Analyzing NYC Subway Dataset

Intro to Data Science - Project 1
Emmanuelle JEAN
emmanuelle.jean@gmail.com

Section 0. References

references :
http://pandas.pydata.org/pandas-docs/stable/
www.udacity.com/wiki/cs101/unit1-python-reference
http://www.cyberciti.biz/faq/python-convert-string-to-int-functions/
http://docs.python.org/2/library/datetime.html#datetime.datetime.strptime
http://www.tutorialspoint.com/python/time_strptime.htm
https://bespokeblog.wordpress.com/2011/07/11/basic-data-plotting-with-matplotlib-part-3-histograms/
http://docs.scipy.org/doc/numpy/reference/index.html
http://www.creative-wisdom.com/teaching/WBI/parametric_test.shtml
http://www.itl.nist.gov/div898/handbook/pri/section2/pri24.htm
http://blog.yhathq.com/posts/facebook-ggplot-tutorial.html
http://discussions.udacity.com/c/nd002-2015-02-04/project-1

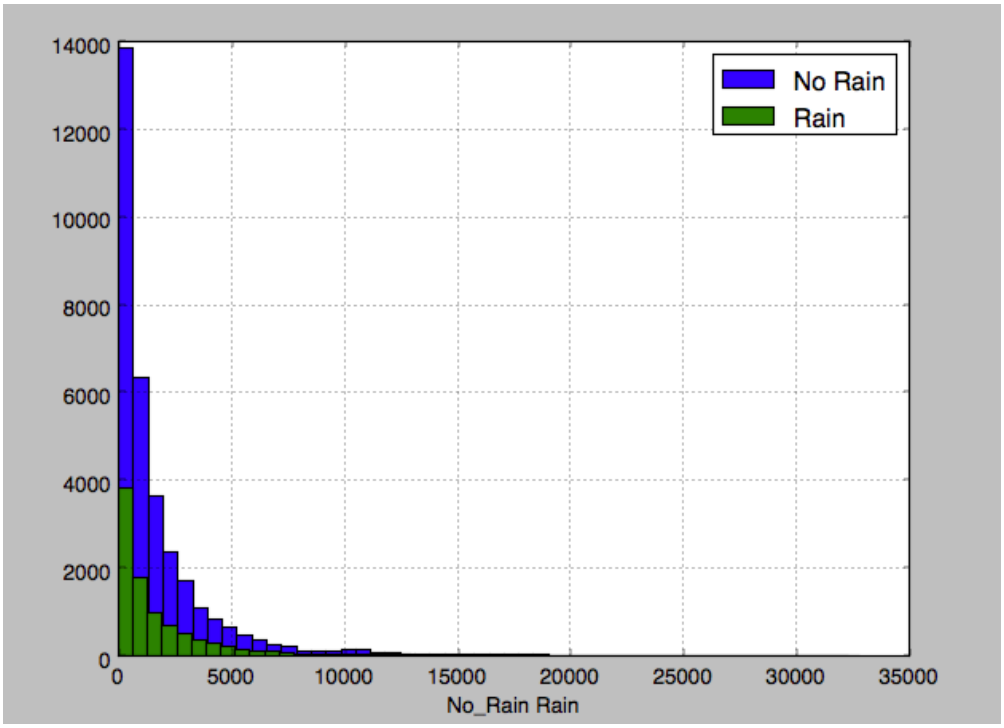
Shapiro Wilk Test :
http://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.shapiro.html
http://unsupervised-learning.com/shapiro-wilk-test-normality/

Section 1. Statistical test

From the first exploratory data analysis done in the problem set 3-1, the histogram shows that the data are not normalized and I concluded that we can not perform the Welch's t test.

Udacity provided an enhanced data source and before going further, I looked at the same histogram than in the problem set 3-1 to see if the histogram of frequencies of ENTRIESn_hours have the same shape than the one for the 1st data set.

```
In [*]: import numpy as np
        import pandas
        from ggplot import *
        import matplotlib.pyplot as plt


        turnstile_data = pandas.read_csv('turnstile_weather_v2.csv')


        def entries_histogram(turnstile_weather):

            fig = plt.figure()
            plt.xlabel("ENTRIESn_hourly")
            plt.ylabel("frequency")
             plt.title("frequency ofENTRIESn_hourly histogram")
            turnstile_weather['ENTRIESn_hourly'][turnstile_weather['rain']==0].hist(bins=50, label="No Rain"
            turnstile_weather['ENTRIESn_hourly'][turnstile_weather['rain']==1].hist(bins=50, label="Rain")
            plt.legend()

            return plt

        entries_histogram(turnstile_data).show()
```



The resulted histogram has the same shape than the histogram of the original dataset.

