

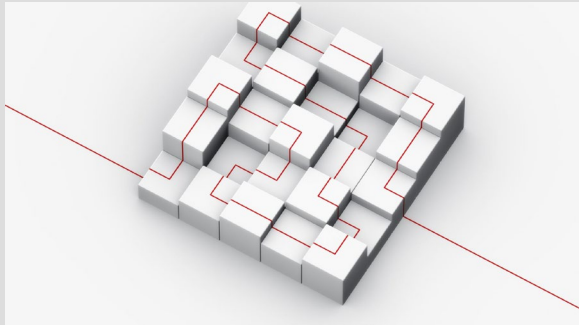



Developing a Project Using Python

John O'Leary

Objectives

- This project involves several steps including defining the input, processing, and output of a program.
- Create a flowchart to plan the steps of the process.
- Create a program that uses decisions and iteration with logical and relational expressions to process data.
- The project required installing and utilizing software including Anaconda, Spyder, and Linux.





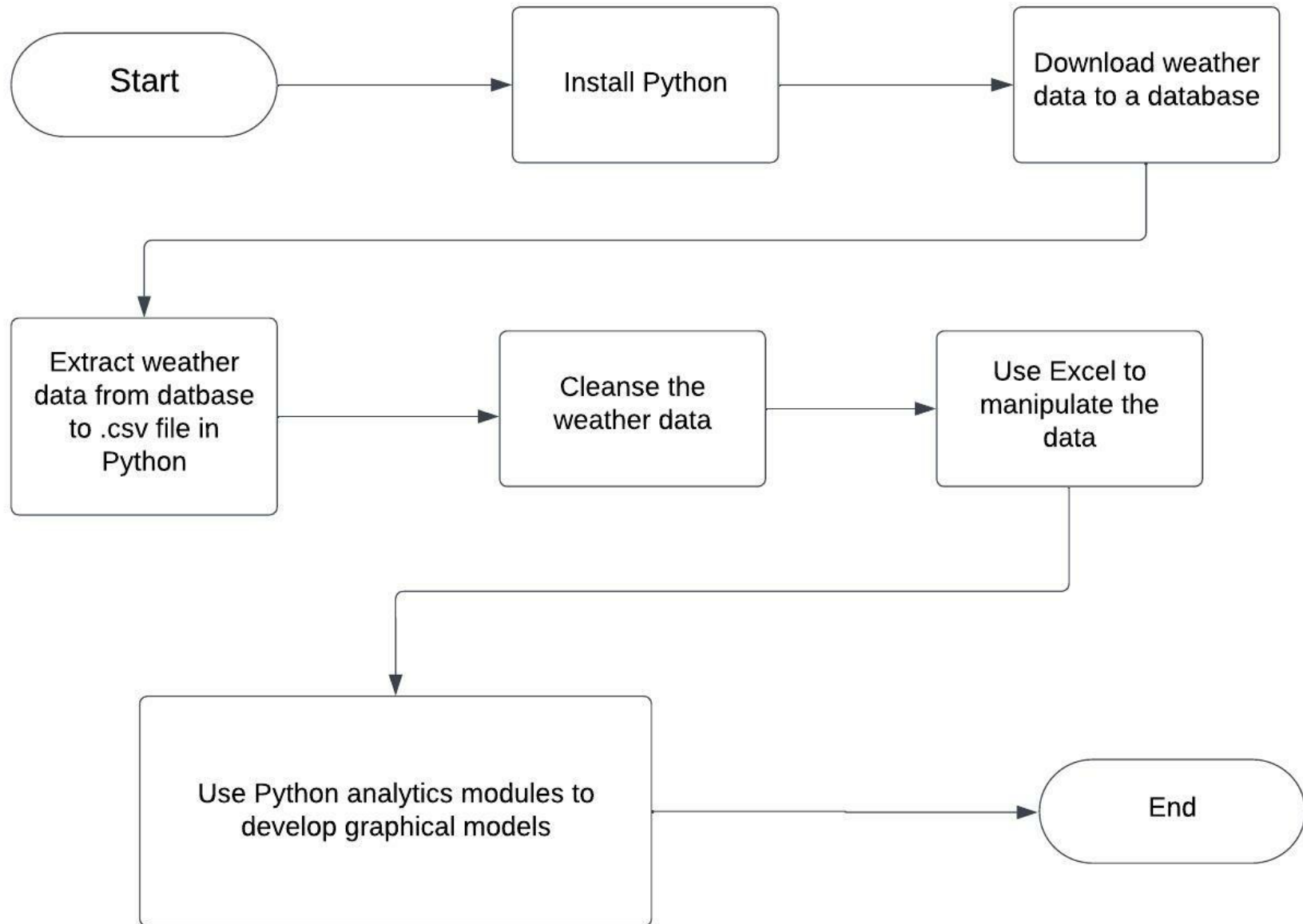
The initial steps to building a program in any programming language are first:

- Planning the steps required to accomplish the objectives. A common form of planning is the use of flowcharts.
- Once a plan is developed, the next step is to retrieve the data needed to be processed.

