Part1 Keep Learning:

1. 5 of favorite website/blog/podcast/RSS/MOOC/Github related to data science/engineering:
https://www.codecademy.com/ I learn Python and SQL in this website.
https://www.datacamp.com/ I took some courses to learn data science, Python, R in this website.
https://www.w3schools.com/ I learn SQL and other programming languages in this website.
https://github.com/rhiever I follows Randy Olson Github to know more about data science.
https://www.analyticsvidhya.com/ This website helps me to learn more about data science.

2. 3 amazing papers:
Building Data Science Teams by DJ Patil
MapReduce: Simplified Data Processing on Large Clusters
Show and Tell: A Neural Image Caption Generator

3. The best book:
Introducing Python Modern Computing in Simple Packages By Bill Lubanovic
This book introduces Python and its data structures, which is very easy to understand and fun to read for a beginner. After reading this book, I master Python and become more interested in data science. In my opinion, the best way to learn a new technique is to practice. And this book includes some practices for testing, debugging, code reuse and some tips. So I really like this book. In addition, this book opens my door to learn data science/data engineering.

Part2 Exploratory Data Analysis

The following pictures shows the part result(id: 0-15) which selects 14 features of the table case_study_data_short_term_rentals.

	id	bedrooms	bathrooms	city	country	is_location_exact	lat	lng	price
0	13050414	4	2.0	Budapest	Hungary	True	47.48950	19.042300	753
1	13052442	1	1.0	Gothenburg	Sweden	False	57.69430	11.982700	113
2	13052871	1	2.0	Lisboa	Portugal	True	38.73940	-9.147540	23
3	13052889	1	3.0	Sankt-Peterburg	Russia	True	59.92790	30.375700	32
4	13052016	1	1.5	Pimville	South Africa	True	-26.27060	27.905100	30
5	13052768	1	1.0	Scuol	Switzerland	True	46.79540	10.296500	65
6	13050317	1	0.5	Eichstegen	Germany	True	47.93100	9.489100	65
7	13051379	2	1.0	Barcelona	Spain	True	41.37270	2.136450	1091
8	13050508	1	3.0	Ostuni	Italy	False	40.71970	17.578800	109
9	13052846	1	1.0	Warszawa	Poland	True	52.23110	21.084500	52
10	13052048	1	1.5	Tarifa	Spain	False	36.01430	-5.605900	16
11	13050541	1	1.0	Split	Croatia	True	43.51410	16.461100	55
12	13050963	4	2.0	Lavaur	France	True	43.68410	1.808910	98
13	13051168	1	1.0	Ingoldingen	Germany	True	48.03750	9.714280	48
14	13052201	1	1.0	Martina Franca	Italy	False	40.69370	17.348400	76
15	13051550	1	2.0	Offida	Italy	False	42.94080	13.766500	164

