```
ror object to mirror
irror_mod.mirror_object
peration == "MIRROR_X":
irror_mod.use_x = True
irror_mod.use_y = False
irror_mod.use_z = False
_operation == "MIRROR_Y":
irror_mod.use_x = False
irror_mod.use_y = True
irror_mod.use_z = False
_operation == "MIRROR_Z":
irror_mod.use_x = False
irror_mod.use_y = False
irror_mod.use_z = True

selection at the end -add
_ob.select= 1
er_ob.select=1
ntext.scene.objects.active
("Selected" + str(modifier
irror_ob.select = 0
bpy.context.selected_obj
ata.objects[one.name].se

int("please select exactly

- OPERATOR CLASSES ----

types.Operator):
X mirror to the selected
ject.mirror_mirror_x"
rror X"
```

CEIS110 Programming with Data

Introduction

Data is growing

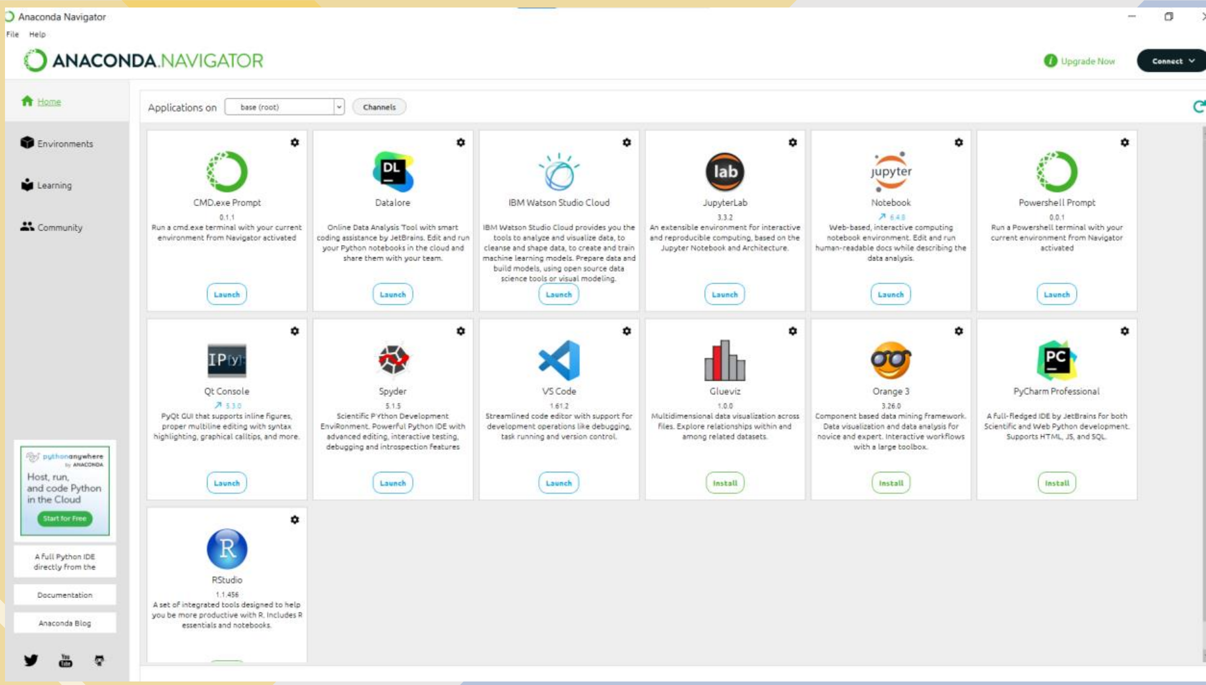
This project uses a cloud system to gather temperature and humidity data

Them the data is analyzed using programming data

Software Inventory

- Before developing programming with data project, all software must bed downloaded and installed
- Software needed included SQLite studio, Excel, and Python IDE - anaconda

Software:
The Software needed for this project includes Microsoft Excel SQLite anaconda Navigation with Spyder



CMD.exe Prompt
0.1.1
Run a cmd.exe terminal with your current environment from Navigator activated

Datalore
Online Data Analysis Tool with smart coding assistance by JetBrains. Edit and run your Python notebooks in the cloud and share them with your team.

IBM Watson Studio Cloud
IBM Watson Studio Cloud provides you the tools to analyze and visualize data, to cleanse and shape data, to create and train machine learning models. Prepare data and build models, using open source data science tools or visual modeling.

JupyterLab
3.3.2
An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.

Notebook
6.4.8
Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.

Powershell Prompt
0.0.1
Run a Powershell terminal with your current environment from Navigator activated

Qt Console
5.3.0
PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.

Spyder
5.1.5
Scientific PYthon Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection Features

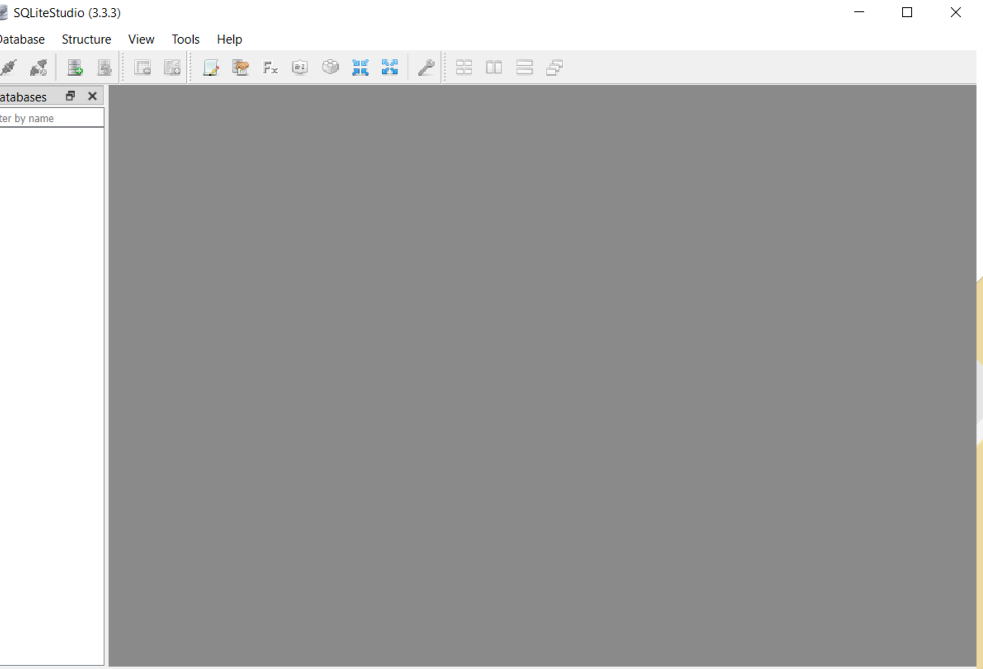
VS Code
1.61.2
Streamlined code editor with support for development operations like debugging, task running and version control.

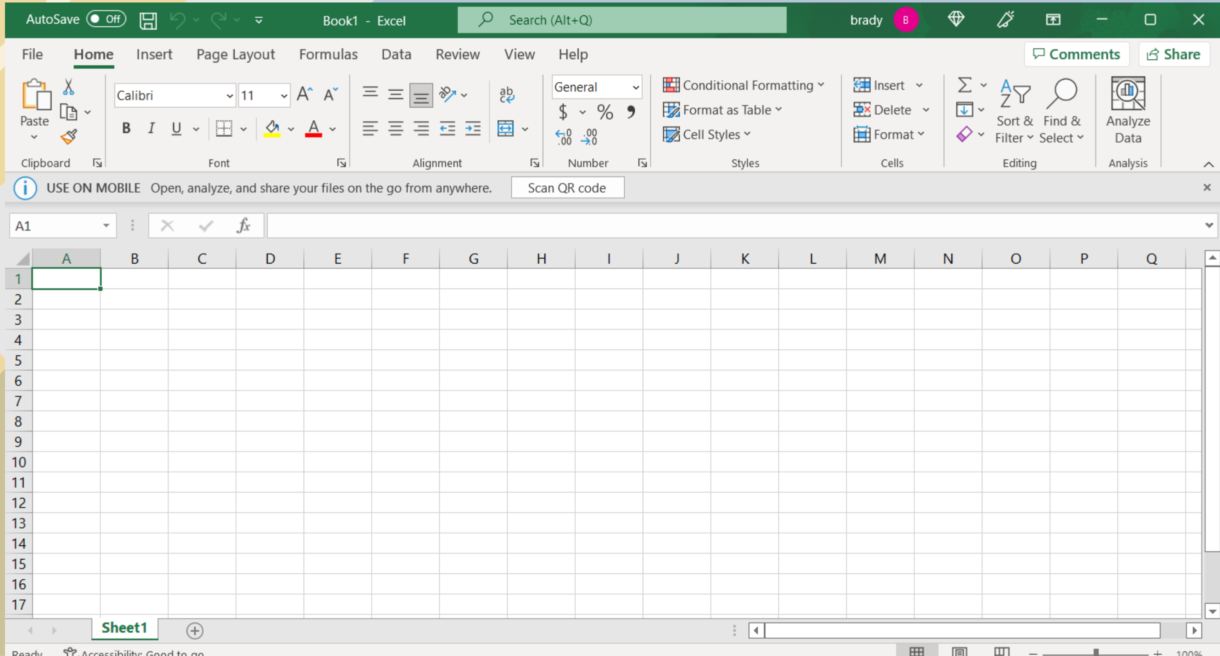
Glueviz
1.0.0
Multidimensional data visualization across files. Explore relationships within and among related datasets.

