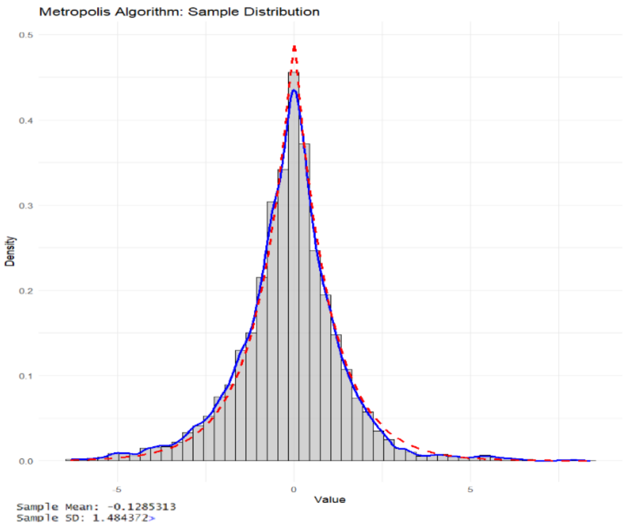
ST2195 REPORT

Programming for Data Science (Coursework)

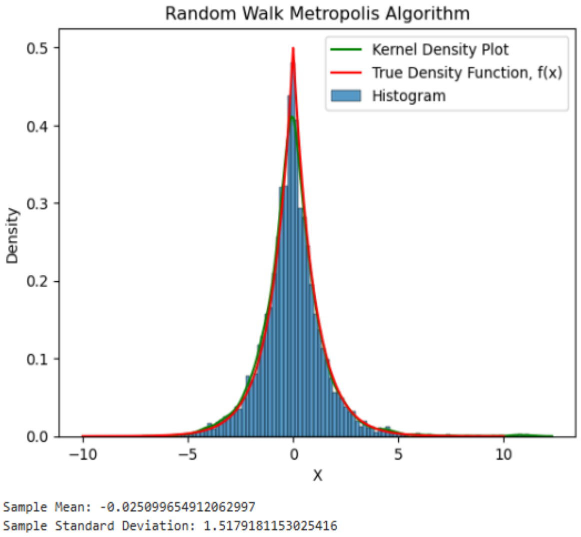
KANG ZHENG RUI
230700659

Part 1a:

In this section, we apply the Random Walk Metropolis Algorithm to generate samples and create a histogram combined with a kernel density plot. We begin by importing the required packages and defining the Probability Density Function $f(x)$. The Random Walk Metropolis Algorithm is then implemented by initializing an array of zeros to store the generated samples, followed by a loop that iteratively generates random x-values. Using the formula provided, the acceptance ratio is calculated, and a random number is drawn from a uniform distribution to decide whether to accept the proposed value. If the random number is smaller than the acceptance ratio, the proposed x-value is accepted and stored in the array. Otherwise, the previous x-value is retained. Using the specified parameters, the generated samples are used to plot a histogram of delays against x, which is overlaid with both a kernel density plot and the $f(x)$. graph. Finally, the sample mean, and standard deviation are calculated and presented. This process is applied to both the R and Python plots respectively.



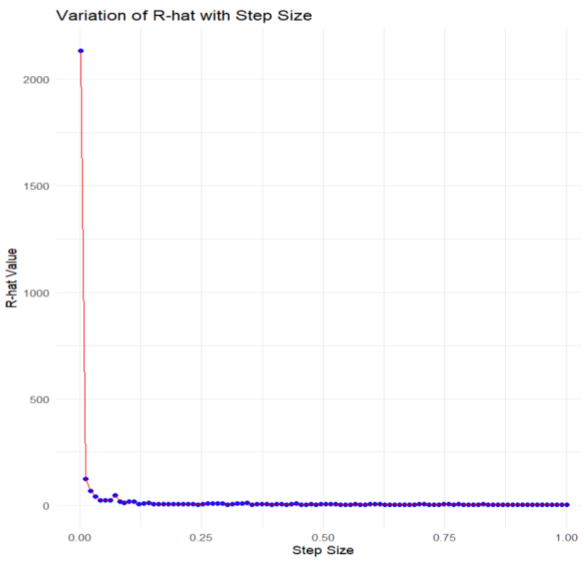
```
Sample Mean: -0.1285313
Sample SD: 1.484372>
```

