

- Answer 5.1 : It means that the intercept represents the log-odds of the probability that an observation belongs to Room_Type 1 (as it is the refrense catagory), when the average price per room is zero. But we have excluded the 0 values in our dataset as room prices, in most of the cases will be greater than 0.
- Answer 5.2 : the intercept represents the estimated log odds of the dependent variable (binary_booking_status) when the value of the predictor variable (lead_time) is zero. Which basically the baseline probability of booking being canceled or not.

Q6) Is the intercept a useful/meaningful value in the context of our data? If yes, explain. If not, explain what purpose it serves.

- Answer 6.1 : Yes and No, the intercept is basically the log odds of the refrense catagory(response) when all the predictor are equal to zero. As we have excluded 0 values it does not make much of a difference. But on the other hand, it provides us with the baseline for comparing every catagories (i.e. all the room types).
- Answer 6.2 : Yes, the intercept (-0.85) in this case suggests that when lead_time is 0 (booking is done near to arrival date) the probability of booking being canceled is very less(i.e. not cancelled).

Q7) Explain what the slope means in the context of the data.

- Answer 7.1 : The changes in log odds of the room_type_reserved with an increase in avg_price_per_room is denoted as slope. When the slope is positive it denotes the increase in log odds of that perticular type and if it's negative the opposite is true.
- Answer 7.2 : The slope of lead_time is 0.013425, which means with the increase in lead_time the log odds of booking being canclled will increase by the value of slope (0.013).

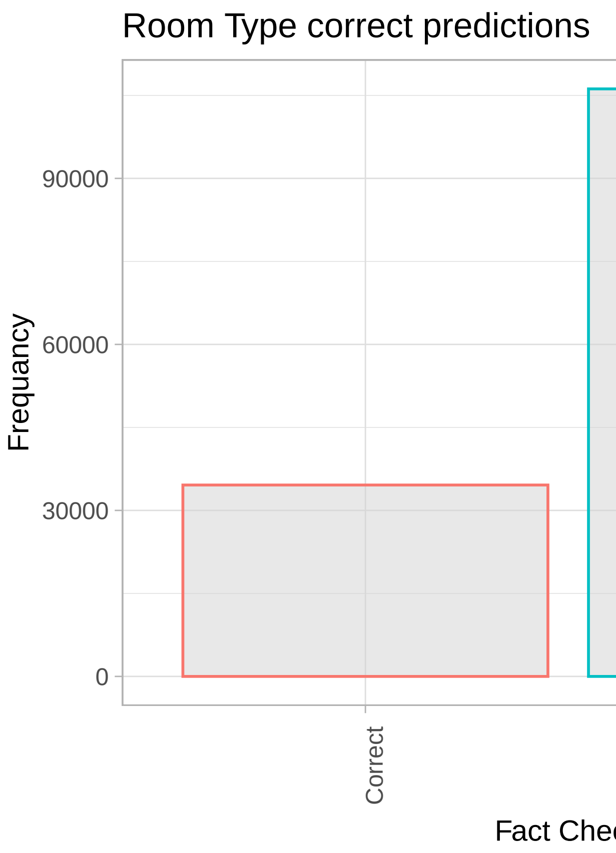
Model Diagnostics

Q1 Create two new data columns based on your best model: predicted values for your response variable and the corresponding residuals.

- Answer : As these are catagorical variables we will just show that if the prediction is correct or not instead of residuals.

