*PROGRAMING FOR DATA SCIENCE*

*COURSE WORK*

Student No: **220648516**



**Contents**

[**PART – 1. Metropolis Hastings Algorithm** 3](#_Toc166179947)

[Simulation using Random Walk Metropolis 3](#_Toc166179948)

[The simulation converges using R̂ values 4](#_Toc166179949)

[**PART – 2. Flight Analysis** 5](#_Toc166179950)

[1. Introduction 5](#_Toc166179951)

[2. Data Cleaning 5](#_Toc166179952)

[3. Best times and days of the week to minimize delays 6](#_Toc166179953)

# **PART – 1. Metropolis Hastings Algorithm**

## Simulation using Random Walk Metropolis

The probability density function was simulated using the random walk Metropolis method.

f (x) = 𝟏/𝟐 𝒆−|𝒙|

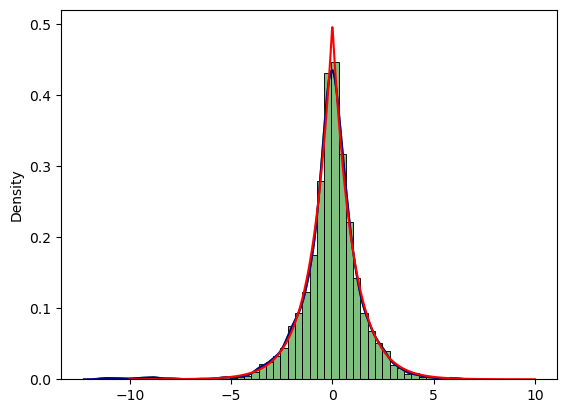


Figure 1 Histogram and kde generated to achieve f(x) in python

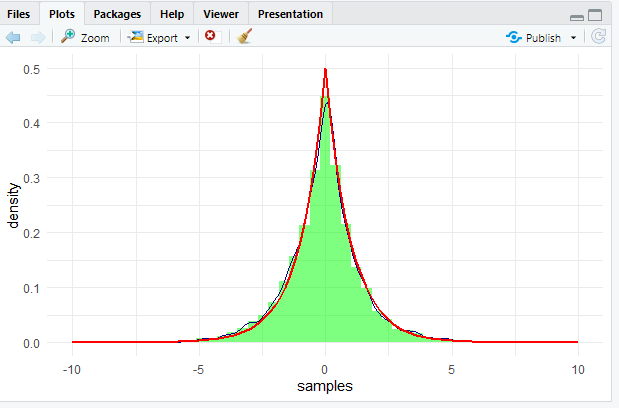
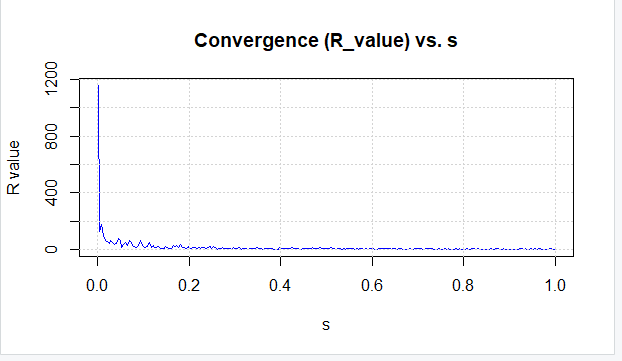


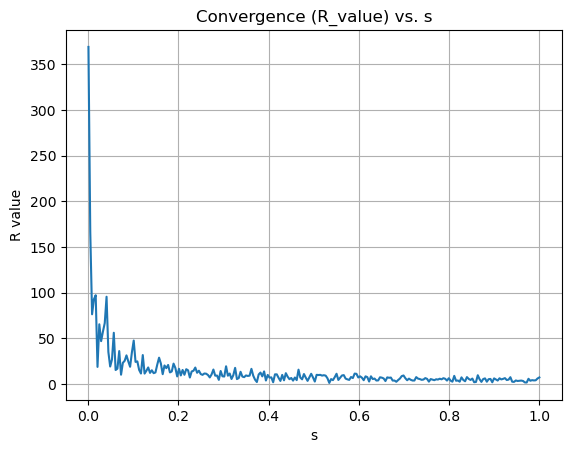
Figure 2 Histogram and kde generated in R

An initial value = x0 was initially configured with a fixed standard deviation and s = 1, to generate the values of N = 10000 values of X. During each repetition, a random number x, was simulated from a normal distribution of mean x𝑖−1 and s = 1 to compute a ratio and was compared with a random number generated from the uniform distribution. The *DARK BLUE* line (KDE) depicts as a result of the simulation for the achieving the *BLACK LINE* f(x).