Using a Flowchart

The following slide contains a flowchart which includes the following processes:

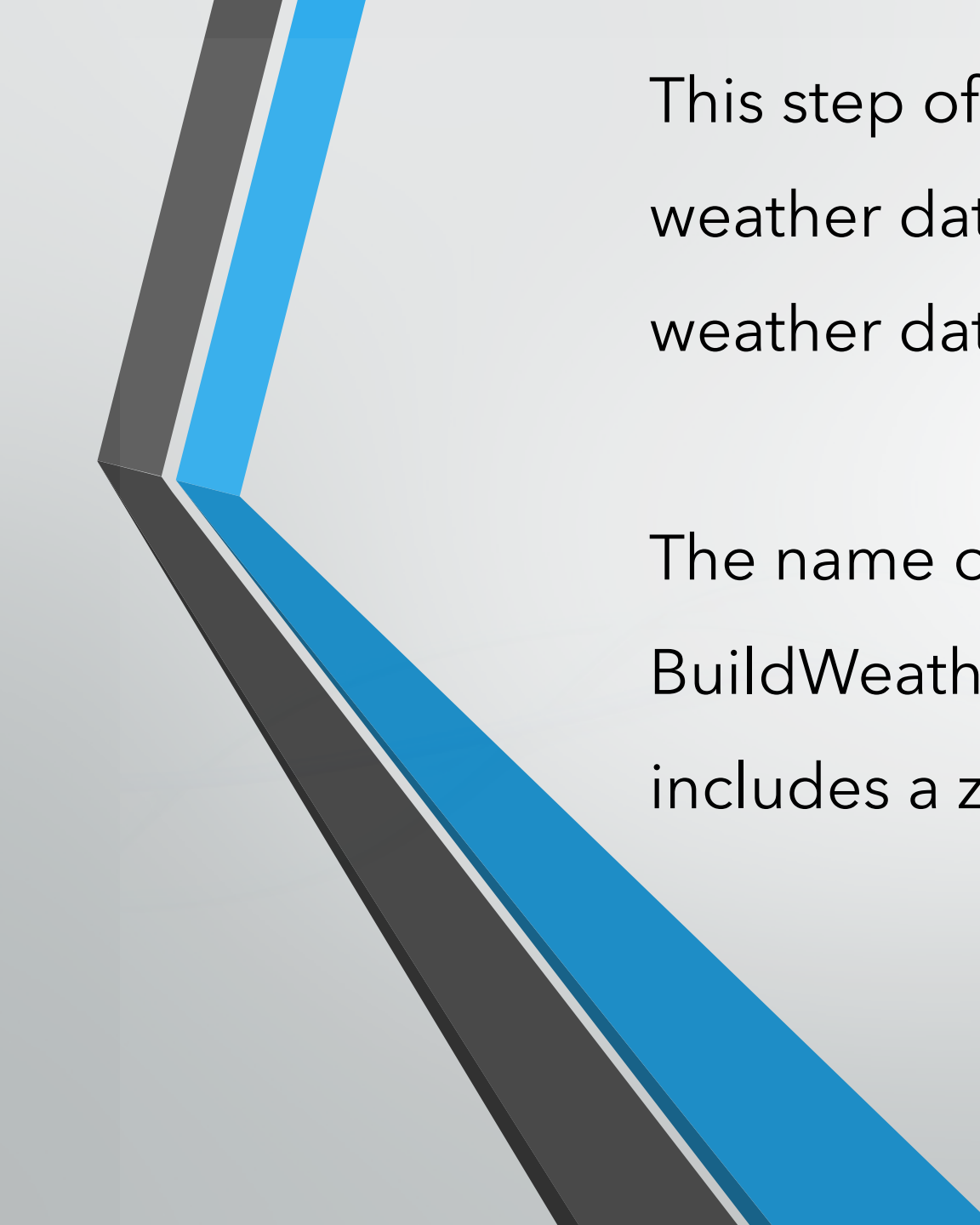
- Install python
- Download weather data to a database.
- Extract weather data from database into a comma separated file with python
- Cleanse weather data
- Use Excel to manipulate data
- Use python data analytics modules to develop graphical models



Screenshot of libraries from noaa-sdk being downloaded successfully.

```
use) C:\WINDOWS\system32>pip install noaa-sdk
Collecting noaa-sdk
  Downloading noaa_sdk-0.1.21-py3-none-any.whl (11 kB)
Requirement already satisfied: requests>=2.22.0 in c:\users\jfo84\anaconda3\lib\site-packages (from noaa-sdk) (2.27.1)
Requirement already satisfied: charset-normalizer~=2.0.0 in c:\users\jfo84\anaconda3\lib\site-packages (from requests>=2.22.0->noaa-sdk) (2.0.4)
Requirement already satisfied: idna<4,>=2.5 in c:\users\jfo84\anaconda3\lib\site-packages (from requests>=2.22.0->noaa-sdk) (3.3)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\jfo84\anaconda3\lib\site-packages (from requests>=2.22.0->noaa-sdk) (2021.10.8)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\jfo84\anaconda3\lib\site-packages (from requests>=2.22.0->noaa-sdk) (1.26.9)
Installing collected packages: noaa-sdk
Successfully installed noaa-sdk-0.1.21
```

Note: The libraries are necessary in order to use data downloaded from noaa in the next steps.



