Name: Yixin Zheng
Course: Si 618

Project I Report

Airbnb dataset exploration

Motivations

Owing to the personal experience in Airbnb in New York. I am quite curious about the elements that contribute to price. There are three questions I'm going to explore. What's the relationship between price of Airbnb and neighborhood? What's the relationship between price of Airbnb and unemployment? What's the relationship between price of Airbnb and income per capita?

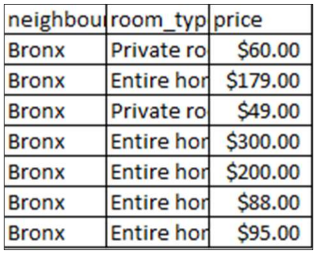
Datasets

1. Airbnb Detailed Listings Data for New York City

Source:

http://data.insideairbnb.com/united-states/ny/new-york-city/2015-12-02/data/listings.csv.gz

The dataset is a csv file covered the period in 2015.
It is downloaded from Inside Airbnb site. It contains field such as id of the hos t, urls, address, neighbourhood_group_cleansed, room types, price, reviews etc. I chose to use neighbourhood_group_cleansed, room type, and price attribute.



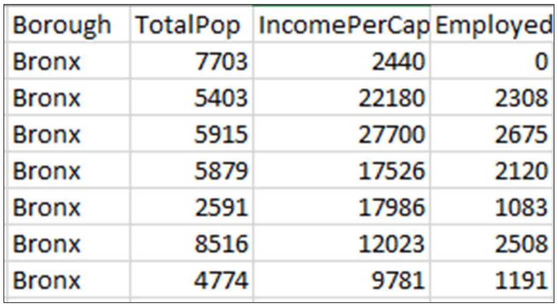
neighbou	room_typ	price
Bronx	Private ro	$60.00
Bronx	Entire hor	$179.00
Bronx	Private ro	$49.00
Bronx	Entire hor	$300.00
Bronx	Entire hor	$200.00
Bronx	Entire hor	$88.00
Bronx	Entire hor	$95.00

2. New York City census tracts

Source:

https://www.kaggle.com/muonneutrino/new-york-city-census-data?select=nyc_census_tracts.csv

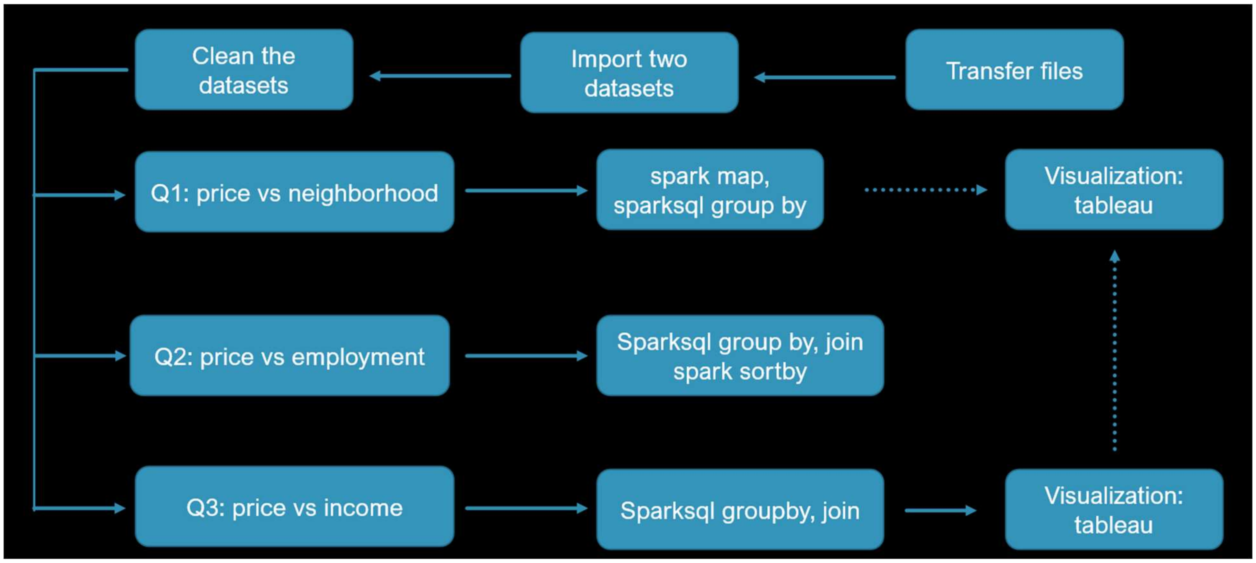
The dataset is a csv file covered the period in 2015.
This file contains a selection of census data taken from the ACS DP03 and DP05 tables. It contains fields such as total population, racial/ethnic demographic information, employment and commuting characteristics, income per cap and more are so on. I chose to use borough and IncomePerCap.



