0



**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 3](#_Toc26683613)

[Introduction 5](#_Toc26683614)

[Data 6](#_Toc26683615)

[Data Fields 7](#_Toc26683616)

[Sample Data 8](#_Toc26683617)

[Design 9](#_Toc26683618)

[Design Principles. 9](#_Toc26683619)

[Modules 9](#_Toc26683620)

[Database Table Diagram 12](#_Toc26683621)

[Sequence Diagrams 13](#_Toc26683622)

[Testing 20](#_Toc26683623)

[Analysis and Visualisation 22](#_Toc26683624)

[Conclusion 38](#_Toc26683625)

[Appendix 1: Reflective Learning Log 42](#_Toc26683626)

[Appendix 2: References 68](#_Toc26683627)

# Figures

[Figure 1 Sample Data. Spreadsheet view 8](#_Toc26683628)

[Figure 2 Sample Data Application Output 8](#_Toc26683629)

[Figure 3 Module Diagram 9](#_Toc26683630)

[Figure 4 Database Table Diagram 12](#_Toc26683631)

[Figure 5 Sequence Diagram File Read to Pair Scatter Plot 13](#_Toc26683632)

[Figure 6 Sequence Diagram Selecting Price Range 2 14](#_Toc26683633)

[Figure 7 Sequence Diagram Analysing data Price > 500 15](#_Toc26683634)

[Figure 8 Sequence Diagram Analysing data Price > 1000 17](#_Toc26683635)

[Figure 9 Sequence Diagram Analysing data Price > 5000 18](#_Toc26683636)

[Figure 10 Sequence Diagram Availability = 0, Availability > 19](#_Toc26683637)

[Figure 11 Unit Test Coverage 20](#_Toc26683638)

[Figure 12 Running all unittests 21](#_Toc26683639)

[Figure 13 Summary of file read 22](#_Toc26683640)

[Figure 14 Data Head 22](#_Toc26683641)

[Figure 15 Data Summary 23](#_Toc26683642)

[Figure 16 Pair Scatter Plots 24](#_Toc26683643)

[Figure 17 Pair Scatter Plots with Correlations 25](#_Toc26683644)

[Figure 18 Price Summary 25](#_Toc26683645)

[Figure 19 Price Distribution - All Data 26](#_Toc26683646)

[Figure 20 Price Summary. Price <=1000 26](#_Toc26683647)

[Figure 21 Price Distribution. Price <= 1000 27](#_Toc26683648)

[Figure 22 Price Summary. Price <=500 27](#_Toc26683649)

[Figure 23 Price Distribution. Price <= 500 28](#_Toc26683650)

[Figure 24 Price Distribution comparison 28](#_Toc26683651)

[Figure 25 Room Listing by Room Type, Pie Chart (Price < 500) 29](#_Toc26683652)

[Figure 26 Room Listing by Room Type, Count Bar Chart (Price < 500) 29](#_Toc26683653)

[Figure 27 Room Listing by Neighbourhood Group, Pie Chart (Price < 500) 30](#_Toc26683654)

[Figure 28 Room Listing by Neighbourhood Group, Count Bar Chart (Price < 500) 30](#_Toc26683655)

[Figure 29 Room listing by Room Type and Neighbourhood Group (Price <=$ 500) 31](#_Toc26683656)

[Figure 30 Room listing by Neighbourhood Group and Room Type (Price <=$ 500) 31](#_Toc26683657)

[Figure 31 Price KDE by Room Type (Price <= $500) 32](#_Toc26683658)

[Figure 32 Price KDE by Neighbourhood Group (Price <= $500) 32](#_Toc26683659)

[Figure 33 Price Summary. Price > 1000, Price > 5000 33](#_Toc26683660)

[Figure 34. Price Distribution. Price > $1000, Price > $5000 33](#_Toc26683661)

[Figure 35 Price Comparison. All Price, Price <$500, Price > $1000, Price > $5000 33](#_Toc26683662)

[Figure 36 Price Grouping 34](#_Toc26683663)

[Figure 37 Price Grouping 35](#_Toc26683664)

[Figure 38 Price Comparison for Availability 35](#_Toc26683665)

[Figure 39 Setting up git repository 42](#_Toc26683666)

[Figure 40 Facing Circular import issue 42](#_Toc26683667)

[Figure 41 Fixing Module not found issue 43](#_Toc26683668)

[Figure 42 Fixing Encoding issue while reading file 44](#_Toc26683669)

[Figure 43 VC++ dependency issue while installation matplotlib 45](#_Toc26683670)

[Figure 44 Installing VS Build tool as a fix 45](#_Toc26683671)

[Figure 45 Python 3.8 is not supported but matplotlib 46](#_Toc26683672)

[Figure 46 Searched and found solution for matplotlib version issue 46](#_Toc26683673)

[Figure 47 Testing major math functions 47](#_Toc26683674)

[Figure 48 Fixing overlapped x, y labels 48](#_Toc26683675)

[Figure 49 Faced overlapped sub plots 49](#_Toc26683676)

[Figure 50 Fixing overlapped subplots by introducing figsize 50](#_Toc26683677)

[Figure 51 Facing empty subplot for variable to same variable scatter plot 50](#_Toc26683678)

[Figure 52 Decided to fill empty subplot by a distribution graph 51](#_Toc26683679)

[Figure 53 Printing summary in a table 52](#_Toc26683680)

[Figure 54 Understanding Standard Deviation 53](#_Toc26683681)

[Figure 55 Sub plotting without index causing issues 55](#_Toc26683682)

[Figure 56 Testing grouped bar charts 56](#_Toc26683683)

[Figure 57 Testing pie charts 56](#_Toc26683684)

[Figure 58 Image saving issue 57](#_Toc26683685)

[Figure 59 Fixing image save 58](#_Toc26683686)

[Figure 60 Unittesting and test coverage 59](#_Toc26683687)

[Figure 61 Unit testing for exceptions 60](#_Toc26683688)

[Figure 62 Python doc Google Standard 60](#_Toc26683689)

[Figure 63 Testing Python doc. See Also links are not working 61](#_Toc26683690)

[Figure 64 Fixing Static code analysis warning and errors 62](#_Toc26683691)

[Figure 65 Finding a way to hide internal methods 62](#_Toc26683692)

[Figure 66 Unit tests are failing intermittently because of output order 63](#_Toc26683693)

[Figure 67 File reading unit test are failing based on the location of test execution. 64](#_Toc26683694)

[Figure 68 Code refactoring 64](#_Toc26683695)

[Figure 69: Large table print casing alignment issues 65](#_Toc26683696)

[Figure 70: Fixed table size issue 65](#_Toc26683697)

[Figure 71: Introducing user friendly logging 66](#_Toc26683698)

# Introduction

According to [[1]](#footnote-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

# Data

[[2]](#footnote-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](http://data.insideairbnb.com/united-states/ny/new-york-city/2019-09-12/data/listings.csv.gz) | Detailed Listings data for New York City |
| 12 September, 2019 | New York City | [calendar.csv.gz](http://data.insideairbnb.com/united-states/ny/new-york-city/2019-09-12/data/calendar.csv.gz) | Detailed Calendar Data for listings in New York City |
| 12 September, 2019 | New York City | [reviews.csv.gz](http://data.insideairbnb.com/united-states/ny/new-york-city/2019-09-12/data/reviews.csv.gz) | Detailed Review Data for listings in New York City |
| 12 September, 2019 | New York City | [listings.csv](http://data.insideairbnb.com/united-states/ny/new-york-city/2019-09-12/visualisations/listings.csv) | Summary information and metrics for listings in New York City (good for visualisations). |
| 12 September, 2019 | New York City | [reviews.csv](http://data.insideairbnb.com/united-states/ny/new-york-city/2019-09-12/visualisations/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](http://data.insideairbnb.com/united-states/ny/new-york-city/2019-09-12/visualisations/neighbourhoods.csv) | Neighbourhood list for geo filter. Sourced from city or open source GIS files. |
| N/A | New York City | [neighbourhoods.geojson](http://data.insideairbnb.com/united-states/ny/new-york-city/2019-09-12/visualisations/neighbourhoods.geojson) | GeoJSON file of neighbourhoods of the city. |

Further, various data exploratory efforts related to this dataset can be found in kaggle

[[3]](#footnote-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 | Consi  dered |
| 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 |

## Sample Data

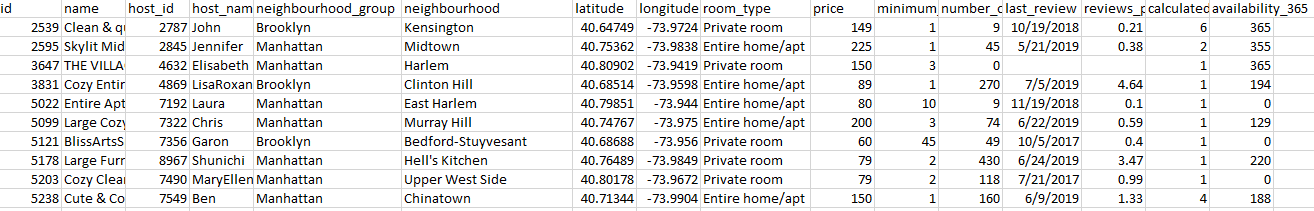


Figure Sample Data. Spreadsheet view

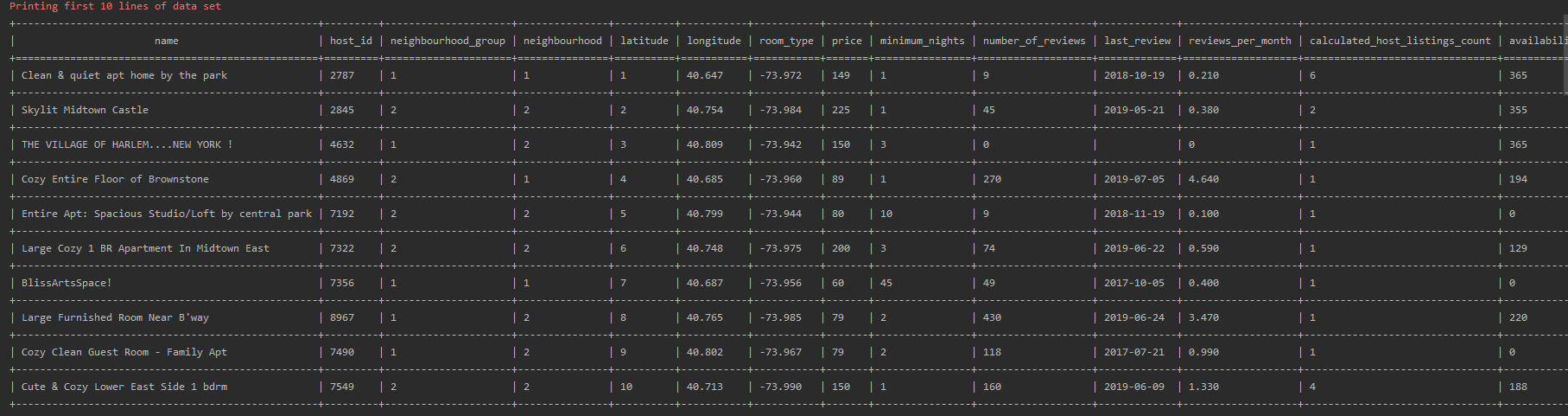


Figure Sample Data Application Output

# Design

## Design Principles.

Following design principles considered when designing modules

Modularity

Modules should be independent from each other as much as possible

[[4]](#footnote-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.

