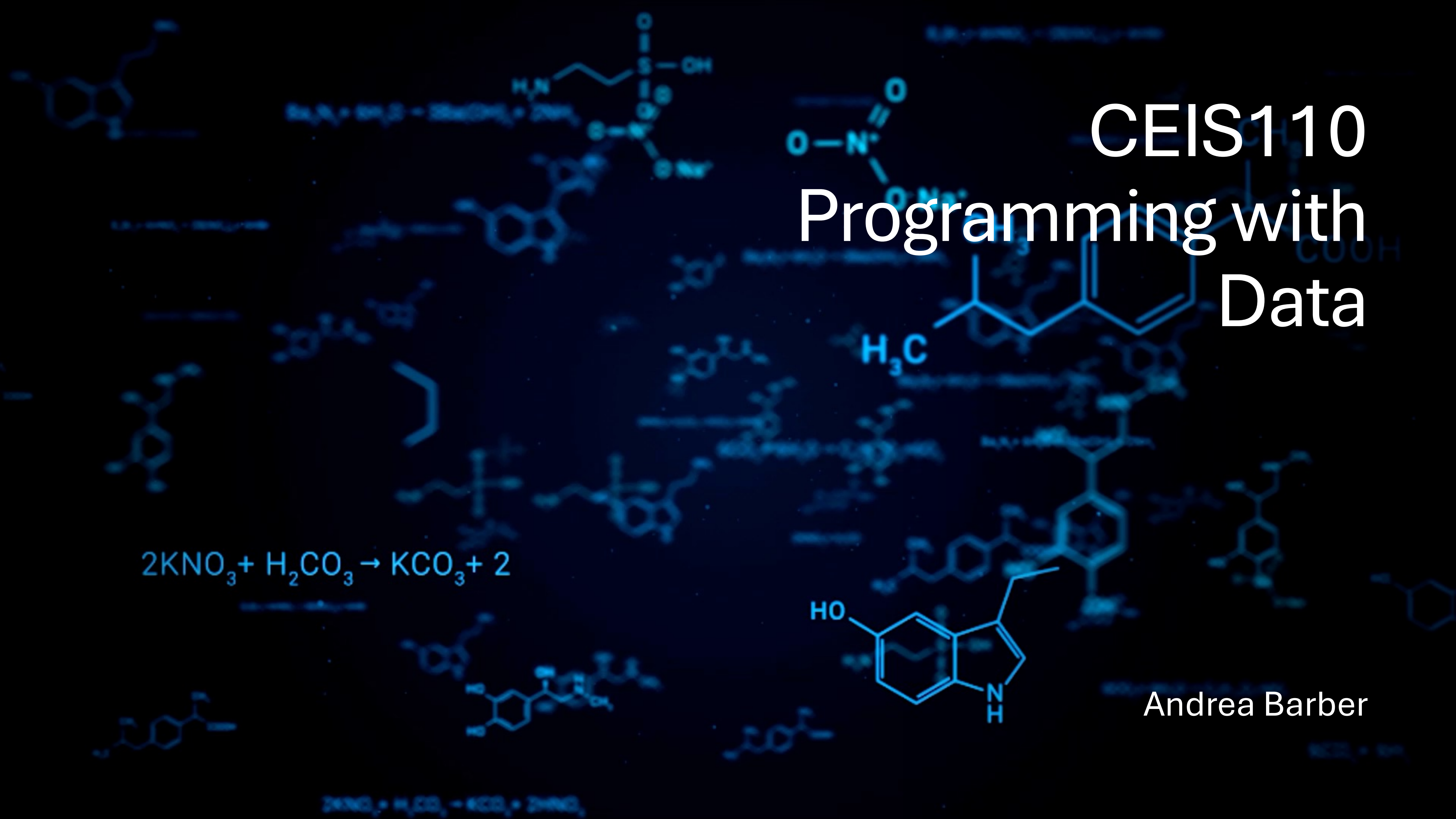


CEIS110 Programming with Data

Andrea Barber

Introduction

ONCE DATA IS COLLECTED IT TYPICALLY REQUIRES PROCESSING AND CLEANING TO BE USEFUL

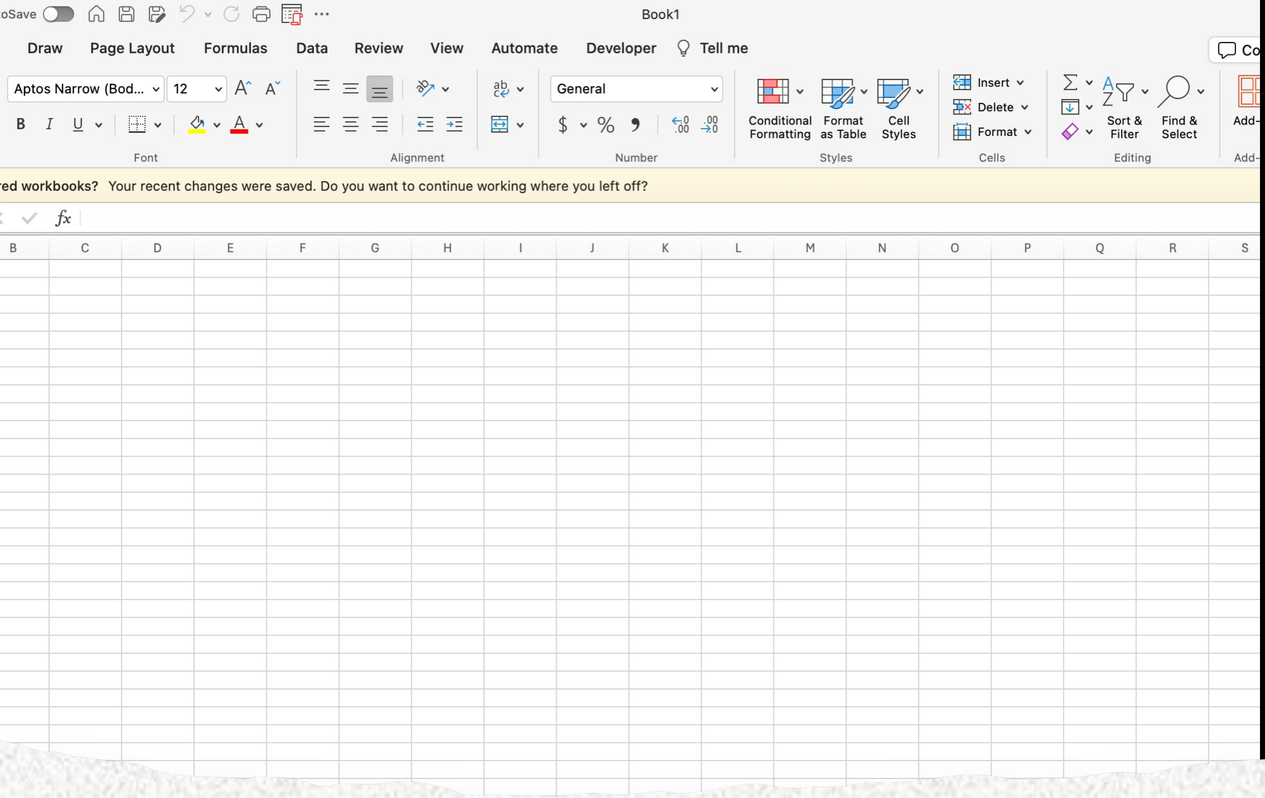
THIS PROJECT LEVERAGES CLOUD TECHNOLOGY TO MONITOR ENVIRONMENTAL CONDITIONS, FOCUSING ON TEMPERATURE AND HUMIDITY.

THE INFORMATION IS EXAMINED THROUGH CODING AND DATA ANALYSIS TECHNIQUES.

