Postgraduate Certificate in Cloud Native Computing
Postgraduate Certificate in Software Design with Artificial Intelligence

Applied Scripting Languages

Assignment 1

Student ID: A00267813

Brief Description

Airbnb is worldwide online marketplace to lend individuals or small business's spare space in competitive manner. New York is one of the most popular and extremely expensive tourist destinations in the world. Hence New York was one of earliest cities where Airbnb started their service. The purpose of this assignment is to analyse factors which influence the price of Airbnb New York listings 2019.

Contents

Figures

Introduction

According to 1 Wikipedia, Airbnb, Inc. is an online marketplace for arranging or offering lodging, primarily homestays, or tourism experiences. The company does not own any of the real estate listings, nor does it host events; it acts as a broker, receiving commissions from each booking. The company is based in San Francisco, California, United States.

In 2008, the founders of AirBnb officially launched their business idea in a site called Airbedandbreakfast.com, then in 2009 they shorten the name Airbnb.com. Just after 2 years, Airbnb recorded 1 millionth booked, and after another year, in February 2012 they announced their 5 millionth and just after another 5 months, doubling numbers Airbnb announced 10 millionth booking.

Main features of the Airbnb are the host who list their lodging with their own price and tenet can rent the lodging through Airbnb, and both tenant and host can share their experience through reviews. So competitive price and tenant's reviews plays a major role for host business.

The purpose of this assignment is to analyse how tenants' reviews corelate to price and other factors which influence the price of Airbnb New York listings 2019

[1] https://en.wikipedia.org/wiki/Airbnb

Data

[2] Insideairbnb publishes various data related to their business for different countries and different cities. The data related to listing in New York 2019 can be downloaded from here http://insideairbnb.com/get-the-data.html

The original dataset has following files in CSV format or archive. As primary focus of the assignment is to analyse price variation, it was decided to analyse listings.csv.

12 September, 2019	New York City	listings.csv.gz	Detailed Listings data for New York City
12 September, 2019	New York City	calendar.csv.gz	Detailed Calendar Data for listings in New York City
12 September, 2019	New York City	reviews.csv.gz	Detailed Review Data for listings in New York City
12 September, 2019	New York City	listings.csv	Summary information and metrics for listings in New York City (good for visualisations).
12 September, 2019	New York City	reviews.csv	Summary Review data and Listing ID (to facilitate time based analytics and visualisations linked to a listing).
N/A	New York City	neighbourhoods.csv	Neighbourhood list for geo filter. Sourced from city or open source GIS files.
N/A	New York City	neighbourhoods.geojson	GeoJSON file of neighbourhoods of the city.

[2] http://insideairbnb.com/

Further, various data exploratory efforts related to this dataset can be found in kaggle [3]https://www.kaggle.com/dgomonov/new-york-city-airbnb-open-data

Data Fields

listings.csv file consists 49080 records with 16 following fields

Field	Type	Description	Considered
id	Integer	AirBnB specific unique number. This field will be ignored for analysis	No
name	Text	Name of the listing	No
host_id	Integer	Host Id assigned by AirBnB	Yes
host_name	Text	Name of the host	Yes
neighbourhood_group	Text	Neighbourhood group where room located	Yes
neighbourhood	Text	Neighbourhood group where room located	Yes
latitude	Real	Latitude coordinate of the room	Yes
longitude	Real	longitude coordinate of the room	Yes
room_type	Text	Room Type	Yes
price	Real	Price of the room	Yes
minimum_nights	Integer	Minimum nights need to be booked	Yes
number_of_reviews	Integer	Total Number of Reviews for the room	Yes
last_review	Date	Last Review date for the room	Yes
reviews_per_month	Real	Average number of reviews per month	Yes
calculated_host_listings_count	Integer	Host listing count	Yes
availability_365	Integer	Availability of the room per year	Yes

[3] https://www.kaggle.com/dgomonov/new-york-city-airbnb-open-data

Sample Data

id	name	host_id	host_nam	neighbourhood_group	neighbourhood	latitude	longitude	room_type	price	minimum	number_c	last_review	reviews_p	calculated	availability_365
2539	Clean & q	2787	John	Brooklyn	Kensington	40.64749	-73.9724	Private room	149	1	9	10/19/2018	0.21	6	365
2595	Skylit Mid	2845	Jennifer	Manhattan	Midtown	40.75362	-73.9838	Entire home/apt	225	1	45	5/21/2019	0.38	2	355
3647	THE VILLA	4632	Elisabeth	Manhattan	Harlem	40.80902	-73.9419	Private room	150	3	0			1	365
3831	Cozy Entir	4869	LisaRoxan	Brooklyn	Clinton Hill	40.68514	-73.9598	Entire home/apt	89	1	270	7/5/2019	4.64	1	194
5022	Entire Apt	7192	Laura	Manhattan	East Harlem	40.79851	-73.944	Entire home/apt	80	10	9	11/19/2018	0.1	1	0
5099	Large Cozy	7322	Chris	Manhattan	Murray Hill	40.74767	-73.975	Entire home/apt	200	3	74	6/22/2019	0.59	1	129
5121	BlissArtsS	7356	Garon	Brooklyn	Bedford-Stuyvesant	40.68688	-73.956	Private room	60	45	49	10/5/2017	0.4	1	0
5178	Large Furr	8967	Shunichi	Manhattan	Hell's Kitchen	40.76489	-73.9849	Private room	79	2	430	6/24/2019	3.47	1	220
5203	Cozy Clea	7490	MaryEllen	Manhattan	Upper West Side	40.80178	-73.9672	Private room	79	2	118	7/21/2017	0.99	1	0
5238	Cute & Co	7549	Ben	Manhattan	Chinatown	40.71344	-73.9904	Entire home/apt	150	1	160	6/9/2019	1.33	4	188

Figure 1 Sample Data. Spreadsheet view

```
Printing first 10 lines of data set
+-----------------------------------------------+---------+------------------+---------------+----------+-----------+-----------------+-------+--------------+------------------+-------------+--------------------+-------------------------------+-------------+
|                     name                      | host_id | neighbourhood_group | neighbourhood | latitude | longitude |    room_type    | price | minimum_nights | number_of_reviews | last_review | reviews_per_month | calculated_host_listings_count | availabili |
+-----------------------------------------------+---------+------------------+---------------+----------+-----------+-----------------+-------+--------------+------------------+-------------+--------------------+-------------------------------+-------------+
| Clean & quiet apt home by the park            |  2787   |        1         |       1       |    1     |  40.647   |    -73.972      |  149  |      1       |        9         | 2018-10-19  |       0.210        |               6               |     365     |
| Skylit Midtown Castle                         |  2845   |        2         |       2       |    2     |  40.754   |    -73.984      |  225  |      1       |        45        | 2019-05-21  |       0.380        |               2               |     355     |
| THE VILLAGE OF HARLEM....NEW YORK !           |  4632   |        1         |       2       |    3     |  40.809   |    -73.942      |  150  |      3       |        0         |             |         0          |               1               |     365     |
| Cozy Entire Floor of Brownstone               |  4869   |        2         |       1       |    4     |  40.685   |    -73.960      |   89  |      1       |       270        | 2019-07-05  |       4.640        |               1               |     194     |
| Entire Apt: Spacious Studio/Loft by central park |  7192   |        2         |       2       |    5     |  40.799   |    -73.944      |   80  |      10      |        9         | 2018-11-19  |       0.100        |               1               |      0      |
| Large Cozy 1 BR Apartment In Midtown East     |  7322   |        2         |       2       |    6     |  40.748   |    -73.975      |  200  |      3       |        74        | 2019-06-22  |       0.590        |               1               |     129     |
| BlissArtsSpace!                               |  7356   |        1         |       1       |    7     |  40.687   |    -73.956      |   60  |      45      |        49        | 2017-10-05  |       0.400        |               1               |      0      |
| Large Furnished Room Near B'way               |  8967   |        1         |       2       |    8     |  40.765   |    -73.985      |   79  |      2       |       430        | 2019-06-24  |       3.470        |               1               |     220     |
| Cozy Clean Guest Room - Family Apt            |  7490   |        1         |       2       |    9     |  40.802   |    -73.967      |   79  |      2       |       118        | 2017-07-21  |       0.990        |               1               |      0      |
| Cute & Cozy Lower East Side 1 bdrm            |  7549   |        2         |       2       |    10    |  40.713   |    -73.990      |  150  |      1       |       160        | 2019-06-09  |       1.330        |               4               |     188     |
+-----------------------------------------------+---------+------------------+---------------+----------+-----------+-----------------+-------+--------------+------------------+-------------+--------------------+-------------------------------+-------------+
```

Figure 2 Sample Data Application Output

Design

Design Principles.

Following design principles considered when designing modules

Modularity
Modules should be independent from each other as much as possible

[4]High cohesion and low coupling

Single Responsibility

Reusability
Modules should be reusable and should be used with other data sets as well.

Modules

It was identified following modules will be implemented for analysing the data set.

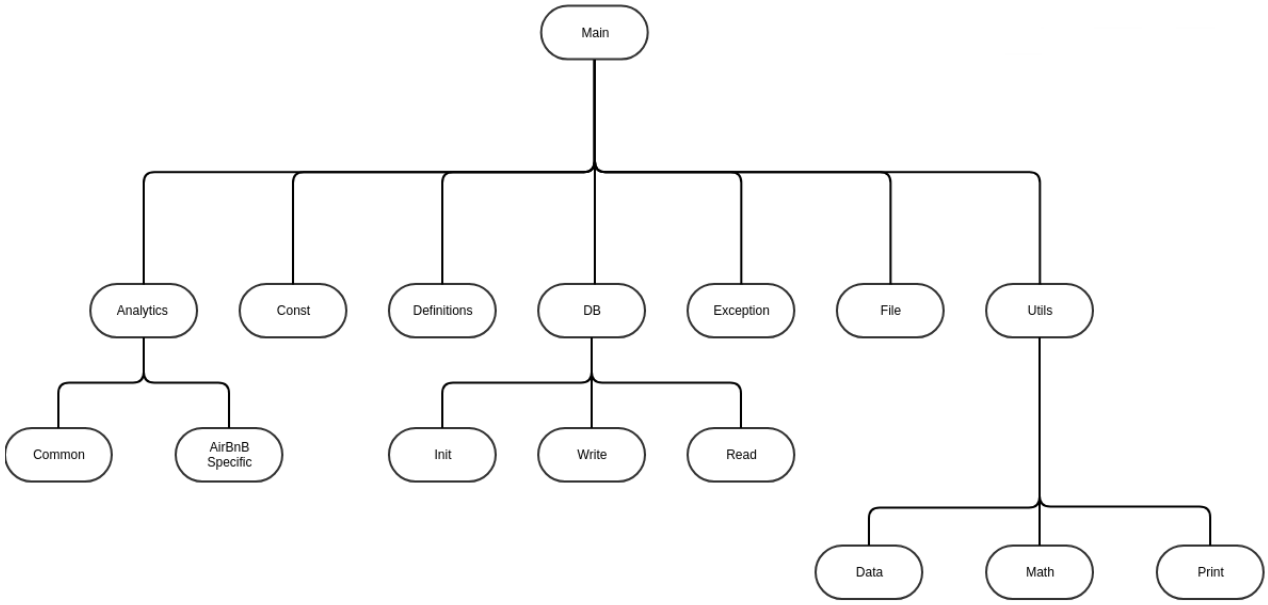


Figure 3 Module Diagram

[4] https://enterprisecraftsmanship.com/posts/cohesion-coupling-difference/

Main

Main module is the starting point of the program

Analytics

Responsibility of this module is all graphs related activities. It consists 2 sub modules for Common and AirBnB specific graphs.

Constant

Shared module which can be used with other modules to specify constants.

Definition

All the application specific definitions / configurations would be defined here

DB

DB module is responsible for all db related operation. This module has 3 sub modules responsible for db Init operations such as creating tables, indexes, Write for all db insert operations. And Read for db sql selects.

Exception.

All application specific exception will be defined in this module.

File

This module will be implemented for csv file read and filtering invalid records.

Utils.

A reusable utility module consists of 3 sub modules Data, Math, and Print

Data

Reusable module can be used with any data set to filter, manipulate, or transform data.

Math

Math module is a reusable module which can be used to calculate main statistical elements such as min, median, mode, standard deviation etc.

Print

Print module is responsible for printing output standard output (console) in user friendly manner.

Database Table Diagram

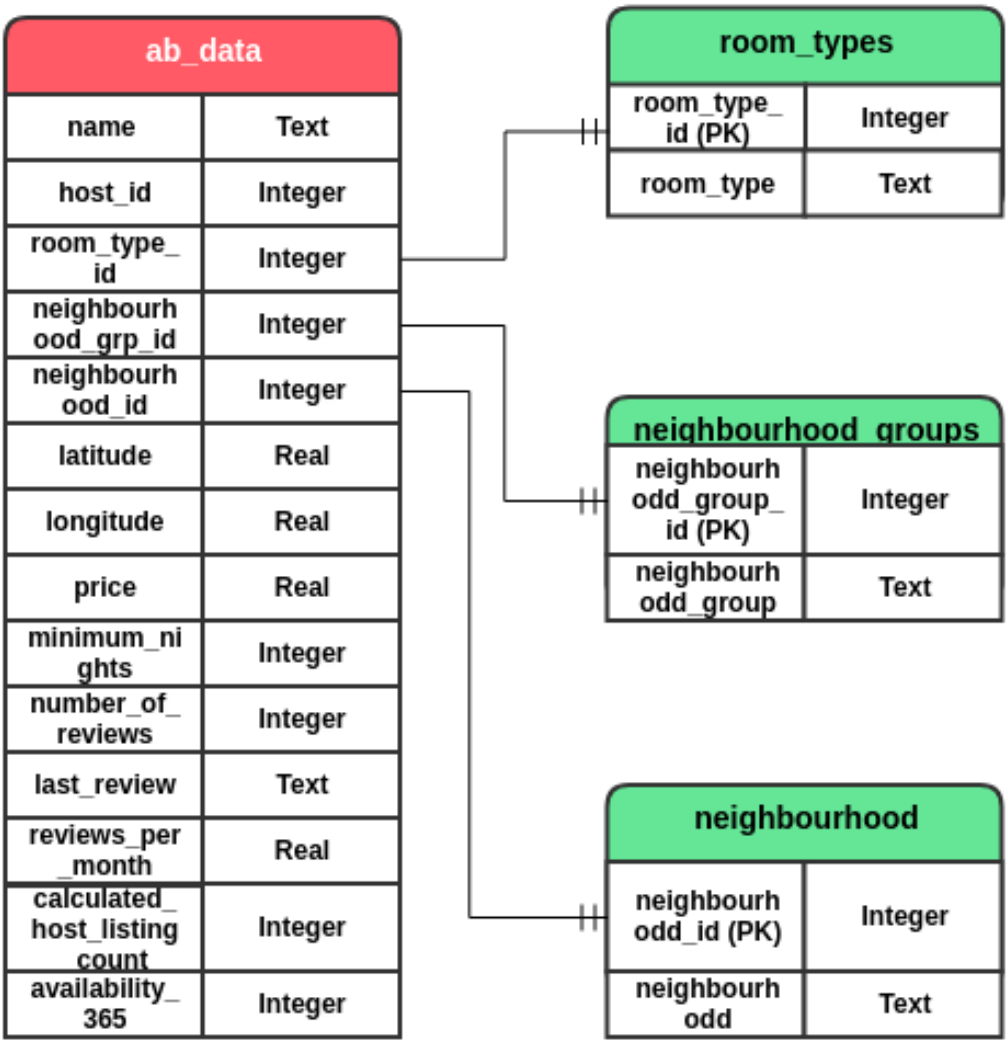


Figure 4 Database Table Diagram

Data repetition was identified with Room Type, Neighbourhood Group and Neighbourhood fields. So, it was decided to normalise these data into separate tables.