1.1 Let's compute the Mann-Whitney U-Test on the improved dataset:

- One-tail p value
- H_0 : the distribution of No Rainy day will have a higher ENTRIESn_hourly that the distribution rainy day.
- p critical value of 0.05

1.2 This Mann-Whitney U-Test is applicable because we do not need to make any assumption regarding the population and we have more than 20 values in each samples.

1.3 The Mann-Whitney U-Test results are :

```
import numpy as np
import scipy
import scipy.stats
import pandas

def mann_whitney_plus_means(turnstile_weather):

    print np.percentile(turnstile_weather['ENTRIESn_hourly'][turnstile_weather['rain']==0],50)
    print np.percentile(turnstile_weather['ENTRIESn_hourly'][turnstile_weather['rain']==1],50)

    without_rain_mean = np.mean(turnstile_weather['ENTRIESn_hourly'][turnstile_weather['rain']==0])
    with_rain_mean = np.mean(turnstile_weather['ENTRIESn_hourly'][turnstile_weather['rain']==1])
    (U,p) = scipy.stats.mannwhitneyu(turnstile_weather['ENTRIESn_hourly'][turnstile_weather['rain']==0],
                                     turnstile_weather['ENTRIESn_hourly'][turnstile_weather['rain']==1])
    return with_rain_mean, without_rain_mean, U, p # leave this line for the grader

turnstile_data = pandas.read_csv('turnstile_weather_v2.csv')
mann_whitney_plus_means(turnstile_data)
```

```
893.0
939.0
(2028.1960354720918, 1845.5394386644084, 153635120.5, 2.7410695712437496e-06)
```

Mann-Whitney U-Test	**U = 153635120.5**	**P = 5.482e-06**
	Non Rainy Day	*Rainy day*
Mean ENTRIESn_hourly	1845.539	2028.196
Median ENTRIESn_hourly	893.0	939.0

1.4 The p value is smaller than the p critical value so we can reject the null hypothesis and say that the non rainy day distribution is most likely to have a smaller ENTRIESn_hourly than the rain day distribution.
This conclusion is confirmed by the comparison of both the mean and the median of the ENTRIESn_hourly for non rainy day and for rain day samples.

****

*

****

However I really want to compute a Welch's t test and by performing the Shapiro-Wilk test, I can hopefully conclude that the data are drawn from a normal distribution.

Shapiro-Wilk test

In [12]:
```
import numpy as np
import pandas
import scipy


turnstile_data = pandas.read_csv('turnstile_weather_v2.csv')
rain_data = turnstile_data['ENTRIESn_hourly'][turnstile_data['rain']==1]
norain_data = turnstile_data['ENTRIESn_hourly'][turnstile_data['rain']==0]

"""Shapiro-Wilk test"""


print "shapiro-Wilk test for rain data", scipy.stats.shapiro(rain_data)
print "shapiro-Wilk test for no rain data", scipy.stats.shapiro(norain_data)
```

```
shapiro-Wilk test for rain data (0.5938820838928223, 0.0)
shapiro-Wilk test for no rain data (0.5956180691719055, 0.0)
```

The Shapiro - Wilk test is performed on samples of more than 5000 point and therefore is not relevant.
In the course "Intro to Data Science", lesson 3 Non parametric Test , the teacher says :

> *" Well, first off there's some math that says that we have enough data. That we have, you know, a large enough sample size we can actually use tests that assume normality. For example, the t test. Even when our data is not normal."*

From that quote, I assume I can run the Welch' two samples t-test on the NYC subway data, as we have more than 5000 data entries.

1.1 I use the Welch's t-test to analyze the NYC subway data.

- the one tail p value.
- H_0 : the mean of hourly entries during rain day is equal to the mean of hourly entries during no rain day.
- p critical value is 0.05.

1.2 This statistical dataset is applicable because we have large samples data. The Welch's t test assumption is :

- The sample data come from a normalised population : Given the number of data, I can assume that the population is normalised.