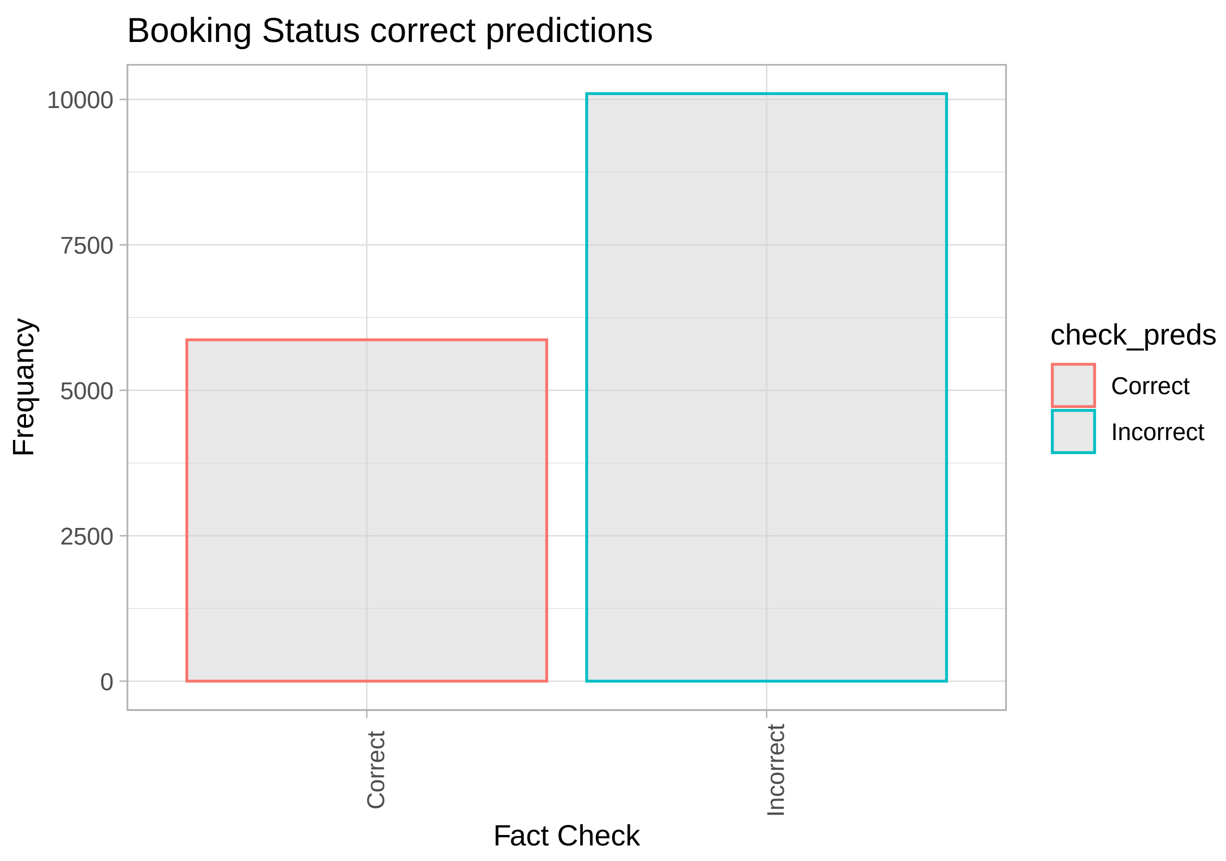
```
# get predicted probabilities for each category
probs <- predict(multinom_model, newdata = balance_data, type = "probs")

# calculate predicted values (category with maximum predicted probability)
balance_data$Room_Type_prediction <- paste0("Room_Type ", apply(probs, 1, which.max))

# Create check column for checking the predictions are true or not?
balance_data$check <- ifelse(balance_data$Class == balance_data$Room_Type_prediction, "Correct", "Incor

# Predict binary_booking_status using the trained logistic regression model
df_balanced$predictions <- predict(model, newdata = df_balanced, type = "response")

# Convert predictions to binary values
df_balanced$predictions_binary <- ifelse(df_balanced$predictions > 0.5, "Not_Canceled", "Canceled")

# Check if predictions are correct
df_balanced$check_preds <- ifelse(df_balanced$predictions_binary == df_balanced$booking_status, "Correc
```

```
ggplot(balance_data)+
  geom_bar(aes(check, color = check), fill = "lightgrey", alpha = 0.5)+
  labs(
    title = "Room Type correct predictions",
    x = "Fact Check",
    y = "Frequancy"
  )+
  theme_light()+
  theme(axis.text.x = element_text(angle = 90))
```



Checking how both of our models work on the whole dataset.

