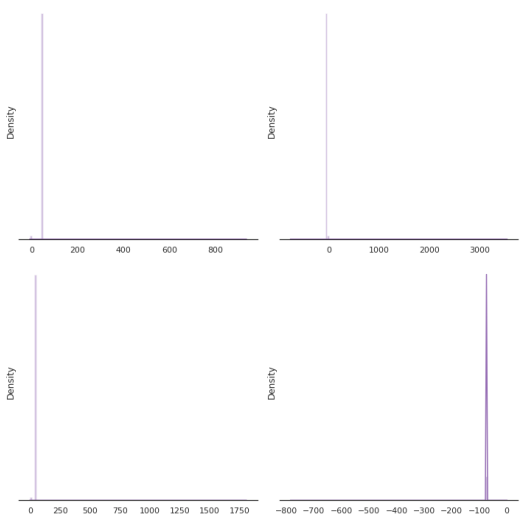
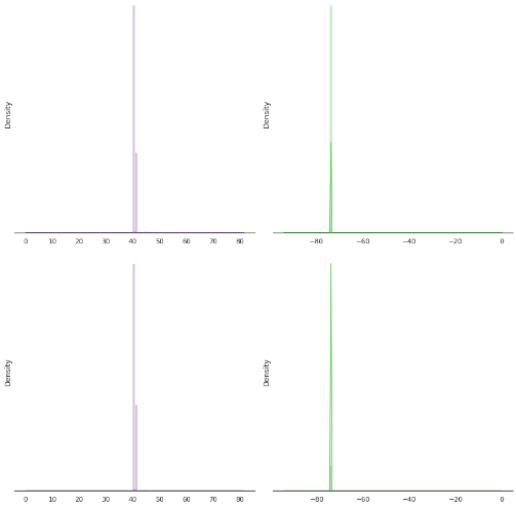
310553019 鄭以昀

Q1: What **regions** have the most pickups? What are the top-5 regions with the **most pickups and drop-offs** (pickups and drop-offs should be counted separately)?

- 收集三個月間所有出發地經緯度和目的地經緯度
- 將經緯度視覺化，發現存在異常值(根本不存在的經緯度等)
- 清除異常值
  - 第一次嘗試，挑選出位於北半球、西半球的經緯度
    考慮美國位置且為開車到達的了的地方(這個方法不夠好)