Borough	TotalPop	IncomePerCap	Employed
Bronx	7703	2440	0
Bronx	5403	22180	2308
Bronx	5915	27700	2675
Bronx	5879	17526	2120
Bronx	2591	17986	1083
Bronx	8516	12023	2508
Bronx	4774	9781	1191

Name: Yixin Zheng
Course: Si 618

Data Manipulation

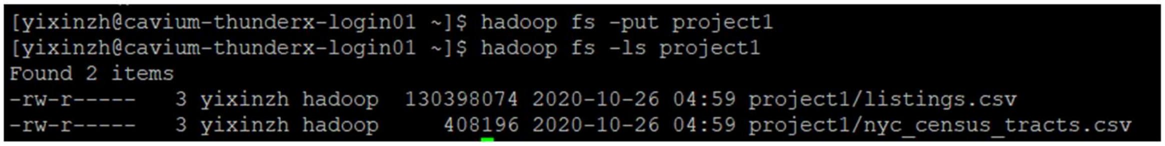


1. Import

Airbnb Detailed Listings Data for New York City is a big dataset with 34376 rows. To transfer it to the server site, I waited until it is total upload, avoiding missing rows because of missing file. I checked the size of two csv files.

```
$hadoop fs -put project1

$hadoop fs -ls project 1
```



```
[yixinzh@cavium-thunderx-login01 ~]$ hadoop fs -put project1
[yixinzh@cavium-thunderx-login01 ~]$ hadoop fs -ls project1
Found 2 items
-rw-r-----   3 yixinzh hadoop  130398074 2020-10-26 04:59 project1/listings.csv
-rw-r-----   3 yixinzh hadoop     408196 2020-10-26 04:59 project1/nyc_census_tracts.csv
```

Run pyspark to get into the shell. Imported all the module needed.

```
$ pyspark

>>>from pyspark import SparkConf, SparkContext

>>>from pyspark.sql import SQLContext
>>>sc = SparkContext(appName="PySparksi618f19project1")
>>>sqlContext=SQLContext(sc)
```

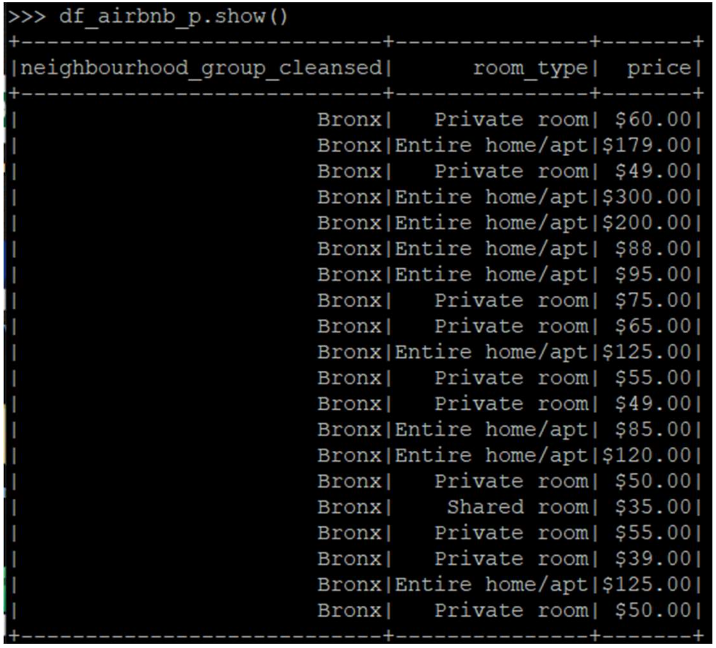
This csv file contains newlines as well as comma in between double quotes, which shall be escaped while read, and then registered it as table. New York City census tracts was read in the same way.

```
>>>df_airbnb=sqlContext.read.option("multiline", "true").option("quote", '"').option("escape", "\\").op
tion("escape", '"').csv('project1/listings.csv', header=True)
>>>df_airbnb.registerTempTable('airbnb')
>>>df = spark.createDataFrame(df_airbnb_p_rdd).toDF("neighbourhood_group_cleansed", "room_ty
pe","price")
>>>df.registerTempTable("nrp")
```

2. Clean

There are several missing values inside the dataset. From Airbnb Detailed Listings Data for New York City dataset, I selected neighbourhood_group_cleansed, room_type, and price, dropping the missing value in those three fields by sparksql. Also dropped the missing value in New York City census tracts dataset.

```
df_airbnb_p=sqlContext.sql("select neighbourhood_group_cleansed, room_type,price from airbnb wh
ere neighbourhood_group_cleansed is not null and price is not null and room_type is not null")
```