This step of the process involves downloading weather data with Python and building a weather database from that data.

The name of the download file is BuildWeatherDb.py and the screenshot includes a zip code to localize the data.

```
7 import sqlite3
8 import datetime
9
0 # parameters for retrieving NOAA weather data
1 zipCode = "01702" # change to your postal code
2 country = "US"
3 #date-time format is yyyy-mm-ddThh:mm:ssZ, times are Zulu time (GMT)
4 #gets the most recent 14 days of data
5 today = datetime.datetime.now()
6 past = today - datetime.timedelta(days=14)
7 startDate = past.strftime("%Y-%m-%dT00:00:00Z")
8 endDate = today.strftime("%Y-%m-%dT23:59:59Z")
9
0 #create connection - this creates database if not exist
1 print("Preparing database...")
2 dbFile = "weather.db"
3 conn = sqlite3.connect(dbFile)
4 #create cursor to execute SQL commands
5 cur = conn.cursor()
6
```

BuildWeatherDb.py Code

```
Console 1/1 *
IPython 8.2.0 -- An enhanced Interactive Python.

In [1]: runfile('C:/Users/jfo84/.spyder-py3/temp.py', wdir='C:/
Users/jfo84/.spyder-py3')
Preparing database...
Database prepared
Getting weather data...
Inserting rows...
216 rows inserted
Database load complete!

In [2]:
```

Python Console

The screenshot shows that the code executed successfully.

Name	Status	Date modified	Type	Size
Week 1	✓	7/20/2022 5:31 PM	File folder	
Week 2	✓	7/20/2022 5:33 PM	File folder	
BuildWeatherDb.py	✓	7/20/2022 6:22 PM	PY File	3 KB
CEIS110_Module3_Project_Guide	✓	7/20/2022 4:10 PM	Microsoft Word D...	329 KB
CEIS110_Week_3_Looping	✓	7/19/2022 4:21 PM	MP4 File	90,544 KB
CEIS1110_Module3_Project_Template	↻	7/20/2022 6:20 PM	Microsoft PowerP...	94 KB
weather.db	✓	7/20/2022 5:56 PM	Data Base File	36 KB

Weather.db file created

This part of the process includes using Python and SQL to create queries for retrieving data based on certain criteria.

Purpose

- Query to retrieve all columns and rows

- Query to retrieve lowest and highest temperature

- Query to find all Clear days

QueryWeatherDB.py x

```

#
"""
Created on Thu Jul 28 08:53:12 2022

@author: John O'Leary
"""

#Purpose: Query database using SQL
#Name: John O'Leary
#Date: July 28 2022
# Run BuildWeatherDB.py to build weather databas

import sqlite3
import pandas as pd

#file names for database and output file
dbFile = "weather.db"

#format output
pd.set_option('display.max_rows', None)
pd.set_option('display.max_columns', None)
pd.set_option('display.width', None)
pd.set_option('display.max_colwidth', None)
pd.set_option('display.expand_frame_repr', False)

#connect to and query weather database
conn = sqlite3.connect(dbFile)
#Create SQL command
selectCmd = selectCmd = " SELECT* FROM observatic

#print out the query
result = pd.read_sql_query(selectCmd, conn)
print(result)

```

Name	Date Modified
BuildWeatherDb.py	7/28/2022 9:02 AM

Help Plots Files

Console 1/A x

```

In [4]: runfile('C:/CEIS110/QueryWeatherDB.py', wdir='C:/CEIS110')

```

	timestamp	windSpeed	temperature	relativeHumidity	windDirection	barometricPressure	vs
textDescription							
0	2022-07-21T12:51:00+00:00	14.76	28.3	67.464688	200.0	100510	
Clear							
1	2022-07-21T13:51:00+00:00	14.76	30.0	63.042819	190.0	100480	
Clear							
2	2022-07-21T14:51:00+00:00	25.92	32.8	52.146765	200.0	100480	
Mostly Clear							
3	2022-07-21T15:51:00+00:00	27.72	33.3	48.872269	210.0	100440	
4	2022-07-21T16:51:00+00:00	25.92	34.4	47.679663	200.0	100340	
Partly Cloudy							
5	2022-07-21T17:51:00+00:00	NaN	35.6	43.002490	NaN	100270	
Mostly Clear							
6	2022-07-21T18:51:00+00:00	29.52	35.0	44.453013	200.0	100200	
Mostly Clear							
7	2022-07-21T19:51:00+00:00	37.08	33.3	48.872269	210.0	100200	
Mostly Cloudy and Windy							
8	2022-07-21T20:51:00+00:00	33.48	33.9	49.027453	NaN	100240	
Mostly Clear and Windy							
9	2022-07-21T21:51:00+00:00	NaN	32.2	55.612936	NaN	100200	
Partly Cloudy							
10	2022-07-21T22:51:00+00:00	20.52	31.7	59.334399	210.0	100240	
Partly Cloudy							
11	2022-07-21T23:51:00+00:00	16.56	30.0	65.383130	210.0	100270	
Partly Cloudy							
12	2022-07-22T00:51:00+00:00	12.96	28.3	72.136180	190.0	100340	
Clear							
13	2022-07-22T01:51:00+00:00	16.56	28.3	72.136180	230.0	100480	
Partly Cloudy							

