DataScope

December 13, 2016

Opening a new business is no small task. It is usually affected by a lot of factors like the location of the business, it's connectivity with the public transport, the social and economic capabilities of that location, cost etc. In this notebook, I try to explore the question of opening a new business with the help of the Chicago 'L' transit system. In my analysis and suggestions, I've tried to incorporate different aspects that would help us decide what is the best location for opening up a new business.

```
In [143]: import pandas as pd
          import numpy as np
          import matplotlib.pyplot as plt
          import seaborn as sns
          %matplotlib inline
          df = pd.read_csv('/Users/Baid/Downloads/CTA_-_Ridership_-__L__Station_Ent
          df.head()
```

After reading in the ~30MB csv file of the 'L', I wanted to check for the basic discrepancies in the data. Since the dataset had been loaded successfully, I wanted to check if there are any NaN values in the dataset.

```
In [207]: df.isnull().sum()

Out[207]: station_id     0
          stationname    0
          date           0
          daytype        0
          rides          0
          dtype: int64
```

And there were none! That's a great dataset!

To check how many unique stations and station_id were present I ran the below code snippet:

```
In [208]: print 'Number of Unique Stations: '+str(len(np.unique(df['stationname']))

Number of Unique Stations: 147
```

```
In [209]: print 'Number of Unique Station ID: '+str(len(np.unique(df['station_id'])
Number of Unique Station ID: 146
```

Now that's an interesting thing to notice! The number of station names is 1 more than the number of station id. I wanted to know what was the reason behind it but that is left for another day!

Next, I wanted to use the date parameter in the dataset to have a better understanding of rides vary throughtout the year, months and also seasons.

```
In [223]: df['date'] = (pd.to_datetime(df['date']))
```

Although it took a couple of minutes, I converted my 'date' parameter to 'Datetime' object because it just became very esy to deal with months and years. In the next cell, I divided the date into months and years and my goal was to analyse the rides in each year and season.

```
In [224]: df['Month'] = df['date'].map(lambda x: x.month)
          df['Year'] = df['date'].map(lambda x: x.year)
```

```
In [225]: df.head()
```

```
Out[225]:    station_id        stationname       date daytype  rides  Month  Year
          0       40010  Austin-Forest Park 2001-01-01       U    290      1  2001
          1       40020         Harlem-Lake 2001-01-01       U    633      1  2001
          2       40030        Pulaski-Lake 2001-01-01       U    483      1  2001
          3       40040       Quincy/Wells 2001-01-01       U    374      1  2001
          4       40050               Davis 2001-01-01       U    804      1  2001
```

Another pre-processing step is to divide the months into seasons. I divided the months into 4 seasons (Summer, Fall, Winter and Spring) based on some temperature information from Wikipedia about Chicago.

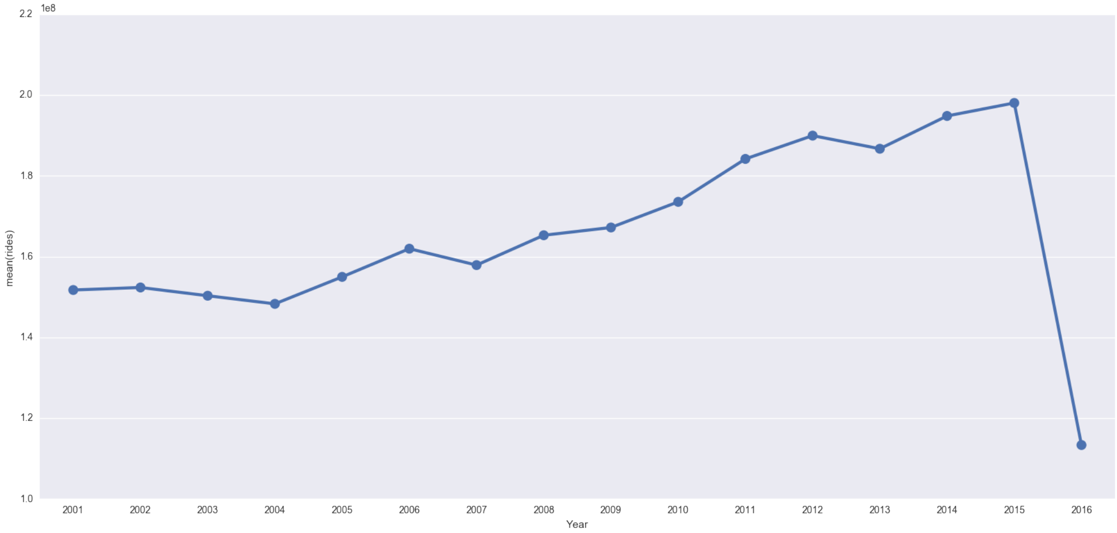
```
In [226]: def season(i):
              if i>=3 and i<=5:
                  return 'Spring'
              elif i>=6 and i<=8:
                  return 'Summer'
              elif i>8 and i<=11:
                  return 'Fall'
              else:
                  return 'Winter'
```

```
In [227]: df['Season'] = df['Month'].apply(season)
```