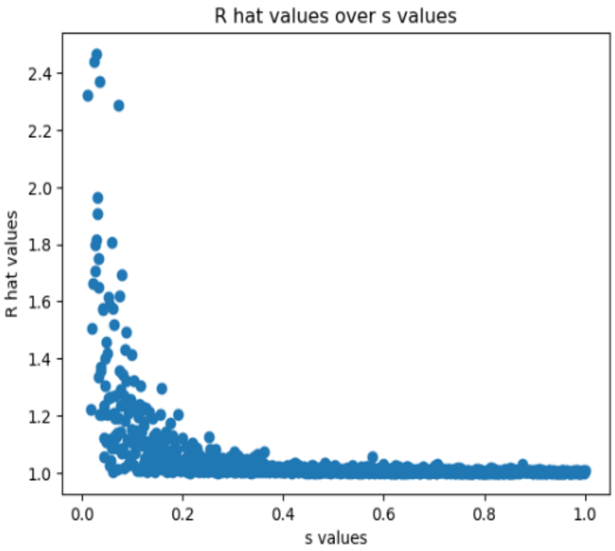


```
Sample Mean: -0.025099654912062997
Sample Standard Deviation: 1.5179181153025416
```

Part 1b:

In this section, we build upon the Metropolis Algorithm introduced in part 1a. A function is created to compute various statistics, including the sample mean, sample variance, overall within-sample variance, overall sample mean, between-sample variance, and the $\hat{R}$ value. Using the new parameters provided in the question and generating initial x_0 values, the function is executed, and a scatter plot is generated to display the $\hat{R}$ values against the corresponding step values. Only $\hat{R}$ values below 2.5 are considered valid, with any values exceeding this threshold regarded as outliers. This applies to both the R and Python plots respectively.

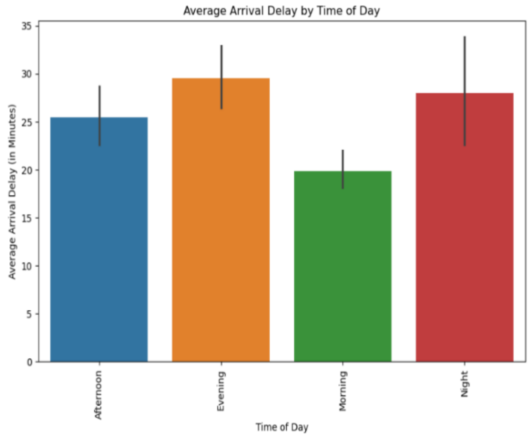


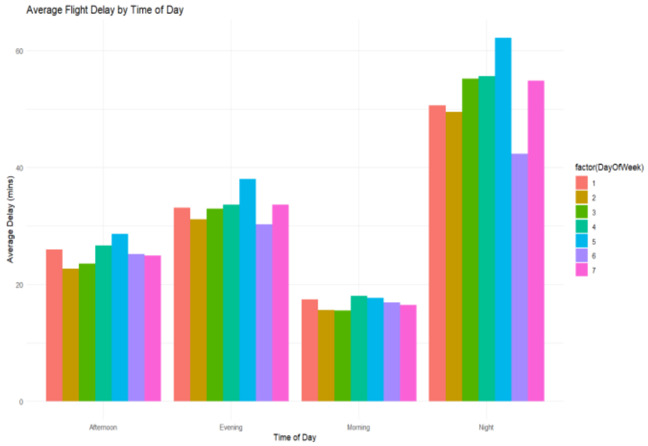


Part 2a:

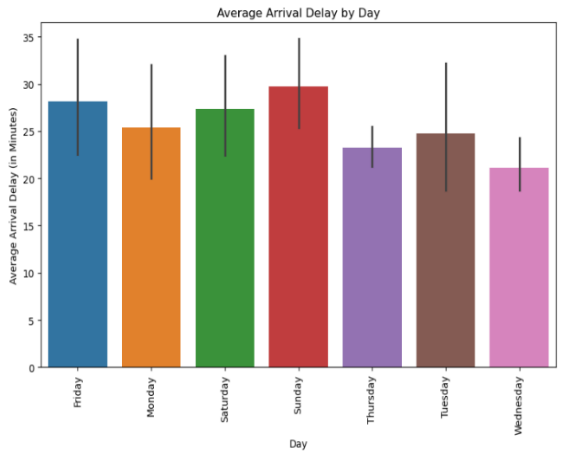
We start by importing the essential packages needed for part 2, then establish our working directory and set up a connection to an SQLite database. After selecting a range of 5 consecutive years from the Harvard Database (2000 to 2004), we proceed to load the relevant CSV files into a database table. Additionally, we create three other tables for airports, carriers, and planes, each corresponding to their respective CSV files.

