Hackwagon Academy - DS101

AirBnB Project

Learning Outcomes:

- Learn how to translate business requirements into workable applications
- Declare variables, and manipulate the variables to perform arithmetic operations
- Create a list, append new elements to a list, remove elements from list, and access elements within a list
- Create a dictionary, access data, and update information within the dictionary
- Be able to aptly make use of if and nested if constructs
- Variable conversion
- Produce visualisations
- Able to come up with insights based on the data

In [ ]:

```
#Before you start, please perform the following 2 steps:
#1. Rename the file to Anurag_Chatterjee_DS101_Airbnb_Project e.g.
john_doe_DS101_Airbnb _Project

#2. Fill in your details here:
#Name : ____________Anurag Chatterjee___________

#Start of Course Class(Edit accordingly): __23 Sep 2019 9.00am____

# FOR TA/INSTRUCTOR:
# Total Marks: / 100
# Part 1: / 5
# Part 2: / 35
# Part 3: / 60
```

This project is split into 4 different parts: 1. Data Cleaning (5 marks) 2. Explorator Data Analysis (25 marks) 3. Interpretation (10 marks) 4. AirBnB Visualisation and Price Recommender App (60 marks) **All questions must follow expected output to be awarded full marks except the following:** 1. Interpretation

References

Important Collections Functions

Creation

Collection Type	Function	Examples

`list`	None	`new_list = []`
		`new_list = [1,2,3,4]`

`dict`	None	`new_dict = {}`
		`new_dict = {'a': 1, 'b':2}`

Add / Appending Data

Collection Type	Functions	Examples	Resulting Output

`list`	`.append()`	`new_list = [1,2,3]`
		`new_list.append(4)`
			`[1,2,3,4]`

`list`	`.extend()`	`new_list = [1,2]`
		`new_list.extend ([3,4])`
			`[1,2,3,4]`

`dict`	None	`new_dict = {}`
		`new_dict['a'] = 1`
		`new_dict['b'] = 2`
			`{'a': 1, 'b':2}`

Updating / Changing Data

Collection Type	Functions	Examples	Resulting Output

`list`	None	`new_list = [1,2,3]`
		`new_list[0] = 5`
			`[5,2,3]`

`dict`	None	`new_dict = {'a': 1, 'b': 2}`
		`new_dict['a'] = 10`
			`{'a': 10, 'b':2}`

Accessing / Taking Out Data

Collection Type	Functions	x to be	Examples

`list`	None	3	`new_list = [1,2,3]`
			`x = new_list[2]`

`list of list`	None	3	`new_list = [[1,2], [3,4]]`
`list of dict`	None	2	

```
x = new_list[1][0]
```

```
new_list =
[{'a':1},{'b':2}]
```

`list of dict` None 2

```
x = new_list[1]
['b']
```

```
new_dict = {'a': 1,
```

`dict` None 2

CITU Framework & Applied Iterations

1. What variables do you need to answer this question?
2. **Create** the results container
3. **Iterate** the input data/list
4. **Take out the variables you needed in step 1**
5. **Test** conditions of each value
6. **Update** the results container when condition is fulfilled

```
x = [10,20,50,2,4]
x.sort()
print(x) # [2,4,10,20,50]
x.sort(reverse=True)
print(x) # [50,20,10,4,2]
```

Further explore the .sort() function in the documentation

Search up 'list .sort() python 3'

```
'b':2}
```

```
x = new_dict['b']
```

Sorting Values

</hr>

Welcome to your final project of Hackwagon Academy DS101! You've come a long way since the start of this course and if you've been on track with our exercises, you should find this doable.

Airbnb is an online marketplace and hospitality service, enabling people to lease or rent short-term lodging including vacation rentals, apartment rentals, homestays, hostel beds, or hotel rooms. The company does not own any lodging; it is merely a broker and receives percentage service fees (commissions) from both guests and hosts in conjunction with every booking. In this project, we aim to use algorithms and libraries to mine the reviews people have submitted on Singapore AirBnB rentals in order to provide descriptive analytics.