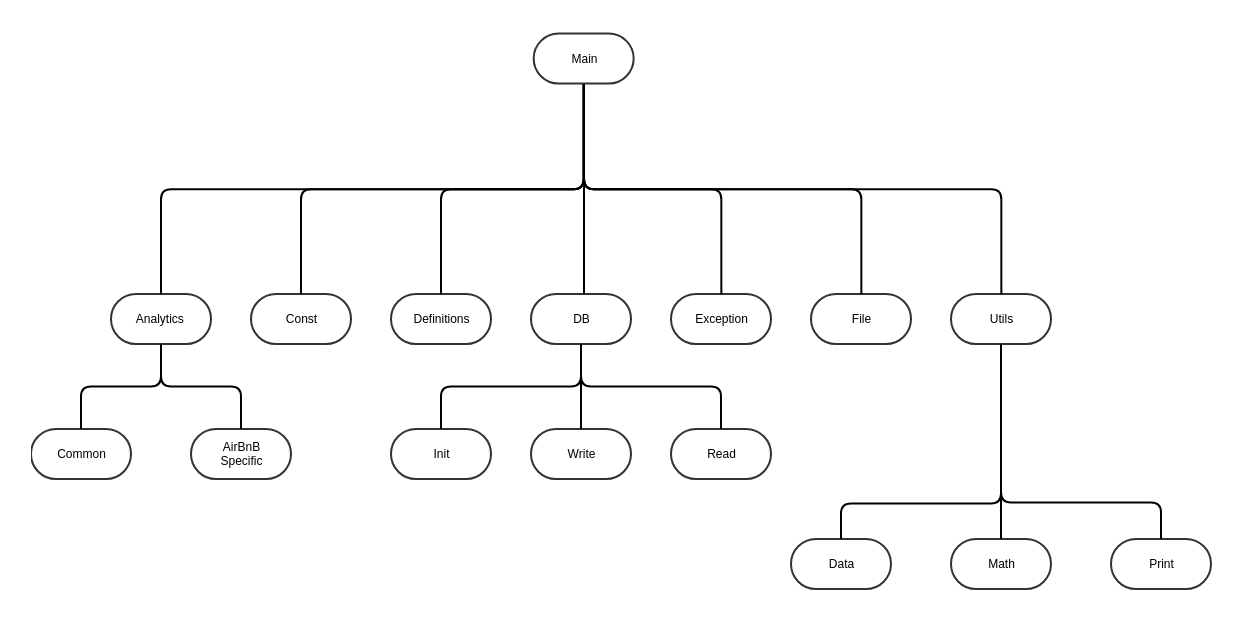


Figure Module Diagram

**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

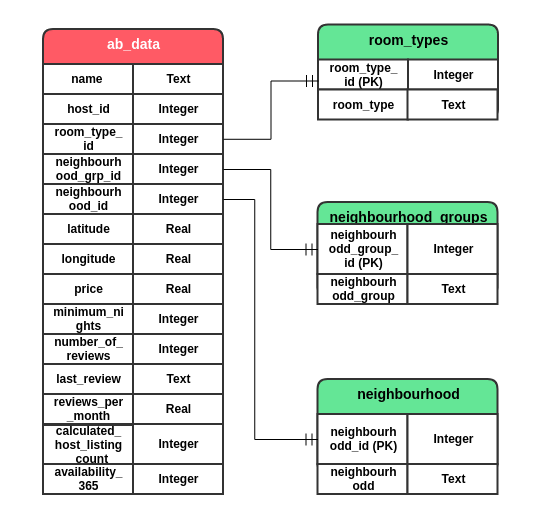


Figure Database Table Diagram

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

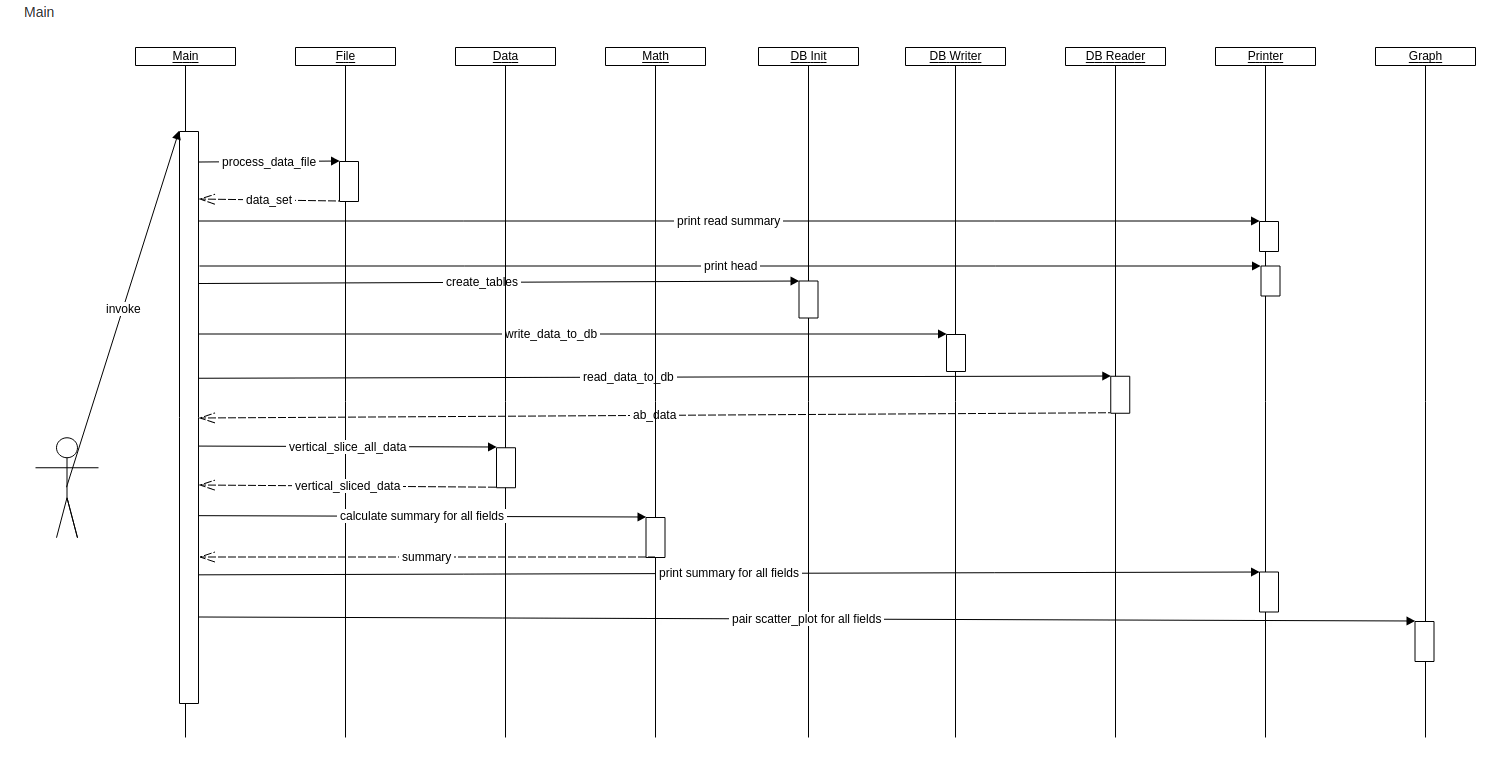


Figure Sequence Diagram File Read to Pair Scatter Plot

Comparing Price

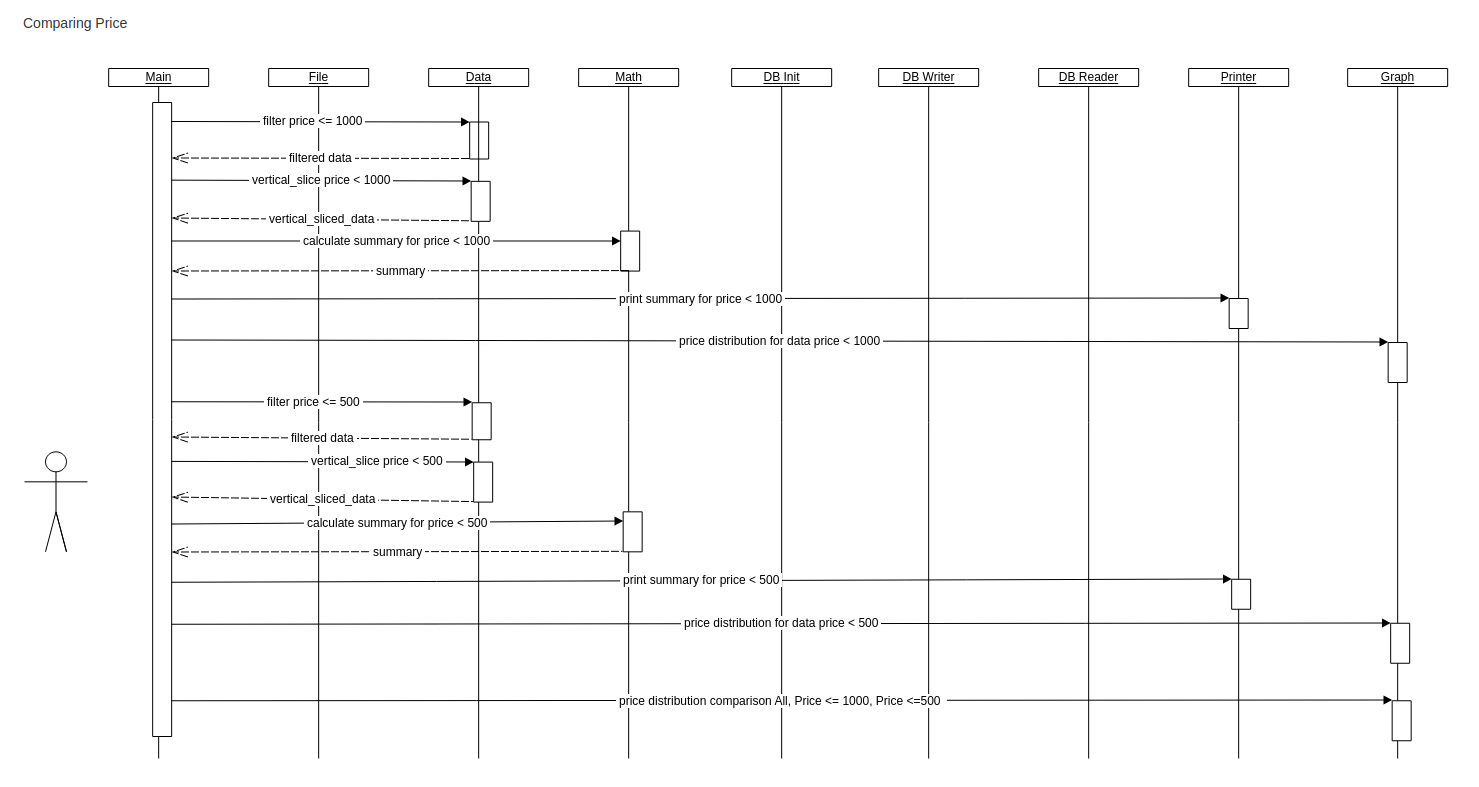


Figure Sequence Diagram Selecting Price Range 2

Analysing data Price < 500

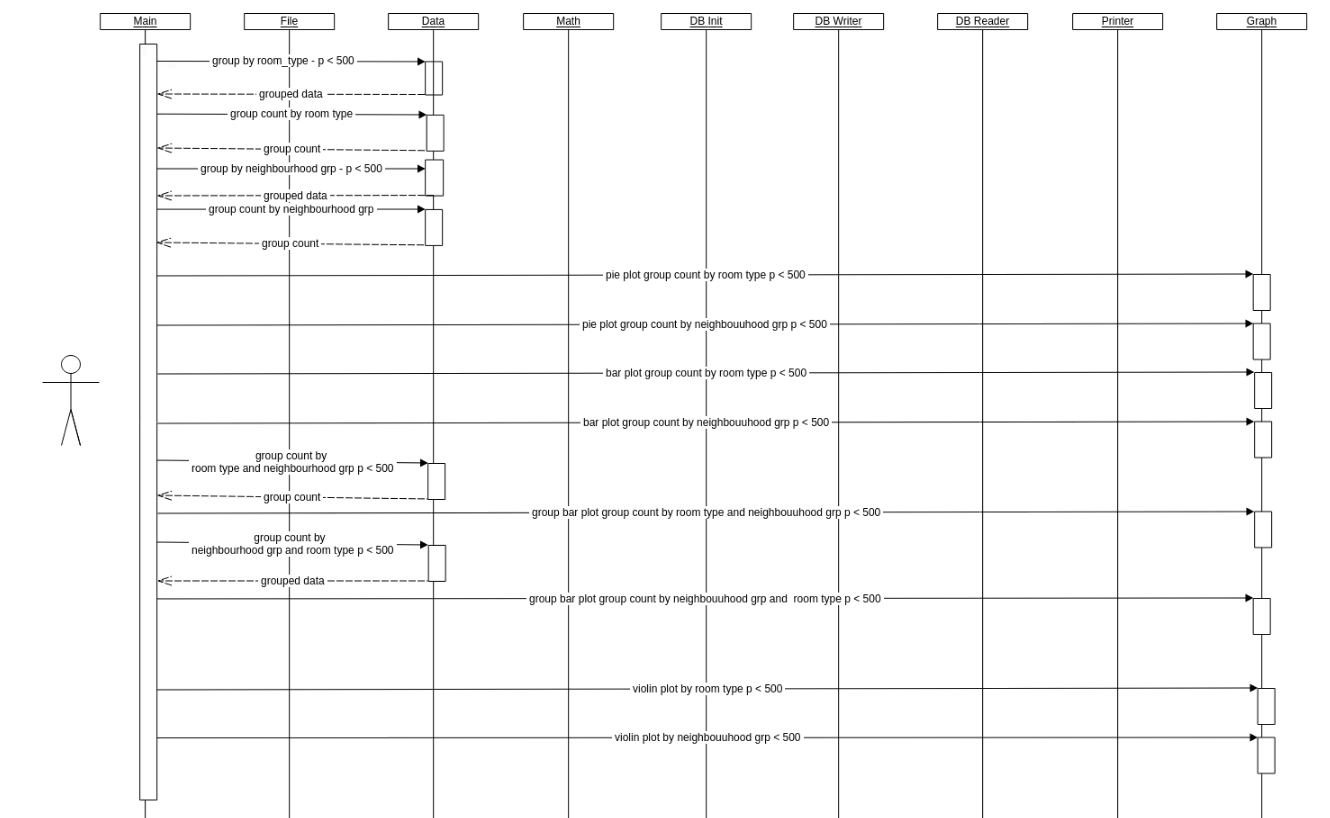


Figure Sequence Diagram Analysing data Price > 500

Analysing data Price > 1000

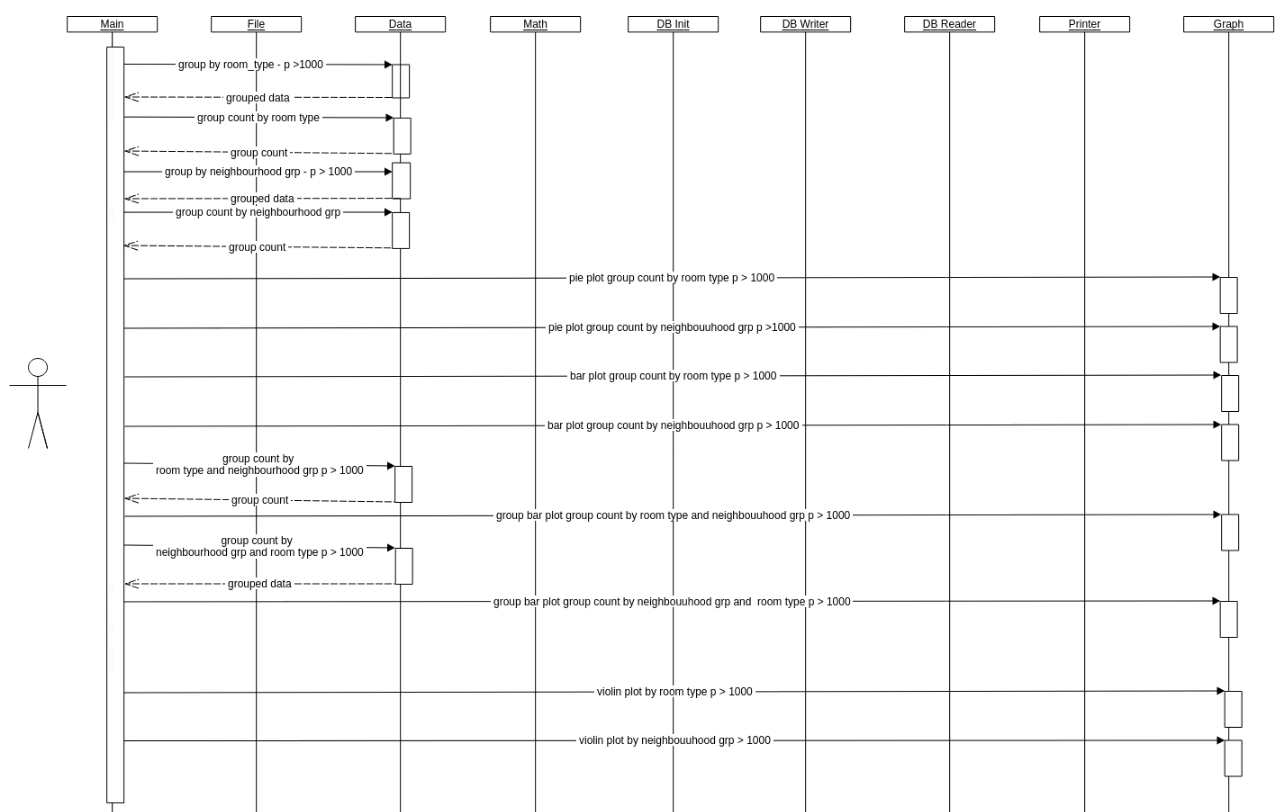


Figure Sequence Diagram Analysing data Price > 1000

Analysing data Price > 5000

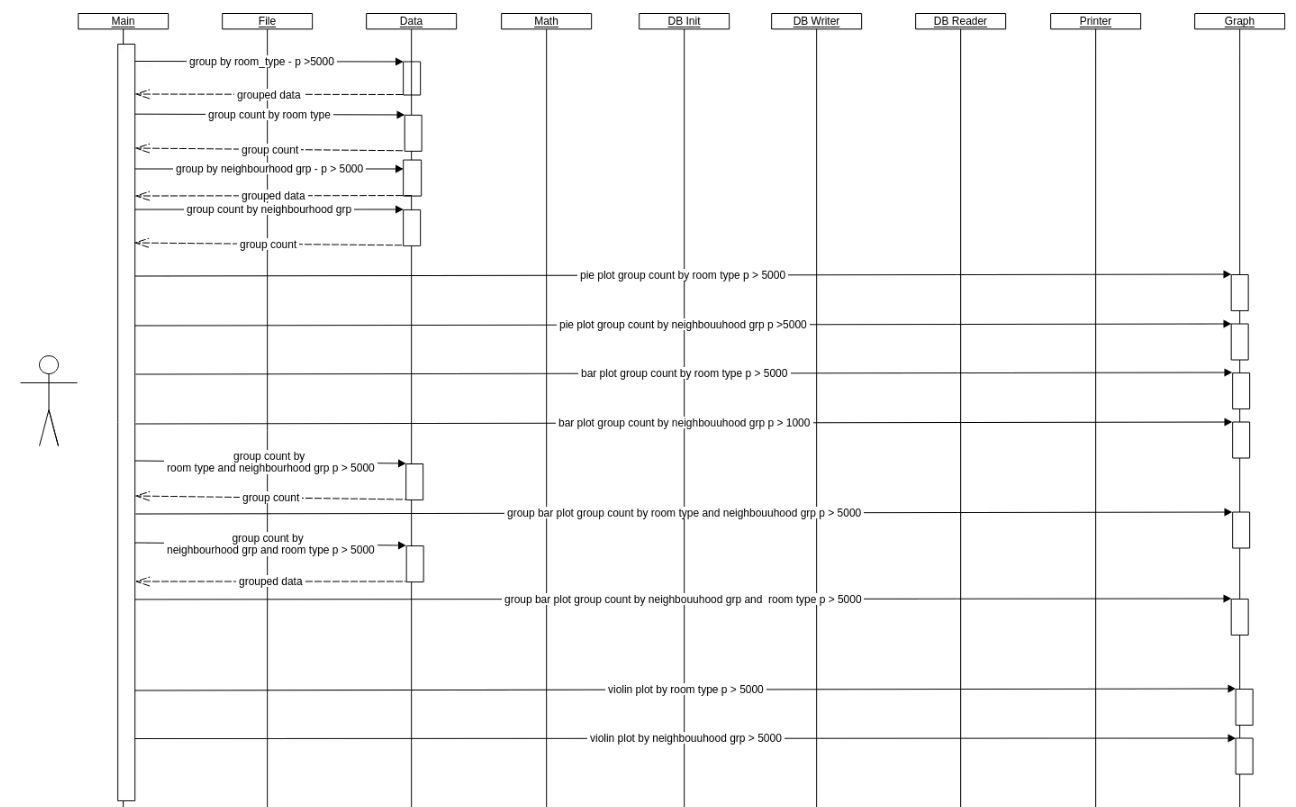


Figure Sequence Diagram Analysing data Price > 5000

Availability = 0, Availability > 0

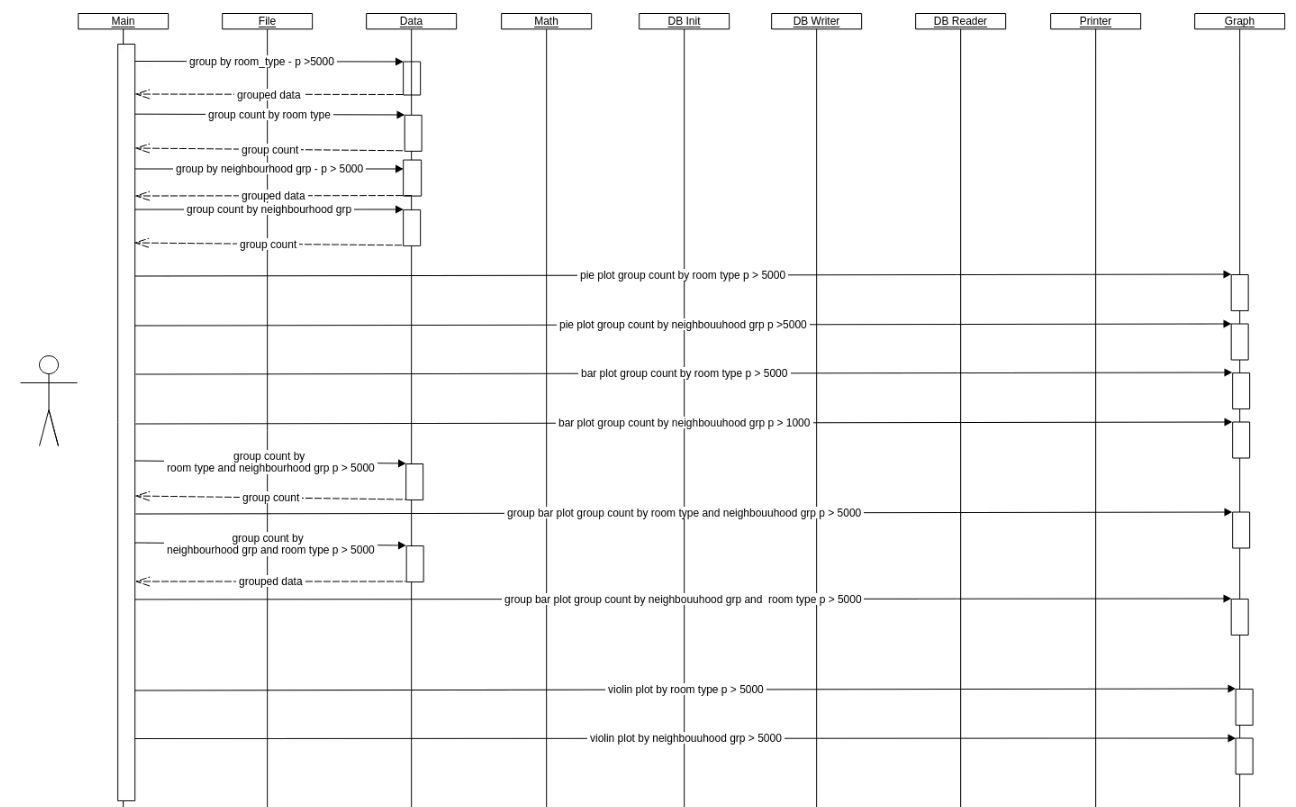


Figure Sequence Diagram Availability = 0, Availability >

# Testing

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



Figure Unit Test Coverage

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

python -m unittest -v

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

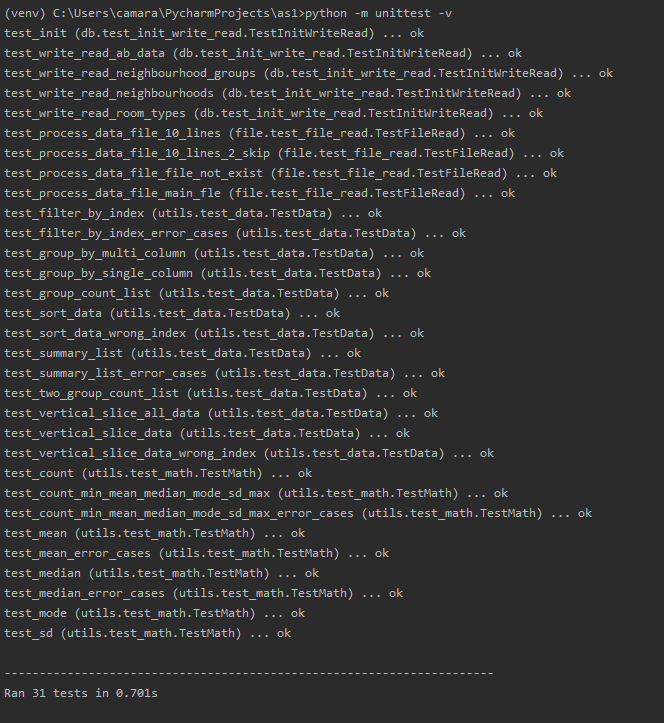
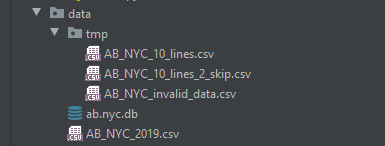


Figure Running all unittests

## Test Data.

Test Data can be found on data/tmp folder



# 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

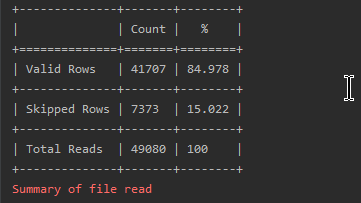


Figure Summary of file read

1. Data Head

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

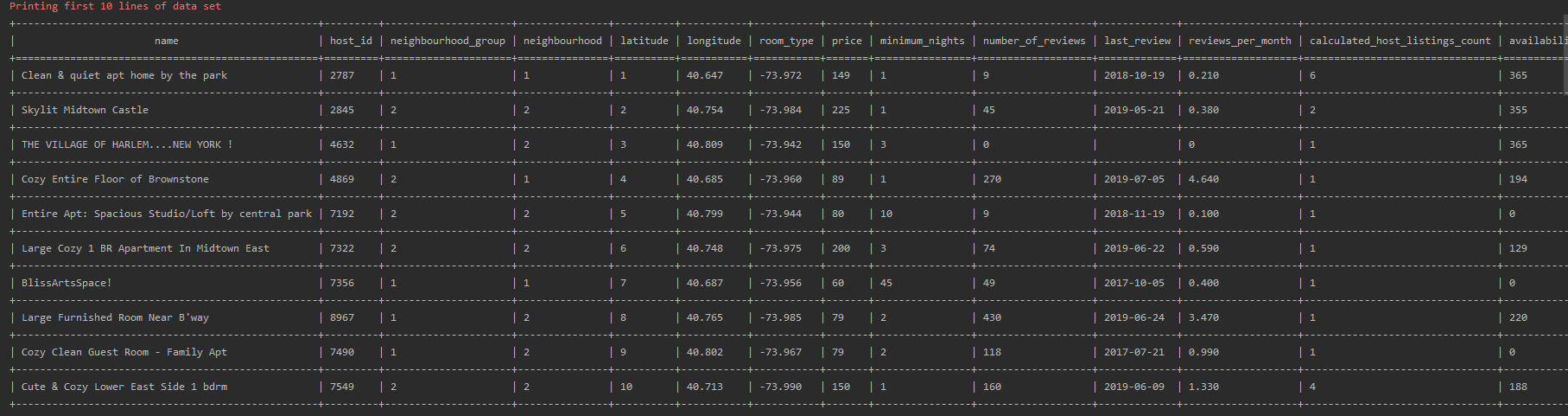


Figure Data Head

1. Data Summary

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

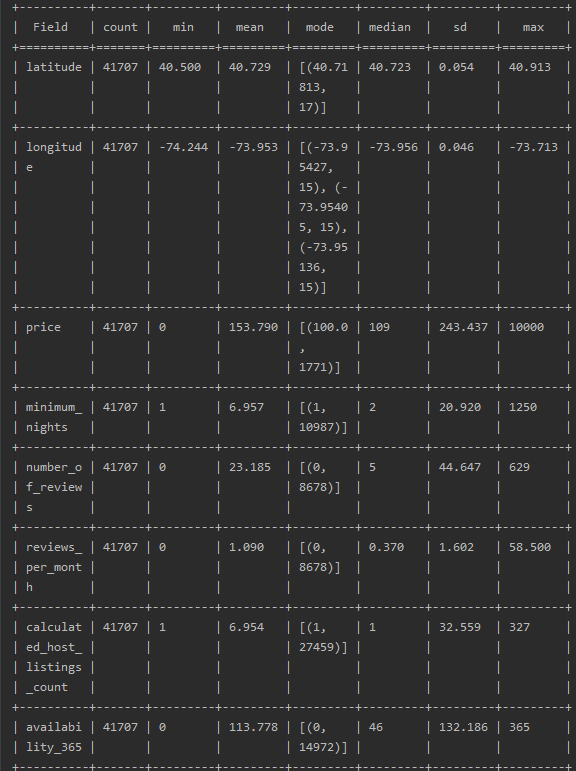


Figure Data Summary

1. 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.

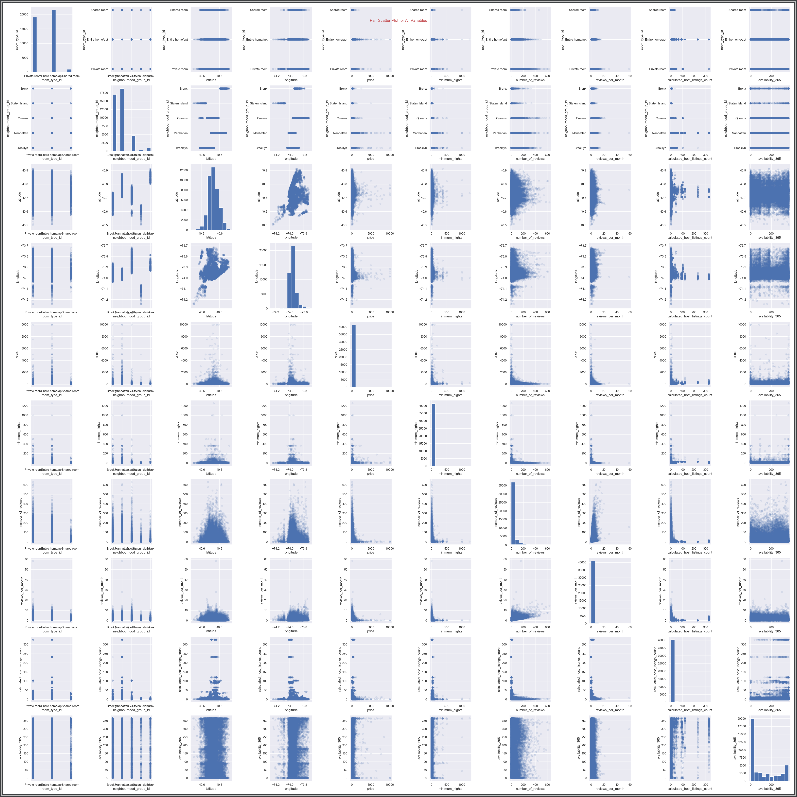


Figure Pair Scatter Plots



Figure Pair Scatter Plots with Correlations

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

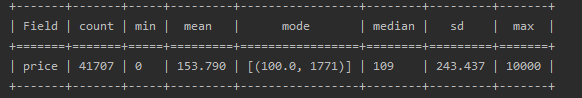


Figure Price Summary

1. Price Distribution graph for all data



Figure Price Distribution - All Data

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

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

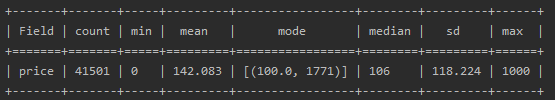


Figure Price Summary. Price <=1000

Standard deviation has been reduced to 118.224

1. Price Distribution for Price <= 1000



Figure Price Distribution. Price <= 1000

1. Considering listings where price <= 500

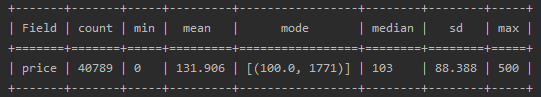


Figure Price Summary. Price <=500

1. Price Distribution for Price <= 500



Figure Price Distribution. Price <= 500

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

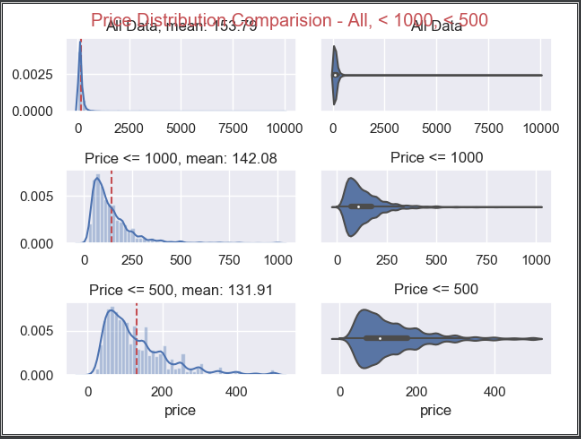


Figure Price Distribution comparison

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

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

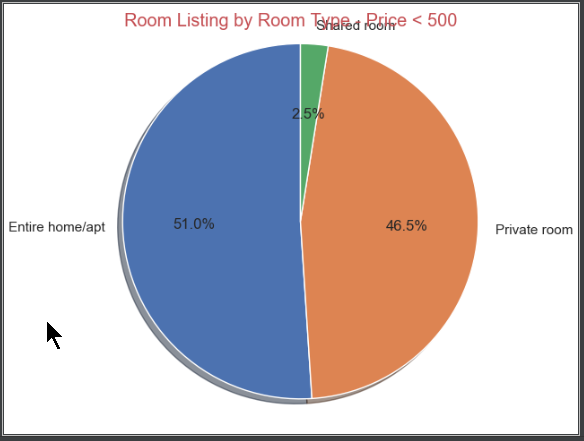


Figure Room Listing by Room Type, Pie Chart (Price < 500)



