Notebook

December 8, 2019

0.0.1 Question 1d

In the following cell, print a summary of the data selection and cleaning you performed. For example, you should print something like: ”Of the original 1000 trips, 21 anomolous trips (2.1%) were removed through data cleaning, and then the 600 trips within Manhattan were selected for further analysis.” (Note that the numbers in this example are not accurate.)

Your Python code should not include any number literals, but instead should refer to the shape of `all_taxi`, `clean_taxi`, and `manhattan_taxi`. Your response will be scored based on whether you generate an accurate description and do not include any number literals in your Python expression, but instead refer to the dataframes you have created.

One way to do this is with Python’s f-strings. For instance,

```
name = "Joshua"
print(f"Hi {name}, how are you?")
```

prints `Hi Joshua, how are you?`.

Please ensure that your Python code does not contain any very long lines, or we can’t grade it.

```
In [21]: all_taxi_size = len(all_taxi)
         clean_taxi_size = len(clean_taxi)
         manhattan_taxi_size = len(manhattan_taxi)
         taxi_size_diff = all_taxi_size - clean_taxi_size
         print(f"There are {all_taxi_size} original trips with pick-up and drop-off locations within the
         print(f"We removed {taxi_size_diff} anomalous trips, that did not have a positive passenger co
         print(f"After this data cleaning,  we have {clean_taxi_size} trips, and by only choosing trips
```

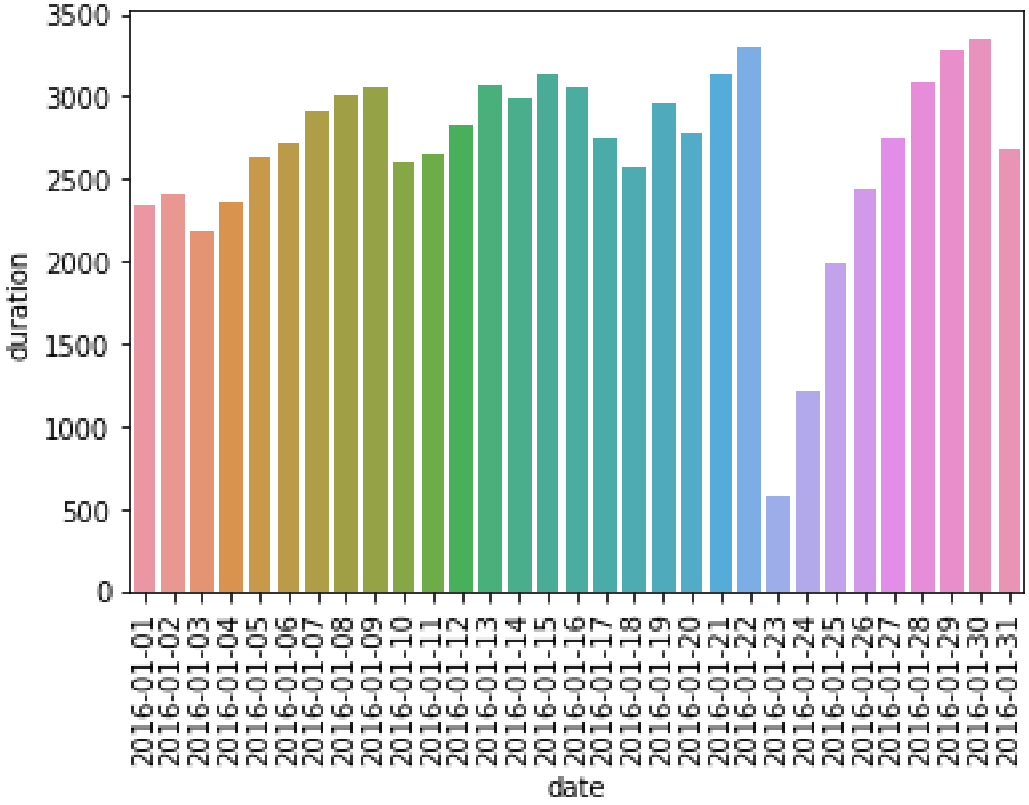
```
There are 97692 original trips with pick-up and drop-off locations within the boundaries of New York Ci
We removed 1247 anomalous trips, that did not have a positive passenger count, a positive distance, a d
After this data cleaning,  we have 96445 trips, and by only choosing trips that start and end within a p
 defines the boundaries of Manhattan Island, we end up with 82800 trips.
```

0.0.2 Question 2b

Create a data visualization that allows you to identify which dates were affected by the historic blizzard of January 2016. Make sure that the visualization type is appropriate for the visualized data.

Hint: How do you expect taxi usage to differ on blizzard days?