Orange 3
3.26.0
Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox.

PyCharm Professional
A full-fledged IDE by JetBrains for both Scientific and Web Python development. Supports HTML, JS, and SQL.

RStudio
1.1.456
A set of integrated tools designed to help you be more productive with R. Includes R essentials and notebooks.

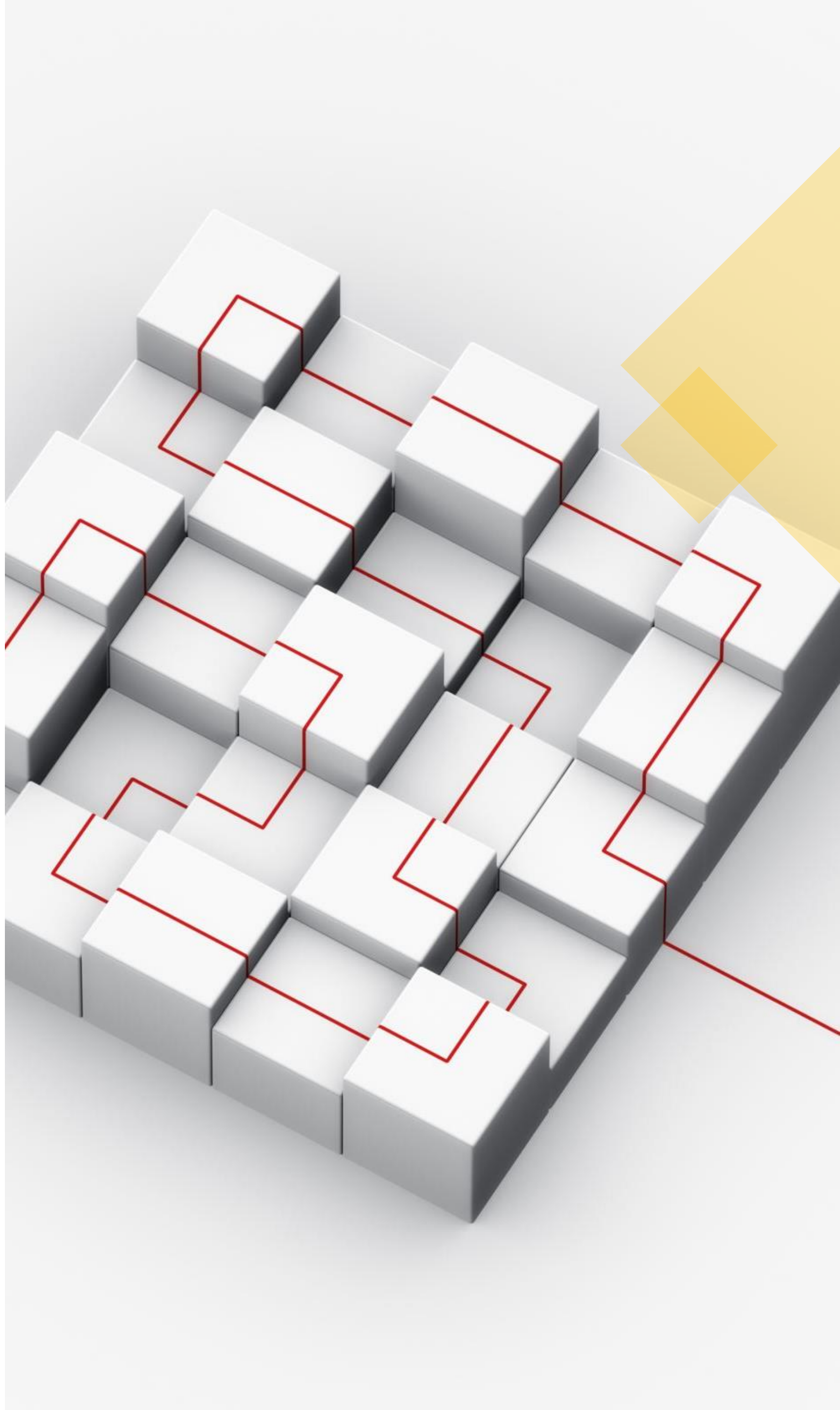




USE ON MOBILE Open, analyze, and share your files on the go from anywhere.

Planning and Design

- After performing of the software needed for the project, a plan was created for the data and temperature project.
- To plan the design of the project, a flowchart was generated.
- Planning and design of the project are crucial steps to understanding the development process.



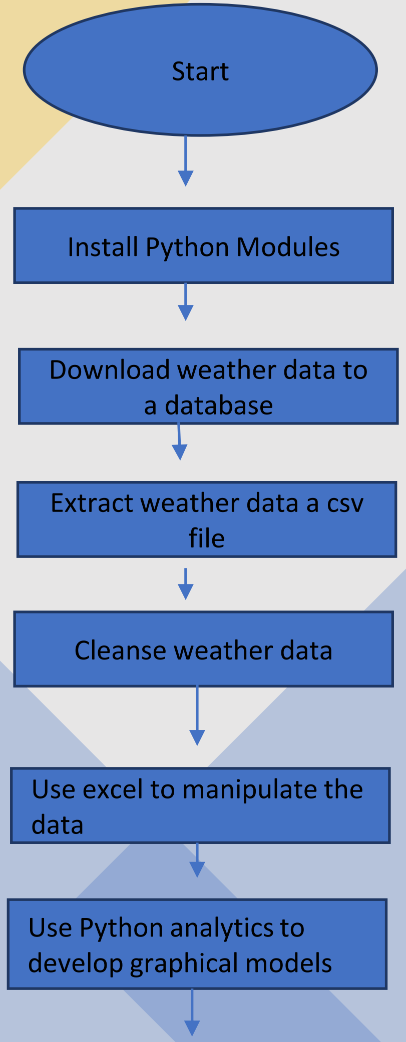
What are Flowcharts?

- Flowcharts are graphical representations of processes of workflows, and the flowchart developed illustrates the process and the output of this software development project.
- Many companies use flowcharts as simple narratives

Flowchart

Include the following processes:

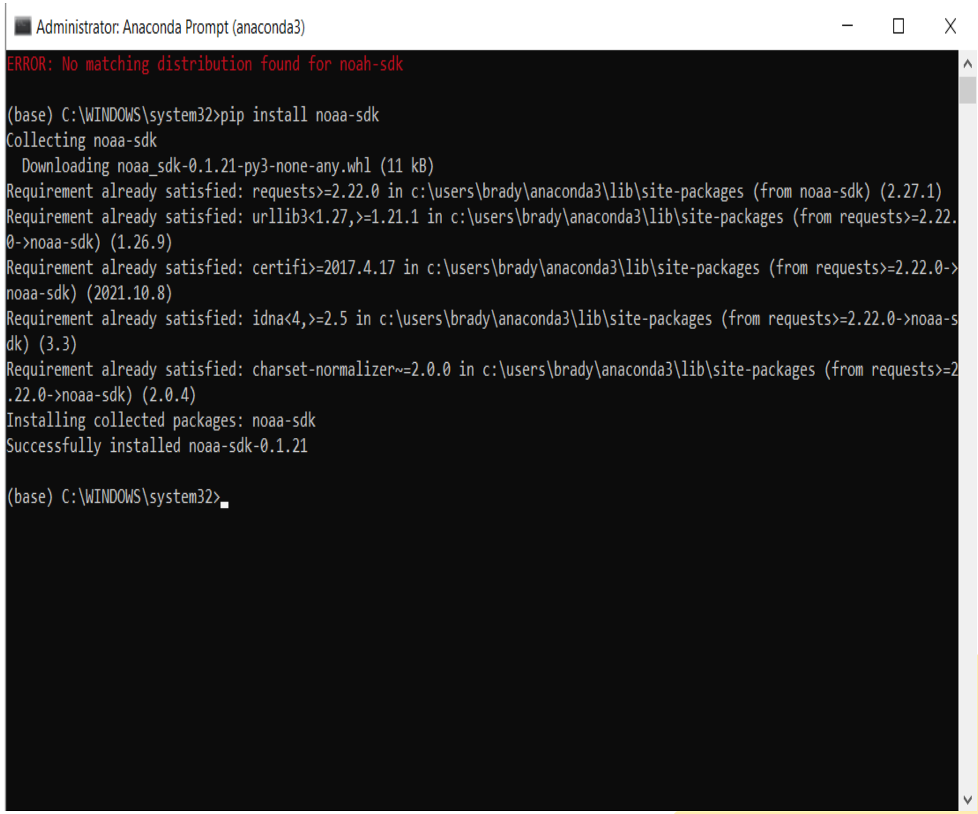
- Install python
- Download weather data to a database.
- Extract weather data from database into a comma separated file with python
- Cleanse weather data
- Use Excel to manipulate data
- Use python data analytics modules to develop graphical models