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

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

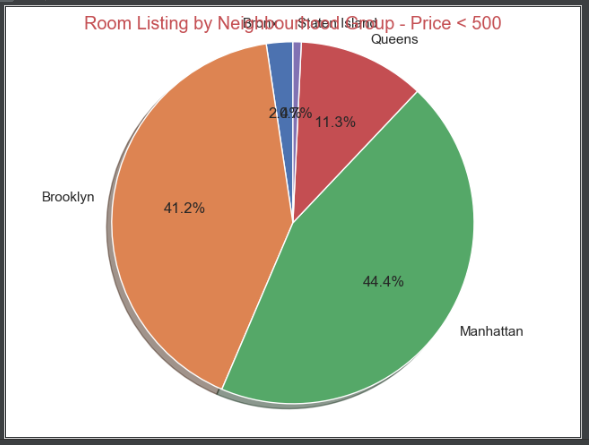


Figure Room Listing by Neighbourhood Group, Pie Chart (Price < 500)

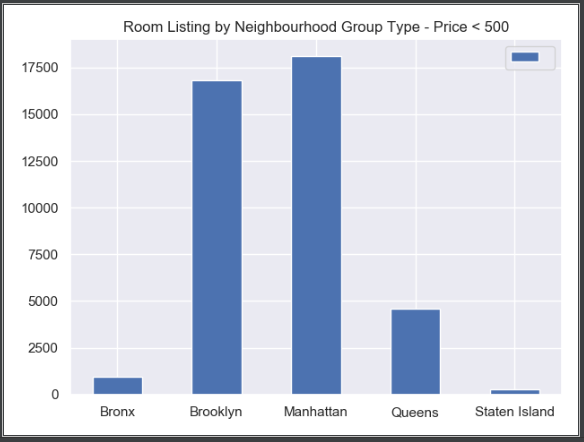


Figure Room Listing by Neighbourhood Group, Count Bar Chart (Price < 500)

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



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

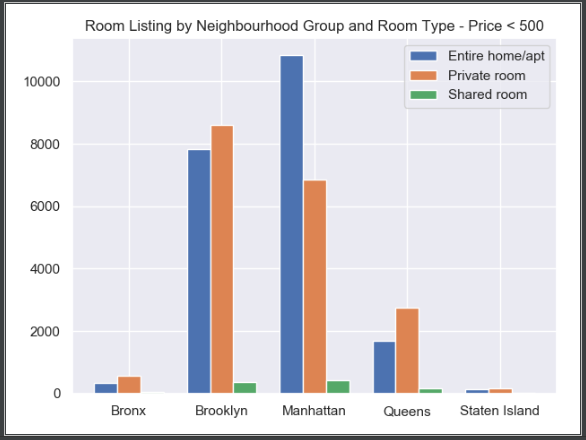


Figure Room listing by Neighbourhood Group and Room Type (Price <=$ 500)

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

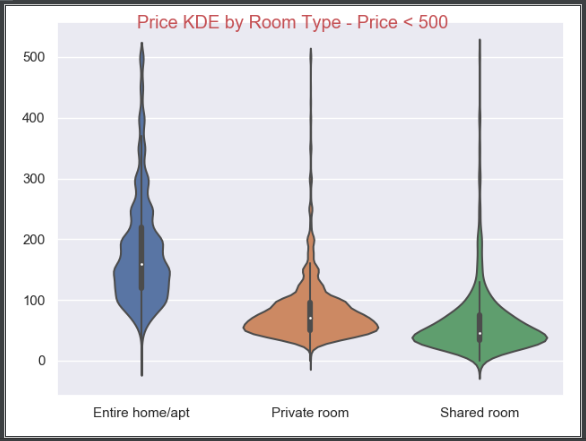


Figure Price KDE by Room Type (Price <= $500)

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

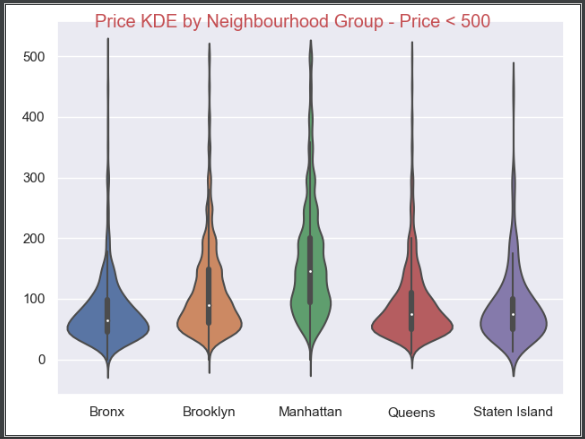


Figure Price KDE by Neighbourhood Group (Price <= $500)

1. Price Analysis. Price > 1000 and Price > 5000

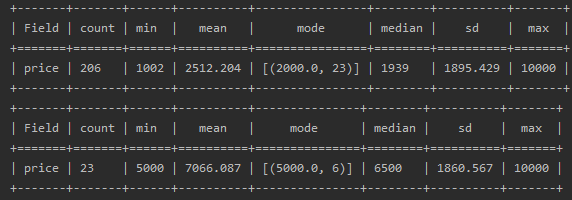


Figure Price Summary. Price > 1000, Price > 5000



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

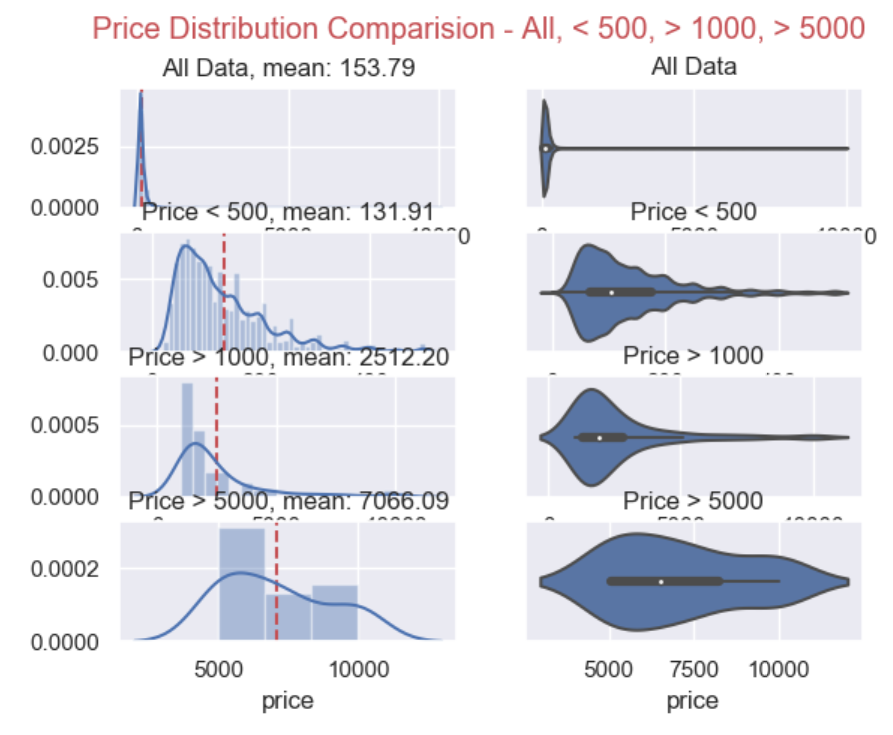
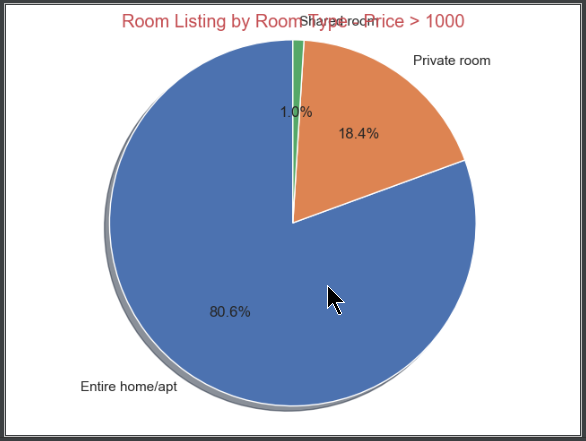
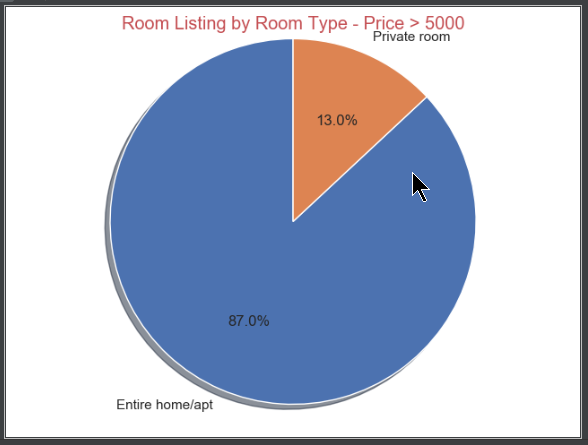
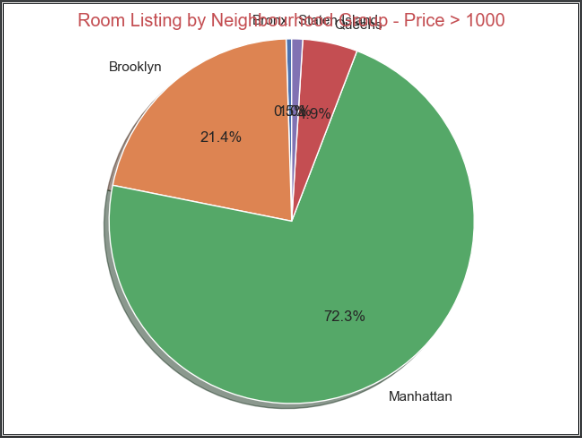
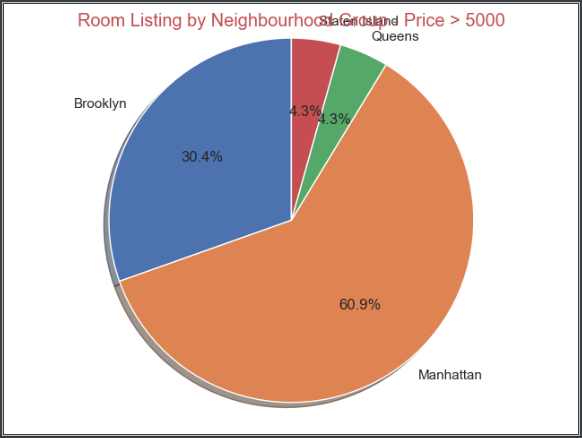


Figure Price Comparison. All Price, Price <$500, Price > $1000, Price > $5000

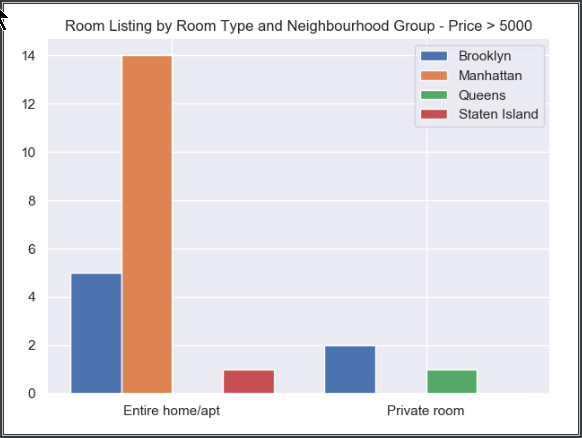
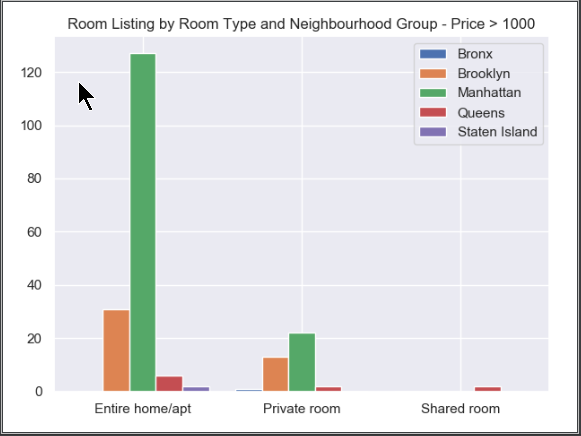


Figure Price Grouping

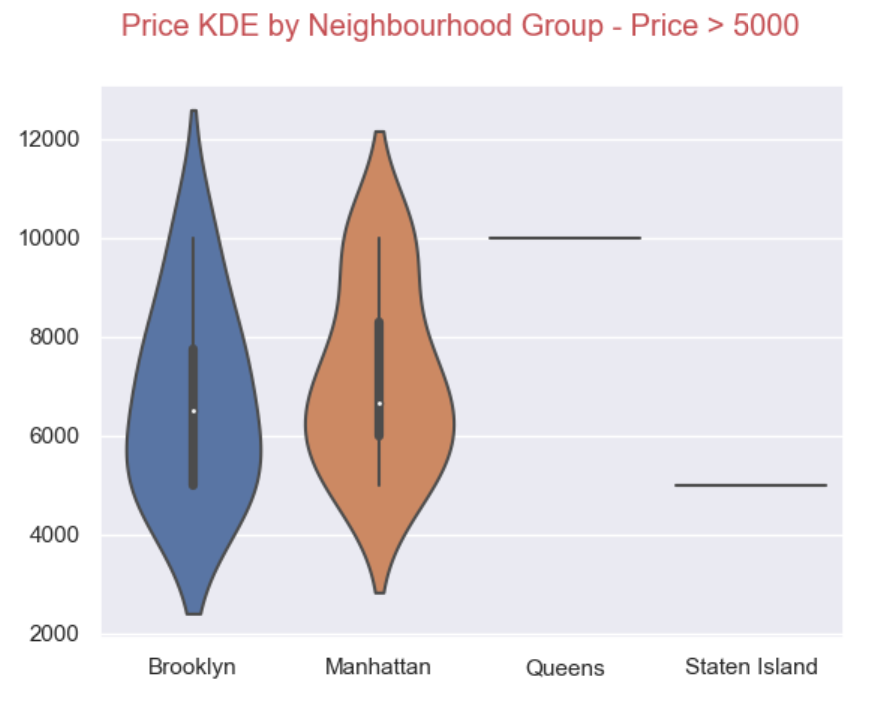
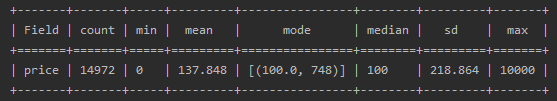


Figure Price Grouping

1. Price Comparison Availability 0 vs Availability > 0



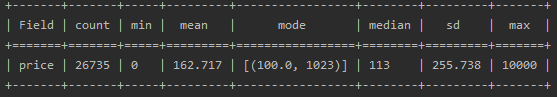
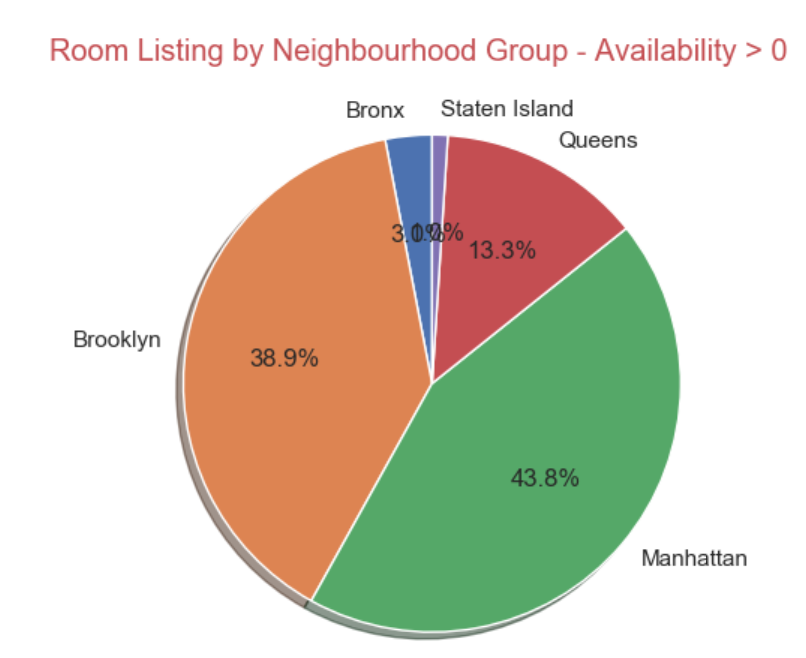
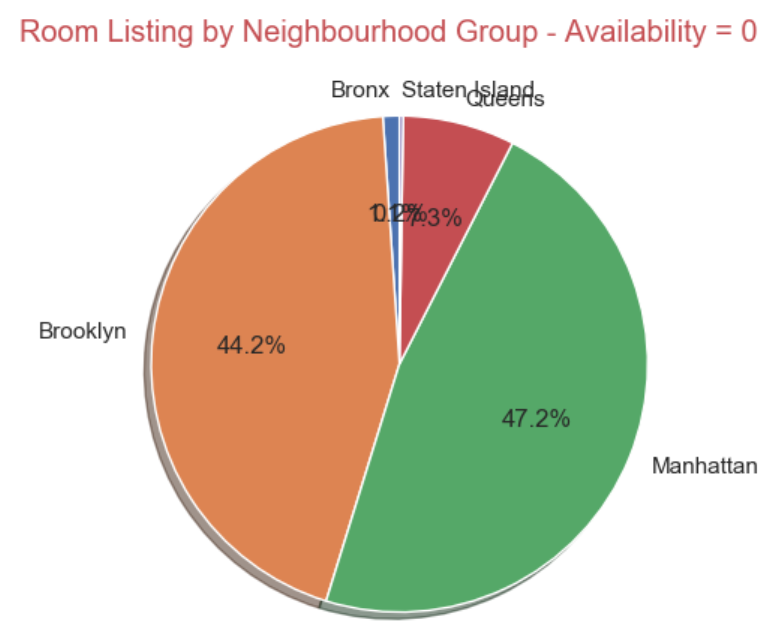
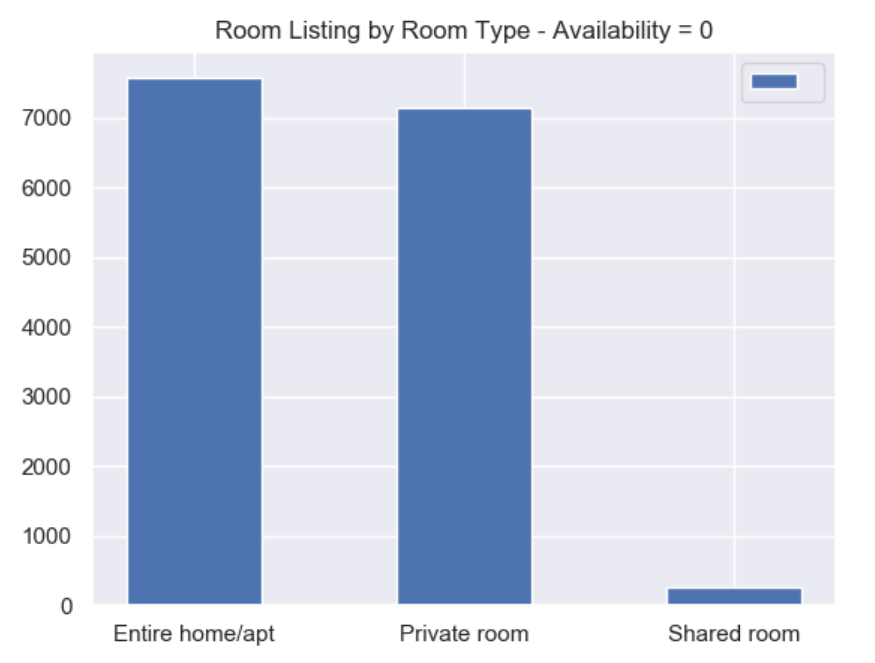
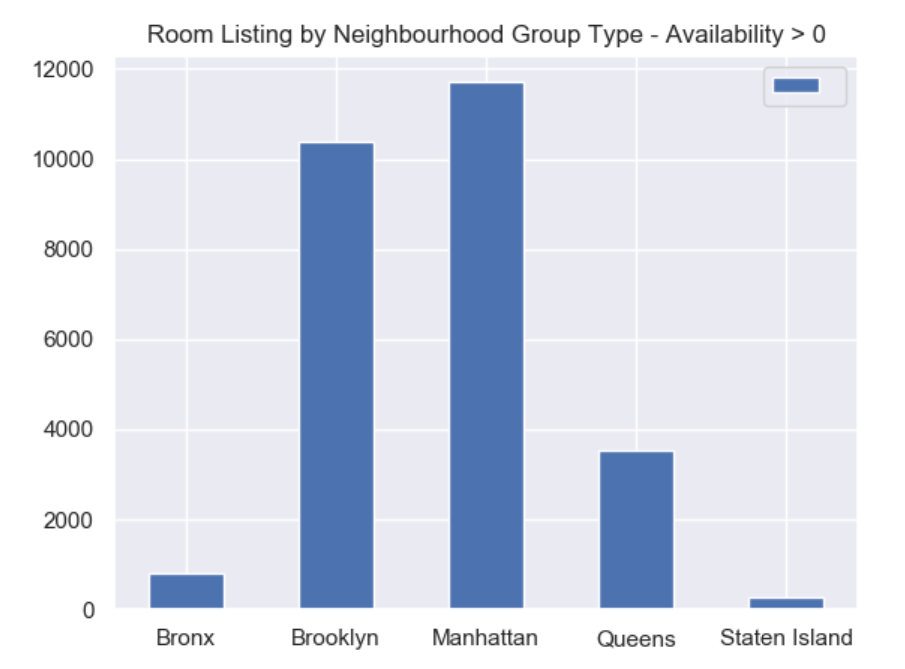
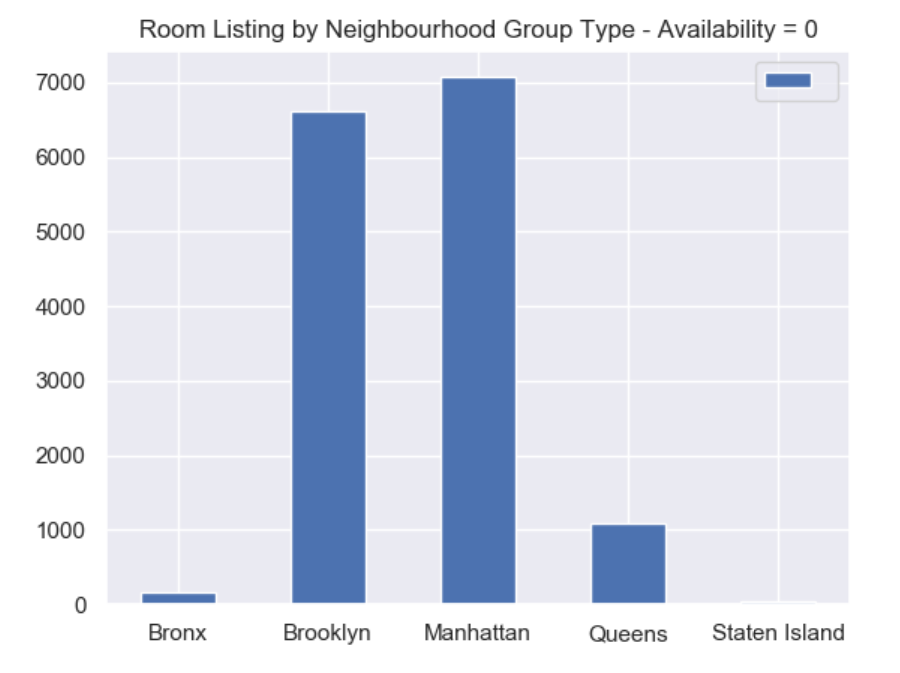
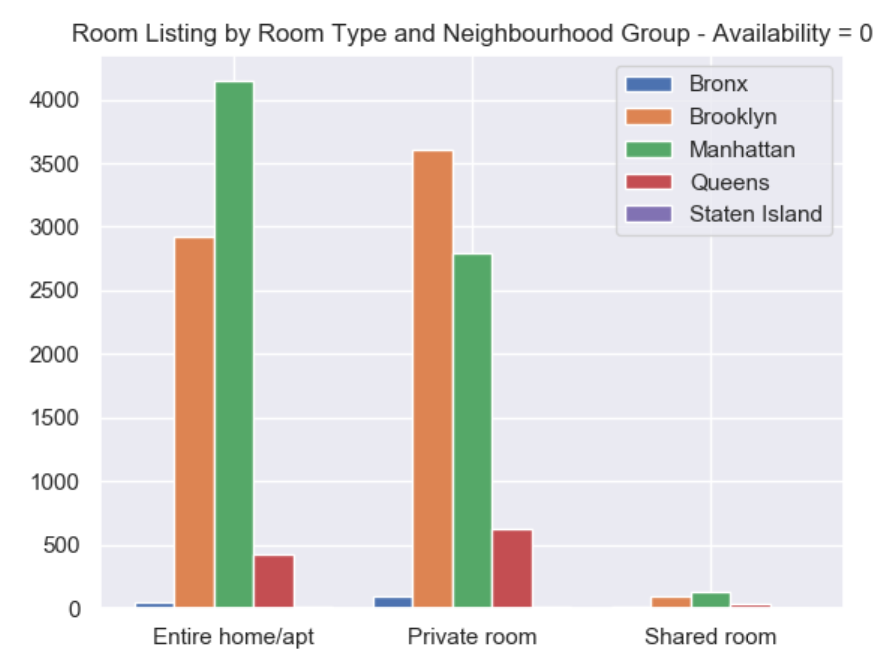