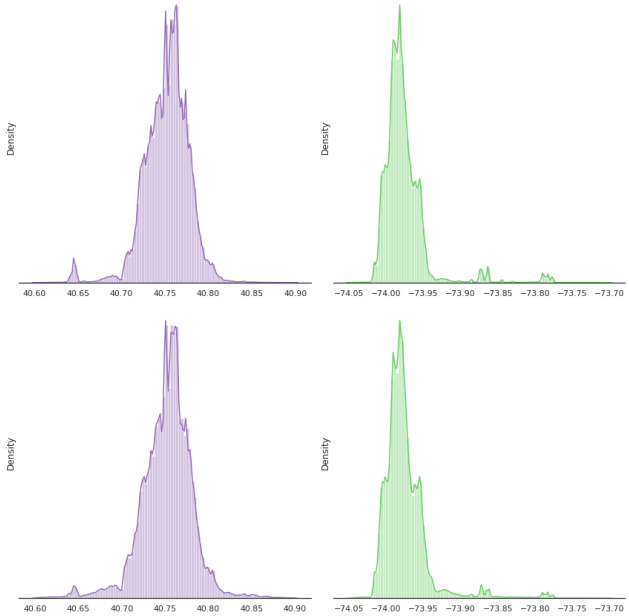


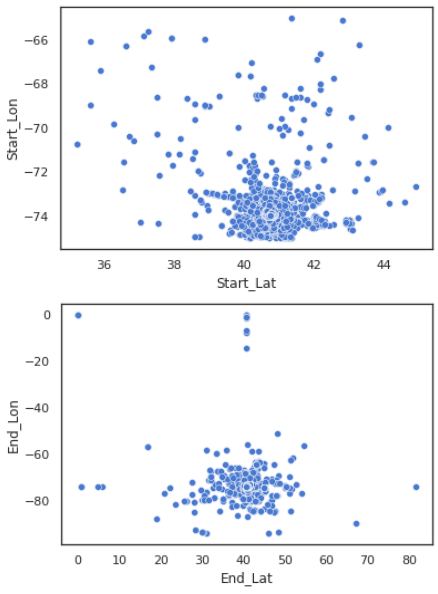
異常值



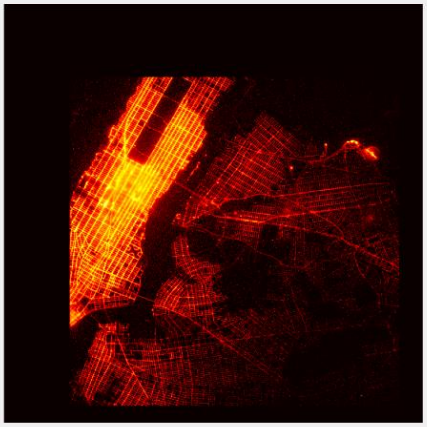
北半球、西半球的經緯度

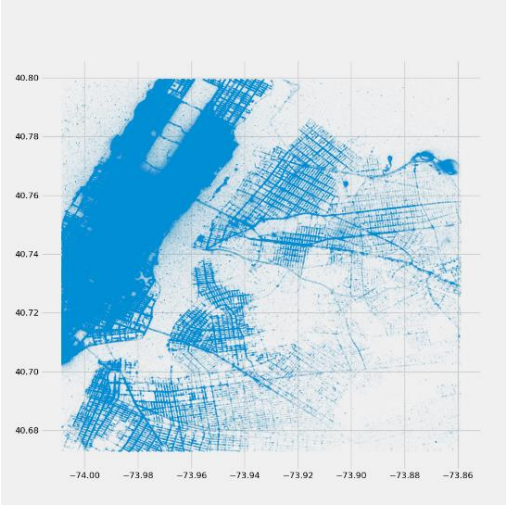
  - 透過觀察上圖自行給定的範圍畫出直方圖和散布圖

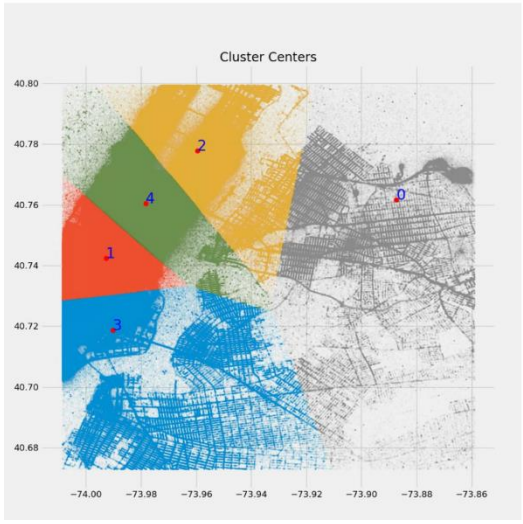




- 了解資料大致狀況(類似常態分佈的形狀)後，將資料頭尾 2.5%視為異常值去除