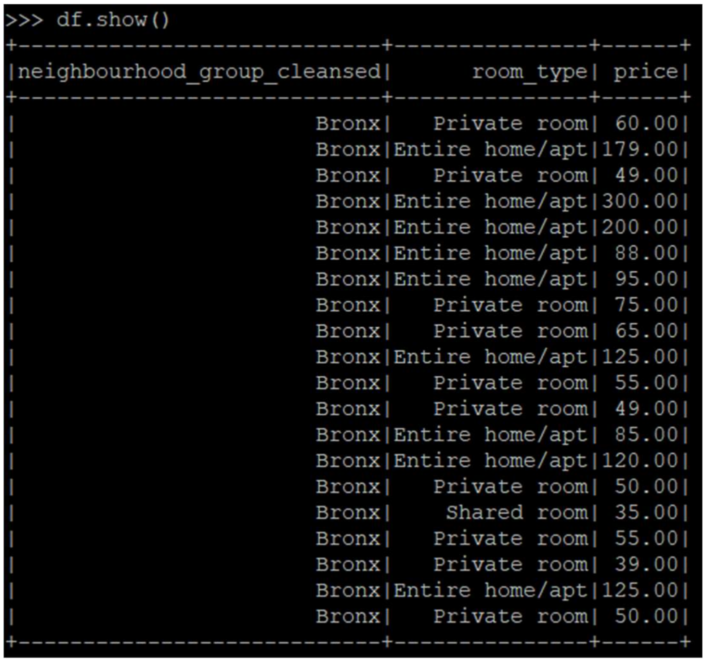


```
>>> df_airbnb_p.show()
+----------------------------+---------------+-------+
|neighbourhood_group_cleansed|      room_type|  price|
+----------------------------+---------------+-------+
|                       Bronx|   Private room| $60.00|
|                       Bronx|Entire home/apt|$179.00|
|                       Bronx|   Private room| $49.00|
|                       Bronx|Entire home/apt|$300.00|
|                       Bronx|Entire home/apt|$200.00|
|                       Bronx|Entire home/apt| $88.00|
|                       Bronx|Entire home/apt| $95.00|
|                       Bronx|   Private room| $75.00|
|                       Bronx|   Private room| $65.00|
|                       Bronx|Entire home/apt|$125.00|
|                       Bronx|   Private room| $55.00|
|                       Bronx|   Private room| $49.00|
|                       Bronx|Entire home/apt| $85.00|
|                       Bronx|Entire home/apt|$120.00|
|                       Bronx|   Private room| $50.00|
|                       Bronx|    Shared room| $35.00|
|                       Bronx|   Private room| $55.00|
|                       Bronx|   Private room| $39.00|
|                       Bronx|Entire home/apt|$125.00|
|                       Bronx|   Private room| $50.00|
+----------------------------+---------------+-------+
```

To get rid of the “$” before numbers in price’s value, I created RDD to map the value of price from the dataframe that has been selected and cleaned, and then created dataframe from the RDD for the sparksql calculation in the next step.

```
>>>df_airbnb_p_rdd=df_airbnb_p.rdd.map(lambda x: (x[0].strip(),x[1],x[2][1:].strip()))
>>>df = spark.createDataFrame(df_airbnb_p_rdd).toDF("neighbourhood_group_cleansed", "room_ty
pe","price")
```

```
>>>df.registerTempTable("nrp")
```



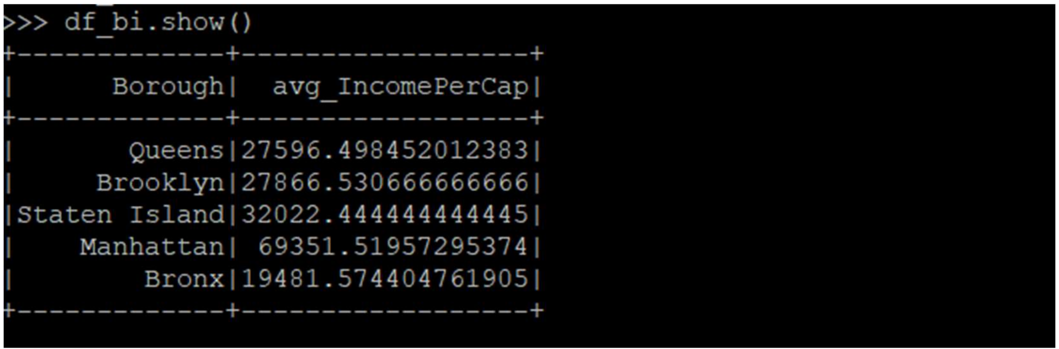
```
>>> df.show()
+--------------------------+---------------+------+
|neighbourhood_group_cleansed|      room_type| price|
+--------------------------+---------------+------+
|                       Bronx|   Private room| 60.00|
|                       Bronx|Entire home/apt|179.00|
|                       Bronx|   Private room| 49.00|
|                       Bronx|Entire home/apt|300.00|
|                       Bronx|Entire home/apt|200.00|
|                       Bronx|Entire home/apt| 88.00|
|                       Bronx|Entire home/apt| 95.00|
|                       Bronx|   Private room| 75.00|
|                       Bronx|   Private room| 65.00|
|                       Bronx|Entire home/apt|125.00|
|                       Bronx|   Private room| 55.00|
|                       Bronx|   Private room| 49.00|
|                       Bronx|Entire home/apt| 85.00|
|                       Bronx|Entire home/apt|120.00|
|                       Bronx|   Private room| 50.00|
|                       Bronx|    Shared room| 35.00|
|                       Bronx|   Private room| 55.00|
|                       Bronx|   Private room| 39.00|
|                       Bronx|Entire home/apt|125.00|
|                       Bronx|   Private room| 50.00|
+--------------------------+---------------+------+
```