Start
Install Python Modules
Download weather data to a database
Extract weather data a csv file
Cleanse weather data
Use excel to manipulate the data
Use Python analytics to develop graphical models

Adding Library

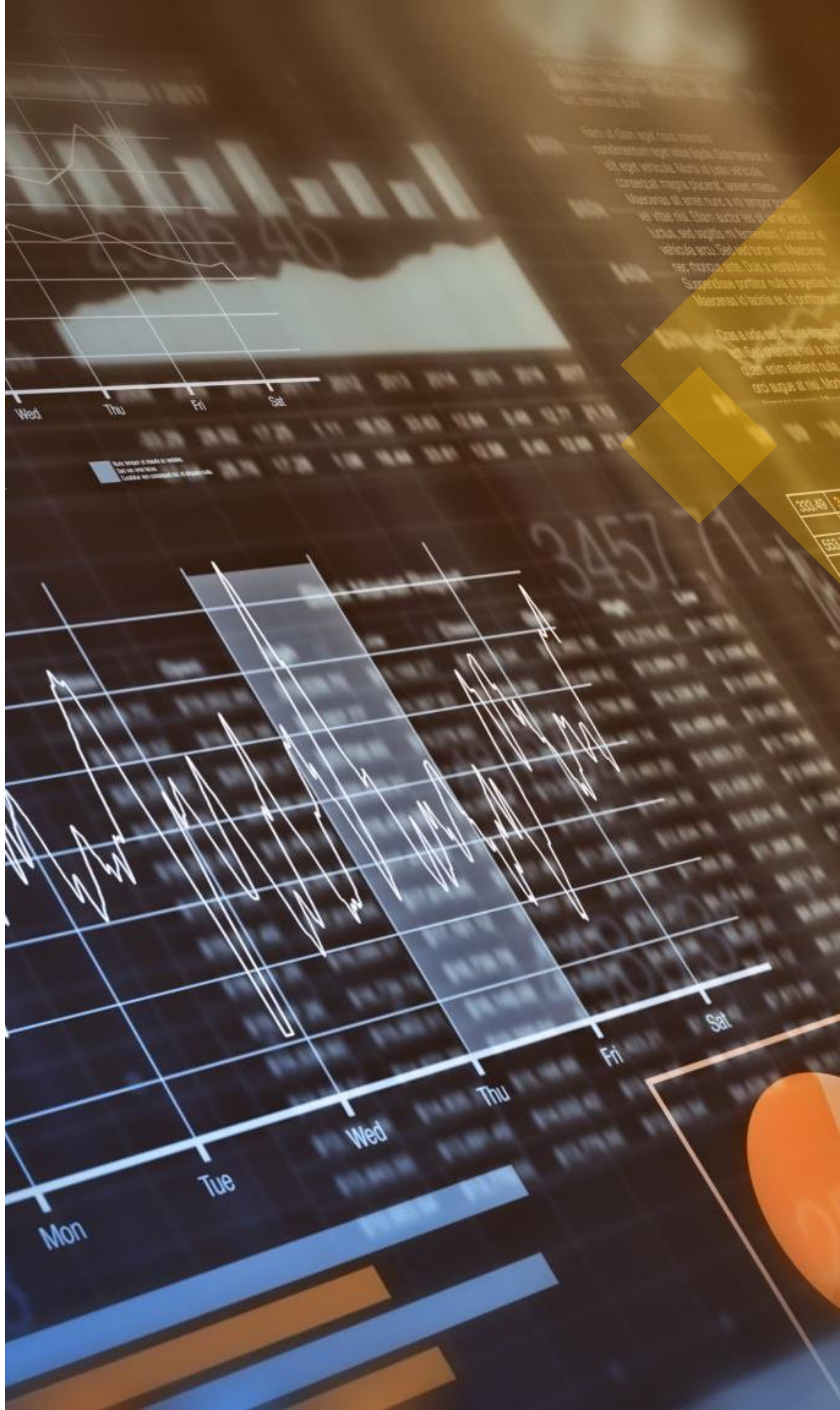
- In order for python to connect to the US government Nation oceanic and Atmospheric Administration (NOAA) weather data service using a cloud-based Application Programming Interface (API), a library module must be installed to use this service.
- The screenshot shows the NOAA-SDK library installed.



```
ERROR: No matching distribution found for noah-sdk

(base) 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\brady\anaconda3\lib\site-packages (from noaa-sdk) (2.27.1)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\brady\anaconda3\lib\site-packages (from requests>=2.22.
0->noaa-sdk) (1.26.9)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\brady\anaconda3\lib\site-packages (from requests>=2.22.0->
noaa-sdk) (2021.10.8)
Requirement already satisfied: idna<4,>=2.5 in c:\users\brady\anaconda3\lib\site-packages (from requests>=2.22.0->noaa-s
dk) (3.3)
Requirement already satisfied: charset-normalizer~=2.0.0 in c:\users\brady\anaconda3\lib\site-packages (from requests>=2
.22.0->noaa-sdk) (2.0.4)
Installing collected packages: noaa-sdk
Successfully installed noaa-sdk-0.1.21

(base) C:\WINDOWS\system32>
```

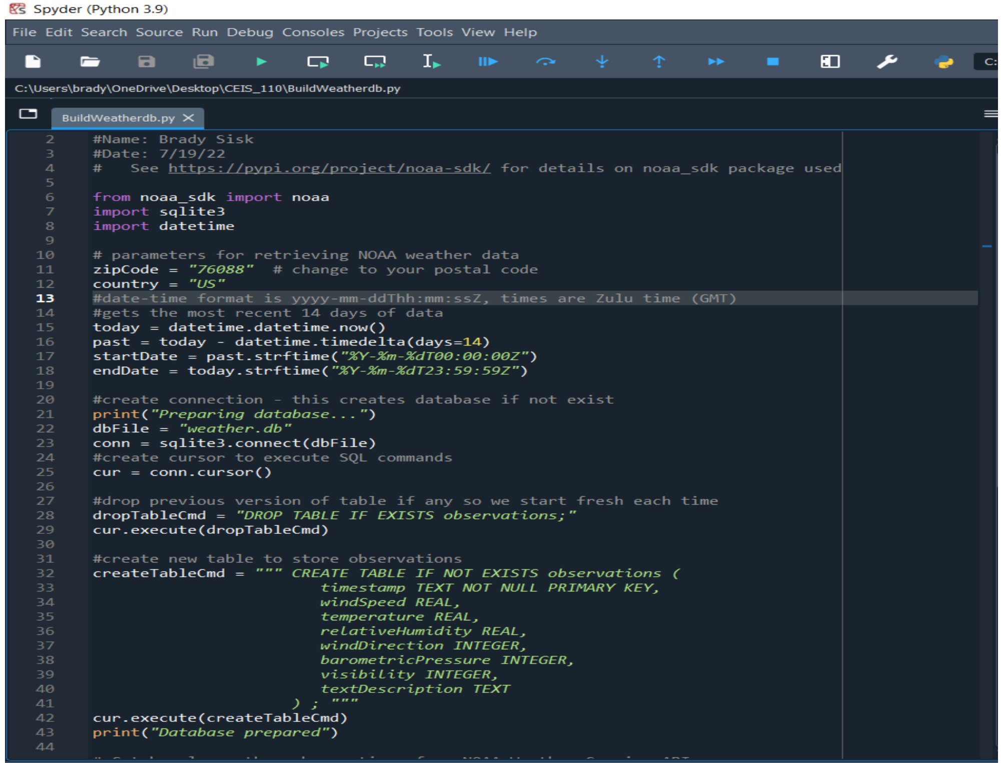
Gathering Temperature and Humidity data

- After planning and design, the code was developed to download a set of weather observations
- This data was stored on local database in table for later analysis



