Table of Contents

[Question 1 - Random Walk Metropolis 3](#_Toc162392563)

[Part (a) 3](#_Toc162392564)

[Part (b) 3](#_Toc162392565)

[Question 2 – Airline On Time Data 4](#_Toc162392566)

[Part (a) – Best Times to Minimise Delays Each Year 4](#_Toc162392567)

[Scope of Analysis 4](#_Toc162392568)

[Data Preprocessing 4](#_Toc162392569)

[Interpretation of Results 4](#_Toc162392570)

[Results 4](#_Toc162392571)

[Part (a) – Best Days of the Week to Minimise Delays Each Year 5](#_Toc162392572)

[Data Preprocessing 5](#_Toc162392573)

[Interpretation of Results 5](#_Toc162392574)

[Results 5](#_Toc162392575)

[Part (b) – Evaluate whether older planes suffer more delays on a year-to-year basis 5](#_Toc162392576)

[Data Preprocessing 5](#_Toc162392577)

[Interpretation of Results 6](#_Toc162392578)

[Results 6](#_Toc162392579)

[Part (c) – Logistic Regression Model 8](#_Toc162392580)

[Data Preprocessing 8](#_Toc162392581)

[Results 8](#_Toc162392582)

# Question 1 - Random Walk Metropolis

## Part (a)

Using set.seed(0) in R, the generated samples had the following descriptive statistics:

* Sample mean: -0.07359532
* Sample standard deviation: 1.540904

Using np.random.seed(0) in Python, the generated samples had the following descriptive statistics:

* Sample Mean: 0.00655248
* Sample Standard Deviation: 1.47114957

The generated samples for both R and Python generated a similar histogram as shown below.

A diagram of a normal distribution

Description automatically generated

*Figure 1: Histogram of samples, along with kernel density plot and overlayed graph of f(x)*

## Part (b)

Running the random walk Metropolis algorithm for both R and Python produced similar results as shown in the histograms below.

A graph with a bar and numbers

Description automatically generated with medium confidence

*Figure 2.1: Histogram of values of Ȓ over entire range of Ȓ*

A graph of a graph

Description automatically generated

*Figure 2.2: Histogram of values of Ȓ for range 1.00 ≤ Ȓ ≤ 1.05*

# Question 2 – Airline On Time Data

*Data from years 1998 to 2007 were used for all parts under Question 2.*

## Part (a) – Best Times to Minimise Delays Each Year

### Scope of Analysis

For our analysis, we will be primarily focusing on whether a flight experiences a delay or not. This means that we will ignore the severity of the delay and how early a flight arrives relative to its scheduled arrival. We assume that passengers are more sensitive to any form of delays compared to arriving at their destination early. We can justify this as delays of any degree can cause a domino effect of creating more delays along a passenger’s itinerary upon reaching their destination.

### Data Preprocessing

To find the best times to minimise delays each year, we will have to first process the data. For each year, we conduct feature engineering to find the hour at which the flight is scheduled to leave. This is done by creating a new column, ‘CRSDepHour’, which is calculated by performing a floor division on the column ‘CRSDepTime’ by a factor of 100. We will also create a new column, ‘isDelayed’, which takes on the value of 1 if any of the delay columns in our yearly dataframe have a value of 1, and 0 otherwise.

Next, we will group the entire yearly dataframe by the ‘CRSDepHour’ column and perform a count aggregation. This gives us a dataframe that shows the number of flights that were scheduled to fly at every hour for that year. Afterwards, we will perform a similar operation but only on a subset of the data where the flights were delayed, i.e. the value for the ‘isDelayed’ columns is 1. This gives us another dataframe that shows number of flights that were delayed at every hour that they were scheduled to fly at for the year. We will then perform a left join operation on these two dataframes, joining them based on the ‘CRSDepHour’ column. We can create a new calculated column, delayed probability, by taking dividing the number of delayed flights by the total number of flights for each hour. This column indicates the probability of delay based on the historical data that we have. Finally, we can create a bar chart with the delayed probability as the y-axis and hour of the day as the x-axis to visualise the data.

