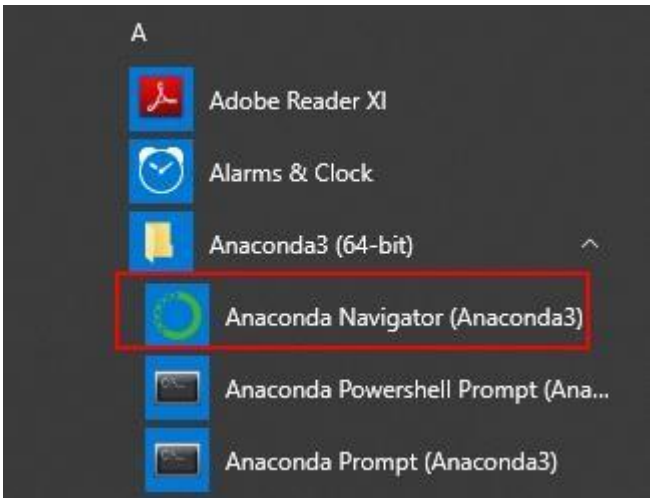
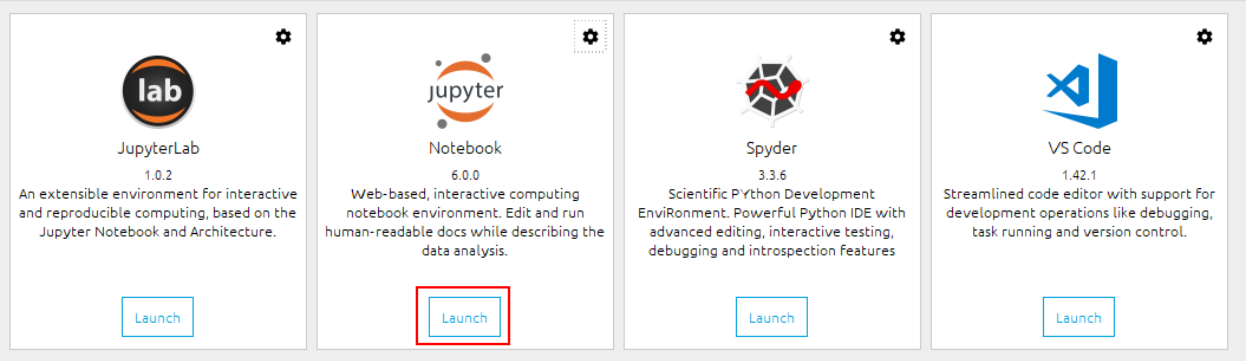
Module 7: Hands-On: 5

Data Cleaning.

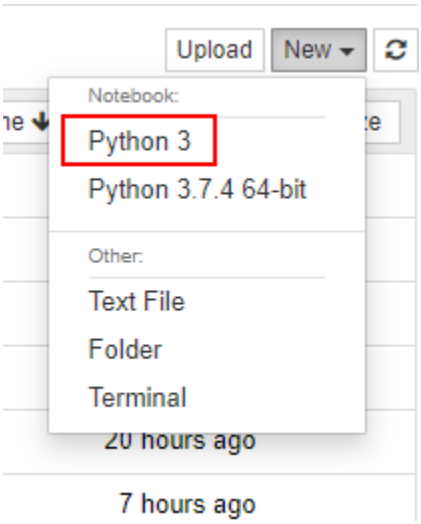
Step 1: Open Anaconda Navigator



Step 2: Click on Launch button under jupyter notebooks.



Step 3: After the notebook opens click on new and Python 3.



Step 4: Import the required packages and read data from patient.csv in a dataframe.

```
In [6]: import pandas as pd
        import numpy as np

In [11]: data = pd.read_csv('patient.csv')

In [12]: data.head()
```

Out[12]:

	id	sex	birth_year	country	region	group	infection_reason	infection_order	infected_by	contact_number	confirmed_date	released_date	deceased_dat
0	1	female	1984.0	China	filtered at airport	NaN	visit to Wuhan	1.0	NaN	45.0	2020-01-20	2020-02-06	Nal
1	2	male	1964.0	Korea	filtered at airport	NaN	visit to Wuhan	1.0	NaN	75.0	2020-01-24	2020-02-05	Nal
2	3	male	1966.0	Korea	capital area	NaN	visit to Wuhan	1.0	NaN	16.0	2020-01-26	2020-02-12	Nal
3	4	male	1964.0	Korea	capital area	NaN	visit to Wuhan	1.0	NaN	95.0	2020-01-27	2020-02-09	Nal
4	5	male	1987.0	Korea	capital area	NaN	visit to Wuhan	1.0	NaN	31.0	2020-01-30	NaN	Nal

Step 5: Take a look at the percentage of null values in each column.

```
In [13]: data.isnull().sum() / data.shape[0]

Out[13]: id                 0.000000
         sex                0.924501
         birth_year         0.930674
         country            0.000000
         region             0.927588
         group              0.981956
         infection_reason   0.969136
         infection_order    0.991690
         infected_by        0.985280
         contact_number     0.992403
         confirmed_date     0.000000
         released_date      0.993352
         deceased_date      0.996914
         state              0.000000
         dtype: float64
```

Step 6: Replace every occurrence of 0, empty string and NULL with np.nan.

```
In [14]: data.replace(to_replace=['0', ' ', 'NULL'], value=np.nan, inplace=True)
```

Step 7: Extract all numeric data and check amount of null values.

```
In [16]: numeric_data = data.select_dtypes(exclude=['object'])

In [18]: numeric_data.isnull().sum()

Out[18]: id                  0
         birth_year       3920
         infection_order  4177
         infected_by      4150
         contact_number   4180
         dtype: int64
```

Step 8: Drop every row with null values and check the shape of data after that.

```
In [24]: not_na_data = numeric_data.dropna()

In [21]: not_na_data.shape
Out[21]: (15, 5)
```

Step 9: Drop every column with null values and check the shape of data after that.

```
In [25]: numeric_data.dropna(axis=1).head()
```

Out[25]:

	id
0	1
1	2
2	3
3	4
4	5

Step 10: Fill every null value with 0 and take a look at the head of data.

```
In [31]: numeric_data.fillna(0).head()
```

Out[31]:

	id	birth_year	infection_order	infected_by	contact_number
0	1	1984.0	1.0	0.0	45.0
1	2	1964.0	1.0	0.0	75.0
2	3	1966.0	1.0	0.0	16.0
3	4	1964.0	1.0	0.0	95.0
4	5	1987.0	1.0	0.0	31.0

Step 11: Fill every null value with mean of that column and take a look at the number of null values after that.

```
In [29]: mean_filled = numeric_data.fillna(numeric_data.mean())

In [30]: mean_filled.isnull().sum()

Out[30]: id                0
         birth_year        0
         infection_order   0
         infected_by       0
         contact_number    0
         dtype: int64
```