Sequence Diagrams

Init

Main

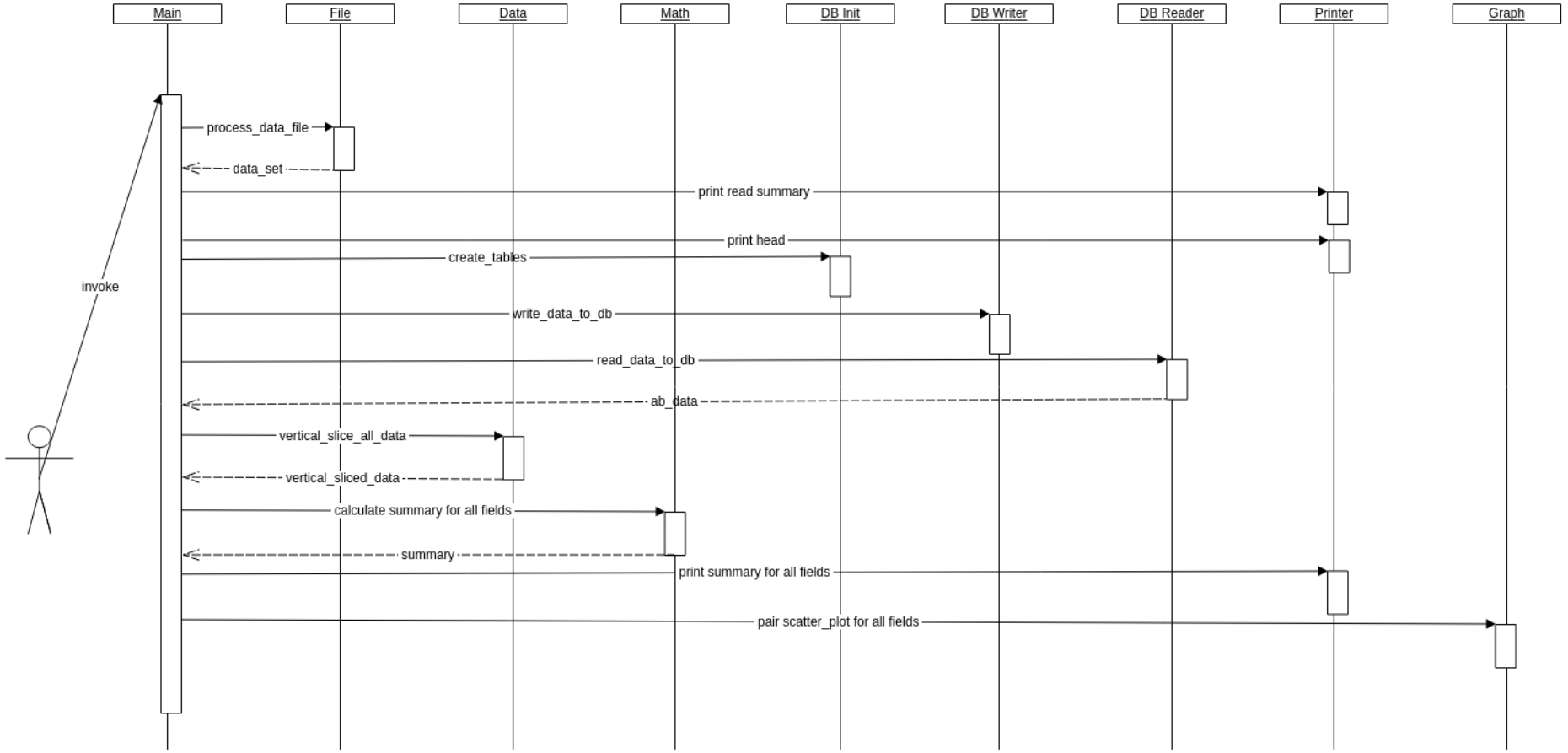


Figure 5 Sequence Diagram File Read to Pair Scatter Plot

Comparing Price

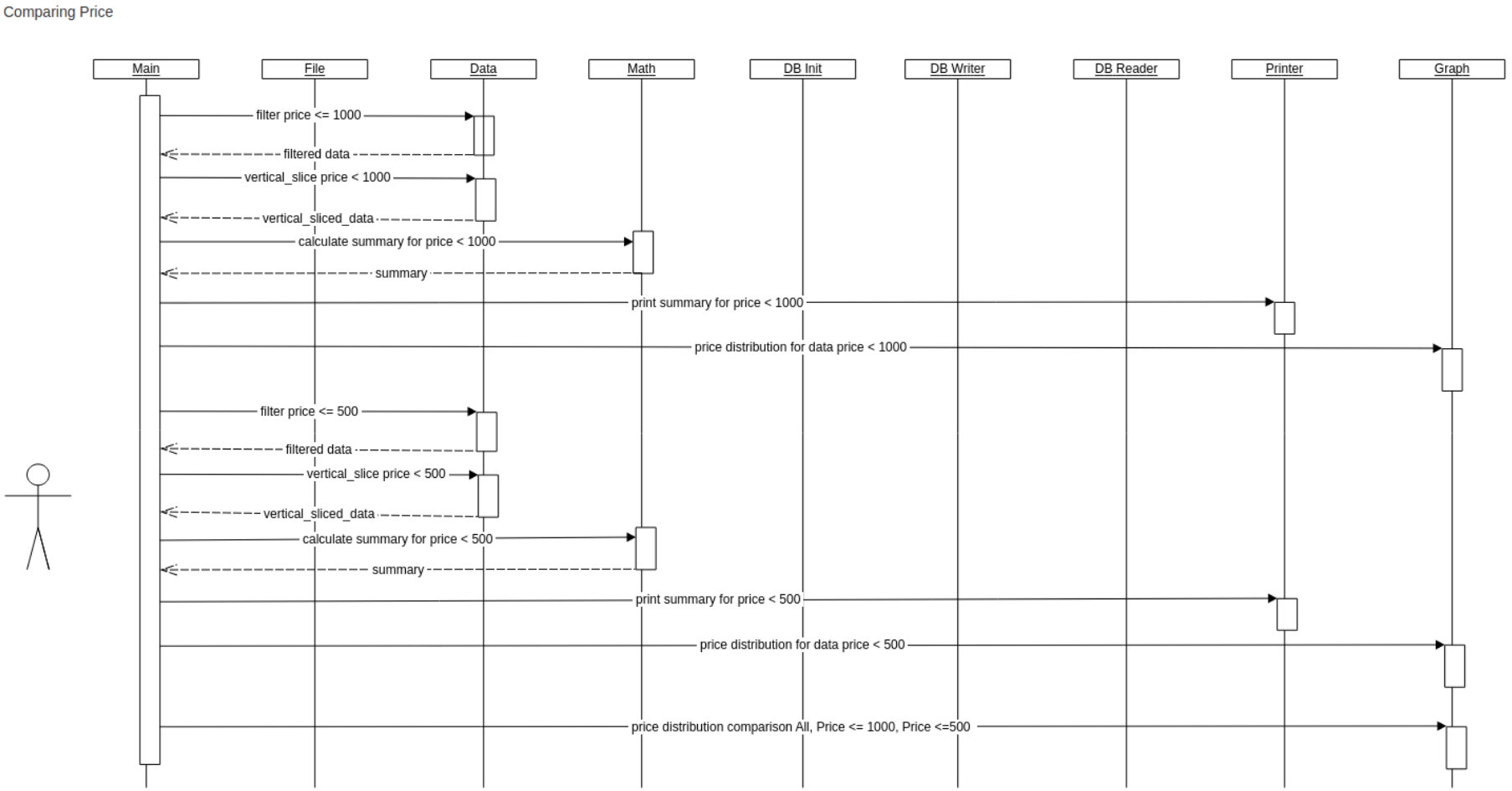


Figure 6 Sequence Diagram Selecting Price Range 2

Analysing data Price < 500

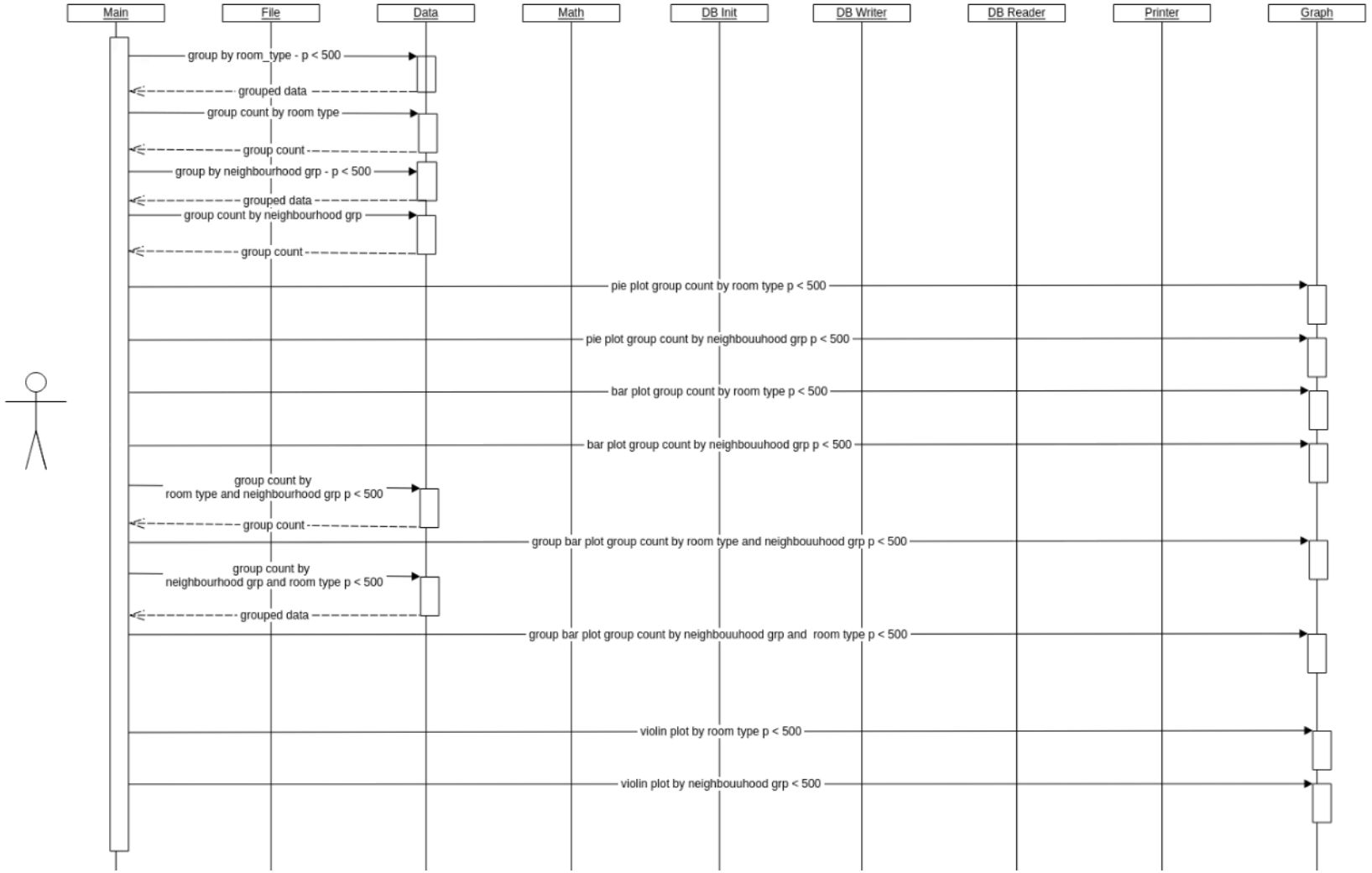


Figure 7 Sequence Diagram Analysing data Price > 500

Analysing data Price > 1000

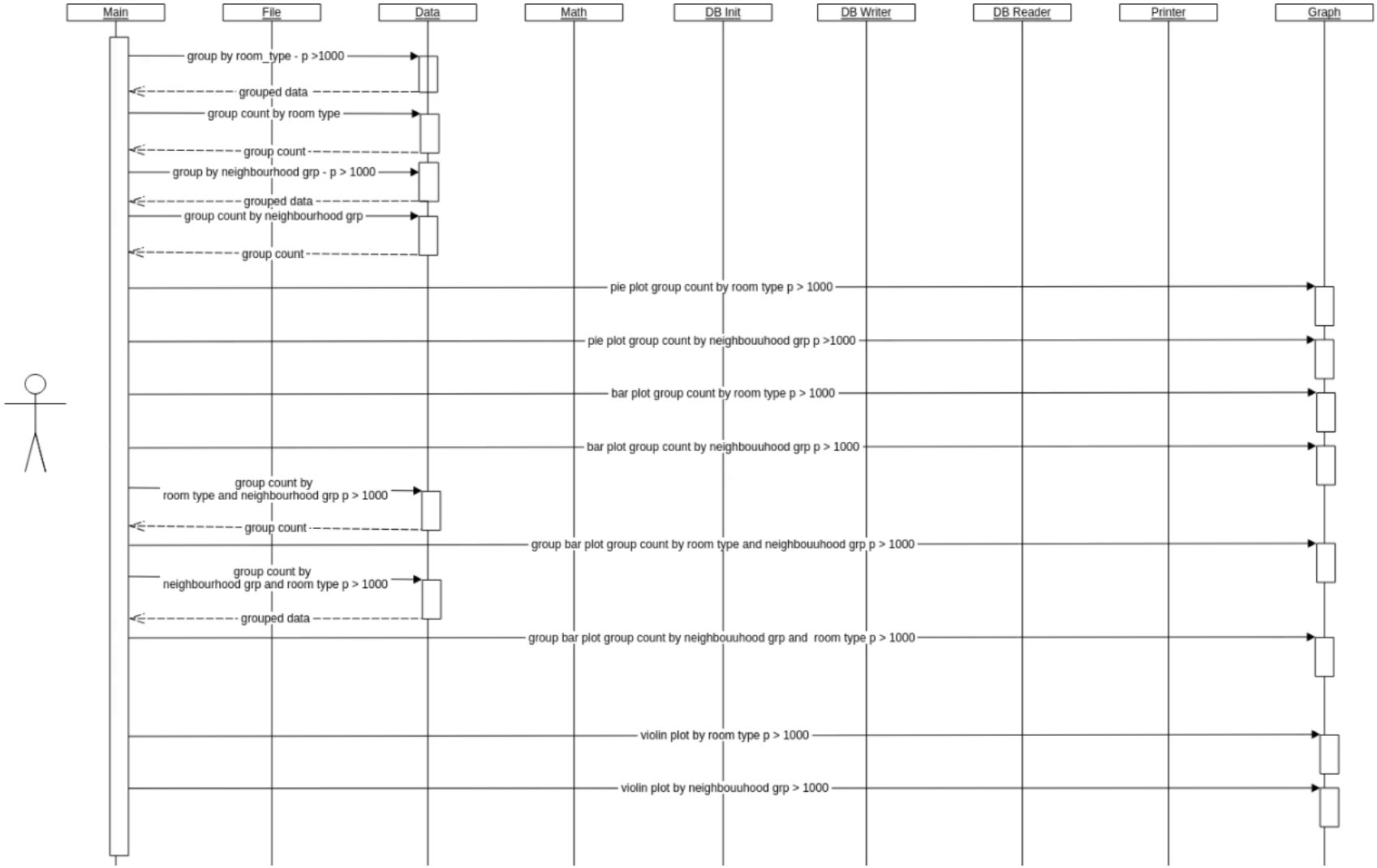


Figure 8 Sequence Diagram Analysing data Price > 1000

Analysing data Price > 5000

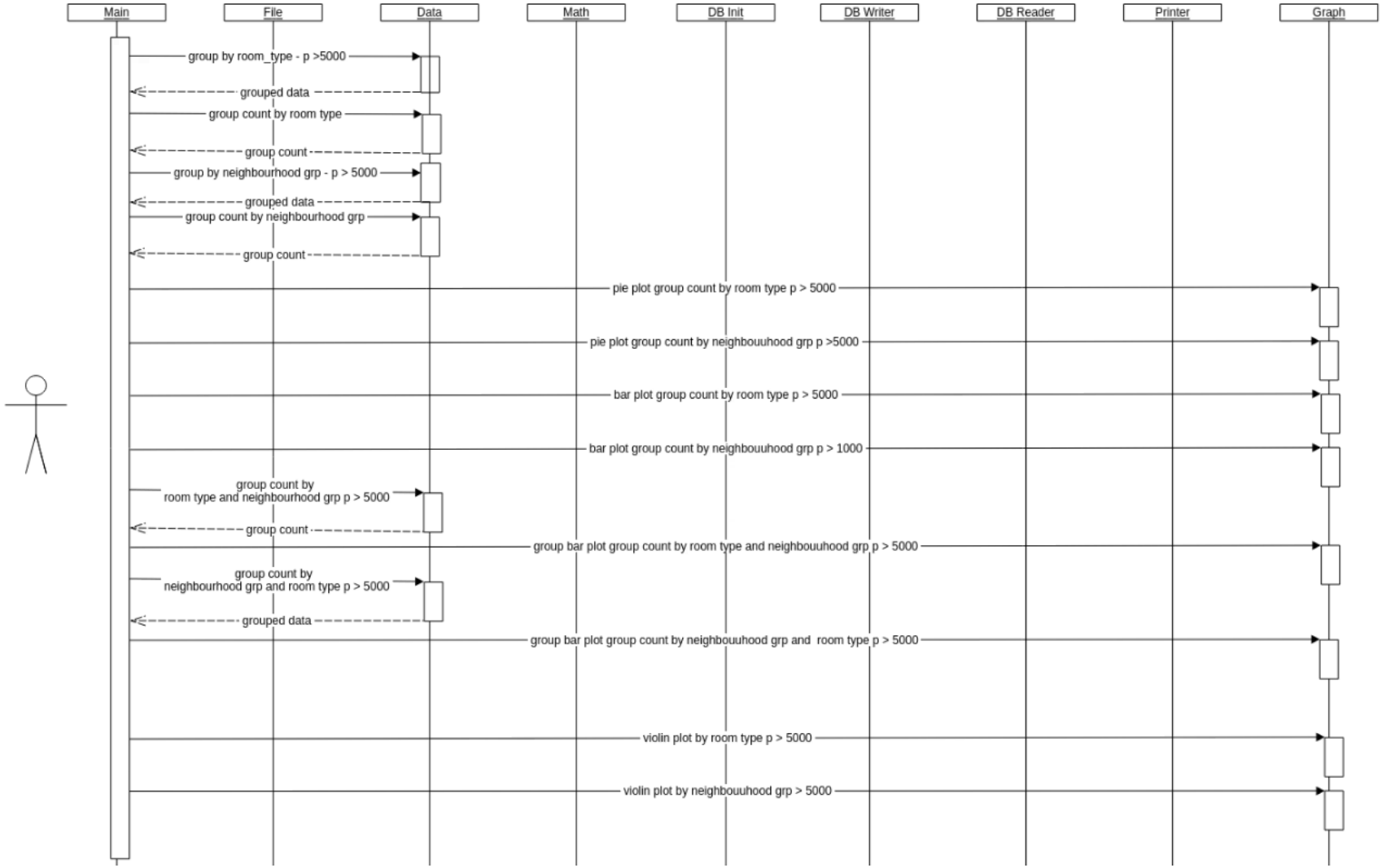


Figure 9 Sequence Diagram Analysing data Price > 5000

Availability = 0, Availability > 0

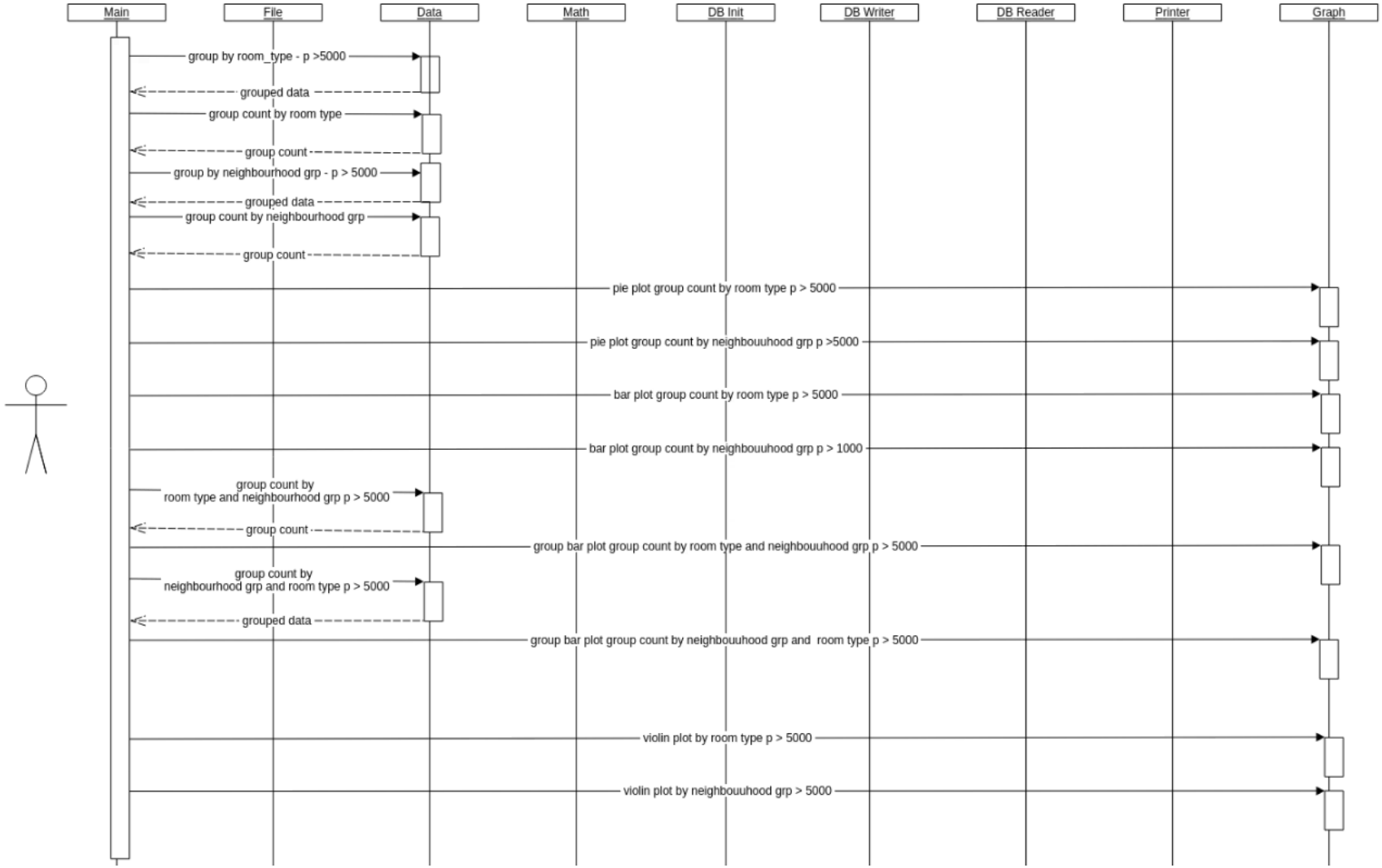


Figure 10 Sequence Diagram Availability = 0, Availability >

Testing

Unittest has been used for testing main modules. 100%-line coverage has been recorded.

Figure 11 Unit Test Coverage

To run all the unit test from the root directory of the project run following command

```
python -m unittest
```

To verify the calculations are correct, numpy, panda, stat modules has been used only in unit testing verification

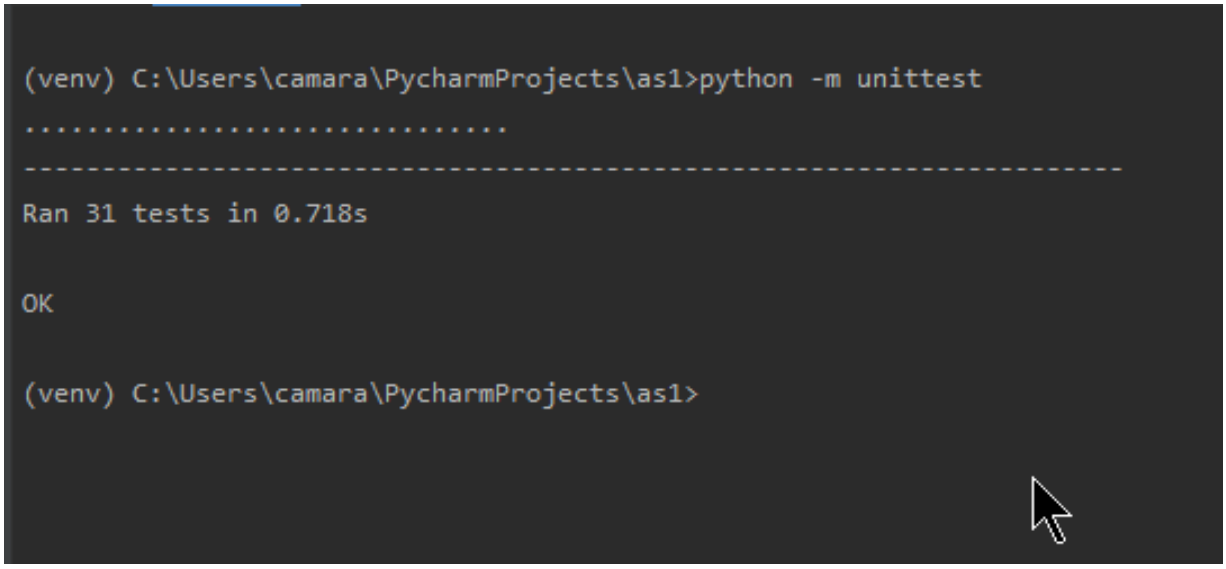


Figure 12 Running all unittests

Analysis and Visualisation

1. File Reading summary

   After initial file reading program generate following report to illustrate Total Valid Rows, Skipped Rows and Total number of rows. This execution is similar to pandas describe method

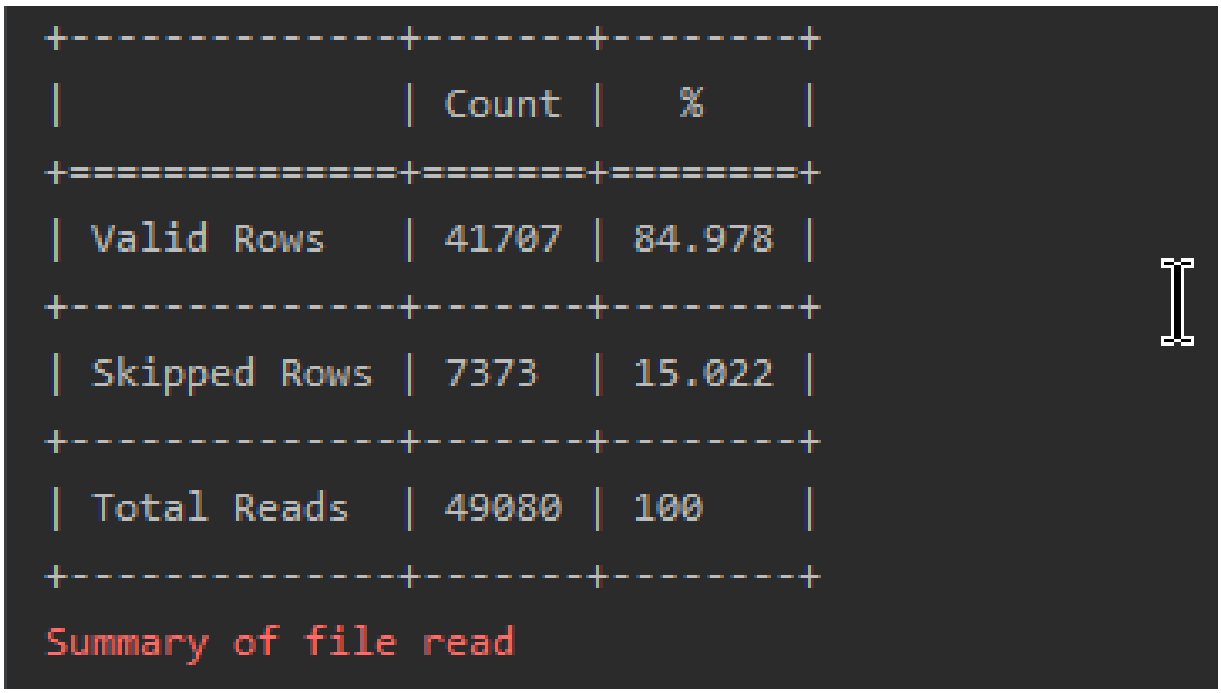


Figure 13 Summary of file read

2. Data Head

   Display first 10 rows of valid data. This functionality is similar to pandas head

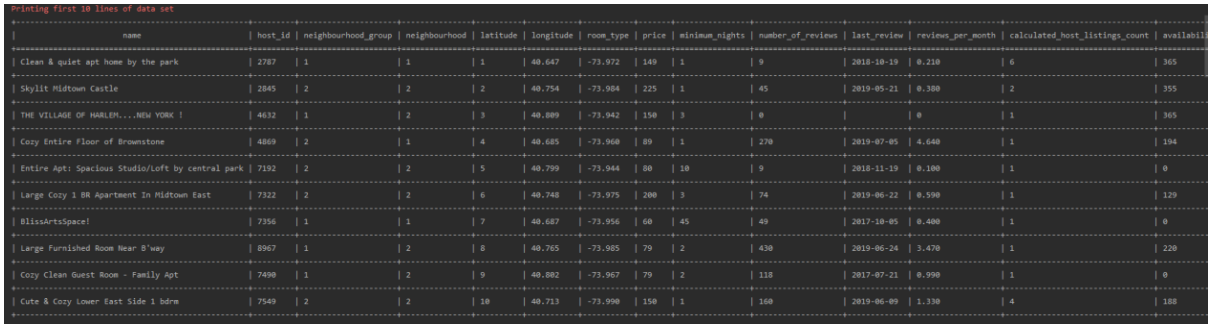


Figure 14 Data Head

3. Data Summary

   Print count, min, mean, mode, median, sd, max all numerical fields.

Field	count	min	mean	mode	median	sd	max
latitude	41707	40.500	40.729	[(40.71813, 17)]	40.723	0.054	40.913
longitude	41707	-74.244	-73.953	[(-73.95427, 15), (-73.95405, 15), (-73.95136, 15)]	-73.956	0.046	-73.713
price	41707	0	153.790	[(100.0, 1771)]	109	243.437	10000
minimum_nights	41707	1	6.957	[(1, 10987)]	2	20.920	1250
number_of_reviews	41707	0	23.185	[(0, 8678)]	5	44.647	629
reviews_per_month	41707	0	1.090	[(0, 8678)]	0.370	1.602	58.500
calculated_host_listings_count	41707	1	6.954	[(1, 27459)]	1	32.559	327
availability_365	41707	0	113.778	[(0, 14972)]	46	132.186	365

Figure 15 Data Summary

4. Pair Scatter Plots

Pair scatter plots are nice feature of seaborn plots to quickly visualize the correlation of a variable against all other variables. But seaborn only support numpys dataframe. Introduced an adapter to generate pair scatter plots from generic list of array.

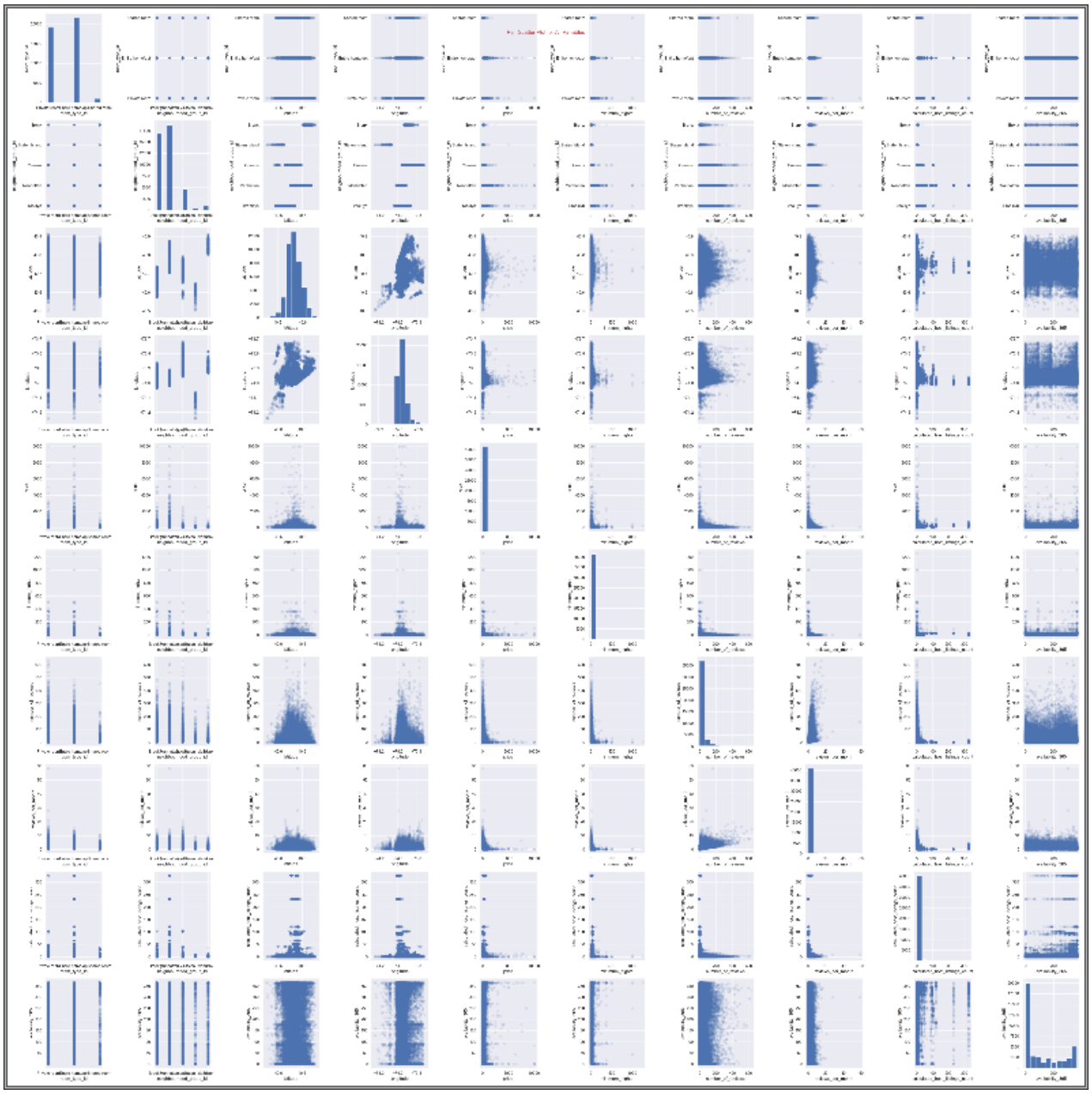


Figure 16 Pair Scatter Plots

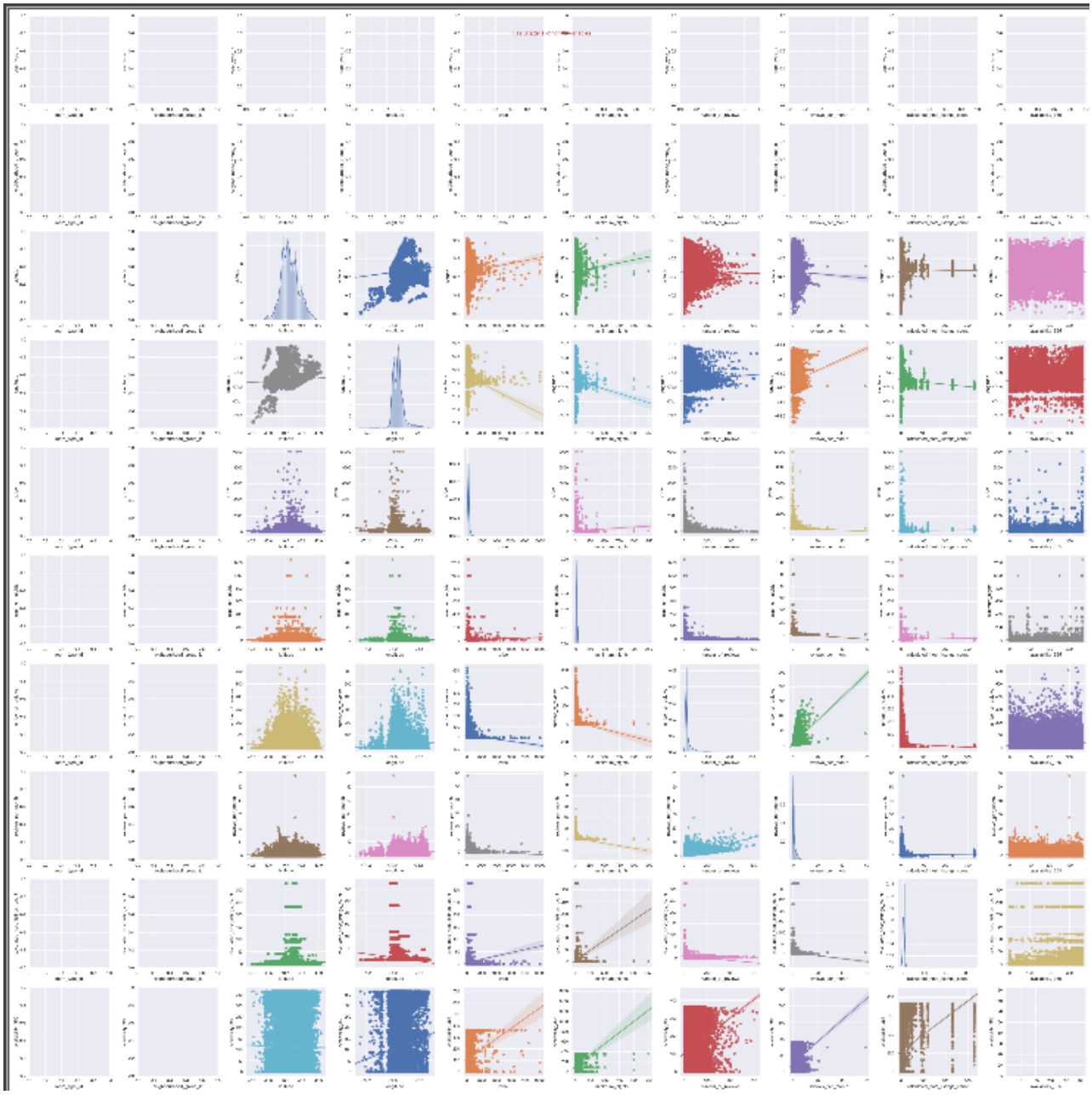


Figure 17 Pair Scatter Plots with Correlations

5. It was decided to focus on price data. Print price summary

Field	count	min	mean	mode	median	sd	max
price	41707	0	153.790	[(100.0, 1771)]	109	243.437	10000

Figure 18 Price Summary

6. Price Distribution graph for all data

Figure 19 Price Distribution - All Data

The graph says the price distribution is between 0 and $1000. Extreme Price listings need to be Filtered.

