Citibike New York City

Jan 2017-Dec 2019 statistical data analysis

**Group 16**

Wang, Su, s399wang

Mabbayad, Mabi, mmabbaya

Xue, Kira, w27xue

Jajcanin, Igor, ijajcani

Cebula, Daniel Adam, dacebula

Shi, Wenwen, w65shi

[Project Objective 5](#_heading=h.9u6iesynoytj)

[Introduction 5](#_heading=h.gjdgxs)

[Data Analysis 5](#_heading=h.30j0zll)

[Background 5](#_heading=h.1fob9te)

[Data Analysis 6](#_heading=h.3znysh7)

[Dataset Overview 6](#_heading=h.2et92p0)

[Data Preparation 6](#_heading=h.tyjcwt)

[Statistical Data Analysis 7](#_heading=h.3dy6vkm)

[Predictive Model 7](#_heading=h.1t3h5sf)

[Conclusion 7](#_heading=h.ob2xo1kjt3br)

# Project Objective

The primary goal of this project is to analyze data gathered for [New York City Citibike](https://www.citibikenyc.com/system-data) Program from Jan 2017 to Dec 2019.

Data collection is based on rental start and end time information i.e., duration of the trip for each bike. It does not collect other factors that could have been related to weather conditions, traffic congestions or trip path information.

The process includes data gathering, data cleaning, grouping and data analysis based on the bike ID (type), origin and destination stations, as well as the time analysis grouped by the day of the week and month. Analysis should also include distance travelled based on the geospatial information provided for start and end stations (latitude and longitude pairings). Distance traveled calculation is simplified through the Haversine formula, rather than calculation of the actual path travelled which is not present in the dataset.

The main aspects of python library and techniques that we used are as follow:

* Data cleaning/manipulation: pandas, numpy
* Visualization: matplotlib, seaborn
* Statistical Modelling: pymc3 and machine learning

# Introduction

## Data Analysis

Data analysis is a process of capturing, cleaning, transforming and modeling data in order to gain insight on the available dataset. The focus here is to understand the data, find patterns, discover relationships within a dataset to answer some questions one might have. Data analysis has a longer history, traditionally data analysis was primarily reactive and focused on reporting, over time, it has evolved from dealing with smaller, structured dataset to steam and process more complex and unstructured data sources. More and more organizations are relying on data analysis to make informed business decisions. In this report, you will see us using the data analysis techniques to analyze the flight delay dataset to answer questions such as which airport or which air carrier has the worst flight delay records.

# Background

Where do Citi Bikers ride? When do they ride? How far do they go? Which stations are most popular and can we make conclusions as to why? What time of the day, days of the week, month, season are most popular to ride the bike? What gender and age group bikes are most popular and used by? There are many more questions that arise for the growing popularity of city bike riders that is becoming worlds wide phenomena.

Bike ridership analysis can be very complex. It depends on multiple factors, obviously weather being one of the key factors, but also safety, road and trails infrastructures, bike qualities, etc.

We have also decided to utilize machine learning to predict the flight delay at the destination airport for the month of January in the upcoming years.

# Data Analysis

## Dataset Overview

There are multiple data sets that we used for our data analysis. They are downloaded from <https://www.citibikenyc.com/system-data>. This data is originally collected by Citibike New York City. These datasets contain all the bike rentals from Oct 2015 to Oct 2020.

Here are the attributes information:

TRIP\_DURATION': Duration of the trip (seconds)

'START\_TIME': Trip Start date and time

'STOP\_TIME': Trip End date and time

'START\_STATION\_ID': Trip Start Station ID

'START\_STATION\_NAME': Trip Start Station Name

'START\_STATION\_LATITUDE': Trip Start Station Latitude

'START\_STATION\_LONGITUDE': Trip Start Station Longitude

'STOP\_STATION\_ID': Trip End Station ID

'STOP\_STATION\_NAME': Trip End Station Name

'STOP\_STATION\_LATITUDE': Trip End Station Latitude

'STOP\_STATION\_LONGITUDE': Trip End Station Longitude

'BIKE\_ID': Bike ID

'USER\_TYPE': User\_type (Customer – 24 hrs or 3-day pass user, Subscriber – Annual Member)

'BIRTH\_YEAR': Year of Birth

'GENDER': Gender (0 – unknown, 1 – male, 2 – female)

According to the data source documentation staff test trips, system inspection trips and any trip below 60 seconds in length were removed from the dataset.

### Data Preparation

An exploratory analysis was done on a subset of the data to better understand the main characteristics of the dataset. Features available from data included geographical coordinates of the stations which the team used to calculate the distance of a bike ride from its starting location to the recorded ending location using the Haversine formula. The records that did not contain both Start Station ID and End Station ID were excluded from the analysis.

Several features with fixed number of possible values were converted using pandas.Categorical function to reduce the size of the on-memory data for analysis. This conversion trimmed the dataframe size from 4.9 GB to 1.5 GB. The resulting dataset was then used for all the analysis and prediction models.

### Statistical Data Analysis

# Predictive Model

## Time series (SARIMA) analysis:

One predictive model used was SARIMA time series analysis. Since the data provides timestamps for each trip, we wondered if we could use the timestamps in order to predict how many trips will be expected in the future.

### Time series data transformation

We extracted the data for the period of October 1st, 2020, to October 31st, 2020. As we were only concerned with the timestamps, we dropped all columns except the “starttime”, and added a column to indicate that 1 trip was initiated for each row of data. From there, we aggregated the table based on hour, and sum of trips initiated within that hour. This created a time series of number of Citibike trips initiated every hour. Based on this data, we can create a model to forecast, on an hourly basis, what’s the expected number of trips in the future.

### Time series visualization & differencing

First, we plotted the time series to visually inspect if there are any trends or seasonality. While it is hard to determine if a trend exists, the high fluctuation in trip counts imply that there is a seasonality at play here.

Next, we inspect the ACF and PACF plots.

ACF plot shows that there is a high autocorrelation at lags 1, 12 (negative autocorrelation), and 24. Viewing the PACF plot shows only partial autocorrelation that is not already accounted for in mutual correlations with other variables.

### Model creation

Based on the ACF and PACT plots, we will use a SARIMA model with order: ARIMA (0,1,1) x (0,1,1)24.

Once the model has been determined, we need to ensure the model is valid by analysing the residuals. The model validation plots show that:

* The residuals plot shows no trend in the residuals
* The histogram shows that residuals are normally distributed and centered on zero
* The normal Q-Q plot is somewhat linear
* The correlogram plot shows that some values are outside of the 95% confidence bounds

To forecast the data, we will use this model and forecast it against November data (October being the training data).

We can see that while the model is able to capture general trends (movement of trip counts), it is not very accurate at predicting the exact values. We may need to change the orders of differencing, or switch to a model which can account for exogenous factors, for even more accurate forecasting.

One factor to consider may be that during COVID situations, ridership trends may be drastically different month to month, based on lockdowns. Factors like that would not be considered in a SARIMA model.

# Correlation Analysis

## Ordinary least squares (OLS) method

### Data Preparation

# Conclusion