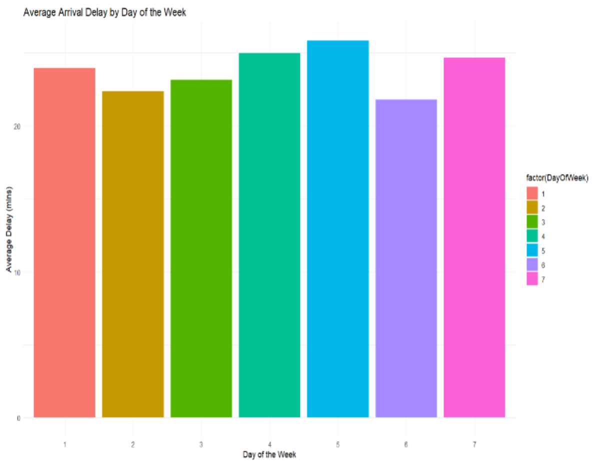
The goal of this section is to determine the optimal times of day and days of the week for flying each year. To achieve this, we begin by creating a function that divides the day into four distinct periods: Morning, Afternoon, Evening, and Night. Next, we develop two more functions: one to categorize the days of the week and another to remove negative ‘ArrDelay’ values. After filtering the data, we group it by year and calculate the average delay by computing the mean of the filtered ‘ArrDelay’ values. Lastly, we use ggplot2 in R and matplotlib in Python to visualize the results.





In the figures shown above, we compare the bar graphs to see which time of the day has the lowest delay, for the right-hand side figure, day 1 represents Monday and so on. We can conclude that the best time of the day to fly is in the morning for every year from 2000 to 2004.



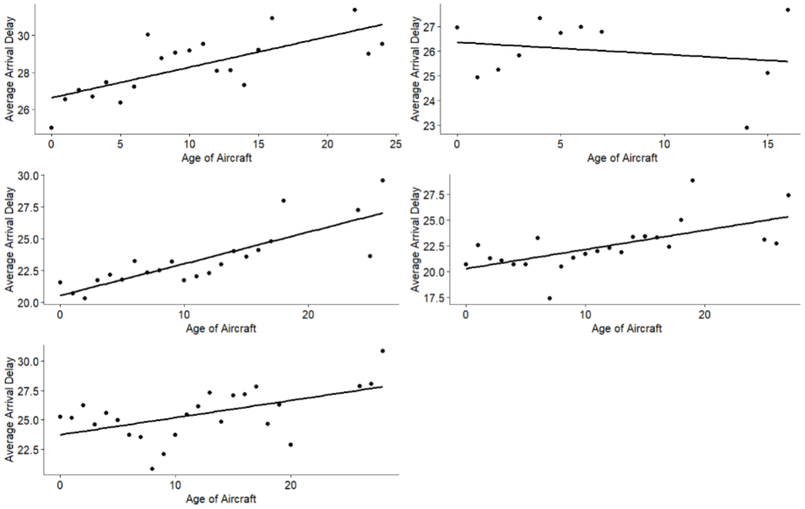


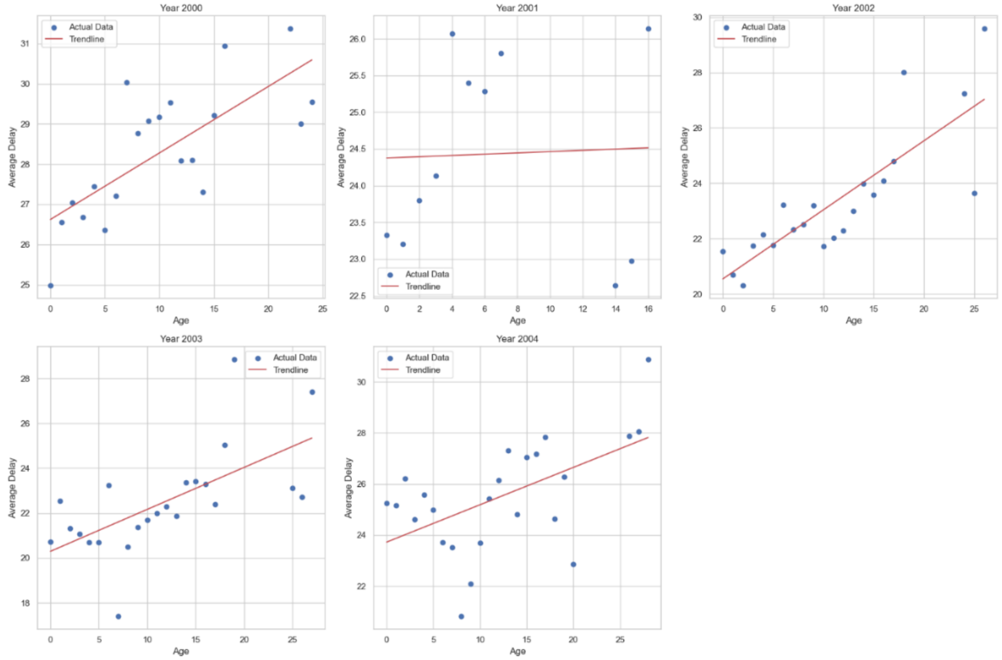
In the figures above, we compare the bar graphs to determine which day of the week has the lowest average delay. In the figure on the right, day 1 represents Monday, and the days continue sequentially. From the analysis, we can conclude that Thursday and Saturday are the best days to fly, as they show the lowest average delays.