Query to retrieve all columns and all rows

Created on Thu Jul 28 08:53:12 2022

@author: John O'Leary
"""

```
#Purpose: Query database using SQL  
#Name: John O'Leary  
#Date: July 28 2022  
# Run BuildWeatherDB.py to build weather database before running this program
```

```
import sqlite3  
import pandas as pd
```

```
#file names for database and output file  
dbFile = "weather.db"
```

```
#format output
```

```
pd.set_option('display.max_rows', None)  
pd.set_option('display.max_columns', None)  
pd.set_option('display.width', None)  
pd.set_option('display.max_colwidth', None)  
pd.set_option('display.expand_frame_repr', False)
```

```
#connect to and query weather database  
conn = sqlite3.connect(dbFile)  
#Create SQL command  
selectCmd = " SELECT MIN(temperature), MAX(temperature) FROM observations; "
```

```
Console 1/A x  
156 2022-07-28T06:51:00+00:00 0.00  
101150 16090 Clear  
157 2022-07-28T07:51:00+00:00 0.00  
101150 16090 Clear  
158 2022-07-28T08:51:00+00:00 0.00  
101120 16090 Partly Cloudy  
159 2022-07-28T09:51:00+00:00 0.00  
101120 16090 Clear  
160 2022-07-28T10:51:00+00:00 5.40  
101150 16090 Clear  
161 2022-07-28T11:51:00+00:00 7.56  
101150 16090 Cloudy  
  
In [5]: runfile('C:/CEIS110/QueryWeatherDB.p  
MIN(temperature) MAX(temperature)  
0 12.2 36.7  
  
In [6]:
```

Query to retrieve lowest and highest temperatures

ed on Thu Jul 28 08:53:12 2022

or: John O'Leary'

ose: Query database using SQL

: John O'Leary

: July 28 2022

un BuildWeatherDB.py to build weather database before running this program

```
import sqlite3
```

```
import pandas as pd
```

```
dbFile = "weather.db"
```

```
conn = sqlite3.connect(dbFile)
```

```
print(conn)
```

```
pd.set_option('display.max_rows', None)
```

```
pd.set_option('display.max_columns', None)
```

```
pd.set_option('display.width', None)
```

```
pd.set_option('display.max_colwidth', None)
```

```
pd.set_option('display.expand_frame_repr', False)
```

```
selectCmd = "SELECT temperature, windspeed, textDescription FROM observations where textDescription = 'Clear';"
```

```
conn.execute(selectCmd)
```

```
result = conn.fetchall()
```

```
selectCmd = "SELECT temperature, windspeed, textDescription FROM observations where textDescription = 'Clear';"
```

```
print(result)
```

```
df = pd.DataFrame(result)
```

```
print(df)
```


Help Plots Files

Console 1/A x

```
In [6]: runfile('C:/CEIS110/QueryWeatherDB.py', working_dir='C:/CEIS110')
```

	temperature	windSpeed	textDescription
0	18.9	5.40	Clear
1	16.1	0.00	Clear
2	16.7	0.00	Clear
3	16.7	0.00	Clear
4	16.1	0.00	Clear
5	17.2	0.00	Clear
6	17.8	0.00	Clear
7	18.9	0.00	Clear
8	21.7	0.00	Clear
9	22.8	11.16	Clear
10	26.1	0.00	Clear
11	28.9	12.96	Clear
12	29.4	12.96	Clear
13	28.9	11.16	Clear
14	28.9	11.16	Clear
15	27.8	11.16	Clear
16	27.8	12.96	Clear
17	25.6	12.96	Clear
18	22.8	11.16	Clear
19	21.1	7.56	Clear
20	16.7	0.00	Clear
21	12.2	0.00	Clear
22	14.4	0.00	Clear
23	14.4	0.00	Clear
24	14.4	0.00	Clear

Query to retrieve data for all clear days



This step of building the program involves using Python to extract data from the weather database and use SQL to query and the manipulate the data.

