Strategizing Airfare Optimization Employing Regression Analysis to Forecast Flight Ticket Prices

1. Introduction

In the dynamic aviation industry, understanding the factors influencing flight ticket prices is crucial for airlines, travel agencies, and passengers. This analysis delves into a comprehensive dataset encompassing flight transaction details to uncover the relationships between various variables and their impact on ticket prices through regression analysis techniques. The central focus lies in identifying the key determinants of price fluctuations, such as airline, source and destination cities, travel timings, number of stops, service class, flight duration, and booking lead time.

1.1 Project Motivation and Objectives The motivation behind this project stems from the pivotal role that flight pricing plays in the aviation sector. Accurate price analysis can empower airlines to refine their revenue strategies, enhance competitiveness, and better cater to customer demands. Furthermore, it enables travelers to make informed decisions, potentially resulting in cost savings and a more transparent booking experience.

- The core objectives of this analysis are:
- Analyzing the relationships between flight characteristics such as airline, travel timings, service class, and ticket prices.
- Understanding the impact of factors like airline preferences, number of stops, flight duration, booking lead time, and more on ticket prices.
- Offering valuable insights to airlines, travel agencies, and policymakers for optimizing pricing strategies and enhancing overall industry efficiency.

1.2 Data Description The dataset utilized for this project, sourced from Kaggle's Flight Price Prediction Dataset, represents a comprehensive repository of flight transaction data invaluable for predicting ticket prices. Encompassing a rich array of information such as airline names, flight identifiers, source and destination cities, departure and arrival times, number of stops, class of service, flight durations, days left until departure, and ticket prices, this integrated dataset serves as a foundational component for understanding the complex dynamics of flight pricing. By leveraging this dataset and employing rigorous regression analysis techniques, the analysis aims to unravel the underlying factors driving fluctuations in ticket prices and uncover meaningful relationships between variables like airline preferences, flight routes, travel timings, and service classes.

1.3 Exploratory Data Analysis Our initial Exploratory Data Analysis (EDA) will focus on understanding the distribution of features like departure_time, source_city, destination_city, airline, and aircraft_type. We'll then explore how these features relate to the target variable, price (ticket price).

1.3.1 Evidence of Data

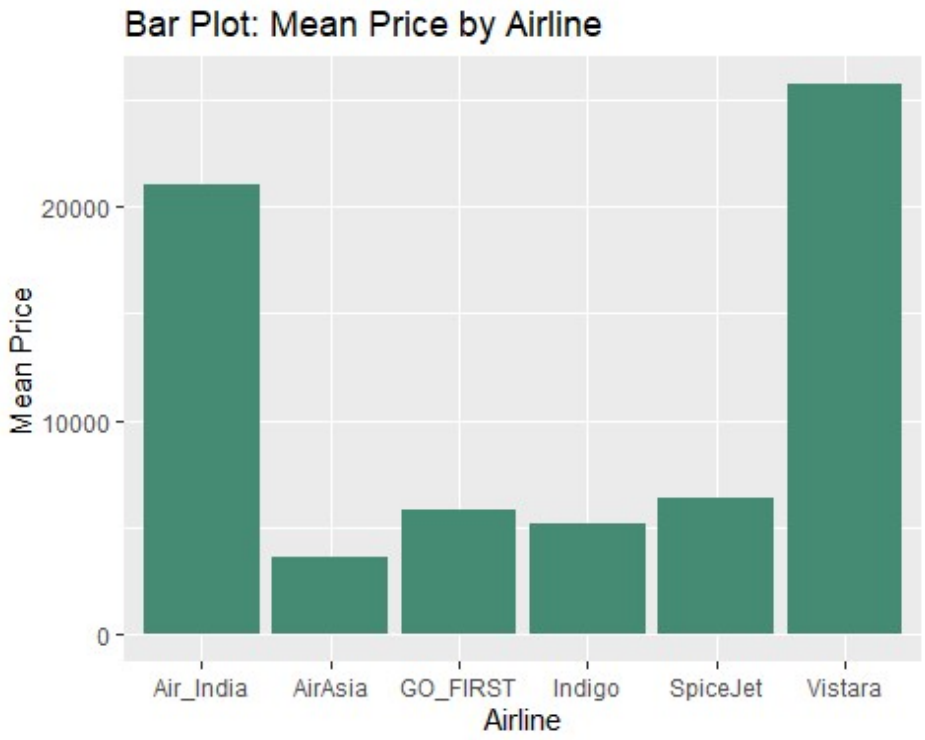
In the initial phase of data exploration, a preliminary examination of the first ten rows was conducted to ascertain the underlying structure of the dataset. The following excerpt presents these initial observations:

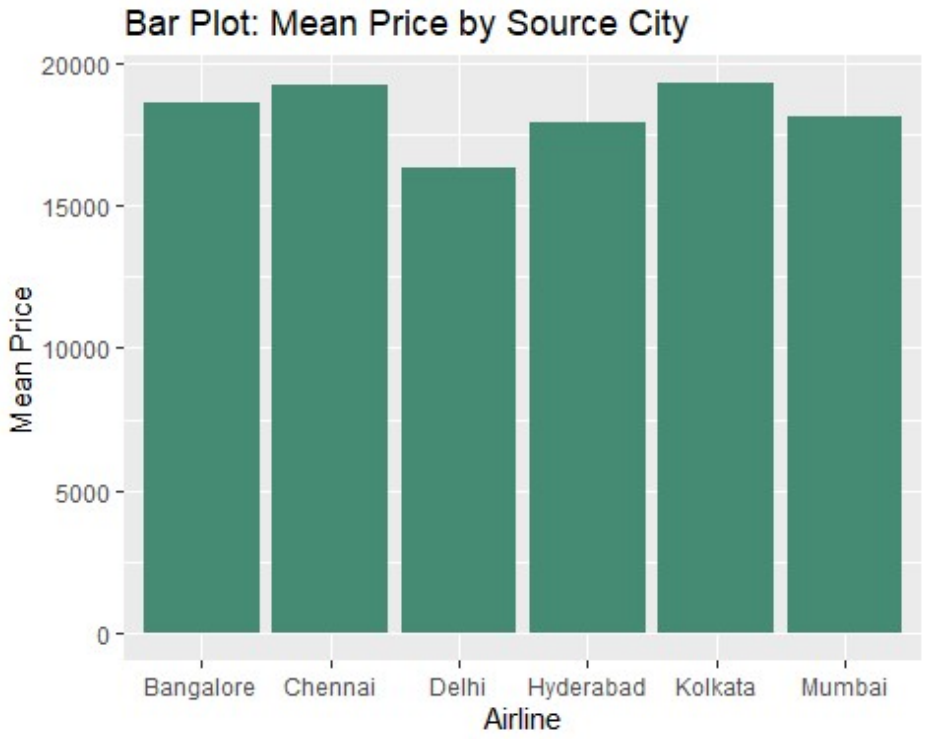
```
##     airline  flight source_city departure_time stops arrival_time
## 0 Air_India  AI-430     Chennai              1     1            3
## 1   Vistara  UK-996      Mumbai              3     1            3
## 2   Vistara  UK-776     Kolkata              3     0            4
## 3 Air_India AI-9720     Kolkata              0     2            1
## 4  GO_FIRST  G8-537     Kolkata              3     1            4
##   destination_city class duration days_left price
## 0           Mumbai     1    10.08        17 49553
## 1        Hyderabad     0    25.42         8 11129
## 2           Mumbai     0     2.75        47  4499
## 3          Chennai     0    25.17        15 13664
## 4           Mumbai     0     5.00         3 18259
```

1.3.2 Variable Relationships: Visualization and Analysis

A. Bar Chat for Key Relationships

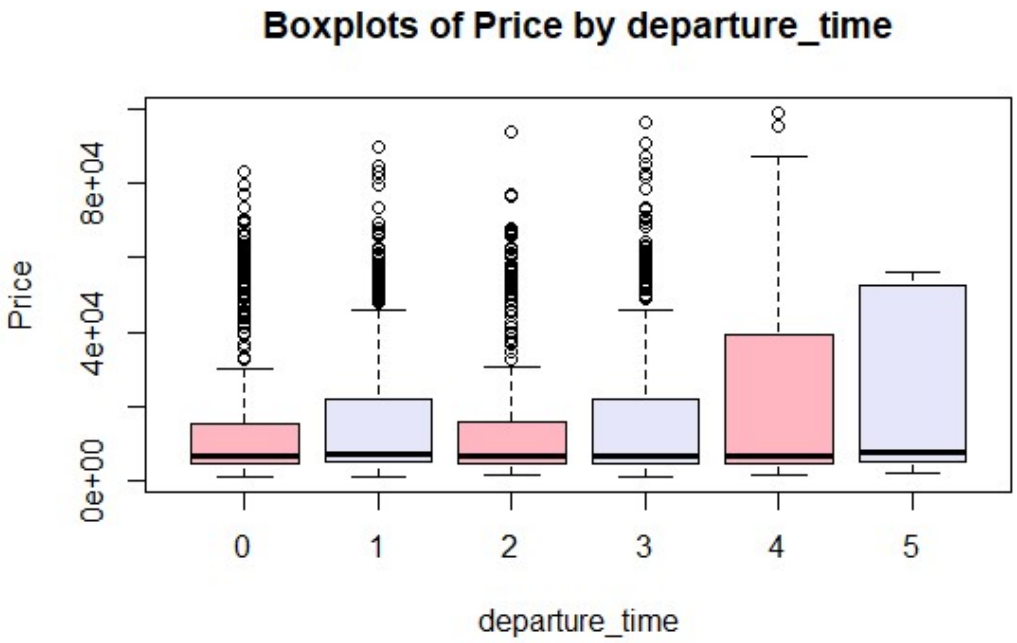
Analyzing the average ticket prices by airline reveals Vistara as the most expensive carrier, followed closely by Air India. This bar chart provides a snapshot of average ticket prices across different airlines. The airlines are listed on the bottom axis (X-axis) while the Y-axis represents the average ticket price. The height of each bar corresponds to an airline's average price. We can see that Vistara has the tallest bar, indicating the highest average ticket price among the airlines listed. Air India follows closely behind with the second-highest average price.