	description	picture_urls	picture_captions	star_rating	recent_review
0	We recommend this accomodation to large famili...	[u'https://a0.muscache.com/im/pictures/d69ffdb...	[u'terrace and view', u'terrace and view', u't...	-1.0	{u'review': {u'listing_id': 13050414, u'review...
1	Mitt boende rymmer par och familjer (med barn).	[u'https://a0.muscache.com/im/pictures/22b44d4...	[u'', u'', u'', u'', u'']	4.5	{u'review': {u'listing_id': 13052442, u'review...
2	Private room with a single bed in a shared apa...	[u'https://a0.muscache.com/im/pictures/007bd36...	[u'The room', u'The room', u'', u'', u'',...	2.5	{u'review': {u'listing_id': 13052871, u'review...
3	-1	[u'https://a0.muscache.com/im/pictures/04938d5...	[u'', u'', u'', u'', u'', u'', u'', u'', u'', ...	-1.0	-1
4	-1	[u'https://a0.muscache.com/im/pictures/36350d1...	[u'', u'', u'', u'', u'', u'', u'', u'', u'', ...	-1.0	-1
5	-1	[u'https://a0.muscache.com/im/pictures/88d9f58...	[u'Ausgangspunkt f\xfcr traumhafte Wanderungen...	-1.0	{u'review': {u'listing_id': 13052768, u'review...
6	-1	[u'https://a0.muscache.com/im/pictures/0659ccf...	[u'Chalet au milieu de la nature', u'La chambr...	5.0	{u'review': {u'listing_id': 13050317, u'review...
7	-1	[u'https://a0.muscache.com/im/pictures/44ceb58...	[u'Livingroom', u'Livingroom', u'Livingroom', ...	-1.0	-1
8	-1	[u'https://a0.muscache.com/im/pictures/377aa7b...	[u'', u'', u'', u'', u'', u'', u'', u'', u'ver...	-1.0	-1
9	-1	[u'https://a0.muscache.com/im/pictures/b9a4639...	[u'Pok\xf3j', u'Pok\xf3j', u'Przedpok\xf3j', u...	5.0	{u'review': {u'listing_id': 13052846, u'review...
10	-1	[u'https://a0.muscache.com/im/pictures/010bea2...	[u'', u'', u'girlsloveboards.com offers kite, ...	5.0	{u'review': {u'listing_id': 13052048, u'review...
11	My place is close to the city center, restaura...	[u'https://a0.muscache.com/im/pictures/87dfe6b...	[u'', u'', u'', u'', u'', u'', u'', u'', u'', ...	-1.0	{u'review': {u'listing_id': 13050541, u'review...
12	-1	[u'https://a0.muscache.com/im/pictures/e460263...	[u'Entr\xe9e ouvrant sur les diff\xe9rents esp...	-1.0	-1
13	Our appartement is just a few minutes away fro...	[u'https://a0.muscache.com/im/pictures/75a140f...	[u'Up the stairs: fully on your own', u'The fi...	-1.0	{u'review': {u'listing_id': 13051168, u'review...
14	Vicina a tutti i servizi immaginabili, climati...	[u'https://a0.muscache.com/im/pictures/2d6518e...	[u'', u'', u'', u'', u'', u'', u'']	-1.0	-1
15		[u'https://a0.muscache.com/im/pictures/6ffa1e8...	[u'']	-1.0	-1

1. The number of unique records is 919
The below picture shows unique records 919 rows * 14 columns.

```
907  {u'review': {u'listing_id': 13050525, u'review...
908                                                 -1
909  {u'review': {u'listing_id': 13051268, u'review...
910  {u'review': {u'listing_id': 13050721, u'review...
911  {u'review': {u'listing_id': 13051184, u'review...
912  {u'review': {u'listing_id': 13050346, u'review...
913  {u'review': {u'listing_id': 13052089, u'review...
914                                                 -1
915  {u'review': {u'listing_id': 13050315, u'review...
916  {u'review': {u'listing_id': 13052268, u'review...
917                                                 -1
918                                                 -1

[919 rows x 14 columns]
```

2. Bermuda has the highest median price for a one bedroom. And the highest median price is 240.0.
The below picture shows the median price for each country.