Then create a graphical representation of the extracted data in Excel.

```
#Purpose: Extract temperature, humidity data from weather database into CSV file
#Name: John O'Leary
#Date: August 1, 2022
# Run BuildWeatherDB.py to build weather database before running this program

import sqlite3

#convert Celsius temperature to Fahrenheit
def convertCtoF(tempC):
    return (tempC*9.0/5.0) + 32.0

#file names for database and output file
dbFile = "weather.db"
output_file_name='formatdata.csv'

#connect to and query weather database and
dbFile = "weather.db"
conn = sqlite3.connect(dbFile)
#create cursor to execute SQL commands
cur = conn.cursor()
selectCmd = """ SELECT temperature, relativeHumidity FROM observations
                ORDER BY timestamp; """
cur.execute(selectCmd)
allRows = cur.fetchall()
#limit the number of rows output to half
rowCount = len(allRows)//2 # double slash does integer division
rows = allRows[:rowCount]

#write data to output file
with open(output_file_name,"w+") as outf:
    outf.write('Celsius,Fahrenheit,Humidity')
    outf.write('\n')
    for row in rows:
```

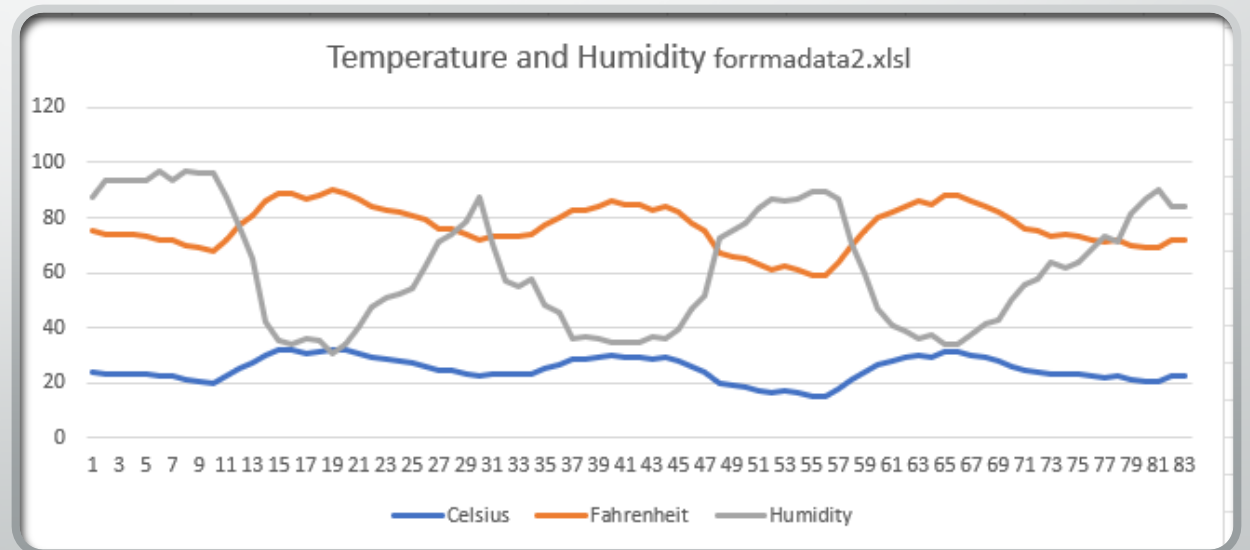
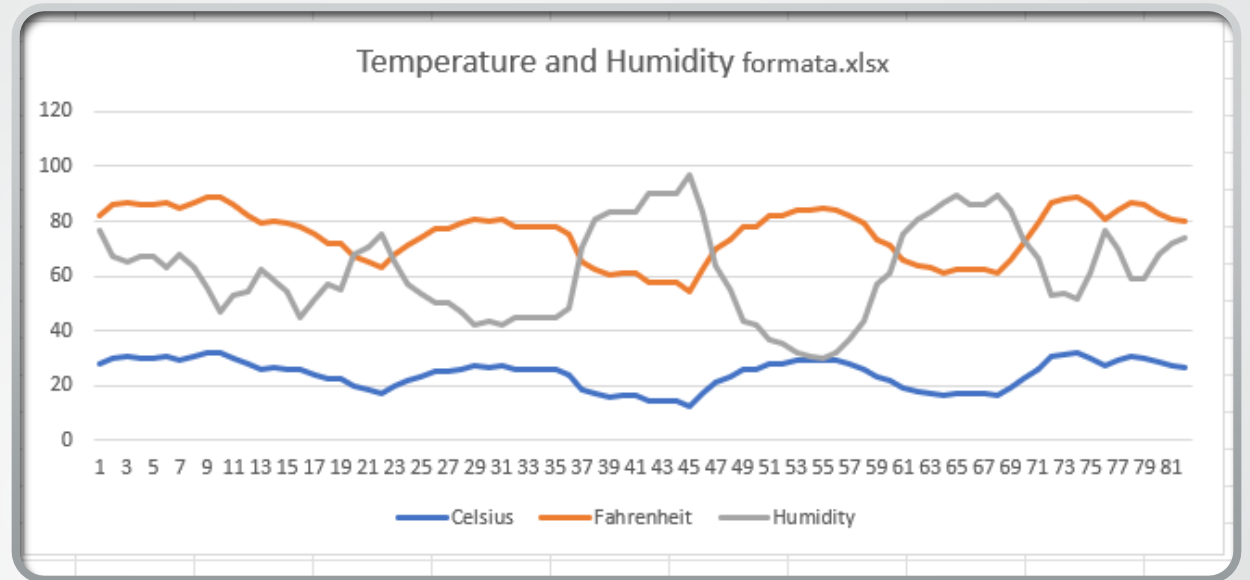
Python code used to extract the data.