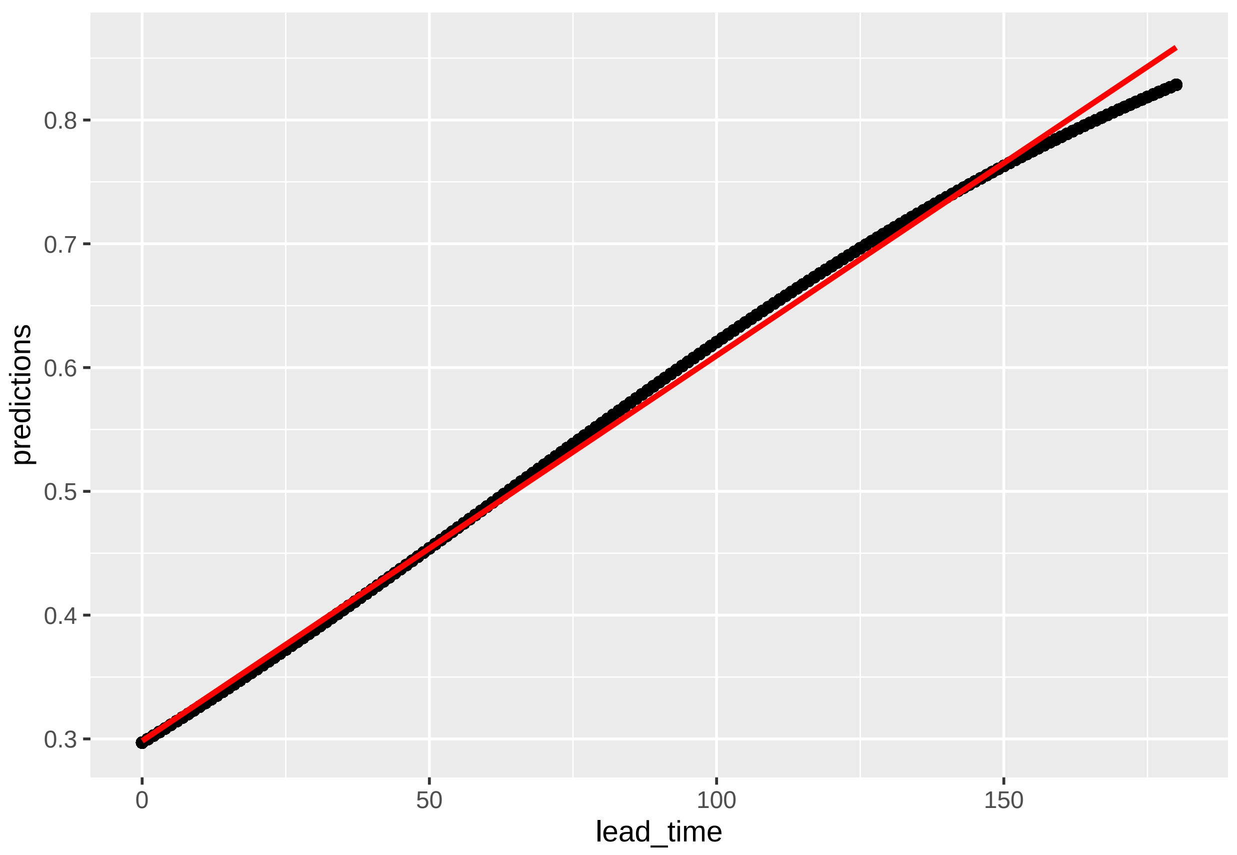
```
ggplot(df_balanced)+
  geom_bar(aes(check_preds, color = check_preds), fill = "lightgrey", alpha = 0.5)+
  labs(
    title = "Booking Status correct predictions",
    x = "Fact Check",
    y = "Frequancy"
  )+
  theme_light()+
  theme(axis.text.x = element_text(angle = 90))
```