Welch's t test code :

```
: import numpy as np
import pandas
import scipy
turnstile_data = pandas.read_csv('turnstile_weather_v2.csv')
rain_data = turnstile_data['ENTRIESn_hourly'][turnstile_data['rain']==1]
norain_data = turnstile_data['ENTRIESn_hourly'][turnstile_data['rain']==0]

rain_mean = np.mean(rain_data)
norain_mean = np.mean(norain_data)
ttest = scipy.stats.ttest_ind(rain_data, norain_data, equal_var = False)
print rain_mean
print norain_mean
print ttest[1]/2
print ttest
```

```
2028.19603547
1845.53943866
2.52144137381
2.32070121582e-07
(5.0428827476194309, 4.6414024316324798e-07)
```

Welch's t test results :

Welch's T test	**t = 5.043**	**P = 2.321e-07**
	Non Rainy Day	*Rainy day*
Mean ENTRIESn_hourly	1845.539	2028.196

Welch's t test interpretations:
We can reject the null hypothesis. The mean of ENTRIESn_hourly is most likely to be different when it rains compared to the non rainy days.

Section 2. Linear Regression

Dataset : turnstile_weather_v2.csv

The code I used to compute the linear gradient, R^2 and the graph of residuals is in the file in the annexe “Intro To Data Science Problem 3-5.html”.

2.1 I used the Gradient descent approach to compute the coefficients theta and produce prediction for ENTRIESn_hourly in my regression model.

2.2 The features I use in my model are : hour, day_week, fog, precipi, pressurei, rain, tempi, wspdi, meanprecipi, meanpressurei, meantempi and meanwspdi .
I used the dummy variables related to UNIT.

2.3 My reasoning was to use all the numerical features hour, day_week, fog, precipi, pressurei, rain, tempi, wspdi, meanprecipi, meanpressurei, meantempi. The more features, the more precise my model will be and the R^2 will be close to one.
I had a R^2 of 0.476.
I reduced the number of features and experimented with different alpha and number of iteration and computed the R^2 value.

features	alpha	iteration	R^2
'hour','day_week','fog','precipi','pressurei','rain','tempi','wspdi','meanprecipi','meanpressurei','meantempi','meanwspdi'	0.1 0.5 0.5 0.1 0.8	75 75 150 1000 75	0.476 0.476 0.476 -1.405
'rain', 'precipi', 'Hour', 'hour','day_week','fog','precipi','pressurei','rain','tempi','wspdi'	0.1 0.8	1000 75	0.471 0.470
'hour','day_week','fog','precipi','rain'	0.1 0.8	1000 75	0.470 0.470
'day_week','rain'	0.8 0.1	75 1000	0.385 0.385

From my different tests, it appears that R^2 decreases a little with less features. I find the the R^2 far from 1 anyway. A higher value of alpha, introduce a risk of missing the lowest point.

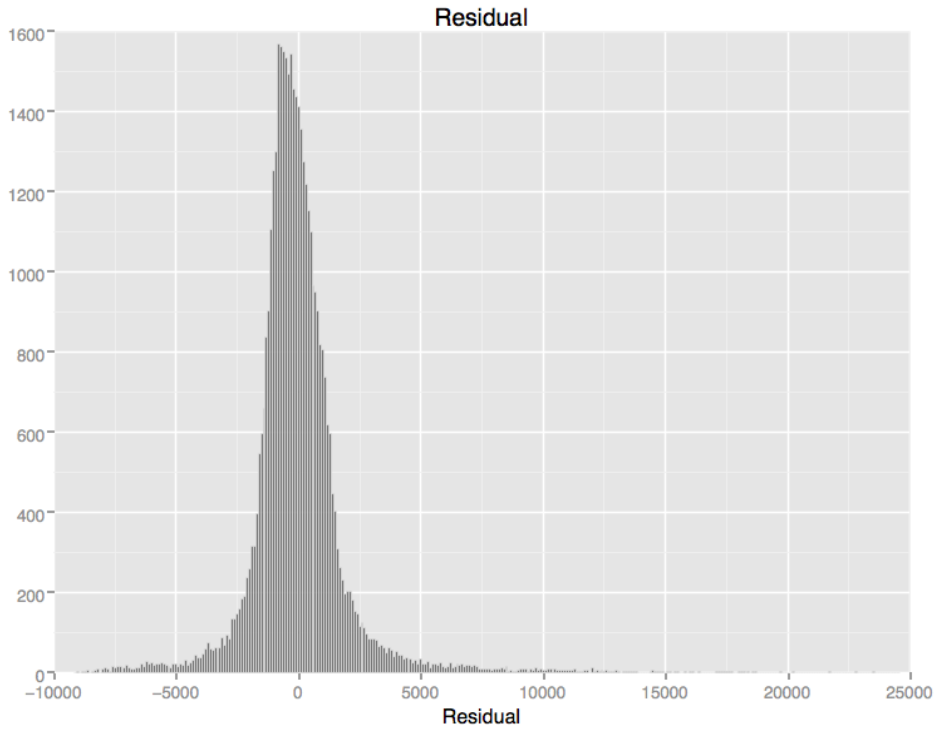
2.4 The coefficients for the 12 features are :