## The simulation converges using R̂ values

To confirm the algorithm converges, the instructions given in the guidelines were followed in order to create a plot for the R̂ value. First the parameters were given as instructed N = 2000, s=1, J=4. 4 different initial value were used to generate 4 different sequences of N = 2000, using the metropolis hastings then many values were generated for R̂ values. Which is plotted into a plot. At the end the R̂ value is converging towards 1-1.05 as required.





# **PART – 2. Flight Analysis**

## Introduction

This report is based on the analysis of planes that have flown over the areas and over time, also the delays they experienced when departing and arriving at airports in the USA. The data set was found on the Harvard dataverse. During the project the year 2006 and 2007 were selected and sampled randomly. For the mentioned process programing languages like Python and R were used.

## Data Cleaning

The year 2006 and 2007 which sampled randomly for 10% was cleaned to ensure the accuracy of the dataset used in the process before data wrangling and visualizing of the findings. During the process first columns and shapes were analyzed. Then the two years were merged since both containing same columns for easy handling. null values and missing values were removed for some datasets and filled with mean values for some datasets depending on the situation for datasets. And duplicate values were removed for all the datasets. Further adjustments were made depending on the question, first question was all about to find best time and day of the week.

## 3. Best times and days of the week to minimize delays

3.1 Best time of the week

Main merged data of 2006 and 2007 data was used for the analysis of this question. After importing the main data set “merged\_data”, ArrDelay column’s and the DepDelay column’s mean values were taken to find the delay easily for plotting. Mean value for null values of the two columns were already filled on data cleaning process. As averages give a clear picture for data, the averages of both delays were taken on this process.

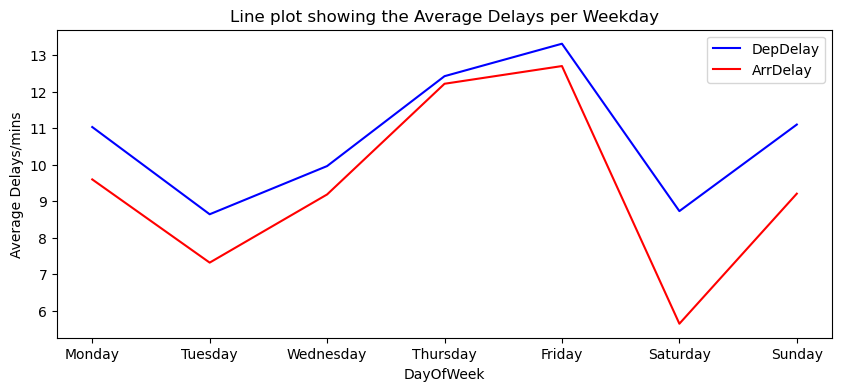


Figure 3 line plot for average delays generated in python

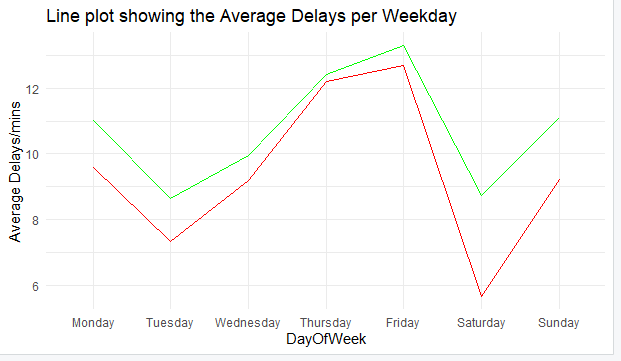
**

Figure 4 lineplot for average delays generated in R

From the above line plots it is clearly visible both the delays indicating Saturday is the best day to minimize the delays.

3.2 Best time of the week

In this part of the question to find the best time of the day to minimize the delays 24 hours were split into 6 slots and each contains 4 hours range, they were 00.00-04.00, 04.00-08.00, 08.00-12.00, 12-00-16.00, 16.00-20.00, 20.00-24.00. and sum of those delays of 4 hours range were fell into the time slot accordingly. A box plot was used to visualize for this question.