3. Group

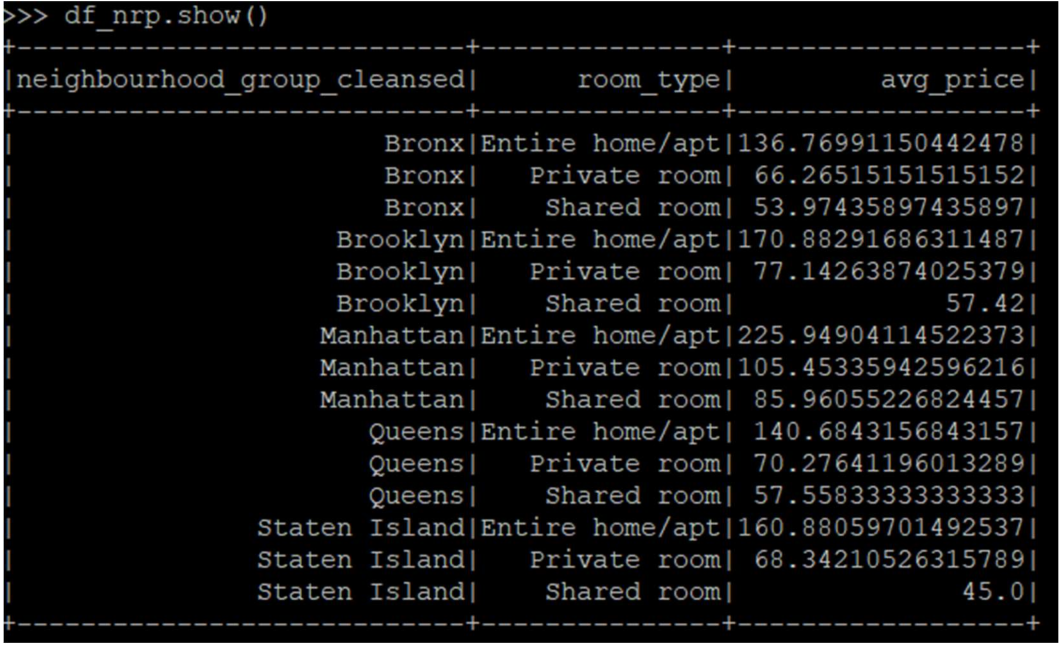
Before joining the two datasets, I use sparksql to calculated the mean of IncomePerCap in New York City census tracts by grouping the borough, the mean of price in Airbnb Detailed Listings Data for New York City dataset by grouping the neighbourhood_group_cleansed. Rename the aggregation values.

```
>>>df_bi=sqlContext.sql('''select Borough, TotalPop, IncomePerCap*TotalPop as Income, employed fr
om census                    where Borough is not null and TotalPop is not null and IncomePerCap is not n
ull and Unemployment is not null

''')
>>>df_bi.registerTempTable('bi')
>>>df_nrp=sqlContext.sql("""select neighbourhood_group_cleansed, room_type, mean(cast(price as f
loat)) as avg_price from nrp
            group by neighbourhood_group_cleansed,room_type
            order by neighbourhood_group_cleansed,room_type
""")
>>>df_nrp.registerTempTable("nrp")
```



```
>>> df_bi.show()
+-------------+------------------+
|      Borough|  avg_IncomePerCap|
+-------------+------------------+
|       Queens|27596.498452012383|
|     Brooklyn|27866.530666666666|
|Staten Island|32022.444444444445|
|    Manhattan| 69351.51957295374|
|        Bronx|19481.574404761905|
+-------------+------------------+
```