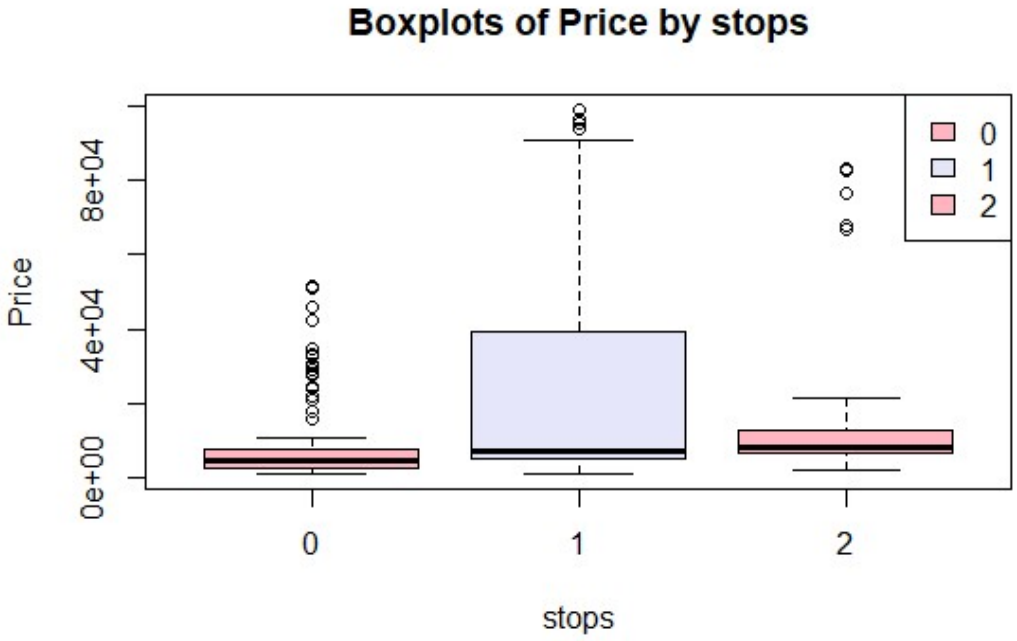


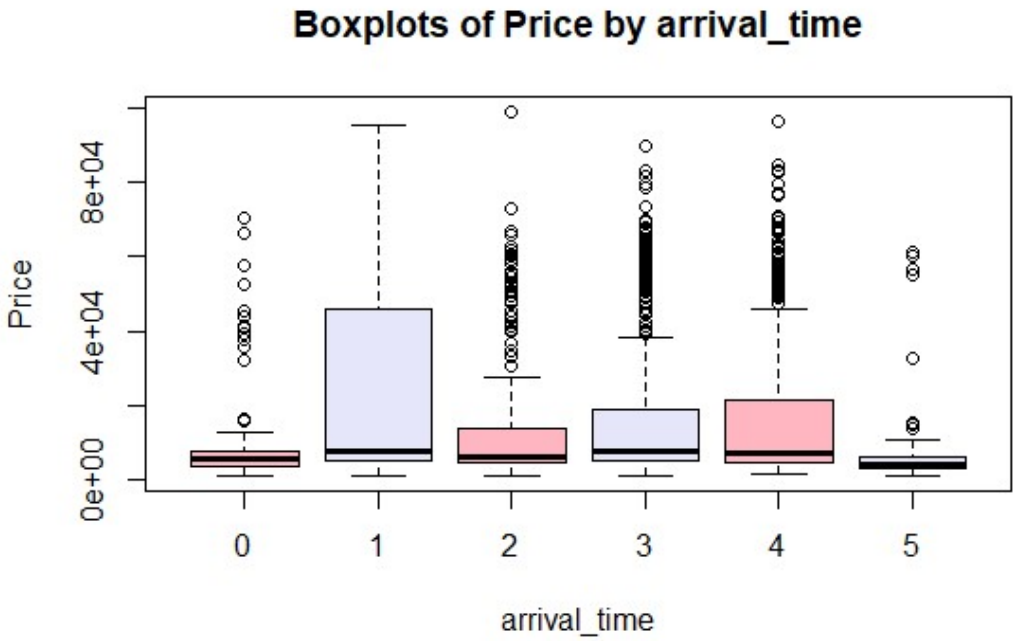


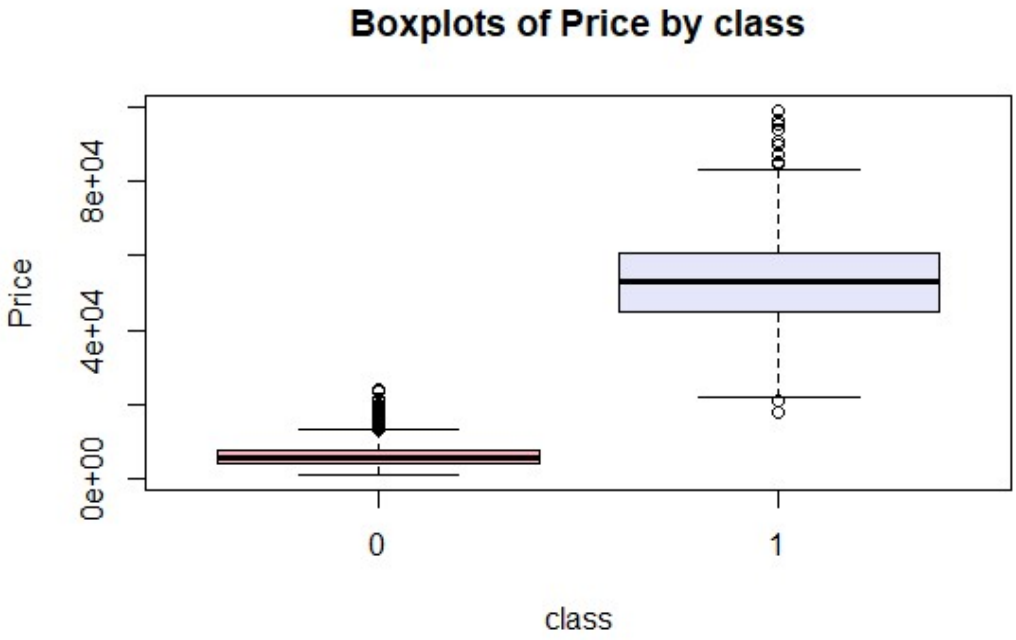
B. Boxplots for Comparative Analysis

The boxplot effectively visualizes price distributions between Business and Economy classes. Business class tends to have higher prices, as indicated by its higher positioning on the y-axis compared to Economy class. Similar boxplots for departure time, stops, and arrival time show that early morning and late night flights have lower prices, flights with one stop are priced higher, and morning flights tend to have higher prices.





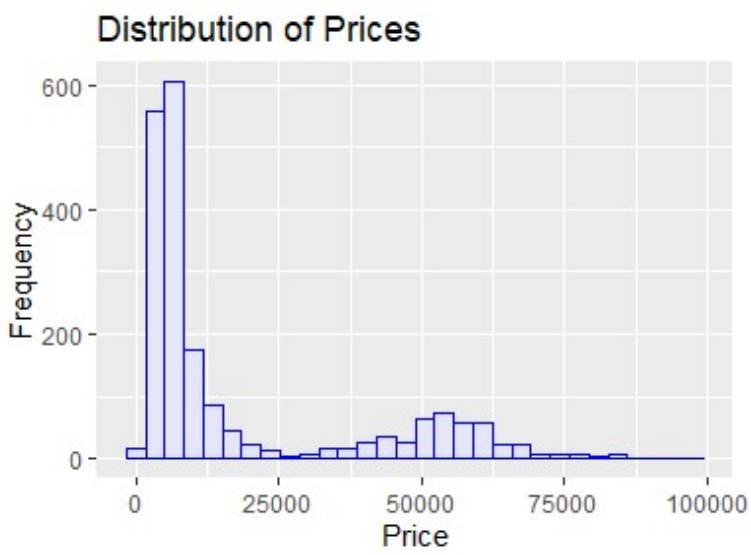




C. Histograms for Variable Insight

The histogram reveals a right-skewed distribution of flight prices.It provides a visual representation of the distribution of prices across the flights in our dataset. Examining the heights of the bars in the histogram reveals the frequency of flights within different price

ranges. In this particular histogram, the most common price range for flights appears to fall approximately within the Rs.5000 - Rs.20,000 range . Flights within this range are more prevalent, indicated by the taller bars, suggesting that a significant portion of the dataset consists of flights priced within this range. Additionally, there are flights priced both lower and higher than this range, but these are comparatively less frequent, as indicated by the shorter bars extending beyond the Rs.5000 - Rs.20,000 range. This visualization provides a clear overview of the distribution of flight prices and highlights the prevalence of flights within specific price ranges.



2. REGRESSION ANALYSIS

2.1 Data Preprocessing Handling Missing Values, Removing Duplicates, and Enhancing Predictors

Handling Missing Values: Following a comprehensive evaluation of the dataset, an assessment was conducted to identify missing values across all columns. The computation involved summing the missing values for each column. The analysis revealed an absence of missing values in any column, indicating a complete dataset suitable for further examination and analysis.

Handling Categorial Values: :In the context of descriptive statistics, non-numeric columns were isolated to identify categorical variables. As part of data preprocessing, manual encoding was conducted on the 'arrival_time' and 'departure_time' columns to transform categorical values into their corresponding numerical counterparts. These numeric representations delineate various time intervals: 0 for early morning, 1 for morning, 2 for afternoon, 3 for evening, 4 for night, and 5 for late night, elucidating details about arrival times within each interval. Concerning the 'Class' column, it serves as a numeric indicator of seat class, with distinct values of 0 denoting Economy and 1 denoting Business, facilitating differentiation between the two classes within the dataset.

The initial steps initiated have laid the foundation for a comprehensive examination of our dataset, enabling us to grasp its organization, detect any missing values, and categorical variables. In the ensuing sections, we will delve further into these aspects, aiming to gain a holistic understanding of our data and inform our analytical strategies. By preprocessing the dataset, we have enhanced its usability by converting categorical variables into numerical representations. This refined dataset now equips us adequately to undertake an in-depth exploration of descriptive statistics and inter-variable relationships.

2.2 Variable Selection

Feature selection is a crucial step in the process of building predictive models, helping to identify the most relevant variables that contribute to the model's performance. We are performing an exhaustive search for the best subset of features for explaining the Flight Price (price) with respect to other variables in a given dataset.

Adjusted R-squared: We then further analyzing the results by focusing on the model with the highest Adjusted R-squared. Adjusted R-squared as a criterion for selecting the best model among the subsets generated by the exhaustive search. The model with 3 predictors has the highest Adjusted R-squared value (0.8900948). The selected predictors for this model include intercept, price, stops, airline, days_left, class, arrival_time, departure_time, and duration. This model provides a good balance between goodness-of-fit and model simplicity, considering the Adjusted R-squared as a criterion for evaluation.

Best Model Selected from Adjusted R2: "price," "class," "stops," and "days_left,".