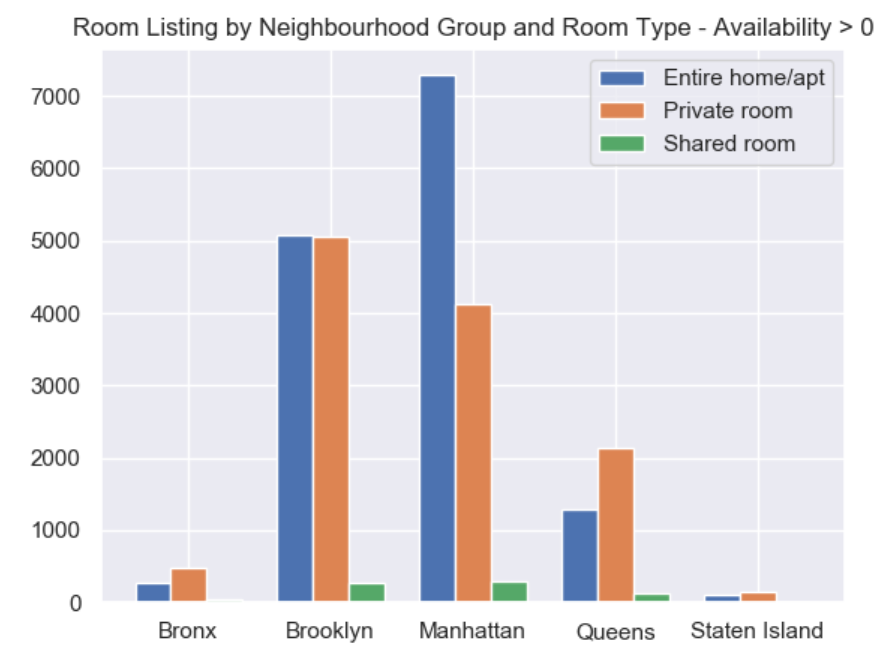
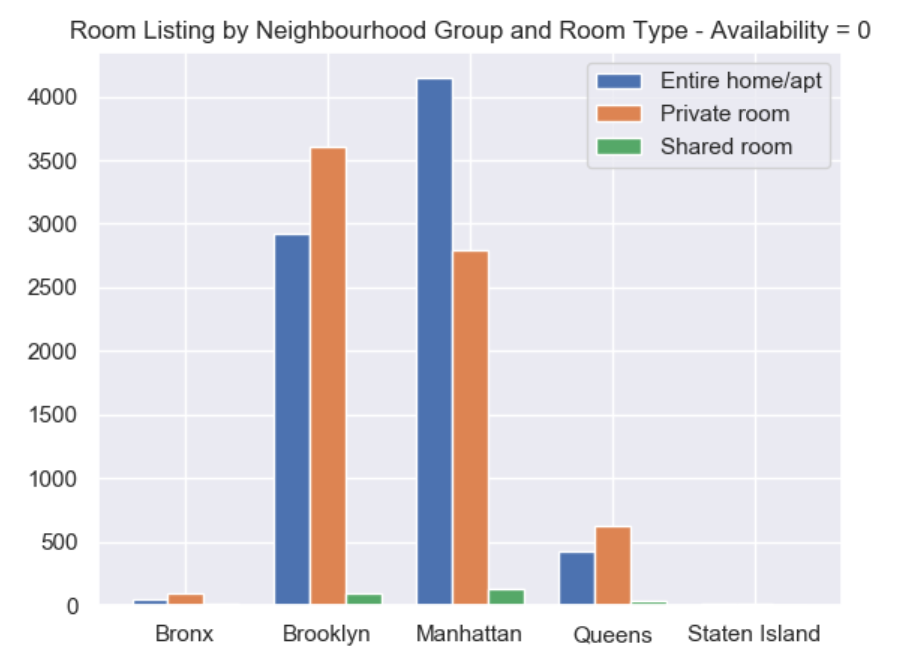
Figure Price Comparison for Availability

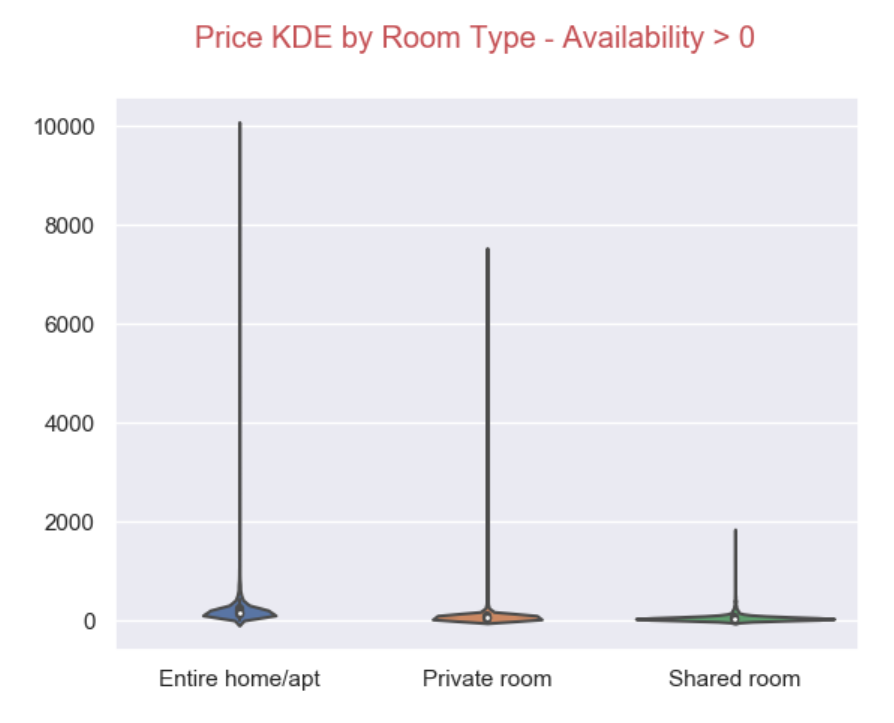








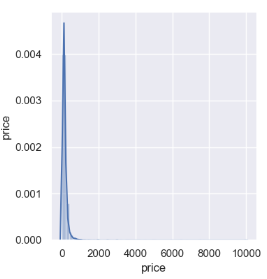




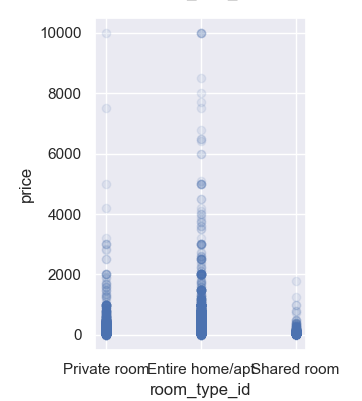


# Conclusion

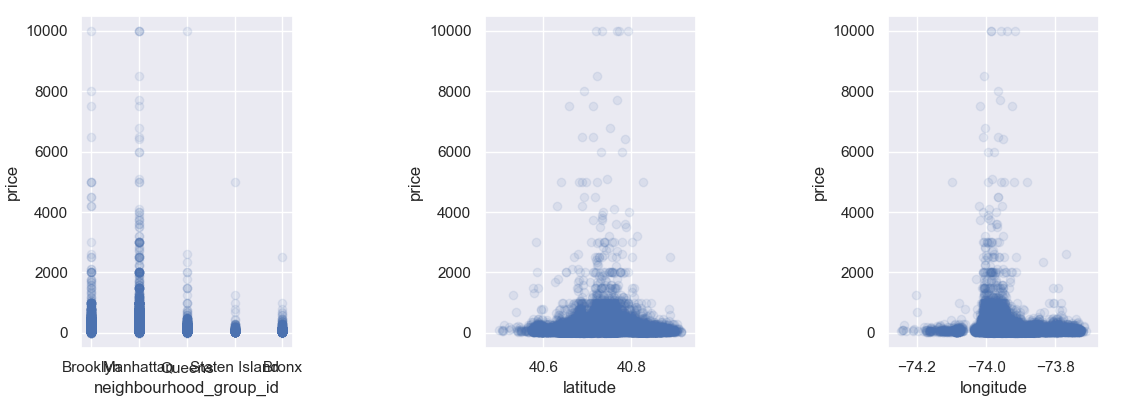
1. Most of the price listing are lower than $2000



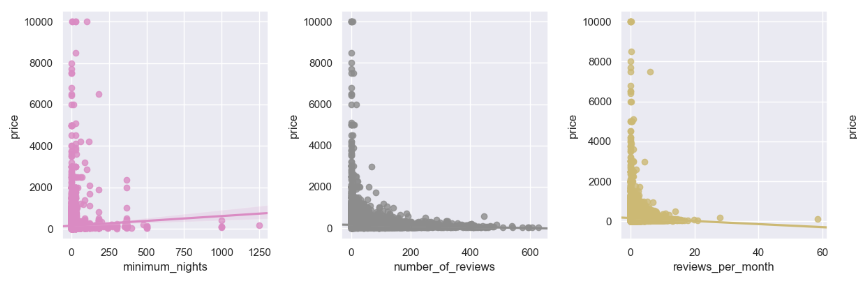
1. Room type has a direct relationship to the price. Entire Home / apt has highest price and always Shared rooms price is lower than $2000.



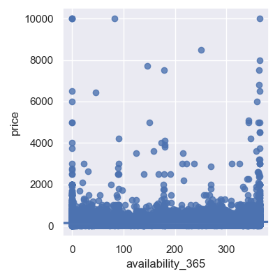
1. The location of the listing has a strong relation to the price. Manhattan and Brooklyn properties have been listed with a higher price.

****

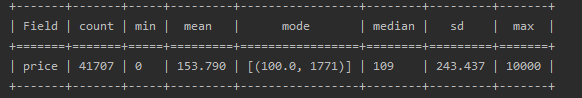
1. Price has a positive correlation with minimum number of nights, while it has a negative correlation with Number of reviews and Number of reviews per month. The reason could be low price attracts higher number of tenants.

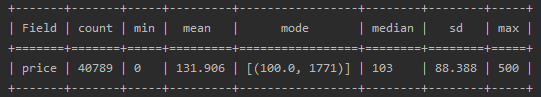


1. Below graph tries to find the relationship between Price and Availability. There is no clear relationship between Price and Availability.

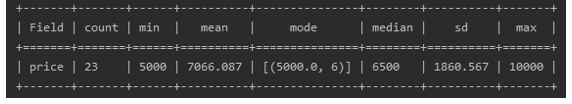


1. More than 97% listing’s price is lower than $500

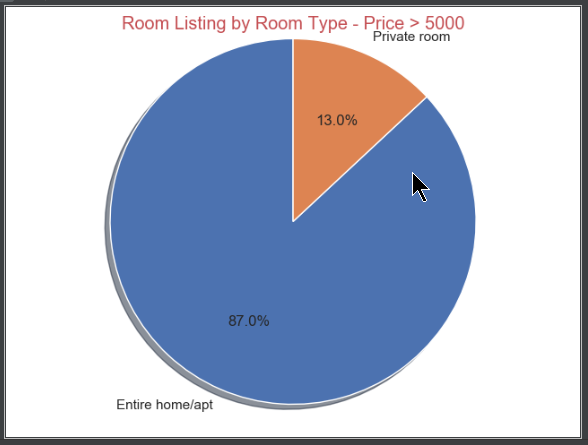
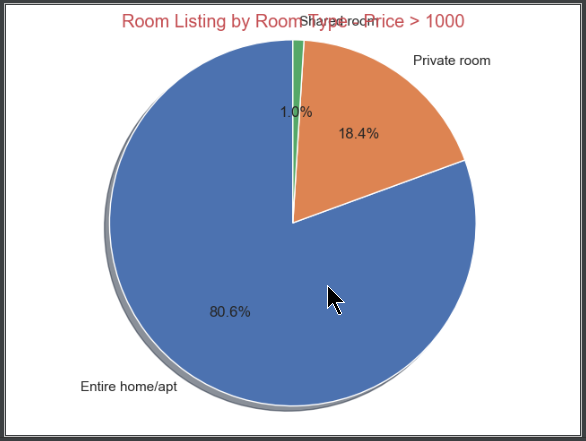
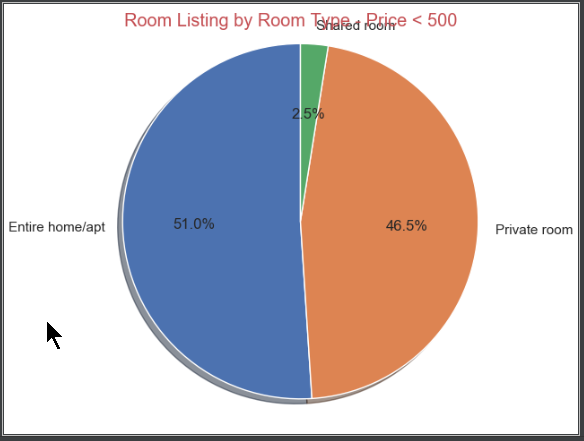


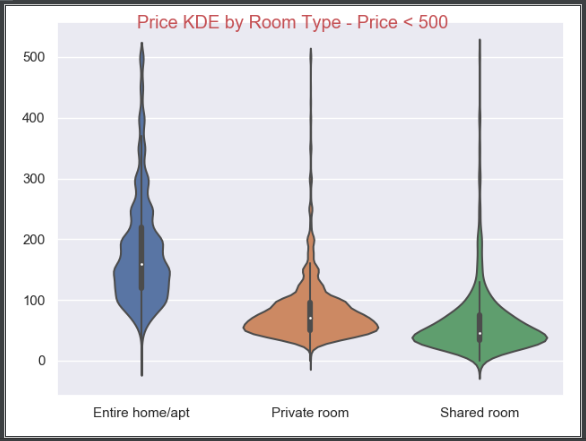


1. Only 0.00006% (23 out of 41707) listings are priced over $5000

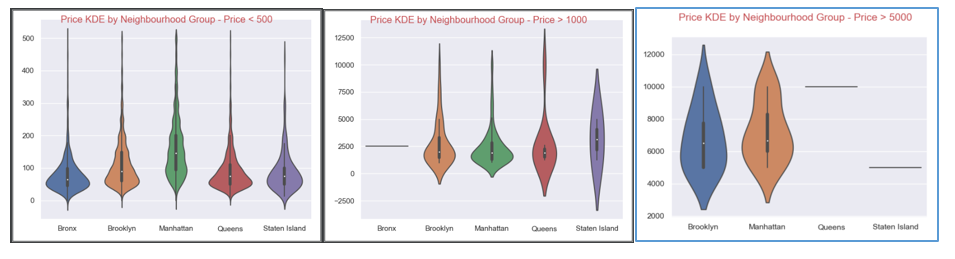


1. Entire Home / apt are popular in all price categories.
2. But when price goes up Entire Home / apt are more popular.
3. No any Shared rooms which are priced over $5000



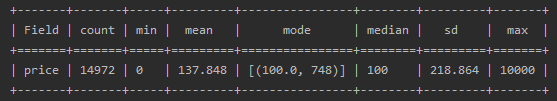


1. Manhattan has the highest median for Price in all 3 price categories, followed by Brooklyn
2. Manhattan has the highest listing count followed by Brooklyn
3. In $500 price category, Private rooms are much popular in all neighbourhood groups, Except Manhattan where Entire Home / Apt has highest listing.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