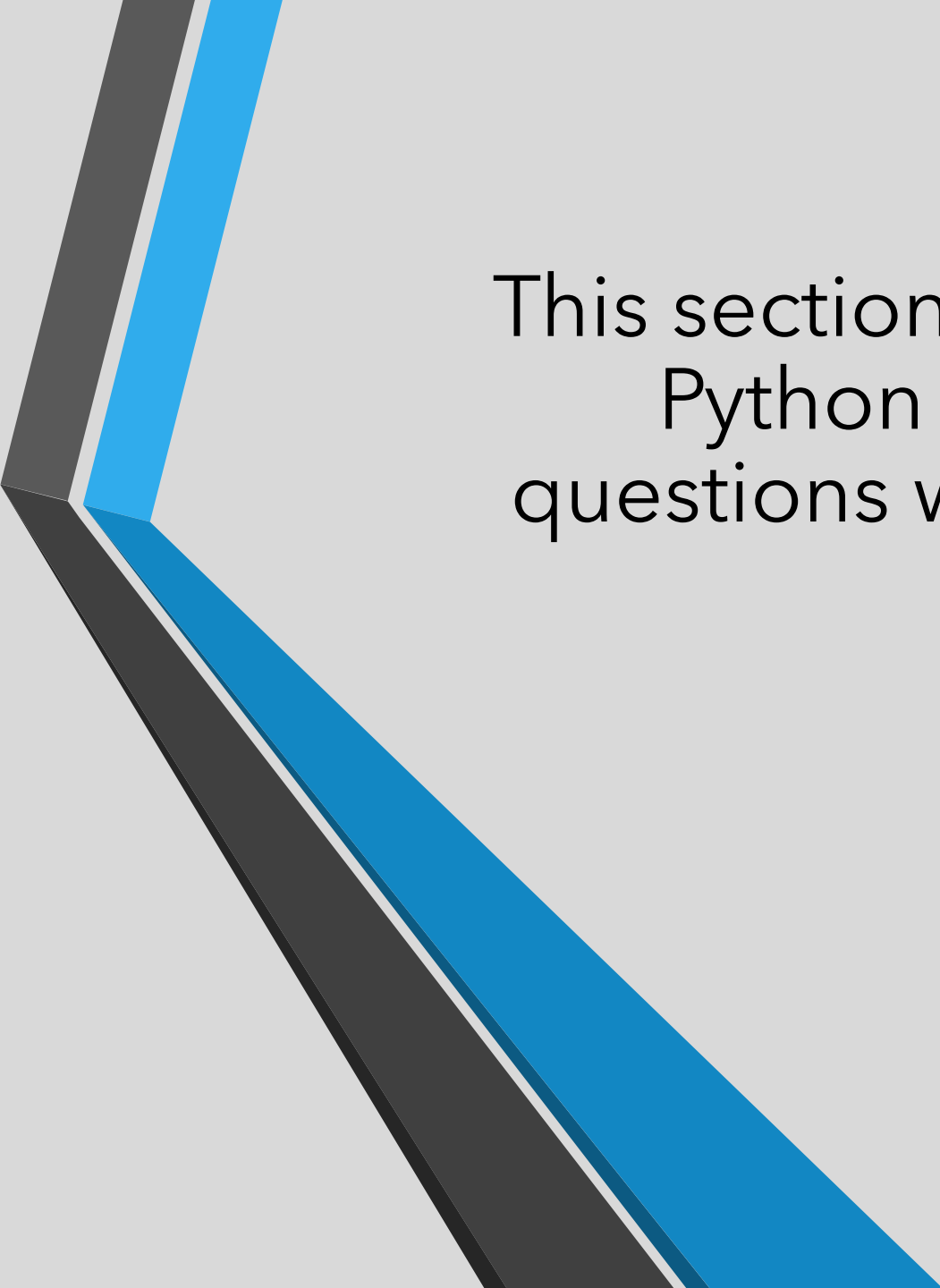
22.2	71.96	96.99591743	30	86	67.391164
22.2	71.96	93.49564967	30	86	67.391164
21.1	69.98	96.9712047	30.6	87.08	63.172227
20.6	69.08	96.36167032	29.4	84.92	67.681664
20	68	96.34534192	30.6	87.08	63.172227
22.2	71.96	87.36520778	31.7	89.06	55.491942
25	77	76.58907524	31.7	89.06	46.678267
27.2	80.96	64.79910978	30	86	53.064596
30	86	41.75007988	27.8	82.04	54.534039
31.7	89.06	35.07281996	26.1	78.98	62.186891
31.7	89.06	33.955371	26.7	80.06	58.167242
			26.1	78.98	54.108554
			25.6	78.08	45.033206

Retrieve and Convert Data to CSV Format

Temperature and Humidity (Line Graphs)

