HW7_Assignment1_citibike_mini_project

yt1369[1] and Cheng Ma[2]

[1]New York University (NYU)
[2]Affiliation not available

November 8, 2017

Abstract

The idea of this project comes from the curiosity of the generation distribution of Citi-bike user, we divide the data into 2 categories, pre-90s and post-90s to see whether younger generation less likely to use Citi-bike to commute, and what is their characters in trip duration, trip time and route pattern. Through the analysis, we reject the null hypothesis that the proportion of pre-90s biking on weekends is the same or higher than the proportion of post-90s biking on weekends(alpha=0.05), hence, we are reasonable to conclude that post-90s are less likely to use Citi-bike for commuting than pre-90s.

Introduction

Citi-bike is the largest bike share program in the US, it gives citizens a healthy, interesting and affordable way to get around town. In this project, we look into the generation because we are curious about whether younger people are less likely to use bike for commuting, generally, public transportation and bicycle are cheap transit methods, either workers or students would love to use, however, since students may have another choice: school bus, it may lead to students(post-90s) less likely to use Citi-bike for commuting. In the end, If we can learn Citi-bike user pattern differences between generations, this project can gives suggestions for location of Citi-bike station, for instance, establish more Citi-bike stations near office building or school to fulfill the larger demand of specific people.

Data

We used the data from citi-bike and we selected datasets from July and December 2016 since we assume that the biking pattern would be slight different from summer to winter and looking into 2 seasons would lead us to the reliable conclusion. As the figures show, the counts of July is greatly higher than December. In the data processing, we convert the "starttime" column which is in string format into "date" using the function "pd.to_datetime", thus, we can learn users counts in specific weekdays.

According to my peer Heci's suggestion, he thinks it's better to dig more into weekdays' rush hours, after careful consideration, I didn't select peak hour to analyze this problem, the reason is the working hour is basically same for both workers(more pre-90s) and students(more post-90s). In future projects, if it is necessary, I would definitely take rush hours into consideration.

Methodology

Along with the suggestion from my peer Heci, we used z-test. Z test is useful in this project to test H0, because the data for pre-90s and post-90s samples both from the same population, and it has one variable as usage quantity of the bike, and two categories(pre and post-90s)). Also, we can easily tell the sample is way over 30.

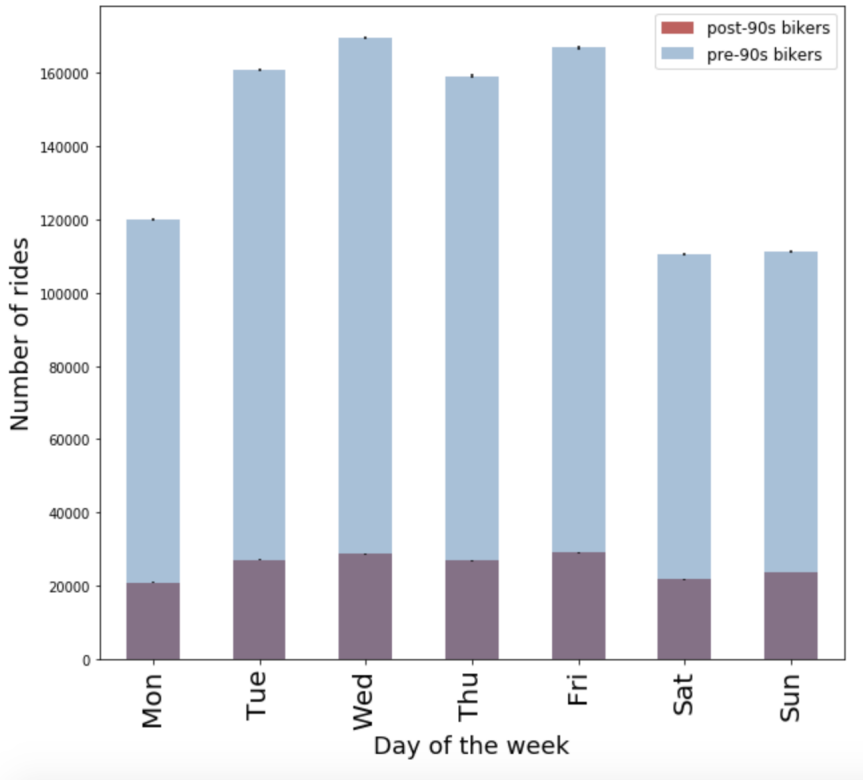

Figure 1: **Distribution of Citi-bike bikers(pre-90s and post-90s)in July 2016, absolute counts, with statistical errors**

Conclusions

The results of analysis match the idea that the post-90s are less likely than pre-90s to choose biking for commuting. And the conclusions of July and December are the same so that our work is robust to sensonality.