Q2 Checking for linearity between independent variable and log odds of the dependent variable.

```
ggplot(df_balanced, aes(x = lead_time, y = predictions)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE, color = "red")
```

```
## 'geom_smooth()' using formula = 'y ~ x'
```



- Answer : As we can see, there is clear and consistent linear pattern, it suggests that linearity assumption in correct and our model is correct for our data.

Q3 - Goodness of Fit of the model

```
summary(multinom_model)
```

```
## Call:
## multinom(formula = room_type_reserved ~ avg_price_per_room, data = train)
##
## Coefficients:
##              (Intercept) avg_price_per_room
## Room_Type 2   0.2464259       -0.002766743
## Room_Type 4  -1.1523014        0.010346983
## Room_Type 5  -1.0314876        0.009387873
## Room_Type 6  -4.1656460        0.029844099
## Room_Type 7  -2.6069937        0.020654229
##
## Std. Errors:
##              (Intercept) avg_price_per_room
## Room_Type 2  0.02178632       0.0002040819
## Room_Type 4  0.02618942       0.0002120920
## Room_Type 5  0.02577298       0.0002108840
## Room_Type 6  0.03725258       0.0002461942
## Room_Type 7  0.03114843       0.0002266155
##
## Residual Deviance: 322094.6
## AIC: 322114.6
```

```
summary(model)
```

```
##
## Call:
## glm(formula = binary_booking_status ~ lead_time, family = "binomial",
##     data = df_balanced)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.878  -1.023  -0.113   1.107   1.558
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.8614904  0.0269686  -31.94   <2e-16 ***
## lead_time    0.0135339  0.0003429   39.46   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 22136  on 15967  degrees of freedom
## Residual deviance: 20373  on 15966  degrees of freedom
## AIC: 20377
##
## Number of Fisher Scoring iterations: 4
```

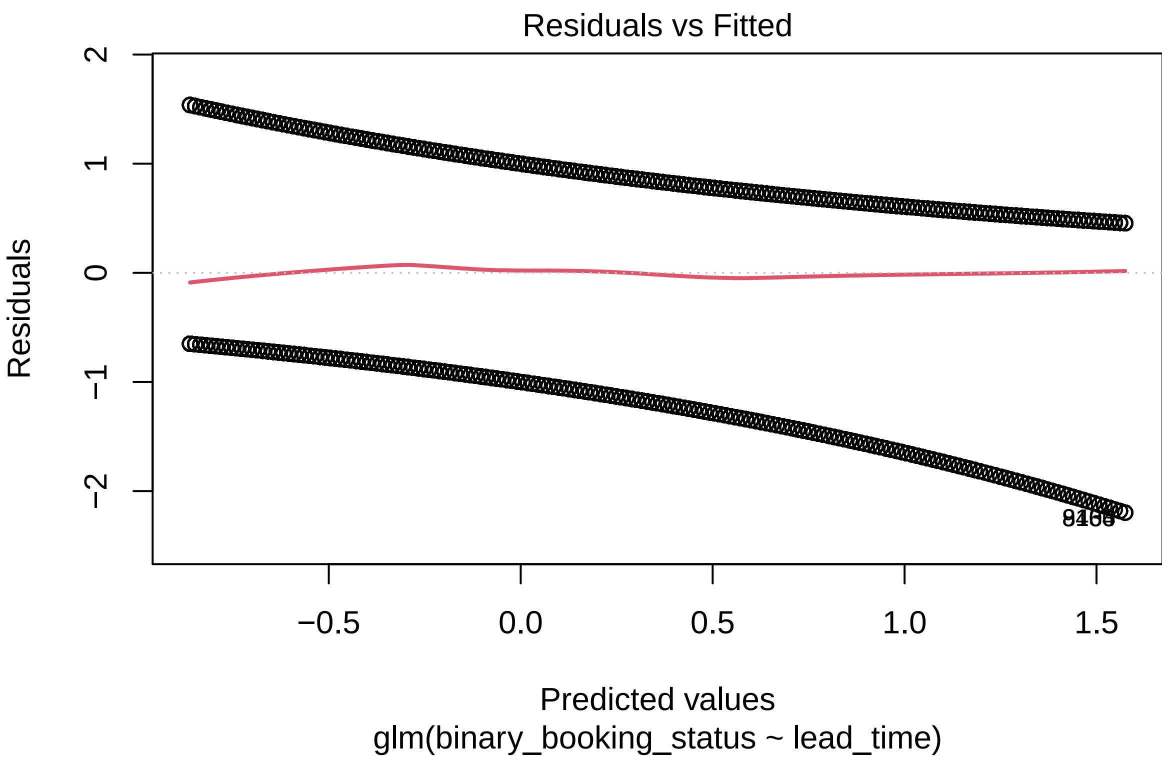
```
# Conduct Hosmer-Lemeshow goodness-of-fit test
hoslem.test(df_balanced$binary_booking_status, fitted(model))
```

```
##
##  Hosmer and Lemeshow goodness of fit (GOF) test
##
## data:  df_balanced$binary_booking_status, fitted(model)
## X-squared = 324.17, df = 8, p-value < 2.2e-16
```