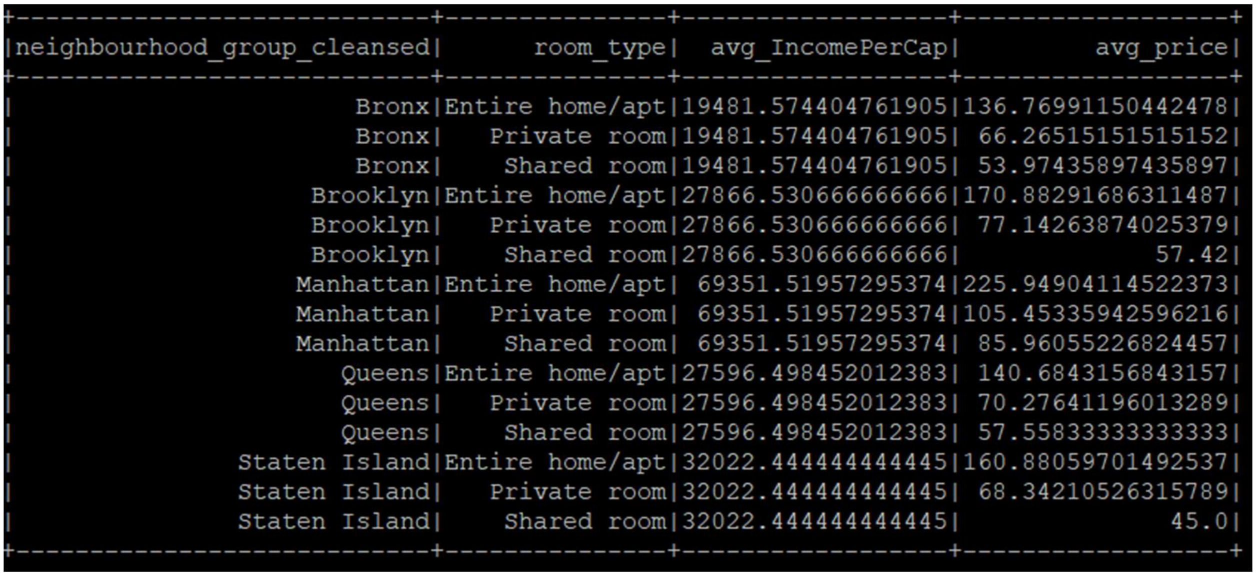


```
>>> df_nrp.show()
+----------------------------+---------------+------------------+
|neighbourhood_group_cleansed|      room_type|         avg_price|
+----------------------------+---------------+------------------+
|                       Bronx|Entire home/apt|136.76991150442478|
|                       Bronx|   Private room| 66.26515151515152|
|                       Bronx|    Shared room| 53.97435897435897|
|                    Brooklyn|Entire home/apt|170.88291686311487|
|                    Brooklyn|   Private room| 77.14263874025379|
|                    Brooklyn|    Shared room|             57.42|
|                   Manhattan|Entire home/apt|225.94904114522373|
|                   Manhattan|   Private room|105.45335942596216|
|                   Manhattan|    Shared room| 85.96055226824457|
|                      Queens|Entire home/apt| 140.6843156843157|
|                      Queens|   Private room| 70.27641196013289|
|                      Queens|    Shared room| 57.55833333333333|
|               Staten Island|Entire home/apt|160.88059701492537|
|               Staten Island|   Private room| 68.34210526315789|
|               Staten Island|    Shared room|              45.0|
+----------------------------+---------------+------------------+
```

4. Join

The two datasets was joined by sparksql. The field neighbourhood_group_cleansed and the field Borough are of the same value， so they are joined together.

```
>>>df=sqlContext.sql("""select nrp.neighbourhood_group_cleansed, nrp.room_type, bi.
IncomePerCap, nrp.avg_price
        from nrp
        join bi on bi.Borough == nrp.neighbourhood_group_cleansed
        order by nrp.neighbourhood_group_cleansed, nrp.room_type
                                   """)
```



```
+----------------------------+---------------+------------------+------------------+
|neighbourhood_group_cleansed|      room_type|  avg_IncomePerCap|         avg_price|
+----------------------------+---------------+------------------+------------------+
|                       Bronx|Entire home/apt|19481.574404761905|136.76991150442478|
|                       Bronx|   Private room|19481.574404761905| 66.26515151515152|
|                       Bronx|    Shared room|19481.574404761905| 53.97435897435897|
|                    Brooklyn|Entire home/apt|27866.530666666666|170.88291686311487|
|                    Brooklyn|   Private room|27866.530666666666| 77.14263874025379|
|                    Brooklyn|    Shared room|27866.530666666666|             57.42|
|                   Manhattan|Entire home/apt| 69351.51957295374|225.9490114522373|
|                   Manhattan|   Private room| 69351.51957295374|105.45335942596216|
|                   Manhattan|    Shared room| 69351.51957295374| 85.96055226824457|
|                      Queens|Entire home/apt|27596.498452012383| 140.6843156843157|
|                      Queens|   Private room|27596.498452012383| 70.27641196013289|
|                      Queens|    Shared room|27596.498452012383| 57.55833333333333|
|               Staten Island|Entire home/apt|32022.444444444445|160.88059701492537|
|               Staten Island|   Private room|32022.444444444445| 68.34210526315789|
|               Staten Island|    Shared room|32022.444444444445|              45.0|
+----------------------------+---------------+------------------+------------------+
```

5. Export

In order to do the visualization, I exported and cat the csv file.

```
>>>df.write.format('csv').option('delimiter','\t').save('project1_1')
>>>exit()
$ hadoop fs -cat project1_1/part-* > project1.csv
```

And then the got the file to my local.

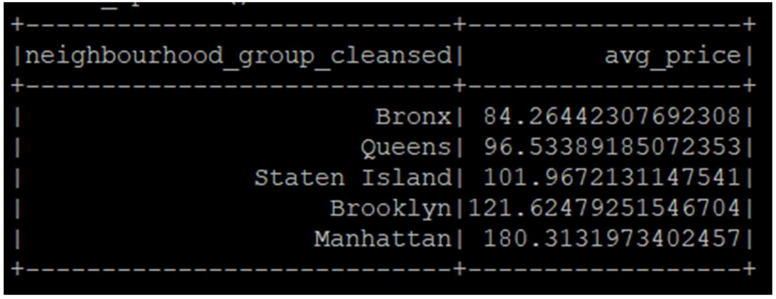
Visualization

By importing the csv file to tableau, I encoded neighborhood and room type in x-axis as independent variable, average price in the y-axis as outcome variable. Also parallelly shooing the income per cap. The room type are also double encoded in color for better identifying.