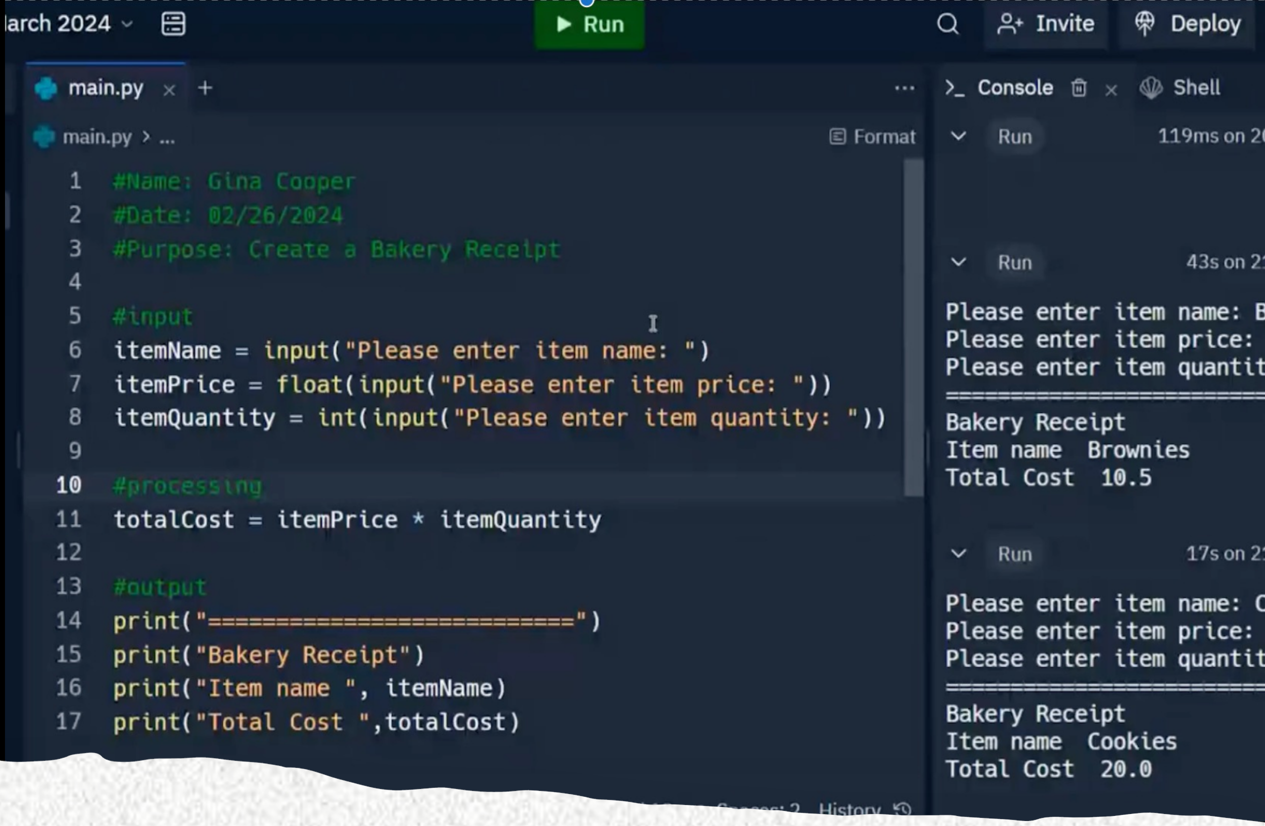
Software Requirements

- The first requirement in developing a project is to install the correct programming language.

- One popular programming language that can be used is Python. However, an analyst may use programming software such as Replit or Microsoft Excel.



ed workbooks? Your recent changes were saved. Do you want to continue working where you left off?



```python
#Name: Gina Cooper
#Date: 02/26/2024
#Purpose: Create a Bakery Receipt

#input
itemName = input("Please enter item name: ")
itemPrice = float(input("Please enter item price: "))
itemQuantity = int(input("Please enter item quantity: "))

#processing
totalCost = itemPrice * itemQuantity

#output
print("==========================")
print("Bakery Receipt")
print("Item name ", itemName)
print("Total Cost ",totalCost)
```