Load File

Load the `airbnb_data.csv` as **a list of dictionaries** into a new variable called `airbnb_data` . Once you load the data, you should see something like this when you print `airbnb_data` :

```
    [
            {
```

```
            'listing_id': '1133718',
            'survey_id': '1280',
            'host_id': '6219420',
            'room_type': 'Shared room',
            'country': '',
            'city': 'Singapore',
            'borough': '',
            'neighborhood': 'MK03',
            'reviews': '9',
            'overall_satisfaction': '4.5',
            'accommodates': '12',
            'bedrooms': '1.0',
            'bathrooms': '',
            'price': '74.0',
            'minstay': '',
            'last_modified': '2017-05-17 09:10:25.431659',
            'latitude': '1.293354',
            'latitude': '1.293354',
            'longitude': '103.769226',

    'location': '0101000020E6100000E84EB0FF3AF159409C69C2F693B1F 43F'In [1]:
```

```
            }
 # Read file into a list called airbnb_data
 # Write your code below:
            ...
 import csv
        ]
with open('airbnb_data.csv') as file:
    airbnb_data = []
    for row in csv.DictReader(file):
        airbnb_data.append(dict(row))
print(airbnb_data[0])

 # Well done
```

```
{'listing_id':  '1133718',  'survey_id':  '1280',  'host_id':  '6219420',  'room_type':
'Share  d  room',  'country':  '',  'city':  'Singapore',  'borough':  '',  'neighborhood':
'MK03',   'rev   iews':   '9',   'overall_satisfaction':   '4.5',   'accommodates':   '12',
'bedrooms':  '1.0',  'bat  hrooms':  '',  'price':  '74.0',  'minstay':  '',  'last_modified':
'2017-05-17   09:10:25.43165   9',   'latitude':   '1.293354',   'longitude':   '103.769226',
'location': '0101000020E6100000E8 4EB0FF3AF159409C69C2F693B1F43F'}
```

Data Cleaning [5 marks]

Once this is done correctly, you do not need to change the type for the remaining parts of your project.

Very Big Hint:

```
for row in data:
    row['score'] = int(row['score'])
```

Preview your data and clean them to appropriate type. Namely these columns:

1. `overall_satisfaction`

2. price
3. longitude
4. latitude
5. reviews

Expected Output:

```
{
        'listing_id': '1133718',
        'survey_id': '1280',
        'host_id': '6219420',
        'room_type': 'Shared room',
        'country': '',
        'city': 'Singapore',
        'borough': '',
        'neighborhood': 'MK03',
        'reviews': 9.0,
        'overall_satisfaction': 4.5,
        'accommodates': '12',
        'bedrooms': '1.0',
        'bathrooms': '',
        'price': 74.0,
        'minstay': '',
        'last_modified': '2017-05-17 09:10:25.431659',
        'latitude': 1.293354,
        'longitude': 103.769226,
        'location':
'0101000020E6100000E84EB0FF3AF159409C69C2F693B1F4 3F'
    }
```

In [2]:

```
# Write your code below:
for row in airbnb_data:
    row['overall_satisfaction'] = float(row['overall_satisfaction'])
    row['price'] = float(row['price'])
    row['longitude'] = float(row['longitude'])
    row['latitude'] = float(row['latitude'])
    row['reviews'] = float(row['reviews'])
print(airbnb_data[0])
# Well done
```

```
{'listing_id': '1133718', 'survey_id': '1280', 'host_id': '6219420', 'room_type':
'Share d room', 'country': '', 'city': 'Singapore', 'borough': '', 'neighborhood':
'MK03', 'rev iews': 9.0, 'overall_satisfaction': 4.5, 'accommodates': '12', 'bedrooms':
'1.0', 'bathr ooms': '', 'price': 74.0, 'minstay': '', 'last_modified': '2017-05-17
09:10:25.43165
9', 'latitude': 1.293354, 'longitude': 103.769226, 'location':
'0101000020E6100000E84EB0 FF3AF159409C69C2F693B1F43F'}
```

Exploratory Data Analysis

The data team at AirBnB wishes to find out the answers to a few simple questions on the existing listings in Singapore. Your goal is to manipulate the data you have stored in the list of dictionaries and **understand some of the basic statistics of your dataset.** The following are some of the common *first* questions asked.

