Big Data Analytics Techniques and Applications

Homework 1

309652022 黃伯丞

- **資料集描述**

這次作業所使用的資料集是 NYC Taxi 從 2009 年一月到三月間的 Yellow Taxi Trip Records，根據 User Guide 的內容可得知資料裡面總共有 19 個欄位，有些欄位沒有資料(例如 RateCodeID 以及 Store_and_fwd_flag 等)，網站上的資料以月為單位分裝，在 2009 年的一月到三月資料每份介於 2~3GB 之間，總共 6GB、41,859,906 筆資料。

- **使用工具**

我利用 Python 跟 Pandas 進行預處理後再將所得到的資料利用 Tableau 呈現。

- **預處理過程**

因為檔案大小比較大的關係所以我並沒有選擇直接下載個別檔案，而是先觀察部分資料後再利用 Pandas 將需要用到的欄位取出來，這樣大小可以降低至少 50%。以下是選取欄位下載再合併的步驟。

```
cols=['Trip_Pickup_DateTime','Trip_Dropoff_DateTime','Trip_Distance','Start_Lon','Start_Lat','End_Lon','End_Lat','Total_Amt']

dtypes = {'Embarked': 'category'}
df=pd.read_csv('https://s3.amazonaws.com/nyc-tlc/trip+data/yellow_tripdata_2009-01.csv',dtype=dtypes,usecols=cols)

df2=pd.read_csv('https://s3.amazonaws.com/nyc-tlc/trip+data/yellow_tripdata_2009-02.csv',dtype=dtypes,usecols=cols)

df3=pd.read_csv('https://s3.amazonaws.com/nyc-tlc/trip+data/yellow_tripdata_2009-03.csv',dtype=dtypes,usecols=cols)

df=pd.concat([df,df2,df3],ignore_index=True)
```

- **結果**

I. Q1

第一題要統計上下車最多次的地點。資料集當中有經緯度小數點後 6 位的紀錄，統計時我取的是小數點後 2 位。其中有些是缺失值或著是偏差過多的資料，因為這題只要找出最大的量所以沒有造成太大影響。我計算的方式是將經緯度轉成小數點後 2 位再將其用,隔開轉成字串，接著建立新欄位存放。然後利用 value_count 計算不同組合的出現次數，將輸出的 Series 格式物件轉成 Dataframe 再把當中的經緯度分別求出來，存至 Start_loc.csv 跟 End_loc.csv 完成統計。

```
df['Start_loc'] = df['Start_Lon'].round(2).apply(str) + ',' + df['Start_Lat'].round(2).apply(str)

Start=df.loc[:,'Start_loc'].value_counts()

df_Start=pd.Series(Start.rename_axis('region'))
df_Start=df_Start.reset_index()

df_Start[['Lon','Lat']]=df_Start.region.str.split(',', expand=True)

df_Start.to_csv('Start_loc.csv')
```

```
df['End_loc'] = df['End_Lon'].round(2).apply(str) + ',' + df['End_Lat'].round(2).apply(str)

End=df.loc[:,'End_loc'].value_counts()

df_End=pd.Series(End.rename_axis('region'))
df_End=df_End.reset_index()

df_End[['Lon','Lat']]=df_End.region.str.split(',', expand=True)

df_End.to_csv('End_loc.csv')
```

接下來是資料視覺化的步驟，因為已經在 Python 統計完成所以留下的檔案很小，讀取快速，方便在 Tableau 進行操作。首先先將各座標出現的次數進行排序再輸出成表格，除了用表格呈現之外也能利用經緯度的資料將其投影在地圖上，觀察哪個區域最密集(限制出現次數至少 60000 次)。從排序資料中可以發現前四名不論起點終點位置都一樣，而第五名第六名對調以及第七名第八名對調，地點主要集中在紐約市區。