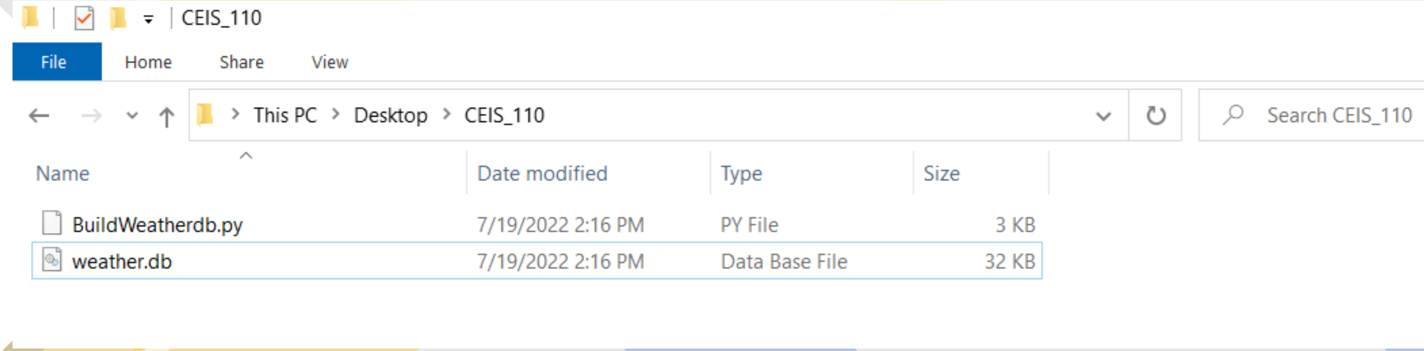
BuildWeatherDb.py

- Screenshot of code in Spyder
- The code will create a table named Observations, the fields timestamp, windspeed, temperature, relativeHumidity, windDirection, barometricPressure, visibility, and textDescription.
- The database will be named weather.db and stored in the same directory as the python code


Spyder (Python 3.9)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\brady\OneDrive\Desktop\CEIS_110\BuildWeatherdb.py
BuildWeatherdb.py

```
#Name: Brady Sisk
#Date: 7/19/22
#   See https://pypi.org/project/noaa-sdk/ for details on noaa_sdk package used

from noaa_sdk import noaa
import sqlite3
import datetime

# parameters for retrieving NOAA weather data
zipCode = "76088"  # change to your postal code
country = "US"
#date-time format is yyyy-mm-ddThh:mm:ssZ, times are Zulu time (GMT)
#gets the most recent 14 days of data
today = datetime.datetime.now()
past = today - datetime.timedelta(days=14)
startDate = past.strftime("%Y-%m-%dT00:00:00Z")
endDate = today.strftime("%Y-%m-%dT23:59:59Z")

#create connection - this creates database if not exist
print("Preparing database...")
dbFile = "weather.db"
conn = sqlite3.connect(dbFile)
#create cursor to execute SQL commands
cur = conn.cursor()

#drop previous version of table if any so we start fresh each time
dropTableCmd = "DROP TABLE IF EXISTS observations;"
cur.execute(dropTableCmd)

#create new table to store observations
createTableCmd = """ CREATE TABLE IF NOT EXISTS observations (
                        timestamp TEXT NOT NULL PRIMARY KEY,
                        windSpeed REAL,
                        temperature REAL,
                        relativeHumidity REAL,
                        windDirection INTEGER,
                        barometricPressure INTEGER,
                        visibility INTEGER,
                        textDescription TEXT
                     ) ; """
cur.execute(createTableCmd)
print("Database prepared")

```

Weather db File



Name	Date modified	Type	Size
BuildWeatherdb.py	7/19/2022 2:16 PM	PY File	3 KB
weather.db	7/19/2022 2:16 PM	Data Base File	32 KB

Screenshot of Windows explorer showing database wether db was created.

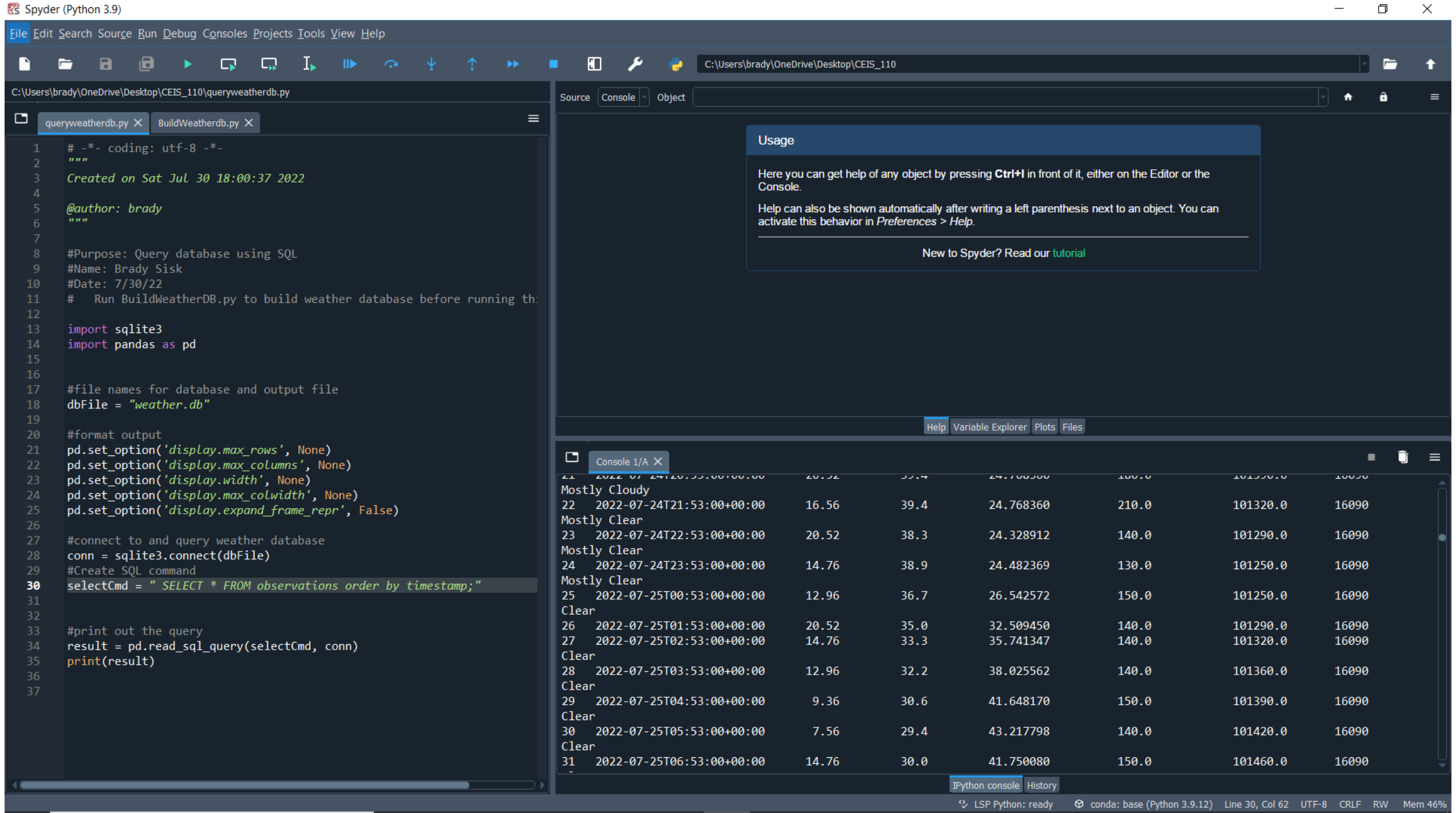
Querying the Database

Querying the Database

- Structured Query Language is a programming language used for working with relational database
- SQLiteStudio was sued to query the database and view the results

Query to retrieve all columns and all rows

- The SQL command “select * from observations” was executed to retrieve all rows and columns from the observations table.