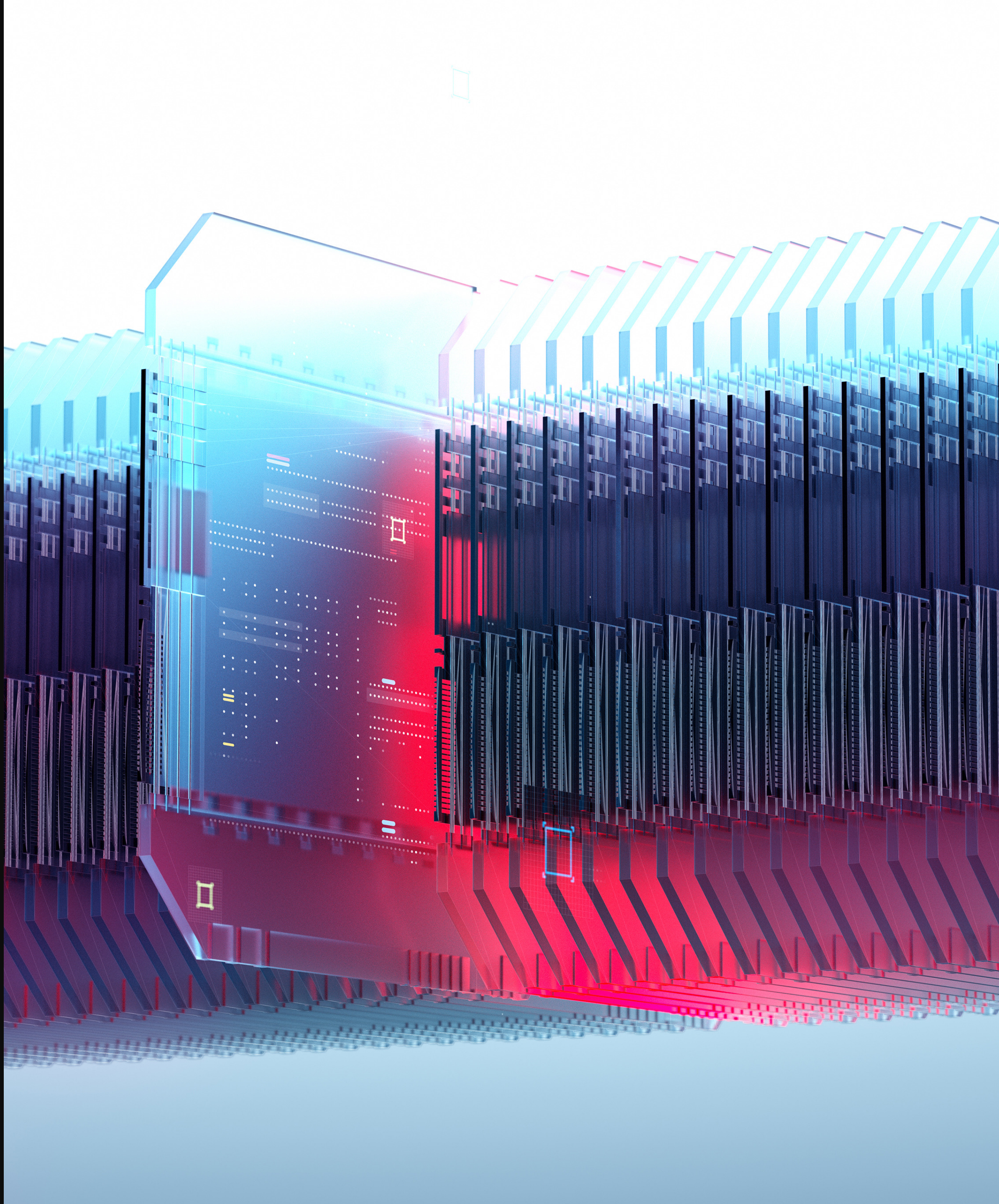
```
Run    119ms on 2

Run    43s on 2
Please enter item name: B
Please enter item price:
Please enter item quantit
=========================
Bakery Receipt
Item name  Brownies
Total Cost  10.5

Run    17s on 2
Please enter item name: C
Please enter item price:
Please enter item quantit
=========================
Bakery Receipt
Item name  Cookies
Total Cost  20.0
```

Software

The software required for the project includes Microsoft Excel and Replit.

Planning and Design

- Following a review of the required programs for the project, a strategy was devised for the temperature data initiative.
- To outline the project structure a flowchart was created.
- The blueprint and configuration of the project are essential phases for comprehending the progression framework.