7. Consider Price less than or equal 1000. Price Summary

Field	count	min	mean	mode	median	sd	max
price	41501	0	142.083	[(100.0, 1771)]	106	118.224	1000

Figure 20 Price Summary. Price <=1000

Standard deviation has been reduced to 118.224

8. Price Distribution for Price <= 1000

Figure 21 Price Distribution. Price <= 1000

9. Considering listings where price <= 500

Field	count	min	mean	mode	median	sd	max
price	40789	0	131.906	[(100.0, 1771)]	103	88.388	500

Figure 22 Price Summary. Price <=500

10. Price Distribution for Price <= 500

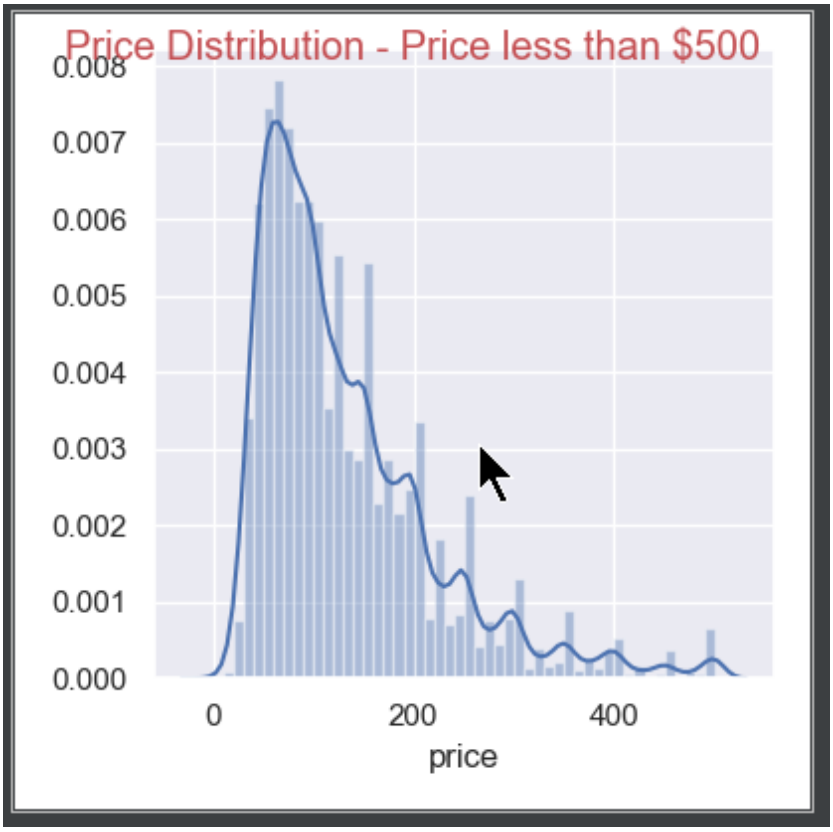


Figure 23 Price Distribution. Price <= 500

11. Price Distribution comparison for All Data, and where Price <= $1000 and Price <= $500

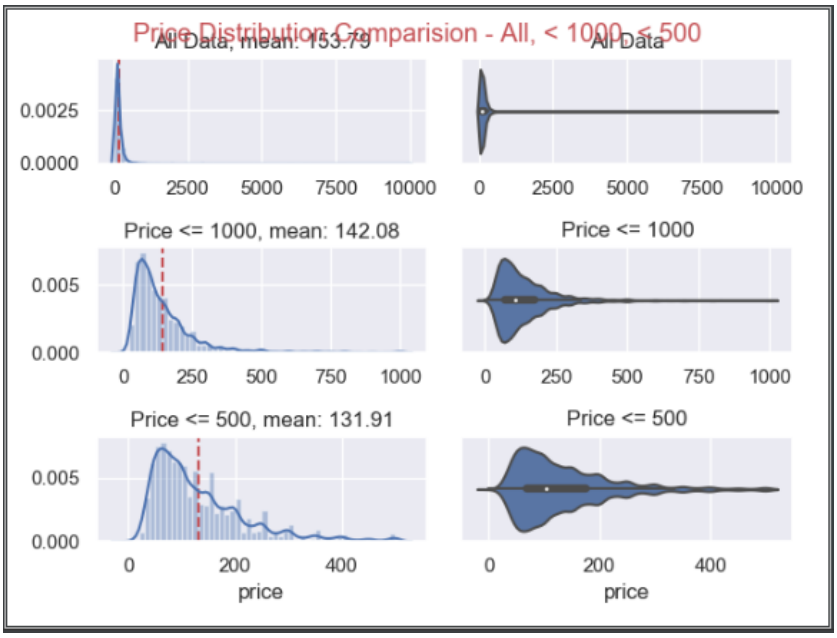


Figure 24 Price Distribution comparison

Considering Price Distribution comparison graph it was decided to analysis data separately where Listing price is 0 to $500 and above $500

12. Room listing by Room Type (Price <= $500)

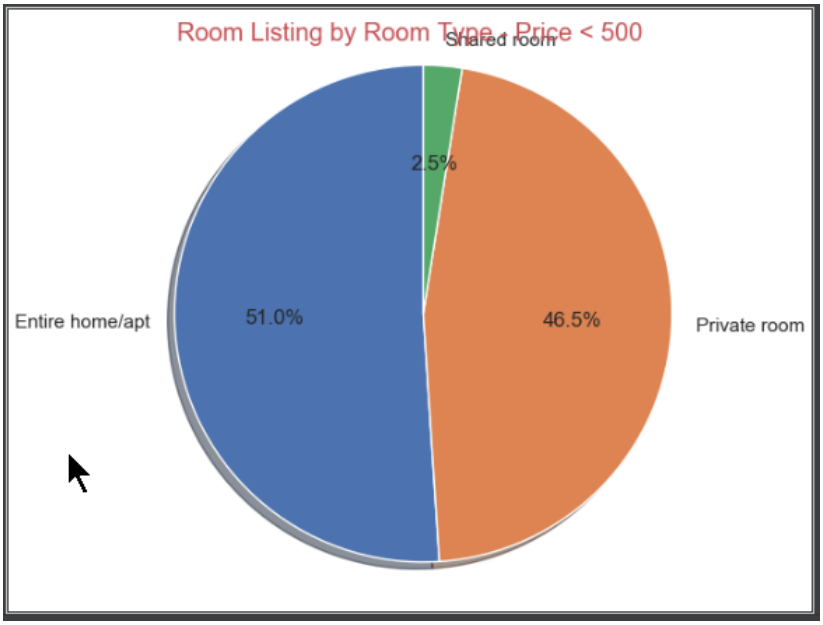


Figure 25 Room Listing by Room Type, Pie Chart (Price < 500)

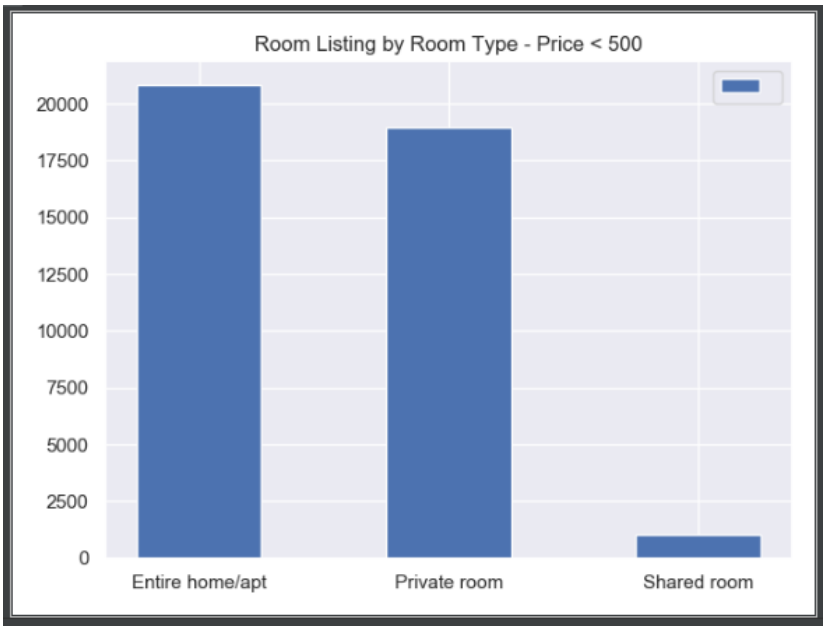


Figure 26 Room Listing by Room Type, Count Bar Chart (Price < 500)

13. Room listing by Neighbourhood Group (Price <= $500)

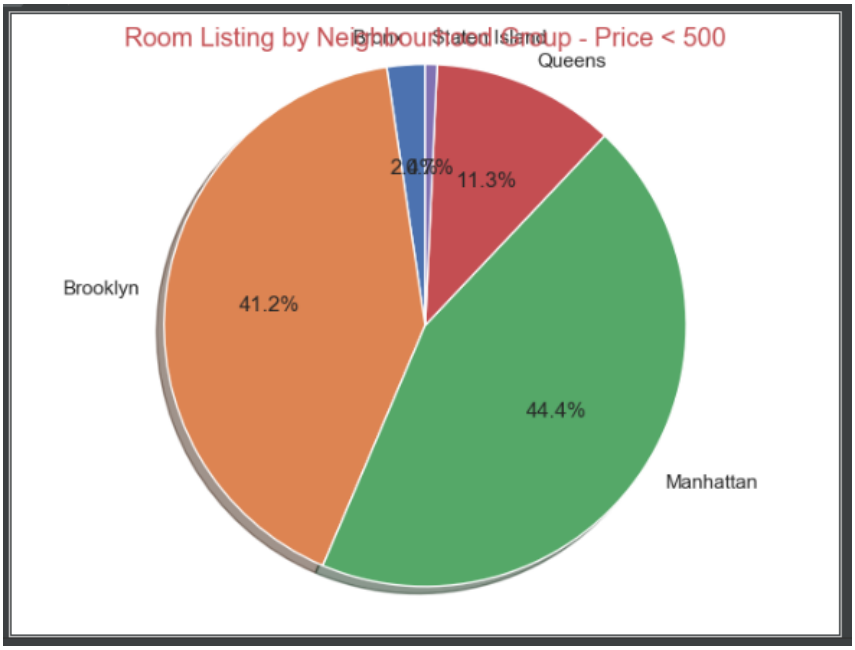


Figure 27 Room Listing by Neighbourhood Group, Pie Chart (Price < 500)

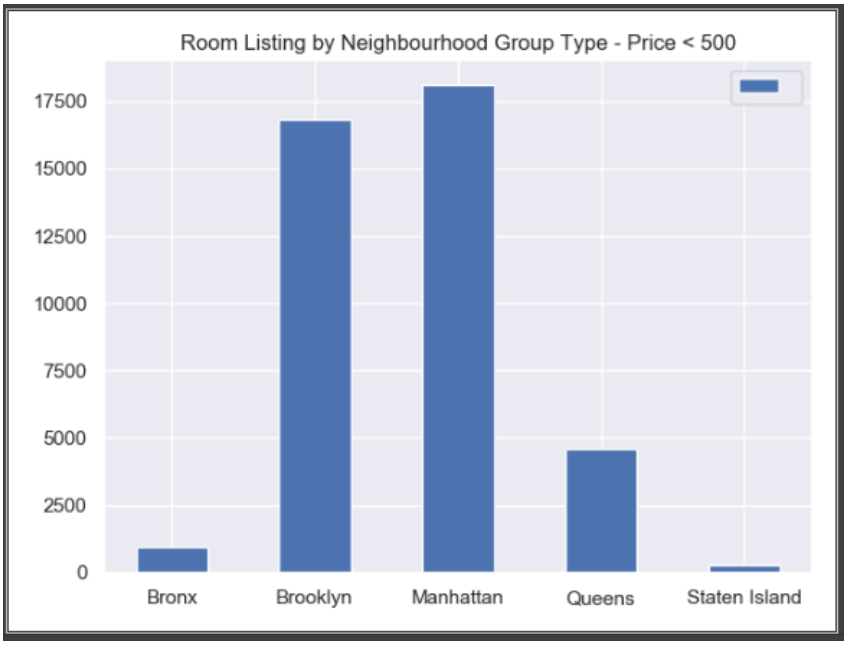


Figure 28 Room Listing by Neighbourhood Group, Count Bar Chart (Price < 500)

14. Room listing by Room Type and Neighbourhood Group (Price <=$ 500)

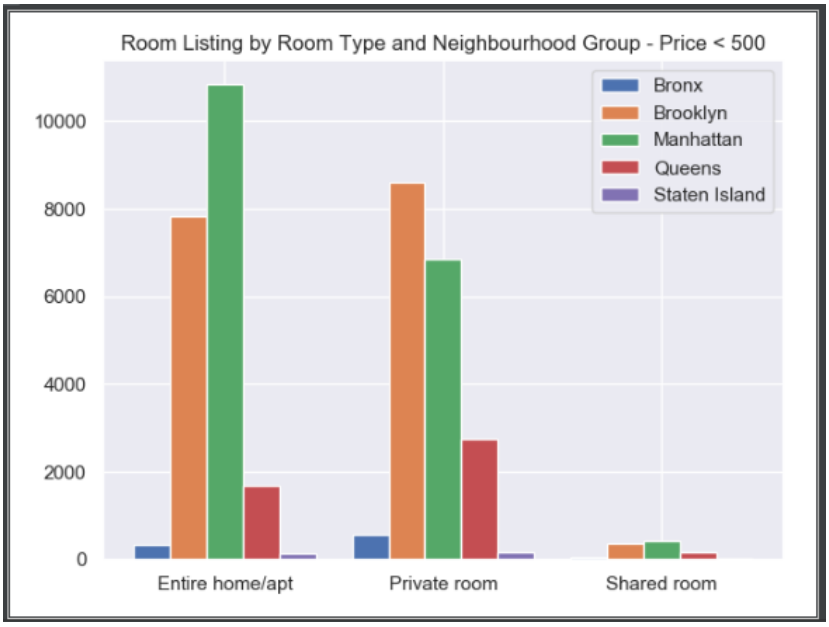


Figure 29 Room listing by Room Type and Neighbourhood Group (Price <=$ 500)

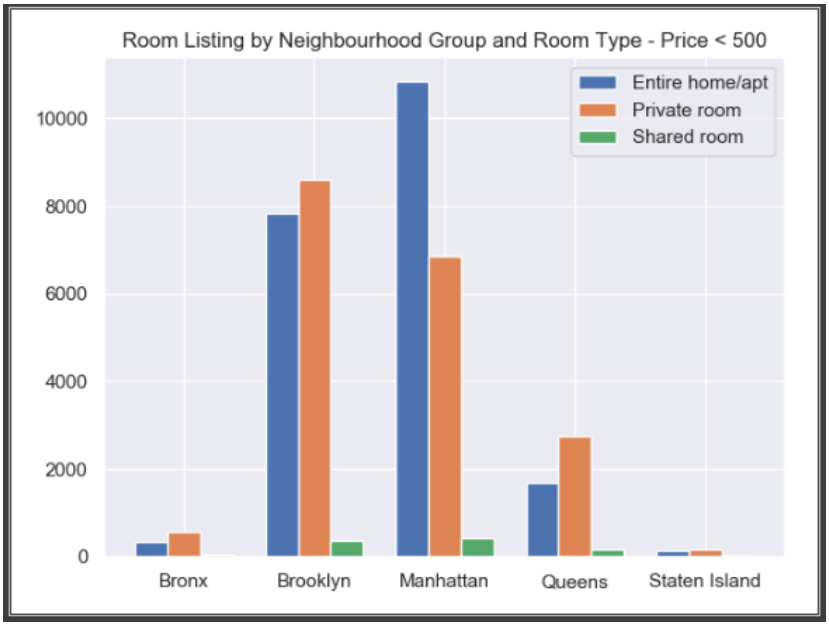


Figure 30 Room listing by Neighbourhood Group and Room Type (Price <=$ 500)

15. Price KDE by Room Type (Price <= $500)

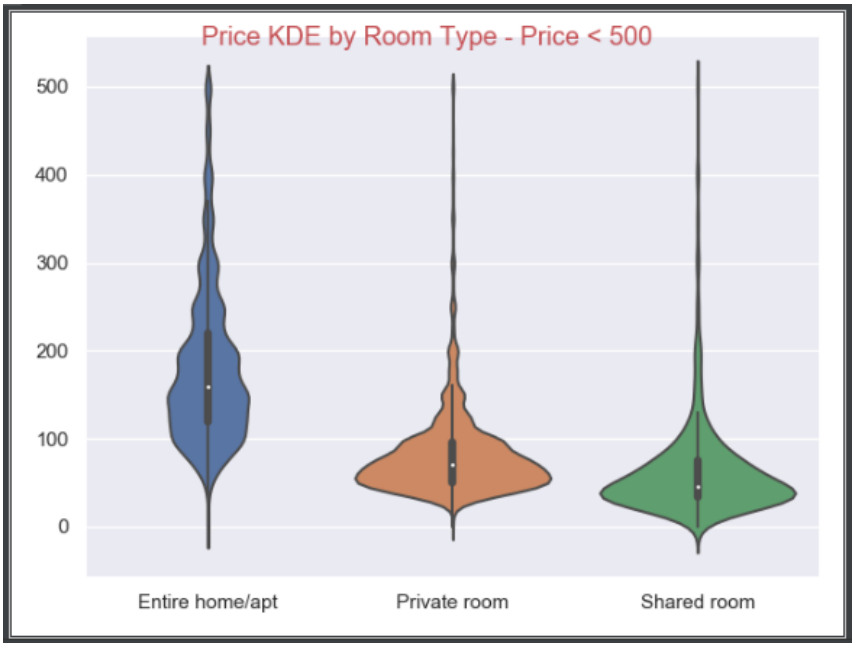


Figure 31 Price KDE by Room Type (Price <= $500)

16. Price KDE by Neighbourhood Group (Price <= $500)

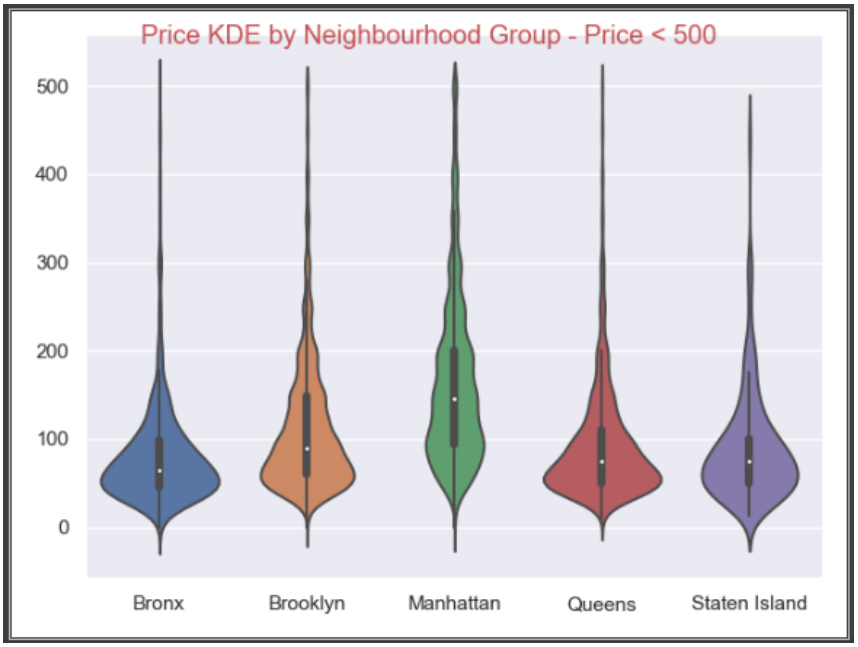


Figure 32 Price KDE by Neighbourhood Group (Price <= $500)

17. Price Analysis. Price > 1000 and Price > 5000

Field	count	min	mean	mode	median	sd	max
price	206	1002	2512.204	[(2000.0, 23)]	1939	1895.429	10000

Field	count	min	mean	mode	median	sd	max
price	23	5000	7066.087	[(5000.0, 6)]	6500	1860.567	10000