Q1. List out each neighborhoods and their number of listings [5 marks]

Hint

1. *Counting with dictionaries, where key is neigbhorhood id, value is counts. </i>*

Expected Output:

```
print(results['TS17']) # 342 counts.
```

In [3]:

```
# Write your code below:
counts = {}
for x in airbnb_data:
    neighborhood = x['neighborhood']
    if neighborhood not in counts:
        counts[neighborhood] = 1
    else:
        counts[neighborhood] += 1
print(counts['TS17'])
# Well done
```

```
342
```

Q2. List out each neighborhood and their average overall_satisfaction [5 marks] Note: You should filter out listings whose reviews are 0.

Hint

1. *Create 2 dictionaries*
    - *Dictionary 1: Key is neighborhood, value is an accumulation of scores*
    - *Dictionary 2: Key is neighborhood, value is an accumulation of counts*
2. *Create 3rd dictionary*
3. *Loop through 1 dictionary (using 1 for loop only because both share same key!), calculate average and store in 3rd dictionary </i>*

Expected Output:

```
print(results['TS17']) # it should give you an average score of
2.859 447004608295.
```

In [4]:

```
counts = {}
total_score = {}
for x in airbnb_data:
    neighborhood = x['neighborhood']
    overall_satisfaction = x['overall_satisfaction']
    reviews = x['reviews']
    if reviews != 0:
        if neighborhood not in counts:
            counts[neighborhood] = 1
        else:
            counts[neighborhood] += 1
        if neighborhood not in total_score:
            total_score[neighborhood] = overall_satisfaction
        else:
            total_score[neighborhood] += overall_satisfaction
results = {}
for neighborhood, scores in total_score.items():
        average_overall_satisfaction = scores / counts[neighborhood]
        if scores == 0:
            results[neighborhood] = 0
        else:
            results[neighborhood] = average_overall_satisfaction
print(results['TS17'])
# Well done
```

```
2.859447004608295
```

Q3. List out each neighborhood and their average price [5 marks]

Hint

1. *Similar to previous question </i>*

Expected Output:

```
print(results['TS17']) # it should give you an average price of
95.56 72514619883.
```

In [5]:

```
# Write your code below:
counts = {}
for x in airbnb_data:
    neighborhood = x['neighborhood']
    if neighborhood not in counts:
        counts[neighborhood] = 1
    else:
        counts[neighborhood] += 1

total_price = {}
for x in airbnb_data:
    neighborhood = x['neighborhood']
    price = x['price']
    if neighborhood not in total_price:
        total_price[neighborhood] = price
    else:
        total_price[neighborhood] += price

results = {}
for neighborhood, prices in total_price.items():
        average_price = prices / counts[neighborhood]
        if counts == 0:
            results[neighborhood] = 0
        else:
            results[neighborhood] = average_price
print(results['TS17'])
# Well done
```

```
95.5672514619883
```

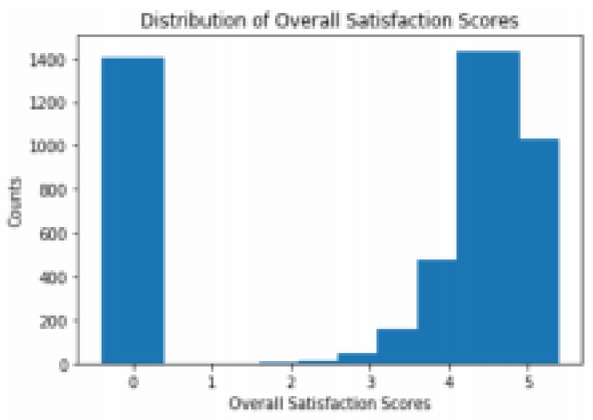
Q4. Plot a distribution of counts of the overall_satisfaction [10 marks]

Note: You should filter out listings whose reviews are 0.