Spyder (Python 3.9)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\brady\OneDrive\Desktop\CEIS_110
C:\Users\brady\OneDrive\Desktop\CEIS_110\queryweatherdb.py
queryweatherdb.py
BuildWeatherdb.py
Source
Console
Object
Help
Variable Explorer
Plots
Files
Console 1/A
IPython console
History
LSP Python: ready
conda: base (Python 3.9.12)
Line 30, Col 62
UTF-8
CRLF
RW
Mem 46%

```python
# -*- coding: utf-8 -*-
"""
Created on Sat Jul 30 18:00:37 2022

@author: brady
"""

#Purpose: Query database using SQL
#Name: Brady Sisk
#Date: 7/30/22
#   Run BuildWeatherDB.py to build weather database before running th

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 * FROM observations order by timestamp;"


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

Usage

Here you can get help of any object by pressing **Ctrl+I** in front of it, either on the Editor or the Console.

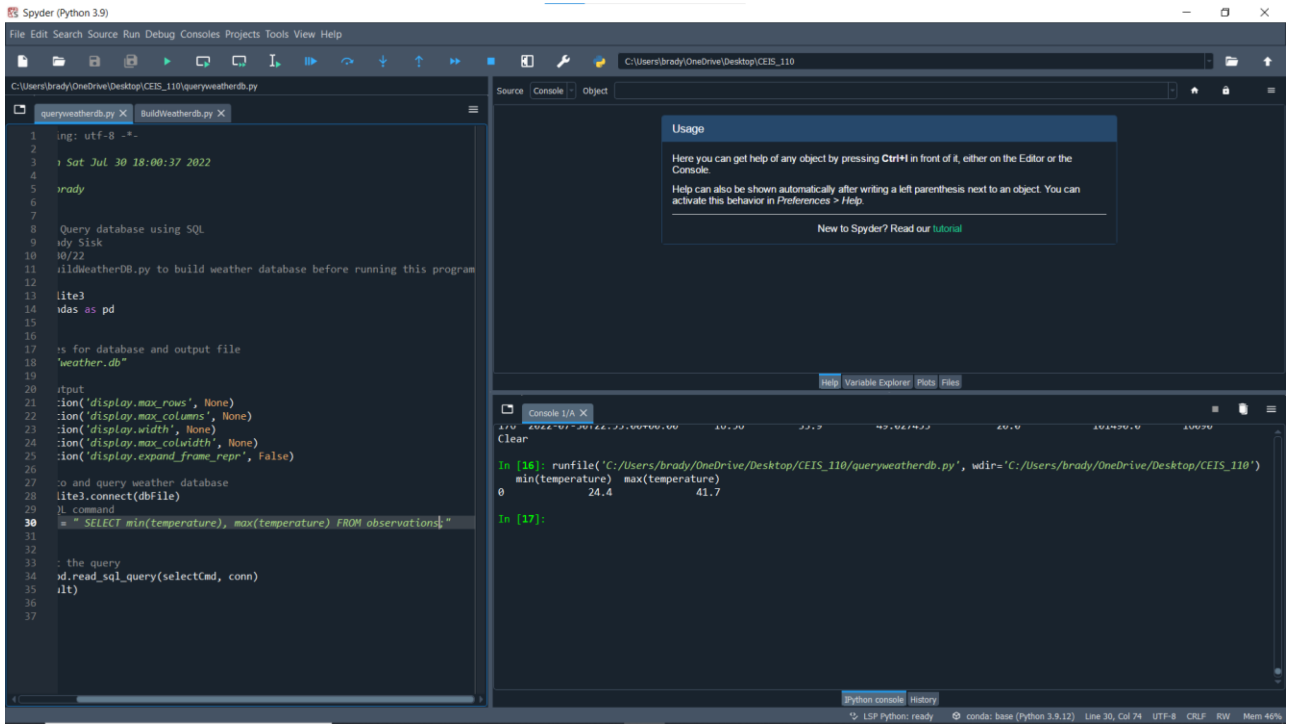
Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in *Preferences > Help*.

New to Spyder? Read our tutorial

```
Mostly Cloudy
22   2022-07-24T21:53:00+00:00      16.56        39.4        24.768360        210.0          101320.0      16090
Mostly Clear
23   2022-07-24T22:53:00+00:00      20.52        38.3        24.328912        140.0          101290.0      16090
Mostly Clear
24   2022-07-24T23:53:00+00:00      14.76        38.9        24.482369        130.0          101250.0      16090
Mostly Clear
25   2022-07-25T00:53:00+00:00      12.96        36.7        26.542572        150.0          101250.0      16090
Clear
26   2022-07-25T01:53:00+00:00      20.52        35.0        32.509450        140.0          101290.0      16090
27   2022-07-25T02:53:00+00:00      14.76        33.3        35.741347        140.0          101320.0      16090
Clear
28   2022-07-25T03:53:00+00:00      12.96        32.2        38.025562        140.0          101360.0      16090
Clear
29   2022-07-25T04:53:00+00:00       9.36        30.6        41.648170        150.0          101390.0      16090
Clear
30   2022-07-25T05:53:00+00:00       7.56        29.4        43.217798        140.0          101420.0      16090
Clear
31   2022-07-25T06:53:00+00:00      14.76        30.0        41.750080        150.0          101460.0      16090
```

Query to retrieve lowest and highest temperature

- The min and max temperatures were retrieved. These temperatures are captured based on the Celsius scale.


Spyder (Python 3.9)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\brady\OneDrive\Desktop\CEIS_110
C:\Users\brady\OneDrive\Desktop\CEIS_110\queryweatherdb.py
queryweatherdb.py
BuildWeatherdb.py
Source
Console
Object
Help
Variable Explorer
Plots
Files
Console 1/A
IPython console
History
LSP Python: ready
conda: base (Python 3.9.12)
Line 30, Col 74
UTF-8
CRLF
RW
Mem 46%

```
ing: utf-8 -*-

ı Sat Jul 30 18:00:37 2022

ɔrady


 Query database using SQL
ıdy Sisk
30/22
ıildWeatherDB.py to build weather database before running this program

lite3
ndas as pd


es for database and output file
'weather.db"

ıtput
tion('display.max_rows', None)
tion('display.max_columns', None)
tion('display.width', None)
tion('display.max_colwidth', None)
tion('display.expand_frame_repr', False)

to and query weather database
lite3.connect(dbFile)
QL command
 = " SELECT min(temperature), max(temperature) FROM observations;"


: the query
od.read_sql_query(selectCmd, conn)
ılt)
```

Usage

Here you can get help of any object by pressing **Ctrl+I** in front of it, either on the Editor or the Console.

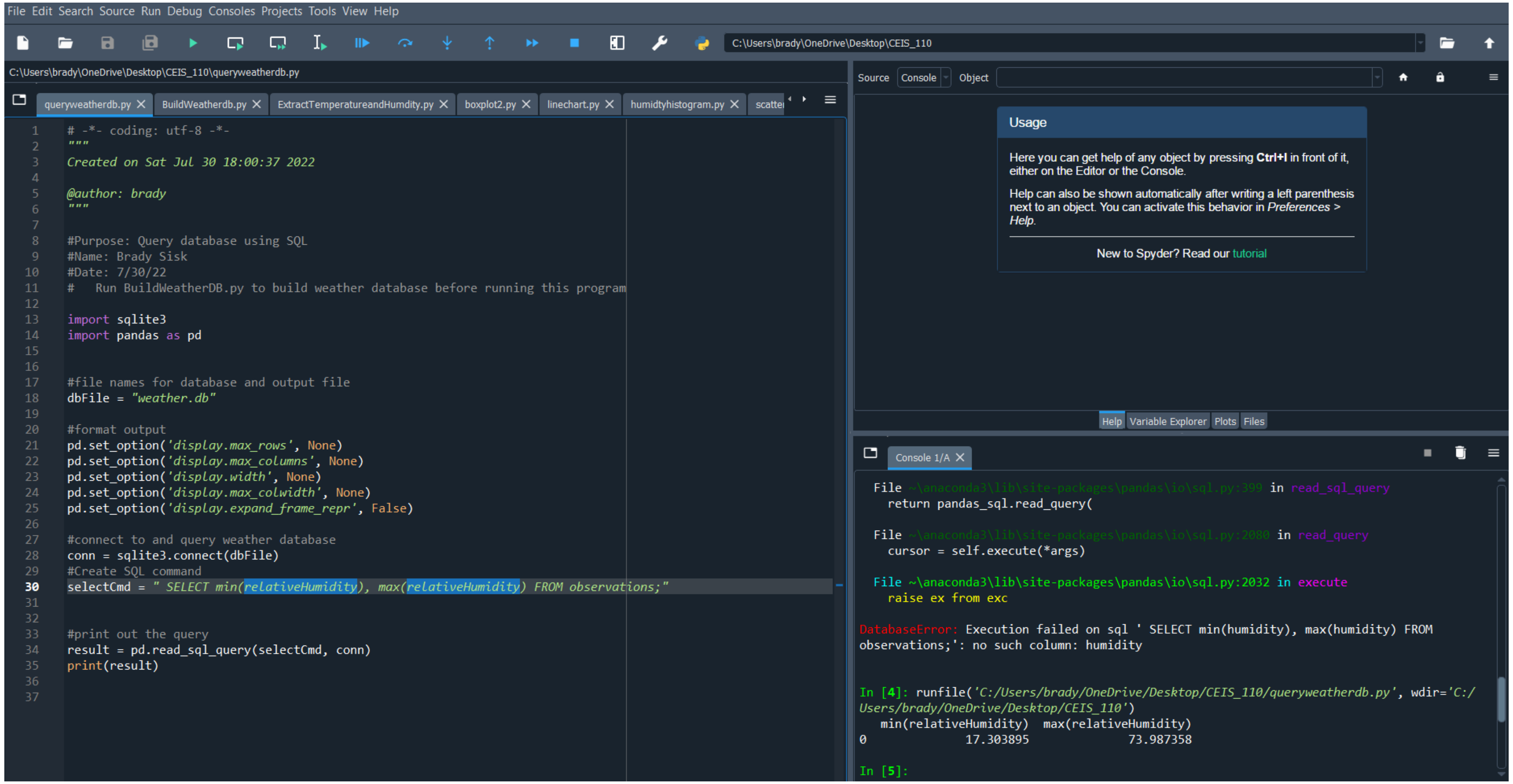
Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in *Preferences > Help*.

New to Spyder? Read our tutorial