Figure 33 Price Summary. Price > 1000, Price > 5000

Figure 34. Price Distribution. Price > $1000, Price > $5000

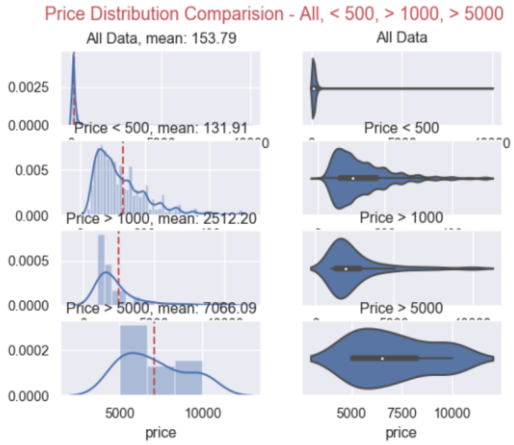


Figure 35 Price Comparison. All Price, Price <$500, Price > $1000, Price > $5000

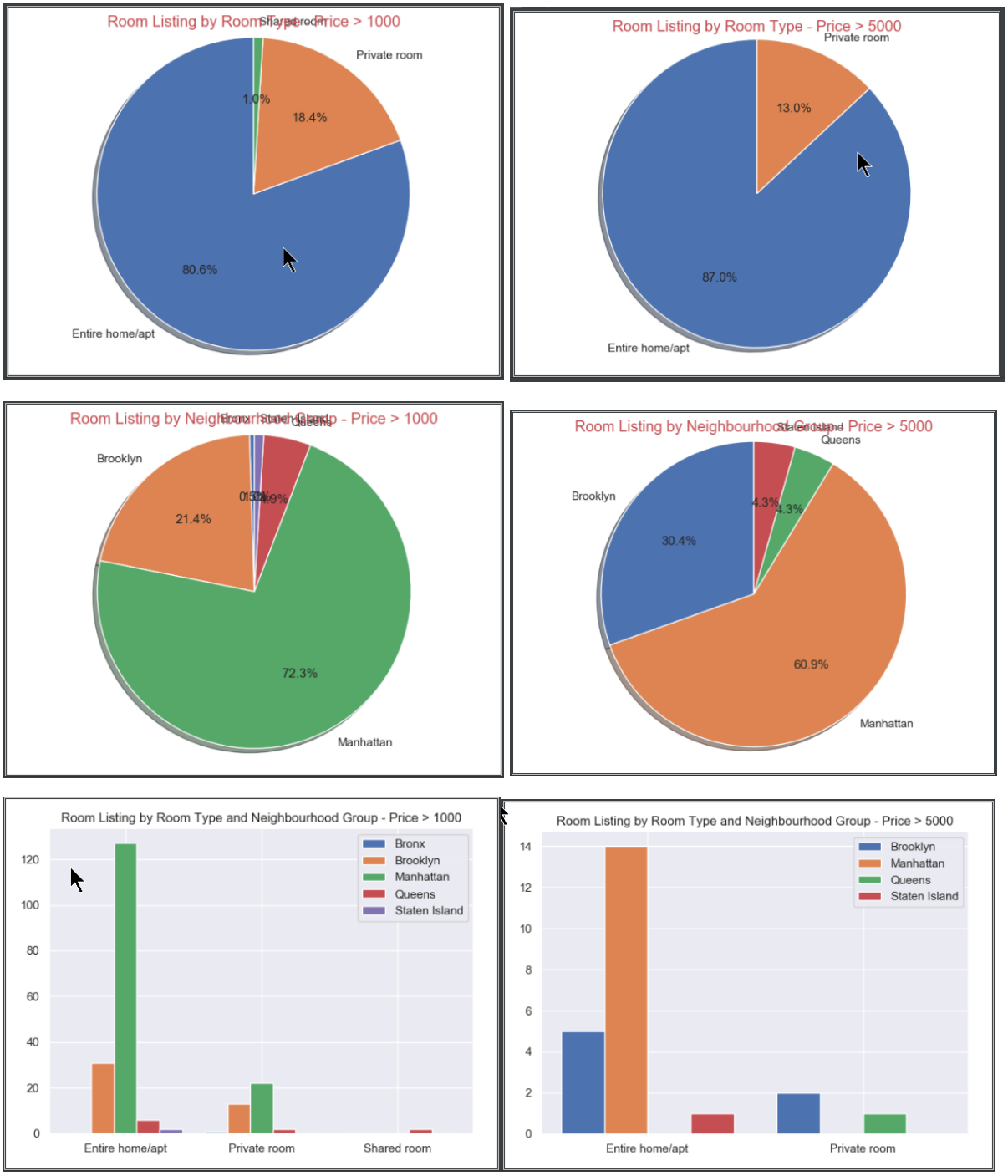


Figure 36 Price Grouping

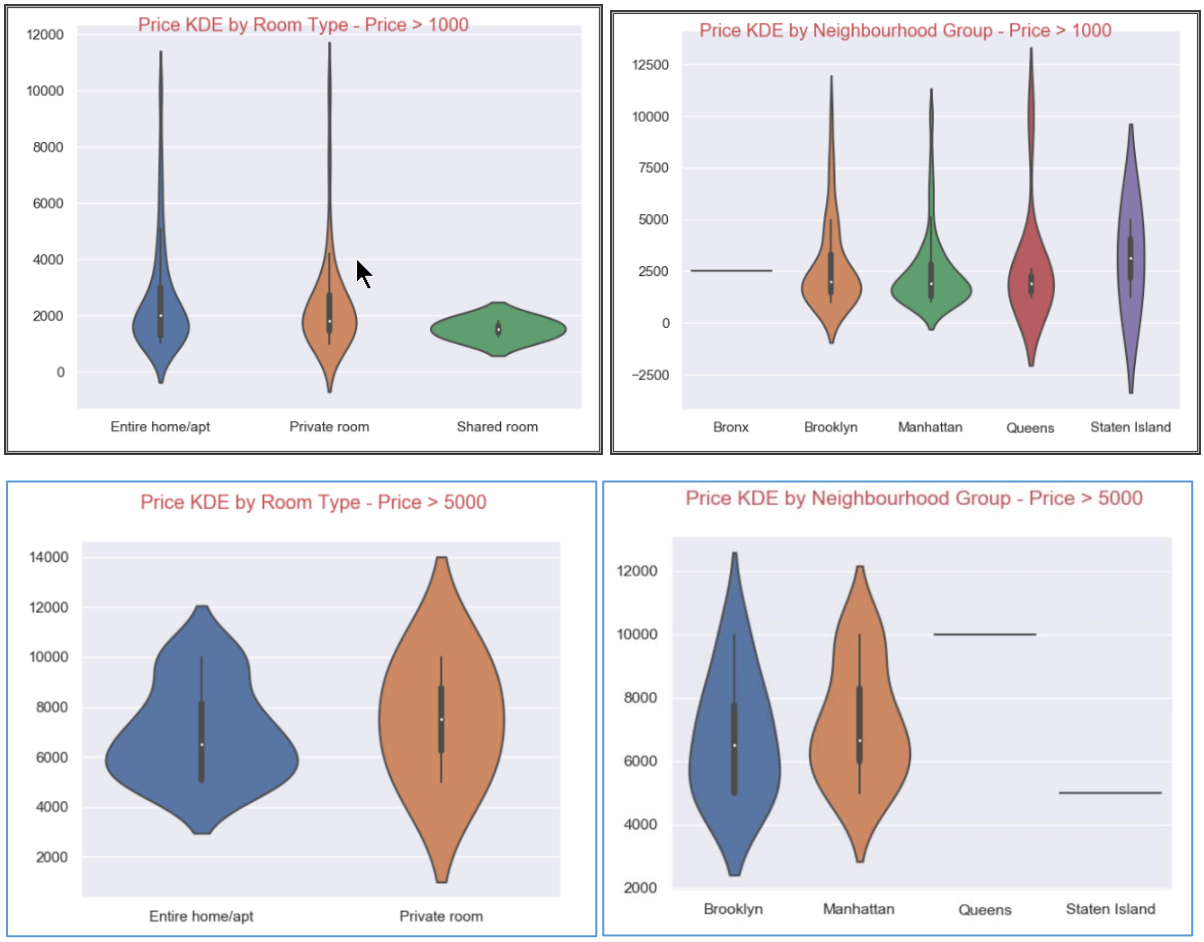


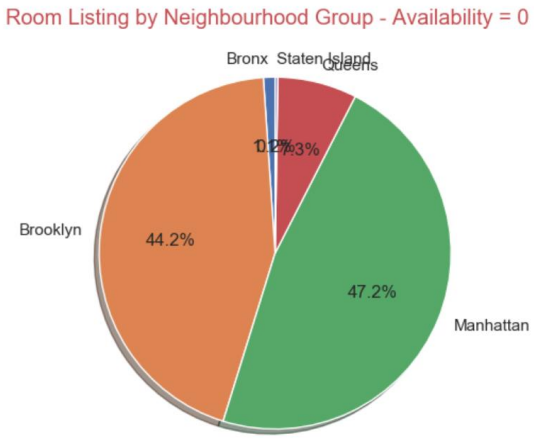
Figure 37 Price Grouping

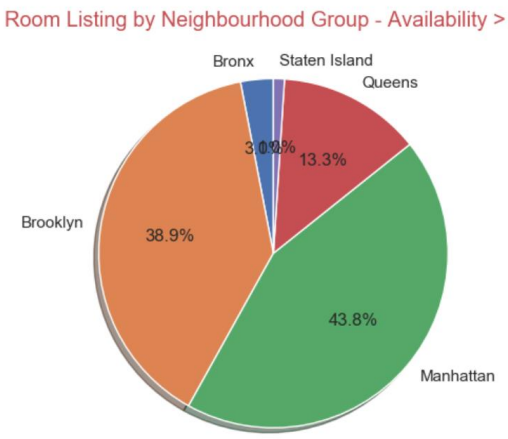
18. Price Comparison Availability 0 vs Availability > 0

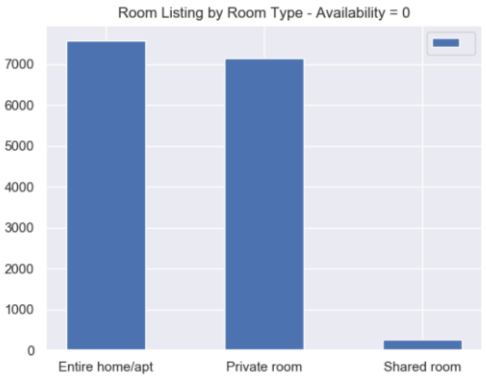
Field	count	min	mean	mode	median	sd	max
price	14972	0	137.848	[(100.0, 748)]	100	218.864	10000

Field	count	min	mean	mode	median	sd	max
price	26735	0	162.717	[(100.0, 1023)]	113	255.738	10000

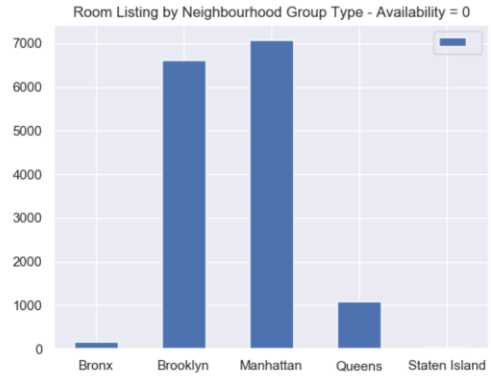
Figure 38 Price Comparison for Availability

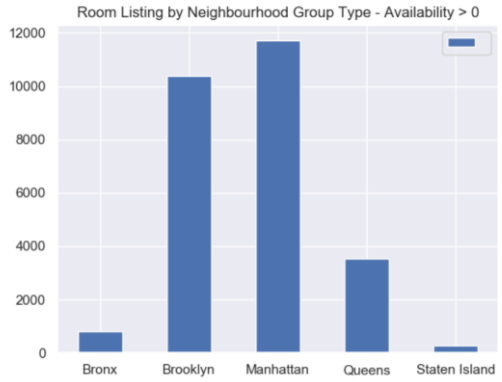

Room Listing by Neighbourhood Group - Availability = 0
Bronx
Staten Island
Queens
Brooklyn
44.2%
47.2%
Manhattan


Room Listing by Neighbourhood Group - Availability > 0
Bronx
Staten Island
Queens
13.3%
Brooklyn
38.9%
43.8%
Manhattan


Room Listing by Room Type - Availability = 0
7000
6000
5000
4000
3000
2000
1000
0
Entire home/apt
Private room
Shared room

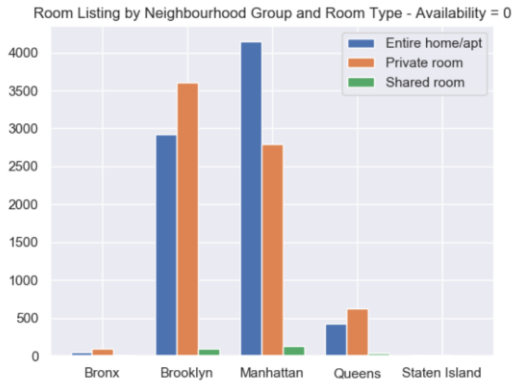

Room Listing by Room Type - Availability > 0
14000
12000
10000
8000
6000
4000
2000
0
Entire home/apt
Private room
Shared room


Room Listing by Neighbourhood Group Type - Availability = 0
7000
6000
5000
4000
3000
2000
1000
0
Bronx
Brooklyn
Manhattan
Queens
Staten Island

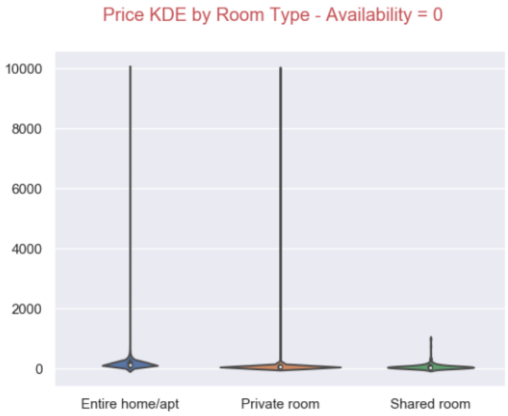

Room Listing by Neighbourhood Group Type - Availability > 0
12000
10000
8000
6000
4000
2000
0
Bronx
Brooklyn
Manhattan
Queens
Staten Island


Room Listing by Room Type and Neighbourhood Group - Availability = 0
4000
3500
3000
2500
2000
1500
1000
500
0
Bronx
Brooklyn
Manhattan
Queens
Staten Island
Entire home/apt
Private room
Shared room


Room Listing by Room Type and Neighbourhood Group - Availability > 0
7000
6000
5000
4000
3000
2000
1000
0
Bronx
Brooklyn
Manhattan
Queens
Staten Island
Entire home/apt
Private room
Shared room


Room Listing by Neighbourhood Group and Room Type - Availability = 0
Entire home/apt
Private room
Shared room
4000
3500
3000
2500
2000
1500
1000
500
0
Bronx
Brooklyn
Manhattan
Queens
Staten Island


Room Listing by Neighbourhood Group and Room Type - Availability > 0
Entire home/apt
Private room
Shared room
7000
6000
5000
4000
3000
2000
1000
0
Bronx
Brooklyn
Manhattan
Queens
Staten Island


Price KDE by Room Type - Availability = 0
10000
8000
6000
4000
2000
0
Entire home/apt
Private room
Shared room

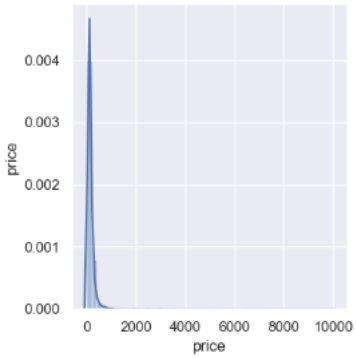

Price KDE by Room Type - Availability > 0
10000
8000
6000
4000
2000
0
Entire home/apt
Private room
Shared room


Price KDE by Neighbourhood Group - Availability = 0
10000
8000
6000
4000
2000
0
Bronx
Brooklyn
Manhattan
Queens
Staten Island

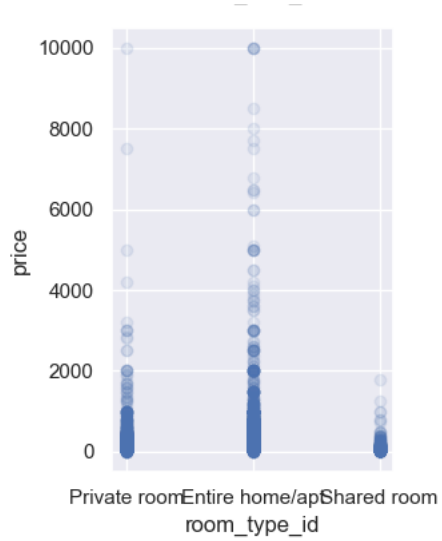

Price KDE by Neighbourhood Group - Availability > 0
10000
8000
6000
4000
2000
0
Bronx
Brooklyn
Manhattan
Queens
Staten Island

Conclusion

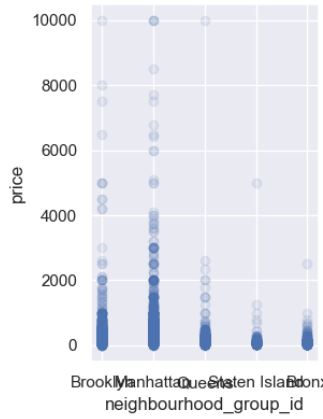
1. Most of the price listing are lower than $2000

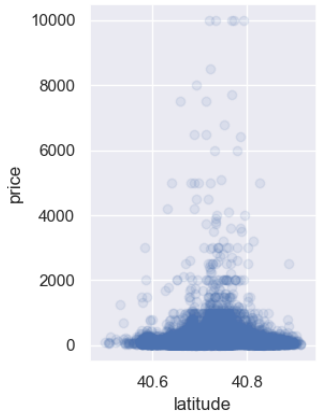


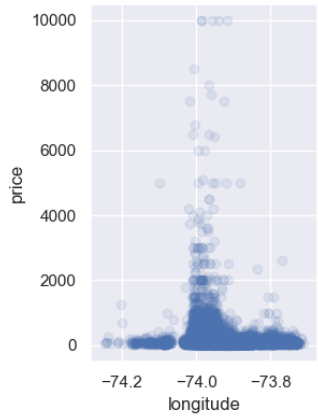
2. Room type has a direct relationship to the price. Entire Home / apt has highest price and always Shared rooms price is lower than $2000.



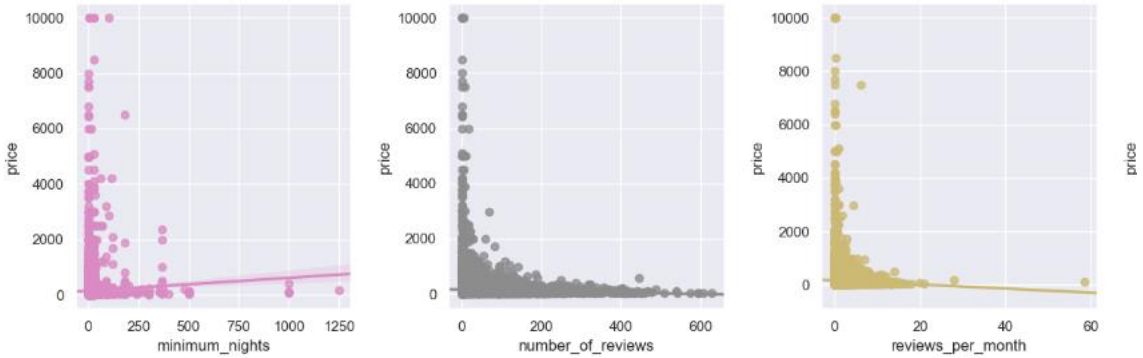
3. The location of the listing has a strong relation to the price. Manhattan and Brooklyn properties have been listed with a higher price.



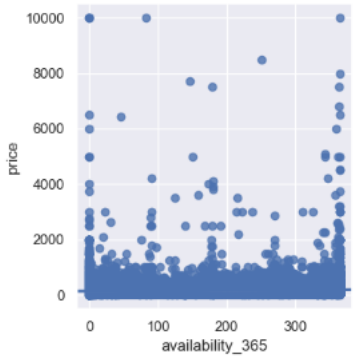




4. Price has a positive correlation with minimum number of nights, while it has a negative correlation with Number of reviews and Number of reviews per month. The reason could be low price attracts higher number of tenants.



5. Below graph tries to find the relationship between Price and Availability. There is no clear relationship between Price and Availability.



6. More than 97% listing's price is lower than $500

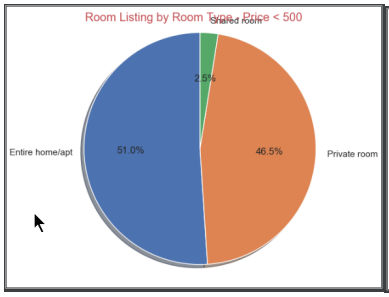
Field	count	min	mean	mode	median	sd	max
price	41707	0	153.790	[(100.0, 1771)]	109	243.437	10000

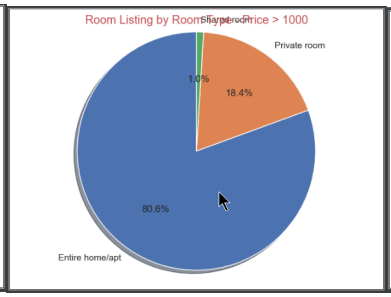
Field	count	min	mean	mode	median	sd	max
price	40789	0	131.906	[(100.0, 1771)]	103	88.388	500

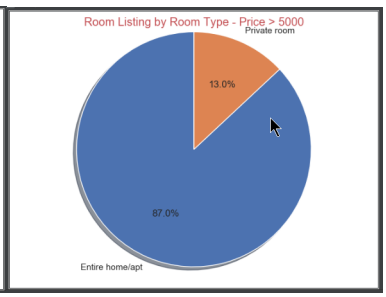
7. Only 0.00006% (23 out of 41707) listings are priced over $5000

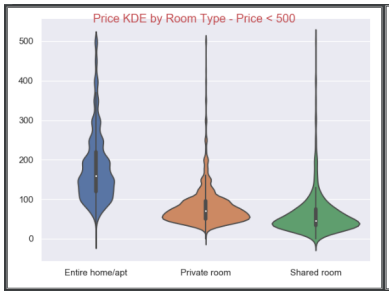
Field	count	min	mean	mode	median	sd	max
price	23	5000	7066.087	[(5000.0, 6)]	6500	1860.567	10000

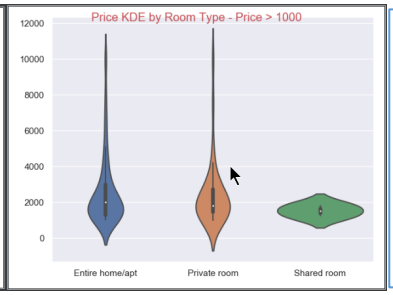
8. Entire Home / apt are popular in all price categories.
9. But when price goes up Entire Home / apt are more popular.
10. No any Shared rooms which are priced over $5000

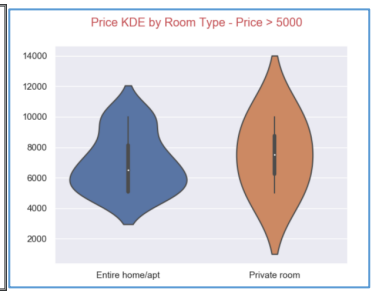




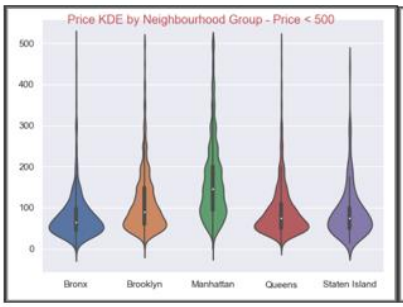


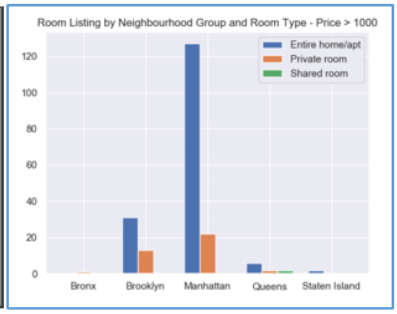


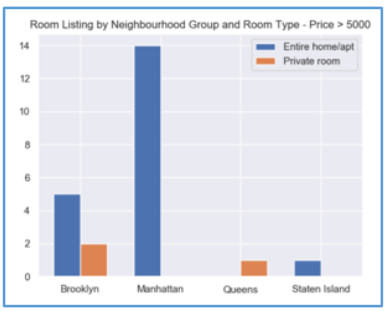




11. Manhattan has the highest median for Price in all 3 price categories, followed by Brooklyn
12. Manhattan has the highest listing count followed by Brooklyn
13. In $500 price category, Private rooms are much popular in all neighbourhood groups, Except Manhattan where Entire Home / Apt has highest listing.
14. For over $5000 category, Private Room nor Shared Room available in Manhattan.







15. Nearly 40% of Total listing are not available.

Field	count	min	mean	mode	median	sd	max
price	14972	0	137.848	[(100.0, 748)]	100	218.864	10000

Appendix 1: Reflective Learning Log

Date: 25/10/2019

Work Completed: Init project. Setup git for version controlling

Understanding Achieved: Understanding python project structure.

```
(venv) C:\Users\camara\PycharmProjects\as11>git init
Initialized empty Git repository in C:/Users/camara/PycharmProjects/as11/.git/

(venv) C:\Users\camara\PycharmProjects\as11>
```

```
(venv) C:\Users\camara\PycharmProjects\as11>git log --pretty=oneline
fatal: your current branch 'master' does not have any commits yet

(venv) C:\Users\camara\PycharmProjects\as11>
```

Figure 39 Setting up git repository

Date: 28/10/2019

Work Completed: Module Structure

Understanding Achieved: Understanding Python Modular programming and importing and installing 3rd party modules

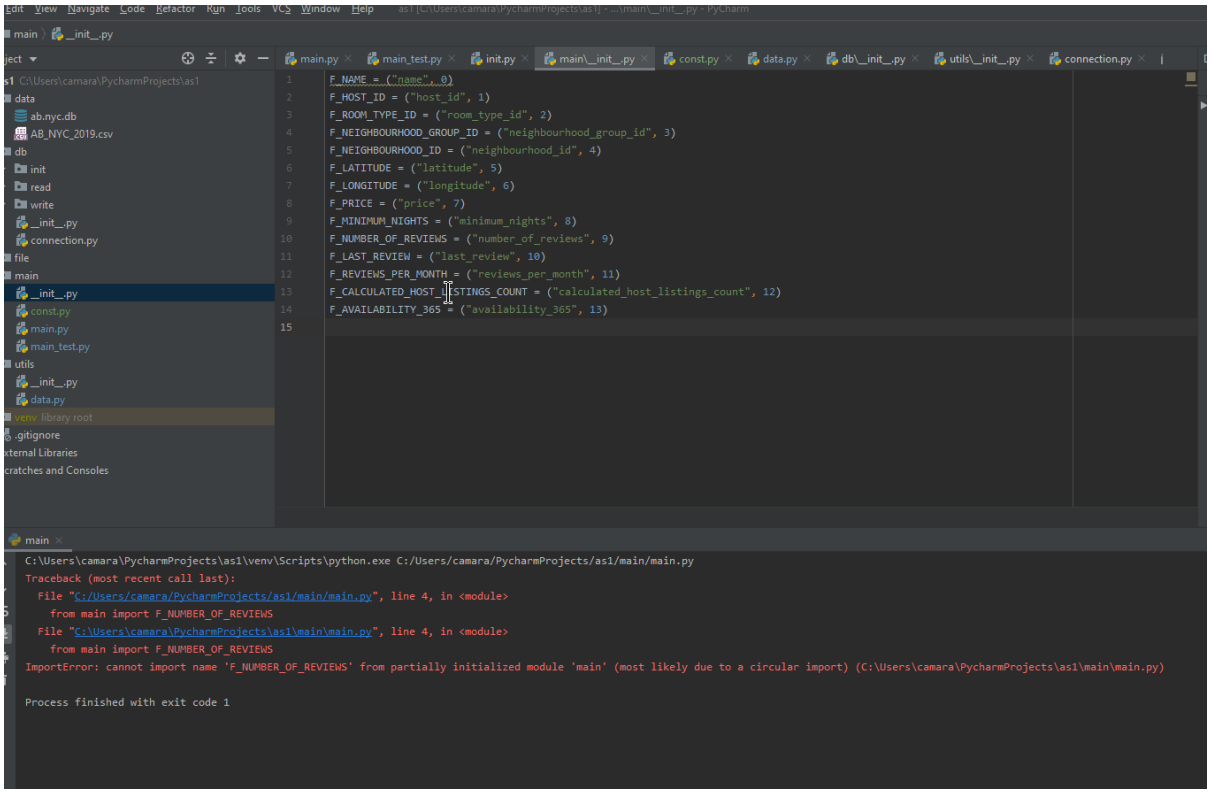


Figure 40 Facing Circular import issue

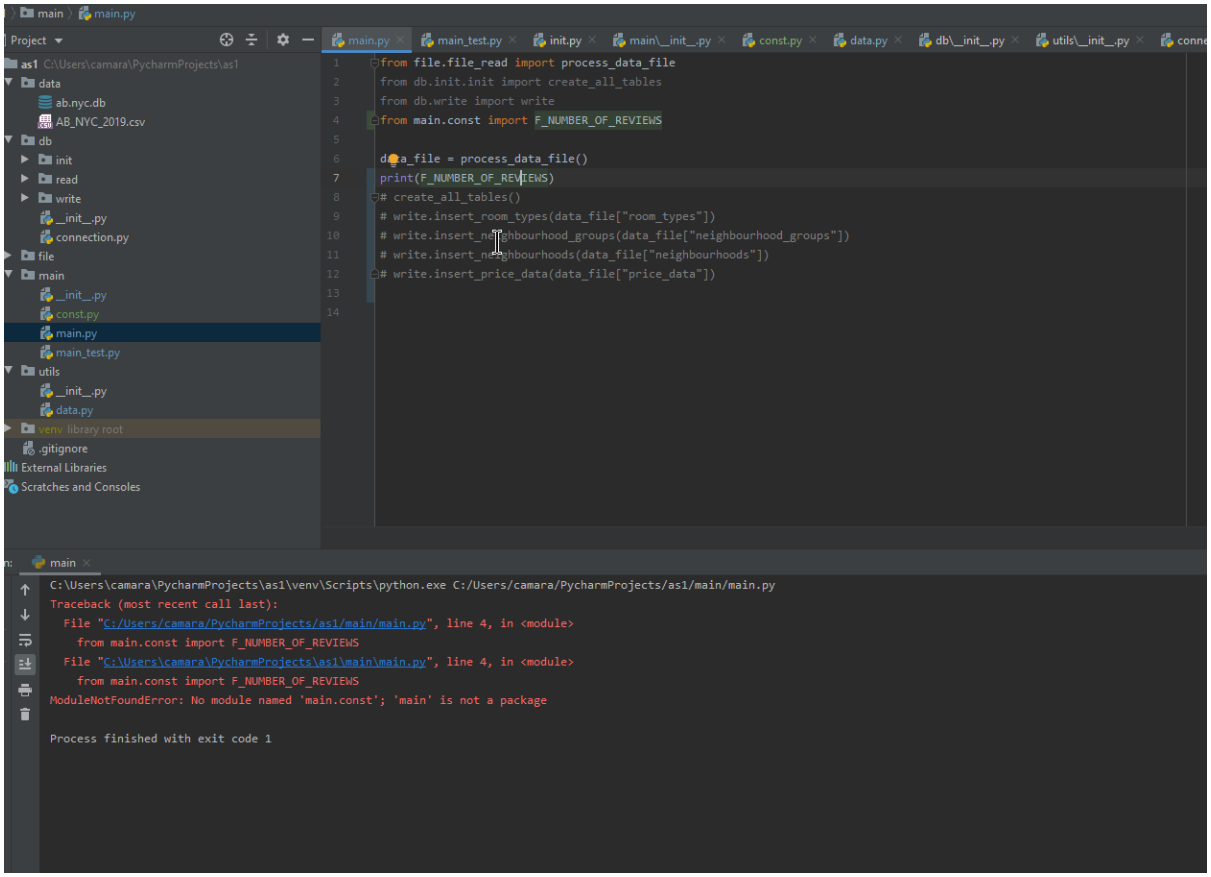


Figure 41 Fixing Module not found issue

Date: 29/10/2019

Work Completed: File Reading

Understanding Achieved: Read CVS file. Encoding issues.