For testing its significance, we calculate the Z-statistics as 30.98 in July and 28.24 in December, we got the corresponding p-value < 0.0002, which is smaller than my chosen α=0.05. So we can reject the Null Hypothesis and my conclusion is statistically significant (by a lot!)

Strength: We got a more reliable conclusion by analyzing two significantly different months in one year.

Weakness: We did not pick gender as a critical character given that different genders make different choices (proved in FBB's instruction). If the majority of post-90s users are females, our work is a simple duplication of FBB's and lose its sense.

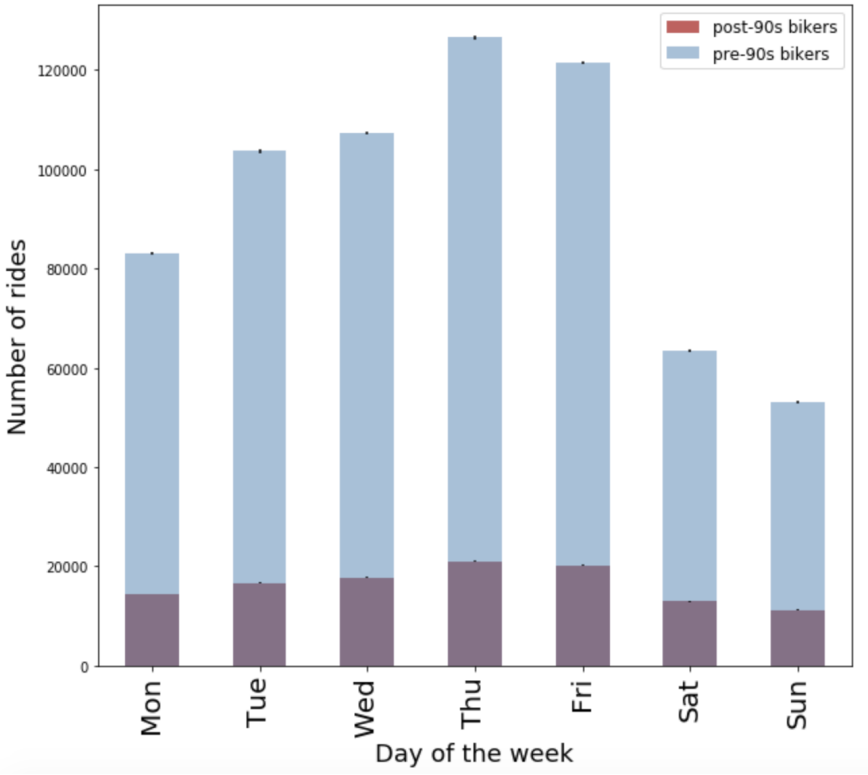

Figure 2: **Distribution of Citi-bike bikers(pre-90s and post-90s)in December 2016, absolute counts, with statistical errors**

```
def getp(p0, p1, n0, n1):
    p = (p0 * n0 + p1 * n1) / (n0 + n1)
    return p

def getse(p, n0, n1):
    se = np.sqrt(p * (1 - p) * (1.0 / n0 + 1.0 / n1)) #standard error
    return se

def getzscore(p0, p1, s):
    zscore = (p0 - p1) / s
    return zscore

# calculations
p = getp(weekend_po, weekend_pr, norm_po, norm_pr)
SE = getse(p, norm_po, norm_pr)
z = getzscore(weekend_po, weekend_pr, SE)

print ("The z statistics is %.10f"%z)
```

```
The z statistics is 30.9816135711
```

Figure 3: **Z statistic for July 2016**

```
sp_stdev_12 = getse(getp(weekend_po12, weekend_pr12, norm_po_12, norm_pr_12), norm_po_12, norm_pr_12)
# print (sp_stdev_mw)
z12 = getzscore(weekend_po12, weekend_pr12, sp_stdev_12)
print ("The z statistics is %.10f"%z12)
```

```
The z statistics is 28.2463866580
```