AIC: We are calculating now the Akaike Information Criterion (AIC) for each model generated by the exhaustive search and then fitting a linear model (lm) using the predictors selected(stops +arrival_time + class + departure_time + duration + days_left)) based on the model with the minimum AIC. **Best Model Selected from AIC:** class + stops + days_left + duration

BIC: We are calculating the Bayesian Information Criterion (BIC) for each model generated during the exhaustive search. The BIC is a criterion for model selection that penalizes model complexity. **Best Model Selected from BIC:** price,"stops".

Summary Table:

criterion	scores	variables_chosen
Adjusted R^2	0.8900948	stops, days_left, class
AIC	35339.25	class, stops, days_left, duration
BIC	39880.14	stops

Based on the summary table provided, a balanced approach for model selection might involve choosing the model with the highest Adjusted R-squared value while also considering competitive AIC and BIC values. In this case, the model with variables "stops," "days_left," and "class" achieves an Adjusted R-squared value of 0.8900948, indicating good goodness of fit. Additionally, the AIC and BIC values for this model are competitive, further supporting its selection.

Therefore, the selected model includes the variables “stops,” “days_left,” and “class.” This model strikes a balance between goodness of fit and simplicity. To operationalize this model, a dataset containing only the selected variables and the target variable was extracted from the original dataset. This updated dataset, containing the essential variables for analysis, was saved for future use.

2.3 Model Diagnostics

2.3.1 Check for collinearity Here we will check the collinearity between the predictors (class, stops, days_left) and response variable.

```
## Warning: package 'corrplot' was built under R version 4.3.3
```

Below are the Eigen value and condition indices for the fitted model after variable selection.

```
##   Eigenvalue Condition Index
## 1     3.0583          1.0000
## 2     0.6894          2.1062
## 3     0.1887          4.0257
## 4     0.0635          6.9385
```

The heatmap depicting the correlation matrix between the response variable price and predictors, namely ‘class’, ‘stops’, and ‘days_left’, reveals interesting insights. Among these predictors, ‘class’ and ‘price’ exhibit a relatively high correlation value of 0.94, indicating a strong linear relationship between them. Conversely, the correlation between ‘class’ and ‘stops’ is notably lower, suggesting less association between these variables.

This observation is beneficial for the regression model, as it indicates that 'class' and 'price' provide unique information and are not strongly redundant. The relatively low correlation between 'class' and other predictors further enhances the model's interpretability, as it reduces the risk of multicollinearity. Overall, this correlation analysis aids in understanding the interplay between predictors and the response variable, thereby contributing to the robustness of the regression model.

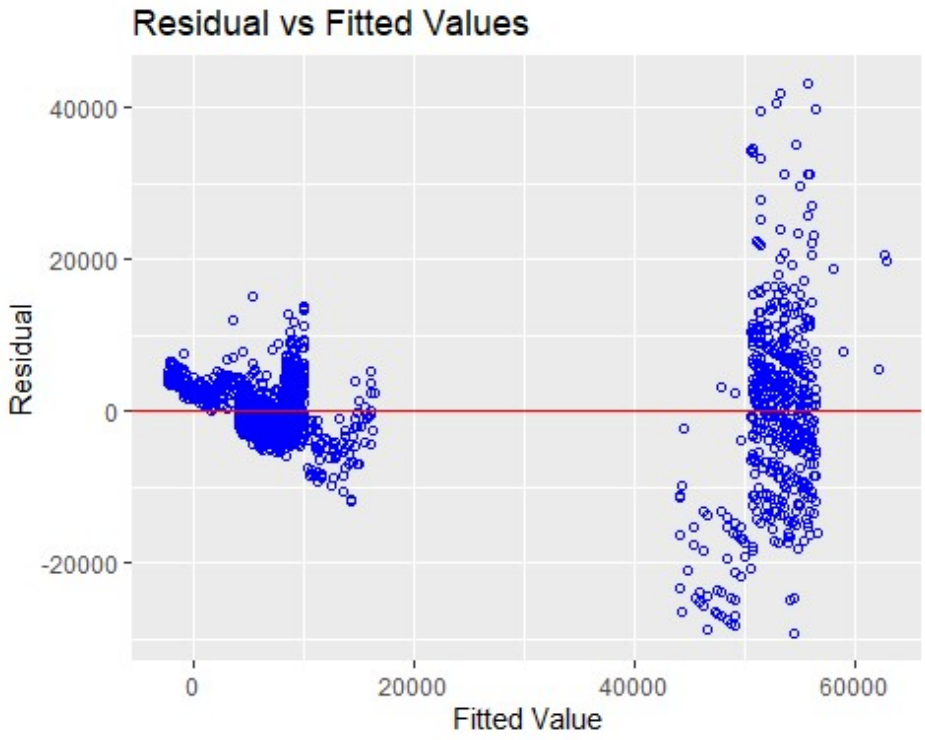
2.3.2 Diagnostics with vif values The 'flight_data' dataset was constructed. The variance inflation factors (VIFs) for key predictors, including 'stops' (VIF: 1.00958), 'days_left' (VIF: 1.000326), and 'class' (VIF: 1.000899), were all found to be close to 1. These low VIF values indicate that there is no substantial multicollinearity among these predictors, contributing to the reliability of the regression model's coefficients. Additionaly, the condition indices are all below 30 which menas we don't have to worry about multicollinearlity.

```
##     class     stops days_left
##  1.000899  1.000958  1.000326
```

Below is the summary for the model fitted after variable selection.

```
##
## Call:
## lm(formula = price ~ class + stops + days_left, data = dataset)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -29346  -2488   -383   2571  43189
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3933.59     495.88   7.933 3.54e-15 ***
## class       46349.90     355.32 130.447  < 2e-16 ***
## stops        6314.78     388.92  16.237  < 2e-16 ***
## days_left    -125.10      11.13 -11.239  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6878 on 1996 degrees of freedom
## Multiple R-squared:  0.8967, Adjusted R-squared:  0.8966
## F-statistic:  5778 on 3 and 1996 DF,  p-value: < 2.2e-16
```

2.3.3 Diagnostics with Fitted Vs residual plot and bp test: Now, we fit the plot for residuals of our final model which only has the three predictors class, days left and stops.




```
##
##  studentized Breusch-Pagan test
##
## data:  final_model
## BP = 398.02, df = 3, p-value < 2.2e-16
```

The fitted-vs-residuals plot does not look good. In particular, the variance tends to increase as the fitted values increase, but there is a gap between the fitted values as there are no points between 20000 to 40000. Additionally, the plot does not have linear relationship. Hence, the linearity and constant variance assumption has been violated.

2.3.4 Diagnostics with Fitted Vs residual plot and bp test: Now, we perform Shapiro wilk test for the model and check normality assumption violation.

```
##
##  Shapiro-Wilk normality test
##
## data:  resid(final_model)
## W = 0.85138, p-value < 2.2e-16
```

The p-value of the Shapiro-Wilk test is < 2.2e-16. So we reject the null hypothesis and conclude that the errors are not normally distributed. So, the normality assumption has been violated.

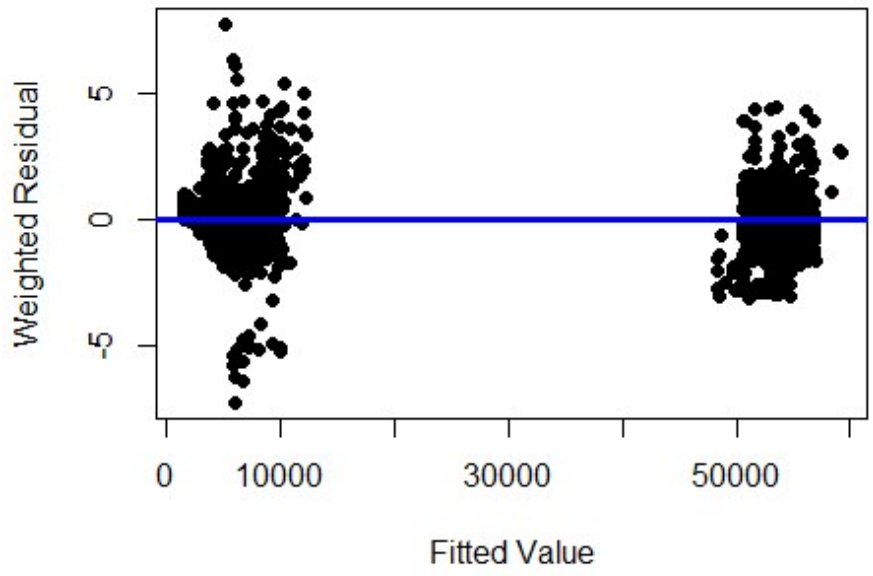
Therefore, the normality and equal variance assumptions are violated here, so we try to correct them using WLS (weighted least squares) model and performing shapiro wilk test. Also, we will be trying to check the normality distribution by removing high influential points.

2.4 Fixing Model Violations Next, we will assess the constant variance assumption for the Weighted Least Squares (WLS) model by using the inverse of the squared fitted values from the model as weights, denoted as weights=1/(fitted values)^2. We will examine the constant variance assumption both graphically and through a hypothesis test at the significance level of alpha=0.05.

Below is the summary for the Weighted least squares model (model_wls).

```
##
## Call:
## lm(formula = price ~ stops + days_left + class, data = dataset,
##     weights = weights)
##
## Weighted Residuals:
##     Min      1Q  Median      3Q     Max
## -7.2474 -0.7704 -0.1329  0.6765  7.7006
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  8031.902    220.433   36.44   <2e-16 ***
## stops        2187.650    135.114   16.19   <2e-16 ***
## days_left    -135.123      5.299  -25.50   <2e-16 ***
## class       46921.982    567.106   82.74   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.315 on 1996 degrees of freedom
## Multiple R-squared:  0.7935, Adjusted R-squared:  0.7931
## F-statistic:  2556 on 3 and 1996 DF,  p-value: < 2.2e-16
```

2.4.1 Checking violations using plotes and hypothesis tests The fitted Vs residual plot for the WLS model is plotted here.



In this case, the spread in the residuals appears to be roughly constant as the fitted values increase. We can also Conduct the BP test.