After all the pre-processing, let's begin with the analyses. Lets begin by seeing if there is actually an increase in the number of riders through the years. If this basic test fails, there wouldn't be any concrete analysis possible. I start off by grouping my data by year and see how the rides are affected throughout the years.

```
In [228]: grouped = df.groupby('Year')
          temp = grouped['rides'].sum().reset_index()
```

```
In [233]: plt.rcParams['figure.figsize'] = (20.0, 9.0)
          linePlot = sns.pointplot(x="Year", y="rides", data=temp)
```



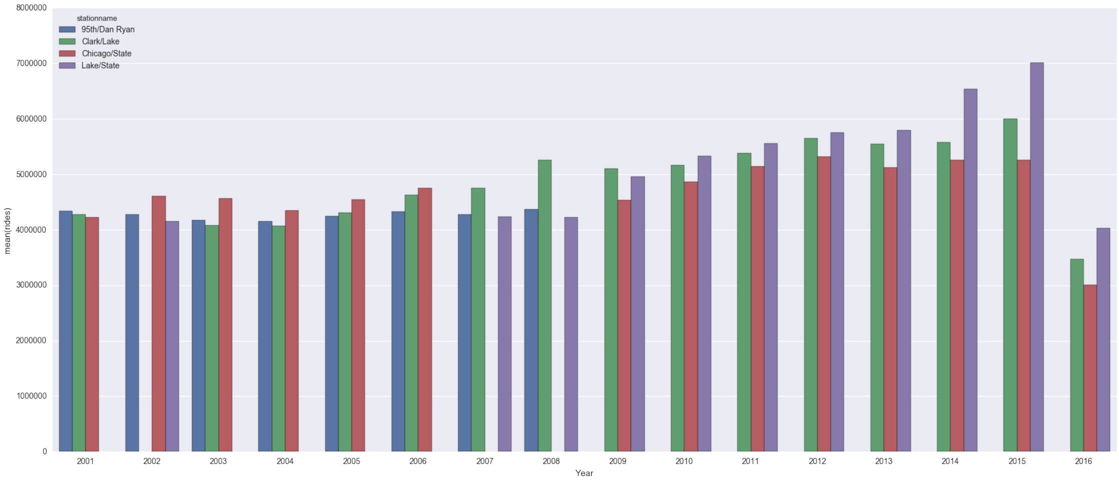
As we can see that there is a constant increasing trend in the plot, it is a good idea to go ahead with opening the business in a new location that is accessible by the 'L'. Now the sudden dip from 2015 to 2016 is because of the fact that 2016's data is only until July and therefore is incomplete.

The next idea is to see which stations were the top stations each year, based on the count of rides. Below is an analysis of the idea:

```
In [234]: grouped = df.groupby(['Year', 'stationname'])
          temp = grouped['rides'].sum().reset_index()
          temp = temp.sort(['Year', 'rides'], ascending=[1, 0])
          years = np.unique(temp['Year'])
          topStations = pd.DataFrame()
          for year in years:
              topStations = topStations.append(temp[(temp['Year']==year)][:3])
          topStations.head()
```

Above code shows us the stations that are sorted by the number of rides for each year. A good idea is to plot the top 3 stations for each year to see if there are stations that appear consistently.

```
In [245]: plt.rcParams['figure.figsize'] = (24.0, 10.0)
          bars = sns.barplot(x="Year", y="rides", hue="stationname", data=topStatic
```