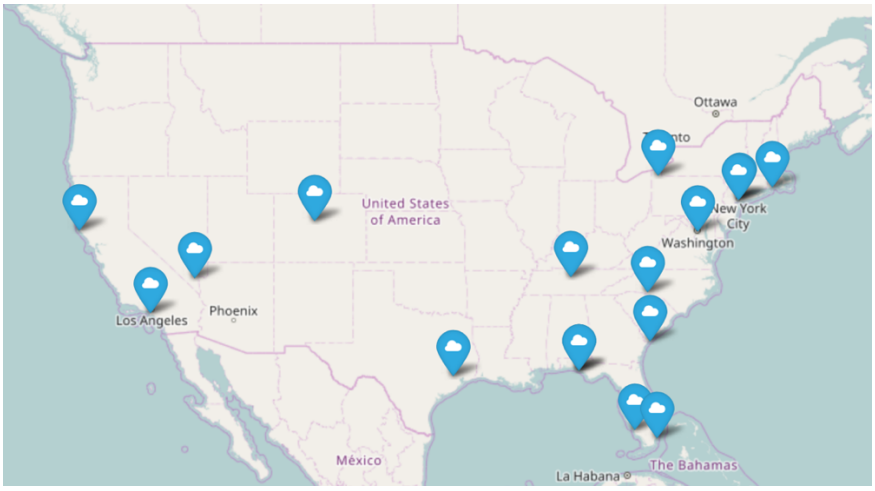
country	price
Armenia	60.000000
Australia	74.333333
Austria	66.800000
Belarus	14.000000
Belgium	43.000000
Bermuda	240.000000
Brazil	37.000000
Bulgaria	20.000000
Cape Verde	28.000000
China	47.857143
Colombia	25.000000
Croatia	60.187500
Cyprus	27.000000
Czech Republic	48.000000
Denmark	75.500000
Dominican Republic	95.000000
Estonia	72.666667
Finland	44.000000
France	62.647059
Germany	48.217391
Ghana	20.000000
Greece	133.166667
Guadeloupe	52.000000
Hong Kong	57.000000
Hungary	60.500000
Iceland	86.000000
India	28.500000

3. Listing 13050401 and listing 13050692 have the shortest distance
The below picture is the result of distance between two listings.

	listing_a	listing_b	distance
0	13050401	13050692	0.000065
1	13050316	13050367	0.000196
2	13050219	13051008	0.000242
3	13050014	13052697	0.000245
4	13050672	13052412	0.000248
5	13050329	13050698	0.000291
6	13050862	13051288	0.000331
7	13050518	13050914	0.000345
8	13050219	13051269	0.000374
9	13050316	13050323	0.000377
10	13050273	13050345	0.000388
11	13051587	13051985	0.000505
12	13050654	13052069	0.000522
13	13050323	13050367	0.000564
14	13050997	13051382	0.000587
15	13051128	13051382	0.000589
16	13051008	13051269	0.000613
17	13051159	13051189	0.000696
18	13051789	13051955	0.000721
19	13050403	13050769	0.000735
20	13051377	13051423	0.000779
21	13052419	13052819	0.000819
22	13051417	13051505	0.000823
23	13052146	13052596	0.000850
24	13050814	13051317	0.000872
25	13052596	13052868	0.000881
26	13051630	13052356	0.000922
27	13050198	13050231	0.000933

4. Visualize all US listings on a map

I use Folium to implement visualization on map.



5. Find out the total number of active listings
The below picture shows the total number of active listings is 919.

```
The number of unique records is 919
Bermuda has the highest median price for a one bedroom
Listing 13050401 and listing 13050692 have the shortest distance
The number of active listings is 919
```

The below picture shows the url of active listings.

```
https://www.airbnb.com/rooms/13050582 exists
https://www.airbnb.com/rooms/13050585 exists
https://www.airbnb.com/rooms/13050586 exists
https://www.airbnb.com/rooms/13050590 exists
https://www.airbnb.com/rooms/13050601 exists
https://www.airbnb.com/rooms/13050603 exists
https://www.airbnb.com/rooms/13050605 exists
https://www.airbnb.com/rooms/13050608 exists
https://www.airbnb.com/rooms/13050610 exists
https://www.airbnb.com/rooms/13050611 exists
https://www.airbnb.com/rooms/13050614 exists
https://www.airbnb.com/rooms/13050622 exists
https://www.airbnb.com/rooms/13050625 exists
https://www.airbnb.com/rooms/13050632 exists
https://www.airbnb.com/rooms/13050634 exists
https://www.airbnb.com/rooms/13050636 exists
https://www.airbnb.com/rooms/13050638 exists
https://www.airbnb.com/rooms/13050641 exists
https://www.airbnb.com/rooms/13050650 exists
```

I found a problem when I implemented this part. At first, I ran my program several times and the number of active listings was different. Then I printed the status code and its associated observations. And I found the reason. My program did not do any rest and tried to blast thousands of request to Airbnb server via URL, which would trigger their website safety mechanism which would identify my process as a virus attack or DDOS attack. Hence I got deny of access to their web service so that the number of active listings was different every time.