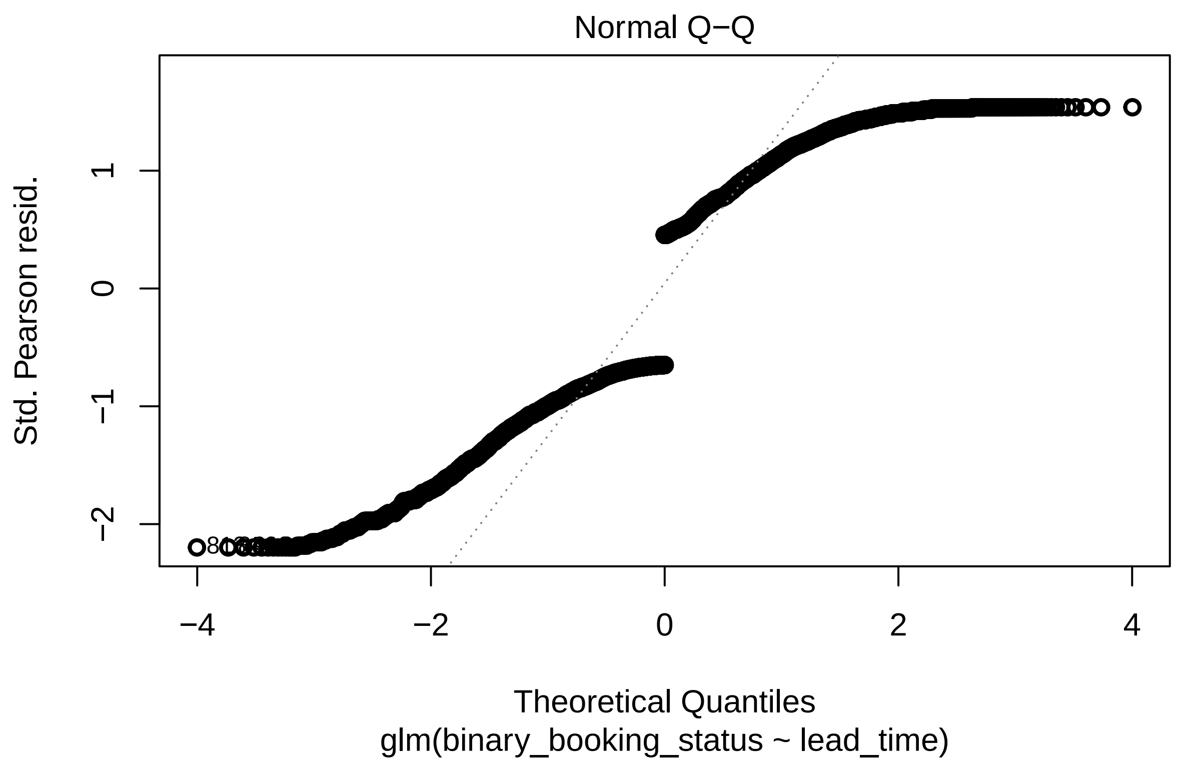
- Answer 3.1: As we can see the AIC value and deviance both are quite high, and this suggest that model in not a well fitted one.
- Answer 3.2: As we can see in the output the AIC value is very high which means that the model is not well fitted with data. To add to this, Hosmer and Lemeshow goodness of fit (GOF) test also gives very low p-value which suggests the same. I.e. there are also other factors contributing to booking cancellation other than lead time.