- 將所有出發地+目的地經緯度視為一個點







Cluster 編號/該 cluster 的個數

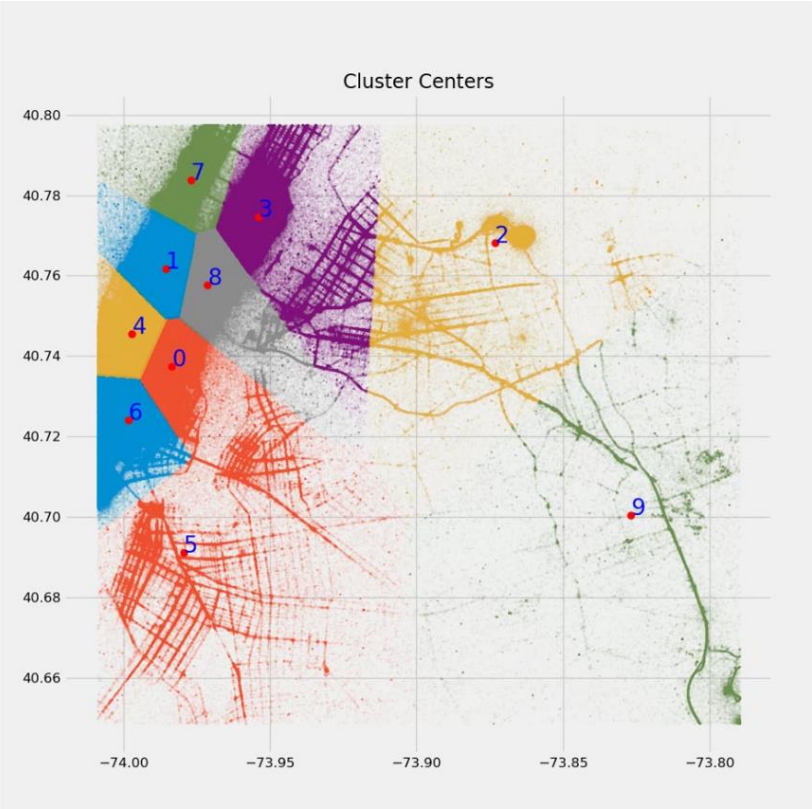
4	24617997
1	19574454
2	14933378
3	10293188
0	1811301

對照地圖(也有使用經緯度確認)後，明顯是曼哈頓下半部這區有最多 pickups + drop-offs

- Pickups

使用相同方法，只看出發地經緯度

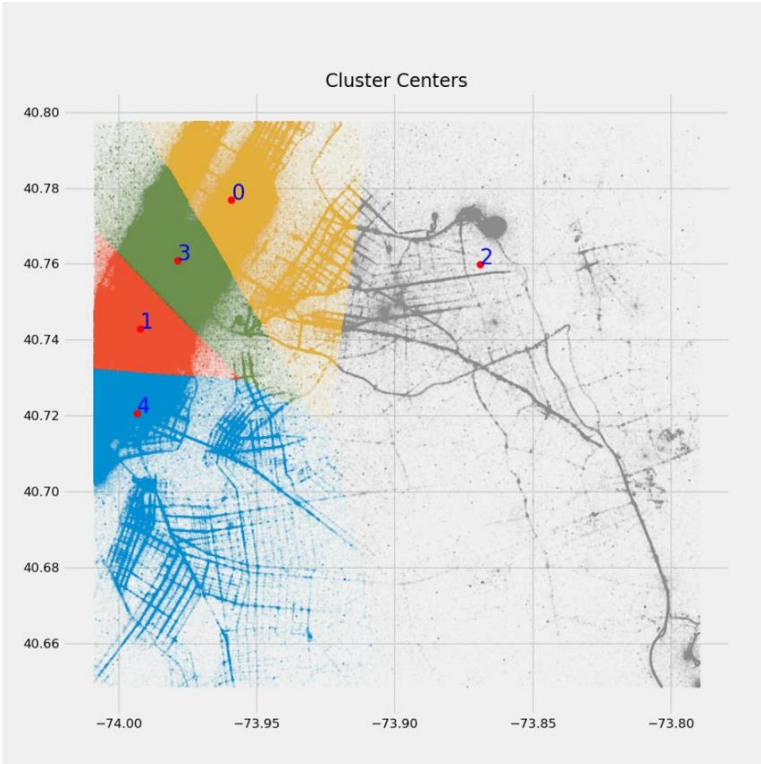
並使用 K means 分出 10 群，最多 pickup 的區域依序為編號 8 1 3 0 4