1. What's the relationship between price of Airbnb and neighborhood?

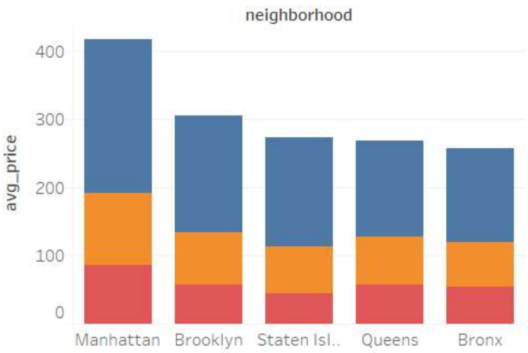
Price and neighborhood are in the same dataset. So I grouped by the neighborhood and calculated the average of price by sparksql. Also use the spark "map()" method. Part of the visualization data is from Q3.

```
>>>df_airbnb_p_rdd=df_airbnb_p.rdd.map(lambda x: (x[0].strip(),x[1],x[2][1:].strip()))
>>>df = spark.createDataFrame(df_airbnb_p_rdd).toDF("neighbourhood_group_cleansed", "room_ty
pe","price")
>>>df.registerTempTable("nrp")
>>>df_np=sqlContext.sql("""select neighbourhood_group_cleansed, mean(cast(price as float)) as avg_
price from nrp
 group by neighbourhood_group_cleansed
order by avg_price""")
```

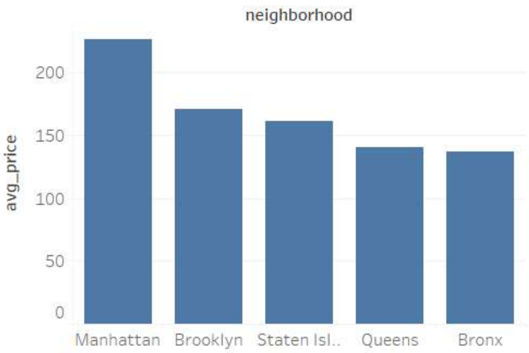


```
+--------------------------+------------------+
|neighbourhood_group_cleansed|         avg_price|
+--------------------------+------------------+
|                     Bronx| 84.26442307692308|
|                    Queens| 96.53389185072353|
|             Staten Island| 101.9672131147541|
|                  Brooklyn|121.62479251546704|
|                 Manhattan| 180.3131973402457|
+--------------------------+------------------+
```

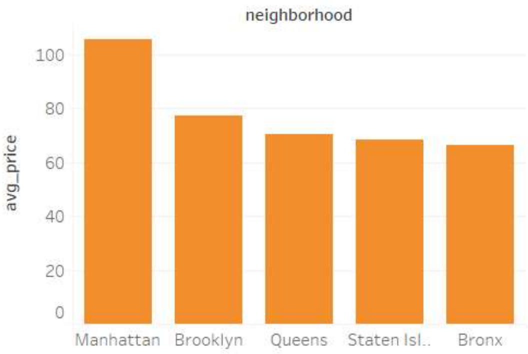
Average of three room types



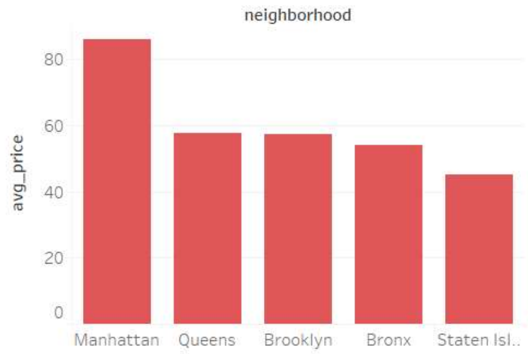
Entire house/apt



Private room



Shared room



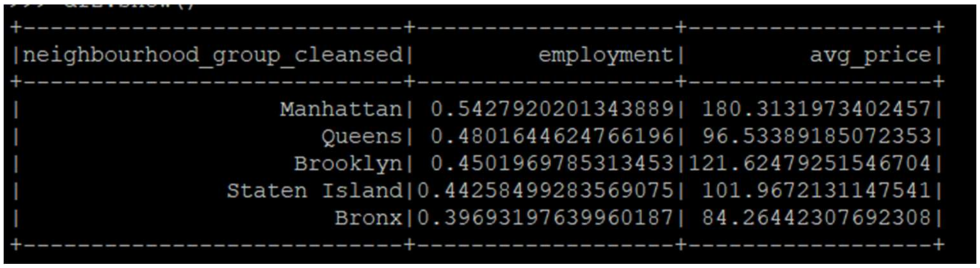
2. What's the relationship between price of Airbnb and employment?