features	coefficients	features	coefficients
hour	7.284e+02	tempi	3.4247e+02
day_week	-3.235e+02	wspdi	6.598e+01
fog	-1.617e+01	meanprecipi	1.030e+02
precipi	-1.314e+02	meanpressuri,	5.188e+01
pressurei	-1.098e+02	meantempi	-3.979e+02
rain	-4.696e+00	meanwspdi	-1.119e+02

The coefficients are the twelve first coefficients of the *theta_gradient_descent* data-frame computed in the *predictions* procedure. They correspond to the first twelve columns of the features dataframe, the rest being the UNIT features using dummies data.

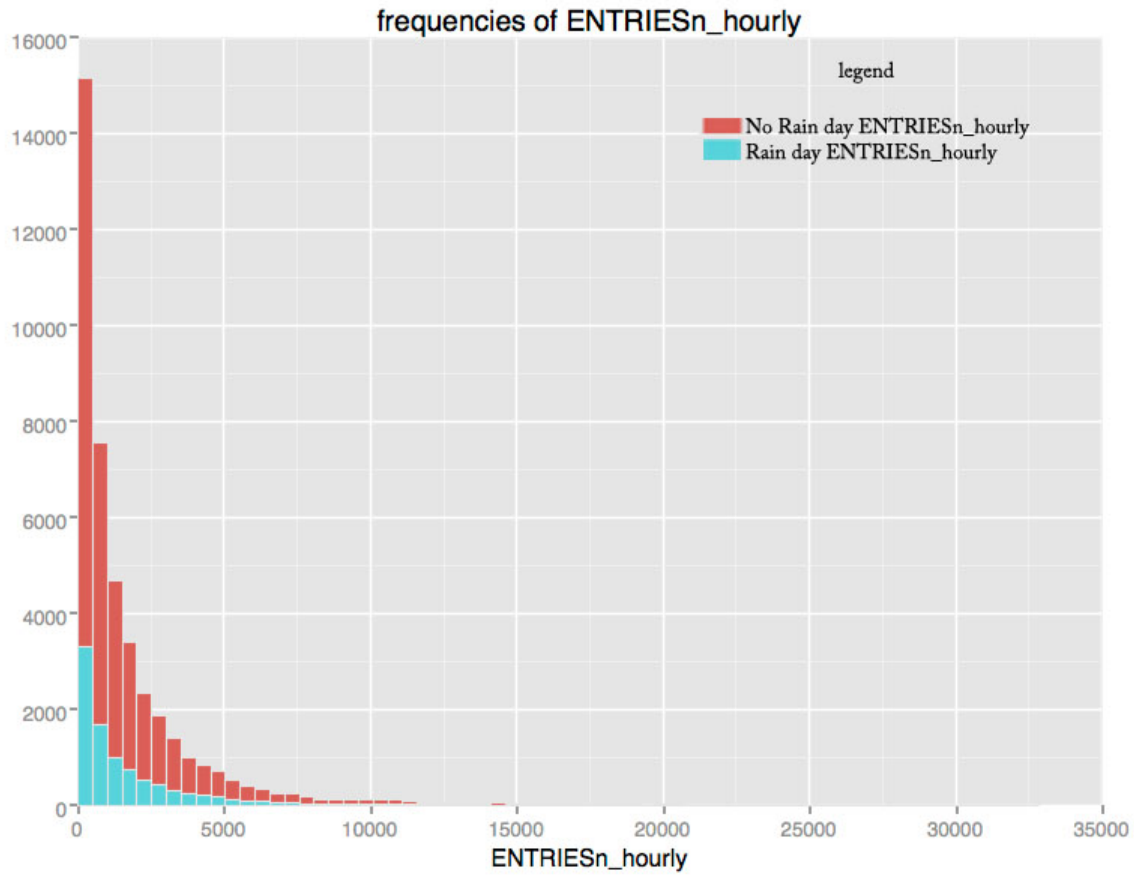
2.5 My model R^2 is 0.476.