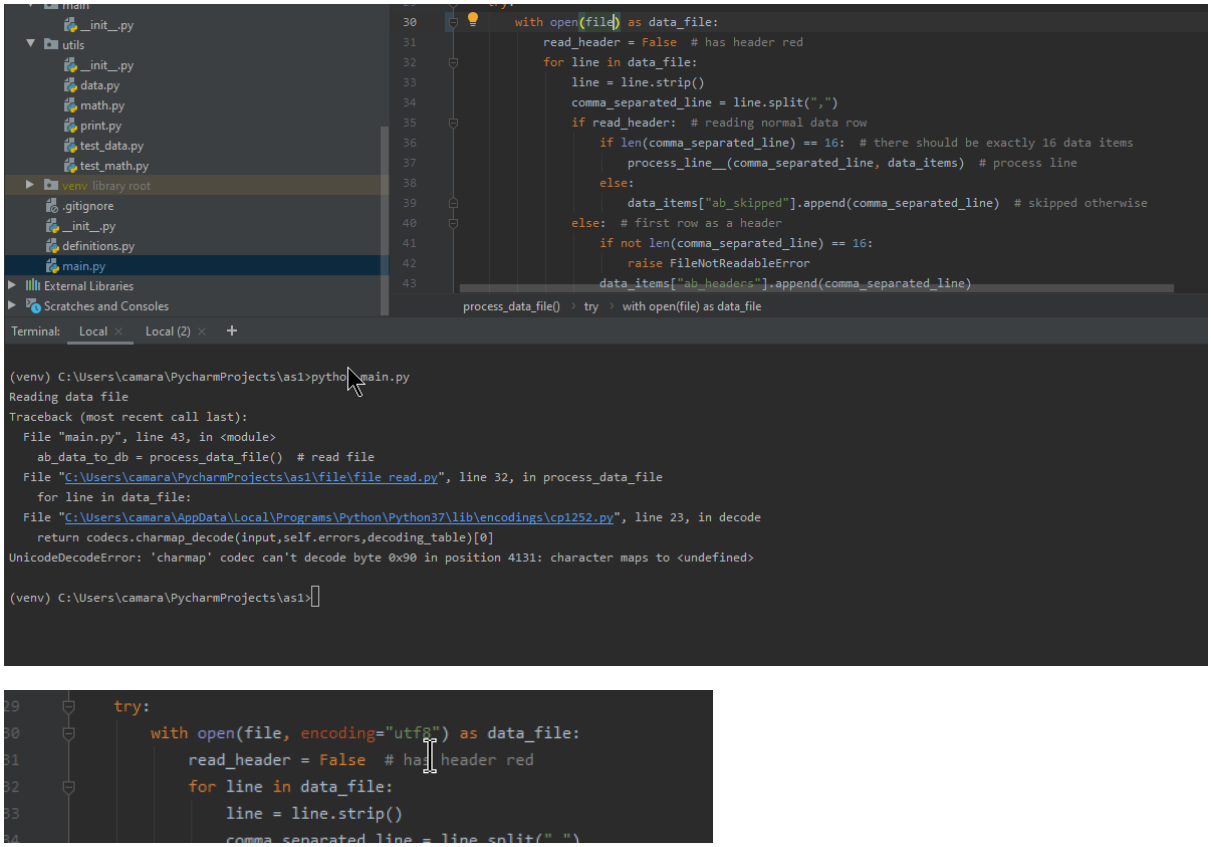


Figure 42 Fixing Encoding issue while reading file

Date: 30/10/2019

Work Completed: Test graph for project

Understanding Achieved: Installing Matplotlib, Matching versions for Matplotlib and python. Visual C++ requirement

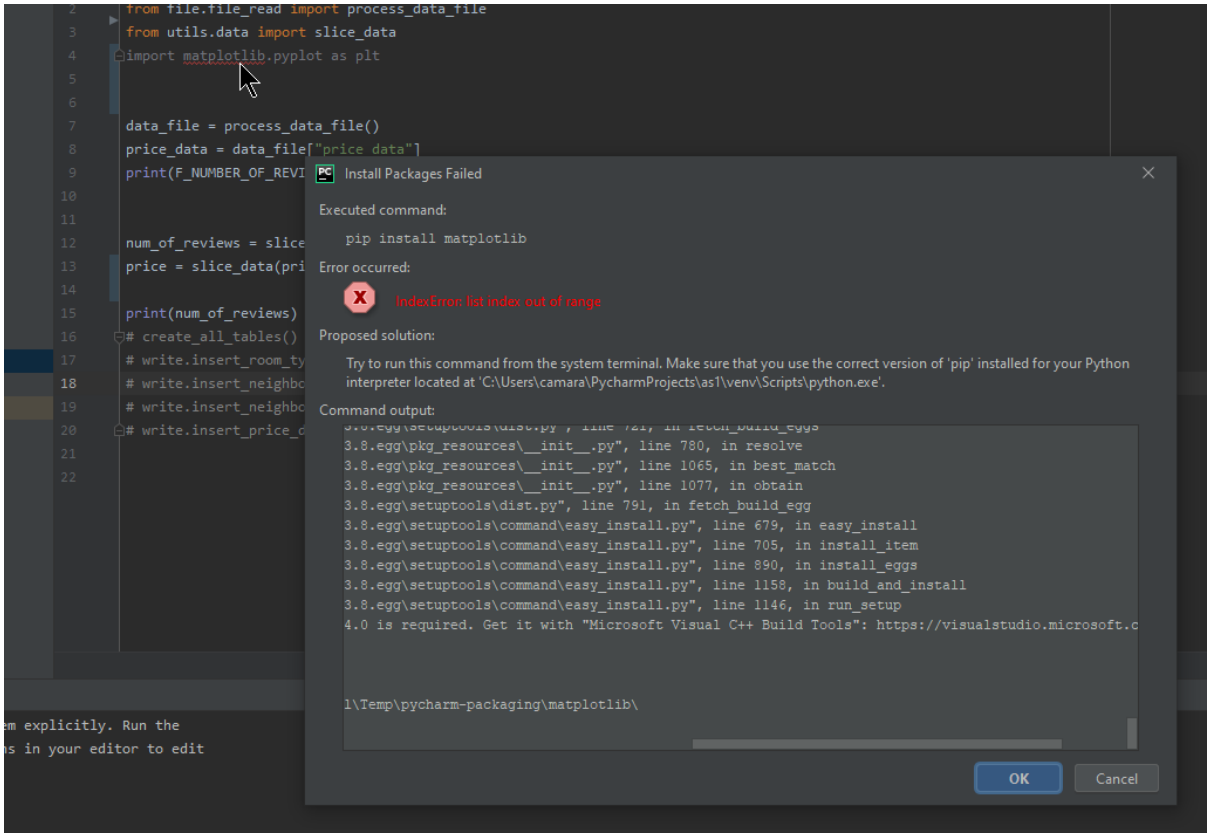


Figure 43 VC++ dependency issue while installation matplotlib

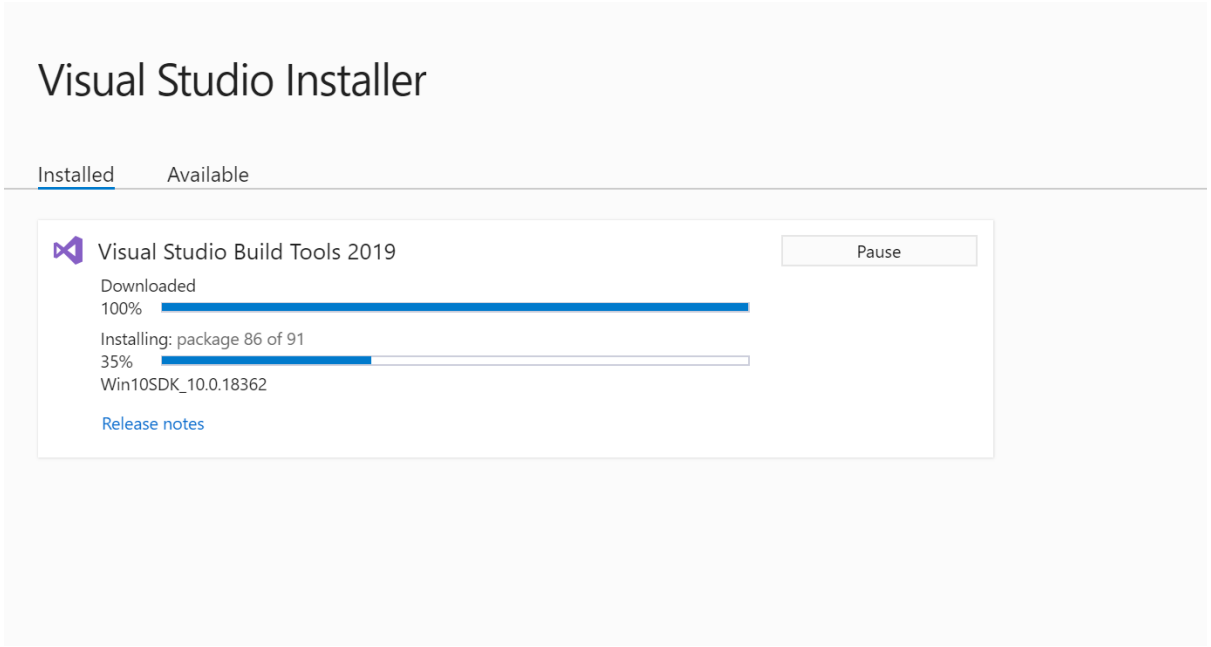


Figure 44 Installing VS Build tool as a fix

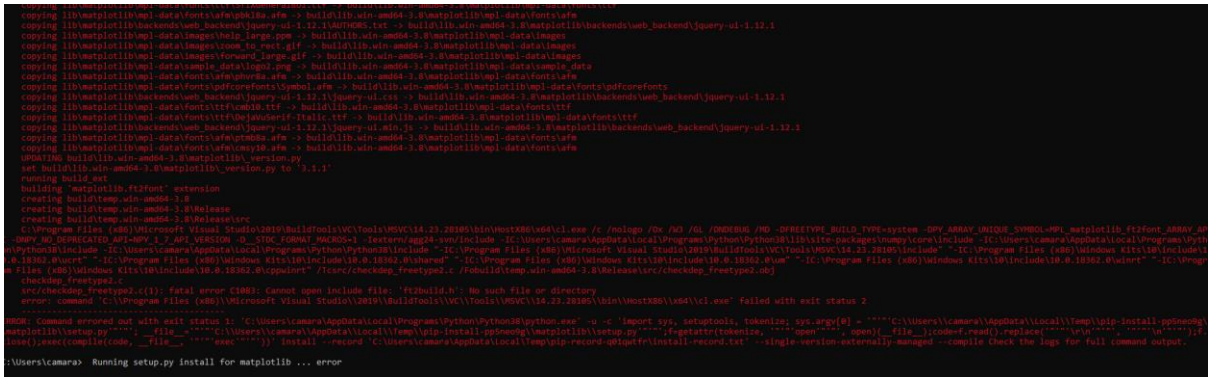


Figure 45 Python 3.8 is not supported but matplotlib

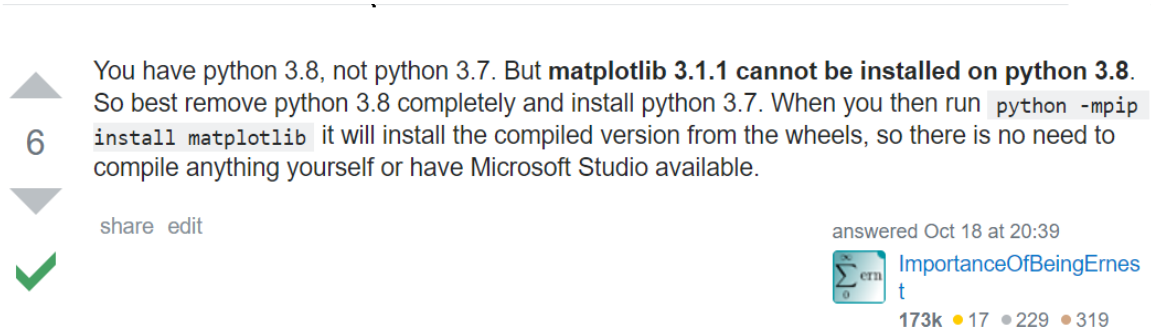


Figure 46 Searched and found solution for matplotlib version issue

Date: 31/10/2019

Work Completed: Basic Maths functions

Understanding Achieved: Meaning of Mean, Mode, Standard Deviation and how to calculate

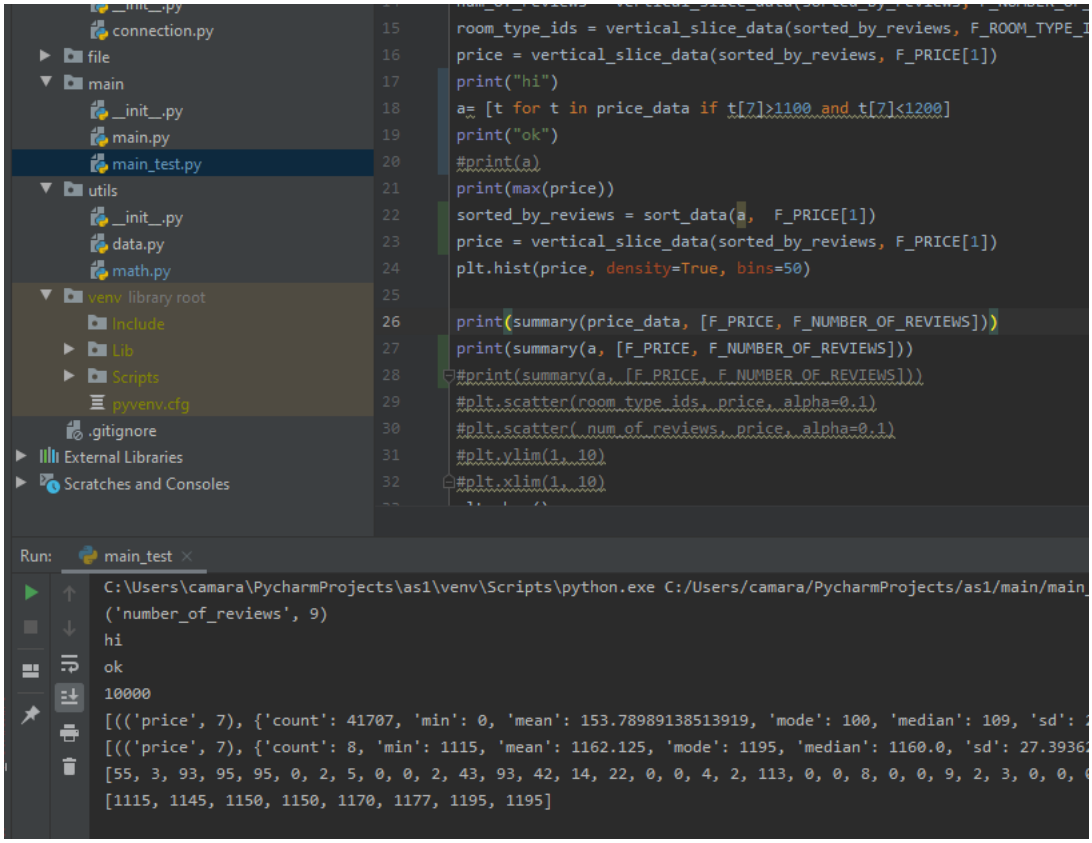


Figure 47 Testing major math functions

Date: 1/11/2019

Work Completed: Testing Scatter Plot

Understanding Achieved: Usage of Scatter Plot, Plotting Matplotlib and Seaborn Scatter plots. Scaling X and Y axis

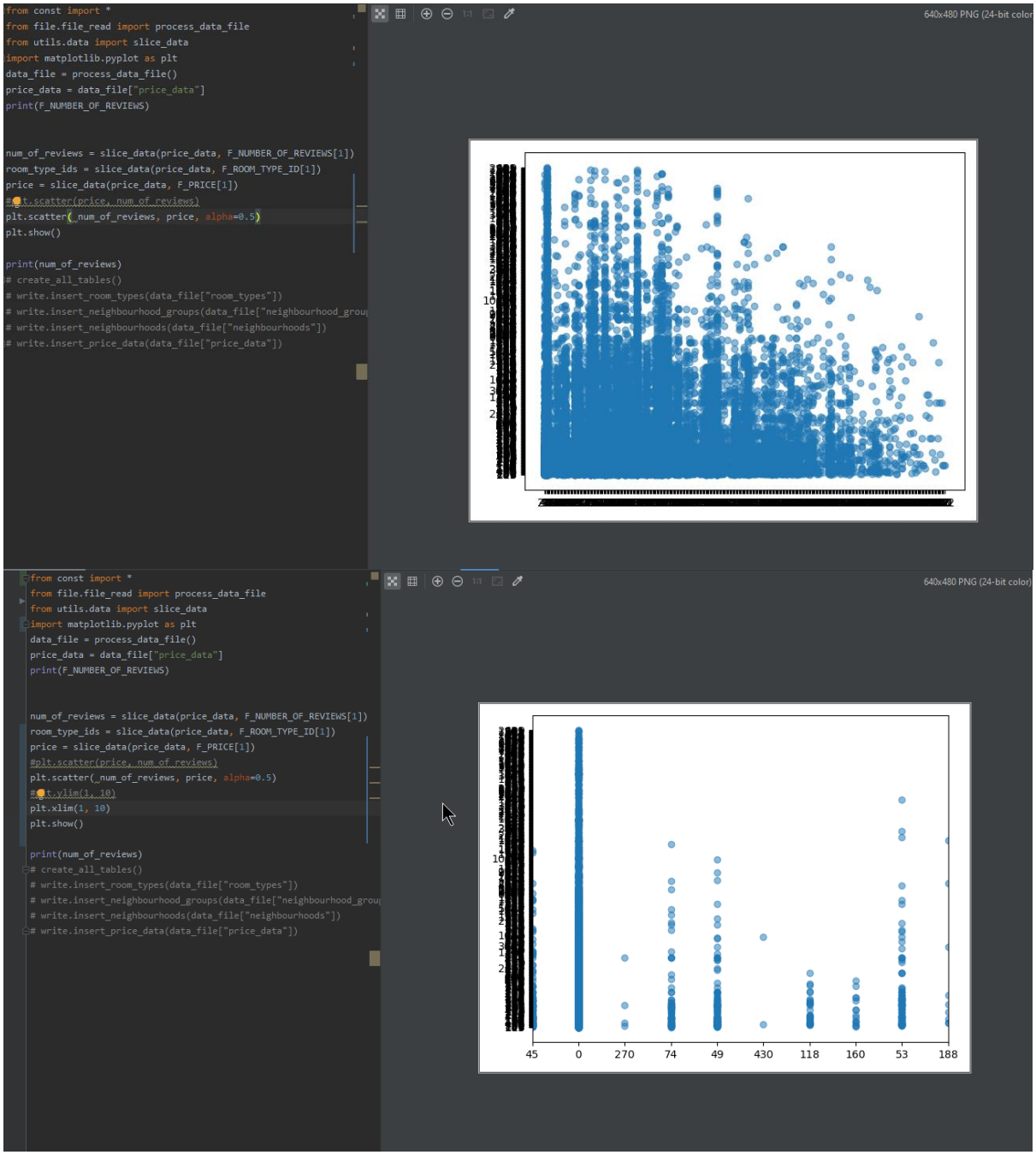


Figure 48 Fixing overlapped x, y labels

Date: 02/11/2019

Work Completed: Pair Scatter Plot

Understanding Achieved: Features of Pair Scatter Plot. DataFrames. Manipulating DataFrames for plotting Pair Scatter Plot without numpy or panda.

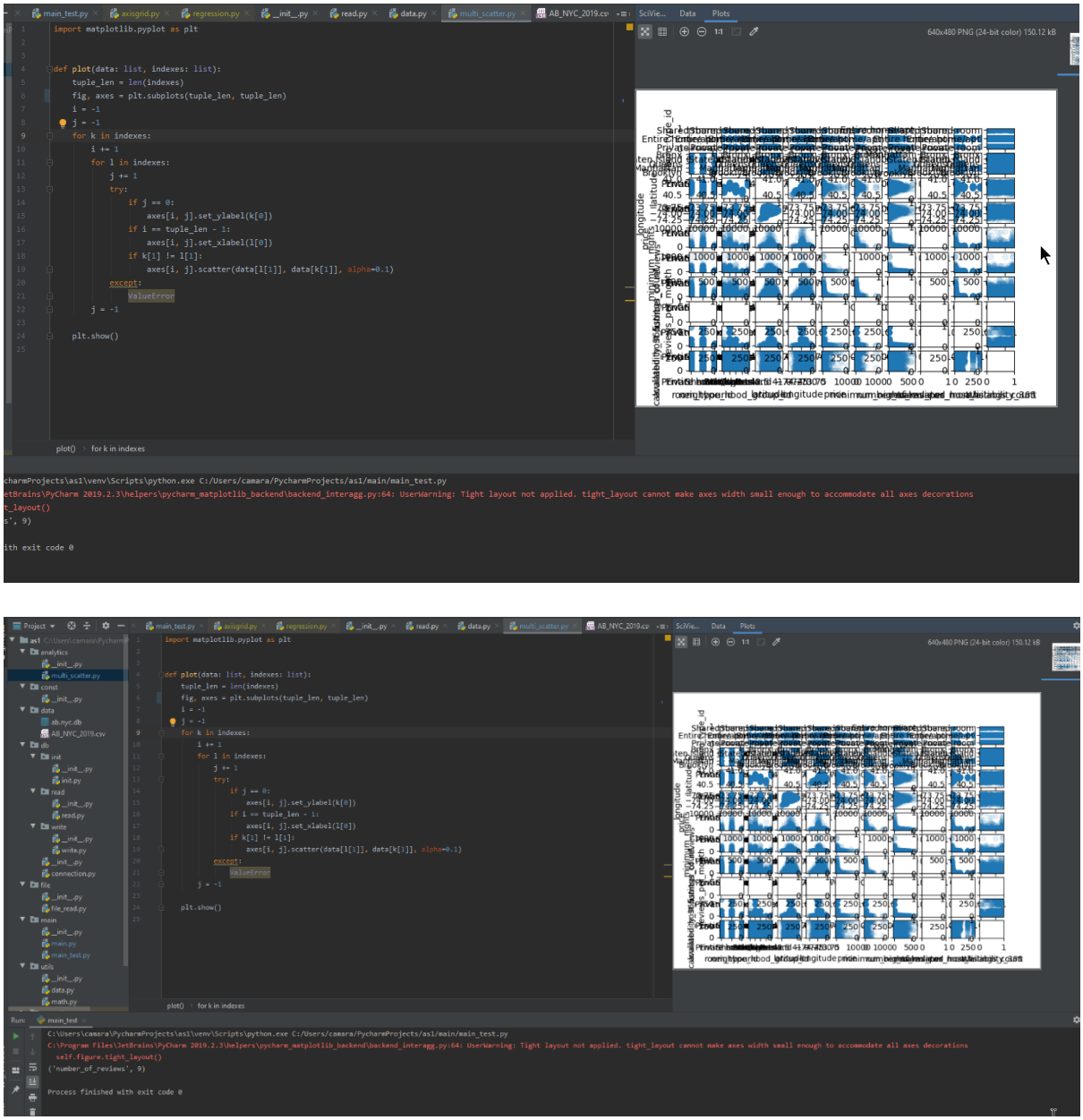


Figure 49 Faced overlapped sub plots

Date: 03/11/2019

Work Completed: Changing the image size based on number of subplots

Understanding Achieved: Change image size.