- Excel charts based on temperature and humidity data from database

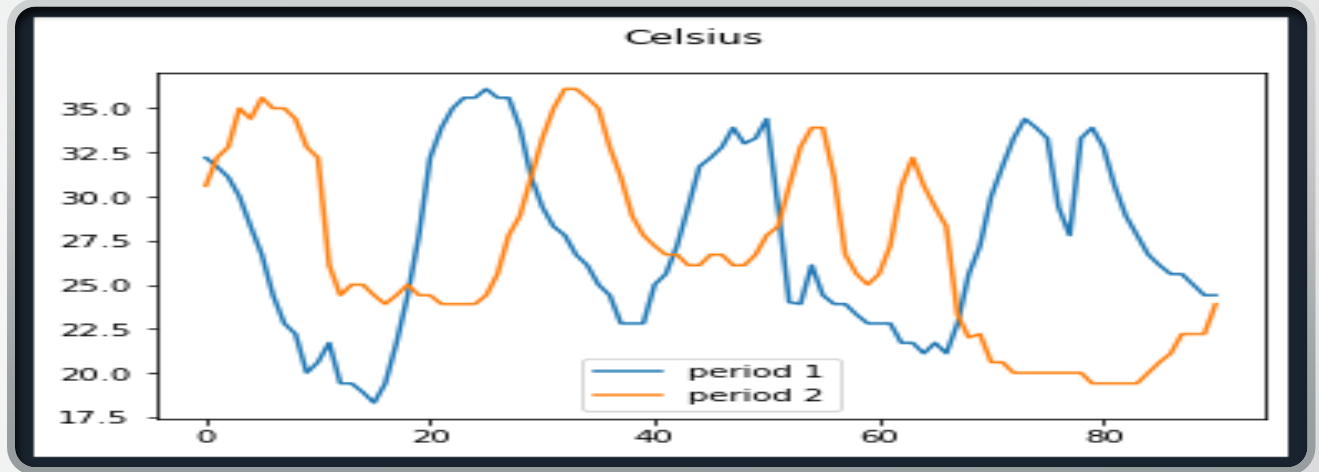




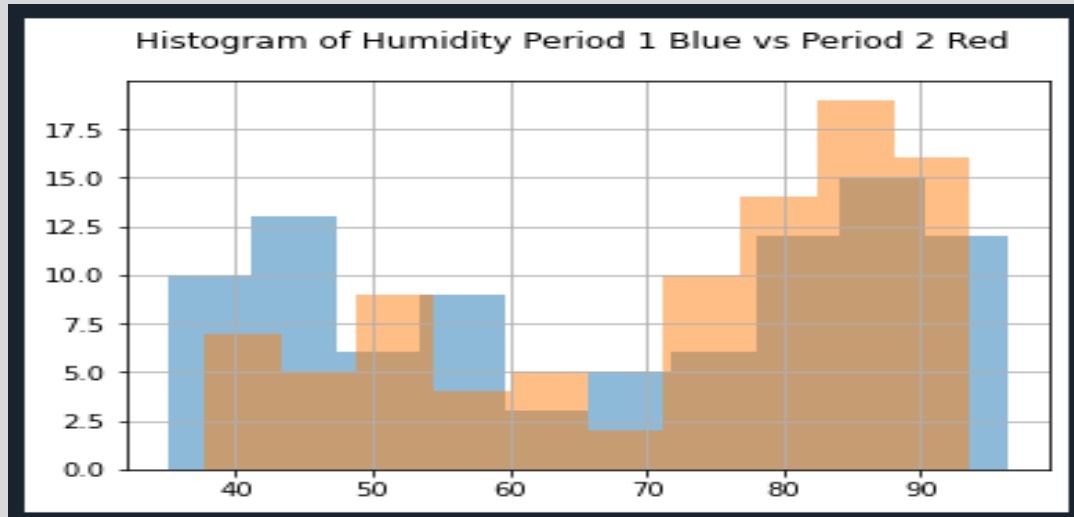
This section contains plots, graphs,
Python code, and data analytic
questions which were a part of the
process.

Plot #1

- This is a line chart depicting the variations in temperature in Celsius from one week to the next.
- The code is what was used to generate the output using Python.



```
#Purpose: Create Celsius plot comparing period 1 and period 2
#Name: John O'Leary
#Date: August 9, 2022
import pandas as pd
import matplotlib.pyplot as plt
df1 = pd.read_csv("formatdata.csv") #baseline data is period 1 (older)
df2 = pd.read_csv("formatdata2.csv") #data for period 2 (more recent)
plt.figure(); df1.Celsius.plot(label = 'period 1');
df2.Celsius.plot(label = 'period 2'); plt.legend(loc='best'); plt.suptitle('Celsius')
plt.show()
```