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

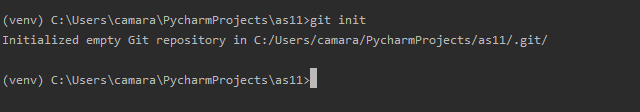


# Appendix 1: Reflective Learning Log

*Date: 25/10/2019*

*Work Completed: Init project. Setup git for version controlling*

*Understanding Achieved: Understanding python project structure.*



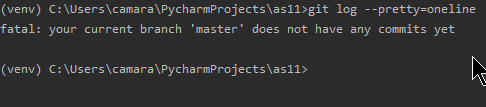


Figure Setting up git repository

*Date: 28/10/2019*

*Work Completed: Module Structure*

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

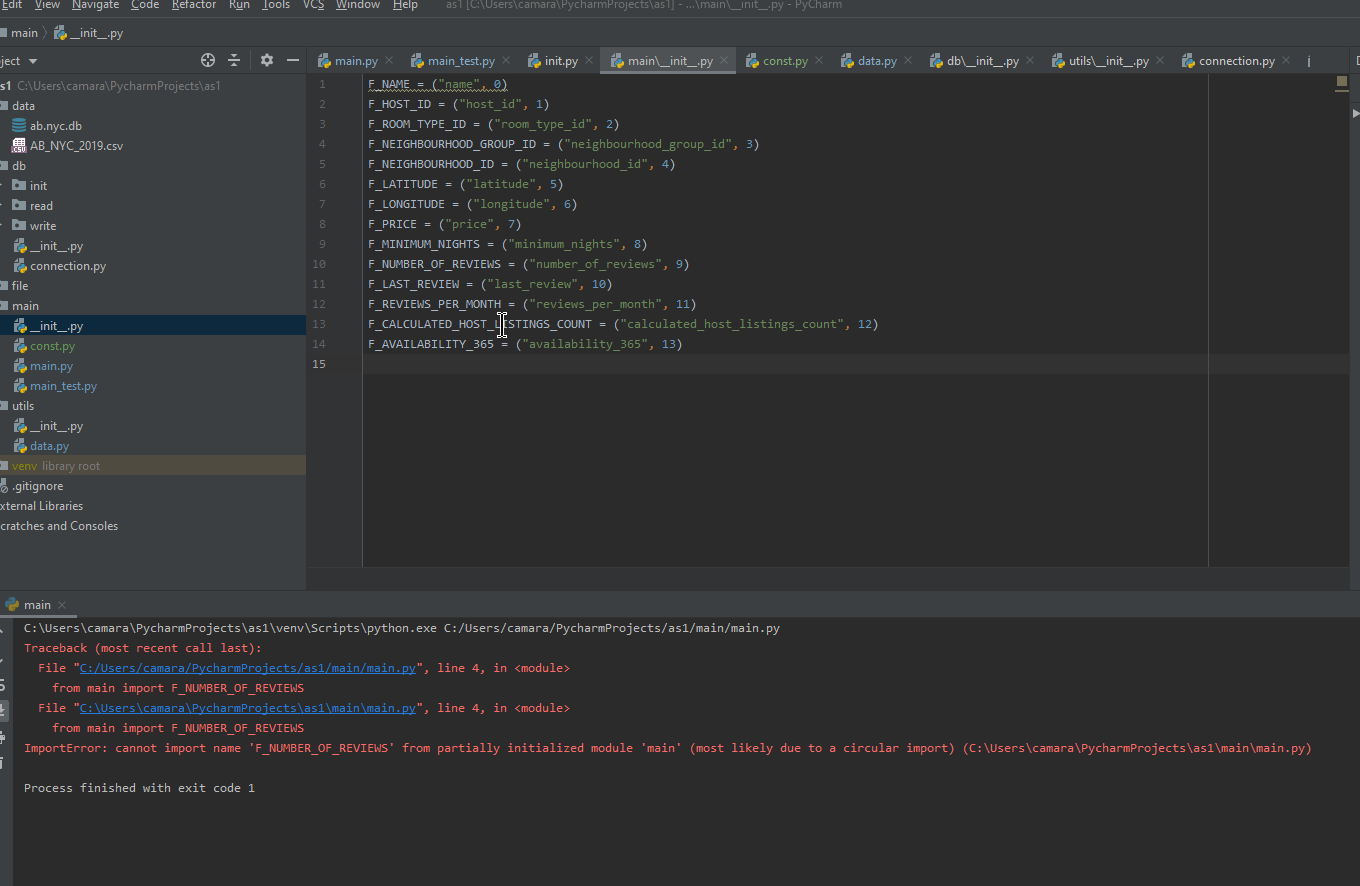


Figure Facing Circular import issue

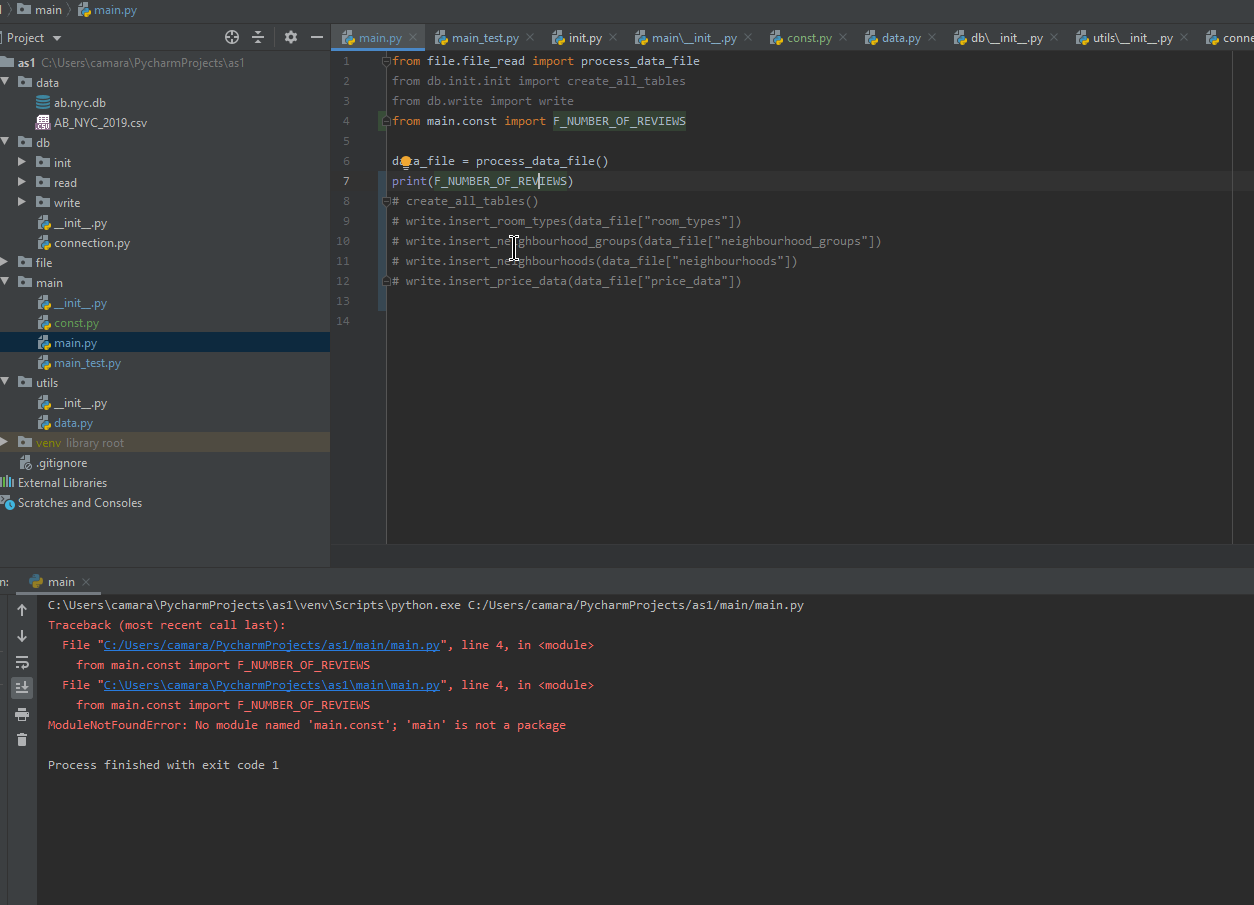
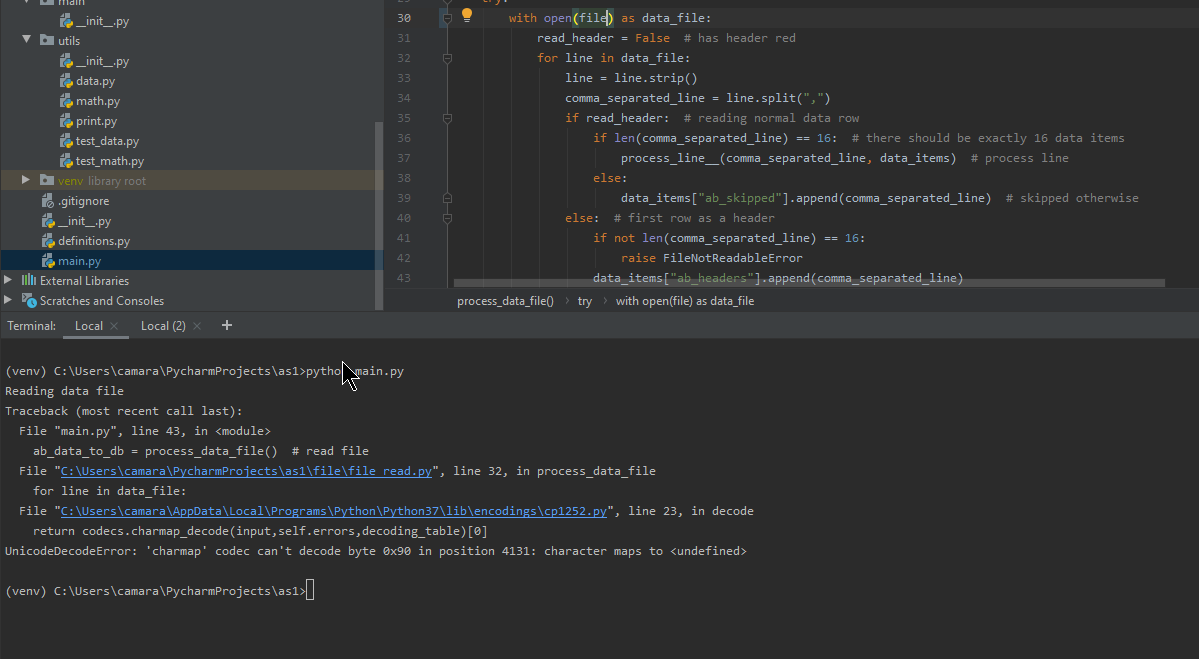


Figure Fixing Module not found issue

*Date: 29/10/2019*

*Work Completed: File Reading*

*Understanding Achieved: Read CVS file. Encoding issues.*



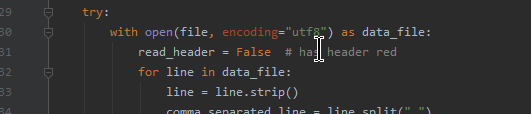


Figure Fixing Encoding issue while reading file

*Date: 30/10/2019*

*Work Completed: Test graph for project*

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

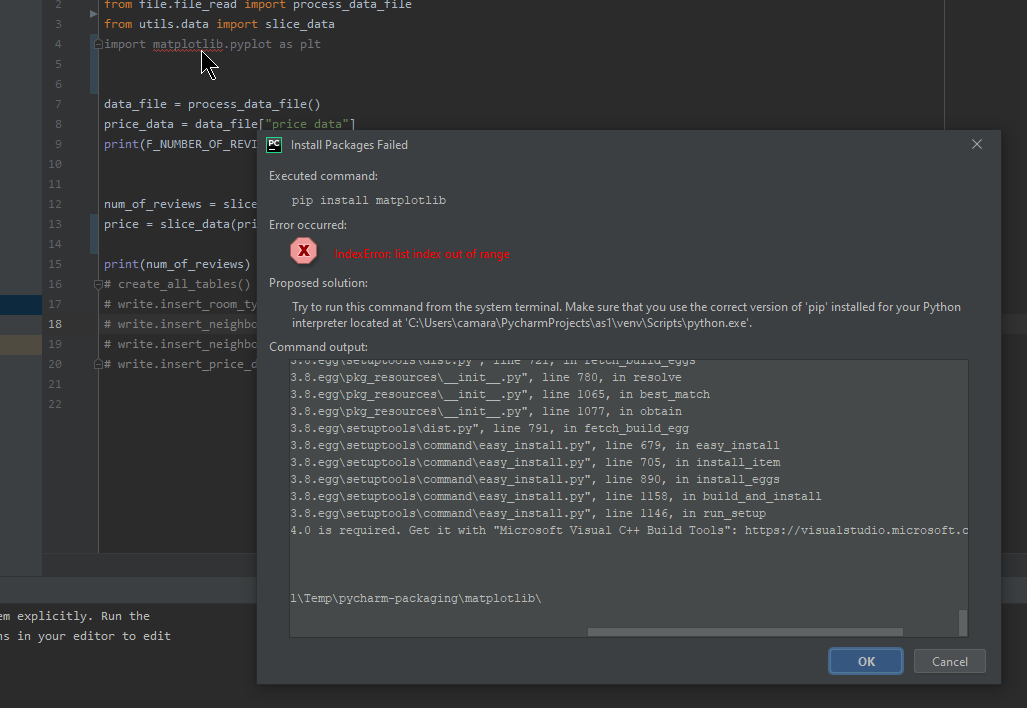


Figure VC++ dependency issue while installation matplotlib

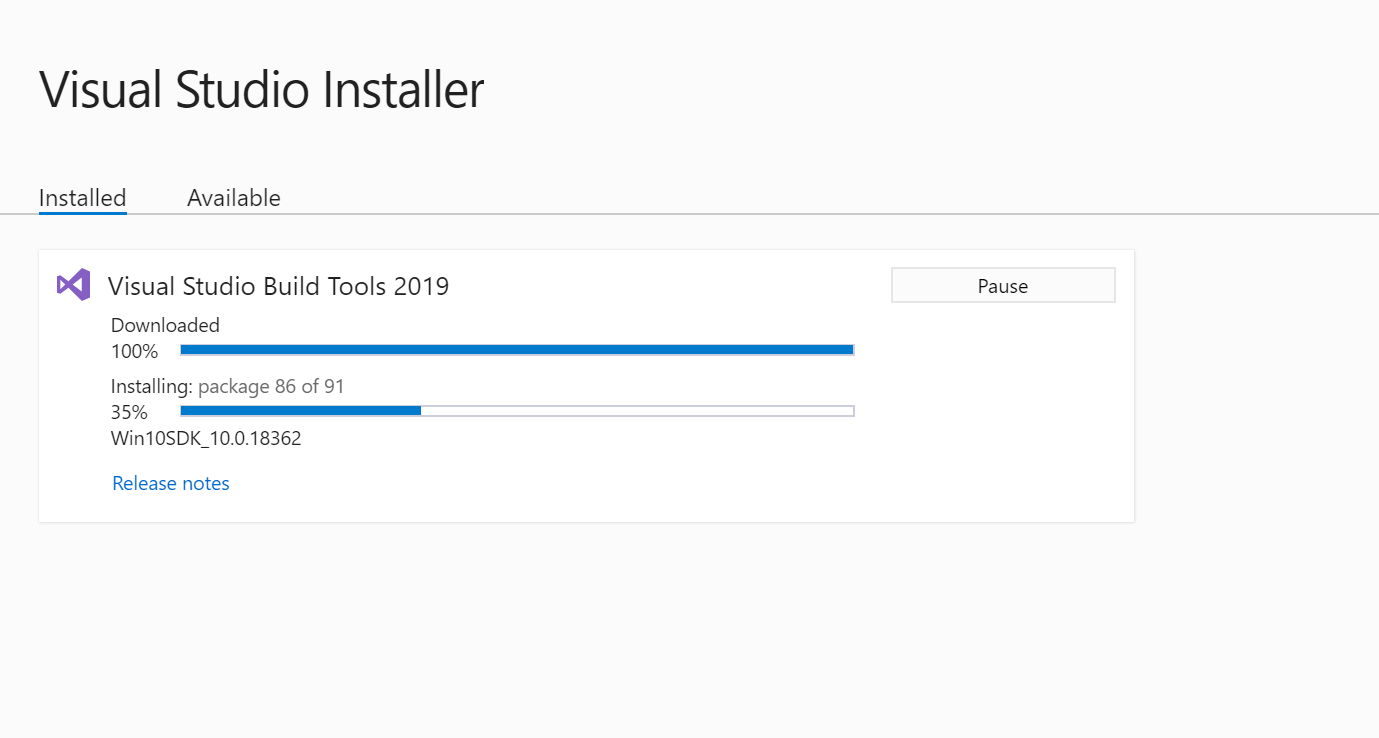


Figure Installing VS Build tool as a fix

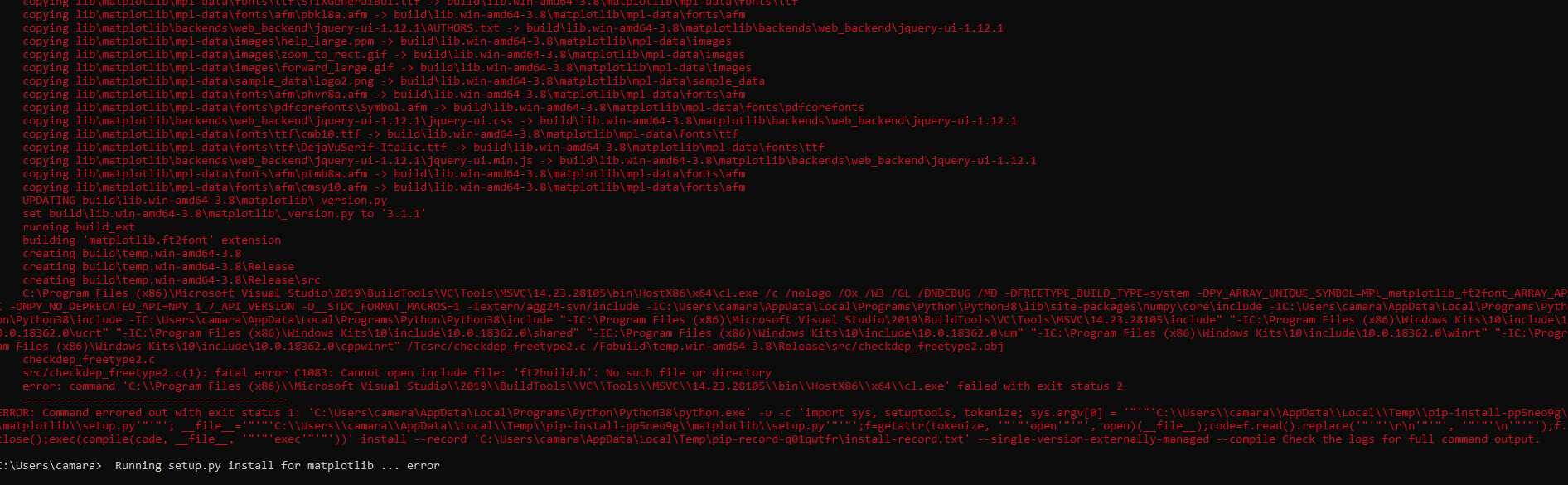


Figure Python 3.8 is not supported but matplotlib

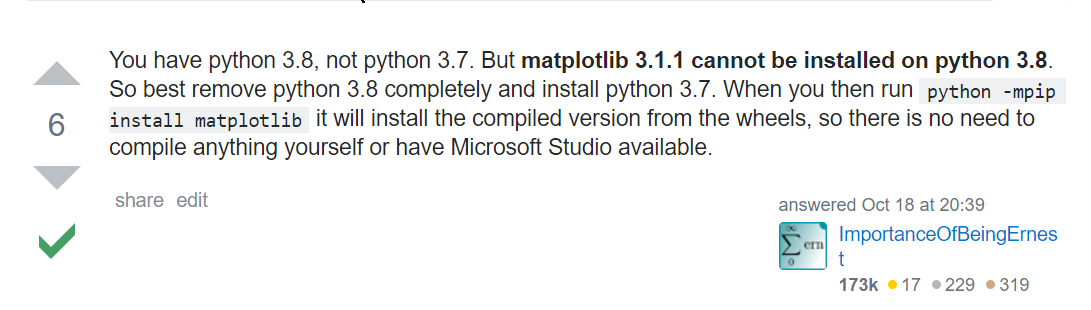


Figure Searched and found solution for matplotlib version issue

*Date: 31/10/2019*

*Work Completed: Basic Maths functions*

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

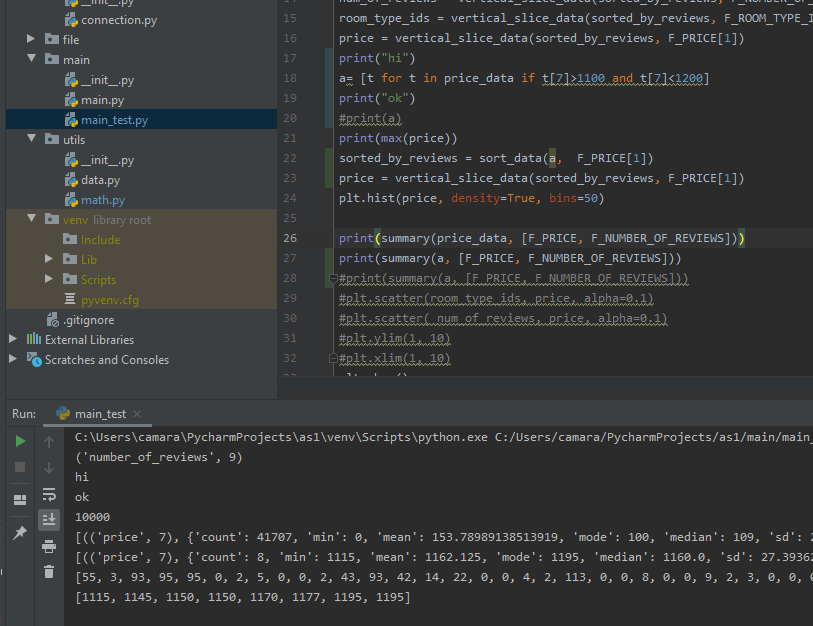
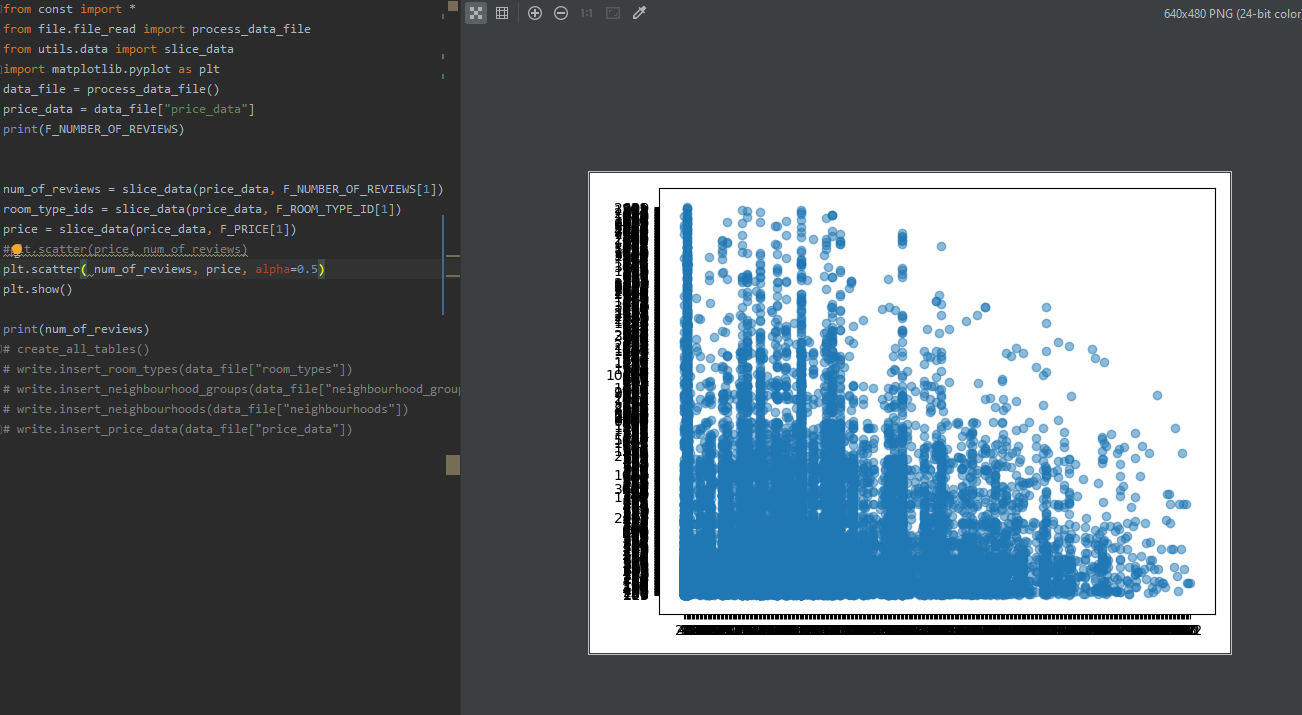
**