```
Clear

In [16]: runfile('C:/Users/brady/OneDrive/Desktop/CEIS_110/queryweatherdb.py', wdir='C:/Users/brady/OneDrive/Desktop/CEIS_110')
   min(temperature)  max(temperature)
0              24.4              41.7

In [17]:
```

Query to retrieve lowest and highest humidity

- Another query performed to retrieve the lowest and highest relative humidity. The relativeHumidity column was used to retrieve this information.


File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\brady\OneDrive\Desktop\CEIS_110
C:\Users\brady\OneDrive\Desktop\CEIS_110\queryweatherdb.py
queryweatherdb.py
BuildWeatherdb.py
ExtractTemperatureandHumdity.py
boxplot2.py
linechart.py
humidtyhistogram.py
scatte
Source
Console
Object
Help
Variable Explorer
Plots
Files
Console 1/A

```python
# -*- coding: utf-8 -*-
"""
Created on Sat Jul 30 18:00:37 2022

@author: brady
"""

#Purpose: Query database using SQL
#Name: Brady Sisk
#Date: 7/30/22
#   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(relativeHumidity), max(relativeHumidity) FROM observations;"


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

Usage

Here you can get help of any object by pressing **Ctrl+I** in front of it, either on the Editor or the Console.

Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in *Preferences > Help*.

New to Spyder? Read our tutorial

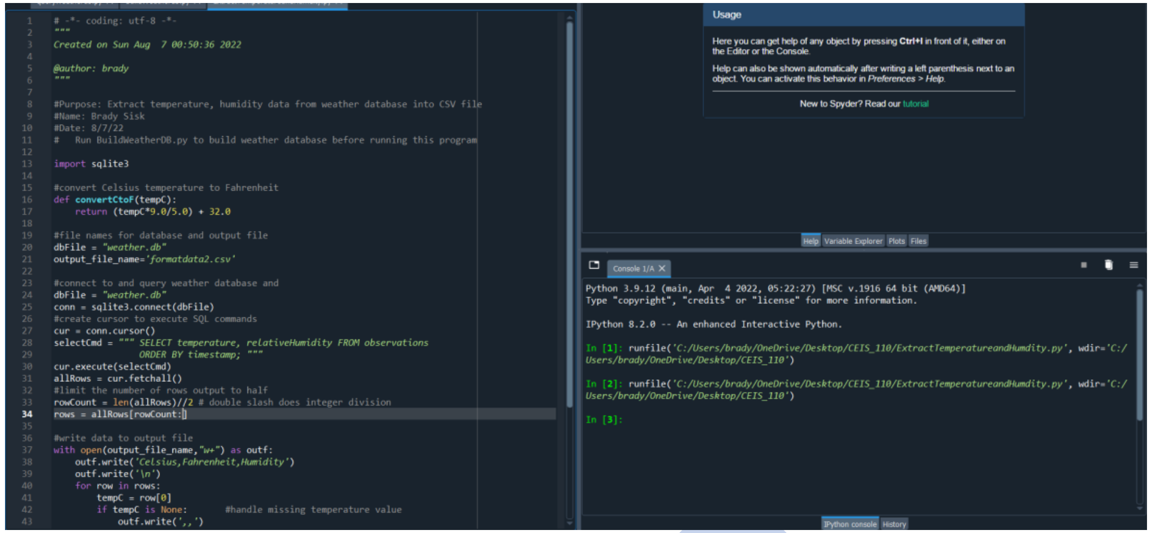
```
  File ~\anaconda3\lib\site-packages\pandas\io\sql.py:399 in read_sql_query
    return pandas_sql.read_query(

  File ~\anaconda3\lib\site-packages\pandas\io\sql.py:2080 in read_query
    cursor = self.execute(*args)

  File ~\anaconda3\lib\site-packages\pandas\io\sql.py:2032 in execute
    raise ex from exc

DatabaseError: Execution failed on sql ' SELECT min(humidity), max(humidity) FROM
observations;': no such column: humidity


In [4]: runfile('C:/Users/brady/OneDrive/Desktop/CEIS_110/queryweatherdb.py', wdir='C:/
Users/brady/OneDrive/Desktop/CEIS_110')
   min(relativeHumidity)  max(relativeHumidity)
0              17.303895              73.987358

In [5]:
```

Data Cleansing

- Data output from machines may have errors or extraneous data. When cleansing the data, programs can automatically put it in the format needed to be read by other programs.
- A python program is reading the data output by the python program and saving it in a csv file so that it can be read by Excel.
- Often data must be cleansed of spurious or missing values in a dataset. The data must be complete, valid, and standardized.

Extracting Temperature and Humidity using Python code

- The weather.db database may contain null or missing values. The code used retrieves only the temperature and humidity and writes them to a comma separated values (CSV) file. Two files are created – a formatdata.csv and formatdata.2 that each contain half of the rows. Missing or invalid values are not written to the file.



```
# -*- coding: utf-8 -*-
"""
Created on Sun Aug  7 00:50:36 2022

@author: brady
"""

#Purpose: Extract temperature, humidity data from weather database into CSV file
#Name: Brady Sisk
#Date: 8/7/22
#   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='formatdata2.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:
        tempC = row[0]
        if tempC is None:       #handle missing temperature value
            outf.write(',,')
```

Usage

Here you can get help of any object by pressing **Ctrl+I** in front of it, either on the Editor or the Console.

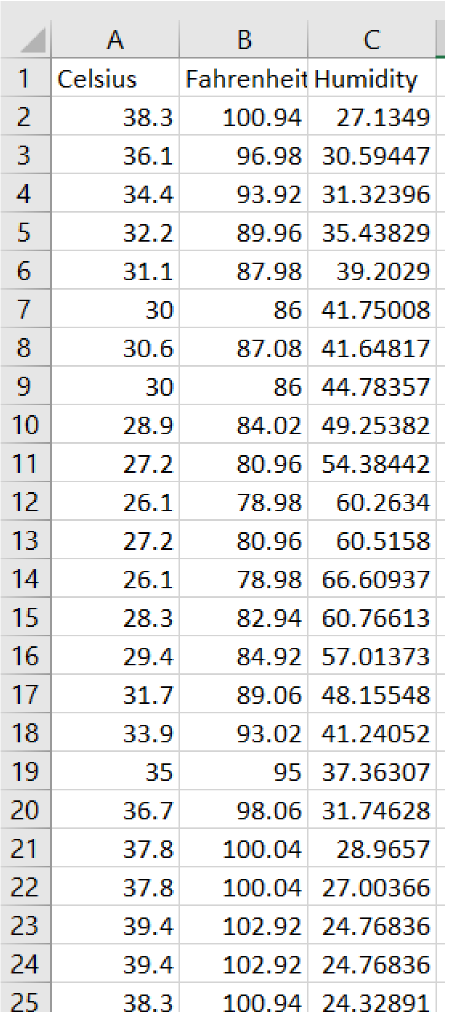
Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in *Preferences > Help*.