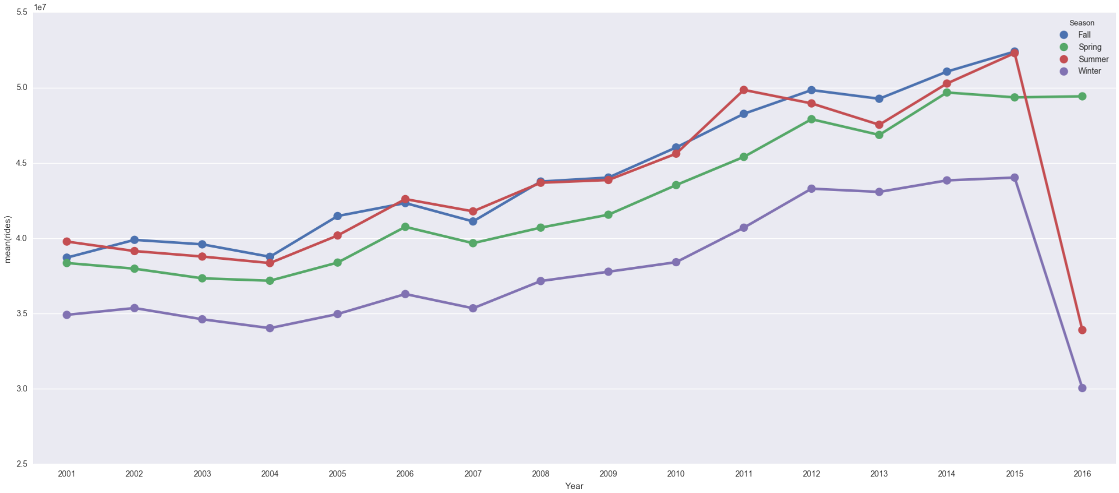


Conclusions that can be drawn from the above plot: 1. 95th/Dan Ryan wasn't in the top three places after 2008 and it wouldn't be wise to open the business in that neighbourhood. 2. Clark/Lake, Chicago/State and Lake/State seem to be the best neighbourhoods to open the new venture. 3. There is a nice increase in the number of rides which again strengthens our claim from plot 1.

The couple of plots above build a good foundation of what to expect. Next I wanted to see how the rides are affected in each season throughout the years. This analysis can let us draw conclusions if the seasons affect business and moreover if the business that needs to be open has some dependency on seasons, those questions could be answered here.

```
In [246]: grouped = df.groupby(['Season','Year'])
          temp = grouped['rides'].sum().reset_index()
```

```
In [248]: point = sns.pointplot(x="Year", y='rides', hue = 'Season', data=temp)
```



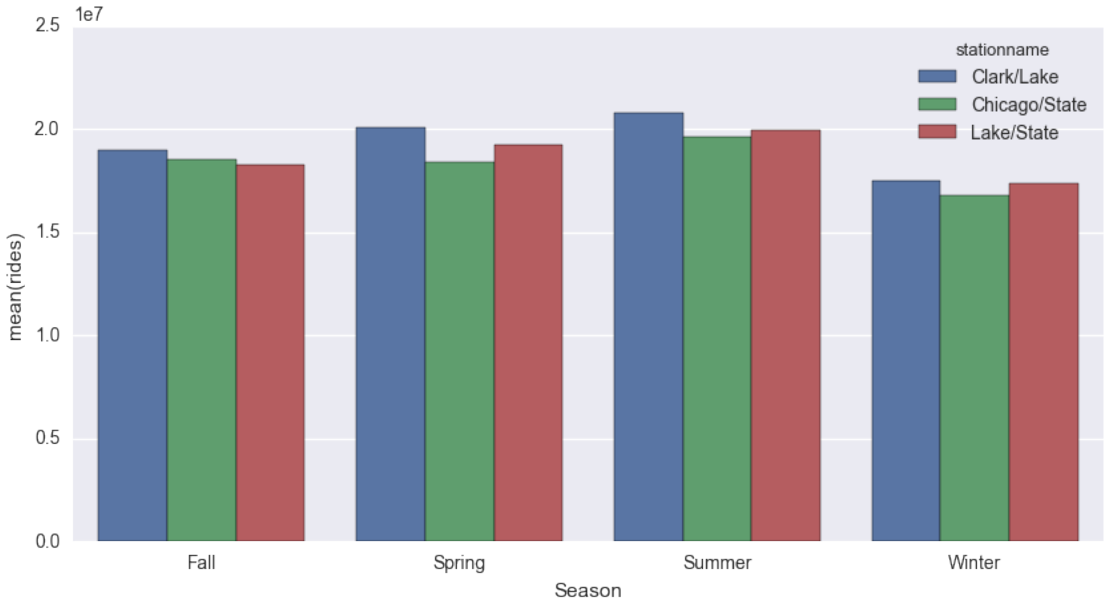
A couple of points that could be under consideration for opening up the new business that can be drawn from the plot are: 1. It is surprising that lesser people use the 'L' in winters. I had assumed that in order to avoid snow, a lot more people would be using the 'L', but that claim is clearly invalid. 2. It would be interseting to see what stations are mostly visited during each season and decision can be made from there.

```
In [249]: grouped = df.groupby(['Season','stationname'])
          temp = grouped['rides'].sum().reset_index()
          temp = temp.sort( ['rides'], ascending=[0])
          s = np.unique(temp['Season'])
          topStations = pd.DataFrame()
          for i in s:
              topStations = topStations.append(temp[(temp['Season']==i)][:3])
```

```
In [253]: topStations.head()
```

```
Out[253]:       Season    stationname     rides
          43      Fall      Clark/Lake  18993920
          38      Fall  Chicago/State  18561613
          90      Fall      Lake/State  18299605
          189   Spring      Clark/Lake  20071793
          236   Spring      Lake/State  19258507
```