Figure 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*

**

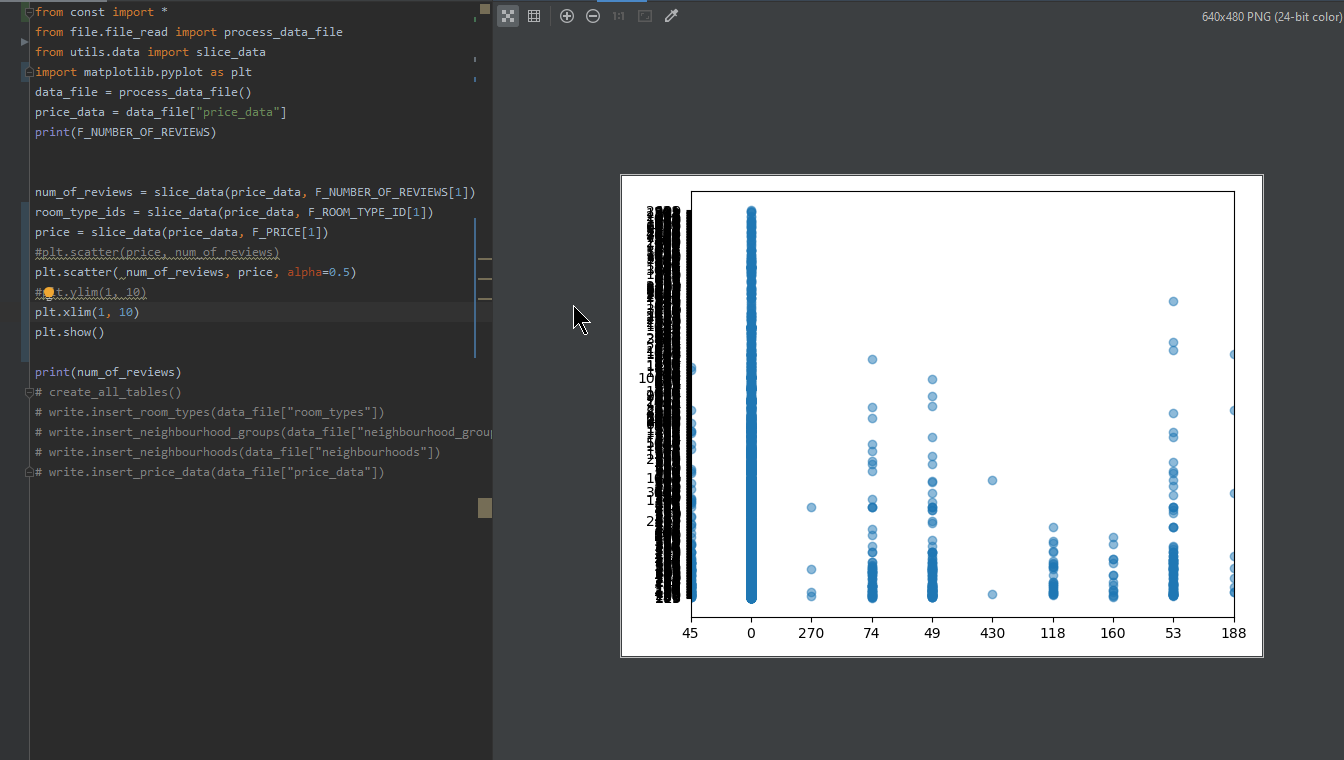


Figure Fixing overlapped x, y labels

*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.*



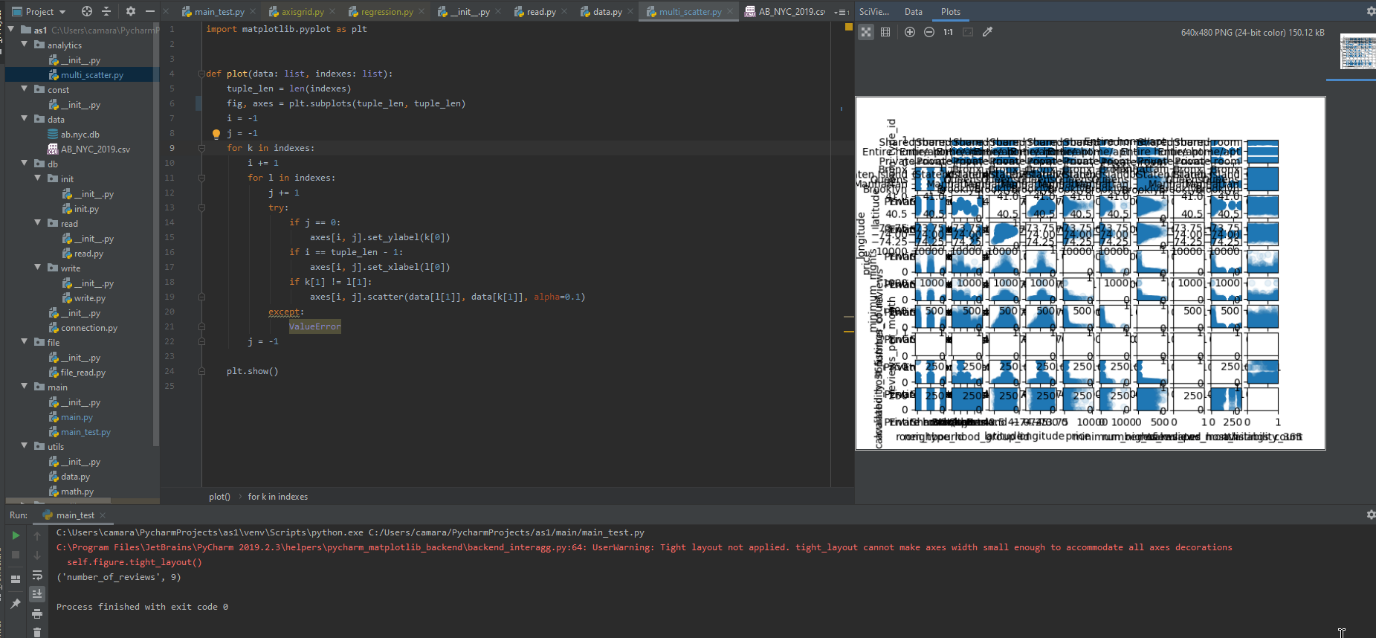


Figure Faced overlapped sub plots

*Date: 03/11/2019*

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

*Understanding Achieved: Change image size.*

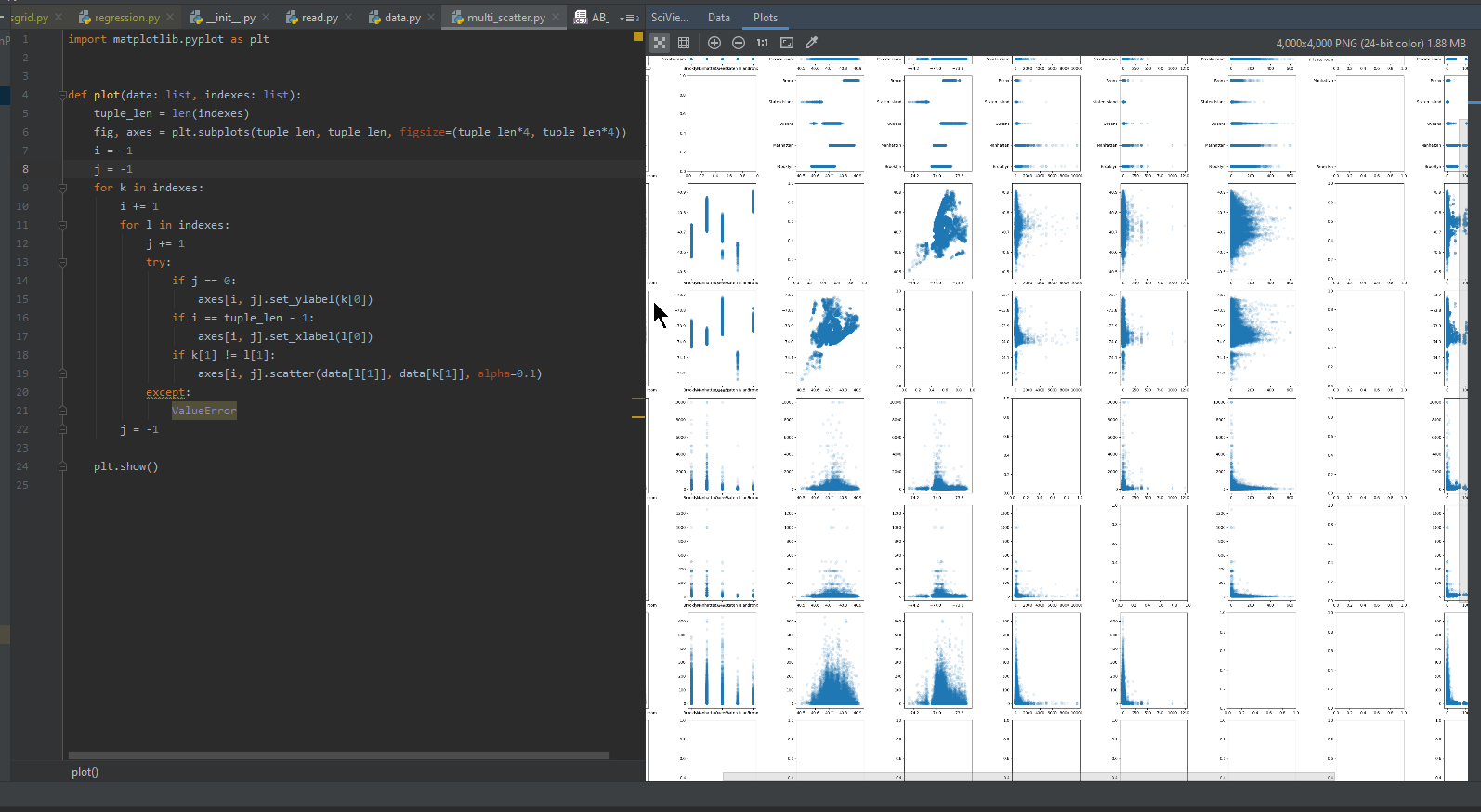


Figure Fixing overlapped subplots by introducing figsize

*Date: 04/11/2019*

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

*Understanding Achieved: Understand how seaborn achieved this*

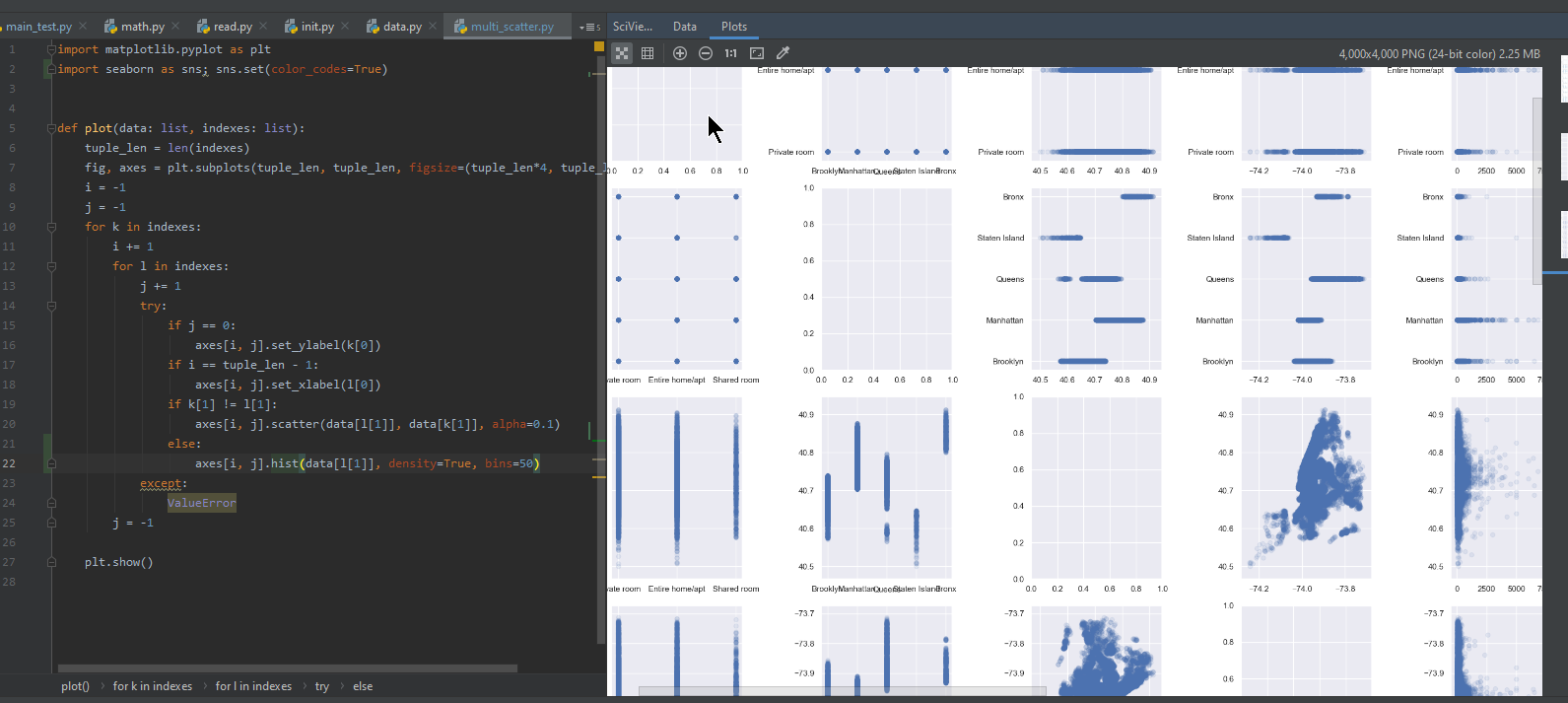


Figure Facing empty subplot for variable to same variable scatter plot

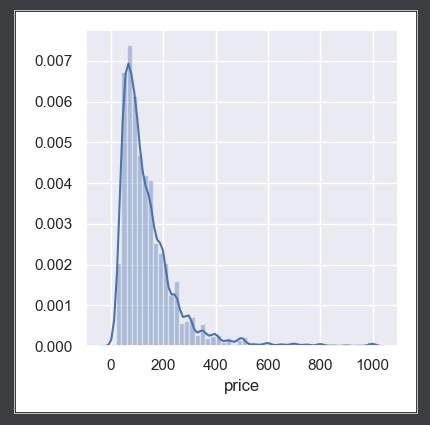
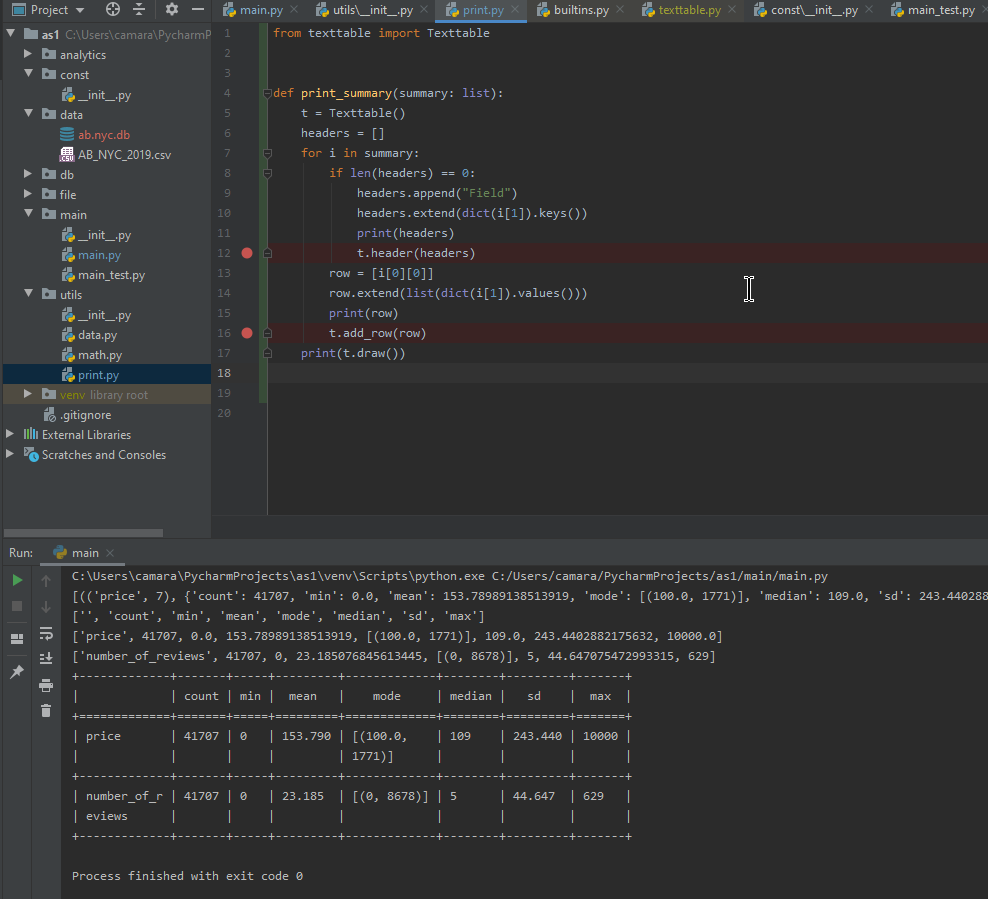


Figure Decided to fill empty subplot by a distribution graph

*Date: 05/11/2019*

*Work Completed: Table Printing*

*Understanding Achieved: Investigated various modules available for table printing*

**

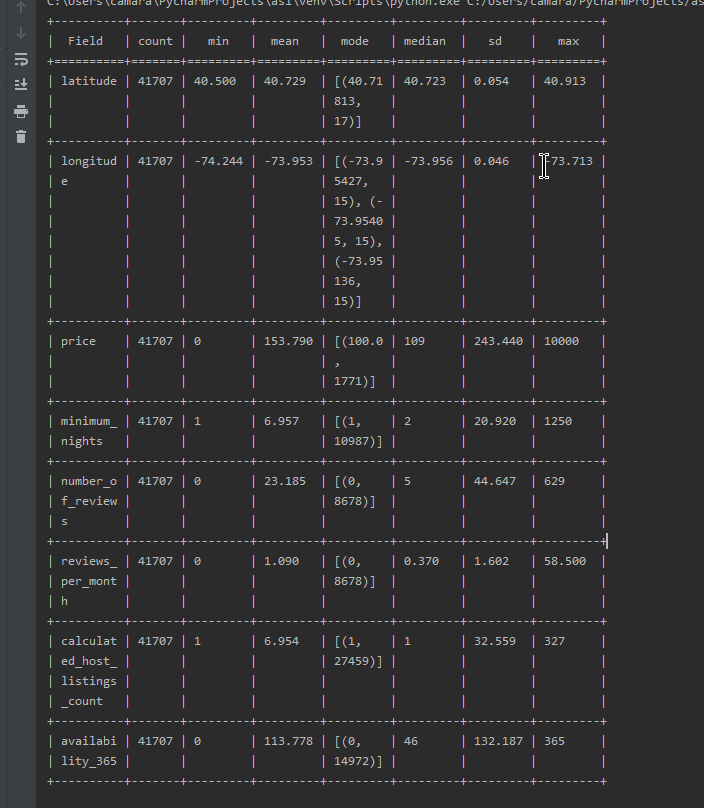
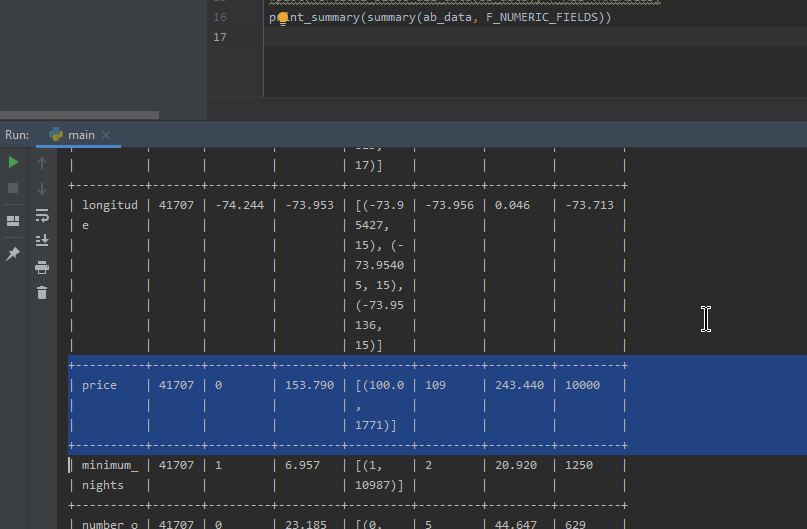
**

