

Groupon Assignment Time-Series

August 11, 2020

```
[1]: import pandas as pd
import numpy as np
from sklearn.experimental import enable_iterative_imputer
from sklearn.impute import IterativeImputer
from datetime import datetime as dt
from datetime import date
import matplotlib.pyplot as plt
import matplotlib.dates as mdates
```

```
[2]: plt.rcParams['figure.figsize'] = [10, 5] #Adjust Plot size
```

1 Data Cleaning

```
[3]: data = pd.read_csv("C:\\Users\\victo\\Desktop\\Data.csv") #converting .csv to DataFrame
print(data.shape)
data.dtypes
```

(138534, 8)

```
[3]: Deal ID          object
Units Sold          float64
Billings            float64
Start Date          object
Deal URL            object
Segment             object
Inventory Type       object
Unnamed: 7          float64
dtype: object
```

```
[4]: data = data[['Billings', 'Start Date', 'Segment']]
data['Start Date'] = data['Start Date'].apply(pd.to_datetime)
data = data.sort_values(by=['Start Date'])
```

2 DataFrame Creation

```
[5]: # imputation function

def imputate(df):
    imp = IterativeImputer(max_iter = 10, random_state = 0)
    imp.fit(df["2013-10"])
    IterativeImputer(random_state = 0)

    res = pd.DataFrame(data = np.random.randint(1,1000, (11,1)),
                        columns = ['Local Billings'])
    res['Local Billings'] = np.NaN

    res = pd.DataFrame(np.round(imp.transform(res), decimals = 2),
                        columns = ['Local Billings'])

    res = res.set_index(pd.date_range(start = '10/20/2013', end = '10/30/2013'))
    res = pd.concat([df,res])['Local Billings']
    return res.sort_index()
```

```
[6]: #Using dictionaries to organize data.

loc_dict = {'Local Billings':[], 'Start Date':[]}
g_dict = {'Goods Billings':[], 'Start Date':[]}
trav_dict = {'Travel Billings':[], 'Start Date':[]}

lcount = -1
gcount = -1
tcount = -1

ldate = pd.Timestamp(2018,1,1,0)
gdate = pd.Timestamp(2018,1,1,0)
tdate = pd.Timestamp(2018,1,1,0)

for index, row in data.iterrows():
    if row[2] == "Local":
        if ldate == row[1]:
            loc_dict['Local Billings'][lcount] += row[0]
        else:
            loc_dict['Start Date'] += [row[1]]
            loc_dict['Local Billings'] += [row[0]]
            lcount += 1
            ldate = row[1]

    elif row[2] == "Goods":
        if gdate == row[1]:
            g_dict['Goods Billings'][gcount] += row[0]
```

```

        else:
            g_dict['Start Date'] += [row[1]]
            g_dict['Goods Billings'] += [row[0]]
            gcount += 1
            gdate = row[1]

    elif row[2] == "Travel":
        if tdate == row[1]:
            trav_dict['Travel Billings'][tcount] += row[0]
        else:
            trav_dict['Start Date'] += [row[1]]
            trav_dict['Travel Billings'] += [row[0]]
            tcount += 1
            tdate = row[1]

```

[7]: *#Convert dict to DataFrame*

```

loc_temp = pd.DataFrame(loc_dict)
loc_temp = loc_temp.set_index('Start Date')

local = imputate(loc_temp) #fix issue with dates

goods = pd.DataFrame(g_dict)
goods = goods.set_index('Start Date')

travel = pd.DataFrame(trav_dict)
travel = travel.set_index('Start Date')

```

3 Calculating Estimated Billings

```

[8]: lsum = np.round(local.sum()/10**6, decimals = 2)
    gsum = np.round(goods['Goods Billings'].sum()/10**6, decimals = 2)
    tsum = np.round(travel['Travel Billings'].sum()/10**6, decimals = 2)
    groupSum = lsum + tsum + gsum

    print('Estimated Billings (Local): $', lsum, sep = '')
    print('Estimated Billings (Goods): $', gsum, sep = '')
    print('Estimated Billings (Travel): $', tsum, sep = '')