Hint
Counting with dictionaries

1. *Iterate over the dictionary (using `.items()`)*
2. *Using what you get in each iteration, populate 2 lists:*
   - *1 list for all the scores labels*
   - *1 list for all the counts*
3. *Plot with the 2 lists </i>*

Expected Output:

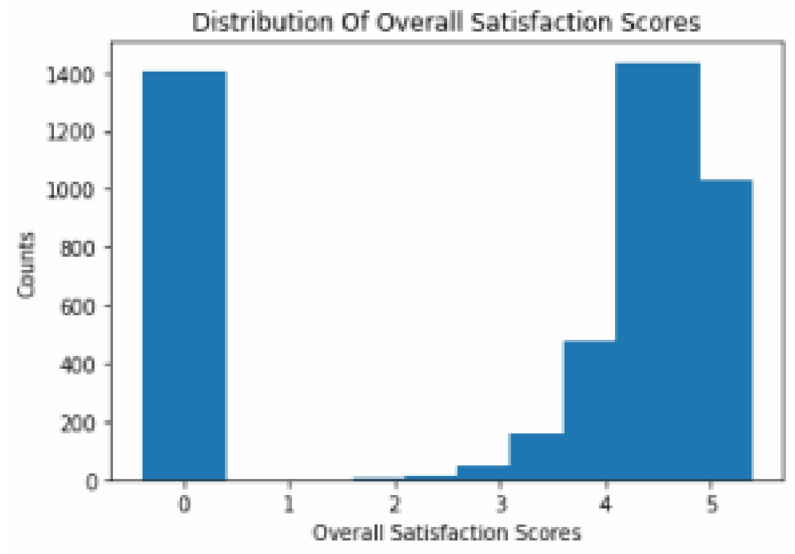


In [7]:

```
import matplotlib.pyplot as plt
scores_counts = {}
for i in airbnb_data:
    overall_satisfaction_scores = i['overall_satisfaction']
    reviews = i['reviews']
    if reviews != 0:
        if overall_satisfaction_scores not in scores_counts:
            scores_counts[overall_satisfaction_scores] = 1
        else:
            scores_counts[overall_satisfaction_scores] += 1

x = []
y = []
for overall_satisfaction_score, counts in scores_counts.items():
    x.append(overall_satisfaction_score)
    y.append(counts)
plt.title('Distribution Of Overall Satisfaction Scores')
plt.xlabel('Overall Satisfaction Scores')
plt.ylabel('Counts')
plt.bar(x, y)
# Well done
```

Out[7]:

```
<BarContainer object of 10 artists>
```

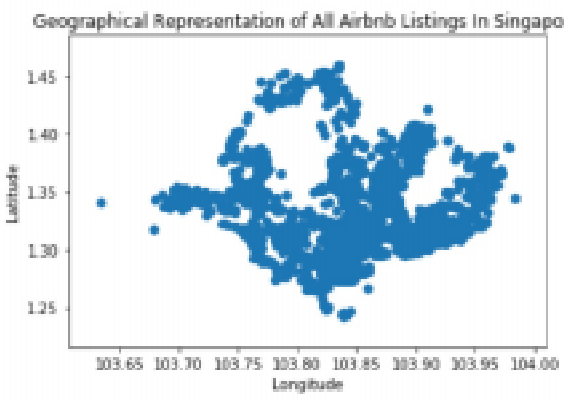


Q5. Plot a geographical representation of all of the listings in Singapore [10 marks]

Hint

1. *Create a list for latitude*
2. *Create a list for longitude*
3. *Append each listing's latitude and logitude to the lists*
4. *Plot a scatter plot using both lists </i>*

Expected Output:

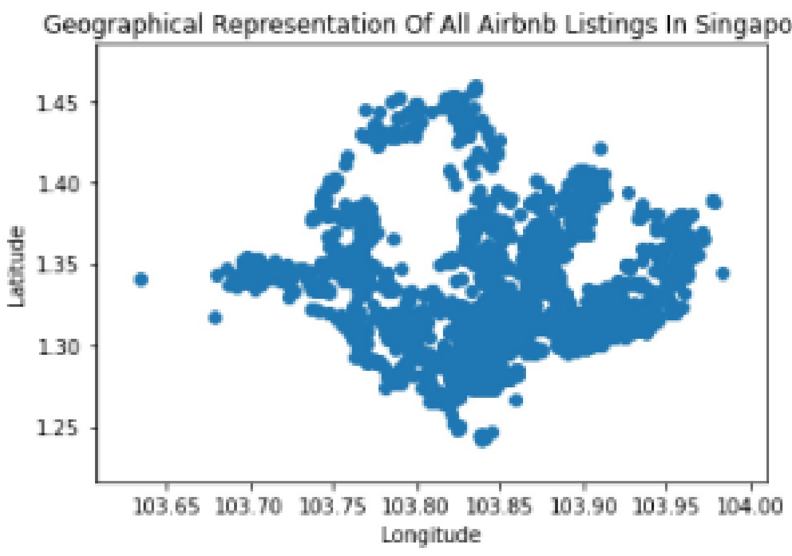


In [8]:

```
import matplotlib.pyplot as plt
latitudes_list = []
longitudes_list = []
for i in airbnb_data:
    latitude = i['latitude']
    longitude = i['longitude']
    latitudes_list.append(latitude)
    longitudes_list.append(longitude)
x = longitudes_list
y = latitudes_list
plt.xlabel('Longitude')
plt.ylabel('Latitude')
plt.scatter(x,y)
plt.title('Geographical Representation Of All Airbnb Listings In
Singapore') # Well done
```

Out[8]:

```
Text(0.5, 1.0, 'Geographical Representation Of All Airbnb Listings In
```



```
Singapore')
```

AirBnB Visualisation and Price Recommender App ***Attempts to create the functions are awarded 2 marks each Scenario:*** *Based on the earlier EDA, we have seen that it is not modular and does not allow the AirBnB team to look into each neighborhood.*

Nevertheless, the AirBnB data team have tasked you to build a simple application to improve the earlier EDA while serving its 2 users: Guests and Hosts. ***Your objective:*** *Develop an* ***app*** *which will serve the 2 main users: 1. Guests - Visualisation tool to recommend them the best listings based on price and overall satisfaction score in a neighborhood 2. Hosts - Recommend a price to set for their listing in a given neighborhood based on better performing listings*

THIS NEXT CELL IS IMPORTANT FOR YOUR APPLICATION. *Run it to install this package called* ***mplleaflet****. How do you know if you installed the library correctly? Try running the cell after this one (not the line that says "!pip install mplleaflet", its the other one), if you don't get an error, you are good to go! If you face any issues, please contact any of your TAs or Instructors.*

In [9]:

```
!pip install mplleaflet
```

```
Requirement already satisfied: mplleaflet in
c:\users\admin\onedrive\anacondapython3.7ve rsion\lib\site-packages (0.0.5)
Requirement already satisfied: jinja2 in
c:\users\admin\onedrive\anacondapython3.7versio n\lib\site-packages (from mplleaflet)
(2.10.1)
Requirement already satisfied: six in
c:\users\admin\onedrive\anacondapython3.7version\l ib\site-packages (from mplleaflet)
(1.12.0)
Requirement already satisfied: MarkupSafe>=0.23 in
c:\users\admin\onedrive\anacondapytho n3.7version\lib\site-packages (from
jinja2->mplleaflet) (1.1.1)
```

How do you know if you installed the library correctly? Try running the next cell, if you don't get an error, you are good to go!

In [10]:

```
import mplleaflet
```

Building the App

To begin building the App, there are 2 things to do:

1. *Building the functions*
2. *Testing the functions*

After we are done building the functions in part 1, we will test them in part 2

Every single function you create must have the ***airbnb_data*** *variable as the* ***first*** *parameter so that you can use it inside the function.*

```
def example_function_1(data, x, y, ..): # << do not name as
airbnb_da ta again!
    for i in data:
        print(i)

# when using it.. notice that airbnb_data is placed first, followed
b y the other parameters
example_function_1(airbnb_data, some_x, some_y, ...)
```

There are a total of 5 functions:

1. `get_all_latitudes`
2. `get_all_longitudes`
3. `listings_recommender`
4. `price_recommender`
5. `visualise_listings`

In [11]:

```
# Running this cell shows no output. This is normal because the function has been
create d but not called/used.
def example_function(interesting_data):
    for i in interesting_data:
        print(i)
```

