EDA on Daegu Metropolitan Rapid Transit System

By, Dharathi Venkatesh - 2020118156

***Abstract*—This report consists of the details of Exploratory Data Analysis (EDA) performed on the data that consisted of the information on the Daegu Metropolitan Rapid Transit System. Using the concepts of EDA various conclusions were made, a few of which included the identification of 'Rush hour' and 'Least rush hour' in train stations and also finding out the train station that is the 'most crowded' and 'least crowded'.**

Keywords—Exploratory Data Analysis (EDA), Daegu Metropolitan Rapid Transit System, Python.

I. Introduction

Exploratory data analysis (EDA) is a technique used by data scientists to explore and investigate data sets and define their fundamental attributes, sometimes employing data visualization tools. It makes it easier for data scientists to uncover patterns, test hypotheses, and check assumptions by assisting them in deciding how to efficiently manipulate data sources to achieve the answers they require. True, EDA makes extensive use of a set of techniques known as "statistical graphics," but it is not the same as statistical graphics in and of itself.

This report talks about the implementation of EDA on the dataset of the Daegu Metropolitan Rapid Transit System. The Daegu Metro is an underground rapid transport system that connects several parts of Deague, South Korea's third biggest metropolitan region. It now consists of three lines: two metro lines and one monorail line. Daegu Metro has a total track length of 81.2 kilometers and links 89 stations.

This analysis helped us get a deeper understanding on the train stations and the number of passengers that use the trains.

II. About the dataset

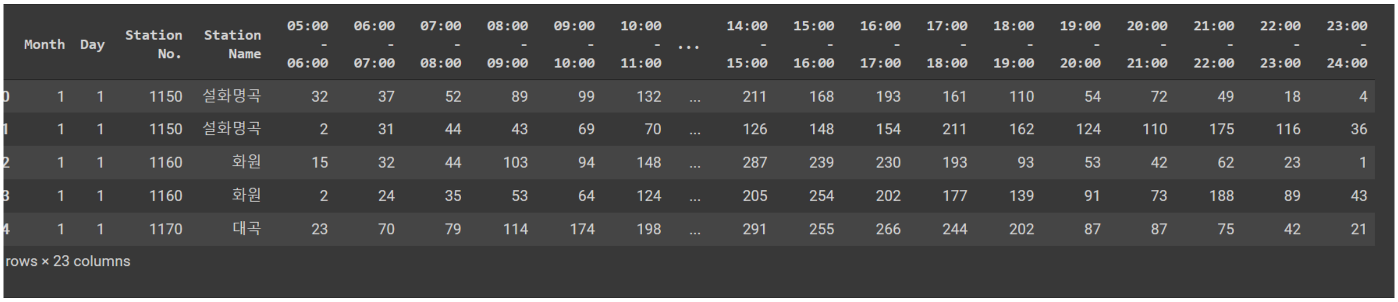
The dataset used in this project was obtained from the open data portal named *data.gov.kor.* The name of the dataset is 'Daegu Metropolitan Rapid Transit Corporation_Number of people getting on and off by station by day and hour by day'. The information in this dataset which is part of the 'Transportation and logistics - railroad' classification.

This dataset provides information about the number of passengers that get on the trains and get off the trains in each station every day on an hourly basis. The data that is recorded here times from January 1, 2022 to May 31, 2022 . Also, the number of passengers in each of the train stations was recorded on an hourly basis starting from 5 am in the morning to 12 am in the night. Every first alternating row consists of the number of passengers that are getting on the trains in that particular train station and the next alternating row contains the number of passengers that are getting off the trains in each of the train stations.

The number of rows in this dataset is 2,7482 and the number of columns is 25, out of which 2 were dropped for convenience while working on the project. The table contains the values which are the month, day, station number, station name and number of passengers on hourly basis.

III. Libraries and frameworks used in the project

Pandas: pandas is a data manipulation and analysis software package for the Python programming language. It includes data structures and methods for manipulating numerical tables and time series, in particular. It's open-source software with a three-clause BSD license.