2.6 My calculated coefficient of determination is pretty far from one.
But the plot of the residual data is normalized and show a higher frequencies around 0. It indicates that our model is pretty accurate.


Residual
1600
1400
1200
1000
800
600
400
200
0
−10000
−5000
0
5000
10000
15000
20000
25000
Residual

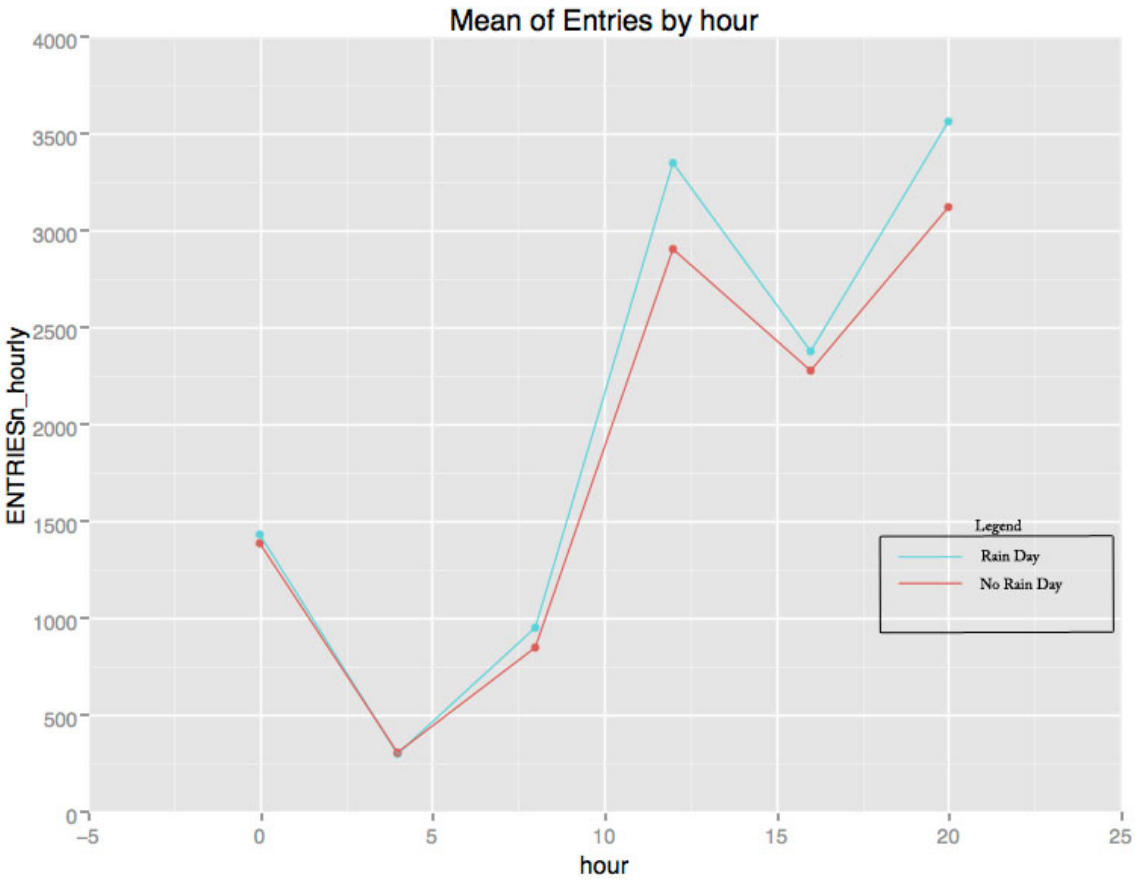
Section 3. Visualization

3.1 Histogram representing the frequencies of ENTRIESn_hourly for rainy days (blue) and non rainy days (red).




```
from pandas import *
from ggplot import *
from datetime import *


"""Histogram of the frequencies of ENTRIESn_hourly """
def plot_weather_data(turnstile_weather):

    pandas.options.mode.chained_assignment = None

    plot= ggplot( aes(x="ENTRIESn_hourly", fill ="rain" ), data = turnstile_weather ) +\
    geom_histogram(binwidth =500)+\
    ggtitle("frequencies of ENTRIESn_hourly")


    return plot

turnstile_weather = pandas.read_csv('turnstile_weather_v2.csv')
plot_weather_data(turnstile_weather)
```