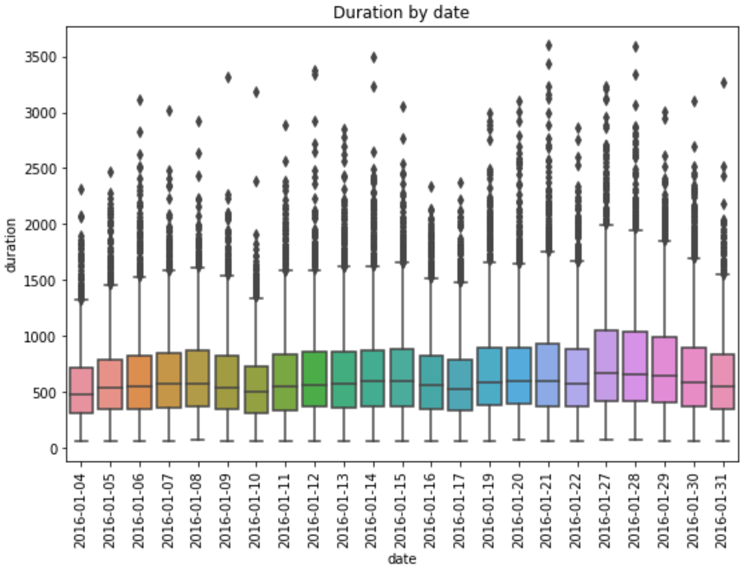
```
In [24]: date_duration = manhattan_taxi.groupby('date').count().reset_index()
         ax_2b = sns.barplot(x='date',y='duration',data=date_duration)
         plt.xticks(rotation=90);
```



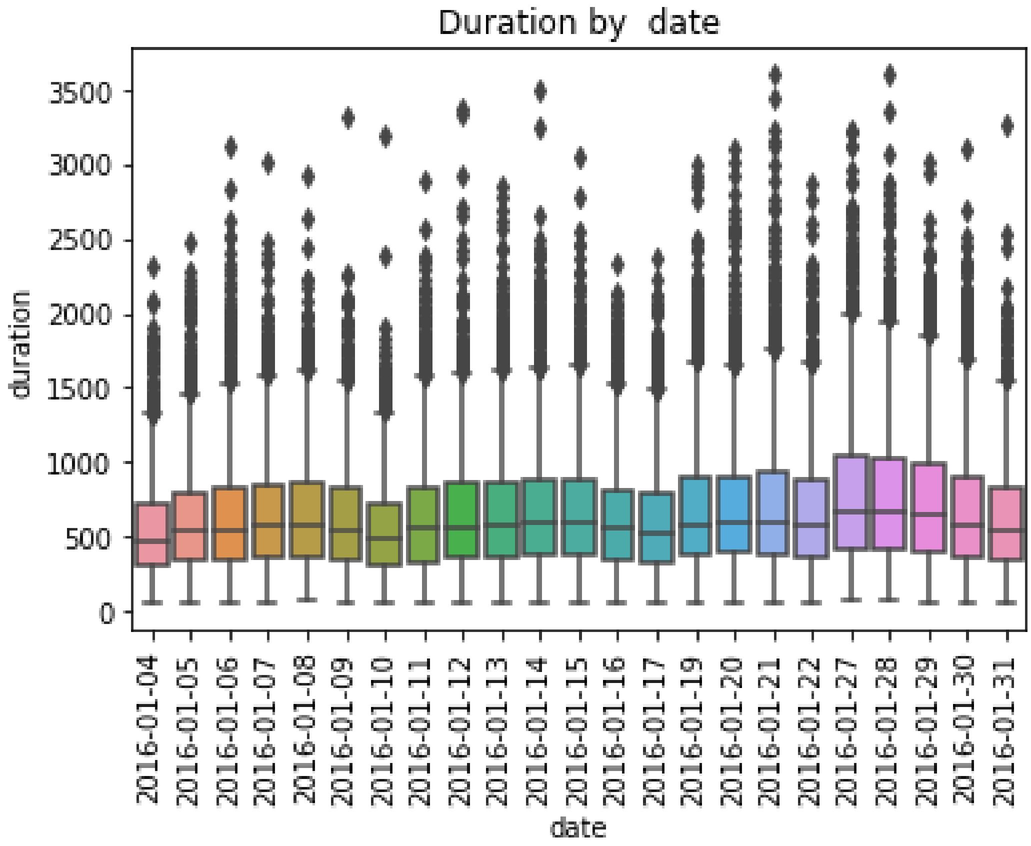
0.0.3 Question 3a

Create a box plot that compares the distributions of taxi trip durations for each day **using `train` only**. Individual dates shoud appear on the horizontal axis, and duration values should appear on the vertical axis. Your plot should look like the following.