In [12]:

```
1
2
3
4
5
```

`get_all_latitudes()` - Functions to get all latitudes given a list of listing_ids [5 marks]

Hint

1. *Create a results list*
2. *Extract the latitude and listing id of each row*
3. *Check listing id exists within the all listings*
4. *If true, append the latitude into the results list*
5. *Return results list </i>*

***Input**: `airbnb_data` as `data` , a `list` of listing_ids*

***Return**: A `list` of latitudes*

In [13]:

```
# Write your code below:
latitudes = []
def get_all_latitude(airbnb_data, all_listing_ids):
    for rows in airbnb_data:
        latitude = rows['latitude']
        listing_id = rows['listing_id']
        if listing_id in all_listing_ids:
            latitudes.append(latitude)
    print(latitudes)
# Well done
```

Tester Cell *- To test the above function to see if it's working.*

Expected Output:

```
[1.305702, 1.296138, 1.304393]
```

In [14]:

```
get_all_latitude(airbnb_data, ['10350448','13507262','13642646'])
```

```
[1.305702, 1.296138, 1.304393]
```

`get_all_longitudes()` - Functions to get all longitudes given a list of listing_ids [5 marks]

Hint

1. *Same as previous question, just that it's about longitudes now </i>*

Input: `airbnb_data` *as* `data` *, a* `list` *of listing_ids*

Return: *A* `list` *of longitudes*

In [15]:

```
# Write your code below:
longitudes = []
def get_all_longitude(airbnb_data, all_listing_ids):
    for rows in airbnb_data:
        longitude = rows['longitude']
        listing_id = rows['listing_id']
        if listing_id in all_listing_ids:
            longitudes.append(longitude)
    print(longitudes)
# Well done
```

Tester Cell *- To test the above function to see if it's working.*

Expected Output:

```
[103.79878, 103.767841, 103.784174]
```

In [16]:

```
get_all_longitude(airbnb_data, ['10350448','13507262','13642646'])
```

```
[103.79878, 103.767841, 103.784174]
```

***listings_recommender()* - Function to recommend all listings based on a given price, satisfaction score and neighborhood [15 marks]**

Note:

1. *Lesser than or equal to that price*
2. *Equal or more than that overall satisfaction score*
3. *In that neighborhood*

Hint

1. *Create a results list*
2. *Extract the relevant of each row*
3. *Check it satisfies all conditions passed into the function*
4. *If true, append the listing id into the results list*
5. *Return results list </i>*

***Input**: `airbnb_data` as `data` , price, overall_satisfaction, neighborhood_id*

***Return**: A `list` of listing_ids*

In [17]:

```
# Write your code below:
listings_list = []
def
    listings_recommender(airbnb_data,price,overall_satisfaction,neighborhood_i
    d): for rows in airbnb_data:
        prices = rows['price']
        overall_satisfaction_score = rows['overall_satisfaction']
        neighborhood = rows['neighborhood']
        listings_id = rows['listing_id']
        if neighborhood == neighborhood_id:
            if prices <= price and overall_satisfaction_score >=
                overall_satisfaction: listings_list.append(listings_id)
    print(listings_list)
# Well done
```

***Tester Cell** - To test the above function to see if it's working.*

Expected Output:

```
['10350448',
 '13507262',
 '13642646',
 '15099645',
 '6451493',
 '4696031',
 '2898794',
 '13181050',
 '9022211',
 '5200263',
 '6529707',
 '14433262']
```

In [18]:

```
listings_recommender(airbnb_data, 60, 5, 'MK03')
```

```
['10350448', '13507262', '13642646', '15099645', '6451493', '4696031', '2898794',
'13181 050', '9022211', '5200263', '6529707', '14433262']
```

`price_recommender()` - Function to recommend a price in a neighborhood based on average price and overall satisfaction [15 marks]

*For this function, we want to build a **simple** price recommendation function that will give a potential host a suggested price.*

To build this, these are the requirements:

1. *Take all listings in that neighborhood and check for listings with a least 1 review and an overall satisfaction score of 4 or more.*
2. *From that filtered listings, calculate the average price and return that as the suggested price rounded to 2 decimal places.*