Part 2b:

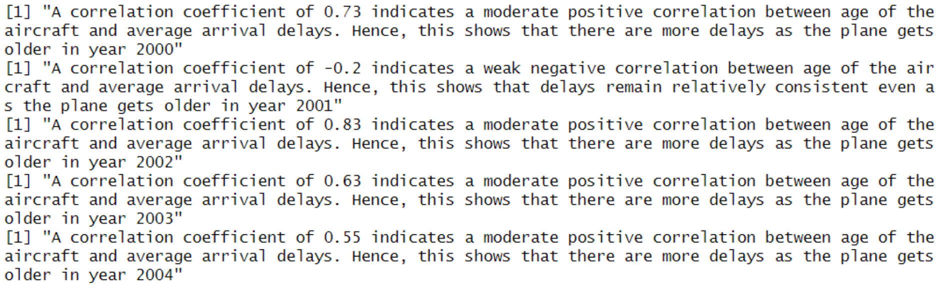
In this section, we explore whether older aircraft tend to experience more delays. To gather the required data, we merge the plane tables with the 'ontime' DataFrame, which is created by processing the data over the years. The age of each aircraft is calculated by subtracting its year of manufacture from the flight year.

To ensure data accuracy, we filter out rows with negative plane ages, negative arrival delays, and missing values. After cleaning the data, we created a scatter plot to visualize the correlation between aircraft age and average delay. (for both R and Python respectively)