Hint: Use `sns.boxplot`.



```
In [29]: q3a = train.copy().sort_values(by=['date'])
         ax_3a = sns.boxplot(x="date", y="duration", data=q3a)
         ax_3a.set(xlabel='date', ylabel='duration', title='Duration by  date')
         plt.xticks(rotation=90);
```


Duration by date
duration
3500
3000
2500
2000
1500
1000
500
0
2016-01-04
2016-01-05
2016-01-06
2016-01-07
2016-01-08
2016-01-09
2016-01-10
2016-01-11
2016-01-12
2016-01-13
2016-01-14
2016-01-15
2016-01-16
2016-01-17
2016-01-19
2016-01-20
2016-01-21
2016-01-22
2016-01-27
2016-01-28
2016-01-29
2016-01-30
2016-01-31
date

0.0.4 Question 3b

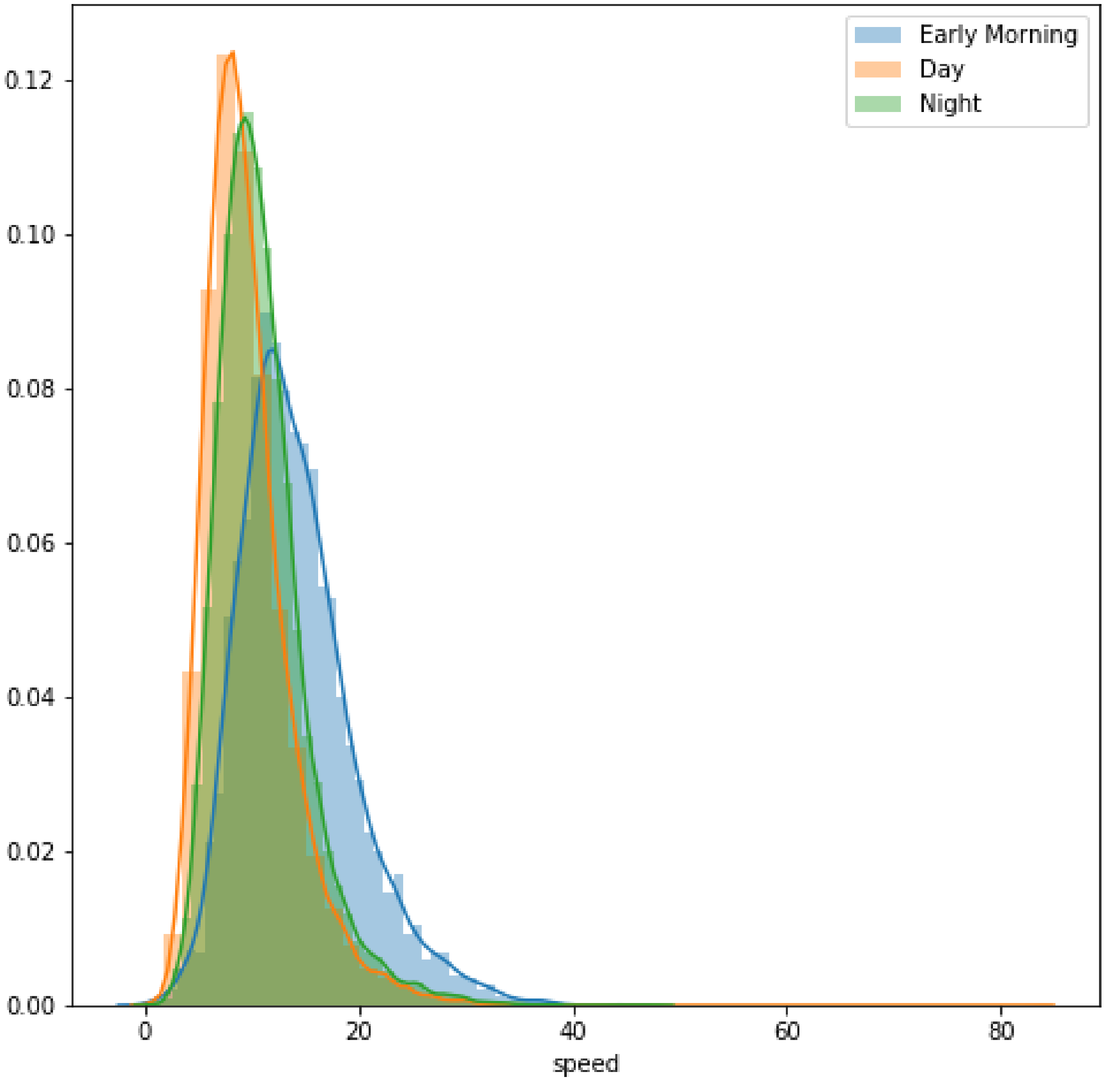
In one or two sentences, describe the assocation between the day of the week and the duration of a taxi trip. This question will be graded on whether your answer is justified by your boxplot and if it is at least somewhat meaningful.