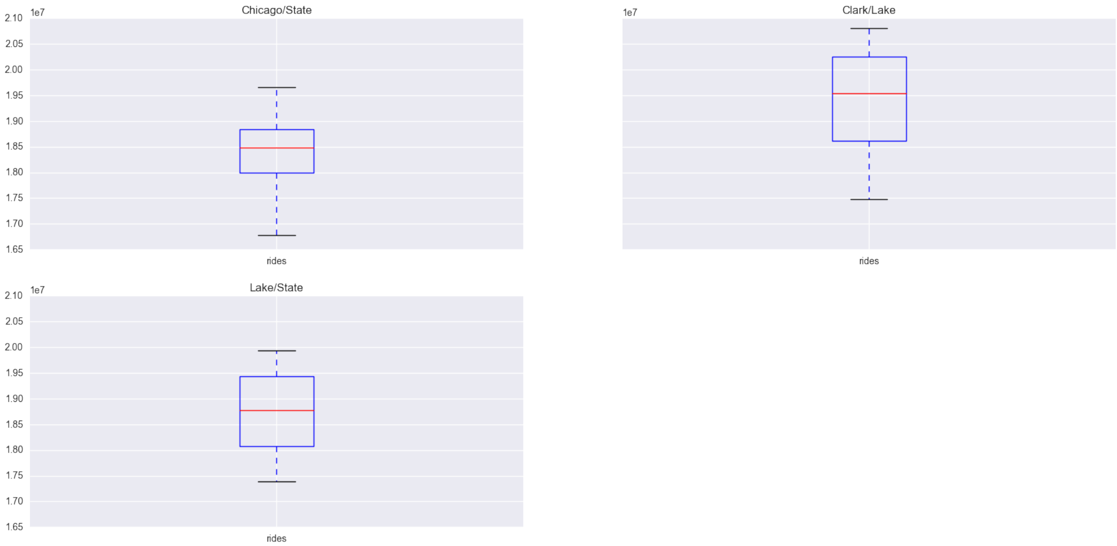
```
In [254]: plt.rcParams['figure.figsize'] = (10.0, 5.0)
          bar = sns.barplot(x="Season", y="rides", hue="stationname", data=topStati
```



A very interesting observation that can be made here is that for each season the top 3 are consistent but from the second plot we saw that Lake/State was doing very well but this plot shows otherwise. Lets look it at in detail:

```
In [255]: x = temp[(temp['stationname']=='Clark/Lake') | (temp['stationname']=='Chi
          x = x.groupby('stationname')
```

```
In [257]: plt.rcParams['figure.figsize'] = (20.0, 10.0)
          box = x.boxplot()
```