New to Spyder? Read our tutorial

```
Python 3.9.12 (main, Apr  4 2022, 05:22:27) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 8.2.0 -- An enhanced Interactive Python.

In [1]: runfile('C:/Users/brady/OneDrive/Desktop/CEIS_110/ExtractTemperatureandHumdity.py', wdir='C:/
Users/brady/OneDrive/Desktop/CEIS_110')

In [2]: runfile('C:/Users/brady/OneDrive/Desktop/CEIS_110/ExtractTemperatureandHumdity.py', wdir='C:/
Users/brady/OneDrive/Desktop/CEIS_110')

In [3]:
```

Data Formatted in an Excel Spreadsheet

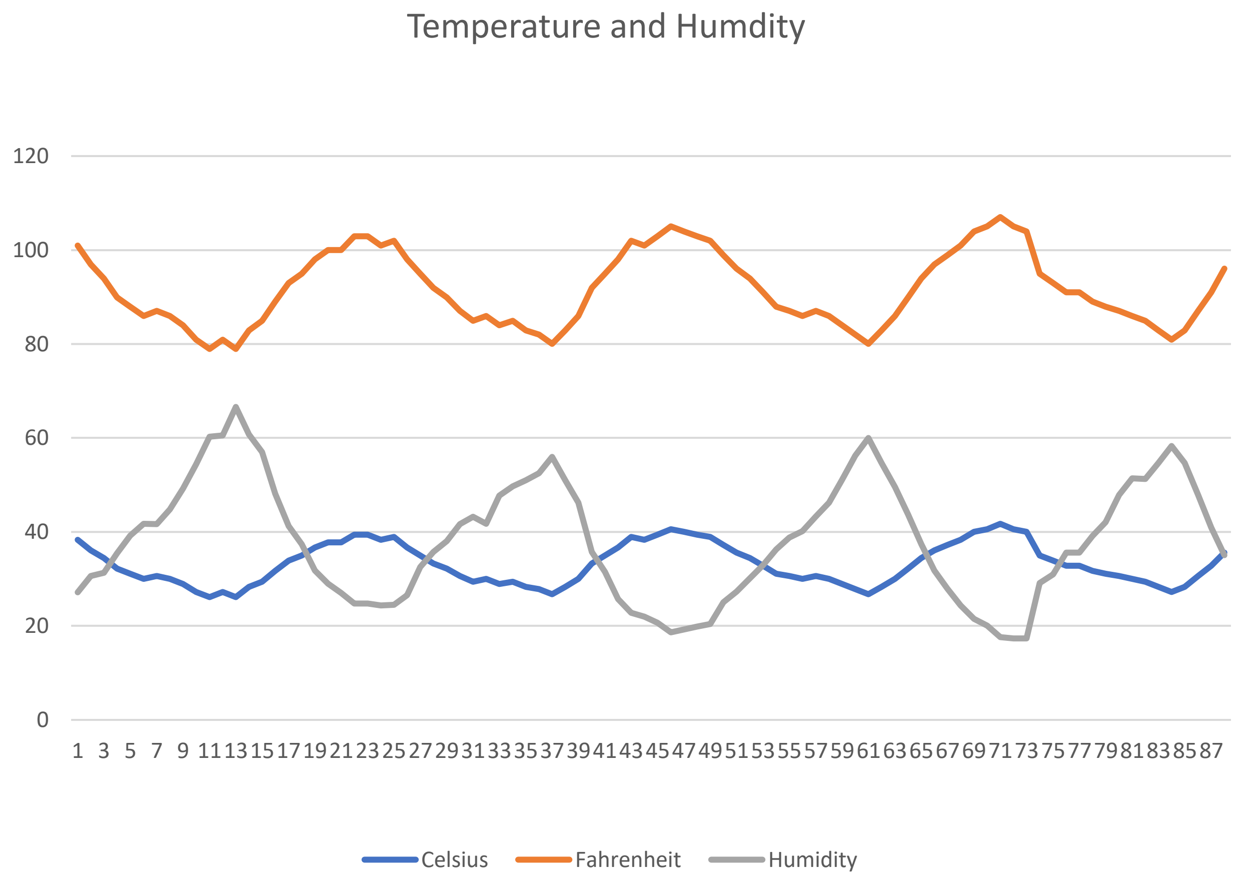
- The python program created a formatdata.csv file
- This file contains 3 columns: Celsius, Fahrenheit, and Humidity
- Statistics can be performed on this spreadsheet



	A	B	C
1	Celsius	Fahrenheit	Humidity
2	38.3	100.94	27.1349
3	36.1	96.98	30.59447
4	34.4	93.92	31.32396
5	32.2	89.96	35.43829
6	31.1	87.98	39.2029
7	30	86	41.75008
8	30.6	87.08	41.64817
9	30	86	44.78357
10	28.9	84.02	49.25382
11	27.2	80.96	54.38442
12	26.1	78.98	60.2634
13	27.2	80.96	60.5158
14	26.1	78.98	66.60937
15	28.3	82.94	60.76613
16	29.4	84.92	57.01373
17	31.7	89.06	48.15548
18	33.9	93.02	41.24052
19	35	95	37.36307
20	36.7	98.06	31.74628
21	37.8	100.04	28.9657
22	37.8	100.04	27.00366
23	39.4	102.92	24.76836
24	39.4	102.92	24.76836
25	38.3	100.94	24.32891

Data Visualization

A line chart was developed in Excel showing the Temperature and Humidity over Period 1

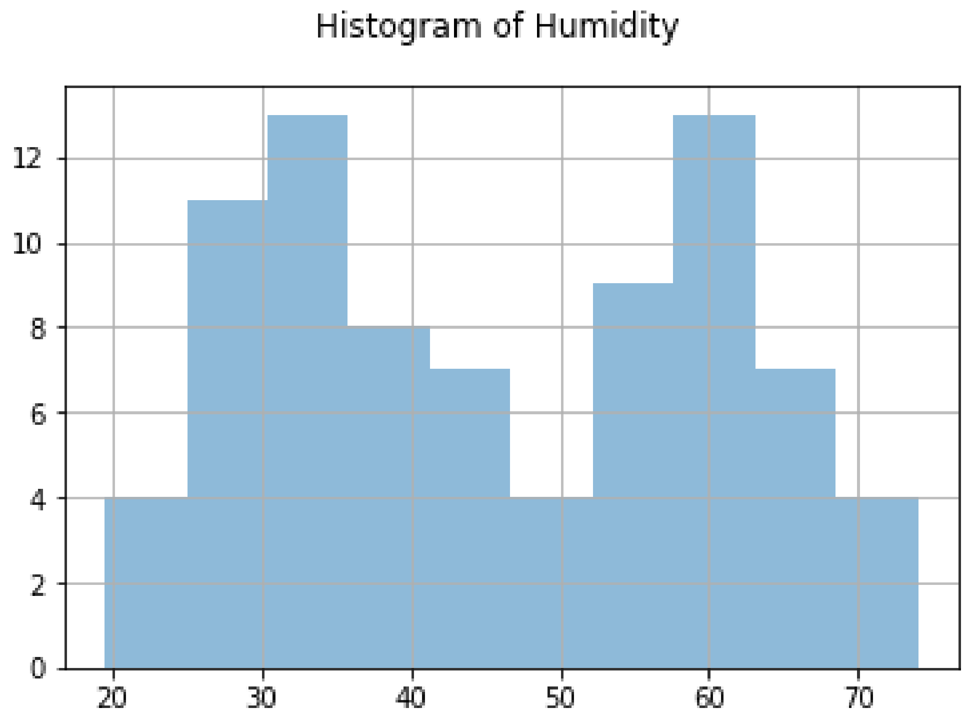


Data Analytics

- Python Data modules allows users to develop charts and graphs depicting data.
- The data sets can be manipulated as well and saved into tabular format
- The data analytics modules are available as part of Anaconda Several plots were generated looking at humidity and temperature
- Then a prediction was made about the data

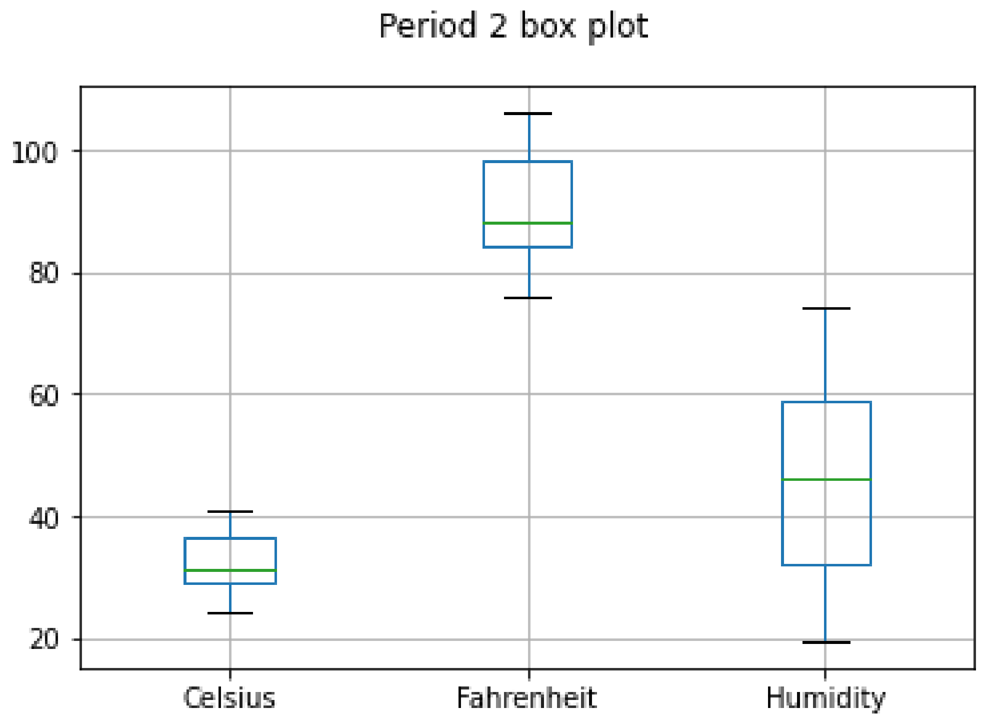
Histogram of Humidity

```
#Purpose: Create a histogram of humidity data from the second period
#Name: Brady Sisk
#Date: 8/13/22
import pandas as pd
import matplotlib.pyplot as plt
df1 = pd.read_csv("formatdata.csv")
df2 = pd.read_csv("formatdata2.csv")
df2['Humidity'].hist(bins=10, alpha=0.5); plt.suptitle('Histogram of Humidity')
```