### Interpretation of Results

Intuitively, the hour with the lowest value of delayed probability should be the best time to minimise delays. However, there are scenarios where this will not be the case. For example, for 2003, flights scheduled between 3am to 4am had a delayed probability of 0.286, the lowest for the year. However, due to the small sample size of 3 total flights during this hour, we cannot be certain that the result is statistically significant. In such cases, we will take the next statistically significant result. These cases will be marked with an \* symbol. As a general rule of thumb, we will be considering hours with at least 100 flights throughout the year.

### Results

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **1998** | **1999** | **2000** | **2001** | **2002** |
|  |  |  |  |  |
| **2003** | **2004** | **2005** | **2006** | **2007** |
|  |  |  |  |  |

The following are the best times to minimise delays for each year:

|  |  |
| --- | --- |
| * 1998: Between 5am to 6am\* * 1999: Between 5am to 6am\* * 2000: Between 3am to 4am * 2001: Between 3am to 4am\* * 2002: Between 5am to 6am\* | * 2003: Between 6am to 7am\* * 2004: Between 5am to 6am\* * 2005: Between 5am to 6am * 2006: Between 5am to 6am * 2007: Between 5am to 6am |

## Part (a) – Best Days of the Week to Minimise Delays Each Year

### Data Preprocessing

Data preprocessing to find the best days of the week to minimise delays each year is conducted in a similar manner to finding the best times to minimise delays each year. The only difference being that we will be focusing on the column ‘DayOfWeek’ instead. Group by, count, left join operations and finding the delayed probability for each day are also conducted in a similar manner.

### Interpretation of Results

Similarly, we will also interpret the results in the same manner. The day with the least delayed probability that is statistically significant will be the best day of the week to minimise delay for that year.

### Results

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **1998** | **1999** | **2000** | **2001** | **2002** |
|  |  |  |  |  |
| **2003** | **2004** | **2005** | **2006** | **2007** |
|  |  |  |  |  |

The following are the best days of the week to minimise delays for each year:

|  |  |
| --- | --- |
| * 1998: Saturday * 1999: Saturday * 2000: Tuesday * 2001: Tuesday * 2002: Saturday | * 2003: Saturday * 2004: Saturday * 2005: Saturday * 2006: Tuesday * 2007: Saturday |

## Part (b) – Evaluate whether older planes suffer more delays on a year-to-year basis

### Data Preprocessing

We first use the plane-data.csv file to create a dataframe containing the tail numbers and issue date of the aircraft. Next, we will use our yearly dataframe and group by tail number to find the total number of flights each unique aircraft undergone to create a new dataframe. We will do a similar operation to find the number of delayed flights each unique aircraft has undergone to create another dataframe. We will then perform an inner join on these two dataframes, joining them by ‘TailNum’. Subsequently, we will use the joined dataframe to perform another inner join with the tail number dataframe. From here, we are able to find the delayed probability for each tail number, similar to that of part (a). We can also find the age of the unique aircraft by extracting the year from the issue date and calculating the age of the aircraft at the year of concern. We will also be filtering for unique aircrafts that have negative age or have lesser than 100 flights for the year, regarding them as noise.

Once we have our final dataframe, run a linear regression to evaluate if older planes suffer more delays on a year-to-year basis. The independent variable will the age of the aircraft while the dependent variable will be the historical delayed probability that we calculated.

### Interpretation of Results

Our null hypothesis for this question will be that there is no difference between the rate of delay of an older plane than a newer plane. The alternative hypothesis will be that older planes have a higher rate of delay than a newer plane. The formulation of our null and hypothesis tests can be seen below. We will be testing our hypothesis at a 5% level of significance.

For each year, we will be examining the result of the linear regression that we ran. In particular, we will be focusing on whether the coefficient for the age variable is statistically significant. If it is, we will also be looking at the value of the coefficient and whether it is positive or negative. We will be testing our hypothesis at a 5% level of significance.