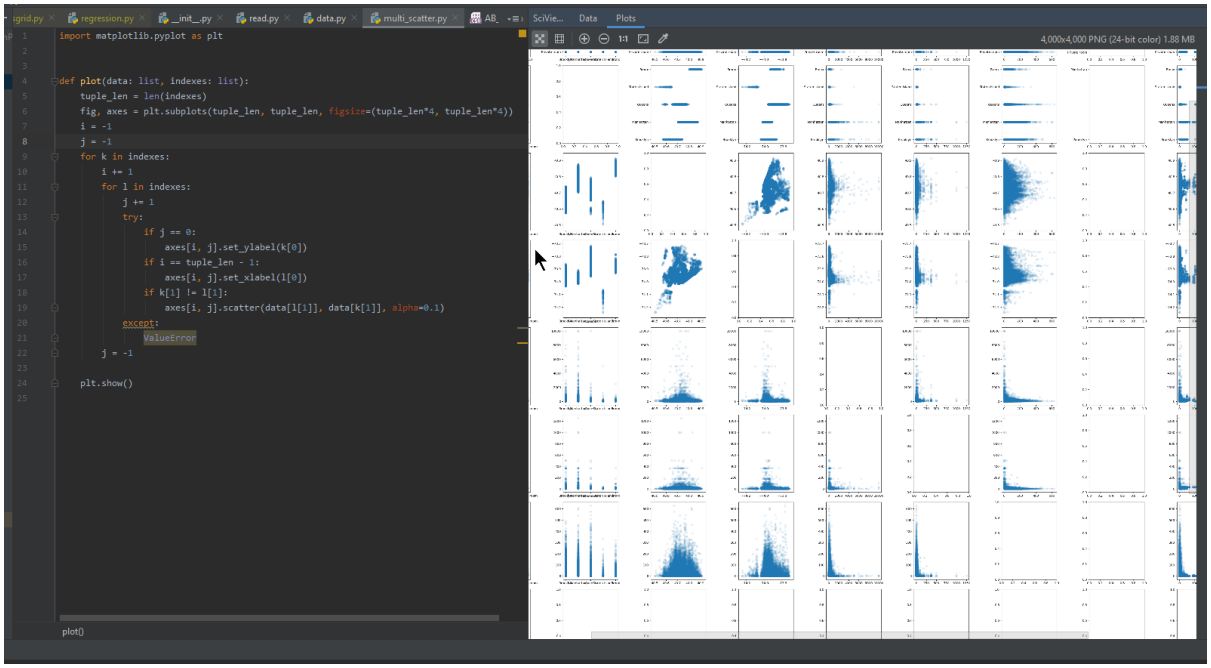


Figure 50 Fixing overlapped subplots by introducing figsize

Date: 04/11/2019

Work Completed: Correcting pair scatter plot for a variable against same variable

Understanding Achieved: Understand how seaborn achieved this

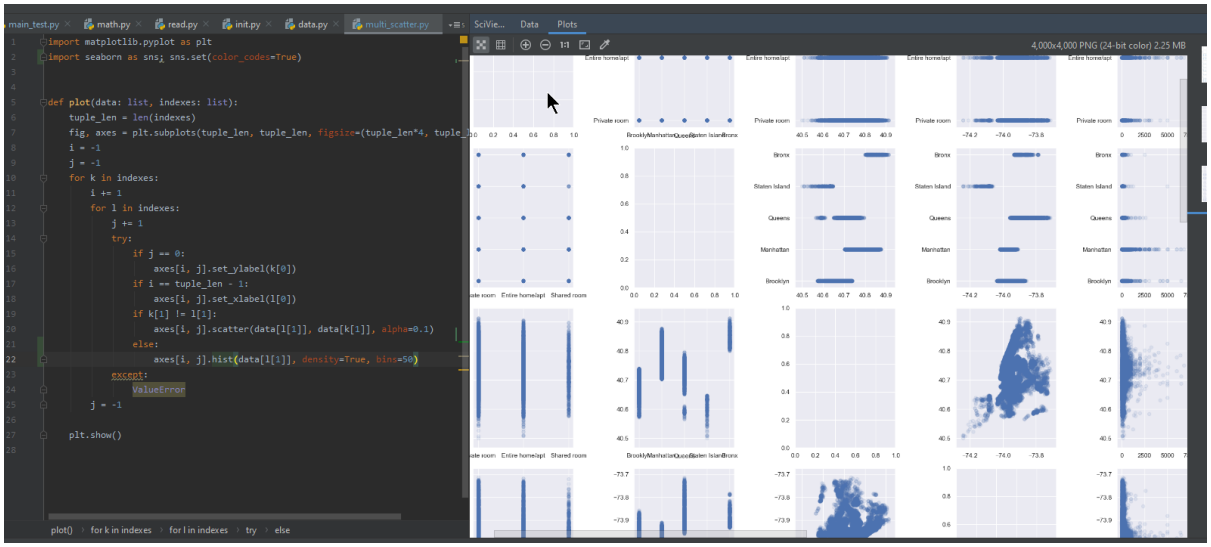


Figure 51 Facing empty subplot for variable to same variable scatter plot

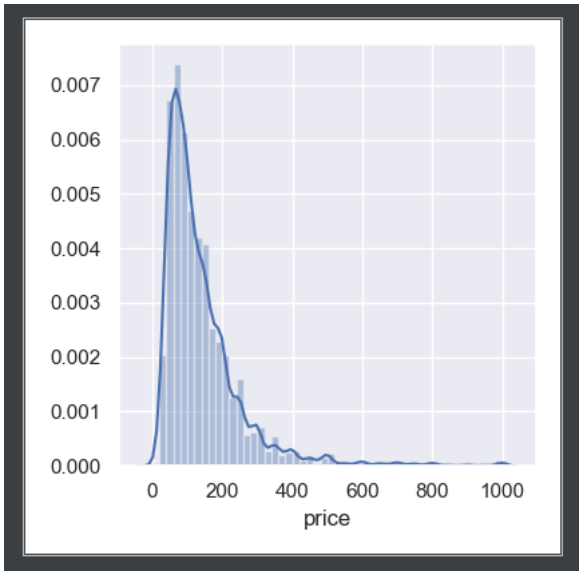
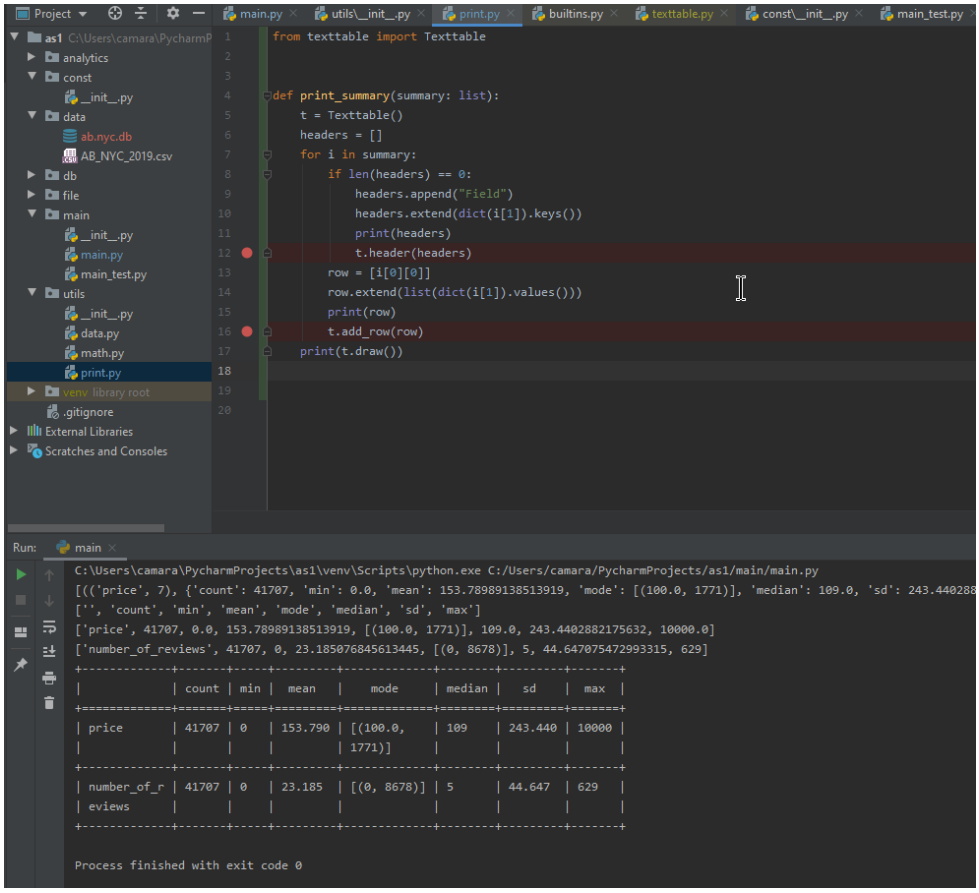


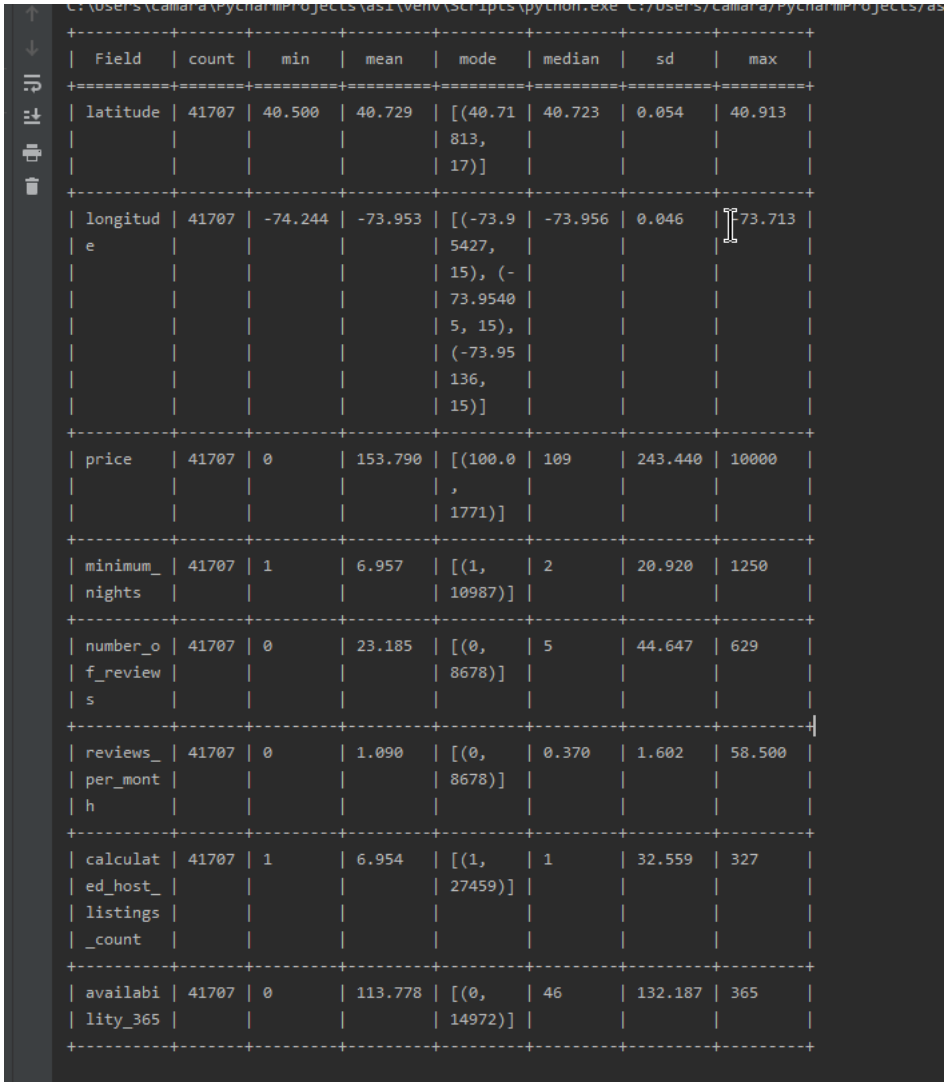
Figure 52 Decided to fill empty subplot by a distribution graph

Date: 05/11/2019

Work Completed: Table Printing

Understanding Achieved: Investigated various modules available for table printing





Field	count	min	mean	mode	median	sd	max
latitude	41707	40.500	40.729	[(40.71813, 17)]	40.723	0.054	40.913
longitude	41707	-74.244	-73.953	[(-73.95427, 15), (-73.95405, 15), (-73.95136, 15)]	-73.956	0.046	-73.713
price	41707	0	153.790	[(100.0, 1771)]	109	243.440	10000
minimum_nights	41707	1	6.957	[(1, 10987)]	2	20.920	1250
number_of_reviews	41707	0	23.185	[(0, 8678)]	5	44.647	629
reviews_per_month	41707	0	1.090	[(0, 8678)]	0.370	1.602	58.500
calculated_host_listings_count	41707	1	6.954	[(1, 27459)]	1	32.559	327
availability_365	41707	0	113.778	[(0, 14972)]	46	132.187	365

Figure 53 Printing summary in a table

Date: 06/11/2019

Work Completed: Categorized data based on Price

Understanding Achieved: Investigating different looping techniques for python. Understand more about Standard Deviation.

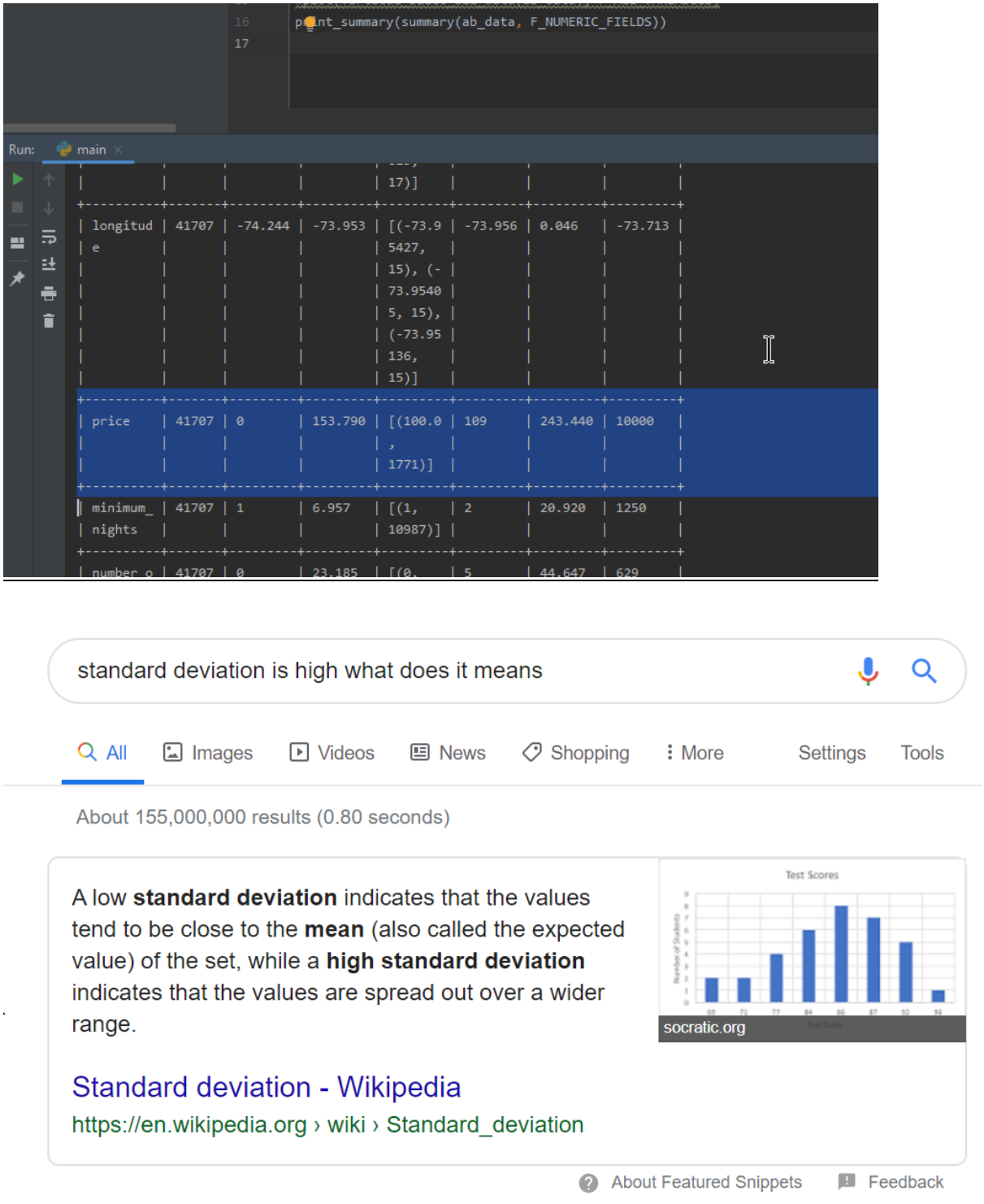
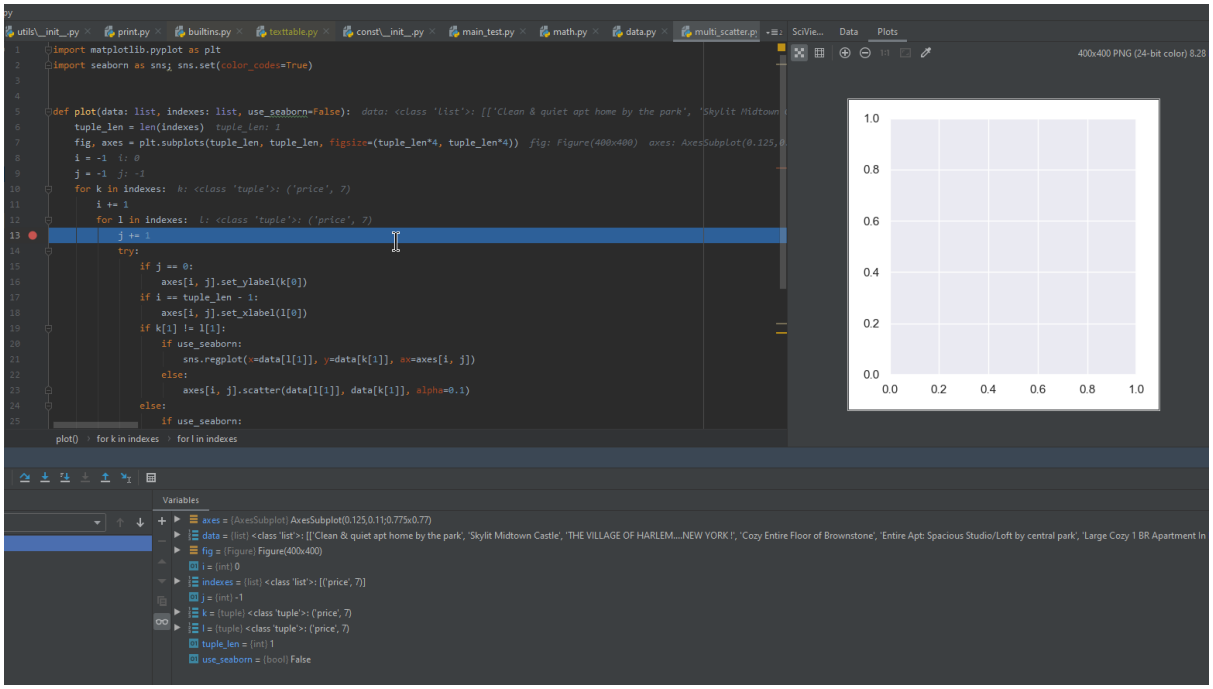


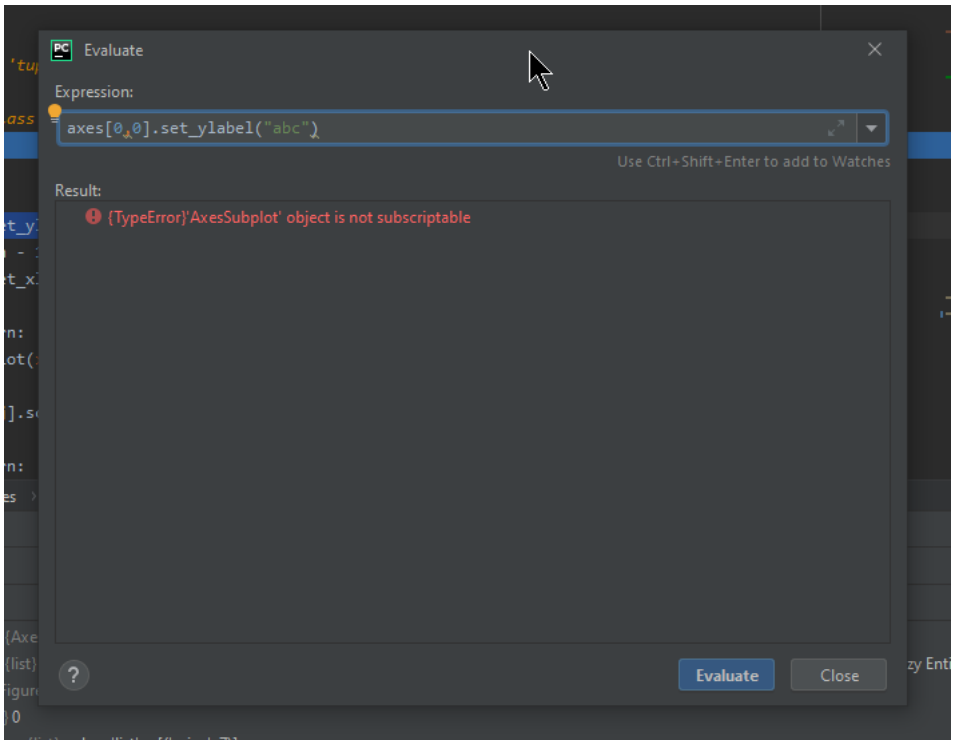
Figure 54 Understanding Standard Deviation

Date: 07/11/2019

Work Completed: Sub plotting

Understanding Achieved: Understand Sub Plotting, figure and axes of matplotlib

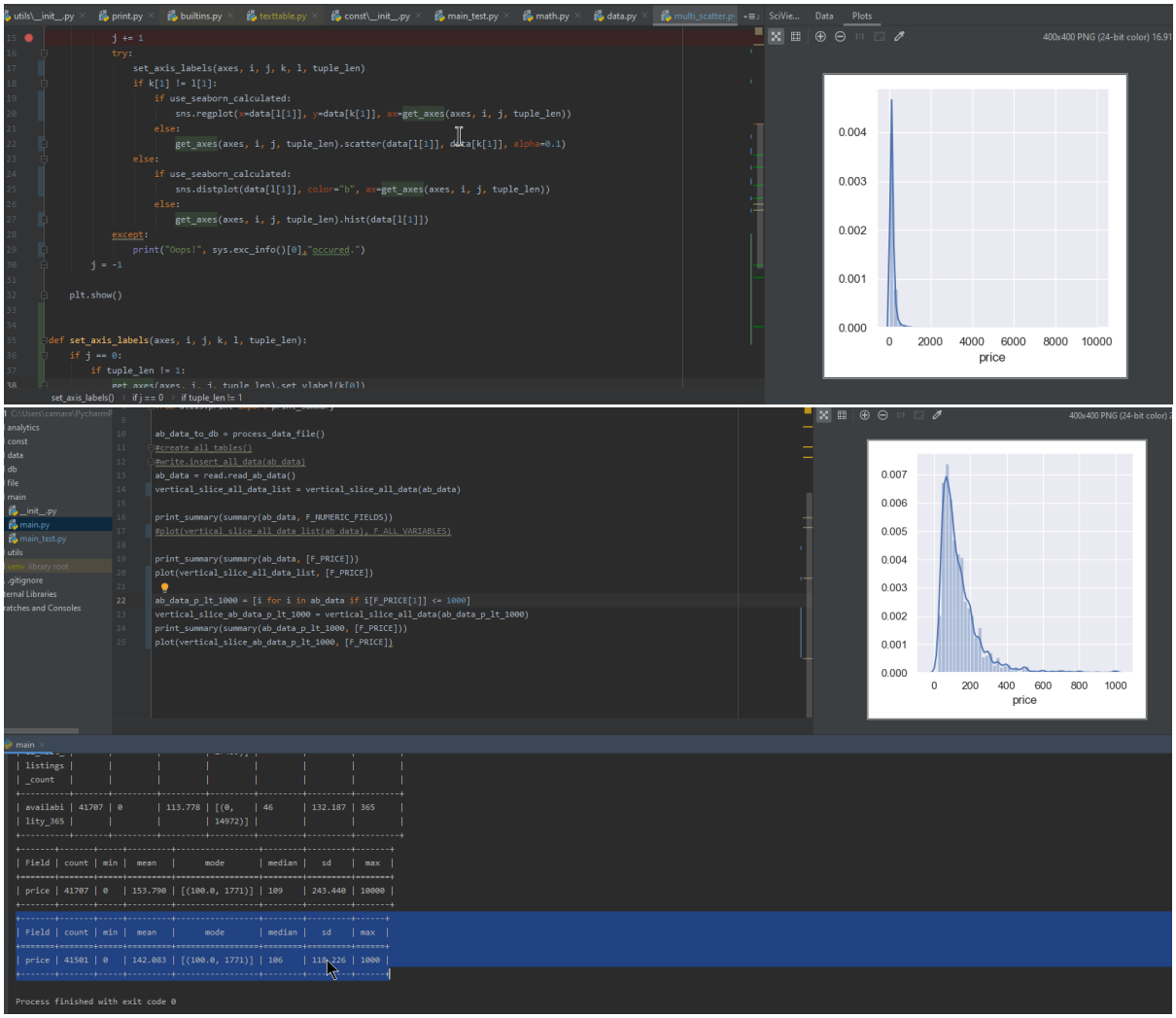


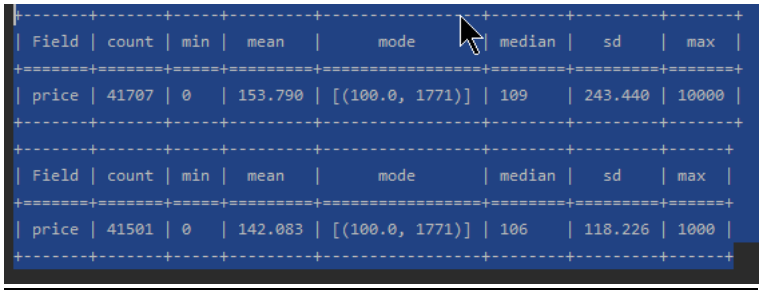



```
fig, axs = plt.subplots()
```

returns a figure with only one single subplot, so axs already holds it without indexing.

Figure 55 Sub plotting without index causing issues





Field	count	min	mean	mode	median	sd	max
price	41707	0	153.790	[(100.0, 1771)]	109	243.440	10000

Field	count	min	mean	mode	median	sd	max
price	41501	0	142.083	[(100.0, 1771)]	106	118.226	1000

Date: 09/11/2019

Work Completed: Grouped bar chat

Understanding Achieved: Data input formats for grouped bar charts.