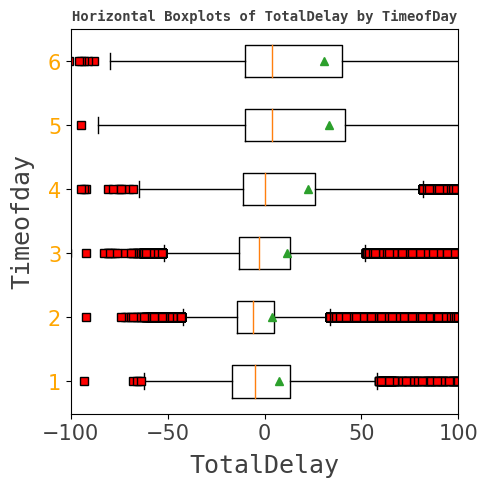
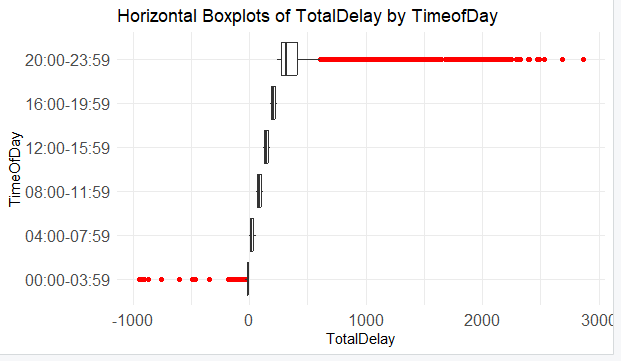
 

Figure 5 Boxplot in python Figure 6 boxplots in R

It is clearly visible that the above boxplots containing lot of outliers, as outliers affect mean values it is best way to refer median values as medians are not affected by the outliers, which is the middle line would be a best way. Seemingly slot 1 and 2 appear to be slightly close slot is a bit lower. Accordingly, Slot 2 04.00-08.00 is the best time possible to minimize delays in a day.

## 4. Do older planes suffer more delays?

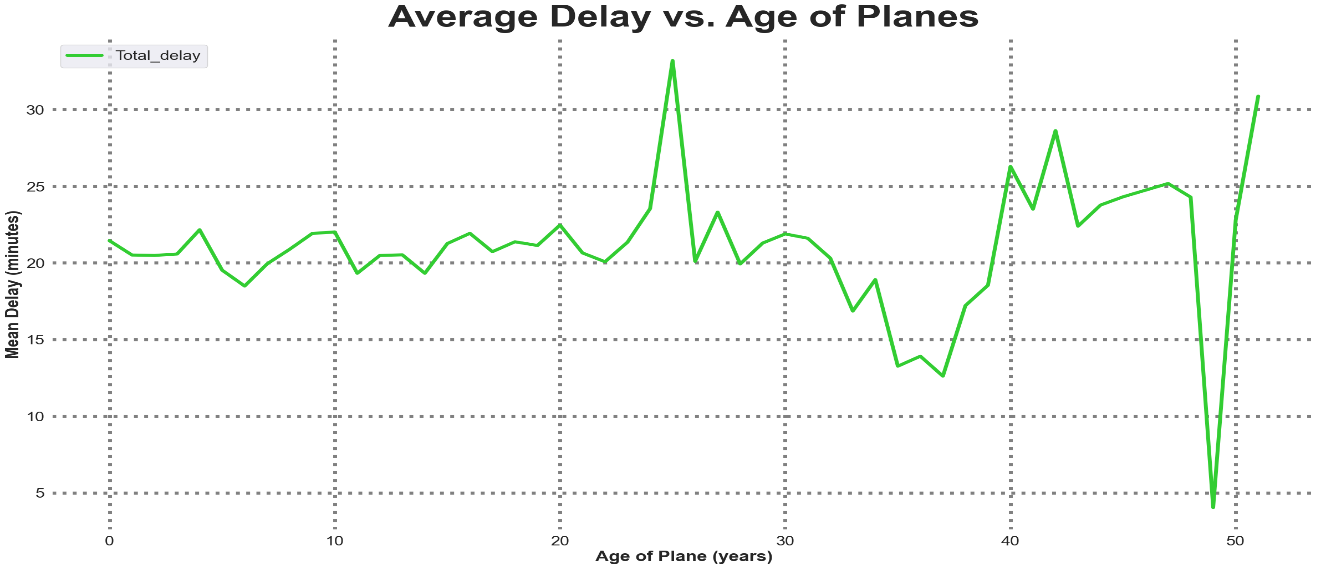
In order to find whether older planes suffer more delays the merged data of 2006 and 2007 years used along with plane data. Those two data sets were merged making the Tailnum column as primary key. Null values of the plane data set were already removed in the data cleaning process. To find the answer, the ages of planes were needed and to achieve that manufactured year and the current year were subtracted from one to each other. After finding the ages, age column was created. Then arrival and departure delays were added up to find total delays. after that delays vs age of the plane data plotted into a line plot.

Figure 7 Average delay vs age of the planes plot generated in python

The above line plot show fluctuations(mean delay) based on the ages of the planes. Theres a large drop in delay between age 47-50 and high fluctuation on planes as age decreasing after around age 47. Even though there have been fluctuations the visualization doesn’t support the argument that older planes suffer more delays.