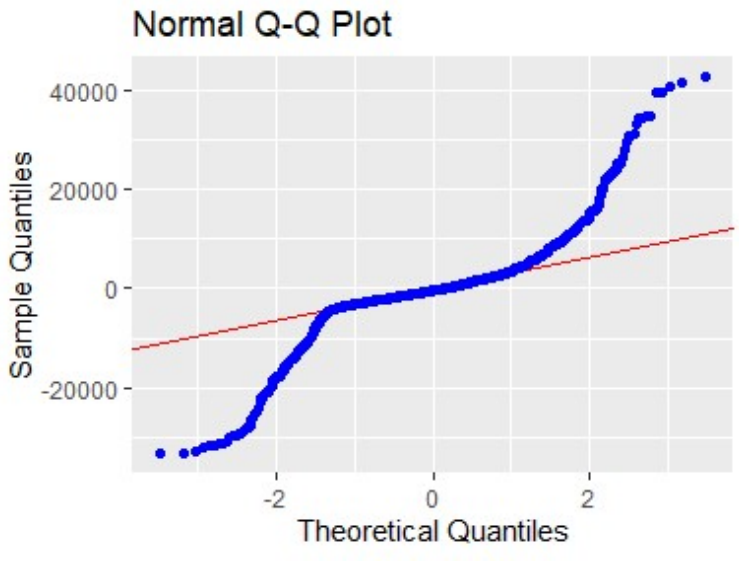
Studentized Breusch-Pagan Test: The Studentized Breusch-Pagan Test evaluates if the variance of residuals varies systematically across different predictor levels in a linear regression model. It helps detect heteroscedasticity, which can impact the reliability of regression analysis.

```
##
##  Shapiro-Wilk normality test
##
## data:  resid(model_wls)
## W = 0.8052, p-value < 2.2e-16

##
##  studentized Breusch-Pagan test
##
## data:  model_wls
## BP = 6.0565e-06, df = 3, p-value = 1
```

In this case, the p-value is very large (p-value = 1), so we do not reject the null hypothesis that the errors are homoscedastic. Hence, we conclude that the constant variance assumption has not been violated for the weighted residuals.

Now, we can check the normality distribution for the WLS model by plotting the QQ plot and histogram of residuals.

Q-Q Plot: The Q_Q plot shows the distribution of residuals for a weighted least squared regression model named model_wls. The plot does not look good and it suggests that the data is not normally distributed.



For further analysis, we'll identify and remove high leverage points, outliers, and influential observations from the data if required, to assess whether the violations of the LINE assumptions are mitigated.

2.5 Finding High Leverage points, Outliers, and Inflential Observations

- ***High Leverage Points:*** About 4.45% high leverage points, ranging from indices 3 to 1999, were detected using a threshold of 2 times the mean of the hat values. These points can substantially influence the estimated coefficients in the regression model, necessitating further investigation into their impact on model fit. Sensitivity checks can help evaluate the robustness of the regression results with and without these influential data points.

```
##    3   35  103  132  162  165  185  216  217  236  248  251  252  290  363
426
##    4   36  104  133  163  166  186  217  218  237  249  252  253  291  364
427
##  472  503  510  530  532  546  591  619  623  661  680  697  706  742  744
745
##  473  504  511  531  533  547  592  620  624  662  681  698  707  743  745
746
##  802  838  844  874  908  922 1002 1004 1014 1021 1032 1058 1059 1072 1075
1210
##  803  839  845  875  909  923 1003 1005 1015 1022 1033 1059 1060 1073 1076
1211
## 1218 1220 1238 1241 1271 1275 1295 1313 1318 1334 1335 1336 1364 1419 1448
1453
## 1219 1221 1239 1242 1272 1276 1296 1314 1319 1335 1336 1337 1365 1420 1449
1454
## 1489 1499 1509 1528 1532 1540 1552 1587 1611 1640 1692 1697 1714 1771 1798
```

```
1807
## 1490 1500 1510 1529 1533 1541 1553 1588 1612 1641 1693 1698 1715 1772 1799
1808
## 1809 1826 1837 1848 1861 1864 1894 1978 1998
## 1810 1827 1838 1849 1862 1865 1895 1979 1999
```

- ***Outliers:*** The residual analysis revealed eighteen observations, with indices 103, 802, 908, 1075, 1210, 1313, 1499, 1509, 1696, 1998, 104, 803, 909, 1076, 1211, 1314, 1500, 1510, 1697 and 1999, identified as outliers based on the test. Outliers, as determined by their studentized residuals exceeding the calculated threshold, can significantly impact the regression model's assumptions and overall fit.

```
##  103  802  908 1075 1210 1313 1499 1509 1696 1998
##  104  803  909 1076 1211 1314 1500 1510 1697 1999
```

- ***High Inflential Points:*** The examination using Cook's distance revealed that 5.55% of the entire dataset, corresponding to indices ranging from 3 to 1999, comprises influential points within the regression model. These specific data points wield a substantial influence on the outcomes of the regression, potentially exerting considerable effects on both the estimated coefficients and the overall model fit.

```
##    3   35   41   54  103  132  154  162  177  199  215  216  236  248  251
252
##    4   36   42   55  104  133  155  163  178  200  216  217  237  249  252
253
##  286  290  298  329  363  388  403  426  437  461  503  505  510  514  530
546
##  287  291  299  330  364  389  404  427  438  462  504  506  511  515  531
547
##  548  609  623  636  655  661  665  689  736  744  745  802  806  816  838
844
##  549  610  624  637  656  662  666  690  737  745  746  803  807  817  839
845
##  874  905  908  922  982 1002 1008 1021 1053 1058 1060 1075 1149 1177 1188
1190
##  875  906  909  923  983 1003 1009 1022 1054 1059 1061 1076 1150 1178 1189
1191
## 1210 1218 1238 1241 1249 1253 1295 1313 1318 1334 1336 1364 1365 1418 1453
1471
## 1211 1219 1239 1242 1250 1254 1296 1314 1319 1335 1337 1365 1366 1419 1454
1472
## 1482 1489 1497 1499 1509 1528 1552 1562 1583 1587 1611 1627 1630 1692 1696
1697
## 1483 1490 1498 1500 1510 1529 1553 1563 1584 1588 1612 1628 1631 1693 1697
1698
## 1751 1752 1771 1799 1826 1827 1837 1848 1861 1864 1875 1949 1976 1988 1998
## 1752 1753 1772 1800 1827 1828 1838 1849 1862 1865 1876 1950 1977 1989 1999
```

2.5.1 Handling High Leverage Points, Outliers, and Influential Observations: The model, constructed with predictors selected through an exhaustive search on the dataset, underwent a thorough refinement process to address influential points, high-leverage

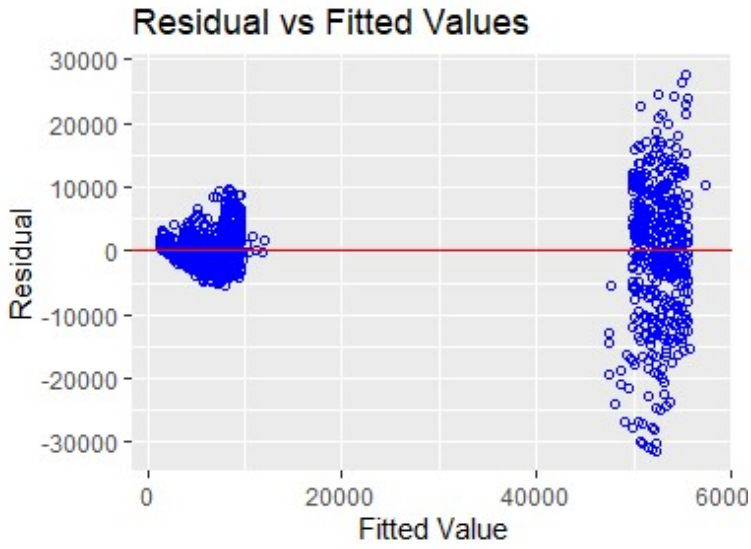
points, and outliers. This meticulous approach aimed to enhance the robustness of the final model (model_wls).

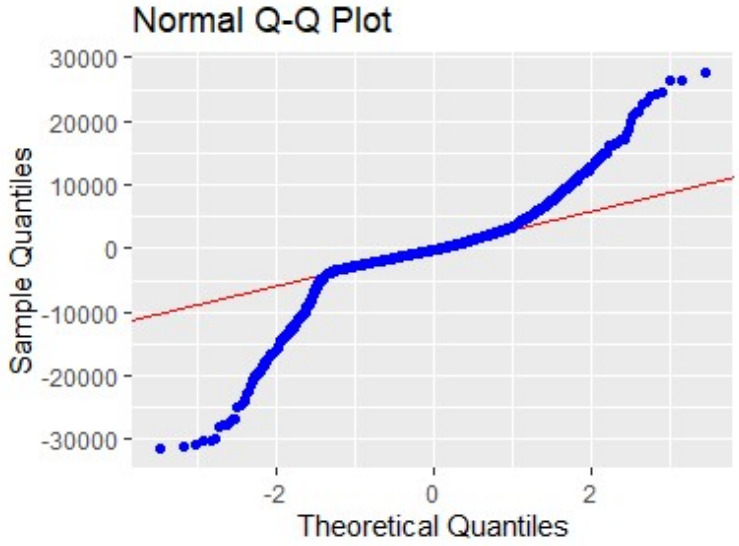
Upon reviewing the summary output of the model: The adjusted model notably improves with an Adjusted R-squared of 0.7047, reflecting a more accurate depiction of predictor-price relationships. Key shifts in coefficients, particularly the intercept and 'class' variable, indicate enhanced precision. The residual standard error of 14640 signifies residual variability around the regression line. A highly significant p-value (< 2.2e-16) underscores the statistical significance of observed relationships, though further refinement is needed for deeper insights into price fluctuations.

```
##
## Call:
## lm(formula = price ~ stops + days_left + class, data = dataset,
##     subset = noninfluential_ids, weights = weights)
##
## Weighted Residuals:
##     Min      1Q  Median      3Q     Max
## -2.8246 -0.6876 -0.0834  0.6472  3.6086
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  7369.76     212.49   34.68   <2e-16 ***
## stops        2429.87     148.52   16.36   <2e-16 ***
## days_left    -124.29       4.63  -26.85   <2e-16 ***
## class       46066.44     442.48  104.11   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9943 on 1885 degrees of freedom
## Multiple R-squared:  0.8633, Adjusted R-squared:  0.8631
## F-statistic:  3967 on 3 and 1885 DF,  p-value: < 2.2e-16
```

2.6 Assessment of Violations in WLS Model after Removing High Influential Points:

In this section, we focus on the model_fix, where we applied Weighted Least Squares (WLS) and removed highly influential points. We will reevaluate potential violations in this model through plots and hypothesis tests.






```
##
##  studentized Breusch-Pagan test
##
## data:  model_fix
## BP = 6.4553e-06, df = 3, p-value = 1

##
##  Shapiro-Wilk normality test
##
## data:  resid(model_fix)
## W = 0.84393, p-value < 2.2e-16
```

Even though,after removing high influential points. there is no much difference in the qq plot compared with the previous qq plot. hence normality assumption is violated

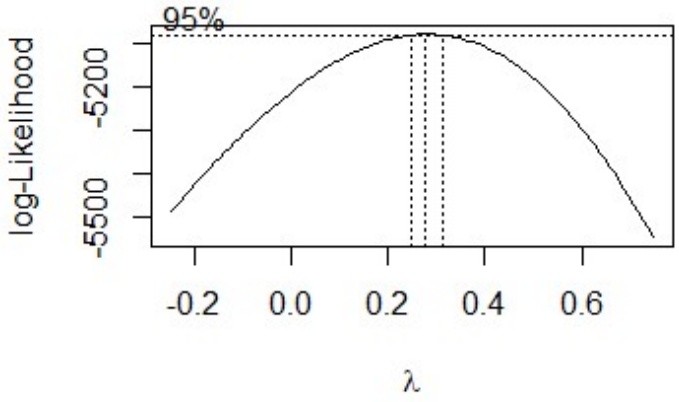
The graph still doesn't look good. Even though there is not a lot of data for large fitted values, it still seems clear that the constant variance assumption is not violated. In addition, the residuals for small fitted values suggest that the regression relationship may not be linear.