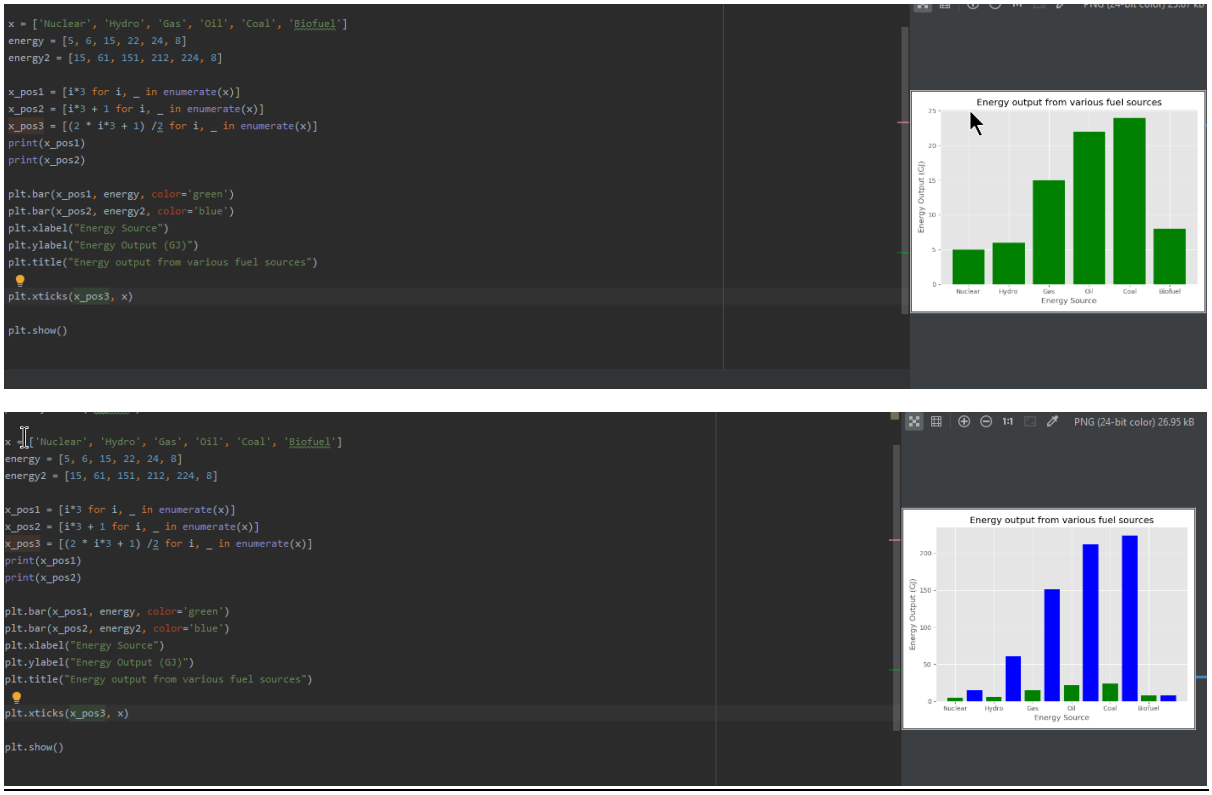


Figure 56 Testing grouped bar charts

Date: 10/11/2019

Work Completed: Pie Charts

Understanding Achieved: Understanding Data format for Pie Charts.

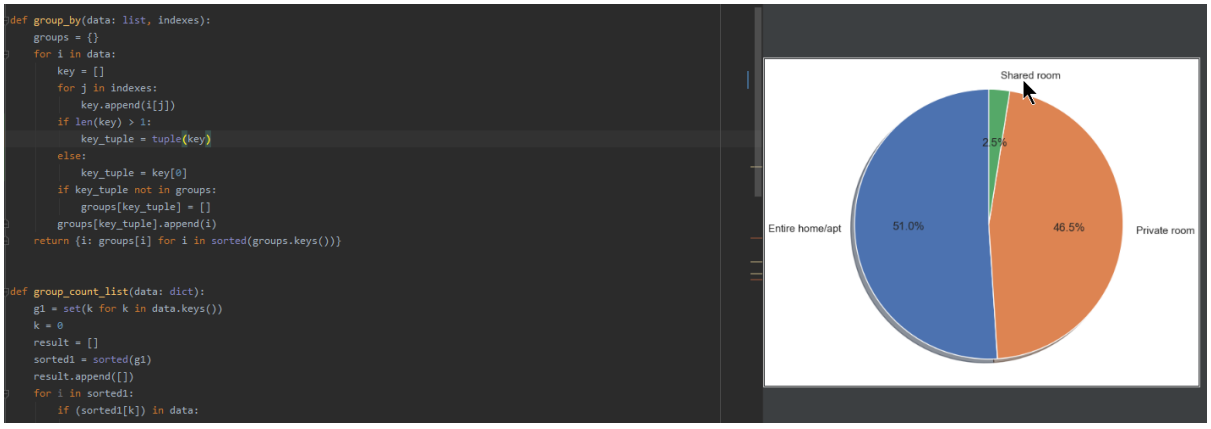


Figure 57 Testing pie charts

Date: 11/11/2019

Work Completed: Graph Saving

Understanding Achieved: Configurations available for graph saving

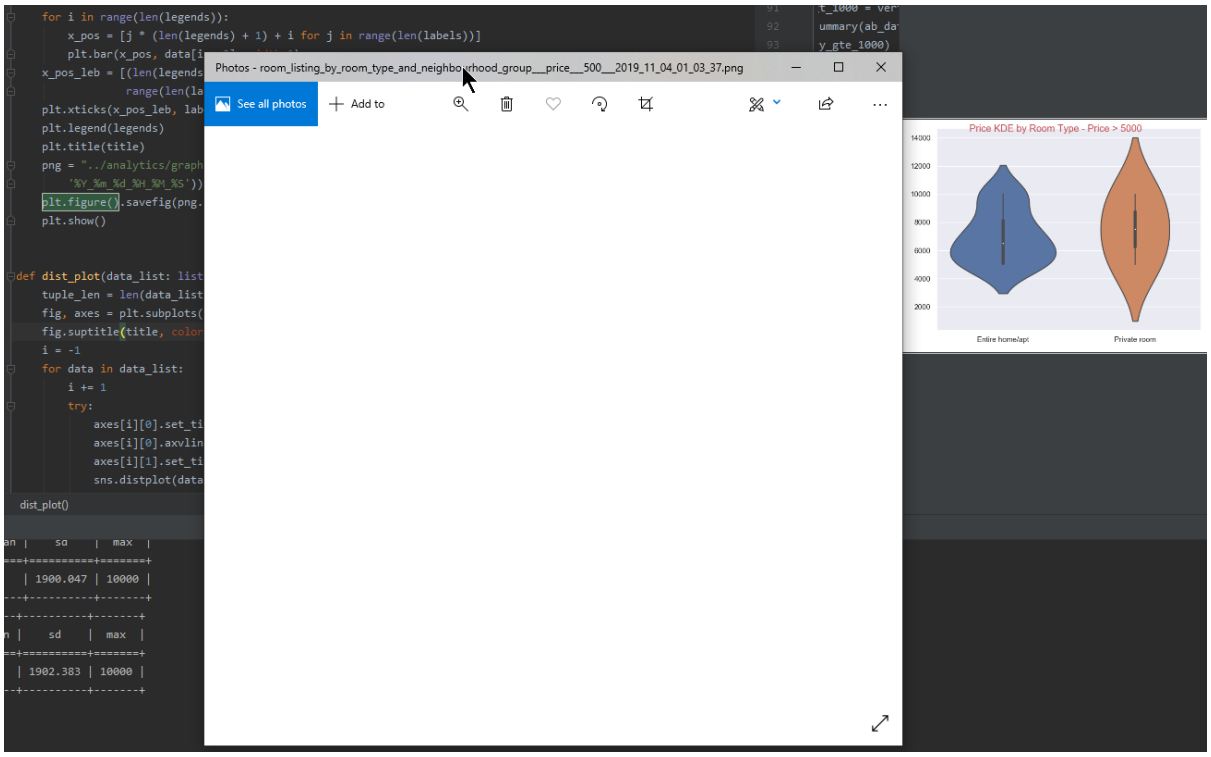


Figure 58 Image saving issue

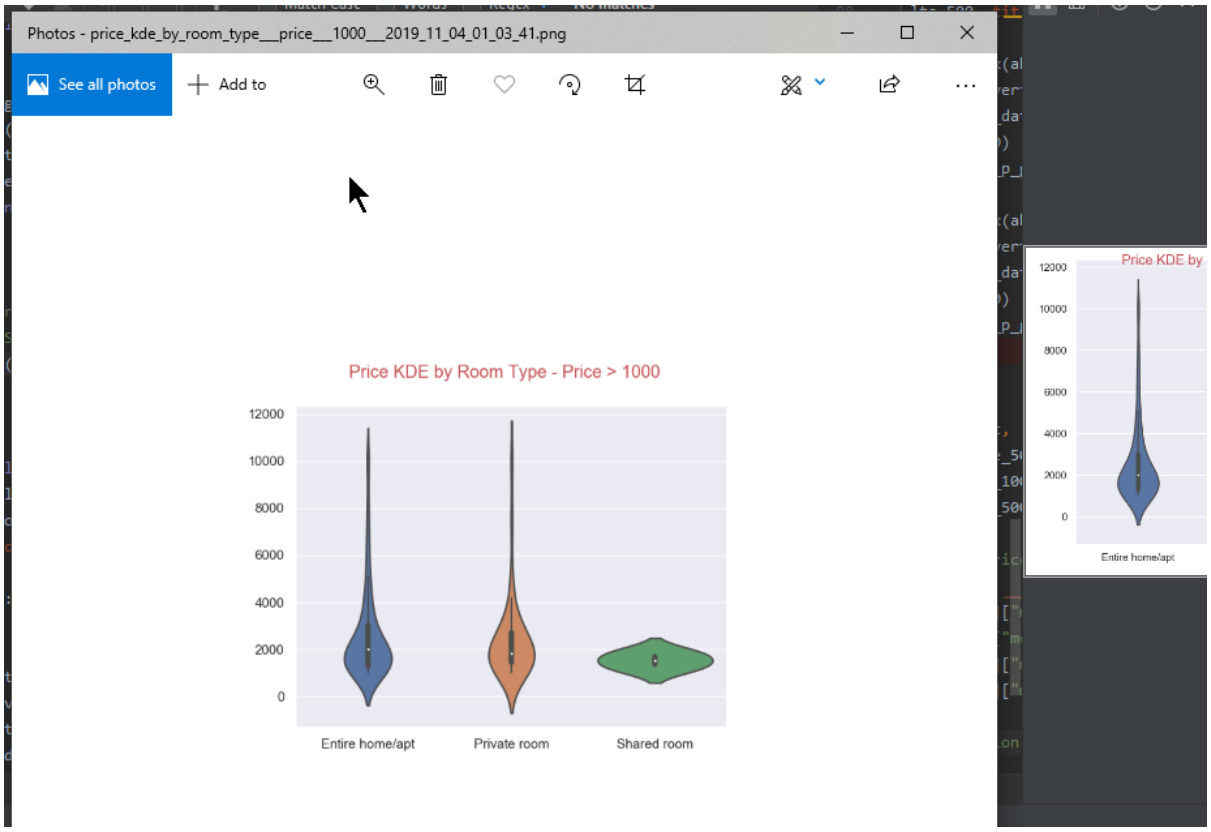


Figure 59 Fixing image save

Date: 13/11/2019

Work Completed: Initial Setup for Unit Testing

Understanding Achieved: Research on available unit testing modules. Unit Test Coverage

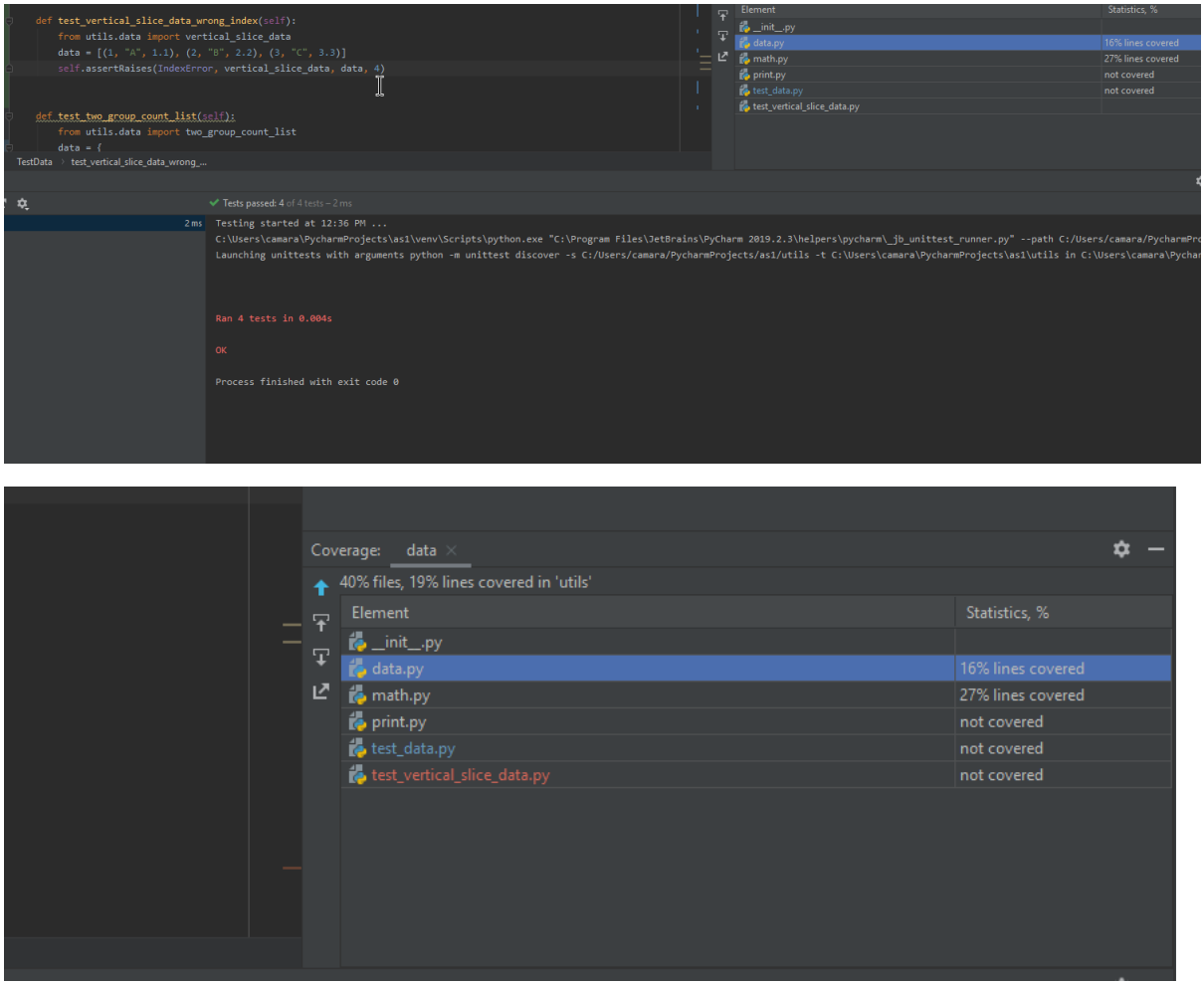


Figure 60 Unittesting and test coverage

Date: 14/11/2019

Work Completed: Unit Testing for Exceptions

Understanding Achieved: Parameter passing for Exception testing

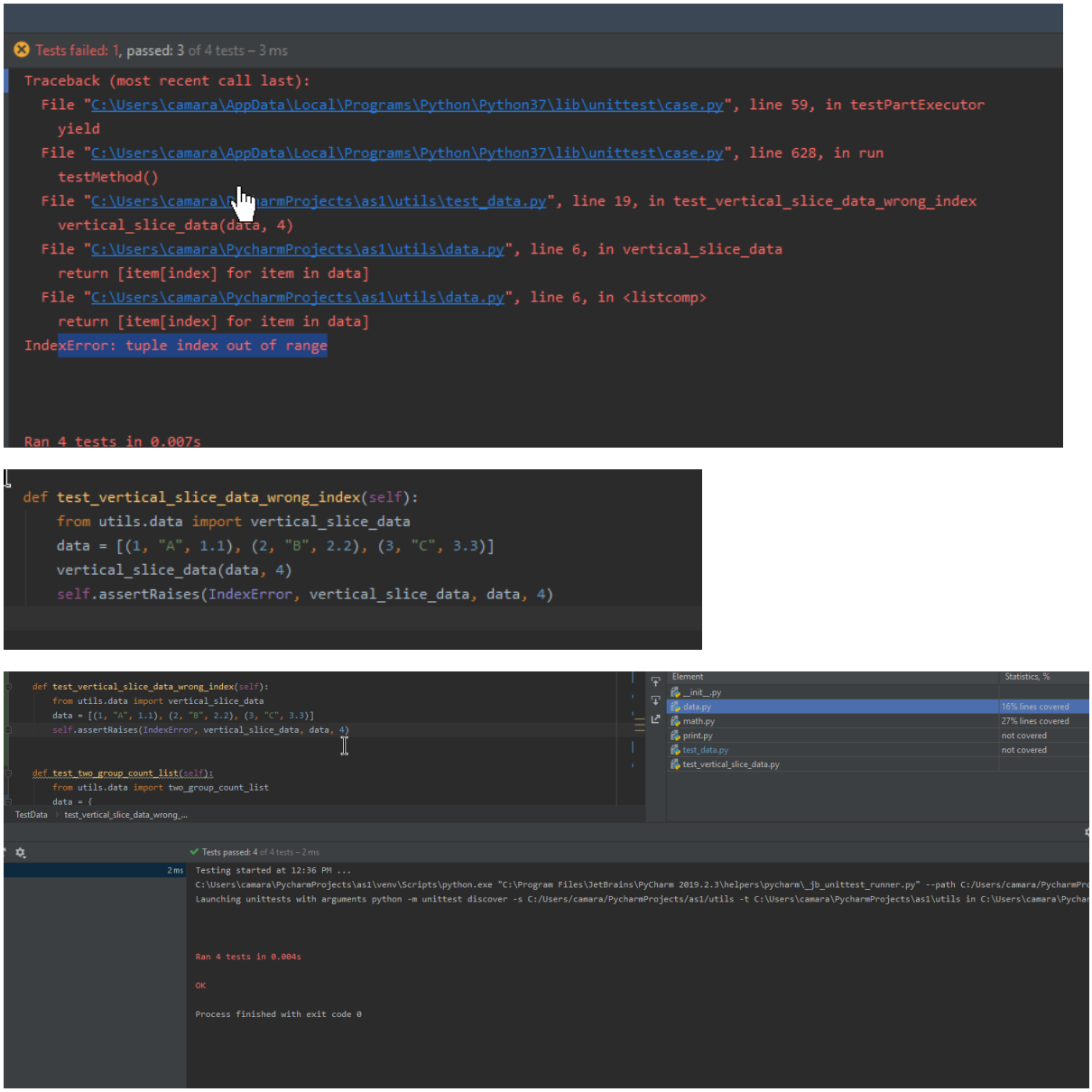


Figure 61 Unit testing for exceptions

Date: 17/11/2019

Work Completed: Initial steps for Pydoc

Understanding Achieved: Research on available documentation patterns

Figure 62 Python doc Google Standard

Date: 14/11/2019

Work Completed: Pydoc

Understanding Achieved: Pydoc

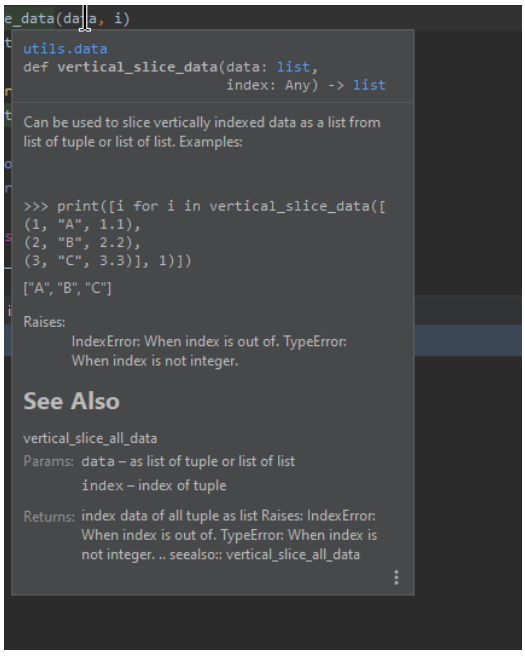


Figure 63 Testing Python doc. See Also links are not working

Date: 18/11/2019

Work Completed: Fixed static code analysis issues

Understanding Achieved: Understanding python coding best practises

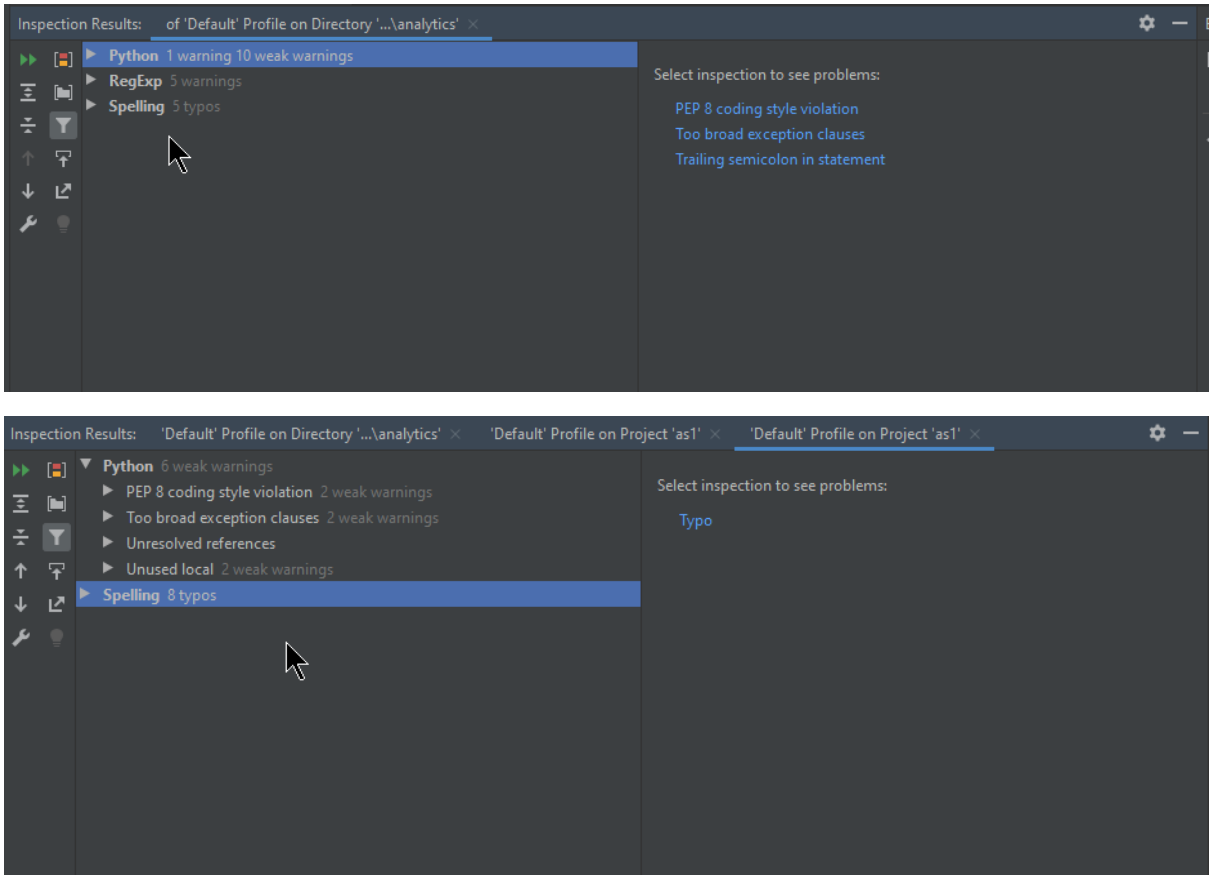
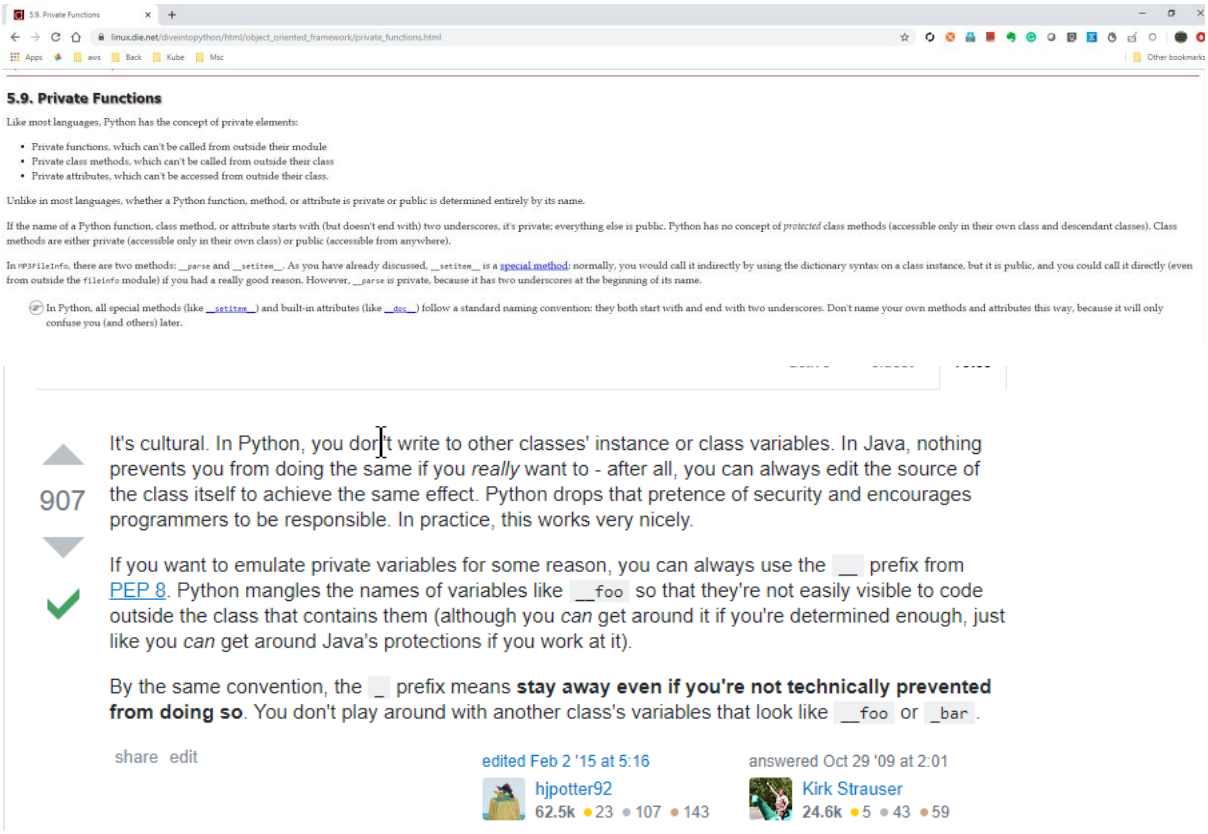


Figure 64 Fixing Static code analysis warning and errors

Date: 20/11/2019

Work Completed: Refactoring code

Understanding Achieved: Research more on Coding best practises



5.9. Private Functions

Like most languages, Python has the concept of private elements:

- Private functions, which can't be called from outside their module
- Private class methods, which can't be called from outside their class
- Private attributes, which can't be accessed from outside their class.