Figure 4: **Z statistic for December 2016, z statistic is larger in the summer**

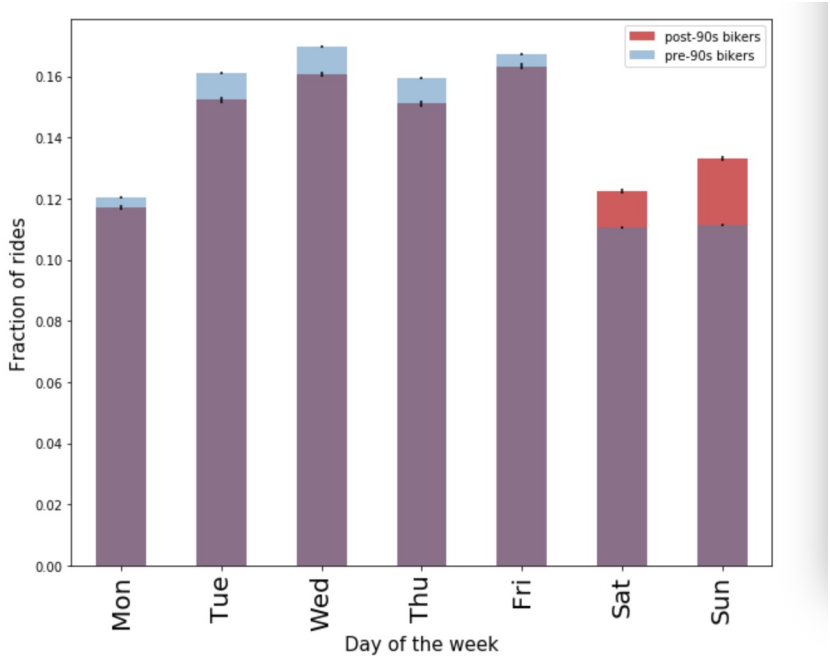

Figure 5: **Distribution of Citi-bike bikers(pre-90s and post-90s)in July 2016, normalized, with statistical errors**

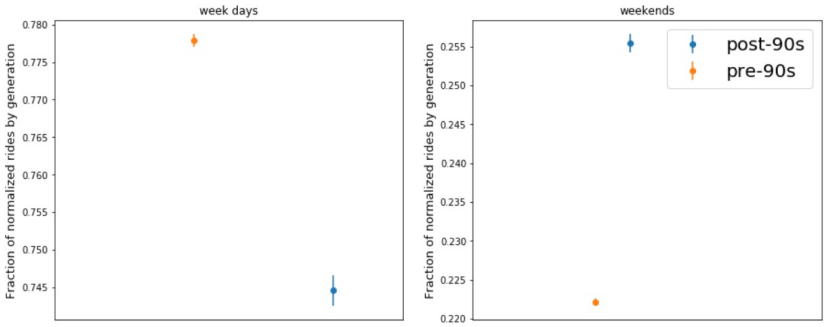

Figure 6: **Distribution of Citi-bike bikers(pre-90s and post-90s)in July 2016, normalized, with statistical errors**

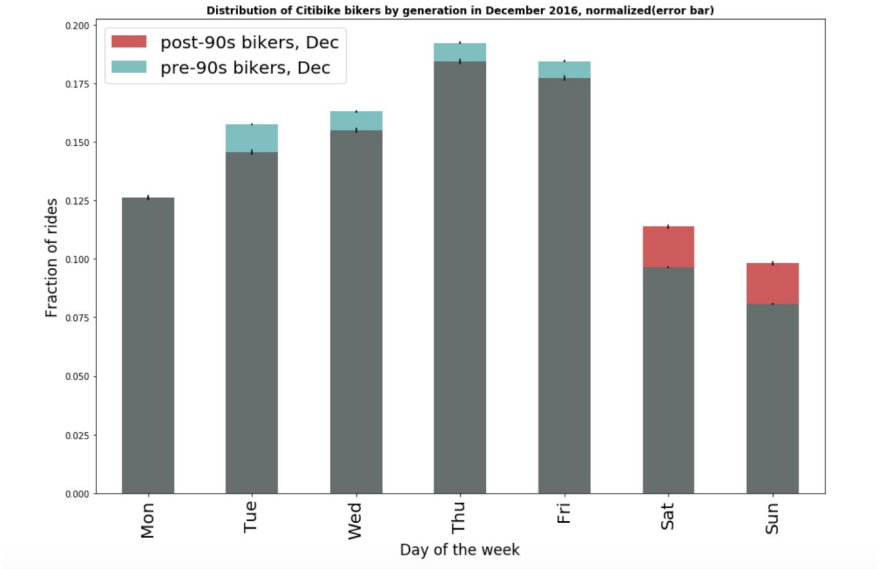

Figure 7: **Distribution of Citi-bike bikers(pre-90s and post-90s)in December 2016, normalized, with statistical errors**