Cluster 編號/該 cluster 的個數

8	6789097
1	5938021
3	5531966
0	5509689
4	5302501
6	5150738
7	2973886
2	824951
5	550876
9	106215

只分成 5 群也能看出類似結果



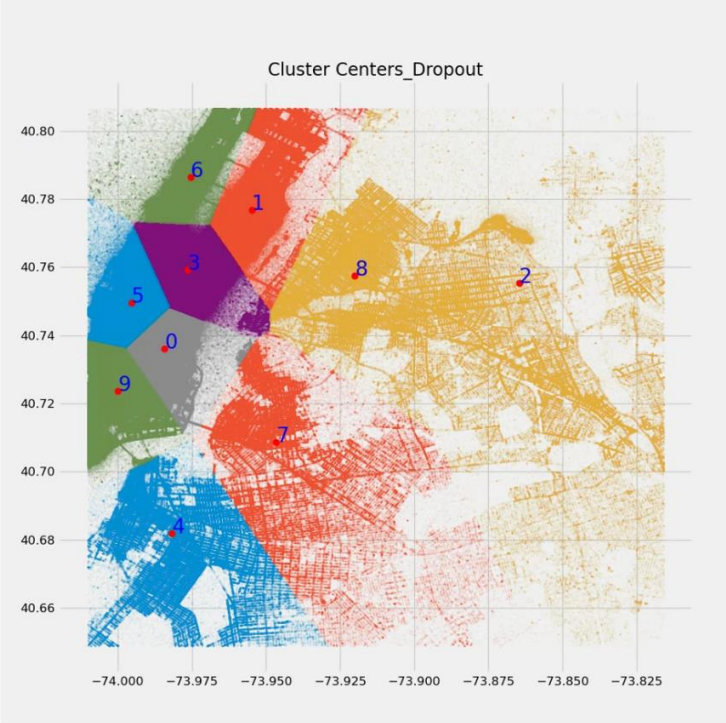
Cluster 編號/該 cluster 的個數

3	13109293
1	10552947
0	7861323
4	6201787
2	952590

- Drop-offs

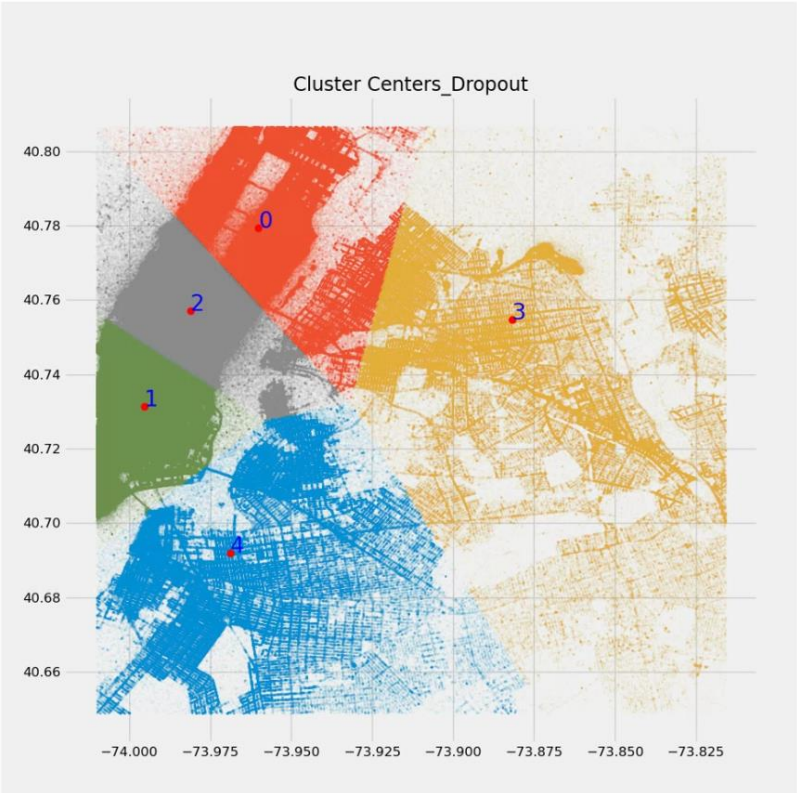
  使用同 Pickups 的方法

  使用 K means 分出 10 群，最多 dropour 的區域依序為編號



3	10327891
5	6082234