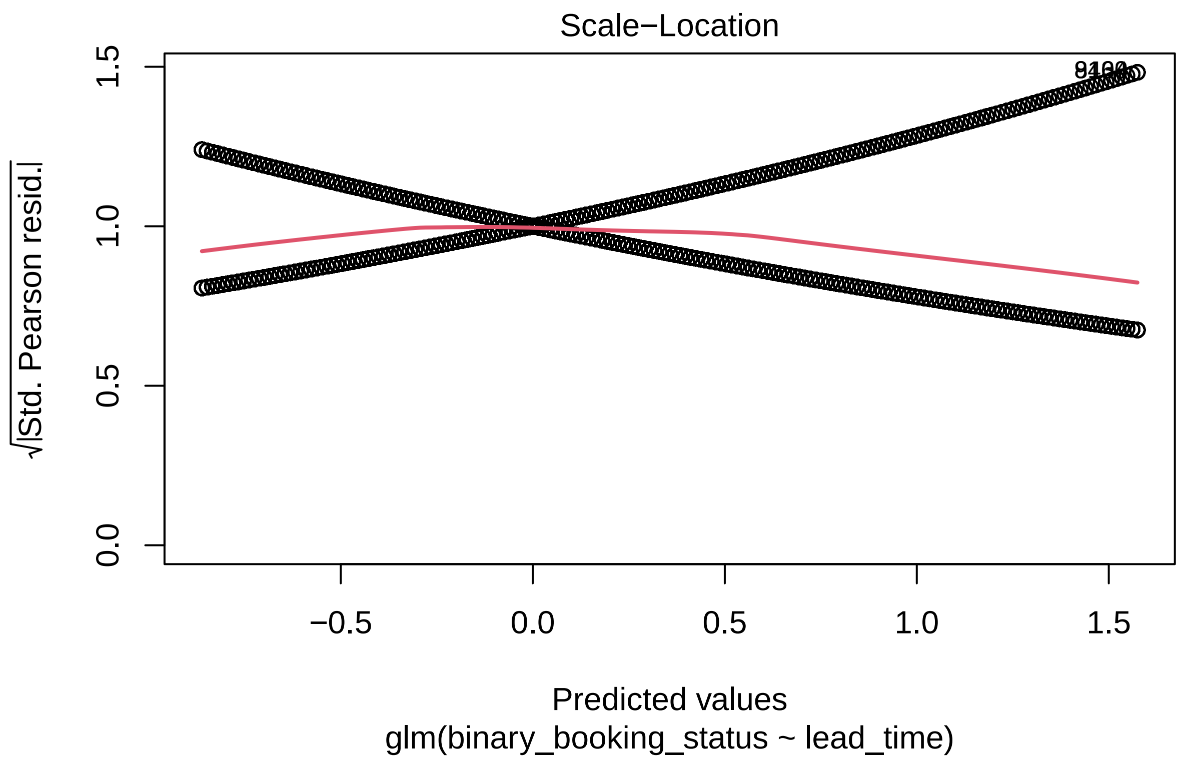
  Understanding the model

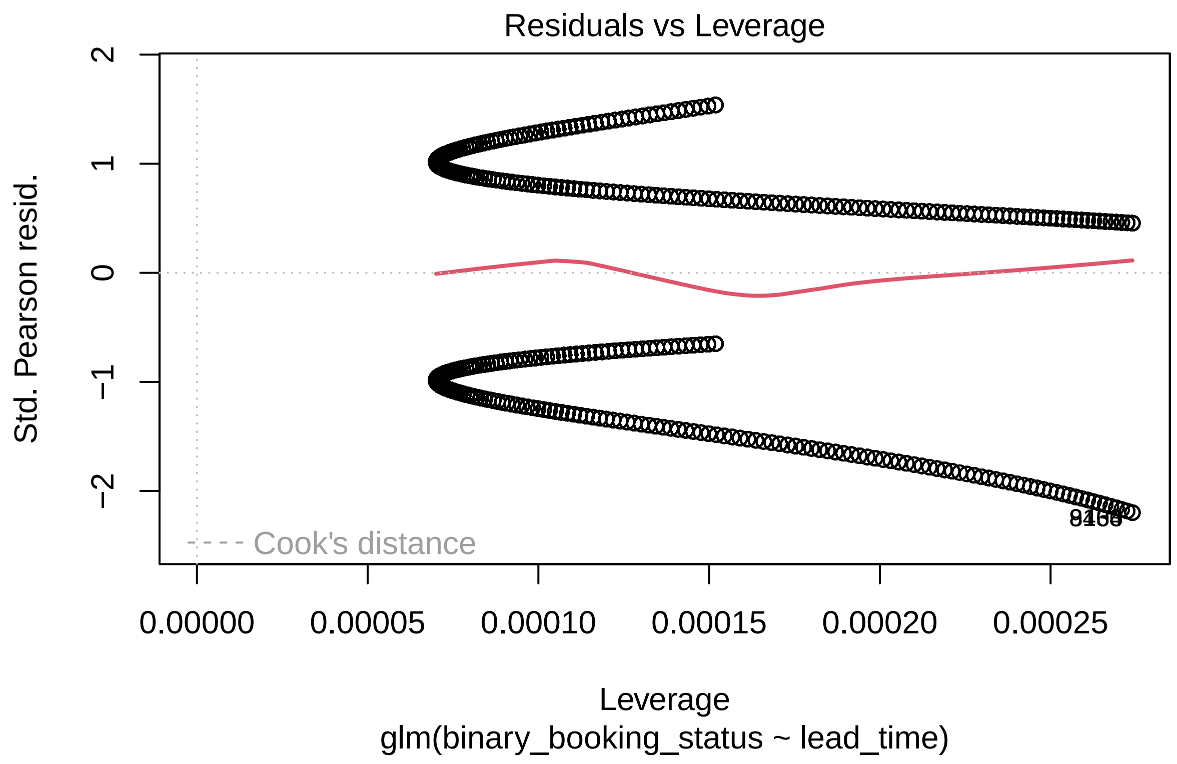
```r
# plot the model
plot(model, col = "black", lwd = 2)
```



Plotting the model




Scale−Location
√|Std. Pearson resid.|
0.0
0.5
1.0
1.5
−0.5
0.0
0.5
1.0
1.5
Predicted values
glm(binary_booking_status ~ lead_time)
9100