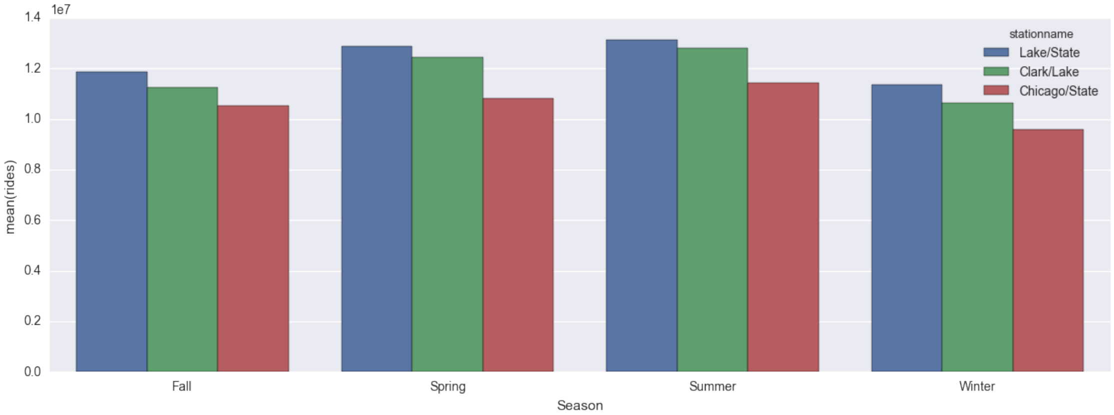
Interesting things to note from the boxplots based on seasons is that in plot 2 it was pretty evident that Lake/State had the most number of passengers across the years, but across all seasons there is a different story to tell. Clark/Lake seems the better option across all the seasons with its median the largest and minimum value, the lowest. Now this is because of the fact that between 2001-2008 Lake/State wasn't in the top used stations and that effect can be seen in the above boxplot.

So now, I'll try to look at the top stations for each season after 2008.

```
In [258]: temp = df[(df['Year']>=2008)]
```

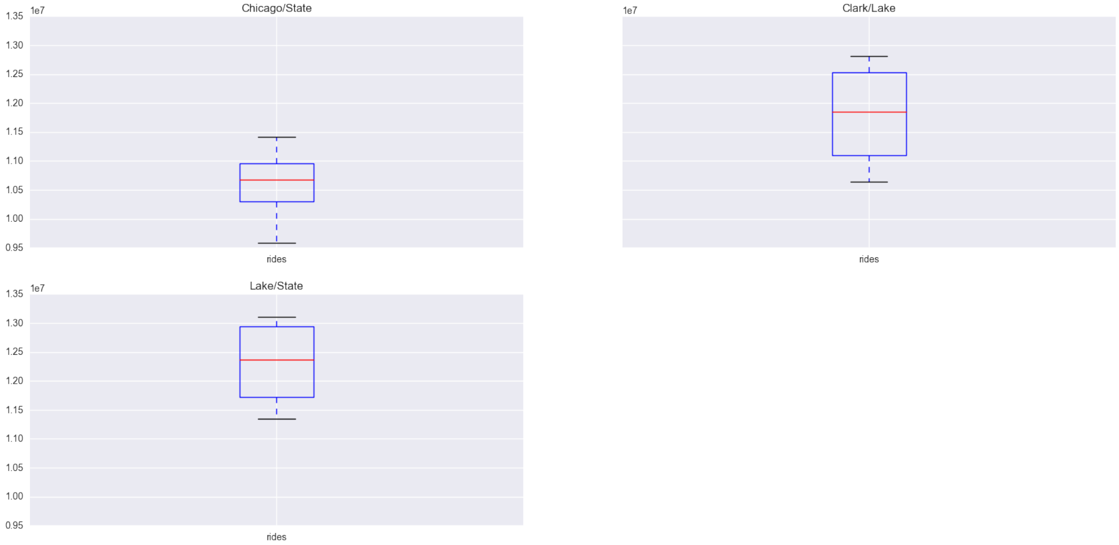
```
In [260]: grouped = temp.groupby(['Season','stationname'])
          temp = grouped['rides'].sum().reset_index()
          temp = temp.sort( ['rides'], ascending=[0])
          s = np.unique(temp['Season'])
          topStations = pd.DataFrame()
          for i in s:
              topStations = topStations.append(temp[(temp['Season']==i)][:3])
```

```
In [264]: plt.rcParams['figure.figsize'] = (15.0, 5.0)
          bar = sns.barplot(x="Season", y="rides", hue="stationname", data=topStati
```