The table from Q1 was kept for joining in Q2. To get the value of employment, I group the dataset by Borough to get the sum of income and total population and employed, them divided them. One thing shall be notice is that this cannot be simply calculate by method "mean" because it is based on the population. The computation techniques I used here is sparksql "group by" and "join" as well as spark "sortBy".

The employment and price seem not that related.

```
>>>df_bi=sqlContext.sql('''select Borough, SUM(Income)/SUM(TotalPop) as IncomePerCap, SUM(empl
oyed)/SUM(TotalPop) as employment from bi
            group by Borough''')
```

```
>>>df_bi.registerTempTable('bi')
>>>df2=sqlContext.sql("""select np.neighbourhood_group_cleansed, bi.employment, np.avg_price
            from np
            join bi on bi.Borough == np.neighbourhood_group_cleansed
            """)
>>>df2_rdd=df2.rdd.sortBy(lambda x:(x[1],x[2]),ascending=False)
>>>df2 = spark.createDataFrame(df2_rdd).toDF("neighborhood","employment","avg_price")
```



```
+--------------------------+------------------+------------------+
|neighbourhood_group_cleansed|        employment|         avg_price|
+--------------------------+------------------+------------------+
|                 Manhattan| 0.542792020134388 9| 180.3131973402457|
|                    Queens| 0.4801644624766196| 96.53389185072353|
|                  Brooklyn| 0.4501969785313453|121.62479251546704|
|             Staten Island|0.44258499283569075| 101.9672131147541|
|                     Bronx|0.39693197639960187| 84.26442307692308|
+--------------------------+------------------+------------------+
```

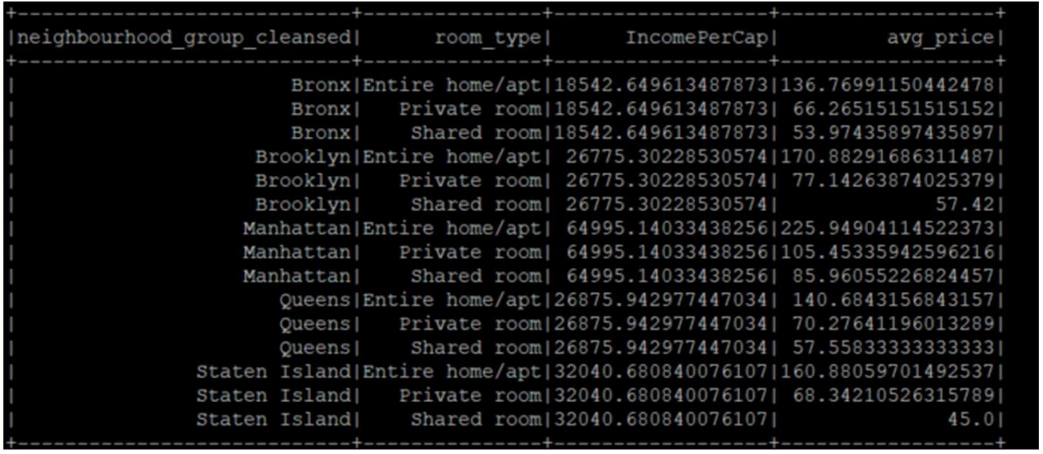
There is no obvious relationship between price of Airbnb and room type count.

3. What’s the relationship between price of Airbnb and income per capita?

The table from Q2 was kept for joining in Q3. As what I previous mentioned in the join in data manipulation part,the two datasets was joined by sparksql on the field neighbourhood_group_cleansed and the field Borough.

All in all, the income per cap and price of Airbnb has positive relationship. But this is not absolute, Staten Island is an exception of all the three-room type. It will be interesting if dive deep into this area to explore the reason.