	Month	Day	Station No.	Station Name	05:00 - 06:00	06:00 - 07:00	07:00 - 08:00	08:00 - 09:00	09:00 - 10:00	10:00 - 11:00	...	14:00 - 15:00	15:00 - 16:00	16:00 - 17:00	17:00 - 18:00	18:00 - 19:00	19:00 - 20:00	20:00 - 21:00	21:00 - 22:00	22:00 - 23:00	23:00 - 24:00
0	1	1	1150	설화명곡	32	37	52	89	99	132	...	211	168	193	161	110	54	72	49	18	4
1	1	1	1150	설화명곡	2	31	44	43	69	70	...	126	148	154	211	162	124	110	175	116	36
2	1	1	1160	화원	15	32	44	103	94	148	...	287	239	230	193	93	53	42	62	23	1
3	1	1	1160	화원	2	24	35	53	64	124	...	205	254	202	177	139	91	73	188	89	43
4	1	1	1170	대곡	23	70	79	114	174	198	...	291	255	266	244	202	87	87	75	42	21

rows × 23 columns

Fig.1.1

Matplotlib: Matplotlib is a Python and NumPy-based cross-platform data visualization and graphical charting package. As a result, it serves as a suitable open-source replacement for MATLAB. The APIs (Application Programming Interfaces) of matplotlib may also be used to incorporate charts in graphical user interfaces. In most cases, a Python matplotlib script can build a visual data plot with just a few lines of code.

Seaborn: Seaborn is a Python package that allows you to create statistical graphs. It's based on matplotlib and tightly interacts with pandas data structures. Seaborn aids in data exploration and comprehension. Its charting functions work with dataframes and arrays containing whole datasets, doing the necessary semantic mapping and statistical aggregation internally to build useful graphs. Its declarative, dataset-oriented API allows you to concentrate on the meaning of your charts rather than the mechanics of drawing them.

IV. Performing EDA on the dataset

Exploratory data analysis is a statistical way of evaluating data sets to summarize their essential properties, commonly utilizing statistical graphics and other data visualization techniques. It contains various steps which are as follows:

A. Cleaning of data

Correcting or deleting inaccurate, corrupted, improperly formatted, duplicate, or incomplete data from a dataset is known as data cleaning. Some of the steps that are included in the process of data cleaning are removing redundant data or unnecessary recordings, filter out unwanted outliers, make corrections in structural errors and handle data that are missing.

Data cleaning was performed on the given dataset. Two of the columns from the dataset were dropped as they were irrelevant. An attempt was made to find missing data and outliers and none of them were found, so further steps of the analysis were continued.

Fig.1.1 shows the first 5 rows of the dataset.

B. Construction of Heat Map

A heat map is a two-dimensional, color-based representation of information. Heat maps may aid in the visualization of basic or complicated data. Heat maps are utilized in a variety of fields, including military, marketing, and consumer behavior analysis.

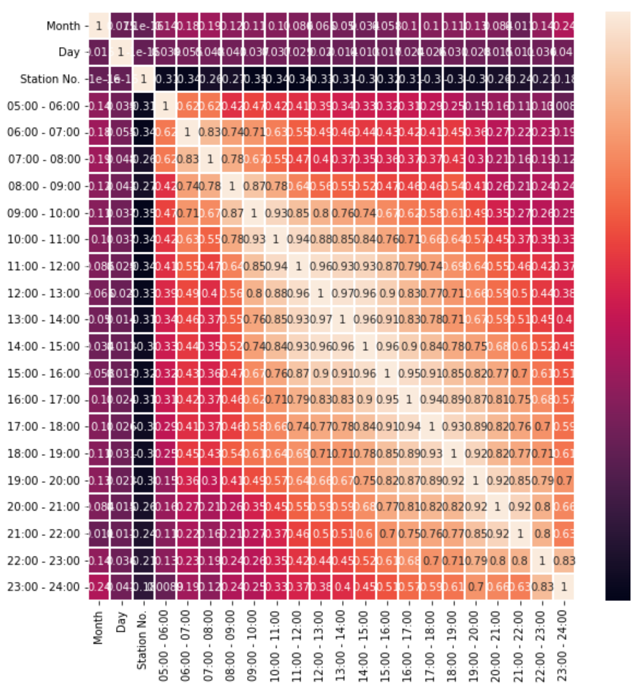


Fig.1.2

Fig.1.2 shows the heat map plotted for the given data.

C. Finding the average number of people that use the train stations all the five months together.

This calculation was done by taking the average of all the columns that contained hourly counts of the number of passengers in the train station. After the calculation was done a graph was plotted which contained time vs. the average number of people or passengers in the train station all together from all the months and train stations and was found to be as follows depicted in *Fig.1.5*.