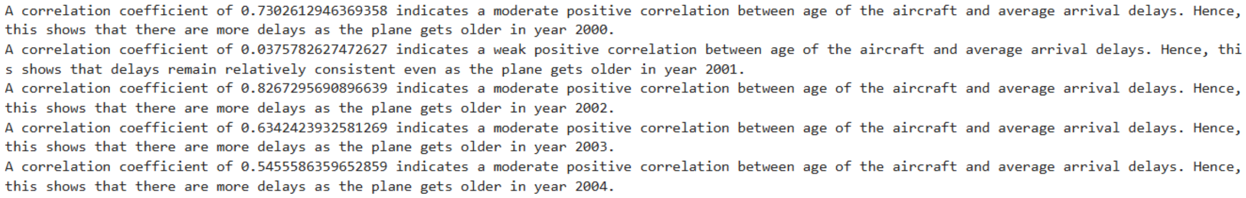




Using Linear Regression in this section provided a clearer representation of the data in both of the figures presented above. The correlation coefficient, derived from the slope of the best-fit line, measures the relationship between aircraft age and average delay. The trendline, a straight line that best fits the data points, serves as an approximation of this relationship. As a result, a slight variation in the correlation coefficient is observed. The conclusions are as follows for both R and Python respectively:



```
[1] "A correlation coefficient of 0.73 indicates a moderate positive correlation between age of the
aircraft and average arrival delays. Hence, this shows that there are more delays as the plane gets
older in year 2000"
[1] "A correlation coefficient of -0.2 indicates a weak negative correlation between age of the air
craft and average arrival delays. Hence, this shows that delays remain relatively consistent even a
s the plane gets older in year 2001"
[1] "A correlation coefficient of 0.83 indicates a moderate positive correlation between age of the
aircraft and average arrival delays. Hence, this shows that there are more delays as the plane gets
older in year 2002"
[1] "A correlation coefficient of 0.63 indicates a moderate positive correlation between age of the
aircraft and average arrival delays. Hence, this shows that there are more delays as the plane gets
older in year 2003"
[1] "A correlation coefficient of 0.55 indicates a moderate positive correlation between age of the
aircraft and average arrival delays. Hence, this shows that there are more delays as the plane gets
older in year 2004"
```



```
A correlation coefficient of 0.7302612946369358 indicates a moderate positive correlation between age of the aircraft and average arrival delays. Hence,
this shows that there are more delays as the plane gets older in year 2000.
A correlation coefficient of 0.0375782627472627 indicates a weak positive correlation between age of the aircraft and average arrival delays. Hence, thi
s shows that delays remain relatively consistent even as the plane gets older in year 2001.
A correlation coefficient of 0.8267295690896639 indicates a moderate positive correlation between age of the aircraft and average arrival delays. Hence,
this shows that there are more delays as the plane gets older in year 2002.
A correlation coefficient of 0.6342423932581269 indicates a moderate positive correlation between age of the aircraft and average arrival delays. Hence,
this shows that there are more delays as the plane gets older in year 2003.
A correlation coefficient of 0.5455586359652859 indicates a moderate positive correlation between age of the aircraft and average arrival delays. Hence,
this shows that there are more delays as the plane gets older in year 2004.
```

The results indicate that, except for 2001, older planes tend to experience more delays across all other years. However, the correlation between plane age and average delays is minimal in most years. Therefore, we cannot definitively conclude that older aircraft are the primary cause of increased delays.

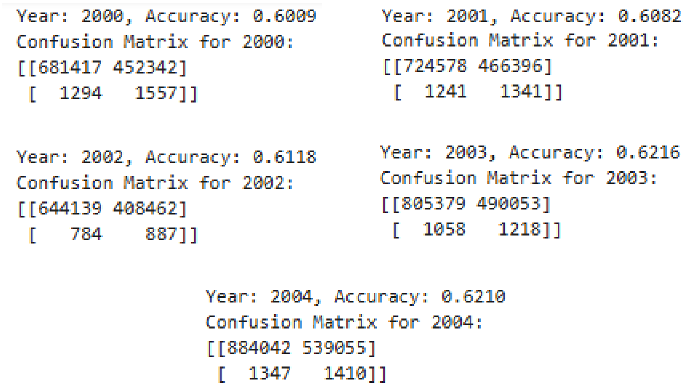
Part 2c:

In this section, we aim to build a logistic regression model to predict the likelihood of flight diversions in the US, and to visualize the impact of different variables over the years. The process begins by loading and processing data from five consecutive years. For each year, data is read in chunks, which are then merged with a separate airports table to add the geographic information (latitude and longitude) for both the origin and destination airports.

For the logistic regression model, we first transform the scheduled departure and arrival times into hourly values to simplify the analysis. Next, categorical variables such as origin, destination, and carrier are converted into numerical representations using R's as.factor() function to prepare them for the logistic regression. Any missing values are handled by replacing them with zeros to ensure the dataset remains complete.

The data is split into training and testing subsets, with 80% used for training the model and 20% reserved for testing. To handle the class imbalance between diverted and non-diverted flights, we apply oversampling techniques on the minority class in the training data. The logistic regression model is then fitted, using flight diversion as the target variable and various extracted features (such as departure time, airport location, and carrier) as predictors. After training, we use the test data to make predictions and evaluate the model's performance by calculating accuracy and generating a confusion matrix to assess how well the model classifies diverted flights.

Finally, we extract the model’s coefficients to examine the influence of each feature on the probability of diversion. These coefficients are visualized through bar graphs, making it easier to interpret the relative importance of each variable in predicting flight diversions.



```
Year: 2000, Accuracy: 0.6009
Confusion Matrix for 2000:
[[681417 452342]
 [  1294   1557]]
```