1	5762296
0	5170846
9	4767651
6	3560400
4	889557
2	758775
8	683408
7	636298

只分成 5 群也能看出類似結果



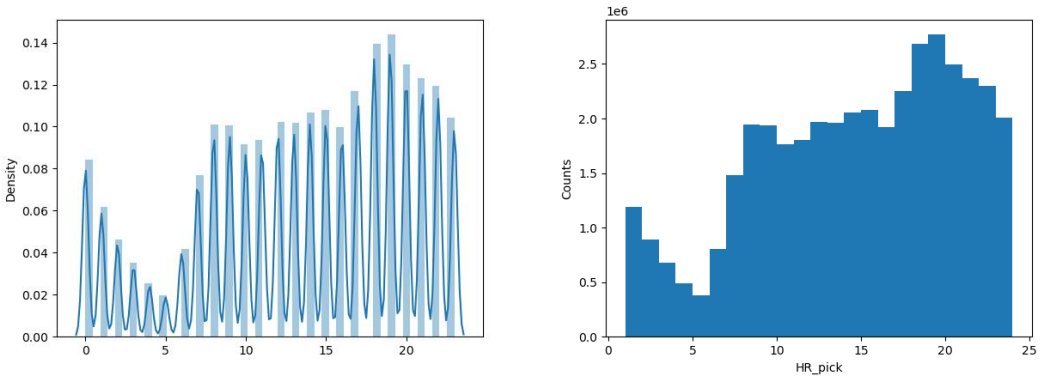
Cluster 編號/該 cluster 的個數

2	15681290
1	10945078
0	9328386
4	1487433
3	1197169

Q2: When are the **peak hours and off-peak hours** for taking a taxi?

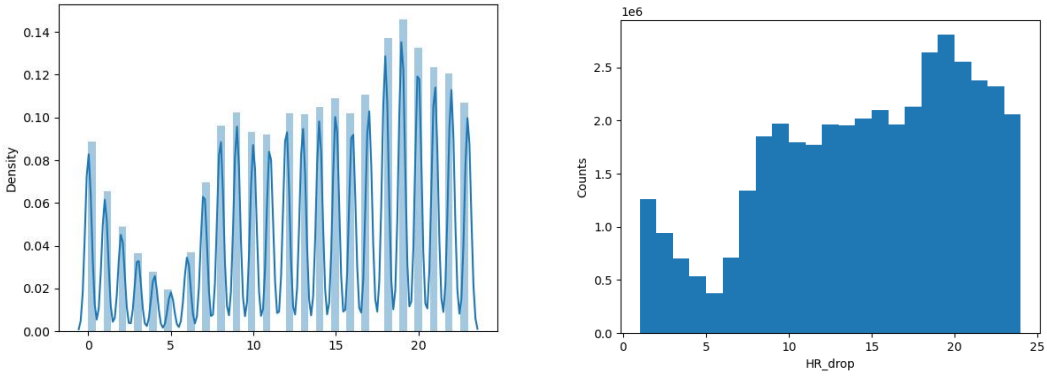
■ hint: You can count the number of pickups in different hours of day.