Figure 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.*



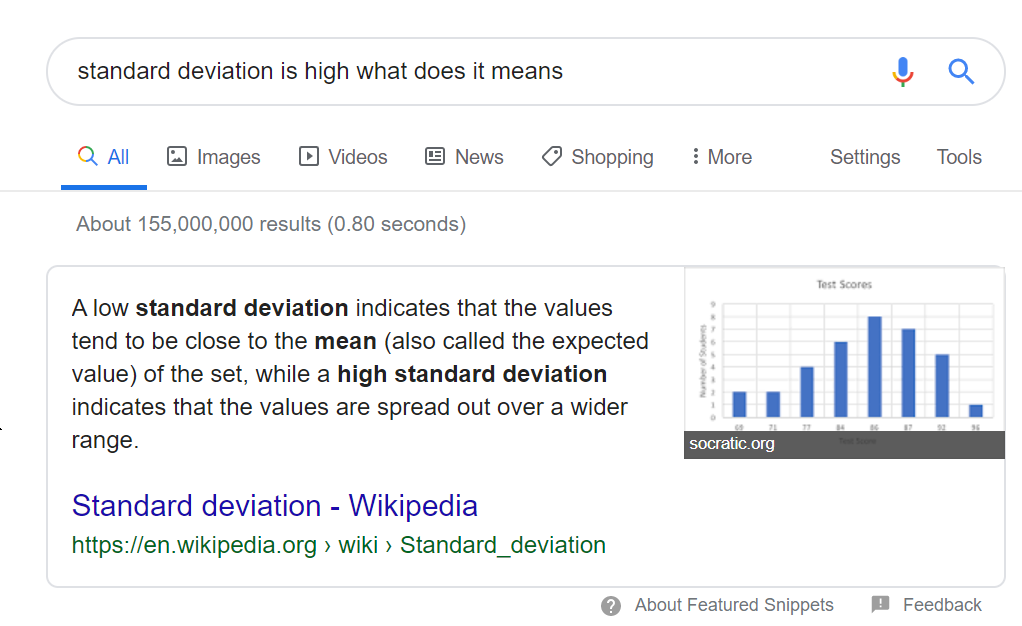
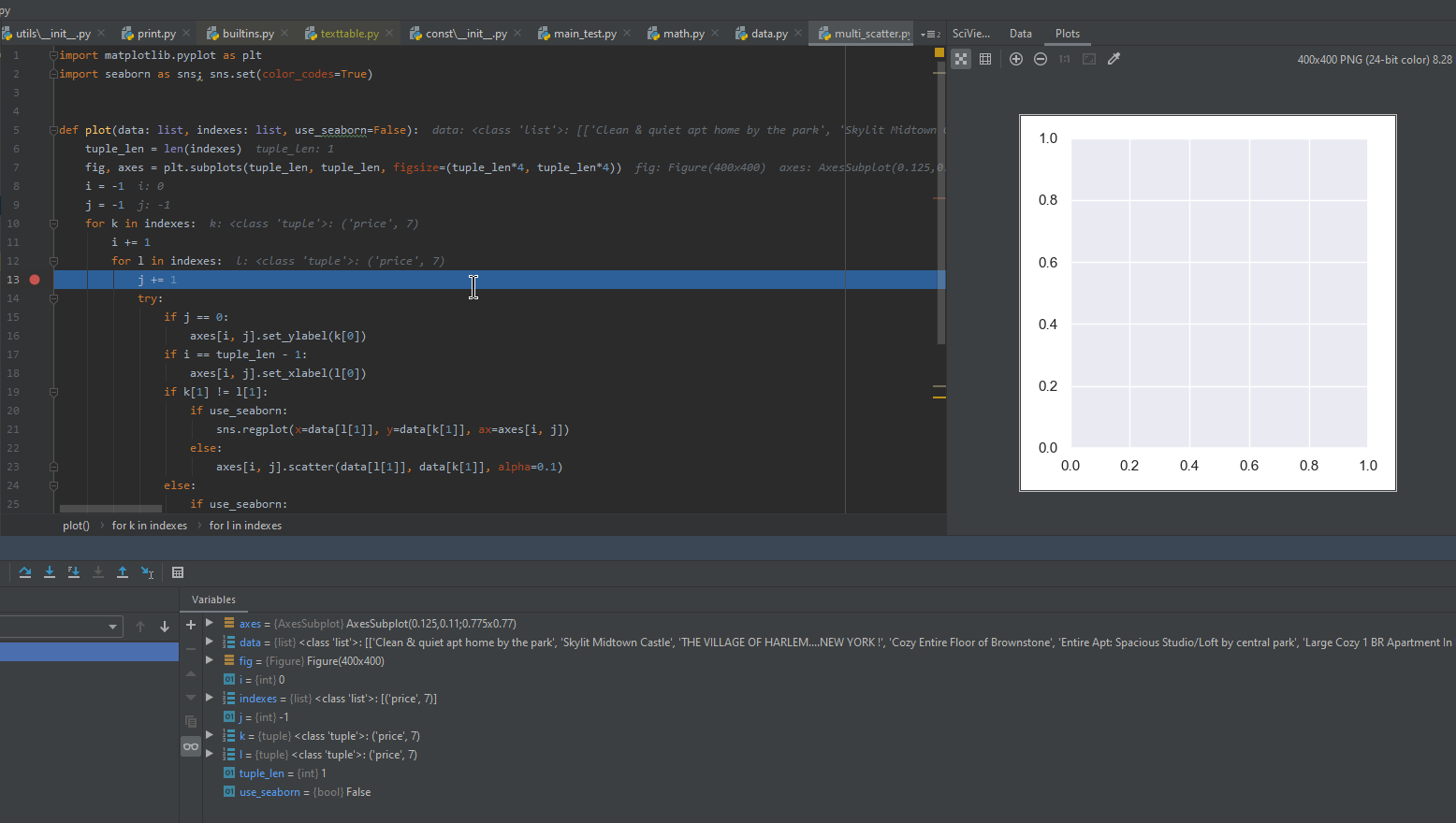


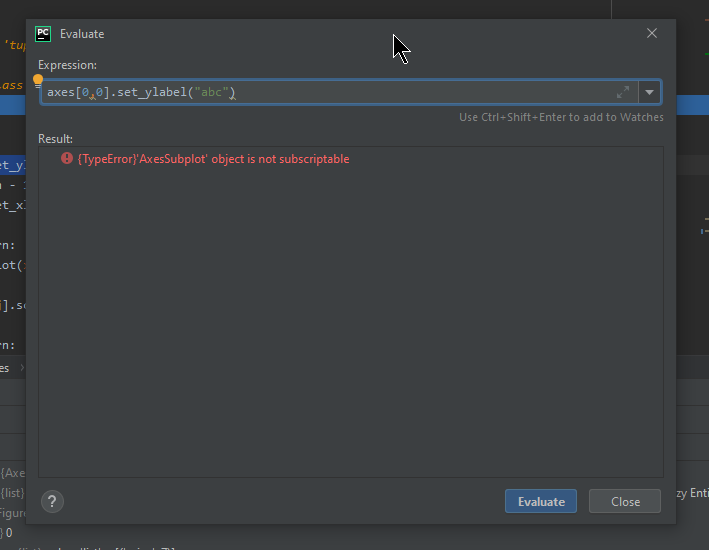
Figure Understanding Standard Deviation

*Date: 07/11/2019*

*Work Completed: Sub plotting*

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



**

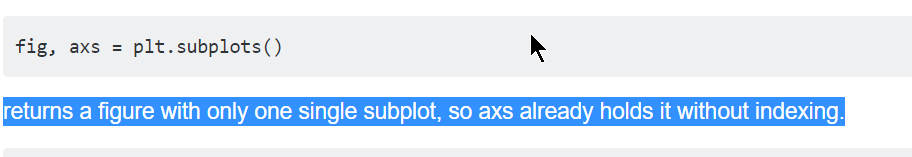
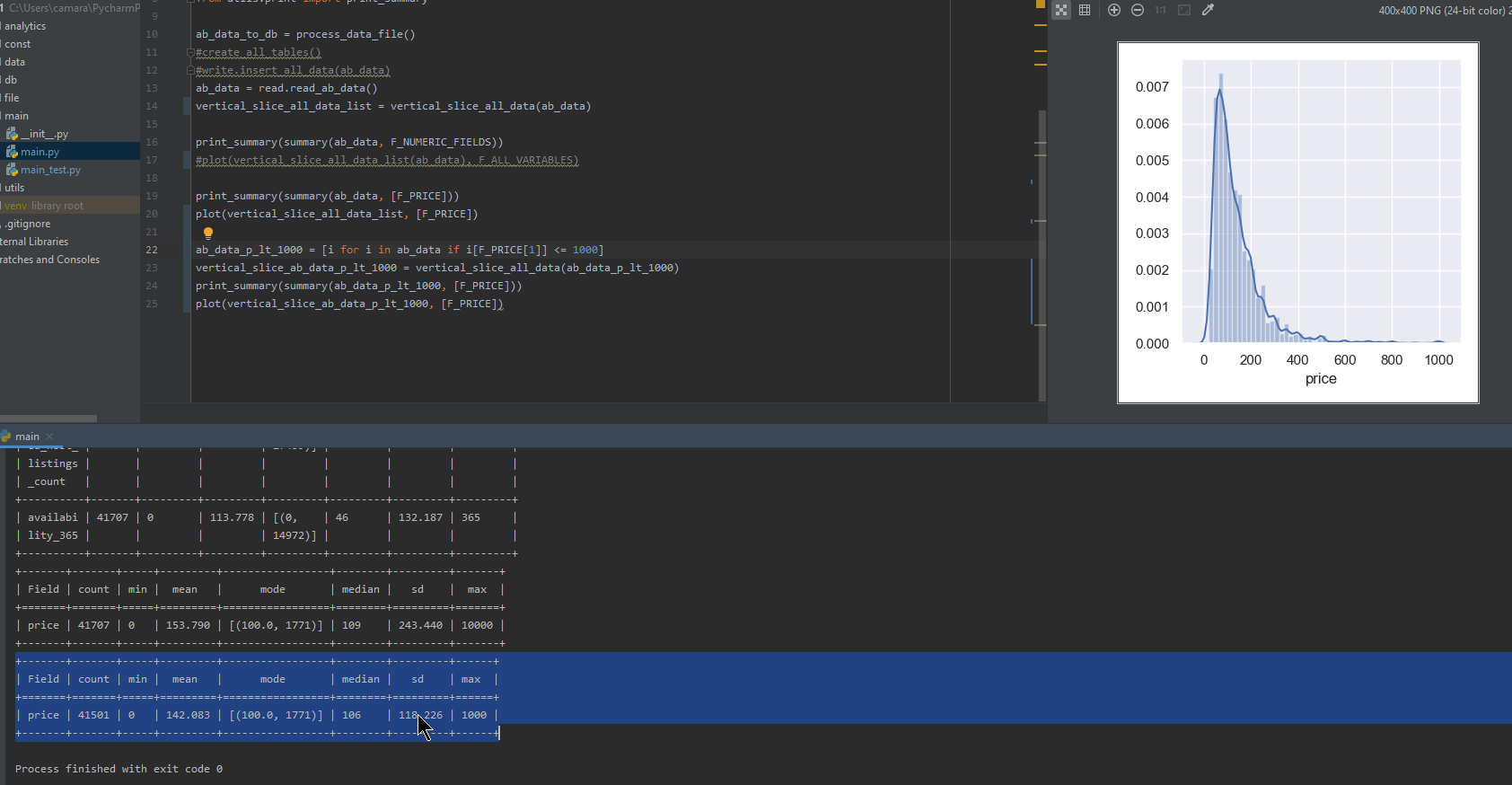
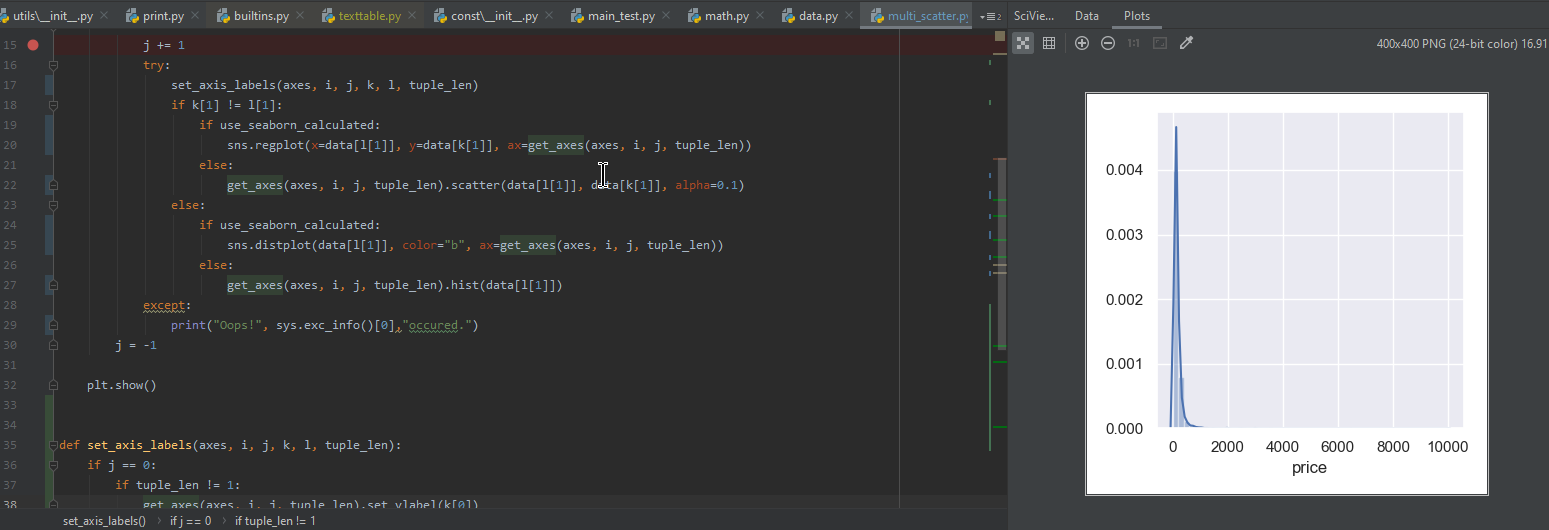
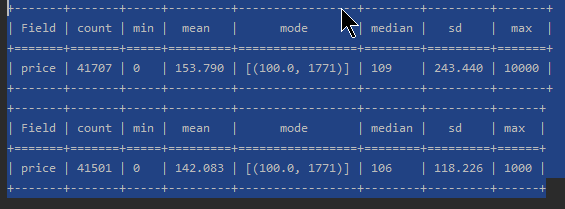
**

