##Hands-on Activity 7.1 Data Collection and Wrangling

Intended Learning Outcomes:

Demonstrate how to gather sensor data, image data and voice data Demonstrate how to gather data (text and images) from web Demonstrate how to prepare data using different data preprocessing techniques

Resources:

Personal Computer Jupyter Notebook Internet Connection

Instruction:

Download the following datasets: Download aapl.csv, amzn.csv Download amzn.csv, fb.csv Download fb.csv, goog.csv Download goog.csv, nflx.csv Download nflx.csv Accomplish the notebook for this activity and submit as a pdf file: Data Wrangling HOA.pdf

Excercise 1

- Read Each file in
- Add a column to each dataframe, called ticker, indicating the ticket symbol it is or (Apple's is AAPL, or example). This is how you look up a stock. Each file's name is also the ticker symbol, so be sue to capitalize it.
- Append them together into a single dataframe.
- Save the result in a CSV file called faan.csv

```
import pandas as pd
import os

directory = '/content/'
file_names = ['aapl.csv', 'amzn.csv', 'fb.csv', 'goog.csv',
'nflx.csv']

dataframes = []

for file_name in file_names:
    ticker = os.path.splitext(file_name)[0].upper()

    file_path = os.path.join(directory, file_name)

    df = pd.read_csv(file_path)
    df['ticker'] = ticker

    dataframes.append(df)

combined_df = pd.concat(dataframes, ignore_index=True)
```

```
combined_df.to_csv('/content/faan.csv', index=False)
```

Exercise 2

- With faang, use type conversion to change the date column into a datetime and the volume column into integers. Then, sort by date and ticker.
- Find the seven rows with the highest value for volume.
- Right now, the data is somewhere between long and wide format. Use melt() to make it completely long format. Hint: date and ticker are our ID variables (they uniquely identify each row). We need to melt the rest so that we don't have separate columns for open, high, low, close, and volume.

```
faang = pd.read_csv('/content/faan.csv')
faang['date'] = pd.to_datetime(faang['date'])
faang['volume'] = faang['volume'].astype(int)
faang.sort_values(by=['date', 'ticker'], inplace=True)

HighestValue = faang.nlargest(7, 'volume')

print(HighestValue)

          date       open       high        low      close     volume
ticker
644 2018-07-26   174.8900   180.1300   173.7500   176.2600  169803668
FB
555 2018-03-20   167.4700   170.2000   161.9500   168.1500  129851768
FB
559 2018-03-26   160.8200   161.1000   149.0200   160.0600  126116634
FB
556 2018-03-21   164.8000   173.4000   163.3000   169.3900  106598834
FB
182 2018-09-21   219.0727   219.6482   215.6097   215.9768   96246748
AAPL
245 2018-12-21   156.1901   157.4845   148.9909   150.0862   95744384
AAPL
212 2018-11-02   207.9295   211.9978   203.8414   205.8755   91328654
AAPL

flong = pd.melt(faang, id_vars=['date', 'ticker'], value_vars=['open',
'high', 'low', 'close', 'volume'], var_name='attribute',
value_name='value')

print(flong)

           date ticker attribute         value
0    2018-01-02   AAPL      open  1.669271e+02
1    2018-01-02   AMZN      open  1.172000e+03
2    2018-01-02     FB      open  1.776800e+02
```

```
3    2018-01-02   GOOG      open  1.048340e+03
4    2018-01-02   NFLX      open  1.961000e+02
...         ...    ...       ...           ...
6270 2018-12-31   AAPL    volume  3.500347e+07
6271 2018-12-31   AMZN    volume  6.954507e+06
6272 2018-12-31     FB    volume  2.462531e+07
6273 2018-12-31   GOOG    volume  1.493722e+06
6274 2018-12-31   NFLX    volume  1.350892e+07

[6275 rows x 4 columns]
```