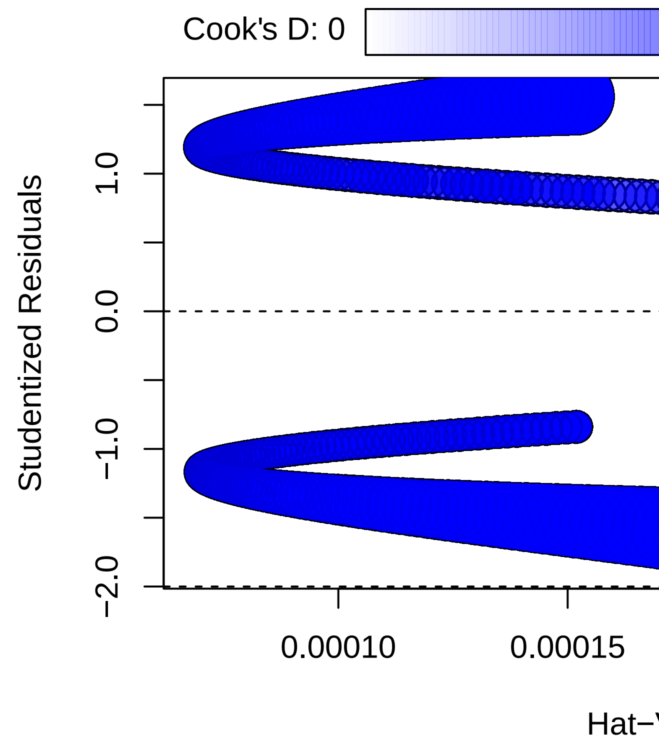
```
# add a title and axis labels
xlab("Lead Time")
```

```
## $x
## [1] "Lead Time"
##
## attr(,"class")
## [1] "labels"
```

```
ylab("Probability of Booking")
```

```
## $y
## [1] "Probability of Booking"
##
## attr(,"class")
## [1] "labels"
```

```
influencePlot(model)
```



Plotting Influencial plot to check for the influencial points.

```
##         StudRes          Hat       CookD
## 91    0.6135774 0.0002739703 2.838344e-05
## 108   0.6135774 0.0002739703 2.838344e-05
## 8134 -1.8780238 0.0002739703 6.618477e-04
## 8400 -1.8780238 0.0002739703 6.618477e-04
```

Conclusion

```
# Convert both predicted and actual values to factors with the same levels
pred <- factor(round(df_balanced$predictions), levels = c("0", "1"))
actual <- factor(df_balanced$binary_booking_status, levels = c("0", "1"))

# Calculate the accuracy using confusionMatrix() from the caret package
accuracy <- confusionMatrix(pred, actual)$overall[1]

# Print the accuracy
cat("Accuracy:", accuracy, "\n")
```

Checking for the accuracy of the model.

```
## Accuracy: 0.632515
```

- In the conclusion, we can safely say that the accuracy of our model is average (63%) and it needs to be hypertuned.
- Also, the response variable's variablity is not just dependent on our predictor variable, there are other hidden factors affecting the response variable as well.
- The multinominal logistic regression model is also performing poorly. The reason might be the data imbalnce.
- the influence plot show influencial points which are at the top left and bottom right corner.

Refrences

1) https://stats.stackexchange.com/questions/297716/how-to-write-a-regression-equation-using-the-output-from-a-summary
2) https://r-graph-gallery.com/index.html
3) https://www.geeksforgeeks.org/
4) https://www.youtube.com/results?search_query=how+to+perform+oversampling+in+R
5) https://stats.oarc.ucla.edu/r/dae/multinomial-logistic-regression/
6) https://www.rstudio.com/wp-content/uploads/2015/02/rmarkdown-cheatsheet.pdf
7) https://www.analyticsvidhya.com/blog/
8) https://stats.stackexchange.com/questions/65244/how-to-determine-the-accuracy-of-logistic-regression-in-r
9) https://topepo.github.io/caret/
10) https://rdrr.io/cran/car/src/R/influencePlot.R
11) https://search.r-project.org/CRAN/refmans/sjPlot/html/plot_model.html
12) https://towardsdatascience.com/visualizing-models-101-using-r-c7c937fc5f04
13) https://topepo.github.io/caret/subsampling-for-class-imbalances.html