Figure Sub plotting without index causing issues

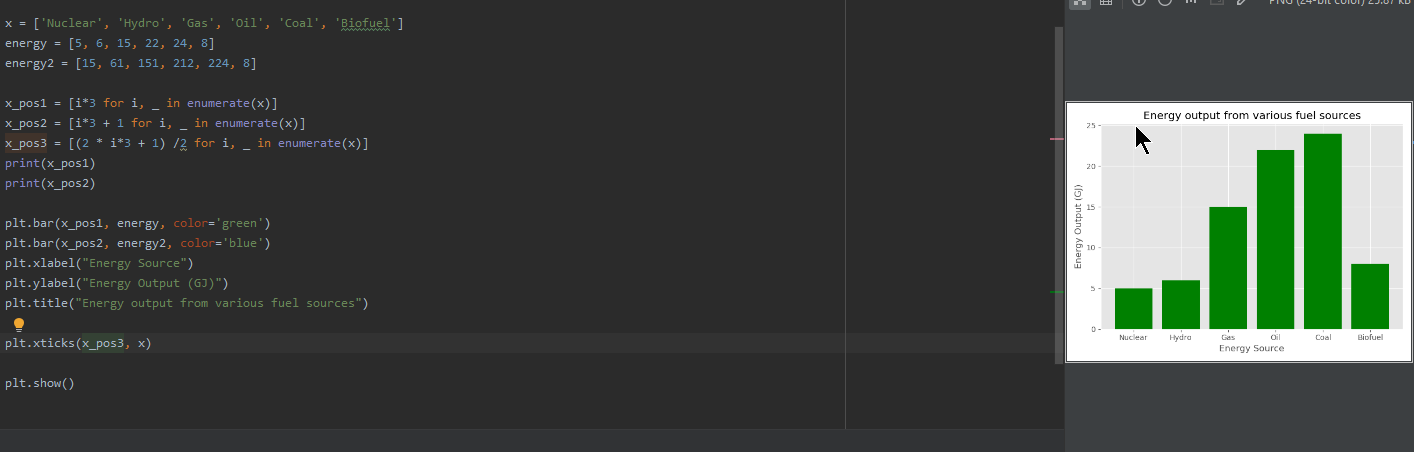
**



*Date: 09/11/2019*

*Work Completed: Grouped bar chat*

*Understanding Achieved: Data input formats for grouped bar charts.*

**

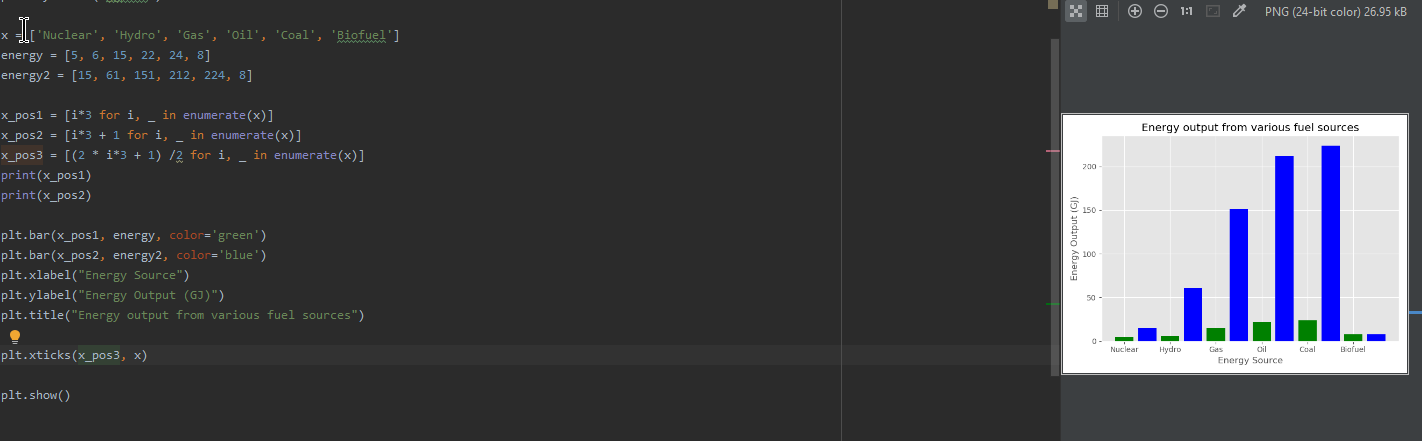


Figure Testing grouped bar charts

*Date: 10/11/2019*

*Work Completed: Pie Charts*

*Understanding Achieved: Understanding Data format for Pie Charts.*

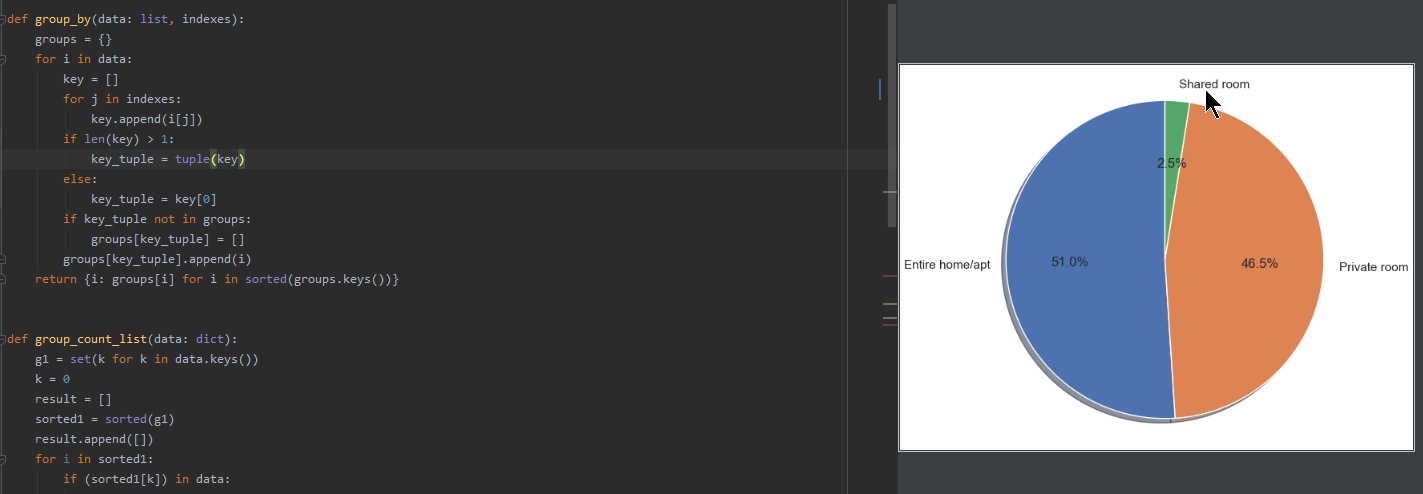
**

Figure Testing pie charts

*Date: 11/11/2019*

*Work Completed: Graph Saving*

*Understanding Achieved: Configurations available for graph saving*

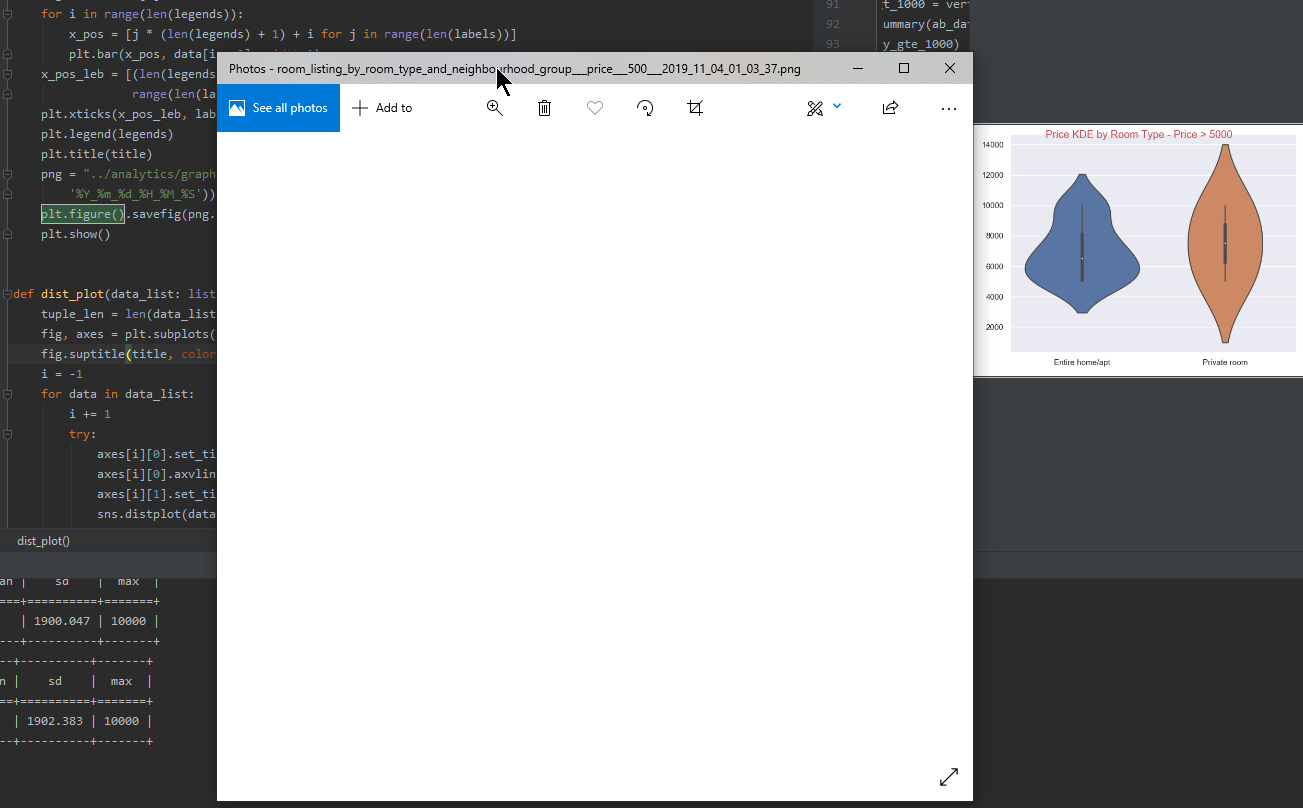
**

Figure Image saving issue

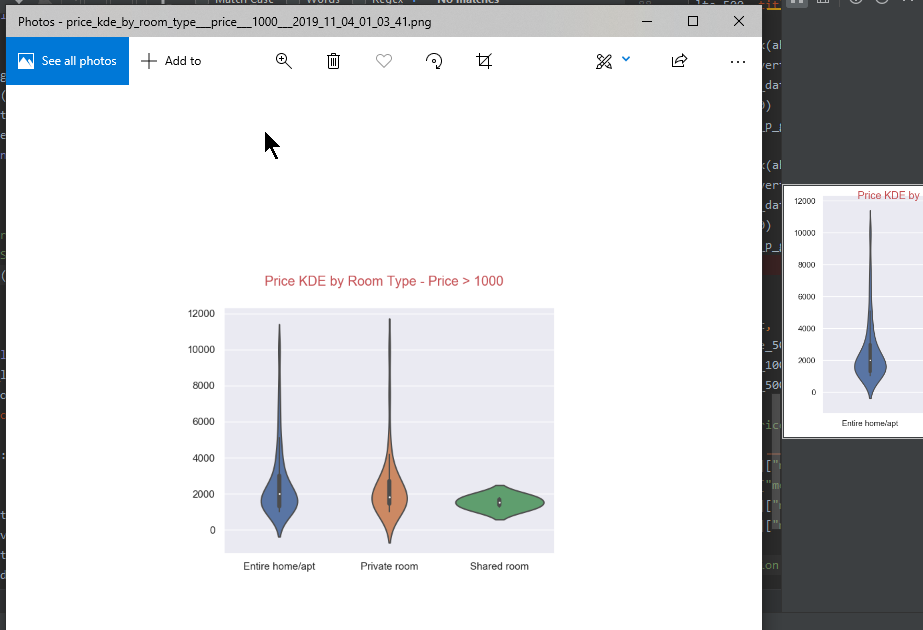
**

Figure 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*

**

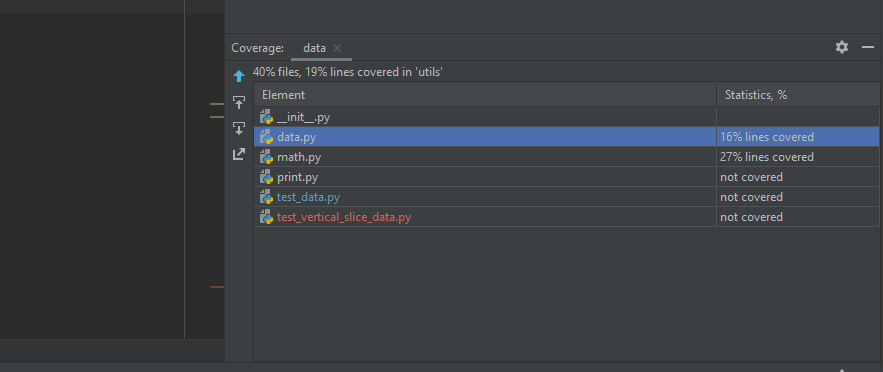
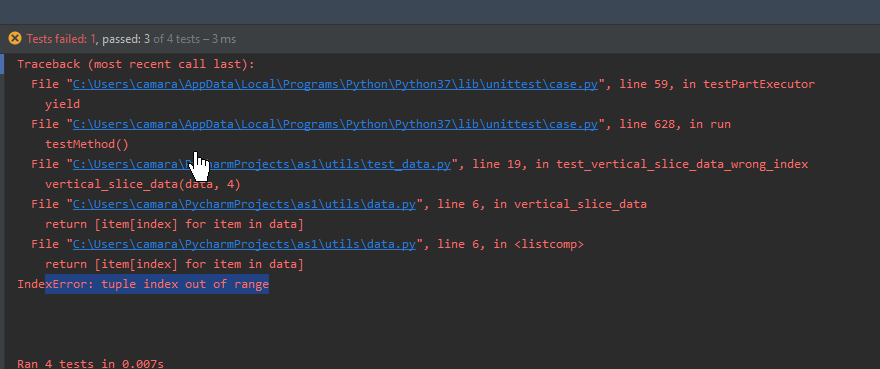
**

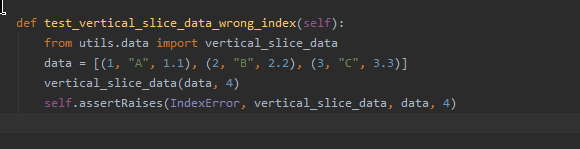
Figure Unittesting and test coverage

*Date: 14/11/2019*

*Work Completed: Unit Testing for Exceptions*

*Understanding Achieved: Parameter passing for Exception testing*

**

**

**

Figure Unit testing for exceptions

*Date: 17/11/2019*

*Work Completed: Initial steps for Pydoc*

*Understanding Achieved: Research on available documentation patterns*

**

Figure Python doc Google Standard

*Date: 14/11/2019*

*Work Completed: Pydoc*

*Understanding Achieved: Pydoc*

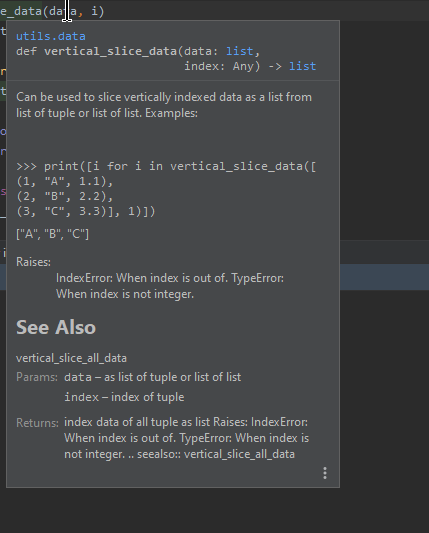
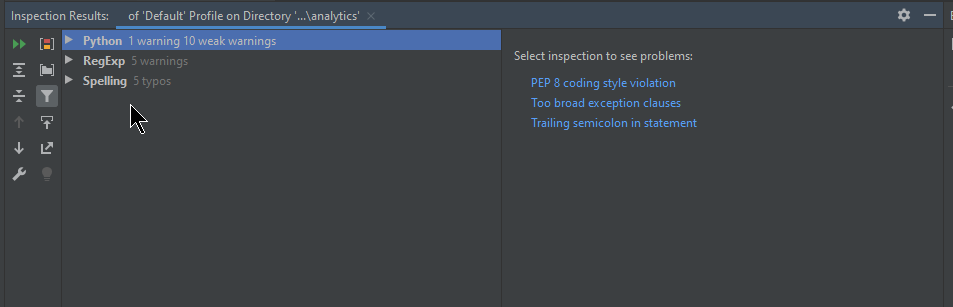
**

Figure 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*

**

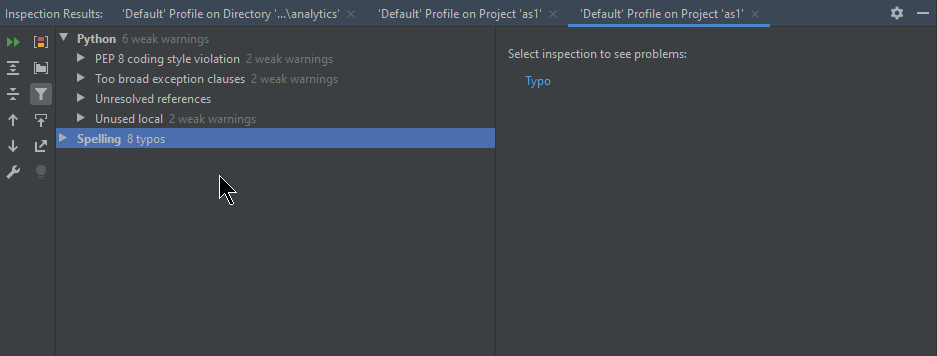
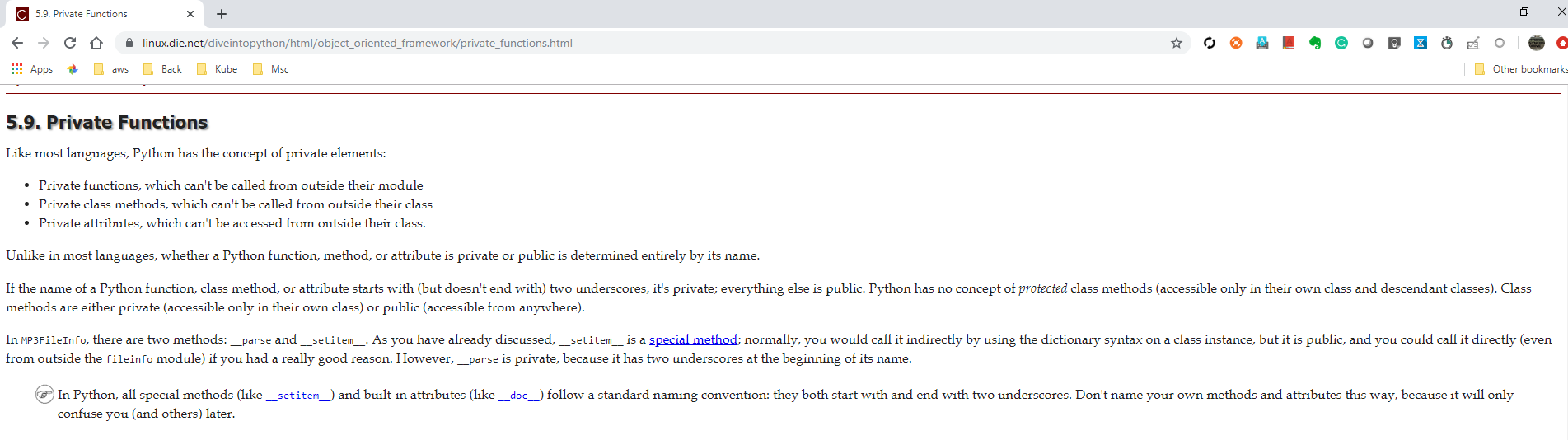
**

Figure Fixing Static code analysis warning and errors

*Date: 20/11/2019*

*Work Completed: Refactoring code*

*Understanding Achieved: Research more on Coding best practises*

**

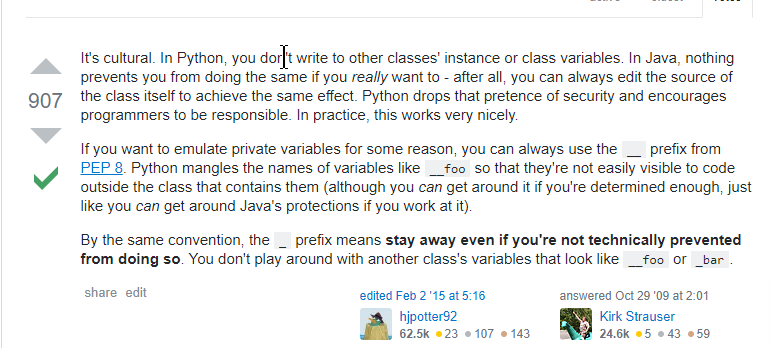
**

Figure 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*

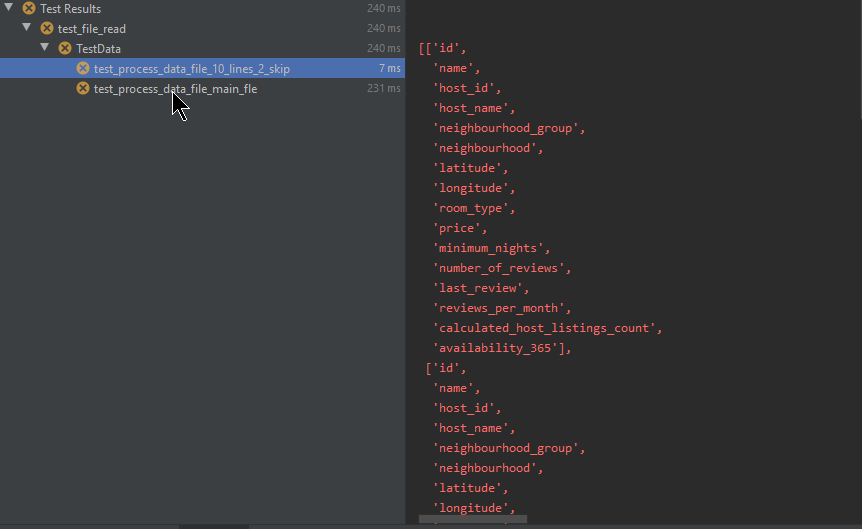
**

Figure 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*

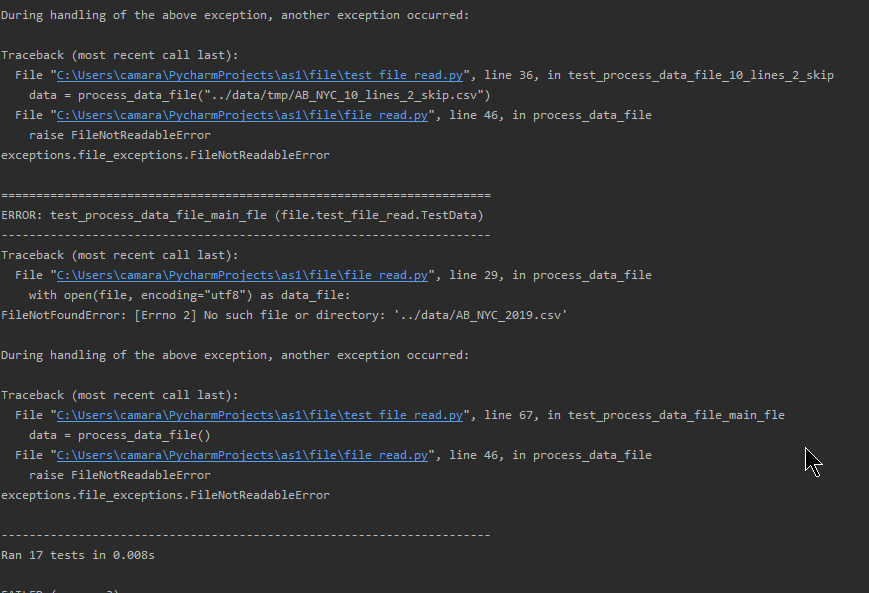
**

Figure 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*

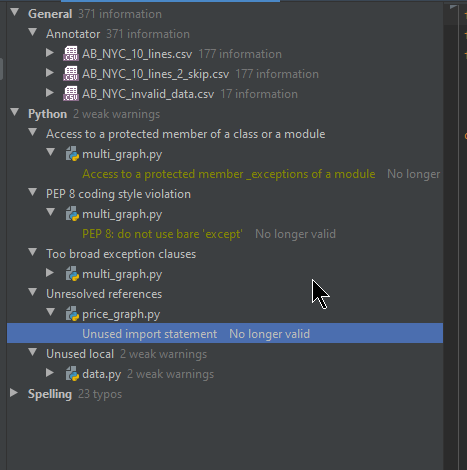
**

Figure Code refactoring

*Date: 26/11/2019*

*Work Completed: Fixed large table printing issues*

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

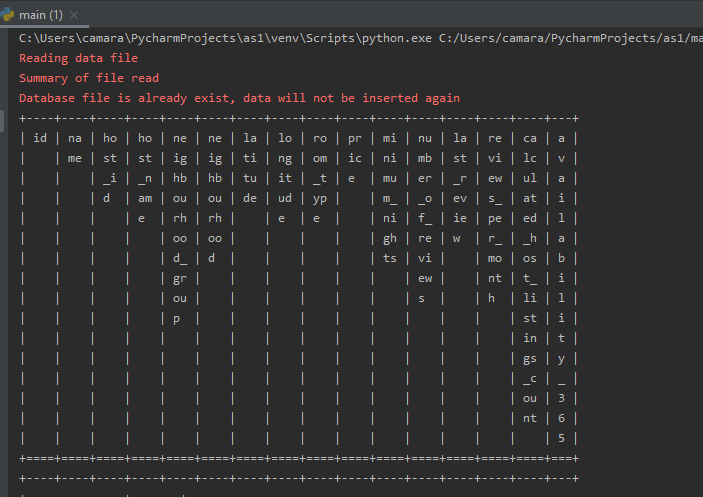
**

Figure : Large table print casing alignment issues

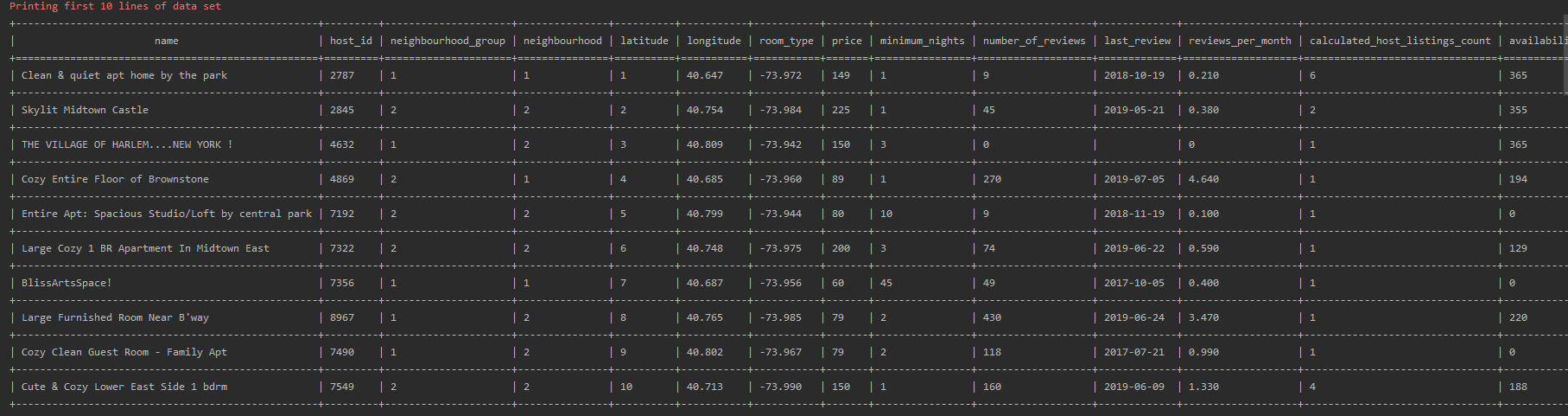
**

Figure : Fixed table size issue

*Date: 27/11/2019*

*Work Completed: User friendly logging for python*

*Understanding Achieved: Understanding available logging options for python*

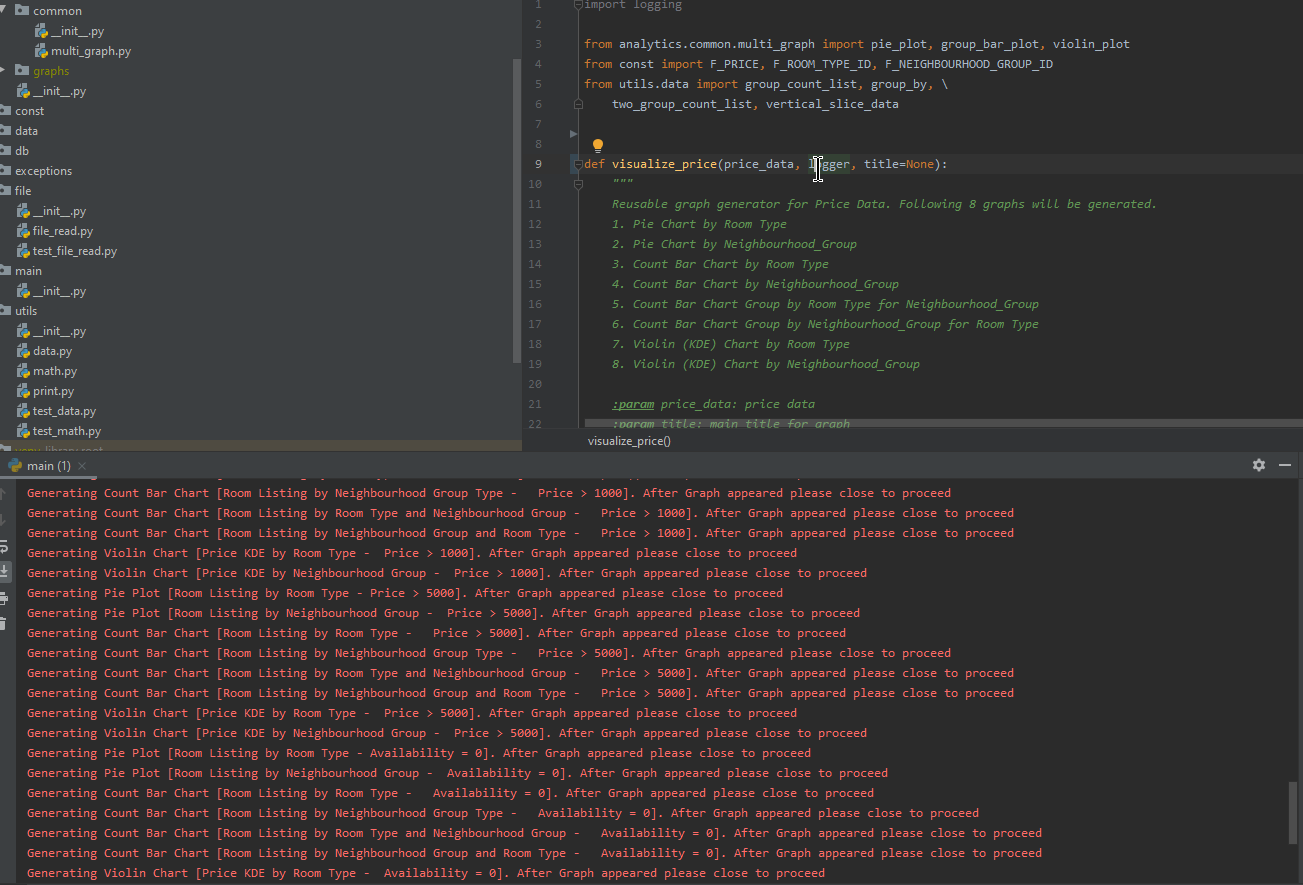
**

Figure : 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

**Turgut, Duygu.** "Airbnb NYC Price Prediction". <https://www.kaggle.com/duygut/airbnb-nyc-price-prediction>>. November 1st 2019

**W., Andrew.** “Smart Pricing with XGB, RFR + Interpretations”. <<https://www.kaggle.com/jrw2200/smart-pricing-with-xgb-rfr-interpretations>>. November 1st 2019.

**Domazet, Josip.** “Mining NYC Airbnb data using R”. < <https://www.kaggle.com/josipdomazet/mining-nyc-airbnb-data-using-r>>. November 1st 2019.

**Inside Airbnb.** “Get Data”**. <**<http://insideairbnb.com/get-the-data.html>**>.** November 1st 2019

1. <https://en.wikipedia.org/wiki/Airbnb> [↑](#footnote-ref-1)
2. <http://insideairbnb.com/> [↑](#footnote-ref-2)
3. <https://www.kaggle.com/dgomonov/new-york-city-airbnb-open-data> [↑](#footnote-ref-3)
4. <https://enterprisecraftsmanship.com/posts/cohesion-coupling-difference/> [↑](#footnote-ref-4)