### Results

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **1998** | **1999** | **2000** | **2001** | **2002** |
|  |  |  |  |  |
| **2003** | **2004** | **2005** | **2006** | **2007** |
|  |  |  |  |  |

#### 1998

Our linear regression model shows that age of an aircraft is statistically significant in predicting the delayed probability for flights in the year 1998 due to a very low p-value that is close to 0. Our variable ‘Age’ has an estimated coefficient of -0.0027543. This suggests that for every unit increase in an aircraft’s age, the probability that a flight involving that same aircraft gets delayed is decreased by 0.0027543, ceteris paribus. As such, we fail to reject our null hypothesis.

#### 1999

Our linear regression model shows that age of an aircraft is not statistically significant in predicting the delayed probability for flights in the year 1999 due to a high p-value of 0.0709. This suggests that the age of an aircraft has no significant impact on the delayed probability of a flight that it is involved in. As such, we fail to reject our null hypothesis.

#### 2000

Our linear regression model shows that age of an aircraft is not statistically significant in predicting the delayed probability for flights in the year 2000 due to a high p-value of 0.345. This suggests that the age of an aircraft has no significant impact on the delayed probability of a flight that it is involved in. As such, we fail to reject our null hypothesis.

#### 2001

Our linear regression model shows that age of an aircraft is statistically significant in predicting the delayed probability for flights in the year 2001 due to a very low p-value that is close to 0. Our variable ‘Age’ has an estimated coefficient of -0.007573. This suggests that for every unit increase in an aircraft’s age, the probability that a flight involving that same aircraft gets delayed is decreased by 0.007573, ceteris paribus. As such, we fail to reject our null hypothesis.

#### 2002

Our linear regression model shows that age of an aircraft is not statistically significant in predicting the delayed probability for flights in the year 2002 due to a high p-value of 0.0641. This suggests that the age of an aircraft has no significant impact on the delayed probability of a flight that it is involved in. As such, we fail to reject our null hypothesis.

#### 2003

Our linear regression model shows that age of an aircraft is statistically significant in predicting the delayed probability for flights in the year 2003 due to a very low p-value of 0.000754. Our variable ‘Age’ has an estimated coefficient of 0.0010651. This suggests that for every unit increase in an aircraft’s age, the probability that a flight involving that same aircraft gets delayed is increased by 0.0010651, ceteris paribus. As such, we have enough evidence to reject our null hypothesis and conclude that older planes suffer more delays for year 2003.

#### 2004

Our linear regression model shows that age of an aircraft is statistically significant in predicting the delayed probability for flights in the year 2004 due to a very low p-value that is close to 0. Our variable ‘Age’ has an estimated coefficient of 0.0023270. This suggests that for every unit increase in an aircraft’s age, the probability that a flight involving that same aircraft gets delayed is increased by 0.0023270, ceteris paribus. As such, we have enough evidence to reject our null hypothesis and conclude that older planes suffer more delays for year 2004.

#### 2005

Our linear regression model shows that age of an aircraft is statistically significant in predicting the delayed probability for flights in the year 2005 due to a low p-value 0.0324. Our variable ‘Age’ has an estimated coefficient of 0.0004806. This suggests that for every unit increase in an aircraft’s age, the probability that a flight involving that same aircraft gets delayed is increased by 0.0004806, ceteris paribus. As such, we have enough evidence to reject our null hypothesis and conclude that older planes suffer more delays for year 2005.

#### 2006

Our linear regression model shows that age of an aircraft is statistically significant in predicting the delayed probability for flights in the year 2005 due to a very low p-value 0.00264. Our variable ‘Age’ has an estimated coefficient of -0.0005763. This suggests that for every unit increase in an aircraft’s age, the probability that a flight involving that same aircraft gets delayed is increased by 0.0005763, ceteris paribus. As such, we have enough evidence to reject our null hypothesis and conclude that older planes suffer more delays for year 2006.