In the beginning, I used a random number generator to let my program sleep for a random time so that my program would not be blocked by the server. In this way, the server would less likely identify my program as a virus attach or DDOS (distributed denial-of-service) attack. But it need spend several hours to run, which was much time-consuming.
Then I try to use another approach. I use urllib this package to open URL and use urllib.HTTPError, urllib.URLError to catch error. It turns out that the running time is faster and it is not blocked by the server.

Part3: Fields of Interest:
I choose option3: web scraping.

1. Airbnb API is not publicly available. But I found some resources online and read other's JavaScript code on GitHub. Here's the link. https://github.com/phamtrisi/airapi Then I tried to use the check_availability_URL = 'https://www.airbnb.com/api/v2/calendar_months' to get calendar information.
So I wrote a function called web_scraping(conn) to test. But when I printed the status code, it showed 404, which means that the URL I found was not correct. And I didn't have enough time to debug about this question.

2. For automatic scalable, the key to scalability is to make program work in its own "data" by dividing the data set into smaller pieces and dispatch the work to different processes/threads.

Since the listing ID is an integer, we can leverage this column to build parallel processes.
Step 1: we need change the web scraping function signature as
def web_scraping (modfactor , mod)

Modfactor is the number of parallel processes we set.
Mod is the number we get by applying mod %.
(eg: 9 % 5 = 4 modfactor = 5, mod = 4)

Step 2: In the data preparation SQL, generate a SQL whose WHERE clause is listing_id % modfactor = mod
(eg: select * from case_study_data_short_term_rentals where id % 5 = 4)

Now the web scraping function will take two parameters to build a subset of data and work on its own data.

Step 3: We need write def main and get command line arguments like following:

```
import sys
def main():
    modfactor = sys.argv[1]
    mod = sys.arg[2]
    web_scraping (modfactor, mod)
```

Step 4: We write a shell script to call the program with python with appropriate parameters.
Sample: if factor is 5 then we can do

```
python mypythonscript.py 5,0  > factor5_mod0_output.txt 2>&1 &
python mypythonscript.py 5,1  > factor5_mod1_output.txt 2>&1 &
python mypythonscript.py 5,2  > factor5_mod2_output.txt 2>&1 &
python mypythonscript.py 5,3  > factor5_mod3_output.txt 2>&1 &
python mypythonscript.py 5,4  > factor5_mod4_output.txt 2>&1 &
```

The above approach will split work into different processes. And such approach can distribute work not only into different processes. Furthermore, such processes can be kicked off from different computers. One good use case is that this program can be kicked off by Hadoop-

streaming package so that the code can be executed on different nodes in a Hadoop cluster. (http://www.michael-noll.com/tutorials/writing-an-hadoop-mapreduce-program-in-python/)

Alternatively, we can make main function takes single parameter mod factor and use loop to kick off the web scraping function in threads. This approach will limit the execution scalability within a single computer.
Eh:

```
import threading
for mod in range(modfactor-1)
    t = threading.Thread(target=crawfunction(modfactor, mod))
    t.start()
    t.join
```

There are three ways to not to be blocked by other websites.
1. Let the program rest. We can use a random number generator to let our program to sleep for a random time. I implemented this in question5: total number of active listings. But this approach was time-consuming because my program need sleep every time.
2. Rotating IPs and Proxy server: we can use different IP addresses for making requests to a server, the detection becomes harder. Create a pool of IPs that we can use and use random ones for each request.
3. User-agent rotation and spoofing: every request made from a web browser contains a user-agent header and using the same user-agent consistently leads to the detection of a bot. We can spoof the user agent by creating a list of user agents and picking a random one for each request. Also we set user-agent to a common web browser instead of using the default user-agent.