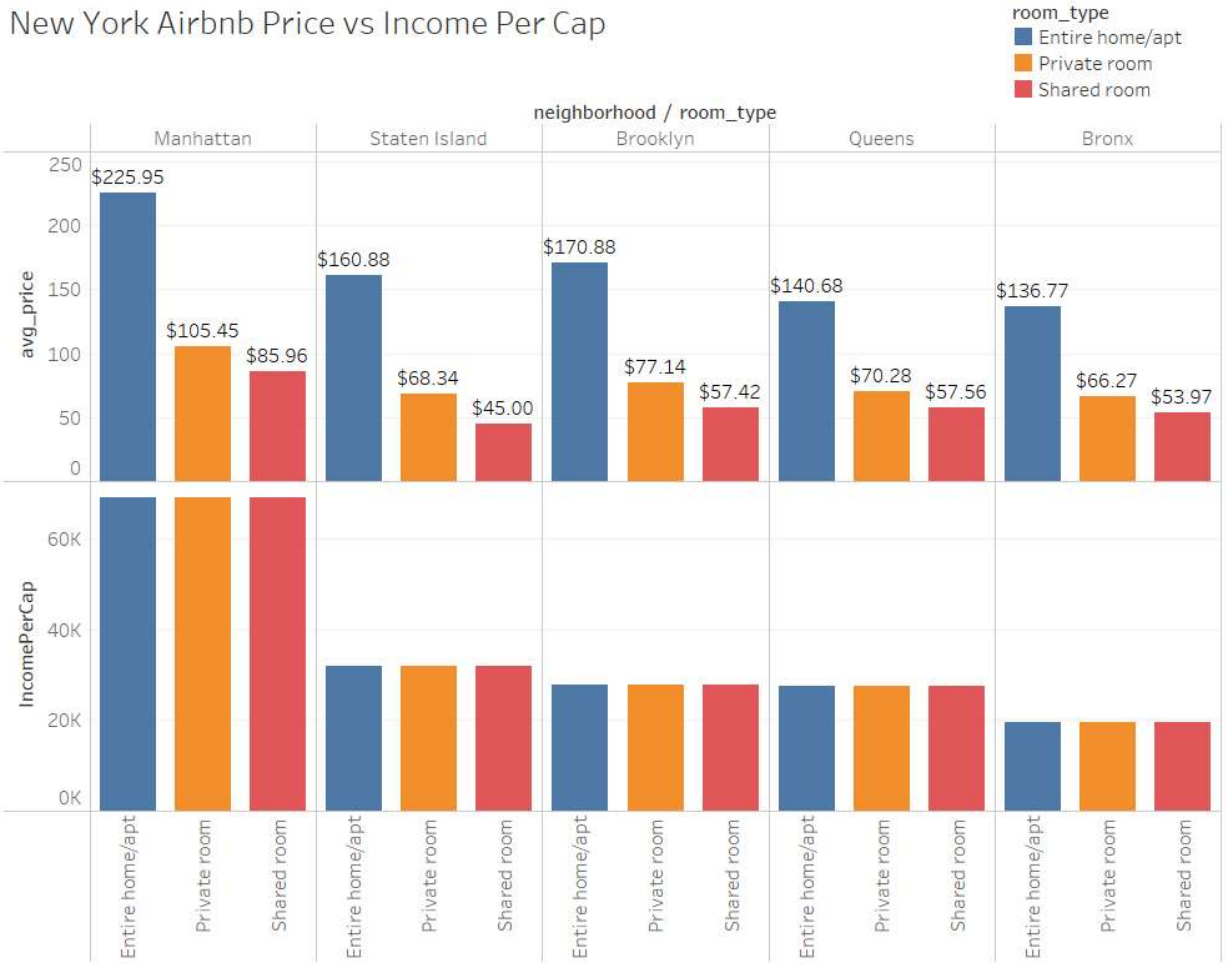
```
>>>df=sqlContext.sql("""select nrp.neighbourhood_group_cleansed, nrp.room_type, bi.IncomePerCap
, nrp.avg_price  from nrp
            join bi on bi.Borough == nrp.neighbourhood_group_cleansed
            order by nrp.neighbourhood_group_cleansed, nrp.room_type
```



```
+----------------------------+---------------+------------------+------------------+
|neighbourhood_group_cleansed|      room_type|      IncomePerCap|         avg_price|
+----------------------------+---------------+------------------+------------------+
|                       Bronx|Entire home/apt|18542.649613487873|136.76991150442478|
|                       Bronx|   Private room|18542.649613487873| 66.26515151515152|
|                       Bronx|    Shared room|18542.649613487873| 53.97435897435897|
|                    Brooklyn|Entire home/apt| 26775.30228530574|170.88291686311487|
|                    Brooklyn|   Private room| 26775.30228530574| 77.14263874025379|
|                    Brooklyn|    Shared room| 26775.30228530574|             57.42|
|                   Manhattan|Entire home/apt| 64995.14033438256|225.94904114522373|
|                   Manhattan|   Private room| 64995.14033438256|105.45335942596216|
|                   Manhattan|    Shared room| 64995.14033438256| 85.96055226824457|
|                      Queens|Entire home/apt|26875.942977447034| 140.6843156843157|
|                      Queens|   Private room|26875.942977447034| 70.27641196013289|
|                      Queens|    Shared room|26875.942977447034| 57.55833333333333|
|               Staten Island|Entire home/apt|32040.680840076107|160.88059701492537|
|               Staten Island|   Private room|32040.680840076107| 68.34210526315789|
|               Staten Island|    Shared room|32040.680840076107|              45.0|
+----------------------------+---------------+------------------+------------------+
```

```
""")
```

Name: Yixin Zheng
Course: Si 618



Challenges

One challenge I encounter was importing Airbnb Detailed Listings Data for New York City dataset. Firstly, it is a large dataset, when first time I tried to read it, it only showed 490 rows, which was far less than 34376 rows in total. This is because while I transferred the file to my login node, only part of the file has been transferred. Secondly, the file contains many newlines and comma in double quotes. If I don’t deal with it, the value of column will be mixed up.

Another challenge is considering how to use aggregate function in sparksql. For the data in Airbnb Detailed Listings Data for New York City, rows represents listing, so I can use mean to calculate the average value; while for the data in census, each row represents a census tract with many people, so I shall use sum and division.