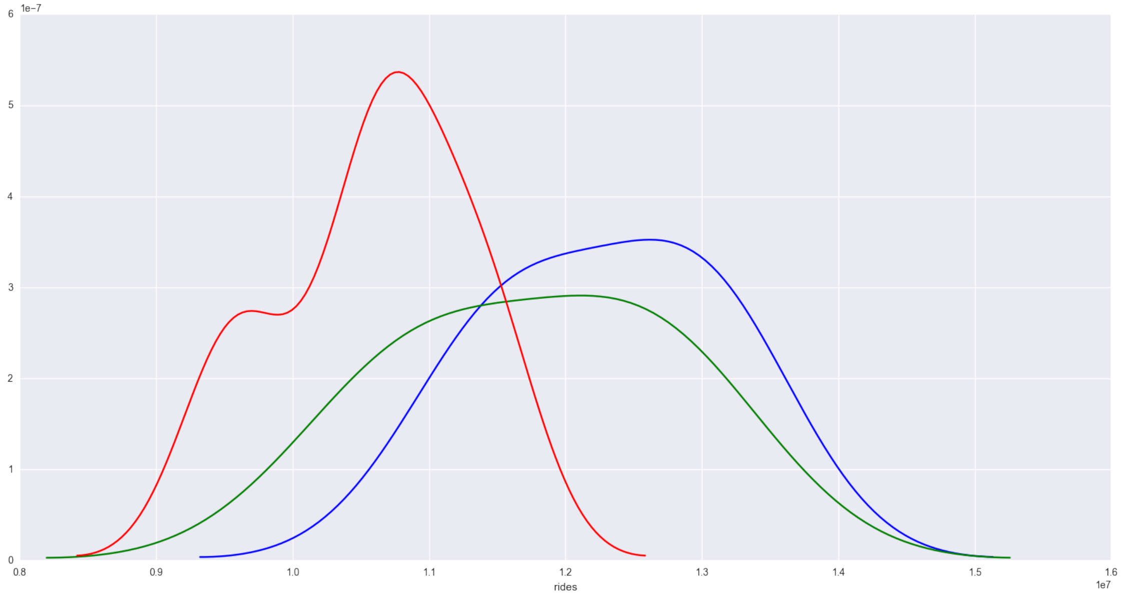


```
In [265]: x = temp[(temp['stationname']=='Clark/Lake') | (temp['stationname']=='Chi
          x = x.groupby('stationname')
```

```
In [267]: plt.rcParams['figure.figsize'] = (20.0, 10.0)
          box = x.boxplot()
```



```
In [268]: x = temp[(temp['stationname']=='Lake/State')]
          ax = sns.distplot(x['rides'], color='blue', hist=False)
          x = temp[(temp['stationname']=='Clark/Lake')]
          ax = sns.distplot(x['rides'], color='green', hist=False)
          x = temp[(temp['stationname']=='Chicago/State')]
          ax = sns.distplot(x['rides'], color='red', hist=False)
```