```

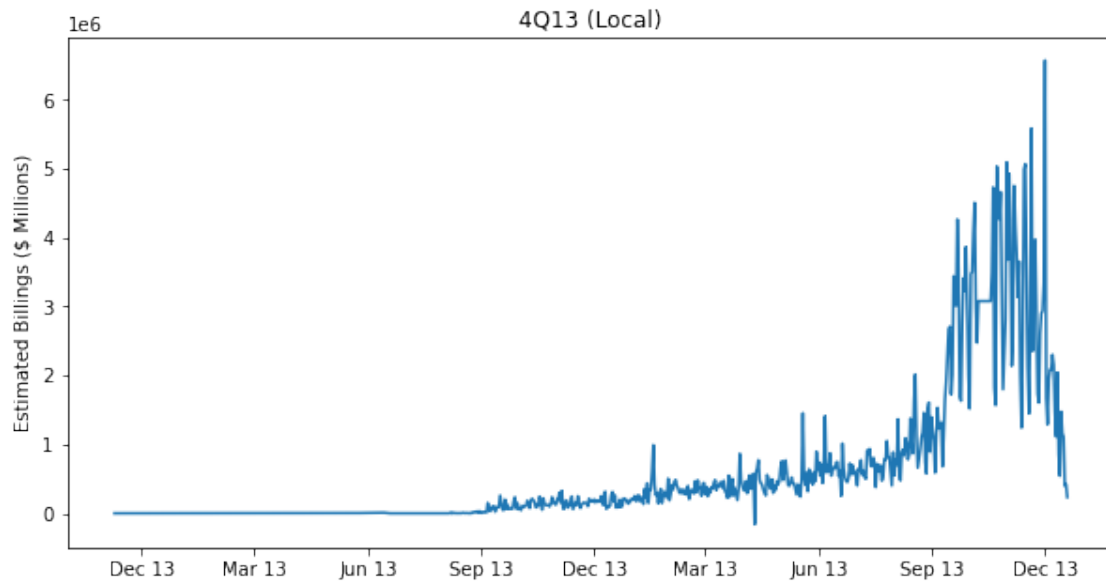
```

Estimated Billings (Local): $443.1
Estimated Billings (Goods): $282.25
Estimated Billings (Travel): $70.55

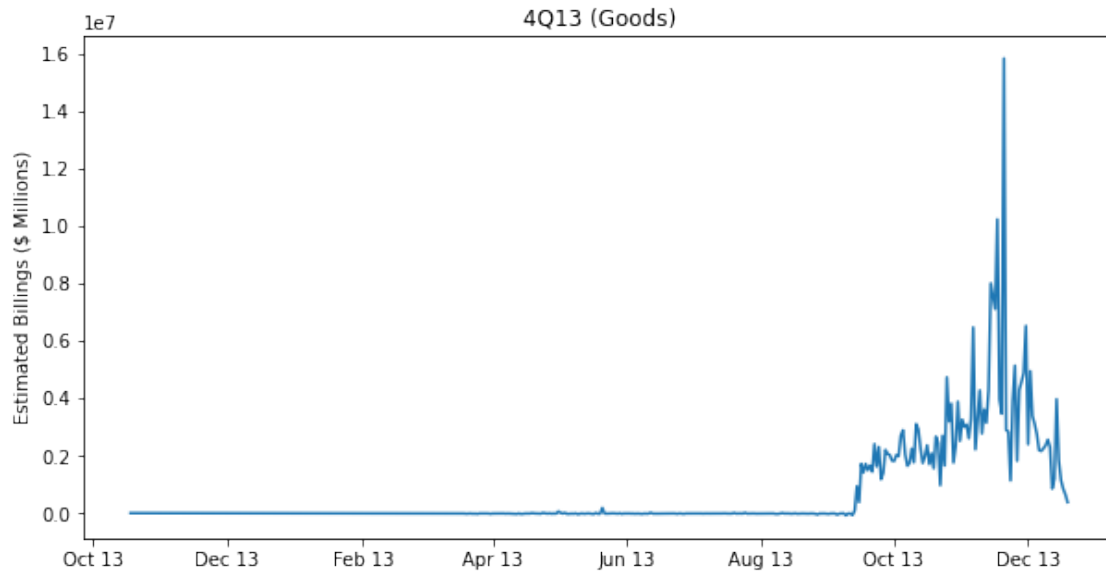
```