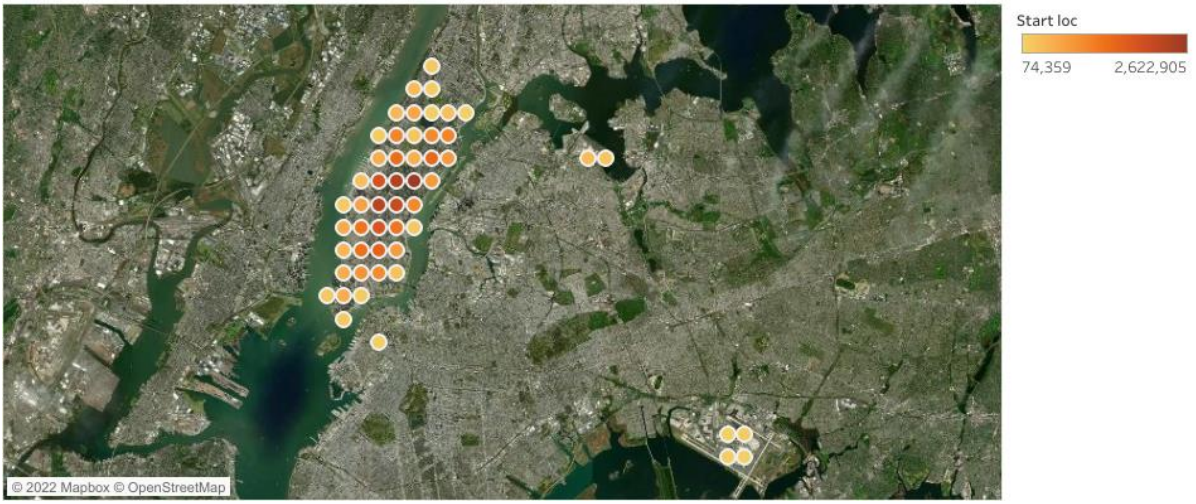
起點排序

起點座標	起點排名	起點總數
-73.97,40.76	1	2,622,905
-73.98,40.76	2	2,331,900
-73.99,40.75	3	2,293,312
-73.98,40.75	4	2,069,723
-73.99,40.76	5	1,885,219
-73.99,40.74	6	1,839,862
-73.99,40.73	7	1,570,153
-73.96,40.77	8	1,492,407
-73.98,40.77	9	1,395,813
-74.0,40.73	10	1,338,418

終點排序

終點座標	終點排名	終點總數
-73.97,40.76	1	2,341,673
-73.98,40.76	2	2,335,210
-73.99,40.75	3	2,126,910
-73.98,40.75	4	2,039,476
-73.99,40.74	5	1,623,692
-73.99,40.76	6	1,552,091
-73.96,40.77	7	1,428,619
-73.99,40.73	8	1,280,443
-73.98,40.74	9	1,219,452
-73.98,40.77	10	1,207,348

起點位置



終點位置

II. Q2

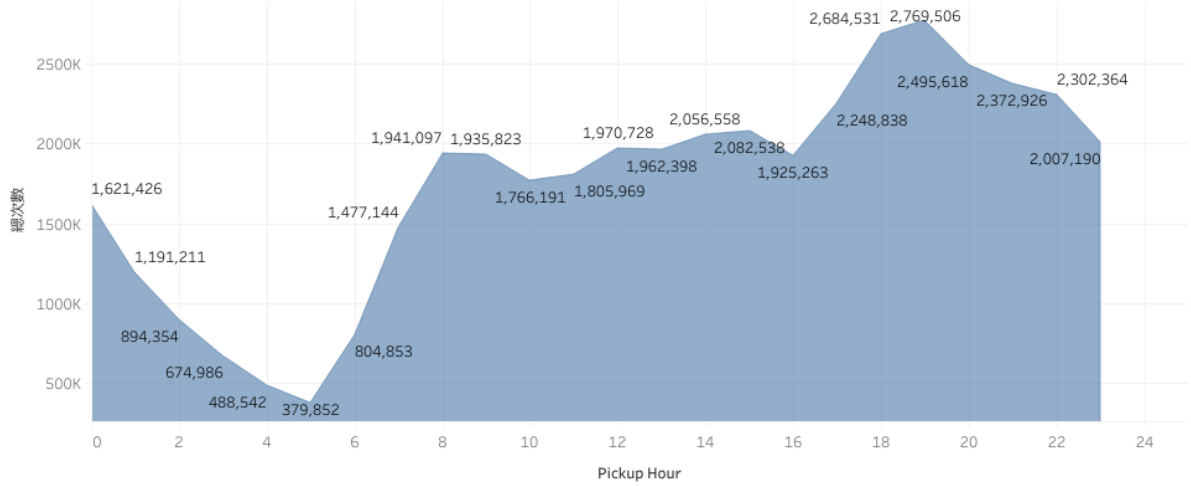
第二題要統計搭乘計程車的尖峰離峰時段，我利用上車的時間來進行分析。將 Trip_Pickup_DateTime 欄位的資料利用 to_datetime 轉成特定的時間資料結構，再把小時的資料提出來加入 Dataframe 的新欄位，然後跟 Q1 一樣用 value_count 計算出現次數存到 Pickup_Hour.csv 完成統計。

```
df['Trip_Pickup_DateTime'] = pd.to_datetime(df['Trip_Pickup_DateTime'], utc=True, format='%Y-%m-%d %H:%M:%S')
df['Pickup_Hour'] = df.Trip_Pickup_DateTime.dt.hour

a=df.loc[:,'Pickup_Hour'].value_counts()
a.to_csv('Pickup_Hour.csv')
```