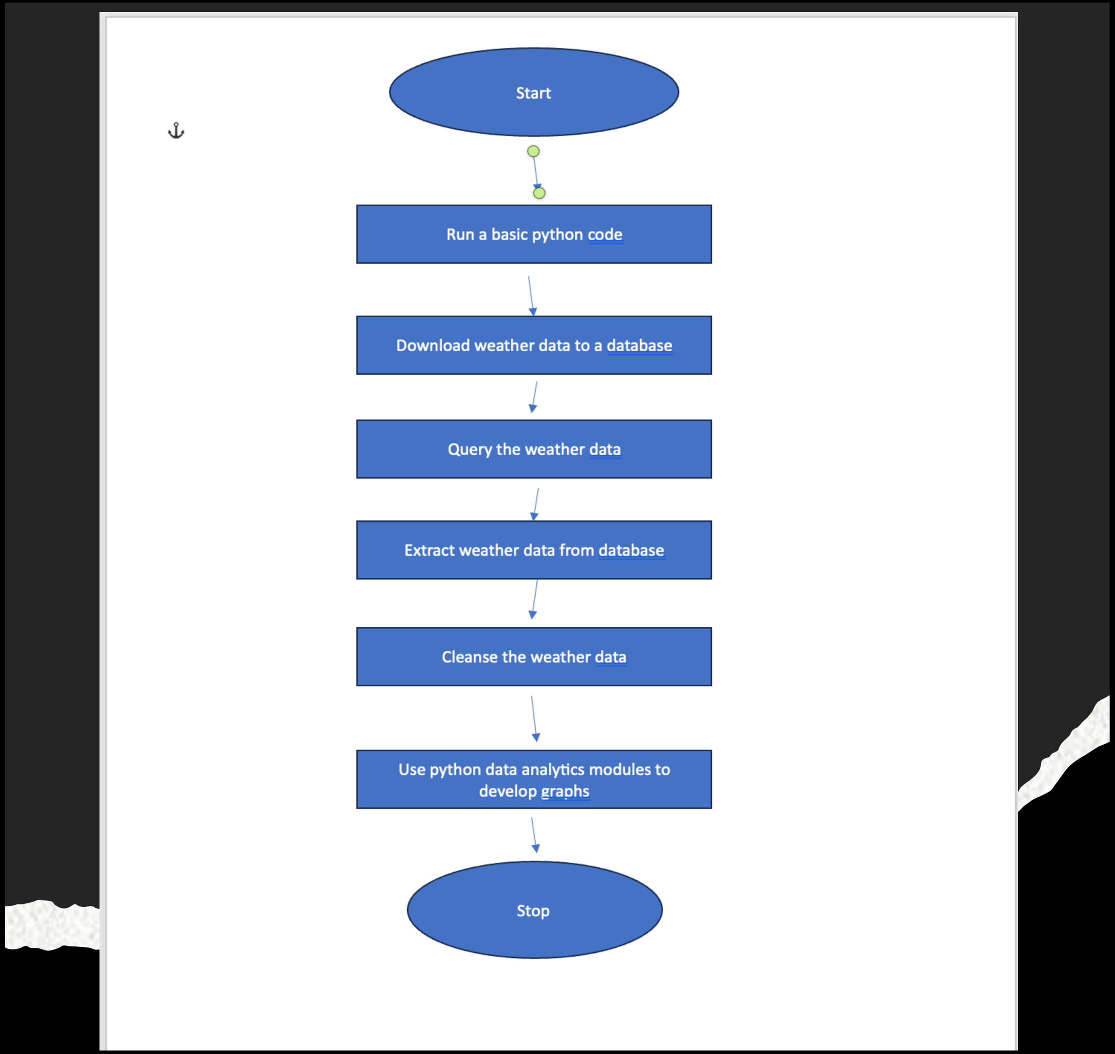
What are Flowcharts?

A Flowchart consists of standard geometric symbols that graphically indicate the actions to be executed and the exact order in which those actions should be executed

Flowcharts help companies visualize the steps involved in processing data from input to output.

Flowchart steps

1. Run a basic Python program
2. Download weather data to a database
3. Query the weather data
4. Extract weather data from database into a comma-separated (CSV) file with Python
5. Cleanse weather data
6. Use Python data analytics modules to develop graphical models