✓ 使用 pickup 時間觀察



約在 18-20 這個時間為 peak hours ，4、5 點為 off-peak 時段

✓ 使用 dropout 時間觀察

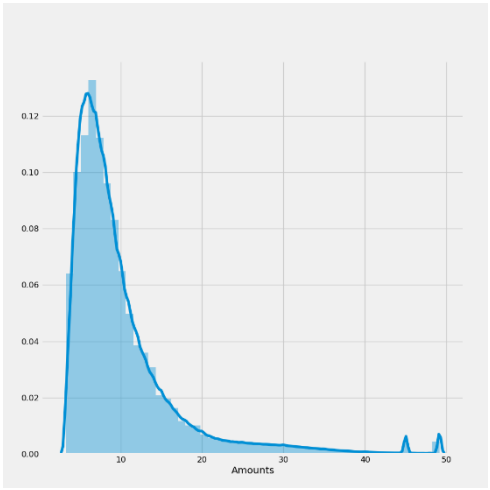


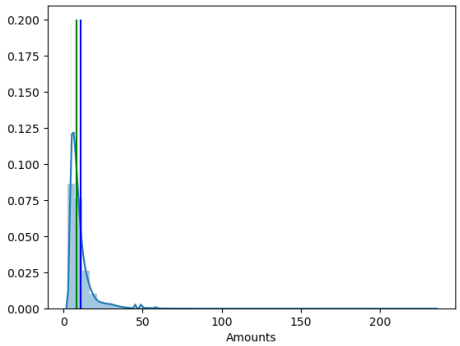
發現結果會是類似的，但 dropout 時間在 19-23 點這個區段又更集中

Q3: What are the differences between big and small total amounts when taking a taxi?

■ hint: First, you should define what big and small total amounts are. And then, you should point out the difference between them. You should at least observe the results of Q1 and Q2

- 首先用直方圖觀察 amounts 的大致趨勢
  (有先去除頭尾各 1%的極端資料)

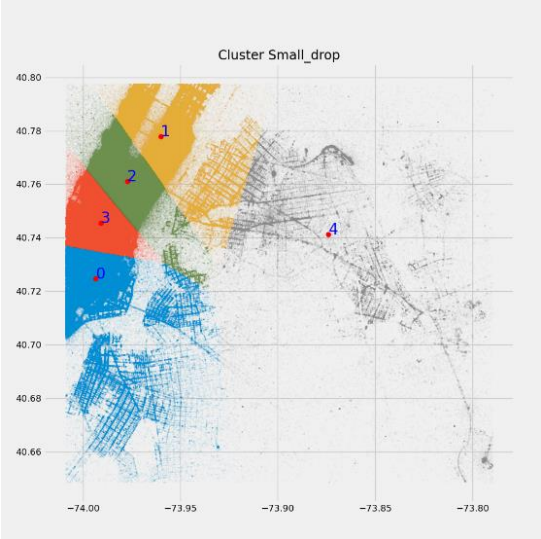


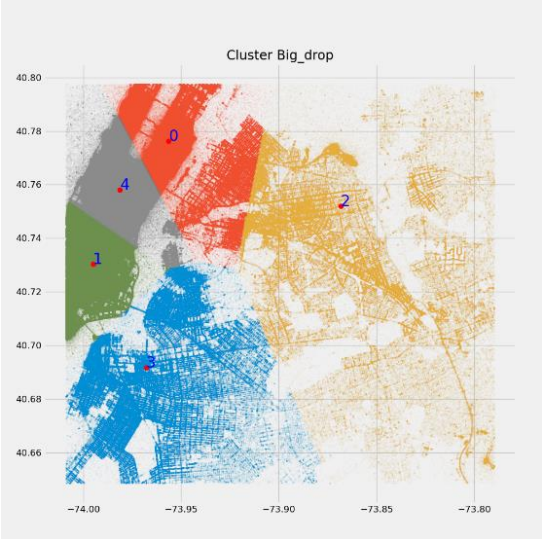


其中金額的 50%(綠線) = 8.099999999999998, 70%(藍線) = 10.7, 90% = 17.9

雖然大金額/小金額兩部分比例會不太一樣，但為了更好的觀察趨勢，將金額大和金額小的界線訂在 10.7 (70%)