Overall, we conclude that the errors are not normally distributed and the constant variance is not violated and linearity assumptions are violated.

2.7 Transformations:

Here, we use the Box-Cox method to determine an appropriate transformation of the response. Which would return the value of λ^ as well as the plot returned by the boxcox function. To search for an appropriate transformation of the response, we use the Box-Cox method. We identify an appropriate value of λ using the boxcox function from the MASS package. The default range of λ ranging from -2 to 2 makes it hard to visualize the CI for λ. Specifying a custom range of -0.25, to 0.75 makes the plot much more legible. To search for an appropriate transformation of the response, we use the Box-Cox method. We identify an appropriate value of λ using the boxcox function.

```
##
## Attaching package: 'dplyr'

## The following object is masked from 'package:car':
##
##     recode

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

## Warning: package 'caret' was built under R version 4.3.3

## Loading required package: lattice

##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##     select

## The following object is masked from 'package:olsrr':
##
##     cement
```



Now, we extract the λ value that maximizes the log-likelihood.

```
## The best λ value is : 0.2752525
```

According to the boxcox plot from above, the 95% confidence interval for λ contains λ = 0.275. Therefore, we are justified in using the transformation $price^{0.275}$.

```
## lower bound upper bound
##   0.2550505   0.3055556
```

In this case, the 95% CI is (0.2550505, 0.3055556), which clearly contains λ = 0.275

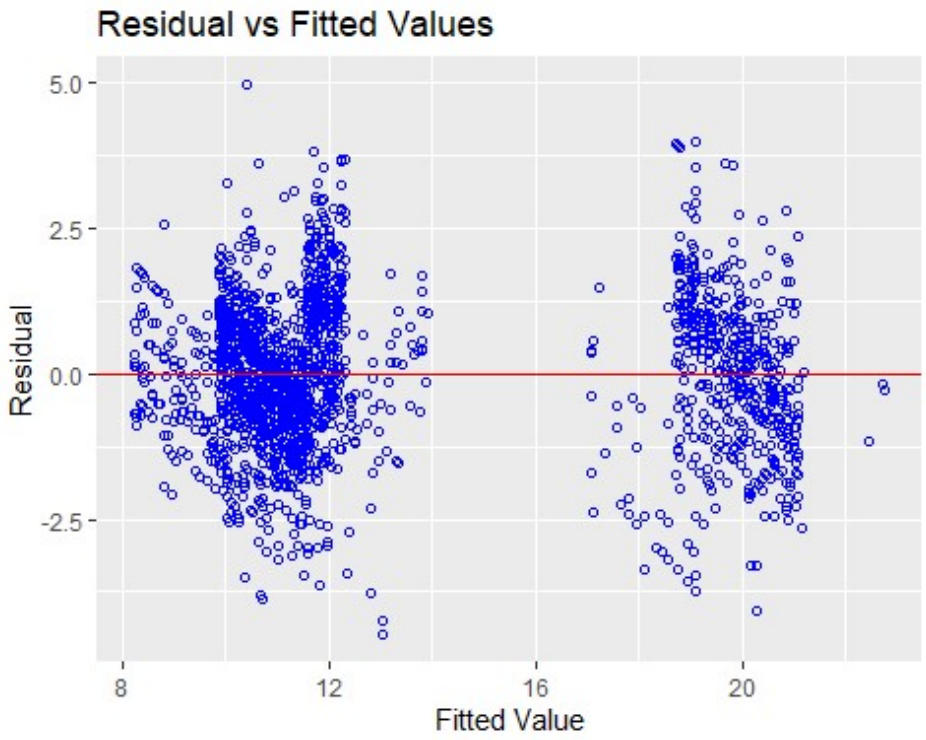
2.7.1 Perform regression analysis and fit a transformed model: Perform OLS regression with $price^{0.275}$ as the response and class, stops, days_left as the predictors.The summary fo the new transformed model (model_bc) is given below.

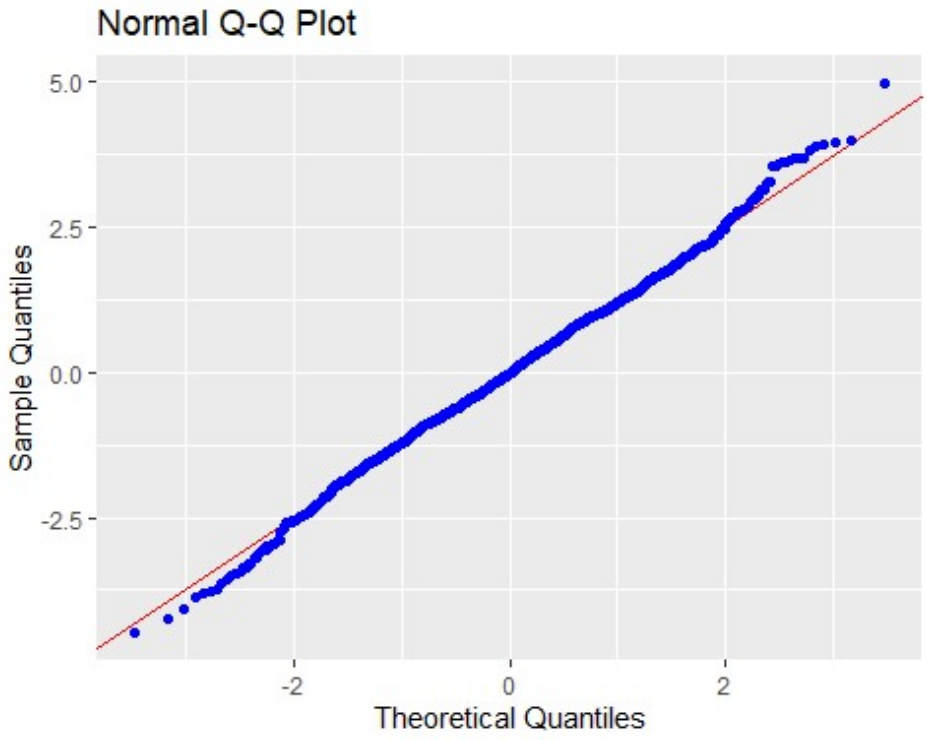
```
##
## Call:
## lm(formula = price^0.275 ~ ., data = dataset)
##
## Coefficients:
## (Intercept)        stops    days_left        class
##    10.73090      1.62138     -0.05083      8.84412

##
## Call:
## lm(formula = price^0.275 ~ ., data = dataset)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.4790 -0.8022 -0.0237  0.8711  4.9664
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept) 10.730897   0.090325  118.80   <2e-16 ***
## stops        1.621380   0.070841   22.89   <2e-16 ***
## days_left   -0.050831   0.002028  -25.07   <2e-16 ***
## class        8.844116   0.064721  136.65   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.253 on 1996 degrees of freedom
## Multiple R-squared:  0.9081, Adjusted R-squared:  0.908
## F-statistic:  6578 on 3 and 1996 DF,  p-value: < 2.2e-16
```

2.7.2 Assessment of Final Model (model_bc) Residuals:

For the final model_bc derived after transformation, we examine the Fitted vs. Residual plot and QQ plot. Subsequently, we conduct the Shapiro-Wilk test to assess violations of equal variance and normality assumptions.





Shapiro-Wilk Normality Test: The flight model's residuals underwent a Shapiro-Wilk test to evaluate their normal distribution assumption. The test, which compares the residuals against the null hypothesis of normality, yielded a Shapiro-Wilk statistic (W) of 0.99755 and a p-value is 0.003311. A low p-value indicates normality violation.

```
##
##  Shapiro-Wilk normality test
##
## data:  resid(model_bc)
## W = 0.99755, p-value = 0.003311
```

Although the p-value of the Shapiro-Wilk test indicates non-normality, the QQ plot for the model displays a satisfactory fit. Thus, despite the lower p-value, the normality assumption appears to be better supported for this model compared to previous ones in this analysis.

```
##
##  studentized Breusch-Pagan test
##
## data:  model_bc
## BP = 24.894, df = 3, p-value = 1.625e-05
```

3. DISCUSSIONS:

3.1 Discussion on Model Diagnosis and Transformation: We began by employing variable selection techniques, including AIC, BIC, and Adjusted R-squared, using the forward method. However, we encountered violations of linearity, constant variance, and normality assumptions, despite no significant multicollinearity issues detected through VIF analysis.

To address these concerns, we applied a Weighted Least Squares (WLS) model and conducted diagnostic tests, such as fitted vs. residual and QQ plots. Initially, the Breusch-Pagan test suggested homoscedasticity, but we found persistent deviations from normality even after addressing outliers, high leverage points, and influential observations.

Implementing the Box-Cox transformation noticeably improved normality, enhancing the QQ plot. However, the Shapiro-Wilk test still indicated a departure from normality. Surprisingly, the Breusch-Pagan test revealed a violation of the constant variance assumption, despite improved alignment in the fitted vs. residual plot, potentially indicating a better model fit.