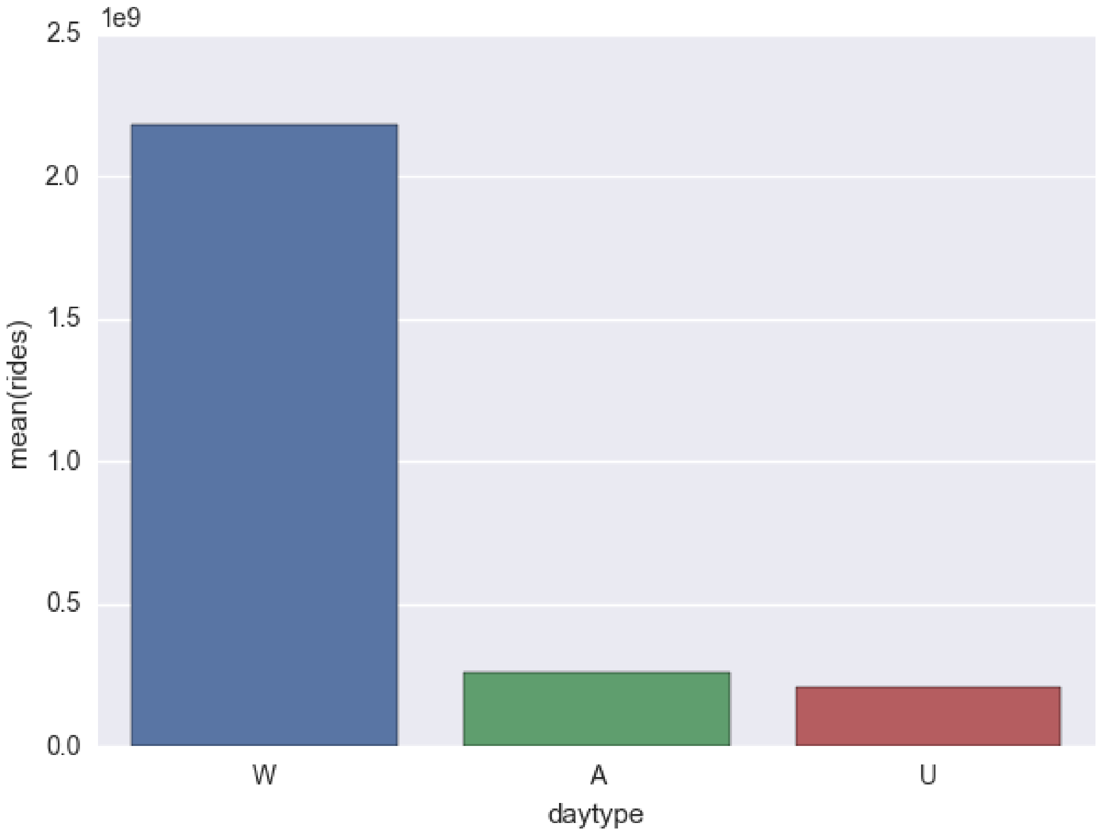
Now we get a better picture for the last 7.5 years where Lake/State has the most number of riders an also has consistent amount of riders across all seasons. This information of having consistent amount of rides across all seasons provides us with a very strong evidence of opening the business in the Lake/State neighbourhood. The distribution and boxplots make it clear that Lake/State has a larger mean, median, maximum count and lowest minimum count for years after 2008.

After learning a little more about where the top ride stations are, I found that all the highly visited stations lie in the downtown region and Lake/State and Clark/Lake are only a couple of blocks away. Now, the best location would be to open a business between the two stations to get the best of both worlds.

Finally to check how things vary based on the type of the day, I tried to build a bar graph that showed count based on the different type of days.

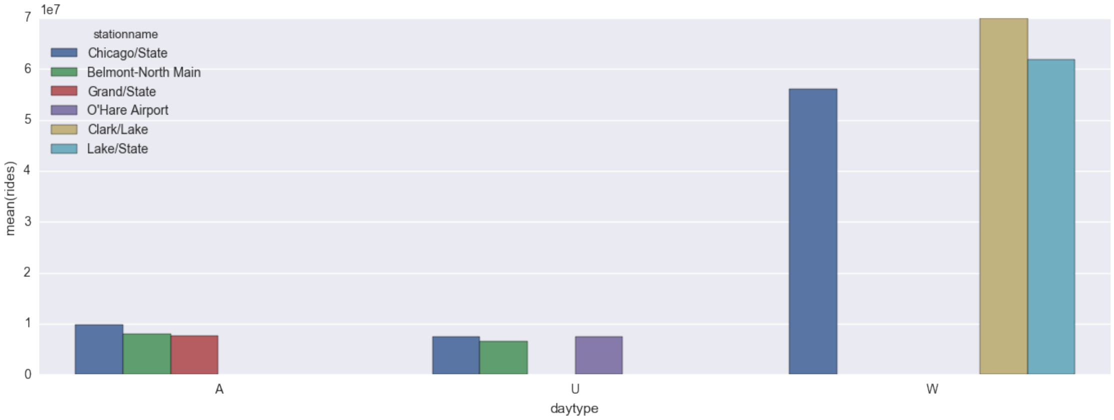
```
In [269]: grouped = df.groupby(['daytype'])
          temp = grouped['rides'].sum().reset_index()
          temp = temp.sort( ['rides'], ascending=[0])
```

```
In [274]: plt.rcParams['figure.figsize'] = (7.0, 5.0)
          bar = sns.barplot(x="daytype", y="rides", data=temp)
```