把 Pickup_Hour.csv 上傳至 Tableau 即可畫出分布圖，可以從當中發現晚上 6~9 點為尖峰時間，而早上 3~6 點為離峰時間。

搭乘時間點



III. Q3

第三題需要指出如何判斷 Total_Amt 的大小。根據 Dataframe 的 describe 函式可以發現其數值至少為 $2.5，另外平均數為 $10.49 以及中位數為 $8.1，代表大部分的數值皆小於 $10 且有部分極值影響了平均數，其中最大值高達 $235.2。因此如果依照中位數為標準的話，超過 $8 的為 big total amounts，反之為 small。如果依照平均數為標準的話，超過 $10.5 的為 big total amounts，反之為 small。

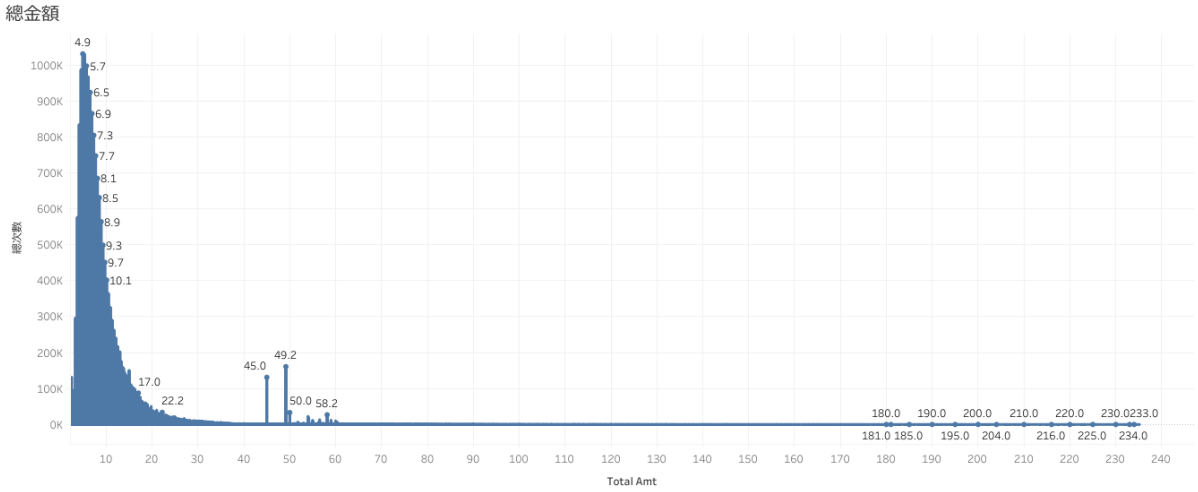
```
df.describe()['Total_Amt']
```

```
count    4.185991e+07
mean     1.049112e+01
std      8.393608e+00
min      2.500000e+00
25%      5.900000e+00
50%      8.100000e+00
75%      1.170000e+01
max      2.352000e+02
Name: Total_Amt, dtype: float64
```

```
a=df.loc[:,'Total_Amt'].value_counts()
```

```
a.to_csv('Total_Amount.csv')
```

透過 value_count 計算出現次數存到 Total_Amount.csv 的方式在 Tableau 上作圖後可以發現 $4.9 跟 $5.3 是出現最多次的，大部分的資料都發生在 $20 以內的範圍，在此範圍間呈現鐘形的結構。其中 $45 跟 $49.2 的出現次數相較其他 big total amounts 而言多次，意味著有些顧客經常需要以計程車進行較遠距離的往返。



- 問題討論

隨著資料量越大，如果只用一般的方式處理容易導致需要較長時間或大量的記憶體資源才能對付，所以需要採用其他的方式來處理大數據。假如實際所需要用到的資訊不需要所有欄位，沒有必要下載整份檔案，因此事先觀察大量資料中的一小部分資料十分重要，比起直接研究所有資料可以省下不少力氣。

另外我利用 Dataframe 的 corr 去觀察 Total_Amt 跟其他欄位的相關係數，發現 Trip_Distance 跟它的相關程度相當高，有 0.88，During_time (以秒為單位的車程時間) 的相關係數只有 0.05。

```
df['time']=df.Trip_Dropoff_DateTime-df.Trip_Pickup_DateTime
```

```
df['During_time']=df.time.dt.total_seconds()/60
```

```
df.corr()['Total_Amt']
```

```
Trip_Distance    0.883570
Total_Amt        1.000000
Pickup_Hour     -0.009815
During_time      0.052260
Name: Total_Amt, dtype: float64
```