Unlike in most languages, whether a Python function, method, or attribute is private or public is determined entirely by its name.

If the name of a Python function, class method, or attribute starts with (but doesn't end with) two underscores, it's private; everything else is public. Python has no concept of *protected* class methods (accessible only in their own class and descendant classes). Class methods are either private (accessible only in their own class) or public (accessible from anywhere).

In `MP3FileInfo`, there are two methods: `__parse` and `__setitem__`. As you have already discussed, `__setitem__` is a special method; normally, you would call it indirectly by using the dictionary syntax on a class instance, but it is public, and you could call it directly (even from outside the `fileinfo` module) if you had a really good reason. However, `__parse` is private, because it has two underscores at the beginning of its name.

In Python, all special methods (like `__setitem__`) and built-in attributes (like `__doc__`) follow a standard naming convention: they both start with and end with two underscores. Don't name your own methods and attributes this way, because it will only confuse you (and others) later.

907

It's cultural. In Python, you don't write to other classes' instance or class variables. In Java, nothing prevents you from doing the same if you *really* want to - after all, you can always edit the source of the class itself to achieve the same effect. Python drops that pretence of security and encourages programmers to be responsible. In practice, this works very nicely.

If you want to emulate private variables for some reason, you can always use the `__` prefix from PEP 8. Python mangles the names of variables like `__foo` so that they're not easily visible to code outside the class that contains them (although you *can* get around it if you're determined enough, just like you *can* get around Java's protections if you work at it).

By the same convention, the `_` prefix means **stay away even if you're not technically prevented from doing so**. You don't play around with another class's variables that look like `__foo` or `_bar`.

share edit

edited Feb 2 '15 at 5:16 hjpotter92 62.5k ● 23 ● 107 ● 143

answered Oct 29 '09 at 2:01 Kirk Strauser 24.6k ● 5 ● 43 ● 59

Figure 65 Finding a way to hide internal methods

Date: 23/11/2019

Work Completed: Fixed intermittently failing test cases because of order of data set

Understanding Achieved: Understanding order of list, tuple and dict

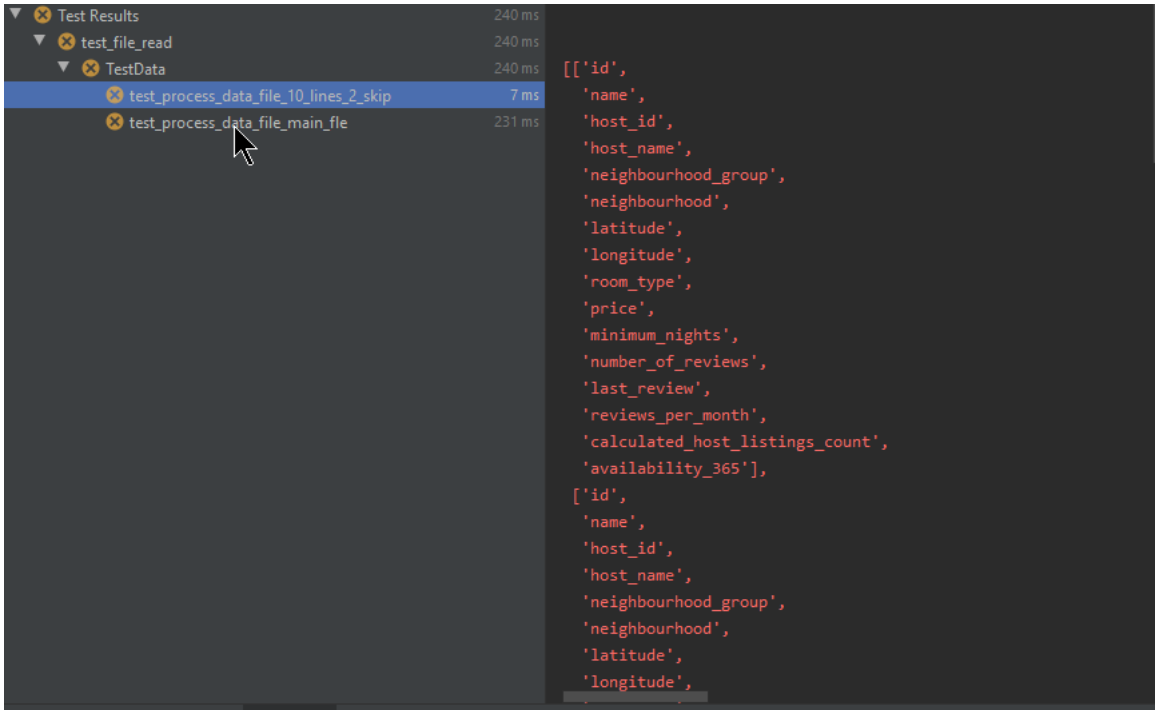


Figure 66 Unit tests are failing intermittently because of output order

Date: 24/11/2019

Work Completed: Fixed relative path and absolute path issue for data files

Understanding Achieved: Understanding how relative path and absolute path works in python

```
During handling of the above exception, another exception occurred:

Traceback (most recent call last):
  File "C:\Users\camara\PycharmProjects\as1\file\test_file_read.py", line 36, in test_process_data_file_10_lines_2_skip
    data = process_data_file("../data/tmp/AB_NYC_10_lines_2_skip.csv")
  File "C:\Users\camara\PycharmProjects\as1\file\file_read.py", line 46, in process_data_file
    raise FileNotReadableError
exceptions.file_exceptions.FileNotReadableError

======================================================================
ERROR: test_process_data_file_main_fle (file.test_file_read.TestData)
----------------------------------------------------------------------
Traceback (most recent call last):
  File "C:\Users\camara\PycharmProjects\as1\file\file_read.py", line 29, in process_data_file
    with open(file, encoding="utf8") as data_file:
FileNotFoundError: [Errno 2] No such file or directory: '../data/AB_NYC_2019.csv'

During handling of the above exception, another exception occurred:

Traceback (most recent call last):
  File "C:\Users\camara\PycharmProjects\as1\file\test_file_read.py", line 67, in test_process_data_file_main_fle
    data = process_data_file()
  File "C:\Users\camara\PycharmProjects\as1\file\file_read.py", line 46, in process_data_file
    raise FileNotReadableError
exceptions.file_exceptions.FileNotReadableError

----------------------------------------------------------------------
Ran 17 tests in 0.008s
```

Figure 67 File reading unit test are failing based on the location of test execution.

Date: 25/11/2019

Work Completed: Fixed static code analysis issue

Understanding Achieved: Understanding python coding best practises

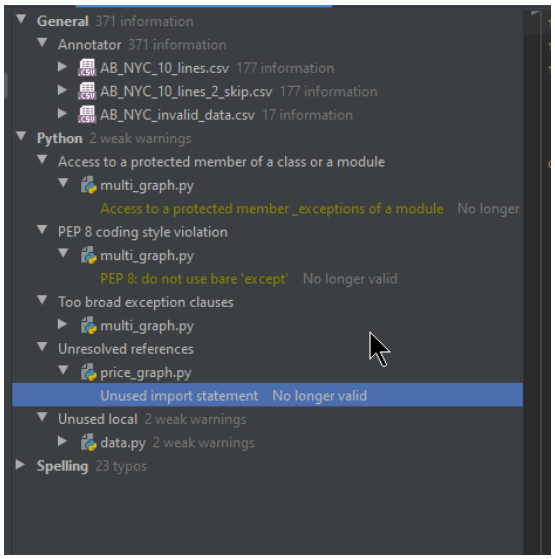


Figure 68 Code refactoring

Date: 26/11/2019

Work Completed: Fixed large table printing issues

Understanding Achieved: Understand how to change the table size for texttable

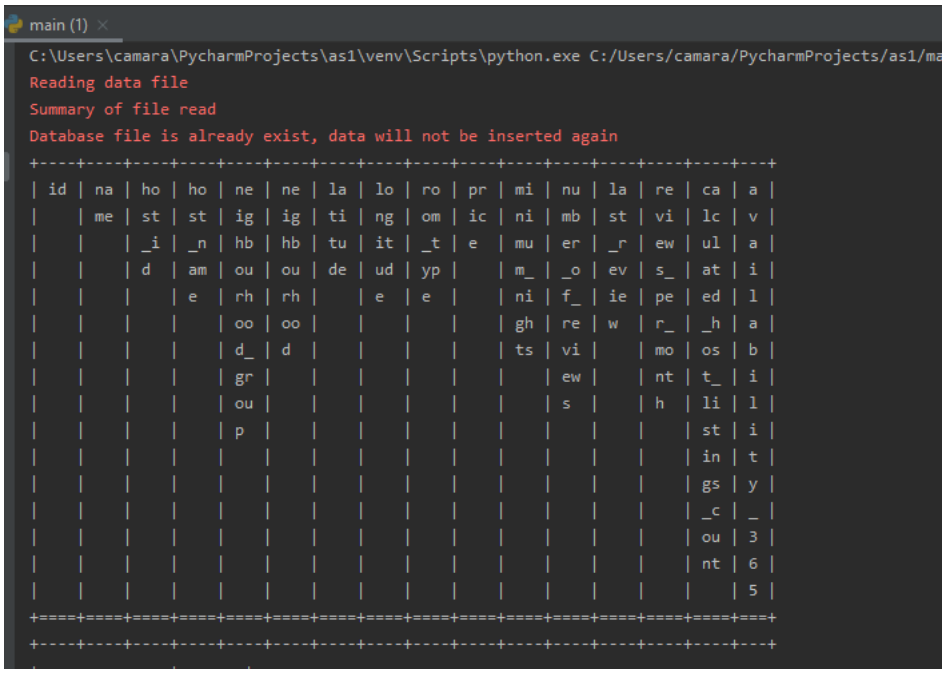


Figure 69: Large table print casing alignment issues

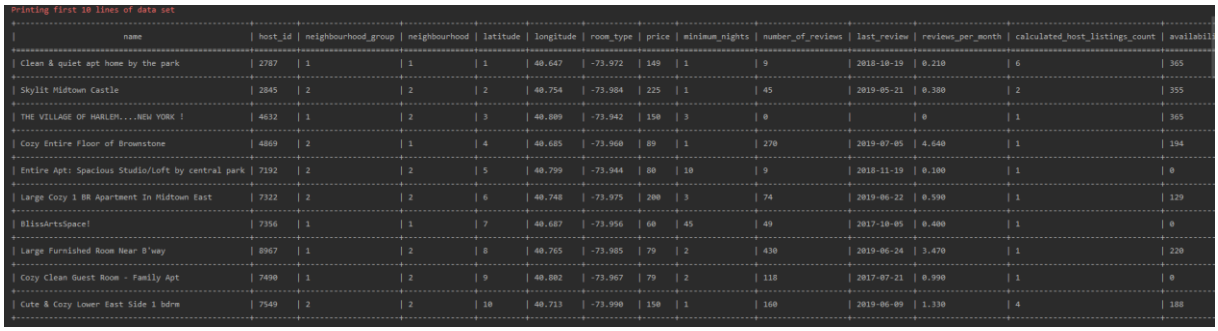


Figure 70: Fixed table size issue

Date: 27/11/2019

Work Completed: User friendly logging for python

Understanding Achieved: Understanding available logging options for python

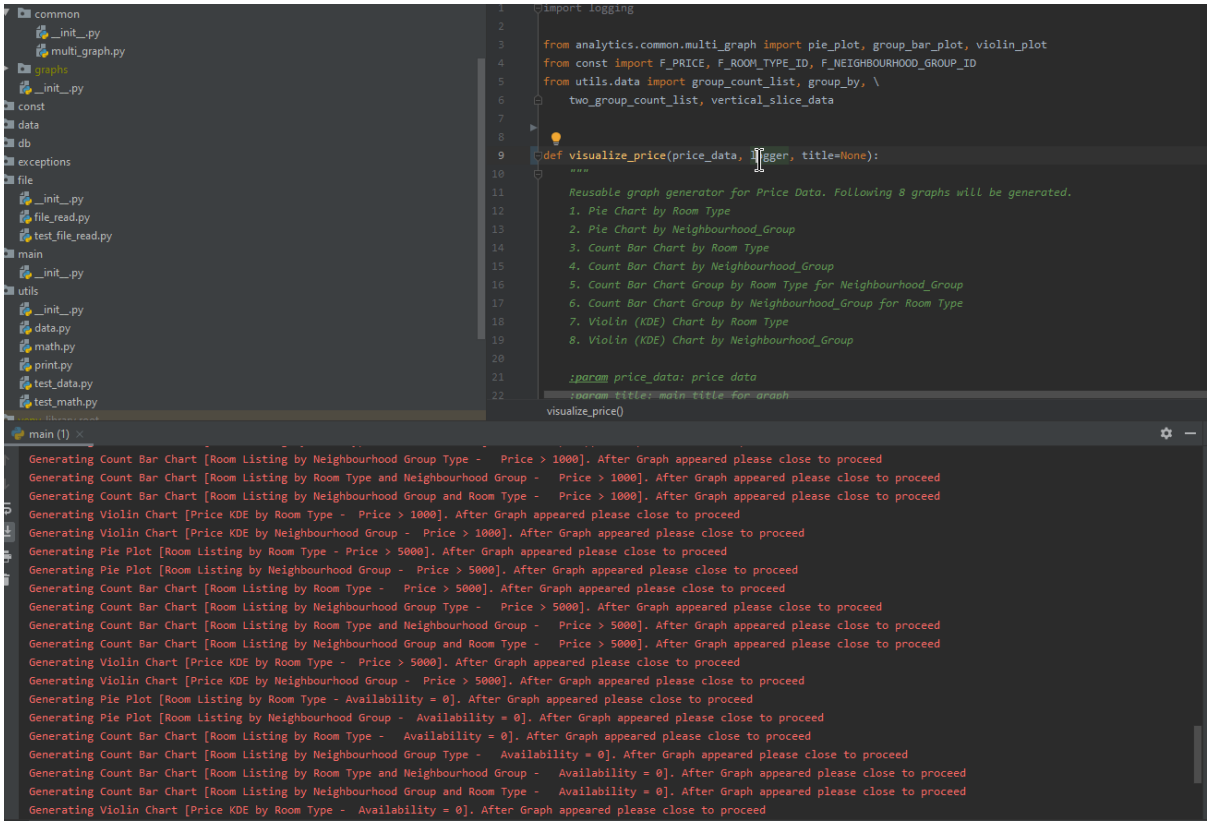


Figure 71: Introducing user friendly logging

Date: 28/11/2019

Work Completed: full git commit logs

```
(venv) C:\Users\camara\PycharmProjects\as1>git log --pretty=format:"%h%x09%an%x09%ad%x09%s"
de1b7b9 Chaminda Amarasinghe    Sat Dec 7 04:23:59 2019 +0000   added price summary for availability
2be2cca Chaminda Amarasinghe    Wed Dec 4 07:38:02 2019 +0000   added % for file read summary
1ad8b2a Chaminda Amarasinghe    Wed Dec 4 07:00:43 2019 +0000   fixed unittests after skipping headers
e73a169 Chaminda Amarasinghe    Tue Nov 19 00:11:48 2019 +0000  typo
0222ec7 Chaminda Amarasinghe    Mon Nov 18 23:53:50 2019 +0000  typo
b93ad9d Chaminda Amarasinghe    Fri Nov 15 14:46:40 2019 +0000  print read summary then print head
cb71fec Chaminda Amarasinghe    Fri Nov 15 14:37:17 2019 +0000  print head of data set
33332d1 Chaminda Amarasinghe    Thu Nov 14 10:35:16 2019 +0000  added pydoc for graph
496a23d Chaminda Amarasinghe    Wed Nov 13 16:22:12 2019 +0000  fixed too broad exception catch
70414ce Chaminda Amarasinghe    Wed Nov 13 16:00:50 2019 +0000  logging in graphs
95acb4e Chaminda Amarasinghe    Wed Nov 13 16:00:16 2019 +0000  logging in graphs
a083c64 Chaminda Amarasinghe    Tue Nov 12 14:26:10 2019 +0000  moved price visualization to ab
4509273 Chaminda Amarasinghe    Tue Nov 12 14:21:40 2019 +0000  logging and comments
09a5123 Chaminda Amarasinghe    Mon Nov 11 16:34:34 2019 +0000  analytics based on availability
6379a50 Chaminda Amarasinghe    Mon Nov 11 16:16:25 2019 +0000  main to root dir
cd0eb5b Chaminda Amarasinghe    Mon Nov 11 15:32:27 2019 +0000  final commit
70ff603 Chaminda Amarasinghe    Mon Nov 11 15:00:51 2019 +0000  unit test for math module
eb3e74e Chaminda Amarasinghe    Mon Nov 11 11:13:41 2019 +0000  added math test
f0ac1a2 Chaminda Amarasinghe    Mon Nov 11 00:49:35 2019 +0000  file read summary
cb50cb0 Chaminda Amarasinghe    Mon Nov 11 00:16:02 2019 +0000  unit tests for db
3c99581 Chaminda Amarasinghe    Sun Nov 10 11:56:51 2019 +0000  pydoc for db module
fc263c2 Chaminda Amarasinghe    Sun Nov 10 10:16:54 2019 +0000  db name as parameter for making testable code
6ae1e82 Chaminda Amarasinghe    Sat Nov 9 16:25:15 2019 +0000   moved test data to tmp folder
5726ec1 Chaminda Amarasinghe    Sat Nov 9 16:15:52 2019 +0000   unit testing for file read
59d16c4 Chaminda Amarasinghe    Sat Nov 9 14:58:51 2019 +0000   unit testing for file read
1136902 Chaminda Amarasinghe    Sat Nov 9 14:58:24 2019 +0000   custom exception for file operations
498228b Chaminda Amarasinghe    Sat Nov 9 13:22:23 2019 +0000   pydoc for file_read
ee0e078 Chaminda Amarasinghe    Fri Nov 8 17:16:08 2019 +0000   added unit test for test_summary_list_error_cases
3de02ad Chaminda Amarasinghe    Fri Nov 8 17:08:03 2019 +0000   fixed code inspection issues
8f7989a Chaminda Amarasinghe    Fri Nov 8 16:52:30 2019 +0000   added unit test and pydoc for test_summary_list
3f4c2da Chaminda Amarasinghe    Thu Nov 7 18:59:41 2019 +0000   typo
2cd128a Chaminda Amarasinghe    Thu Nov 7 18:45:34 2019 +0000   unit test and pydoc for filter_by_index
157aa1c Chaminda Amarasinghe    Thu Nov 7 17:55:42 2019 +0000   group by doc correction
b95bd28 Chaminda Amarasinghe    Thu Nov 7 17:51:42 2019 +0000   filter_by_index should be generic to accept index instead tuple
5c891e1 Chaminda Amarasinghe    Thu Nov 7 17:46:32 2019 +0000   unit tests and doc for group_count_list and two_group_count_list
dc42ecf Chaminda Amarasinghe    Mon Nov 4 17:30:05 2019 +0000   pydoc and unit test for data group_by
b8a1d3e Chaminda Amarasinghe    Mon Nov 4 17:20:02 2019 +0000   pydoc and unit test for data group_by
bd2af8a Chaminda Amarasinghe    Mon Nov 4 16:29:03 2019 +0000   pydoc and unit test for data sort_data
```

```
f64017f Chaminda Amarasinghe    Mon Nov 4 15:06:51 2019 +0000   vertical_slice_all_data
e376b2f Chaminda Amarasinghe    Mon Nov 4 13:07:33 2019 +0000   unit test and py doc for vertical_slice_data
5804f9b Chaminda Amarasinghe    Mon Nov 4 01:06:14 2019 +0000   fixed bar chart saving issue
c30911e Chaminda Amarasinghe    Mon Nov 4 00:44:16 2019 +0000   graph saving
48d137f Chaminda Amarasinghe    Sun Nov 3 23:54:40 2019 +0000   added title to all graphs
82deab6 Chaminda Amarasinghe    Sun Nov 3 22:47:09 2019 +0000   title addition to graphs
9167b27 Chaminda Amarasinghe    Sun Nov 3 22:35:14 2019 +0000   refactored visualizing price
55f9bd7 Chaminda Amarasinghe    Sun Nov 3 22:08:28 2019 +0000   rename duplicate variables
e39e362 Chaminda Amarasinghe    Sun Nov 3 21:54:58 2019 +0000   analize price greater than 1000
5e8616a Chaminda Amarasinghe    Sun Nov 3 21:51:05 2019 +0000   analize price greater than 1000
99b7f0c Chaminda Amarasinghe    Sun Nov 3 17:32:48 2019 +0000   violin plots for price group by room type and neighbourhood group
71609ee Chaminda Amarasinghe    Sun Nov 3 17:26:34 2019 +0000   violin plot support
83fc9dc Chaminda Amarasinghe    Sat Nov 2 23:38:11 2019 +0000   mean value support for dis_plot
246077a Chaminda Amarasinghe    Sat Nov 2 23:03:56 2019 +0000   violinplot plot support for dist_plot
a56ebbf Chaminda Amarasinghe    Sat Nov 2 22:20:16 2019 +0000   group bar chart neighbourhood group id and room type
e2c8326 Chaminda Amarasinghe    Sat Nov 2 22:04:51 2019 +0000   pie chart and bar chart by room type, neighbourhood group id, group bar
chart room type and neighbourhood group id
f7e0c75 Chaminda Amarasinghe    Sat Nov 2 21:33:42 2019 +0000   pie plot support
c18768b Chaminda Amarasinghe    Sat Nov 2 21:33:10 2019 +0000   group by single index should use exact key instead single element tuple
to avoid ('text',)
ddfd39f Chaminda Amarasinghe    Sat Nov 2 20:28:10 2019 +0000   group bar chart support for common graphs
99f37c5 Chaminda Amarasinghe    Sat Nov 2 20:27:04 2019 +0000   group count list and two group count list
e4e950d Chaminda Amarasinghe    Sat Nov 2 01:15:57 2019 +0000   added 2 group support
9cedbf0 Chaminda Amarasinghe    Fri Nov 1 23:05:25 2019 +0000   analytics into sub modules. ab and common
264fa04 Chaminda Amarasinghe    Fri Nov 1 22:27:40 2019 +0000   corrected group by first element
15b0a3c Chaminda Amarasinghe    Fri Nov 1 17:01:45 2019 +0000   xlable and ylable for all pairplots
0ea44df Chaminda Amarasinghe    Fri Nov 1 15:13:08 2019 +0000   multi distribution plot support, started using logging
3f2be25 Chaminda Amarasinghe    Fri Nov 1 13:40:06 2019 +0000   multi distribution plot support
b3ebf2d Chaminda Amarasinghe    Fri Nov 1 13:39:37 2019 +0000   multi distribution plot support
72dcc6a Chaminda Amarasinghe    Fri Nov 1 12:41:33 2019 +0000   filter by index, limiting data
7de21d9 Chaminda Amarasinghe    Fri Nov 1 11:02:55 2019 +0000   multi_scatter for single field
d781e5e Chaminda Amarasinghe    Thu Oct 31 18:15:13 2019 +0000  print summary in table
a77c0bf Chaminda Amarasinghe    Thu Oct 31 16:58:58 2019 +0000  count chart and seaborn support to find regression
d9d5567 Chaminda Amarasinghe    Thu Oct 31 14:45:17 2019 +0000  refactored price_data to ab_data
8c4176f Chaminda Amarasinghe    Thu Oct 31 14:13:29 2019 +0000  multi scatter plot support for any data set
42bfdae Chaminda Amarasinghe    Thu Oct 31 14:12:06 2019 +0000  vertical_slice_all_data method added
27a5d48 Chaminda Amarasinghe    Wed Oct 30 18:13:53 2019 +0000  removed unused imports
5996899 Chaminda Amarasinghe    Wed Oct 30 17:48:01 2019 +0000  reading from db
81c0466 Chaminda Amarasinghe    Tue Oct 29 17:44:27 2019 +0000  group by index for any data set
81d0340 Chaminda Amarasinghe    Tue Oct 29 17:04:15 2019 +0000  corrected mode calculation
eb5887d Chaminda Amarasinghe    Tue Oct 29 00:09:40 2019 +0000  basic math functions count, min, mean, median, mode, sd, max and summary
for fields
7443d14 Chaminda Amarasinghe    Mon Oct 28 20:35:10 2019 +0000  sort a tuple by given field index
16b6e77 Chaminda Amarasinghe    Mon Oct 28 02:29:28 2019 +0000  read csv data and convert to correct type
7ea02c5 Chaminda Amarasinghe    Mon Oct 28 00:23:23 2019 +0000  price data slicing
ced071f Chaminda Amarasinghe    Mon Oct 28 00:22:25 2019 +0000  const module
feb7c4f Chaminda Amarasinghe    Sun Oct 27 23:57:44 2019 +0000  added main_test for testing
fe82cb6 Chaminda Amarasinghe    Sun Oct 27 23:55:48 2019 +0000  added constants for field names
8f2b1d5 Chaminda Amarasinghe    Sun Oct 27 19:04:29 2019 +0000  add data to price_data table
```

Appendix 2: References

Müller, Andreas C. and Guido, Sarah. "Introduction to Machine Learning with Python". O'Reilly Media, Inc., 2016. **ISBN**: 1449369898

Barry, Paul. "Head First Python, 2nd Edition". O'Reilly Media, Inc., 2010. **ISBN**: 9781491919521

Yim, Aldrin. Chung, Claire. and Yu, Allen. "Matplotlib for Python Developers". Packt Publishing Ltd. 2018. **ISBN**: 9781788625173

Boschetti, Alberto. and Massaron, Luca. "Python Data Science Essentials" Packt Publishing Ltd. 2015. **ISBN**: 9781785280429

Semi, Koen. "5 Key Principles of Software Architecture" < https://towardsdatascience.com/5-key-principles-of-software-architecture-e5379cb10fd5>. November 1st, 2019

Python Software Foundation. "Python Documentation" < http://docs.python.org/3/>. November 1st, 2019

NumPy developers. "NumPy Documentation" < https://numpy.org/doc/>. November 10st 2019

the Matplotlib development team. "Matplotlib Documentation" <https://matplotlib.org/3.1.1/contents.html>. November 15th 2019

Waskom, Michael. "Seaborn Documentation" < https://seaborn.pydata.org/>. November 15th 2019