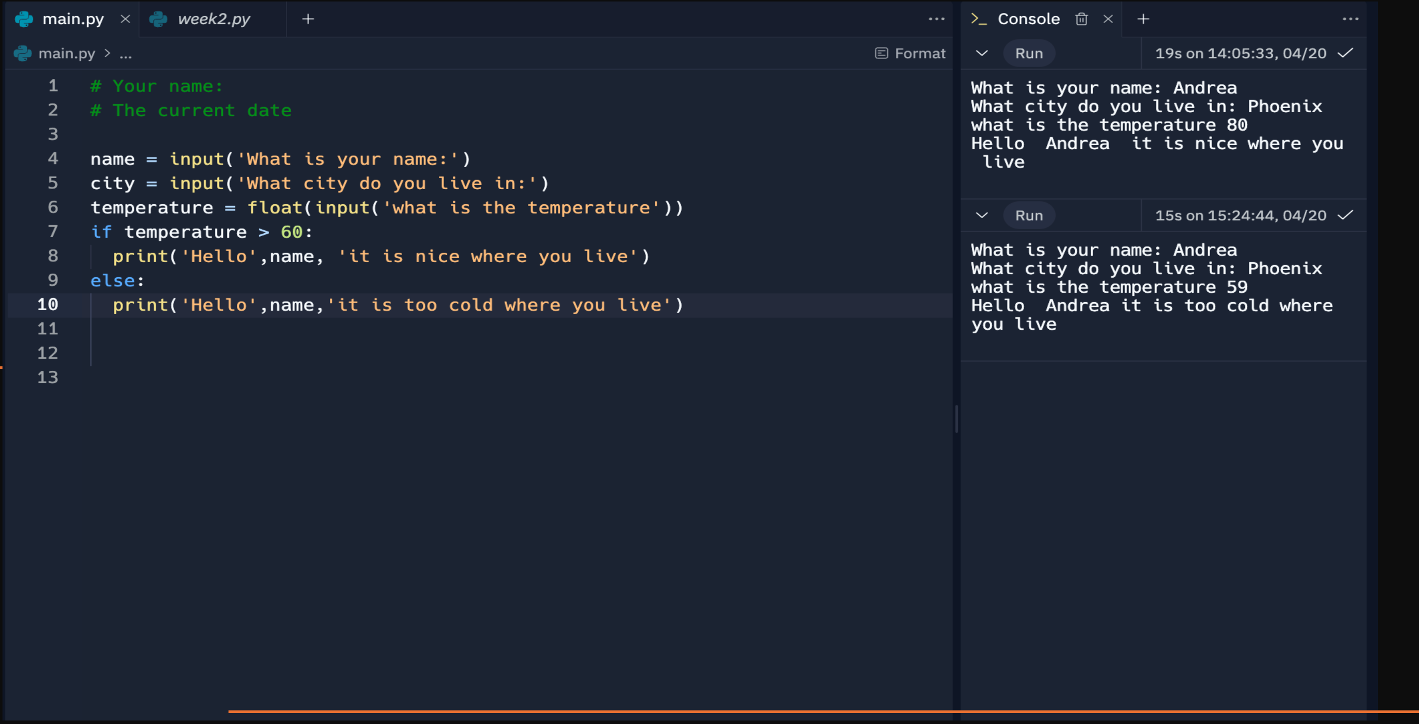

Start
Run a basic python code
Download weather data to a database
Query the weather data
Extract weather data from database
Cleanse the weather data
Use python data analytics modules to develop graphs
Stop

Introduction to Python

To write a program in Python you need an integrated development environment also known as an IDE.

Python combines an editor, debugger, and programming aid in one package.

Spyder, Visual Studio Code, Pycharm, and other tools are also available.



```python
# Your name:
# The current date

name = input('What is your name:')
city = input('What city do you live in:')
temperature = float(input('what is the temperature'))
if temperature > 60:
  print('Hello',name, 'it is nice where you live')
else:
  print('Hello',name,'it is too cold where you live')
```