4 Time-Series Graphs

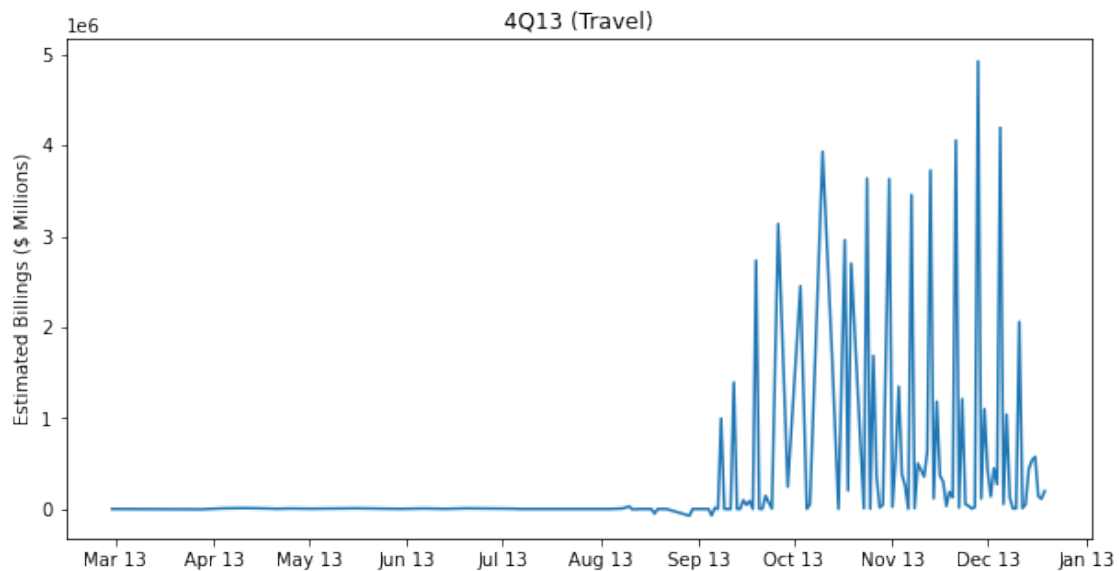
```
[9]: fig, axis = plt.subplots()
axis.plot(local, linestyle = '-')
axis.set_ylabel('Estimated Billings ($ Millions)')
axis.set_title('4Q13 (Local)')
axis.xaxis.set_major_locator(mdates.MonthLocator(bymonthday = 13, interval = 3))
axis.xaxis.set_major_formatter(mdates.DateFormatter('%b %d'));
```



```
[10]: fig, axis = plt.subplots()
axis.plot(goods, linestyle = '-')
axis.set_ylabel('Estimated Billings ($ Millions)')
axis.set_title('4Q13 (Goods)')
axis.xaxis.set_major_locator(mdates.MonthLocator(bymonthday = 13, interval = 2))
axis.xaxis.set_major_formatter(mdates.DateFormatter('%b %d'));
```



```
[11]: fig, axis = plt.subplots()
axis.plot(travel, linestyle = '-')
axis.set_ylabel('Estimated Billings ($ Millions)')
axis.set_title('4Q13 (Travel)')
axis.xaxis.set_major_locator(mdates.MonthLocator(bymonthday = 13, interval = 1))
axis.xaxis.set_major_formatter(mdates.DateFormatter('%b %d'));
```



```
[ ]:
```