***Input**: `airbnb_data` as `data` , a neighborhood_id*

***Return**: A `float` of recommended price*

In [19]:

```
# Write your code below:
filtered_neighborhood_listings = []
total_price = []
def price_recommender(airbnb_data,neighborhood_id):
    for rows in airbnb_data:
        reviews = rows['reviews']
        neighborhood_listings = rows['listing_id']
        overall_satisfaction_score = rows['overall_satisfaction']
        prices = rows['price']
        neighborhood = rows['neighborhood']
        if neighborhood == neighborhood_id:
            if float(reviews) >= 1.0 and float(overall_satisfaction_score) >=
            4.0: filtered_neighborhood_listings.append(neighborhood_listings)
                total_price.append(float(prices))
    total_price_value = 0
    for number in total_price:
        total_price_value += number
    average_price = sum(total_price)/len(total_price)
    print(average_price)

# round to 2 decimal!
```

***Tester Cell** - To test the above function to see if it's working.*

Expected Output:

```
66.28
```

In [20]:

```
price_recommender(airbnb_data, 'TS17')
```

```
66.28
```

`visualise_listings()` - Function to geographically visualise a given list of listings [15 marks]

NOTE

```
# To do any visualtion, the last part of the code you would normally
do this..
    # some code to prepare data
    # some code to design the visualisation
    plt.show() #<< last line in code to show visualisation
```

With the new mplleaflet package, you can do the same thing, but just change from `plt.show()` to `mplleaflet.show()` .

DO NOT DO THIS

```
for i in data:
        # some codes here
        mplleaflet.show() # << Do not put the .show() function
insid e the for loop

    mplleaflet.show()# .show() should exist outside your for loop
```

Hint

1. *Use the 2 functions you've created earlier and make 2 lists, latitude and longitude*
2. *Use .scatter() to plot the scatter plot*
3. *.show() the scatter plot </i>*

***Input**: `airbnb_data` as `data` , a list of listing_ids*

***Output**: Visualisation of locations the listings (nothing to return)*

In [21]:

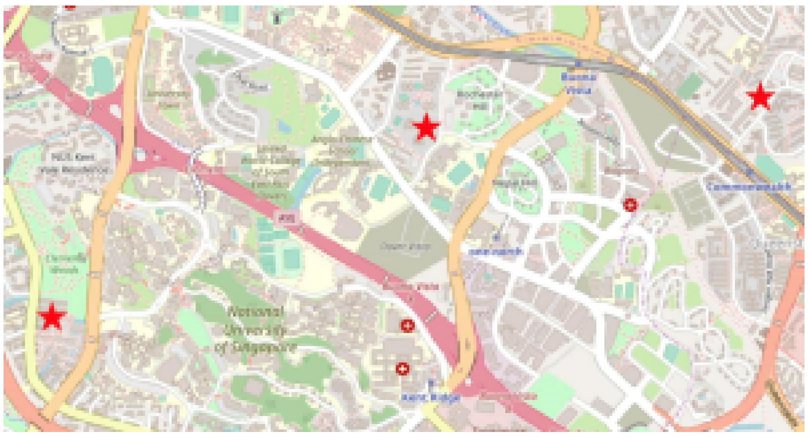
```
import mplleaflet
import matplotlib.pyplot as plt
def visualise_listings(airbnb_data, all_listing_ids):
    x = longitudes
    y = latitudes
    plt.scatter(x,y, c='red', alpha =1.0, marker = '*')
    mplleaflet.show()

# what is longitudes, what is latitudes?
```

***Tester Cell** - To test the above function to see if it's working.*

Expected Output:

Do not have to look exactly like this, as long as the locations are the same, it is fine!

If it's working, a new tab will pop out. This is normal.



In [22]:

```
visualise_listings(airbnb_data, ['10350448','13507262','13642646'])
```

Testing

Here, we will test if your functions are working as they are supposed to.

***Your task**: Use the functions created above, if necessary interchangeably, to answer the questions below.*

User - An Airbnb Host

Imagine now you're an Airbnb host and you are going to use the app you've developed to ask for a recommended price to list your place.