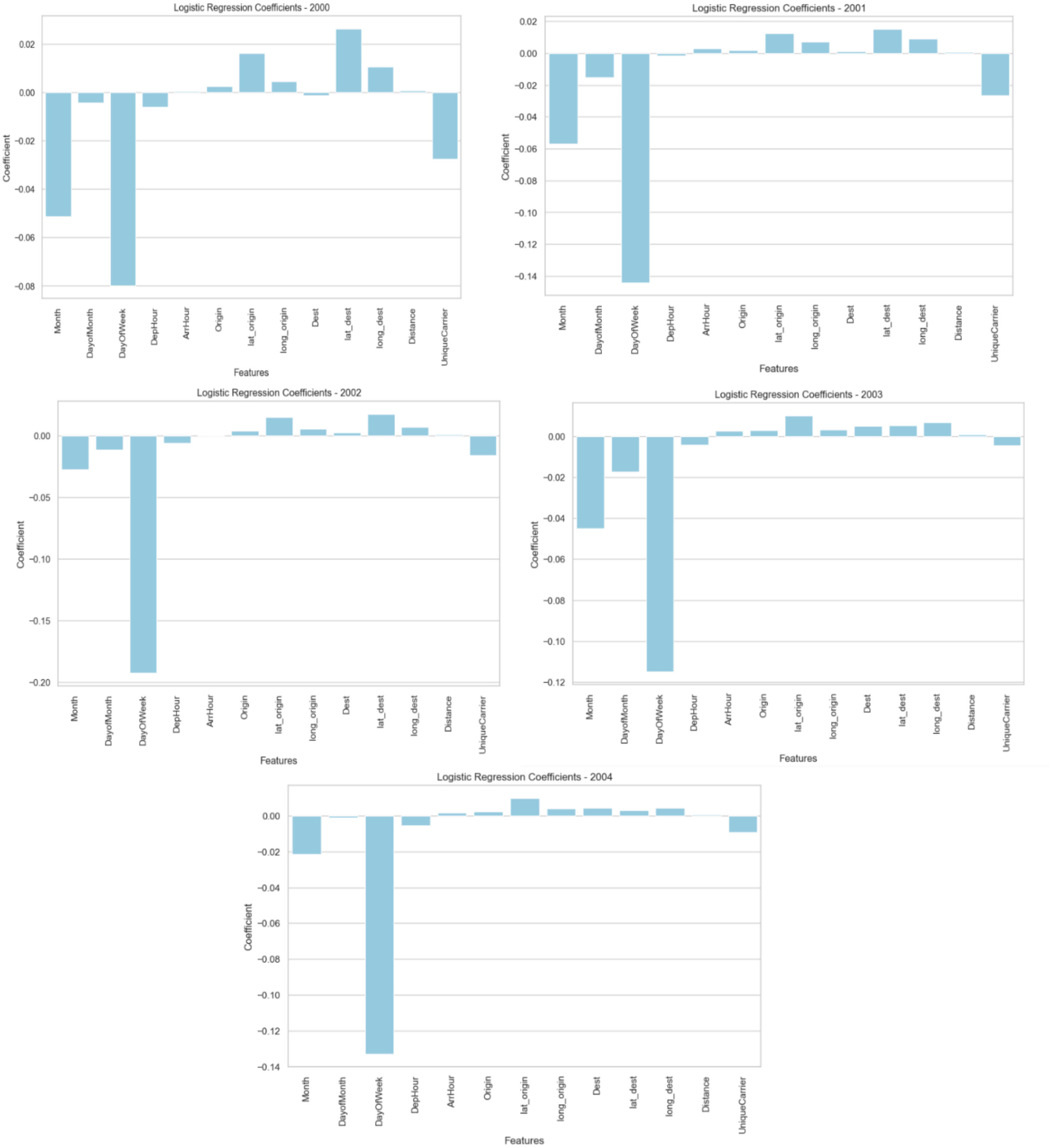
```
Year: 2001, Accuracy: 0.6082
Confusion Matrix for 2001:
[[724578 466396]
 [  1241   1341]]
```

```
Year: 2002, Accuracy: 0.6118
Confusion Matrix for 2002:
[[644139 408462]
 [   784    887]]
```

```
Year: 2003, Accuracy: 0.6216
Confusion Matrix for 2003:
[[805379 490053]
 [  1058   1218]]
```

```
Year: 2004, Accuracy: 0.6210
Confusion Matrix for 2004:
[[884042 539055]
 [  1347   1410]]
```

As shown in the figure above, the accuracy across the years 2000 to 2004 ranges between 60%-62%. The confusion matrix shows that the mentioned issue of diverted and non-diverted flights imbalance is being resolved. The model is relatively accurate after data prediction balancing.



In the figure above, the plotted coefficients are being plotted against the various variables, revealing that ‘Month’ and ‘DayOfWeek’ emerges as the top 2 most significantly influential variables.

This logistic regression analysis has provided valuable insights into the factors influencing flight diversion. It highlights the importance of transforming raw flight data into a well-organized format, utilizing logistic regression for analysis, and evaluating the model's performance, ultimately improving our ability to understand and predict flight diversion events.