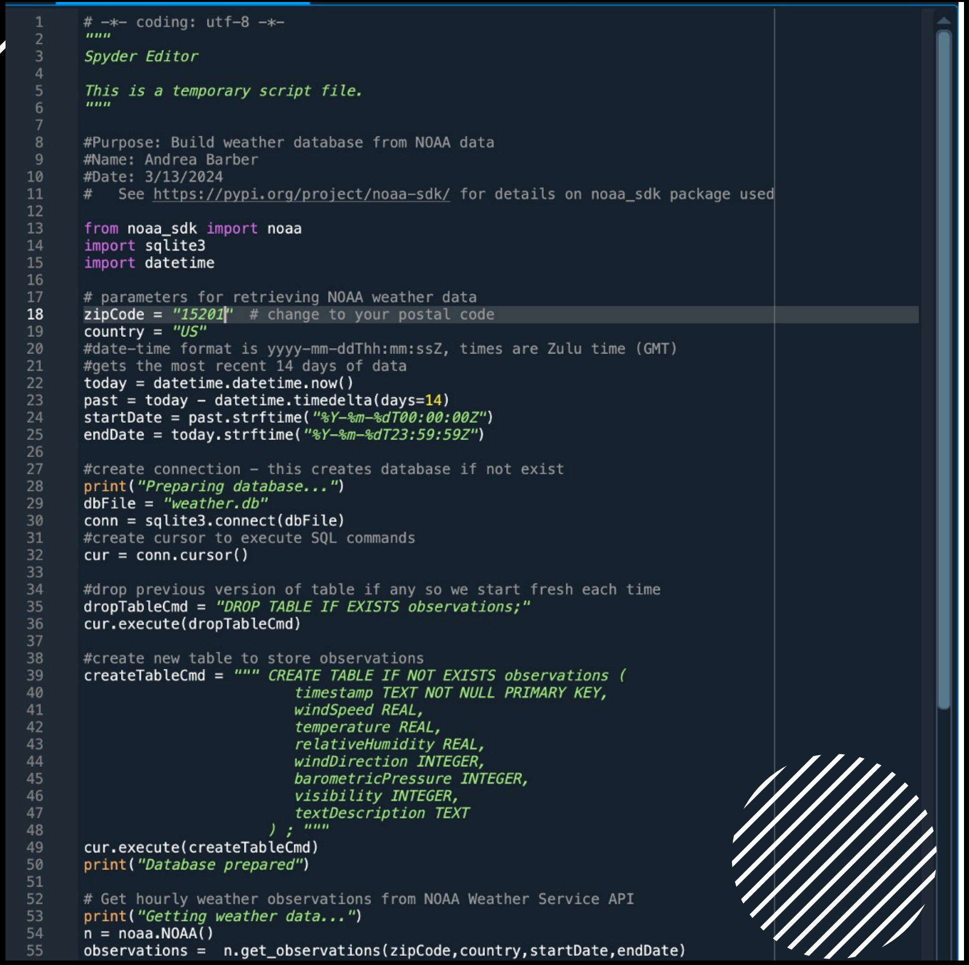
```
What is your name: Andrea
What city do you live in: Phoenix
what is the temperature 80
Hello  Andrea  it is nice where you
 live
```

```
What is your name: Andrea
What city do you live in: Phoenix
what is the temperature 59
Hello  Andrea it is too cold where
you live
```

Gathering Temperature and Humidity data

- Following the strategy and blueprinting phase, the program was created to retrieve a collection of weather data.
- The information was saved in a nearby data repository in a tabular format for subsequent examination in Replit.

BuildWeatherDb.py Code (Screenshot)

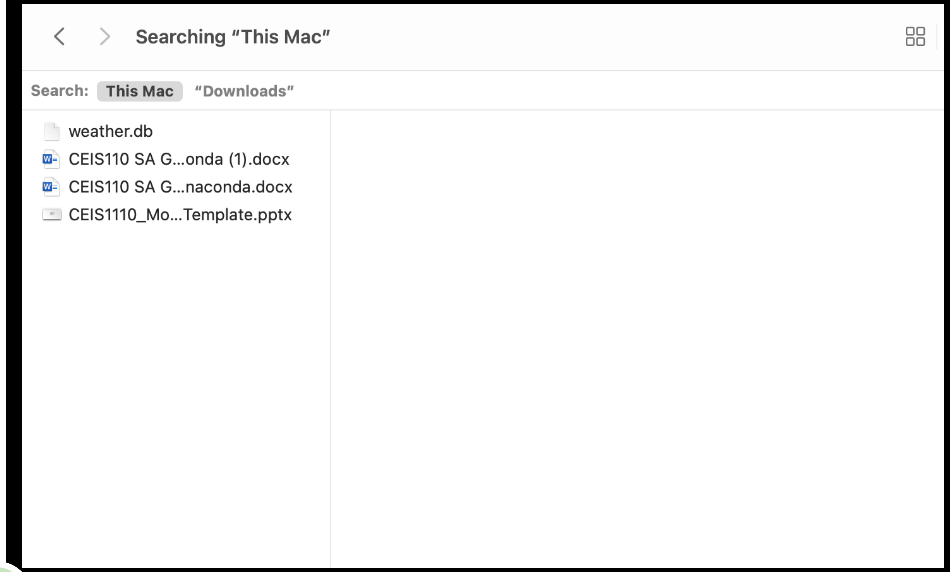


```python
# -*- coding: utf-8 -*-
"""
Spyder Editor

This is a temporary script file.
"""

#Purpose: Build weather database from NOAA data
#Name: Andrea Barber
#Date: 3/13/2024
#   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 = "15201"  # 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")

# Get hourly weather observations from NOAA Weather Service API
print("Getting weather data...")
n = noaa.NOAA()
observations =  n.get_observations(zipCode,country,startDate,endDate)
```