D. Finding the average number of people that get on the train from each of the train stations every month.

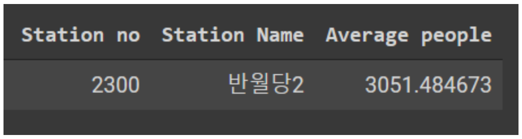
This calculation was done by taking the average of all the columns of the first alternative rows that contained hourly counts of the number of passengers in the train station, in a month wise fashion. After the calculation was done a graph was plotted which contained time vs. the average number of people or passengers that get on the train as follows depicted in *Fig.1.6*.

E. Finding the average number of people that get off the train from each of the train stations every month.

This calculation was done by taking the average of all the columns of the second alternative rows that contained hourly counts of the number of passengers in the train station, in a month wise fashion. After the calculation was done a graph was plotted which contained time vs. the average number of people or passengers that get on the train as follows depicted in *Fig.1.7*.

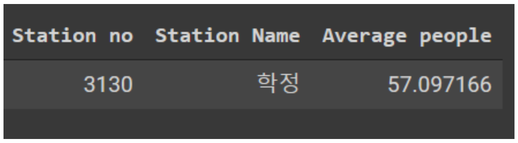
F. Finding the 'most crowded' and 'least crowded' train station in Daegu.

This calculation was done by taking the average of the count of passengers per day in all the stations. After which all the values for each station were taken separately and average was taken. Finally the maximum and minimum count of people was found which in turn gave the details of the 'most crowded' and 'least crowded' train stations.



Station no	Station Name	Average people
2300	반월당2	3051.484673

Fig.1.3 Most crowded station



Station no	Station Name	Average people
3130	학정	57.097166

Fig.1.4 Least crowded station

All these calculations and findings gave us a deeper understanding of the pattern of travel of all the passengers in the train stations of Daegu. The graphs made it easier for us to visualize the data and understand it better. The rush hour of train stations, the free hour of train stations and the most and least crowded stations were identified.

Graphs:

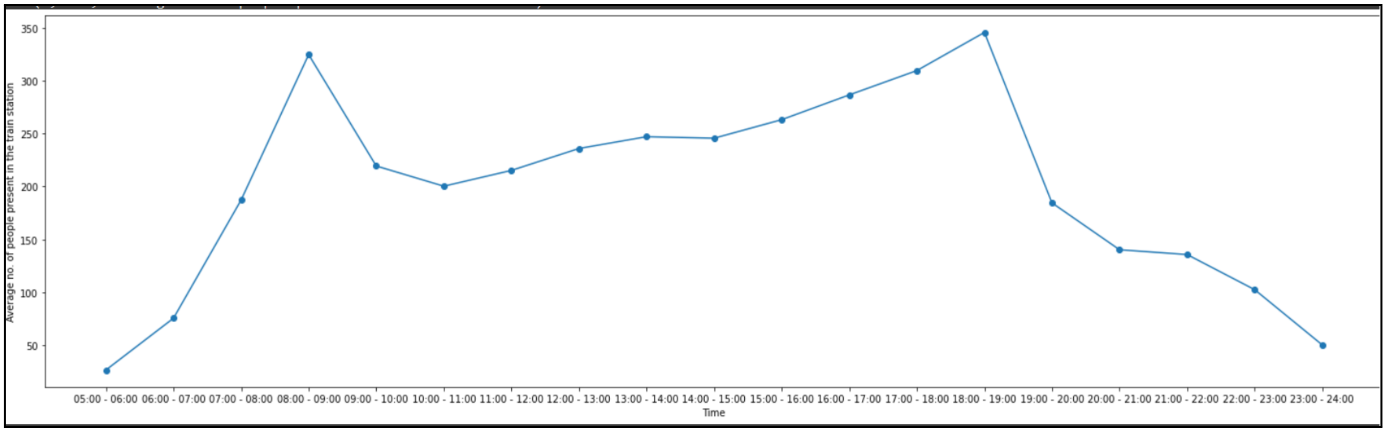


Fig.1.5

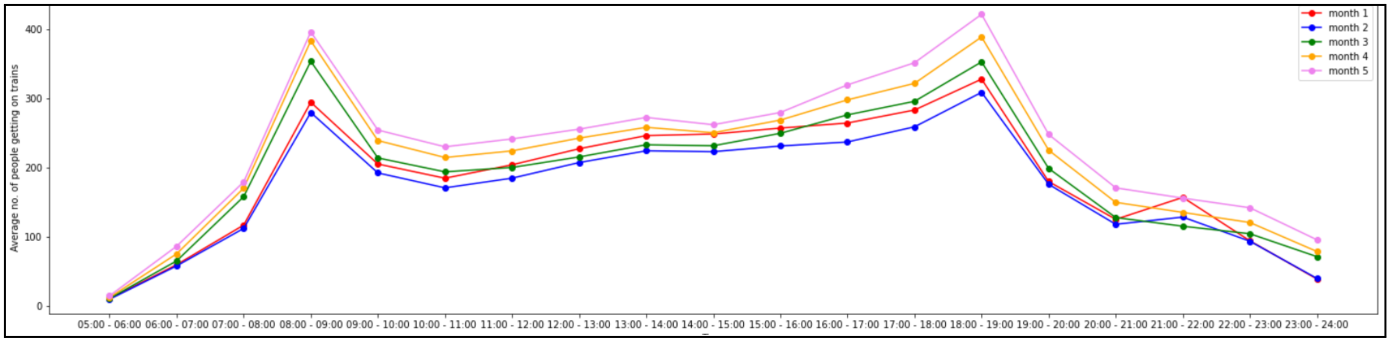


Fig.1.6

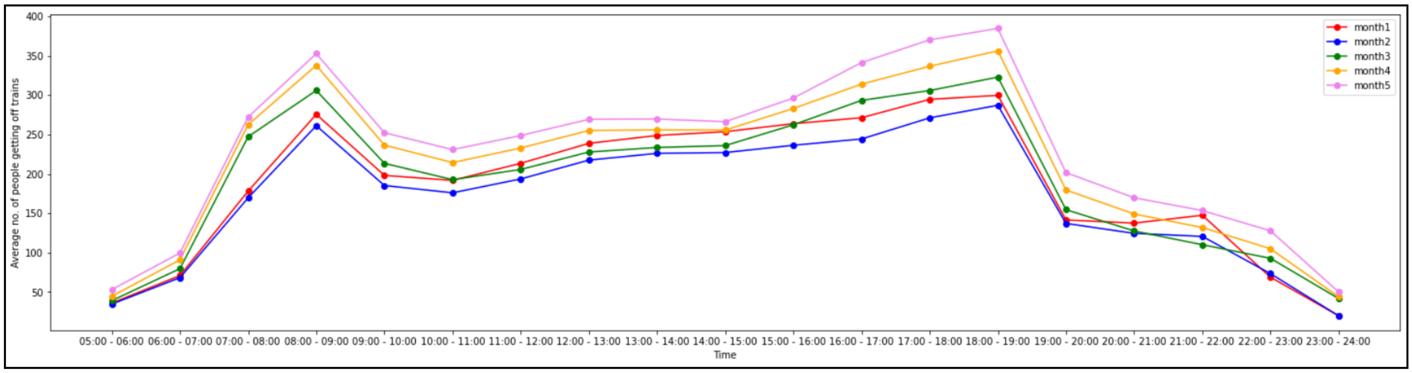


Fig.1.7

V. Conclusion

By looking at Graphs we can make various conclusions. From all the graphs given above we can identify the ‘Rush hours’ as 08:00 - 09:00 and 18:00 - 19:00. These are the peak hours at the train station when it is the most crowded. This is because these are the hours people are leaving for work or coming back from work.

The hours that are least crowded are 05:00 - 06:00 and 23:00 - 24:00.

The most crowded station in Daegu is identified to be 반월당*2 (Banwoldang2)* and the least crowded station is 학정 *(Hakjeong)*.

Another pattern that is very prominent in the above graphs is the rise of the number of people using the trains every month.

The performance of EDA on the given dataset gave us a deeper understanding of the pattern of travel of all the passengers in the train stations of Daegu. The graphs made it easier for us to visualize the data and understand it better. The rush hour of train stations, the free hour of train stations and the most and least crowded stations were identified.

References

[1] https://www.data.go.kr/en/data/15002503/fileData.do (dataset)

[2] https://colab.research.google.com/drive/1o-B7C0UuUiUSh4M5Ta7YMhcj7V8L52V9?usp=sharing (Google Colab link for the code written)

[3] https://www.itl.nist.gov/div898/handbook/eda/section1/eda11.htm

[4] https://seaborn.pydata.org/introduction.html

[5] https://www.activestate.com/resources/quick-reads/what-is-matplotlib-in-python-how-to-use-it-for-plotting/

[6] https://en.wikipedia.org/wiki/Pandas_(software)