Exercise 3

- Using web scraping, search for the list of the hospitals, their address and contact information. Save the list in a new csv file, hospitals.csv.
- Using the generated hospitals.csv, convert the csv file into pandas dataframe. Prepare the data using the necessary preprocessing techniques.

```
import requests as req
from bs4 import BeautifulSoup
import pandas as pd

url =
'https://en.wikipedia.org/wiki/List_of_hospitals_in_the_Philippines'
reqq = req.get(url)
soup = BeautifulSoup(reqq.content, 'html.parser')
tab = soup.find('table')
df = pd.read_html(str(tab), header=0)[0]
df.to_csv('hospitals.csv', index=False)
read = pd.read_csv("hospitals.csv")

read
```

```
{"summary":"{\n  \"name\": \"read\",\n  \"rows\": 49,\n  \"fields\":
[\n    {\n      \"column\": \"Name of Hospital\",\n
\"properties\": {\n        \"dtype\": \"string\",\n
\"num_unique_values\": 49,\n        \"samples\": [\n          \"Manila
Naval Hospital\",\n          \"Lung Center of the Philippines\",\n
\"Quirino Memorial Medical Center\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Location\",\n      \"properties\":
{\n        \"dtype\": \"string\",\n        \"num_unique_values\": 48,\
n        \"samples\": [\n          \"Taft Avenue, Ermita, Manila\",\n
\"V. Luna Road, Quezon City\",\n          \"Honorio Lopez Boulevard.,
Balut, Tondo, Manila\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Class\",\n      \"properties\": {\n        \"dtype\": \"category\",\
n        \"num_unique_values\": 8,\n        \"samples\": [\n
\"DOH Retained\",\n          \"University\",\n          \"LGU\"\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    }\n  ]\n}","type":"dataframe","variable_name":"read"}
```

```python
df = pd.read_csv('hospitals.csv')

print("Original DataFrame:")
print(df.head())


# 1. Handle missing values
missing_values = df.isnull().sum()
print("\nMissing Values:")
print(missing_values)

# Fill missing values with appropriate values
df['location'].fillna('Unknown', inplace=True)

# Drop duplicate rows
df.drop_duplicates(inplace=True)

# Extracting area or city from the Location column
df['Area'] = df['location'].str.split(',').str[0].str.strip()


print("\nPreprocessed DataFrame:")
print(df.head())
```

```
Original DataFrame:
                               name_of_hospital  \
0                  Caloocan City Medical Center   
1                            Ospital ng Malabon   
2             San Lorenzo Ruiz General Hospital   
3  Gat Andres Bonifacio Memorial Medical Center   
4                              Ospital ng Tondo   

                                             location         class  
0                    450 A. Mabini St., Caloocan City           LGU  
1          F. Sevilla Boulevard, Tañong, Malabon City           LGU  
2  O. Reyes St., Rosita Subdivision, Santulan, Ma...  DOH Retained  
3                      8001 Delpan St., Tondo, Manila           LGU  
4              Jose Abad Santos Avenue, Tondo, Manila           LGU  

Missing Values:
name_of_hospital    0
location            0
class               0
dtype: int64

Preprocessed DataFrame:
                               name_of_hospital  \
0                 Caloocan City Medical Center
1                           Ospital ng Malabon
2            San Lorenzo Ruiz General Hospital
3  Gat Andres Bonifacio Memorial Medical Center
4                             Ospital ng Tondo

                                            location         class  \
0                   450 A. Mabini St., Caloocan City           LGU   
1         F. Sevilla Boulevard, Tañong, Malabon City           LGU   
2  O. Reyes St., Rosita Subdivision, Santulan, Ma...  DOH Retained   
3                     8001 Delpan St., Tondo, Manila           LGU   
4             Jose Abad Santos Avenue, Tondo, Manila           LGU   

                      Area
0        450 A. Mabini St.
1     F. Sevilla Boulevard
2             O. Reyes St.
3          8001 Delpan St.
4  Jose Abad Santos Avenue
```