#### 2007

Our linear regression model shows that age of an aircraft is not statistically significant in predicting the delayed probability for flights in the year 2002 due to a high p-value of 0.162. This suggests that the age of an aircraft has no significant impact on the delayed probability of a flight that it is involved in. As such, we fail to reject our null hypothesis.

## Part (c) – Logistic Regression Model

### Data Preprocessing

In order to fit a logistic regression model for the probability of diverted US flights, we will first have to decide on the features that we want to use. Features that we will be using from the yearly dataframes are ‘Month’, ‘DayOfWeek’, ‘Origin’, ‘UniqueCarrier’, ‘Dest’, ‘Distance’ and ‘Diverted’. We will also be using the coordinates of the origin airport and the destination airport, which can be obtained from the airports.csv file. We can create a dataframe that stores the data from the airports.csv file and use our yearly dataframes to perform two left join operations, joining on the IATA codes for the origin airports and the destination airports separately. This operation creates four new columns, a latitude and longitude column for each origin and destination airport. These columns are called ‘origin lat’, ‘origin long’, ‘dest lat’ and ‘dest long’ respectively. Since each airport has its own unique coordinates, we will then drop the ‘Origin’ and ‘Dest’ columns to prevent multicollinearity.

Furthermore, since ‘UniqueCarrier’ is a categorical variable, we will perform one-hot encoding to convert it into a numerical format that our model can understand and utilise effectively. Lastly, we will also perform feature engineering on the columns ‘CRSDepTime’, ‘CRSArrTime’ and ‘DayofMonth’. We will be performing a floor division on the first two variables by a factor of 100 to obtain the columns, ‘CRSDepHour’ and ‘CRSArrHour’, which indicate the scheduled departure hour and scheduled arrival hour of a flight respectively. Floor division on ‘DayofMonth’ by a factor of 7 will also be done to obtain ‘WeekofMonth’, which indicates which week of the month the flight takes place on.

Once all the preparation done, we can then run a logistic regression model with the following independent variables: ‘Month’, ‘WeekofMonth’, ‘DayOfWeek’, ‘CRSDepHour’, ‘CRSArrHour’, every category of ‘UniqueCarrier’ for that year, ‘Distance’, ‘origin lat’, ‘origin long’, ‘dest lat’ and ‘dest long’. These independent variables will be used to predict the dependent variable ‘Diverted’. This can be formulated in the equation below.

After running the logistic regression model for each year, we will store the coefficients of each variable along with the year of the dataset. This allows us to easily plot a line chart of each variable over the years in our dataset.

### Results

|  |  |  |  |
| --- | --- | --- | --- |
| **Intercept** | **Month** | **WeekofMonth** | **DayOfWeek** |
|  |  |  |  |
| **CRSDepHour** | **CRSArrHour** | **Distance** | **origin lat** |
|  |  |  |  |
| **origin long** | **dest lat** | **dest long** | **UniqueCarrier\_AS** |
|  |  |  |  |
| **UniqueCarrier\_CO** | **UniqueCarrier\_DL** | **UniqueCarrier\_HP** | **UniqueCarrier\_NW** |
|  |  |  |  |
| **UniqueCarrier\_TW** | **UniqueCarrier\_UA** | **UniqueCarrier\_US** | **UniqueCarrier\_WN** |
|  |  |  |  |
| **UniqueCarrier\_AQ** | **UniqueCarrier\_MQ** | **UniqueCarrier\_B6** | **UniqueCarrier\_DH** |
|  |  |  |  |
| **UniqueCarrier\_EV** | **UniqueCarrier\_FL** | **UniqueCarrier\_HA** | **UniqueCarrier\_OO** |
|  |  |  |  |
| **UniqueCarrier\_TZ** | **UniqueCarrier\_XE** | **UniqueCarrier\_OH** | **UniqueCarrier\_F9** |
|  |  |  |  |
| **UniqueCarrier\_YV** | | **UniqueCarrier\_AA** | |
|  | |  | |