3.2 Summary of Model Evaluation and Hypothesis Testing:

Model	R2	P value of BP test	P value of Shapirowilk test
Adjusted R square model (final_model)	0.8967	< 2.2e-16	< 2.2e-16
Weighted Least squares model (model_wls)	0.7935	1	< 2.2e-16
WLS Model after removing highly influential points (model_fix)	0.8633	1	< 2.2e-16
The model after transformation (model_bc)	0.9081	1.625e-05	0.003311

After conducting a series of analyses to address model robustness, we focused on improving LINE assumptions. Ultimately, the model_bc, derived after Box-Cox transformation, emerged as the most enhanced. Despite a slight decrease in the p-value of

the Breusch-Pagan test, this model exhibited the highest R-squared value (0.9081) and notably improved p-values from the Shapiro-Wilk test (0.003311). Visually, both the fitted vs. residual and QQ plots for the model_bc appeared favorable, affirming its overall superiority.

We performed hypothesis tests on the final model (model_bc) by examining test statistics and p-values for each predictor's effect on the response variable. The results and conclusions from the t-tests for each predictor are outlined below.

3.3 Hypothesis Testing Results for Predictors in the Final Model (model_bc)

- ***stops:*** Null hypothesis (H0): The coefficient for stops is zero. Alternative hypothesis (H1): The coefficient for stops is not zero. The p-value for stops is < 2e-16, indicating that stops is highly statistically significant.
- ***days_left:*** Null hypothesis (H0): The coefficient for days_left is zero. Alternative hypothesis (H1): The coefficient for days_left is not zero. The p-value for days_left is < 2e-16, indicating that days_left is highly statistically significant.
- ***class:*** Null hypothesis (H0): The coefficient for class is zero. Alternative hypothesis (H1): The coefficient for class is not zero. The p-value for class is < 2e-16, indicating that class is highly statistically significant.

3.3.1 Hypothesis Testing: Relationship between Response Variable (Price) and each Predictor seperately(Stops, Class, Days_Left)

Now, we can perform the hypothesis tests to check the significant relationship between response variable (price) and rest of the predictors (stops, class, days_left)

Based on model 1 (simple linear regression model between price and stops), the interpretation is : The Null Hypothesis: The $H0{:}\beta 1 = 0$ suggests that there is no significant relationship between the number of stops and the price. The Alternate Hypothesis: The $H1{:}\beta 1 \neq 0$ indicates that there is a significant relationship between the number of stops and the price.

The test statistic is calculated as t = βˆ1/SE[βˆ1], which equals 4.025 in this case. The p-value of the test is 5.92e-05. At a significance level of α = 0.05, the p-value for the coefficient of "stops" is less, leading to rejection of the null hypothesis. This signifies a significant linear relationship between the number of stops and the price. Thus, we conclude that the number of stops has a statistically significant effect on the price, satisfying all linear regression assumptions.

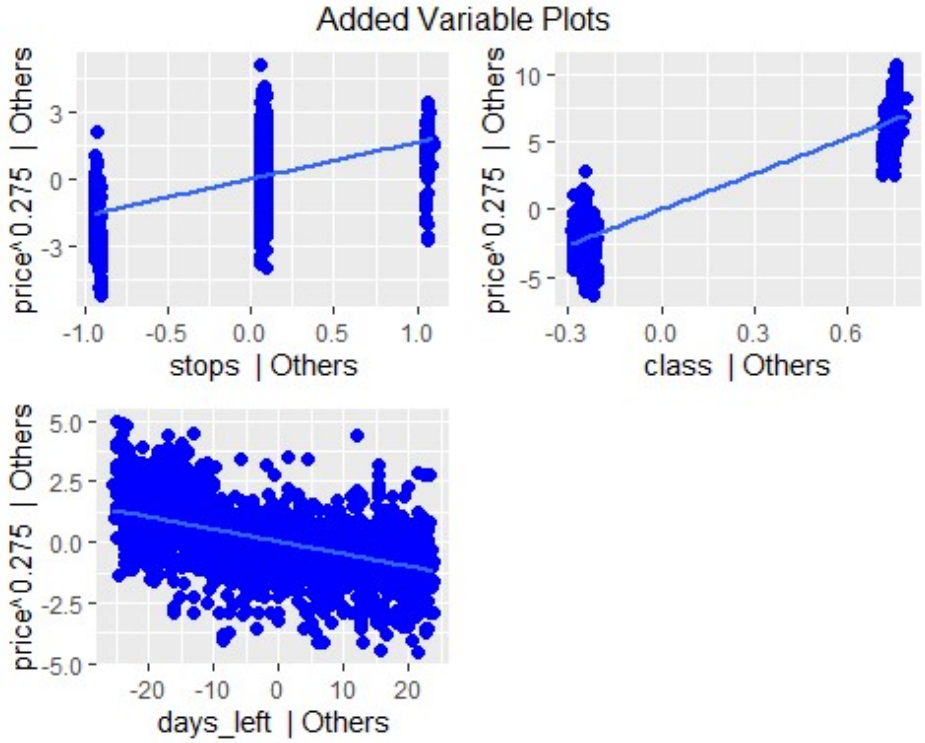
Based on model 2 (simple linear regression model between price and days_left), the interpretation is : For the simple linear regression model between price and days_left, the test statistic is -4.051 with a p-value of 5.3e-05. At a significance level of α = 0.05, the p-value is less, leading to rejection of the null hypothesis, indicating a significant linear relationship between days_left and price. Thus, days_left has a statistically significant effect on price, satisfying all linear regression assumptions.

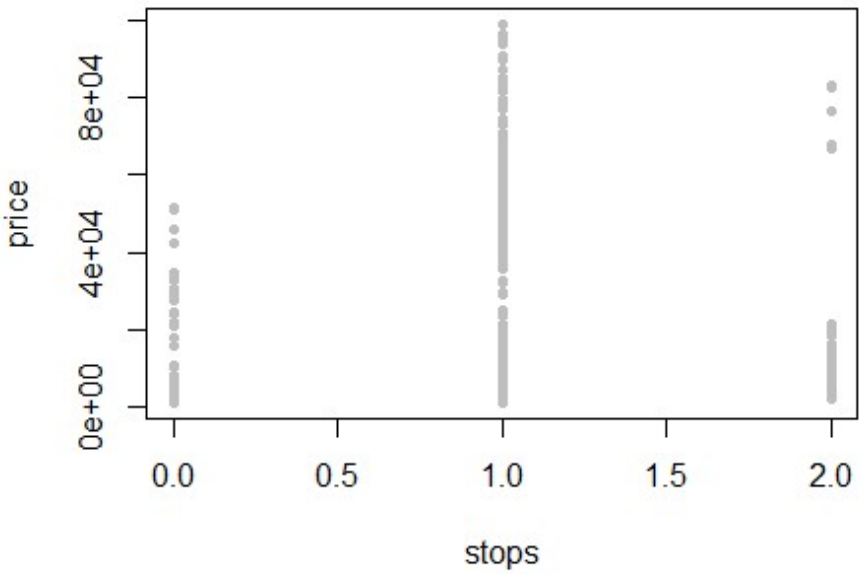
Based on model 3 (simple linear regression model between price and class), the interpretation is :

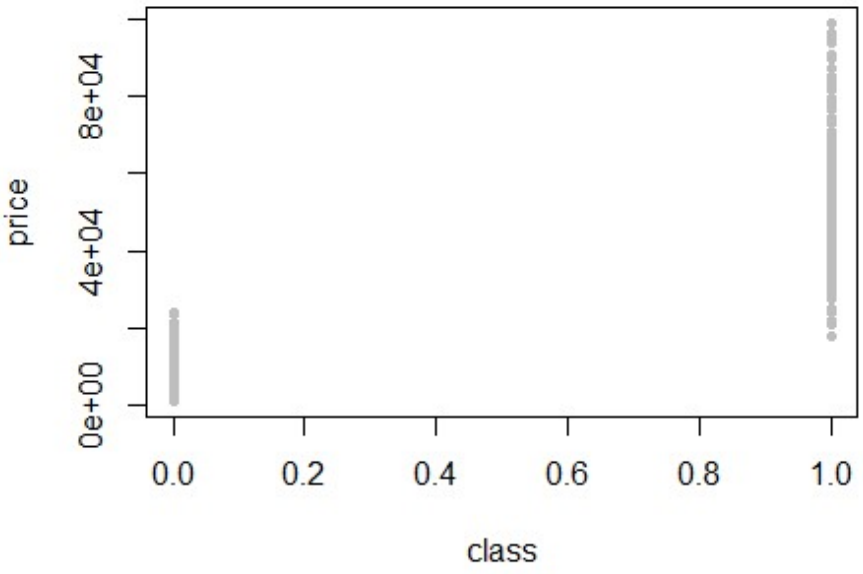
The Null Hypothesis: The $H0{:}\beta 1 = 0$ suggests that there is no significant relationship between the class and the price. The Alternate Hypothesis: The $H1{:}\beta 1 \neq 0$ indicates that there is a significant relationship between the class and the price. The test statistic is calculated as t = βˆ1/SE[βˆ1], which equals 119.26 in this case.The p-value of the test is < 2.2e-16. At α = 0.05, the p-value for the coefficient of "class" is significantly lower, leading to the rejection of the null hypothesis. This implies a significant linear relationship between the class and the price. Therefore, in the context of the problem, we conclude that the class indeed has a statistically significant effect on the price, satisfying all linear regression assumptions. **3.4 Analysis of Flight Characteristics and Ticket Prices:**

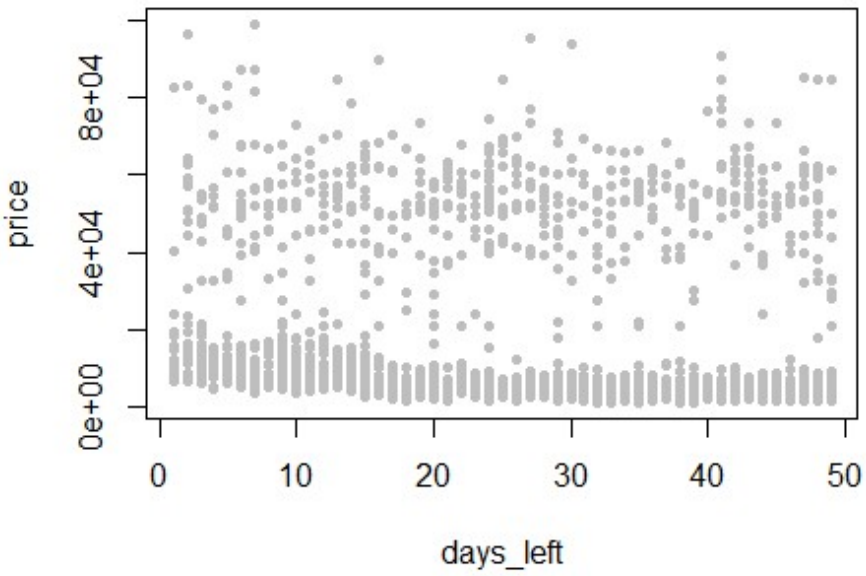
We examined added variable plots for each predictor in the final model (model_bc), including stops, class, and days_left, to analyze their impact on ticket prices.