Note: The end of Part 2 showed a calendar for these dates and their corresponding days of the week.

Looking at the boxplots, we see that taxi duration is shorter during the weekends (Saturday and Sunday). We see this for the dates 9,10,16,17,30,31 of January. This is the case, because during the weekends, the boxplots are shorter and are shifted downwards relative to the y axis.

0.0.5 Question 3c

Use `sns.distplot` to create an overlaid histogram comparing the distribution of average speeds for taxi rides that start in the early morning (12am-6am), day (6am-6pm; 12 hours), and night (6pm-12am; 6 hours). Your plot should look like this:



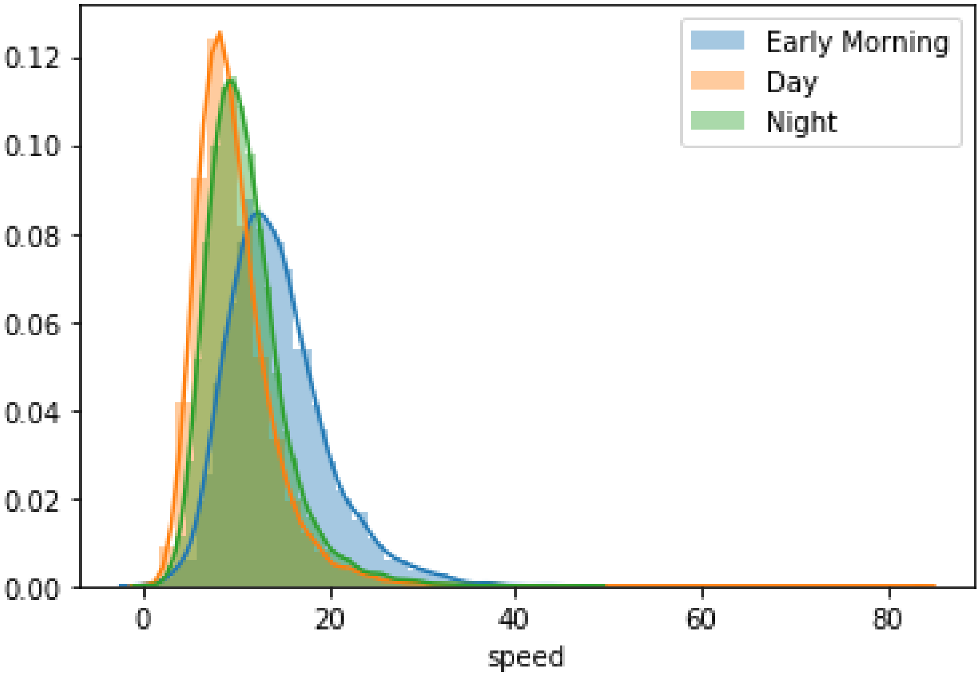
```
In [31]: early_morning = train[train['hour']>=0][train['hour']<=6]['speed']
         day = train[train['hour']>=6][train['hour']<=18]['speed']
         night = train[train['hour']>=18][train['hour']<=24]['speed']

         sns.distplot(early_morning,label='Early Morning')
         sns.distplot(day,label='Day')
         sns.distplot(night,label='Night')
         plt.legend()
```

```
/srv/conda/envs/data100/lib/python3.6/site-packages/ipykernel_launcher.py:2: UserWarning: Boolean Serie

/srv/conda/envs/data100/lib/python3.6/site-packages/ipykernel_launcher.py:3: UserWarning: Boolean Serie
  This is separate from the ipykernel package so we can avoid doing imports until
```

```
Out[31]: <matplotlib.legend.Legend at 0x7f204c11aa90>
```


Early Morning
Day
Night
0.12
0.10
0.08
0.06
0.04
0.02
0.00
0
20
40
60
80
speed

0.0.6 Question 4e

In one or two sentences, explain how the `period` regression model could possibly outperform linear regression model, even when the design matrix of the latter includes one feature for each possible hour.

The `period` regression model outperforms the linear regression model because we are training our model based on each period. Each of the three periods (morning, afternoon, night) would definately have an impact on the duration of a taxi ride. Additionally, even if the design matrix of the latter includes one feature for each possible hour, there are 24 hours in a day, and these many features would not be able to build a robust model. On the other hand, having periods splits up all those hours into 3 periods, which gives the model more correlated data to work with. Thus, the RMSE for the `period` regression model is lower.