Based on your assigned neighborhood, what is the recommended price for your neighborhood [2.5 marks]

Expected output: 98.52

In [23]:

```
neighborhood_to_test = 'TS23'
filtered_neighborhood_listings = []
total_price = []
def price_recommender(airbnb_data,neighborhood_id):
    for rows in airbnb_data:
        reviews = rows['reviews']
        neighborhood_listings = rows['listing_id']
        overall_satisfaction_score = rows['overall_satisfaction']
        prices = rows['price']
        neighborhood = rows['neighborhood']
        if neighborhood == neighborhood_id:
            if float(reviews) >= 1.0 and float(overall_satisfaction_score) >=
            4.0: filtered_neighborhood_listings.append(neighborhood_listings)
                total_price.append(float(prices))
    total_price_value = 0
    for number in total_price:
        total_price_value += number
        average_price = total_price_value/len(total_price)
    print(average_price)
price_recommender(airbnb_data,neighborhood_to_test)

# 2 decimals!
```

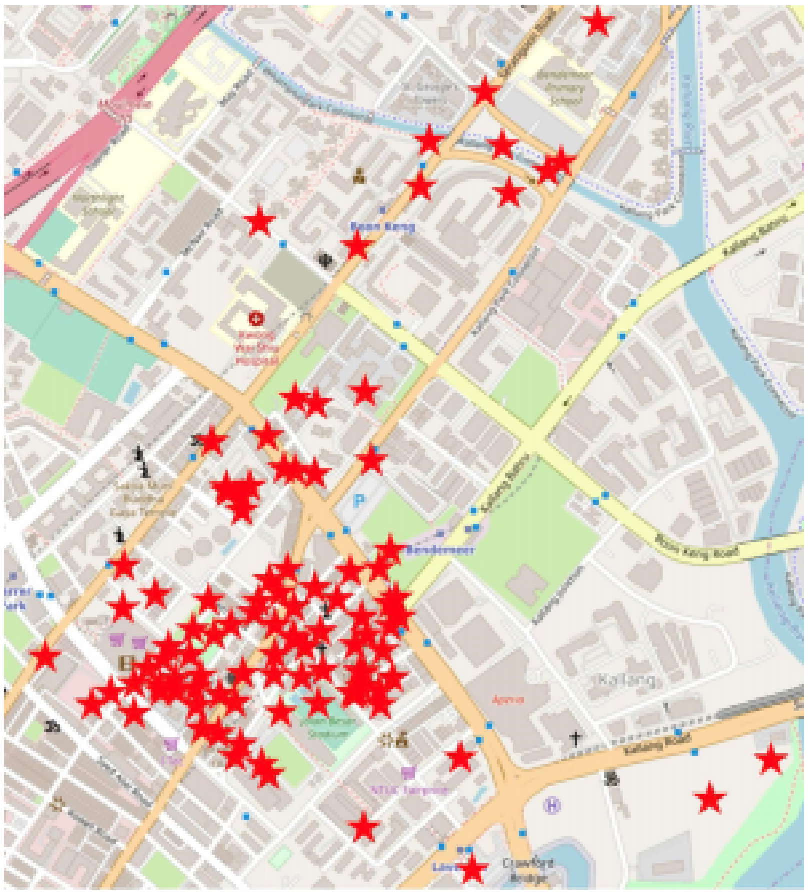
```
98.51515151515152
```

User - An Airbnb Guest

Imagine now you're an Airbnb guest and you are going to use the app to find a list of listings you want based on your search filter/restrictions.

Based on your assigned price, overall_satisfaction and neighborhood, using the functions created above and plot them out on a map [2.5 marks]

Expected output:



If it's working, a new tab will pop out. This is normal.

In [34]:

```python
import mplleaflet
import matplotlib.pyplot as plt
def visualise_all_listings(airbnb_data, all_listing_ids):
    neighborhood_to_test = 'TS17'
    price_to_test = 100
    overall_satisfaction_to_test = 4
    x = longitudes
    y = latitudes
    plt.scatter(x,y)
    mplleaflet.show()
visualise_all_listings(airbnb_data, all_listing_ids)

# what is latitudes, what is longitudes?
```