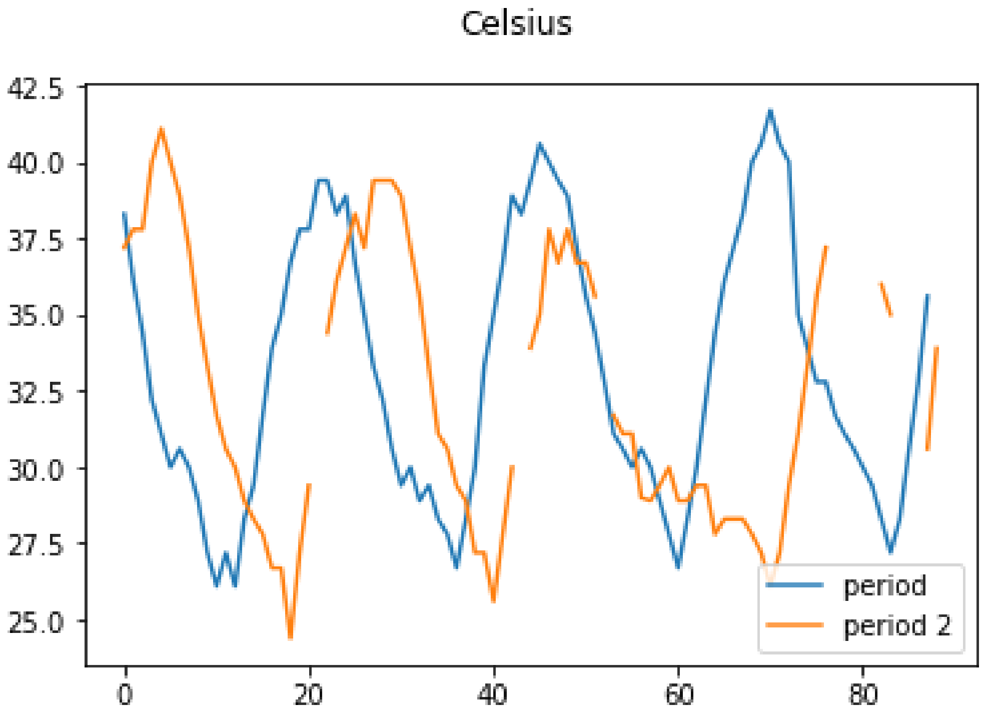
Period 2 Box Plot

```
#Purpose: Create box plot for period 2 data
#Name: Brady Sisk
#Date: 8/13/22
import pandas as pd
import matplotlib.pyplot as plt
df2 = pd.read_csv("formatdata2.csv")
df2.boxplot(); plt.suptitle('Period 2 box plot')
plt.show()
```



Analysis

- Think of your own question and create a chart/graph to answer it
- Your own question:
- Was there a big difference between period 1 and
- Period 2?
- Answer supported by Chart:
- No, judging by the chart I would say the periods were very similar



Code

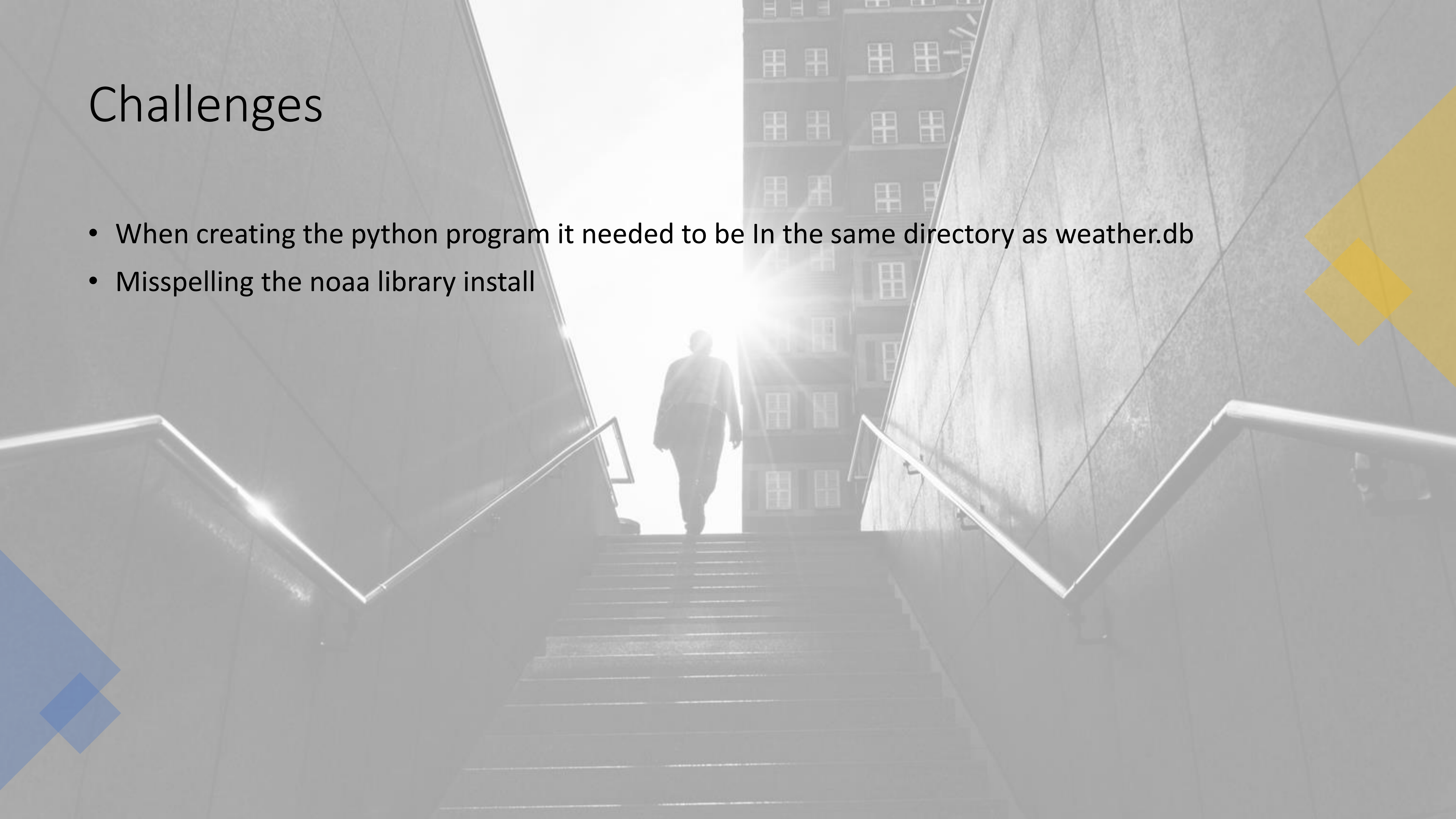
```
#Purpose: Create Celsius plot comparing period 1 and period 2
#Name: Brady Sisk
#Date: 8/13/22
import pandas as pd
import matplotlib.pyplot as plt
df1 = pd.read_csv("formatdata.csv") #baseline data is period 1 (older)
df2 = pd.read_csv("formatdata2.csv") #data for period 2 (more recent)
plt.figure(); df1.Celsius.plot(label = 'period '); df2.Celsius.plot(label = 'period 2'); plt.legend(loc='best'); plt.suptitle('Celsius')
plt.show()
```

Prediction

- Develop a prediction based on the data. What variations in temperature and humidity do you expect over the next few hours or days? How would humidity change if temperature goes up or down?

- Over the next few hours I would predict that the temperature will drop and the humidity rise.

- “As air temperature increases, air can hold more water molecules, and its relative humidity decreases. **When temperatures drop, relative humidity increases**”



Challenges

- When creating the python program it needed to be In the same directory as weather.db
- Misspelling the noaa library install

Career Skills

- Several career skills were gained in this project.
  - Communication – using flowcharts to depict the plan of a project
  - Database Development
  - Programming using python
  - Troubleshooting errors in the code and data cleansing
  - Analysis – Reviewing and graphs to make prediction on the data

Conclusion

- This project covered the fundamental topics of programming with data by using data gathered from cloud service to perform data analytics operations
- Building this project provided a hands on learning opportunity to put into practice the topics covered in the course.