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




price
0e+00
4e+04
8e+04
0.0
0.5
1.0
1.5
2.0
stops


price
0e+00
4e+04
8e+04
0.0
0.2
0.4
0.6
0.8
1.0
class



After meticulous data preparation and model refinement, we conducted hypothesis tests and examined added variable plots for each predictor in the final model.

Our analysis revealed significant relationships between flight characteristics and ticket prices. Specifically, we observed a linear increase in flight prices with an increase in class and stops, indicating higher fares for premium services and longer routes. Conversely, the number of days left before the flight displayed a negative slope, suggesting that prices tend to decrease as the departure date approaches.

These findings provide valuable insights into the pricing dynamics of airline tickets and underscore the importance of considering various factors in pricing strategies. By understanding these relationships, airlines and travel agencies can make informed decisions to optimize pricing structures and enhance customer satisfaction.

4. Limitations

- ***Categorical Columns Not Included:*** The omission of categorical columns in the model represents a significant limitation, affecting the model's accuracy and interpretability.
- ***Assumption Limitations:*** Despite efforts to address linearity and constant variance, residual normality concerns persist, indicating potential unaccounted factors or data complexities.
- ***Model Fit Considerations:*** The Box-Cox transformation improved normality but introduced constant variance issues, underscoring the challenge of balancing multiple assumptions. Further assessment, such as cross-validation, is needed for comprehensive model evaluation.

5. Conclusion

In this comprehensive analysis of flight ticket pricing dynamics, we aimed to uncover the underlying factors influencing ticket prices and provide valuable insights for industry stakeholders. By employing regression analysis techniques and hypothesis testing, we identified significant relationships between various flight characteristics and ticket prices.

Our investigation revealed that factors such as service class, number of stops, and days left before the flight significantly impact ticket prices. Premium services and longer routes were associated with higher fares, while ticket prices tended to decrease as the departure date approached. These findings underscore the complex interplay between supply and demand dynamics in the aviation industry.

Throughout our analysis, we encountered and addressed several challenges, including violations of linearity, constant variance, and residual normality assumptions. Despite efforts to mitigate these issues through model refinement techniques such as Weighted Least Squares and Box-Cox transformation, residual normality concerns persisted. The omission of categorical columns in the model further limited its accuracy and interpretability.

In conclusion, our analysis provides valuable insights into the intricate pricing dynamics of airline tickets. While our model offers a significant improvement in understanding the relationships between flight characteristics and ticket prices, further research and refinement are needed to address the remaining limitations. By continuing to explore and refine pricing models, airlines, travel agencies, and policymakers can make informed decisions to optimize pricing strategies and enhance the overall efficiency and competitiveness of the aviation industry.

6. Additional work :

Furthermore, we conducted an additional assessment for high leverage points, outliers, and highly influential points on the final model obtained after the Box-Cox transformation. Upon identifying and removing the highly influential points, we refitted the model to evaluate if its robustness improved.

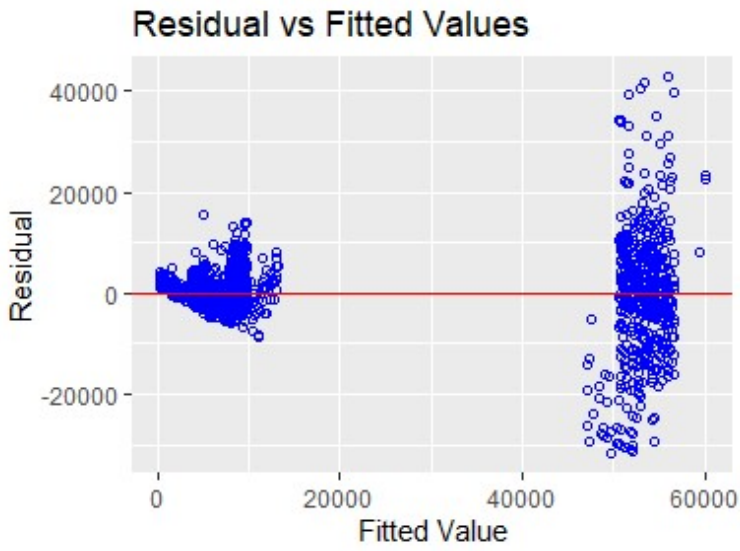
- Handling High Leverage Points, Outliers, and Influential Observations: Following the identification of high influential points, we excluded them from the dataset and refitted the model (model_fix_bc). Below is the summary of the updated model.

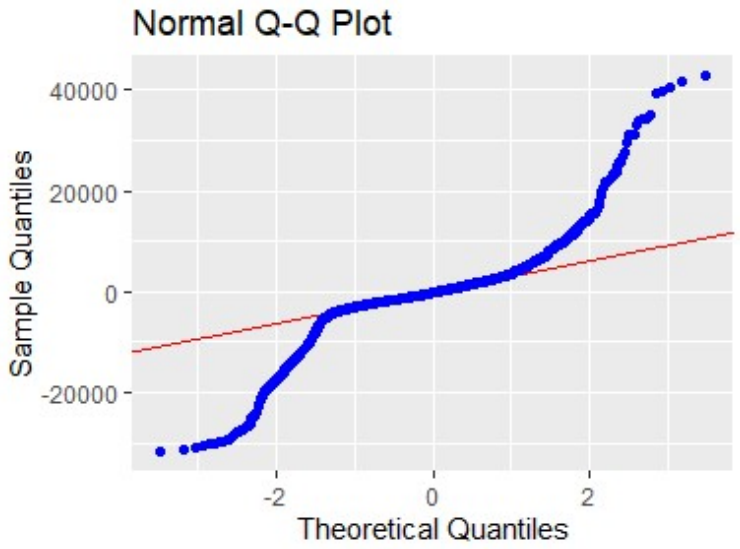
For the fitted model (model_fix_bc), we conducted Shapiro-Wilk tests, residual vs. fitted plots, and QQ plots. However, the results showed no significant improvement, indicating persistent issues with model diagnostics.

Based on the plot, it's evident that the model's performance hasn't significantly improved, with continued violations of the equal variance and normality assumptions. In comparison, the model after Box-Cox transformation without removing high influential points exhibited better performance. This indicates that the step of removing high influential points did not positively impact the model's robustness.

- Robust Regression: Furthermore, we explored a robust regression model using IRWLS with a limit of 100 iterations. However, examination of the QQ plots and fitted vs. residuals plot revealed that the model did not meet the necessary assumptions. Consequently, we opted not to utilize this regression model due to its inadequate performance in addressing the violations of variance and normality.

```
##
## Call: rlm(formula = price ~ stops + class + days_left, data = dataset,
##     maxit = 100)
## Residuals:
##      Min       1Q   Median       3Q      Max
## -31863.7  -2084.6   -176.3   2095.2  43019.7
##
## Coefficients:
##             Value      Std. Error t value
## (Intercept)  6433.5138   257.7302    24.9622
## stops        3481.0041   202.1356    17.2211
## class       46853.9792   184.6724   253.7141
## days_left    -125.1665     5.7853   -21.6352
##
## Residual standard error: 3102 on 1996 degrees of freedom

##
##  Shapiro-Wilk normality test
##
## data:  resid(model_hub)
## W = 0.81263, p-value < 2.2e-16
```