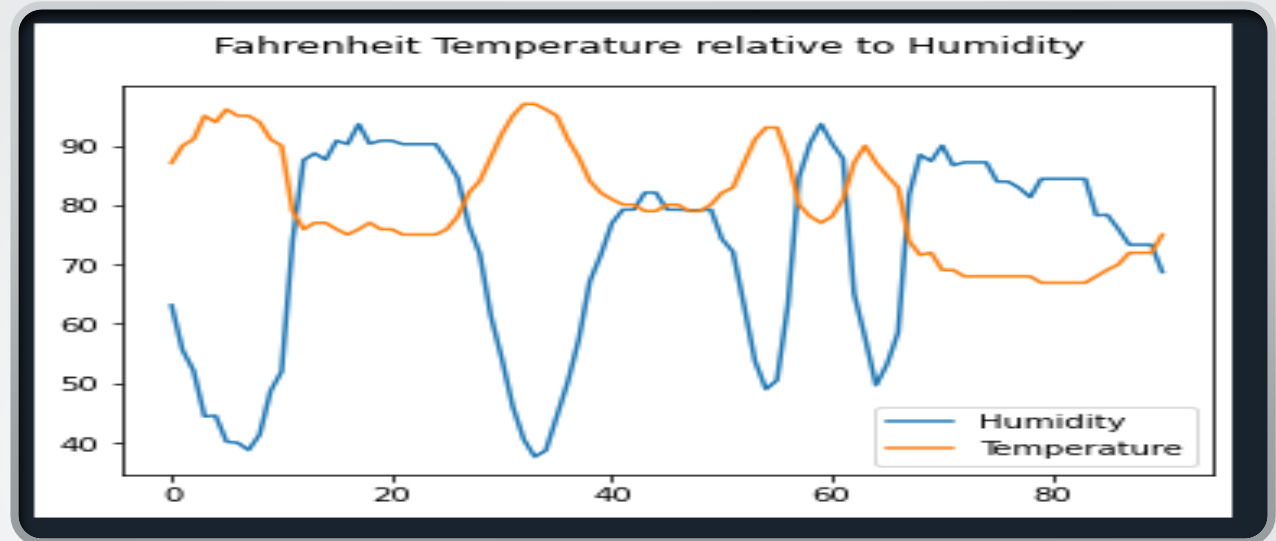
Plot #2

- Included is a histogram showing difference in humidity between week one and week two.
- The code is what was used to generate the graphical representation in Python.

```
#Purpose: Create a histogram of humidity data comparing both periods
#Name: John O'Leary
#Date: August 11, 2022
import pandas as pd
import matplotlib.pyplot as plt
df1 = pd.read_csv("formatdata.csv")
df1['Humidity'].hist(bins=10, alpha=0.5)
df2 = pd.read_csv("formatdata2.csv")
df2['Humidity'].hist(bins=10, alpha=0.5);
plt.suptitle('Histogram of Humidity (Period 1 Blue vs. Period 2 Red)');
```

Analysis

- My own question:
- What is the correlation between temperatures and humidity?
- Answer supported by Chart:
- Higher temperatures most often coincide with low humidity ranges. Spikes in temperature are usually preceded by dropping humidity.



```
#Purpose: Create a line graph to see the correlation between
#temperature and humidity in period 2
#Name: John O'Leary
#Date: August 11, 2022
import pandas as pd
import matplotlib.pyplot as plt
df1 = pd.read_csv("formatdata.csv") #data for period 1 (older)
df2 = pd.read_csv("formatdata2.csv") #data for period 2 (more recent)
plt.figure(); df2.Humidity.plot(label = 'Humidity ')
df2.Fahrenheit.plot(label = 'Temperature')
plt.legend(loc='best')
plt.suptitle('Fahrenheit Temperature relative to Humidity'); plt.show()
```



Prediction

Based on the current conditions, and if they follow the pattern displayed in the line graph, I expect that in the next few days, the humidity will continue to rise. It's currently at 52% and the heat index is at 73 degrees.

Another factor I considered in my prediction is based on my geographical location. Wind speed and direction, as well as barometric pressure have an impact on temperature and humidity levels. Therefore, here in New England the usual pattern does not always follow that when temperatures are high, humidity is low.



Challenges

The challenges to this project were the initial introduction to Python and learning the many syntax rules and commands.

At times, the error codes which were generated were difficult to understand and find the reason for the error message.

Learning to develop skills to plan and create programs is something which will take time.

The more I learned about using Python, the more I found I needed to learn.



Career Skills

Adding Linux to my skillset is a tremendous advantage for opening new doors in the IT job market.

Critical thinking is an important aspect of planning and creating programs and was very much exercised in this project.


Using Anaconda and Spyder are new skills I gained from this project.

Using Excel to create graphical models based on data is another new skill I acquired during this project.

My communications skills were improved every week during discussion postings and live lectures.

Conclusion

- In conclusion I want to thank Professor Boulware for his excellent and readily available guidance and sharing his wealth of knowledge and expertise on the subject. I also want to thank the other Professors, many of whom I gained some valuable information from. This course is one of the most difficult yet rewarding courses I've taken so far.
- This project is a representation of everything I learned during this course. I hope to continue learning about Python and have opportunities to put the knowledge to good use.



Thank you for
watching!