3.2 Mean of ENTRIESn by hour, for rain and no rain days



Code

```
from pandas import *
from ggplot import *
from datetime import *

def plot_weather_data(turnstile_weather):
    turnstile_weather['DateDatetime'] = pandas.to_datetime(turnstile_weather['DATEn'])
    #turnstile_weather['weekDay'] = turnstile_weather['DateDatetime'].map(lambda x: x.weekday())
    #******Visualisation of the mean of Entries by hour for rain and no rain day for the business day
    rain_day_group = turnstile_weather.groupby(["rain","hour"], as_index=False)
    mean_rain_day= rain_day_group["rain","hour", "ENTRIESn_hourly"].aggregate(np.mean)
    plot= ggplot(mean_rain_day, aes(y='ENTRIESn_hourly', x="hour", color='rain')) + geom_point() +geom_line()+\
    ggtitle("Mean of Entries by hour")
    return plot

turnstile_weather = pandas.read_csv('turnstile_weather_v2.csv')
plot_weather_data(turnstile_weather)
```

On this visualization, we see how the average of entries varies during the day by hour block, for non rainy day (red) and rainy day (blue).

We can see that there is more people, in average, riding the subway when it rains, between 8:00 am and 8:00 pm. The numbers of riders entering the subway between 12:00 am and 4:00 am does not seems as much influenced by rain than riders using the subway after 8:00 am.

Section 4. Conclusion

4.1 From my analysis of the NYC subway improved dataset and interpretation of the computed information, more people ride the NYC subway when it rains.

4.2
First, I compared the average of riders during rainy days to the average of riders during days without rain. There are more people who take the subway on rainy days. I compared, as well, the median of the entries during rainy days to the median of the entries of non rainy days. The comparison led to the same conclusion.

But I needed to make sure that this conclusion is representative for every months of the year. I had to verify that the data is representative to the whole population of entries in the NYC Subway.
I looked at the histogram of the two sets to see if they are normally distributed. The histogram did not show a Gaussian curve. The only test I could use was the Mann-Whitney U-Test. The Mann-Whitney U-Test does not need the assumption that the data follows a normal distribution. From the calculated Mann-Whitney U and p values, I interpreted that indeed, the sets of data differs and we have a difference of number of NYC Subway riders when it rains.

My first analysis of the shape of the distribution was only graphical, so I computed the Shapiro-Wilk test to really see if the population where the data has been drawn from is normal. The obtained value of p was zero, and with some research on internet, I understood that this test is meaningful for samples with less than 5000 points. On the other hand, some math tell us that if we have enough data, we can concluded that we have a normal distribution.
This assumption allows me to compute the Welch's test.

The Welch's test p value is smaller than my chosen p critical value so I can reject the null hypothesis and say that the mean of entries in the NYC subway is more likely to be different when it rains and when it does not rain.

I chose the linear gradient as the prediction model, and use 12 features to compute the prediction values. Even if the R^2 value of 0.476 is far from one. The residual data plot is normal and indicates that my prediction model is accurate.

Section 5. Reflection

5.1 shortcoming of the method of my analysis :

Dataset :

- The number of datapoint for rainy days is very low compared to to the number of data points for non rainy days.

- The differences of results between the 2 datasets provided shows the importance of cleaning up our values before computing any tests.

- I am wondering if we would have the same results with data from a winter month. That would validate the assumption I made : given the number of data, we can consider the data being normalised.
-
- I have been a NYC subway commuter and my first reflexion was that ridership is not influenced by rain. One takes the NYC subway because it is the most convenient compared to the bus, the car, bike or feet. The analysis proved me that I was wrong in my way of thinking.

Analysis :

- I have been able to justify using two tests and fortunately their results were not in opposition, but I realize that the choice of the test and the choice of assumptions are very important and but also one can easily argue the choice made in my analysis.

- In my linear regression, I did not included any non numerical features such as UNIT or station.

- I only used the methods presented in the Intro to Data Science but I am aware that more methods are available.