✓ 下車位置 drop out





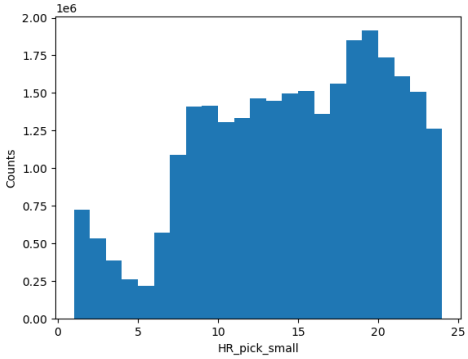
Small total amounts

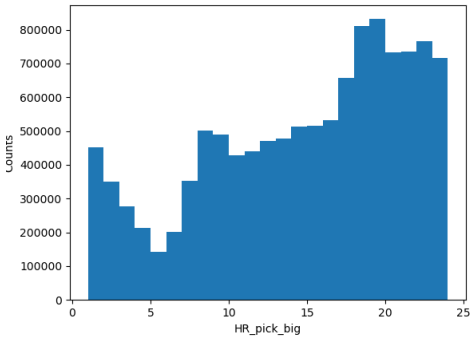
Big total amounts

金額小的位置較集中，金額大的則散布更廣，也有更多在東側地區下車。

✓ 尖峰/離峰時段

■ Pickup time



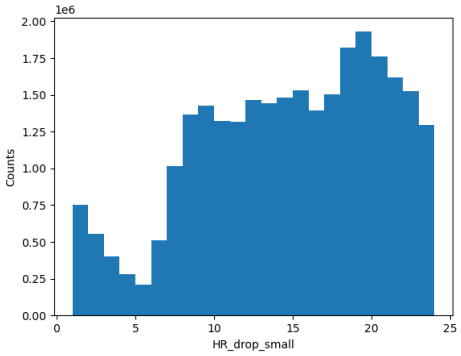


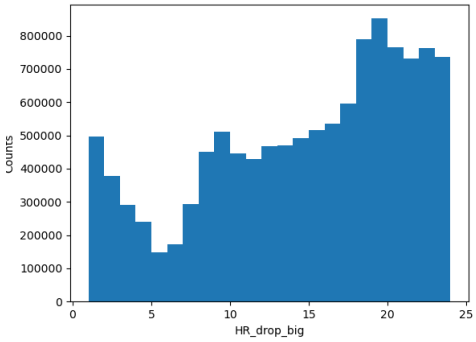
Small total amounts　　Big total amounts

小金額的部分在較晚時段(20~24 點)人數會有明顯逐步減少的趨勢

但大金額在同樣時段人數維持的差不多，且尖峰主要都集中在這個時段

■ Dropoff time





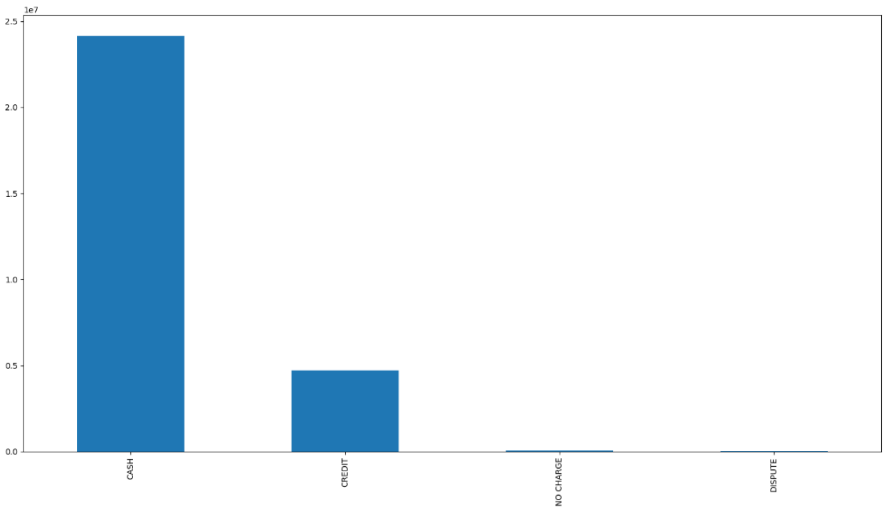
Small total amounts　　Big total amounts

小金額的部分在 8~16 點人數維持的差不多，和後續高峰比也沒有太大差距

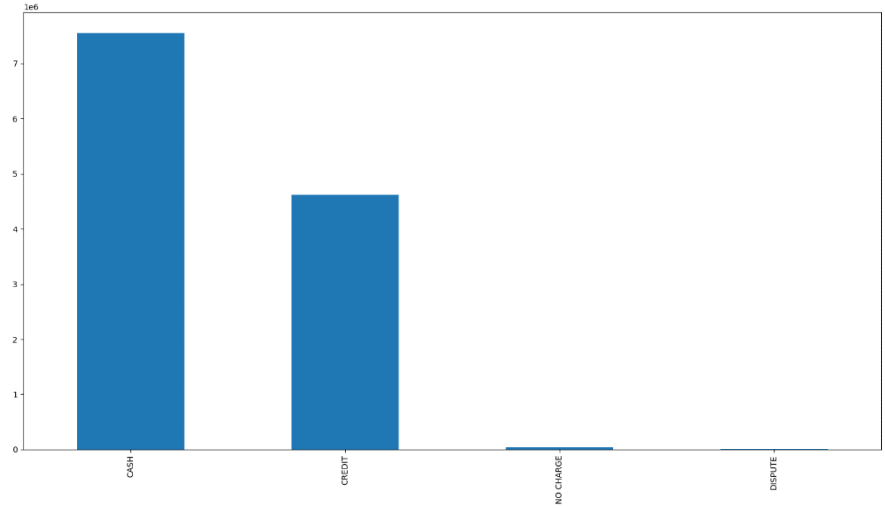
大金額在這個時段比例相對較少，人數要一直到 19 點左右才有急遽的增加

✓ 付款方式 payment type

Small amount



Big amount



在小金額的部分，現金支付多了不少。信用卡支付的約為現金的 1/6。
大金額的部分信用卡支付的約為現金的 1/2，相對比例增加許多。

✓ Difficulties

