# Project


## Temirlan Karataev (tkaratae) - 031957685, Berkay Belly (bbelli) - 030537383


# Path 1: Bike traffic

## Analysis:

1 - You want to install sensors on the bridges to estimate overall traffic across all the bridges. But you only have enough budget to install sensors on three of the four bridges. Which bridges should you install the sensors on to get the best prediction of overall traffic?

Based on the data that was given, we calculated the percentage of traffic that goes through each bridge every single day, then we took the average of total sum of each bridge's percentage. In the end, our data indicated that the lowest percentage of traffic was flowing through Brooklyn Bridge, therefore, we recommend installing sensors to Manhattan, Williamsburg, Queensboro Bridges.

2- The city administration is cracking down on helmet laws, and wants to deploy police officers on days with high traffic to hand out citations. Can they use the next day's weather forecast to predict the number of bicyclists that day?

Approaching this question with logic, and without any data analysis, our answer would be yes, the next day's weather could help us predict the number of bicyclists that day. In order to understand whether our logical approach correlates with our data analysis, we used linear regression with 5-degree polynomials based on the independent variables that we had mentioned in our introduction. We will also use linear regression to find the MSE and R squared values at best-fit lambda based on our predicted model.

3- Can you use this data to predict whether it is raining based on the number of bicyclists on the bridges?

Again, logically, the number of bicyclists is dependent on an enormous scale of factors, due to this it would be very hard to find a concrete relationship between whether it is raining or not and the number of bikers out that day.

Regardless, we will still do our data analysis using two regression models. Our first method will be linear regression with a similar approach to what we had made in question 2. Our second model will be random forest regression with correlation of the MSE and R values.

Our data in Path 1, NYC_Bicycle_Counts_2016_Corrected.csv, gives information on bike traffic across a number of bridges in New York City. The data consists of several dependent and independent variables such as the temperature, the precipitation, the data and of course the traffic amongst each bridge and the total traffic.

Initially, we cleared out the data from any variables that might cause unexpected behaviors when we would convert the raw data to certain data types such as float or integers.

## Part 1

In [37]:
```
import numpy as np
import csv
import pandas as pd
import re

def sensors():
    # col_list = ["Date", "Day", "HighTemp", "LowTemp", "Precipitation", "Brooklyn", "Manhattan", "Will
iamsburg", "Queensboro", "Total"]
    data = pd.read_csv("NYC_Bicycle_Counts_2016_Corrected.csv", thousands=',')
    data.columns = data.columns.str.replace(' ', '')
    # total = pd.read_csv("NYC_Bicycle_Counts_2016_Corrected.csv", converters={"Total":int})
    total = data.Total.to_list()
    brooklyn = data.BrooklynBridge.to_list()
    manhattan = data.ManhattanBridge.to_list()
    williamsburg = data.WilliamsburgBridge.to_list()
    queensboro = data.QueensboroBridge.to_list()

    total = [float(i) for i in total]
    brooklyn = [float(i) for i in brooklyn]
    manhattan = [float(i) for i in manhattan]
    williamsburg = [float(i) for i in williamsburg]
    queensboro = [float(i) for i in queensboro]

    for i in range(len(total)):
        brooklyn[i] = brooklyn[i] / total[i]
        manhattan[i] = manhattan[i] / total[i]
        williamsburg[i] = williamsburg[i] / total[i]
        queensboro[i] = queensboro[i] / total[i]

    avg_brooklyn = sum(brooklyn) / len(brooklyn)
    avg_manhattan = sum(manhattan) / len(manhattan)
    avg_williamsburg = sum(williamsburg) / len(williamsburg)
    avg_queensboro = sum(queensboro) / len(queensboro)

    t_t = sum(total)
    t_b = sum(brooklyn) / sum(total)
    t_m = sum(manhattan) / sum(total)
    t_w = sum(williamsburg) / sum(total)
    t_q = sum(queensboro) / sum(total)

    print("Results:")

    print("Brooklyn Bridge:",avg_brooklyn)
    print("Manhattan Bridge:",avg_manhattan)
    print("Queensboro Bridge:",avg_queensboro)
    print("Williamsburg Bridge:",avg_williamsburg)
    # total = data.Total.to_list()


    # print(total)

if __name__ == '__main__':
    sensors()
```

```
Results:
Brooklyn Bridge: 0.16131907987022118
Manhattan Bridge: 0.27018858783146046
Queensboro Bridge: 0.23508279151302222
Williamsburg Bridge: 0.33340954078529605
```

So, based on the average value that we got from the code above, we decided to exclude the Brooklyn Bridge

## Part 2

In [49]:
```
import pandas as pd
import numpy as np

df = pd.read_csv("NYC_Bicycle_Counts_2016_Corrected.csv", thousands=',')
#print(df.head())


df.drop(['Date'], axis=1, inplace=True)
df.drop(['Day'], axis=1, inplace=True)
df.drop(['High Temp (°F)'], axis=1, inplace=True)
df.drop(['Low Temp (°F)'], axis=1, inplace=True)
df.drop(['Brooklyn Bridge'], axis=1, inplace=True)
df.drop(['Manhattan Bridge'], axis=1, inplace=True)
df.drop(['Williamsburg Bridge'], axis=1, inplace=True)
df.drop(['Queensboro Bridge'], axis=1, inplace=True)


df.Precipitation[df.Precipitation == 'T'] = 0
df.Precipitation[df.Precipitation == '0.47 (S)'] = 0.47

Y = df['Total'].values
Y = Y.astype('int')

X = df.drop(labels = ['Total'], axis = 1)

from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, random_state = 0)

from sklearn.linear_model import LinearRegression

linear_regress = LinearRegression(fit_intercept=True)
linear_regress.fit(X_train,Y_train)

prediction = linear_regress.predict(X_test)

from sklearn.metrics import mean_squared_error, r2_score

print('Mean squared error: %.2f' % mean_squared_error(Y_test, prediction))
print('Coefficient of determination: %.2f' % r2_score(Y_test, prediction))
```

```
Mean squared error: 21655031.51
Coefficient of determination: 0.03
```

```
/srv/conda/envs/notebook/lib/python3.6/site-packages/ipykernel_launcher.py:18: SettingWithCopyWarnin
g:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexin
g.html#returning-a-view-versus-a-copy
/srv/conda/envs/notebook/lib/python3.6/site-packages/ipykernel_launcher.py:19: SettingWithCopyWarnin
g:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexin
g.html#returning-a-view-versus-a-copy
```

Based on the values of MSE and R-Squared we can conclude that the administration can not rely on the traffic of bicycles for the next day because the data has a high number of outliers.

## Part 3

In [65]:
```
Y = df['Precipitation'].values
#Y = Y.astype('int')

X = df.drop(labels = ['Precipitation'], axis = 1)
X = X.astype('float')

X_train, X_test, Y_train, Y_test = train_test_split(X, Y, random_state = 0)

from sklearn.preprocessing import StandardScaler
standart = StandardScaler()
X_train = standart.fit_transform(X_train)
X_test = standart.transform(X_test)

from sklearn.ensemble import RandomForestRegressor

regr = RandomForestRegressor(n_estimators=20, random_state=0)
regr.fit(X_train, Y_train)
pred = regr.predict(X_test)

print('Mean squared error: %.2f'% mean_squared_error(Y_test, pred))

print('Coefficient of determination: %.2f'% r2_score(Y_test, pred))
```

```
Mean squared error: 0.06
Coefficient of determination: -1.94
```

Based on the results of the MSE and Coeff. of Determination we can conclude that the bike traffic is not dependent from precipitation rate and can not be used to predict the raining.