Code Appendix

```
#1.3.1 Evidence of Data
# Read the CSV file

library(readr)
flight <-
read.csv("C:/Users/divya/Downloads/flight_filtered_data_final_ff.csv",
row.names = 1)
first_10_columns <- head(flight, n = 5)
first_10_columns

#1.3.2 Variable Relationships: Visualization and Analysis
# Assuming the flight dataset is already loaded into your environment
# Load the ggplot2 library
library(ggplot2)
```

```r
ggplot(flight, aes(x = airline, y = price)) +
  stat_summary(fun = mean, geom = "bar", fill = "aquamarine4") +
  labs(x = "Airline", y = "Mean Price", title = "Bar Plot: Mean Price by
Airline")

#1.3.2 Variable Relationships: Visualization and Analysis
# Assuming the flight dataset is already loaded into your environment
# Load the ggplot2 library
library(ggplot2)

ggplot(flight, aes(x = source_city, y = price)) +
  stat_summary(fun = mean, geom = "bar", fill = "aquamarine4") +
  labs(x = "Airline", y = "Mean Price", title = "Bar Plot: Mean Price by
Source City")
#1.4.2 Variable Relationships: Visualization and Analysis
# Define the columns of interest
Data <-
read.csv("C:/Users/divya/Downloads/flight_filtered_data_final_ff.csv",
row.names = 1)
columns_of_interest <- c("departure_time", "stops", "arrival_time", "class")

# Loop through each column
for (col in columns_of_interest) {
  # Convert stops to factor to ensure distinct levels
  if (col == "stops") {
    Data$stops <- factor(Data$stops)
  }

  # Side-by-side boxplots for price by each column with non-default colors
  boxplot(price ~ Data[[col]], data = Data,
          col = c("lightpink", "lavender"), xlab = col,
          ylab = "Price", main = paste("Boxplots of Price by", col))

  # Add a legend if there are levels
  if (is.factor(Data[[col]]) && length(levels(Data[[col]])) > 0) {
    legend("topright", legend = levels(Data[[col]]), fill = c("lightpink",
"lavender"))
  }
}



#1.4.2 Variable Relationships: Visualization and Analysis
# Load the ggplot2 library
library(ggplot2)

# Create a histogram plot for the price variable with specified binwidth
price_histogram <- ggplot(flight, aes(x = price)) +
  geom_histogram(bins = 30, fill = "lavender",color="blue") +
```

```r
  labs(title = "Distribution of Prices",
       x = "Price",
       y = "Frequency")

# Print the histogram plot
print(price_histogram)
#2.1 Data Preprocessing Handling Missing Values, Removing Duplicates, and
Enhancing Predictors
# Check for missing values
any(is.na(flight))
# We have already submitted the dataset after handling the categorical
values. This is just for reference.

# Load necessary libraries
#library(dplyr)

# Assuming 'arrival_time' and 'departure_time' are categorical columns in
your dataset

# Manual encoding for 'arrival_time' column
#flight$arrival_time <- factor(flight$arrival_time, levels = c("early
morning", "morning", "afternoon", "evening", "night", "late night"))
#flight$arrival_time <- as.integer(as.character(flight$arrival_time))

# Manual encoding for 'departure_time' column
#flight$departure_time <- factor(flight$departure_time, levels = c("early
morning", "morning", "afternoon", "evening", "night", "late night"))
#flight$departure_time <- as.integer(as.character(flight$departure_time))

# Manual encoding for 'Class' column
#flight$Class <- as.integer(flight$Class)

#2.2 Variable Selection
# Load the flight data
flight_data <-
read.csv("C:/Users/divya/Downloads/flight_filtered_data_final_ff.csv",
row.names = 1)

##AIC forward
mod_start1 <- lm(price ~ 1, data = flight_data)
mod_forwd_aic1 <- step(mod_start1, scope = price ~ stops +arrival_time +
class + departure_time + duration + days_left, direction = 'forward')
coef(mod_forwd_aic1)

##BIC forward
n1 <- nrow(flight_data)
mod_forwd_bic1 <- step(mod_start1, scope = price ~ stops, direction =
'forward', k = log(n1))
coef(mod_forwd_bic1)
```

```r
# The best Forward AIC model value and its coefficients
extractAIC(mod_forwd_aic1)
coef(mod_forwd_aic1)

# The best Forward BIC model and its coefficients
extractAIC(mod_forwd_bic1, k = log(n1))
coef(mod_forwd_bic1)

# Create a table with each quality criterion
criterion <- c("AIC", "BIC")
scores <- c(AIC = extractAIC(mod_forwd_aic1)[2], BIC =
extractAIC(mod_forwd_bic1, k = log(n1))[2])
variables_chosen <- c(AIC = paste(names(coef(mod_forwd_aic1))[-1], collapse =
", "),
                      BIC = paste(names(coef(mod_forwd_bic1))[-1], collapse =
", "))
question11 <- data.frame(criterion, scores, variables_chosen)
knitr::kable(question11, "pipe", align = c("l" , "c", "c"))
#2.2 Variable Selection

## Load necessary library
library(leaps)

# Load the flight data
flight_data <-
read.csv("C:/Users/divya/Downloads/flight_filtered_data_final_ff.csv",
row.names = 1)

## Subset of predictors
predictors_flight <- c("price", "stops", "airline", "days_left","class",
"arrival_time", "departure_time", "duration")

## Perform an exhaustive search
mod_exhaustive_flight <- summary(regsubsets(price ~ ., data = flight_data[,
predictors_flight], nvmax = 3))

## Display the selected subset of features
mod_exhaustive_flight$which

## Display the residual sum of squares (RSS) for the selected models
mod_exhaustive_flight$rss


## Displaying the Adjusted R-squared values
mod_exhaustive_flight$adjr2

## Finding the index with the highest Adjusted R-squared value
which.max(mod_exhaustive_flight$adjr2)
```

```r
## Assigning the index with the highest Adjusted R-squared value to a
variable
best_r2_ind_flight <- which.max(mod_exhaustive_flight$adjr2)

## Displaying the selected subset of features for the best Adjusted R-squared
best_features_flight <- mod_exhaustive_flight$which[best_r2_ind_flight,]

## Get the number of columns in the selected subset
p_flight <- ncol(mod_exhaustive_flight$which)

mod_exhaustive_flight$which[best_r2_ind_flight,]
criterion = c("Adjusted R^2", "AIC", "BIC")
scores = c("0.8900948", "35339.25", "39880.14")
variables_chosen = c("stops, days_left, class", "class, stops, days_left,
duration", "stops")
question14 = data.frame(criterion, scores, variables_chosen)
knitr::kable(question14, "pipe", align=c("l" , "c", "c"))
#2.3.1 Check for collinearity
dataset <- flight[, c("price", "stops", "days_left", "class") ]

library(corrplot)
corrplot(cor(dataset),
method = 'color', order = 'hclust', diag = FALSE,
addCoef.col = 'black', tl.pos= 'd', cl.pos ='r')

library(olsrr)

# Fit a new model using the subsetted data
final_model <- lm(price ~ class + stops + days_left, data = dataset)

#check eigen value and condition index for the selected model
round(ols_eigen_cindex(final_model)[, 1:2], 4)
#2.3.2 Diagnostics with vif values
# Load the car package
library(car)
# fit model to data

vif(final_model)

# Summary of the model fitted with selected variables
summary(final_model)
#2.3.3 Diagnostics with Fitted Vs residual plot and bp test:
# check fitted vs residual and bp test for the model with selected variables
library(olsrr)
library(lmtest)

ols_plot_resid_fit(final_model)
bptest(final_model)
```

```r
#perform Shapiro wilk test for the model with selected variables
shapiro.test(resid(final_model))

#2.4 Fixing Model Violations
model_ws = lm(abs(resid(final_model)) ~ stops + days_left + class, data =
dataset)

# Calculate the weights as 1 / (fitted values)^2
weights <- 1 / fitted(model_ws)^2

# Run WLS
model_wls <- lm(price ~ stops + days_left + class, data = dataset, weights =
weights)

# Print the model summary
print(summary(model_wls))
#2.4.1 Checking violations using plotes and hypothesis tests
# Plot fitted values vs. weighted residuals
plot(fitted(model_wls), weighted.residuals(model_wls),
     pch = 16, xlab = 'Fitted Value', ylab = 'Weighted Residual')

# Change the range of weighted residuals
ylim <- c(-10, 10)  # Adjust as needed
abline(h = 0, lwd = 3, col = 'blue')

library(lmtest)
shapiro.test(resid(model_wls))
bptest(model_wls)
ols_plot_resid_qq(model_wls)
#2.5 Finding High Leverage points, Outliers, and Inflential Observations

high_lev_ids <- which(hatvalues(model_wls) > 2 * mean(hatvalues(model_wls)))

# Viewing the indices of high leverage points
high_lev_ids

outlier_test_cutoff = function(model, alpha = 0.05) {
    n = length(resid(model))
    qt(alpha/(2 * n), df = df.residual(model) - 1, lower.tail = FALSE)
}

# vector of indices for observations deemed outliers.
cutoff = outlier_test_cutoff(model_wls, alpha = 0.05)

which(abs(rstudent(model_wls)) > cutoff)

high_inf_ids <- which(cooks.distance(model_wls) > 4 /
length(cooks.distance(model_wls)))
print(high_inf_ids)
```

```r
#2.5.1 Handling High Leverage Points, Outliers, and Influential Observations:
noninfluential_ids = which(
    cooks.distance(model_wls) <= 4 / length(cooks.distance(model_wls)))

# fit the model on non-influential subset
model_fix = lm(price ~ stops + days_left + class, data = dataset, weights =
weights,
               subset = noninfluential_ids)

# return coefficients
summary(model_fix)
ols_plot_resid_fit(model_fix)
ols_plot_resid_qq(model_fix)
bptest(model_fix)
#install.packages("lmtest")
shapiro.test(resid(model_fix))
#2.7 Transformations
# Load the dplyr package
library(dplyr)
# Load the caret package
library(caret)

# Box-Cox transformation
# Load necessary library
library(corrplot)

library(MASS)


bc = boxcox(model_fix, lambda = seq(-0.25, 0.75, by = 0.05), plotit = TRUE)
lambda <- bc$x[which.max(bc$y)]
cat('The best λ value is :', lambda)
get_lambda_ci = function(bc, level = 0.95) {
    # lambda such that
    # L(lambda) > L(hat(lambda)) - 0.5 chisq_{1, alpha}
    CI_values = bc$x[bc$y > max(bc$y) - qchisq(level, 1)/2]

    # 95 % CI
    CI <- range(CI_values)

    # label the columns of the CI
    names(CI) <- c("lower bound","upper bound")

    CI
}

# extract the 95% CI from the box cox object
get_lambda_ci(bc)
#2.7.1 Perform regression analysis and fit a transformed model:
```

```r
# unlike the predictors, raising the response to a power is not a problem

# though the model is new with the response variable raising to the power
0.275(best lambda value), we are using the "dataset" with our selected
variables (stops, class, days_left) and  using the old model's predictors
only but not all predictors.

model_bc = lm(price ^ 0.275 ~ ., data = dataset)
model_bc

summary(model_bc)
#2.7.2 Assessment of Final Model (model_bc) Residuals:
ols_plot_resid_fit(model_bc)
ols_plot_resid_qq(model_bc)
shapiro.test(resid(model_bc))

bptest(model_bc)
#3.3 Hypothesis Testing Results for Predictors in the Final Model (model_bc)
summary(model_bc)
#3.3.1 Hypothesis Testing: Relationship between Response Variable (Price) and
each Predictor seperately(Stops, Class, Days_Left)
model1 = lm(price ~ stops, data = Data)
summary(model1)
model2 = lm(price ~ days_left, data = Data)
summary(model2)
model3 = lm(price ~ class, data = Data)
summary(model3)
#3.4 Analysis of Flight Characteristics and Ticket Prices:
ols_plot_added_variable(model_bc)
plot(price ~ stops, data = dataset, pch = 20, col = 'grey')
plot(price ~ class, data = dataset, pch = 20, col = 'grey')
plot(price ~ days_left, data = dataset, pch = 20, col = 'grey')
#6.1 Additional work - checking high influential points again and removing
them to fit and check for a better model

high_lev_ids <- which(hatvalues(model_bc) > 2 * mean(hatvalues(model_bc)))

# Viewing the indices of high leverage points
high_lev_ids

#Function for outliers
outlier_test_cutoff = function(model, alpha = 0.05) {
    n = length(resid(model))
    qt(alpha/(2 * n), df = df.residual(model) - 1, lower.tail = FALSE)
}

# vector of indices for observations deemed outliers.
cutoff = outlier_test_cutoff(model_bc, alpha = 0.05)
```

```r
which(abs(rstudent(model_bc)) > cutoff)
high_inf_ids <- which(cooks.distance(model_bc) > 4 /
length(cooks.distance(model_bc)))
print(high_inf_ids)
noninfluential_ids_bc = which(
    cooks.distance(model_bc) <= 4 / length(cooks.distance(model_bc)))

# fit the model on non-influential subset
model_fix_bc = lm(price ~ stops + days_left + class, data = dataset, weights
= weights,
                subset = noninfluential_ids_bc)

# return coefficients
summary(model_fix_bc)
shapiro.test(resid(model_fix_bc))
ols_plot_resid_fit(model_fix_bc)
ols_plot_resid_qq(model_fix_bc)
library(MASS)

# IRWLS with a limit of 100 iterations.
model_hub = rlm(price ~ stops + class + days_left, maxit = 100, data =
dataset)

summary(model_hub)

shapiro.test(resid(model_hub))
ols_plot_resid_fit(model_hub)
ols_plot_resid_qq(model_hub)
```