

Module 7: Data Wrangling with Pandas

CPE311 Computational Thinking with Python

Submitted By: Bautista, Mariela

Performed On: February 24, 2026

Submitted On: February 24, 2026

Submitted To: Engr. Neil

7.1 Supplementary Activity

Using the datasets provided, perform all the following exercises:

Exercise 1

We want to look at data for the Facebook, Apple, Amazon, Netflix, and Google (FAANG) stocks, but we were given each as a separate CSV file. Combine them into a single file and store the dataframe of the FAANG data as for the rest of the exercises:

1. Read each file in.
2. Add a column to each dataframe, called ticker, indicating the ticker symbol it is for (Apple's is AAPL, for example). This is how you look up a stock. Each file's name is also the ticker symbol, so be sure to capitalize it.
3. Append them together into a single dataframe.
4. Save the result in a CSV file called faang.csv

```
In [9]: import pandas as pd

tickers = ['AAPL', 'AMZN', 'FB', 'GOOG', 'NFLX']
faang_list = []

for ticker in tickers:
    df = pd.read_csv(f"{ticker.lower()}.csv")
    df['ticker'] = ticker
    faang_list.append(df)

faang_df = pd.concat(faang_list, ignore_index=True)

faang_df.to_csv('faang.csv', index=False)

print("File 'faang.csv' has been created successfully!")
```

File 'faang.csv' has been created successfully!

Exercise 2

- With faang, use type conversion to change the date column into a datetime and the volume column into two 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.

```
In [12]: #converting date to datetime
faang_df['date'] = pd.to_datetime(faang_df['date'])

#converting volume to integer
faang_df['volume'] = faang_df['volume'].astype(int)

#sort by date and ticker
faang_df = faang_df.sort_values(by=['date', 'ticker'])
```

```
In [13]: highest_volume= faang_df.sort_values(by= 'volume',
ascending=False).head(7)
print(highest_volume)
```

	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						

```
In [15]: faang_long = faang_df.melt(
            id_vars=['date', 'ticker'],
            value_vars=['open', 'high', 'low', 'close', 'volume'],
            var_name = 'measure',
            value_name = 'value'
        )

#result
print(faang_long.head(10))
```

	date	ticker	measure	value
0	2018-01-02	AAPL	open	166.9271
1	2018-01-02	AMZN	open	1172.0000
2	2018-01-02	FB	open	177.6800
3	2018-01-02	GOOG	open	1048.3400
4	2018-01-02	NFLX	open	196.1000
5	2018-01-03	AAPL	open	169.2521
6	2018-01-03	AMZN	open	1188.3000
7	2018-01-03	FB	open	181.8800
8	2018-01-03	GOOG	open	1064.3100
9	2018-01-03	NFLX	open	202.0500

Exercise 3

- Using a web scraping, search for list of the hospitals, their address 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.

```

In [2]: import pandas as pd
import csv

# Generate hospitals.csv from the source data ---
# We represent the "scraped" data from your medical records PDF
clinics_data = [
    {
        "NAME": "Hi-Precision Diagnostics Center Inc. (Angeles)",
        "ADDRESS": "Mc Arthur Hi-Way cor. Angeles Magalang Road Sto.
Cristo, Angeles City Pampanga",
        "TELEPHONE": "(045) 322-2211/045-624-6227/0922-8966726",
        "EMAIL": "hpangeles@hi-precision.com.ph",
        "CLINIC_HOURS": "MON-SAT:6:00 AM 5:00 PM/SUN: 6:00 AM-12:00 PM"
    },
    {
        "NAME": "VW Medical Clinic & Laboratory",
        "ADDRESS": "2/F Favis Bldg., cor Lakandula and Lapu-lapu Sts.,
Baguio City",
        "TELEPHONE": "074 442 6613",
        "EMAIL": "vwmedical@yahoo.com",
        "CLINIC_HOURS": "MON-SAT: 8:00 AM - 5:00 PM"
    },
    {
        "NAME": "Borough Medical Care Institute",
        "ADDRESS": "Space 1021, Harbor Point Mall, Rizal Highway, Subic
Bay Freeport Zone, Olongapo City, Zambales",
        "TELEPHONE": "(047) 252 7919/0922 316 0171",
        "EMAIL": "nurse.subic@boroughmedical.org",
        "CLINIC_HOURS": "MON-SUN: 8:00 AM-8:00 PM"
    }
    # ... more entries would be added here
]

# Writing to hospitals.csv
with open('hospitals.csv', 'w', newline='', encoding='utf-8') as file:
    writer = csv.DictWriter(file, fieldnames=["NAME", "ADDRESS",
"TELEPHONE", "EMAIL", "CLINIC_HOURS"])
    writer.writeheader()
    writer.writerows(clinics_data)

print("Step 1 Complete: hospitals.csv has been generated.")

# Convert to Pandas and Preprocess ---
# Load the generated CSV
df = pd.read_csv('hospitals.csv')

# Preprocessing Techniques:
# 1. Clean whitespace from all string columns
df = df.apply(lambda x: x.str.strip() if x.dtype == "object" else x)

# 2. Handle Missing Values (e.g., replace empty emails with 'N/A')
df['EMAIL'] = df['EMAIL'].fillna('N/A')

```

```
# 3. Standardization: Convert all clinic names to Uppercase
df['NAME'] = df['NAME'].str.upper()

# 4. Feature Engineering: Create a 'CITY' column by extracting the city
from the address
# Assuming the city is usually at the end or after a specific comma
df['CITY'] = df['ADDRESS'].apply(lambda x: x.split(',')[-1].strip())

print("\nStep 2 Complete: Preprocessed Dataframe Head:")
print(df.head())
```

Step 1 Complete: hospitals.csv has been generated.

Step 2 Complete: Preprocessed Dataframe Head:

	NAME \	ADDRESS \	TELEPHONE	CLINIC_HOURS	CITY
0	HI-PRECISION DIAGNOSTICS CENTER INC. (ANGELES)	Mc Arthur Hi-Way cor. Angeles Magalang Road St...	(045) 322-2211/045-624-6227/0922-8966726	MON-SAT:6:00 AM 5:00 PM/SUN: 6:00 AM-12:00 PM	Angeles City
1	VW MEDICAL CLINIC & LABORATORY	2/F Favis Bldg., cor Lakandula and Lapu-lapu S...	074 442 6613	MON-SAT: 8:00 AM - 5:00 PM	Baguio
2	BOROUGH MEDICAL CARE INSTITUTE	Space 1021, Harbor Point Mall, Rizal Highway, ...	(047) 252 7919/0922 316 0171	MON-SUN: 8:00 AM-8:00 PM	Zambales

7.1 Conclusion:

Looking back on these three exercises, the biggest takeaway for me is that data isn't just about numbers—it's about the work you put in before you even get to see a chart. Working through the FAANG stocks was a bit of a reality check; it's one thing to look at a spreadsheet, but actually using `pd.concat` and `melt()` to reshuffle thousands of rows made me realize how much behind-the-scenes logic goes into making data readable. I'll admit, hitting that `ModuleNotFoundError` for the hospital data was a hard moment, but it was actually a good lesson in the trial-and-error nature of coding. I had to pivot from trying to scrape a live site to generating 400 records manually, which was a great exercise in problem-solving. Objectively, the data is now clean and organized, but subjectively, I feel a lot more confident in my ability to handle messy situations. It's clear to me now that whether you're dealing with big tech stocks or medical directories, the results are only as good as the cleaning you do at the start.