- 光是讀取資料的時間要花費數分鐘很長，資料條件篩選也同樣很久
  一個月大約有千萬筆 data，所以光是讀取 csv 檔都很耗 RAM (原先使用 Colab 開啟會顯示 RAM 空間已用完(平台提供應<13GB))
  => 可以使用 random 的方式只看大致趨勢。我試用隨機挑選點(50 萬筆)來替代真實的各個點，結果對照下我認為已能充分具代表性
  (不過後來改用其他設備因此不需要了)

- ✓ The scale of data
  三個 CSV 檔分別為 2.36GB 2.24GB 2.42GB
  共 4 千多萬筆資料，一筆資料有 18 個欄位

- ✓ Analytical tools
  Python 3.7
  pandas 1.3.4
  matplotlib 3.5.1
  scikit-learn 1.0.2

- ✓ Spec of the platform
  處理器　Intel(R) Core(TM) i7-6700 CPU @ 3.40GHz　3.40 GHz
  已安裝記憶體(RAM)　16.0 GB
  系統類型　64 位元作業系統

- ✓ Reference:
  https://www.kaggle.com/drgilermo/dynamics-of-new-york-city-animation
  https://www.kaggle.com/selfishgene/yellow-cabs-tell-the-story-of-new-york-city

- ✓ My code and results
  https://github.com/f74066357/Bigdata_analyze/tree/main/Hw1