As expected, the number of users are way too high for weekdays than the weekends mainly because of teh scale of weekdays throughtout the years. Lets look at other plots:

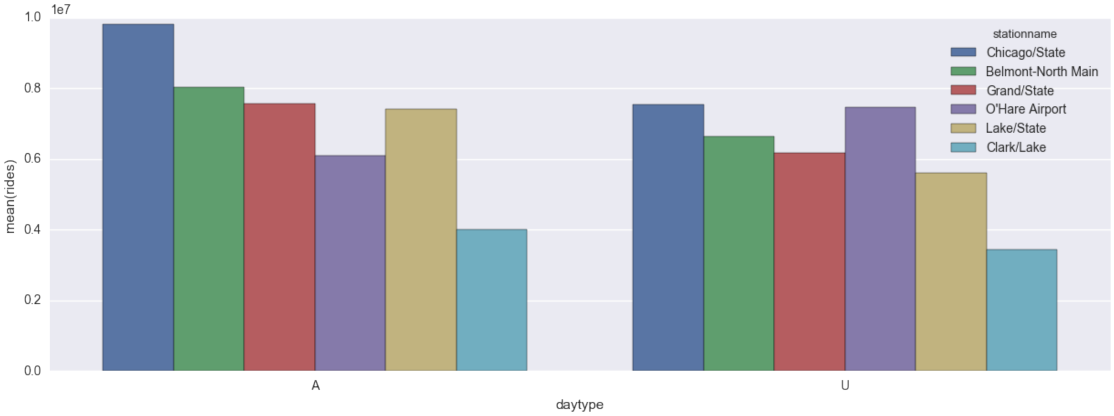
```
In [300]: grouped = df.groupby(['daytype','stationname'])
          temp = grouped['rides'].sum().reset_index()
          temp = temp.sort( ['rides'], ascending=[0])
          s = np.unique(temp['daytype'])
          topStations = pd.DataFrame()
          for i in s:
              topStations = topStations.append(temp[(temp['daytype']==i)][:3])

In [283]: plt.rcParams['figure.figsize'] = (15.0, 5.0)
          bar = sns.barplot(x="daytype", y="rides", hue="stationname", data=topStat
```



Another interesting observation to be made here is that although the stations Lake/State and Clark/Lake do not get as many riders during the weekends and holidays, they make up for it during the weekdays. Since a business always wants consistency, Chicago/State features in all the plots and can be a viable option. But before jumping to any conclusions, I want to see how many riders use the Lake/State and Clark/Lake stations.

```
In [302]: x = temp[(temp['stationname']=='Clark/Lake') | (temp['stationname']=='Lak
          x = x[(x['daytype']!='W')]
```

```
In [306]: plt.rcParams['figure.figsize'] = (15.0, 5.0)
          bar = sns.barplot(x="daytype", y="rides", hue="stationname", data=x)
```



From the above plots, it is quite evident that Lake/State and Clark/Lake are not the popular choices during the weekends and holidays. It is understandable that O'Hare is in the mix because a lot of people travel to and from the airport during the weekends and holidays. I would still say that Lake/State and Clark/Lake are the safer locations to open the new business because of the scale of numberof weekdays as compared to weekends.

Coming to the second part of the question, I believe there are a lot of other factors that could help me decide the best location for opening a new business. Things like the social and economic development of the neighbourhood would play a major impact in deciding the location. Also, depending on the type of business looking into the time of riders would also give us a better picture. For example, restaurants are usually busy during lunch, dinner and weekends. To see what stations are prevalent then, would be interesting. So, the time factor usually would be used to see how it impacts the business.

Another approach to the problem could be to convert the dataset into a time-series problem and predict the count of rides for a particular station. This additional information would also give us a peek into the future and that would definitely help us make better decisions and learn even better.

Finally, I want to see data that is simple and easy to comprehend. I know that is not always the case and a lot of time is utilized in fetching the right kind of data. Once the data is gathered, the representation should also be very meaningful and easily understood. If the data is not presented in simple terms, it can neither be understood nor it can be explained and that wouldn't lead to good decisions.