An image capture of the code in Replit

The code will create a table named observations with the following fields: timestamp, windspeed, temperature, relativehumidity, windDirection, barometricpressur e, visibility and textDirection.

The database will be called "weather.db" and placed in the same folder as the Replit repository. script.


Searching "This Mac"

Search: This Mac "Downloads"

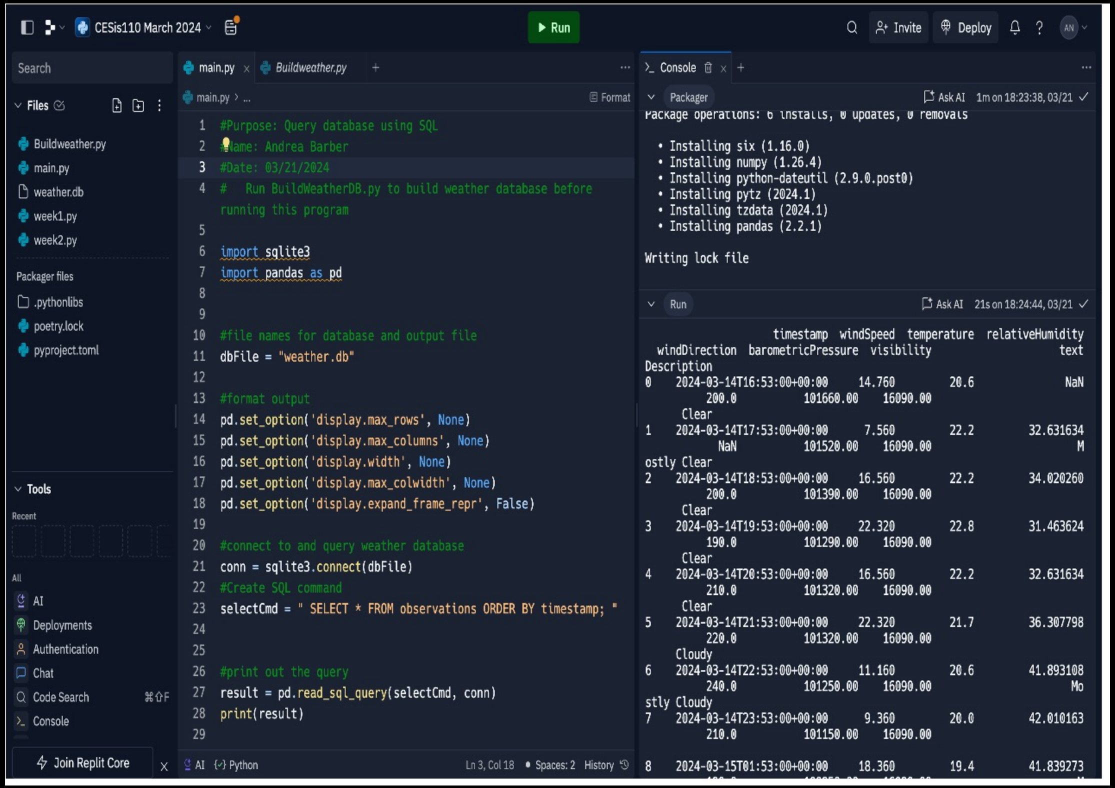
weather.db
CEIS110 SA G...onda (1).docx
CEIS110 SA G...naconda.docx
CEIS1110_Mo...Template.pptx

Weather.db File (Screenshot)

Screenshot of Filepath way on Mac OS showing database file Weather.db was created

Querying the Database

- Structure Query Language(SQL) is a specialized programming language for working with a relational database.
- Programs written in a general-purpose programming language like Python, issue SQL commands to the database “under the hood” and receive and display the results to the user.



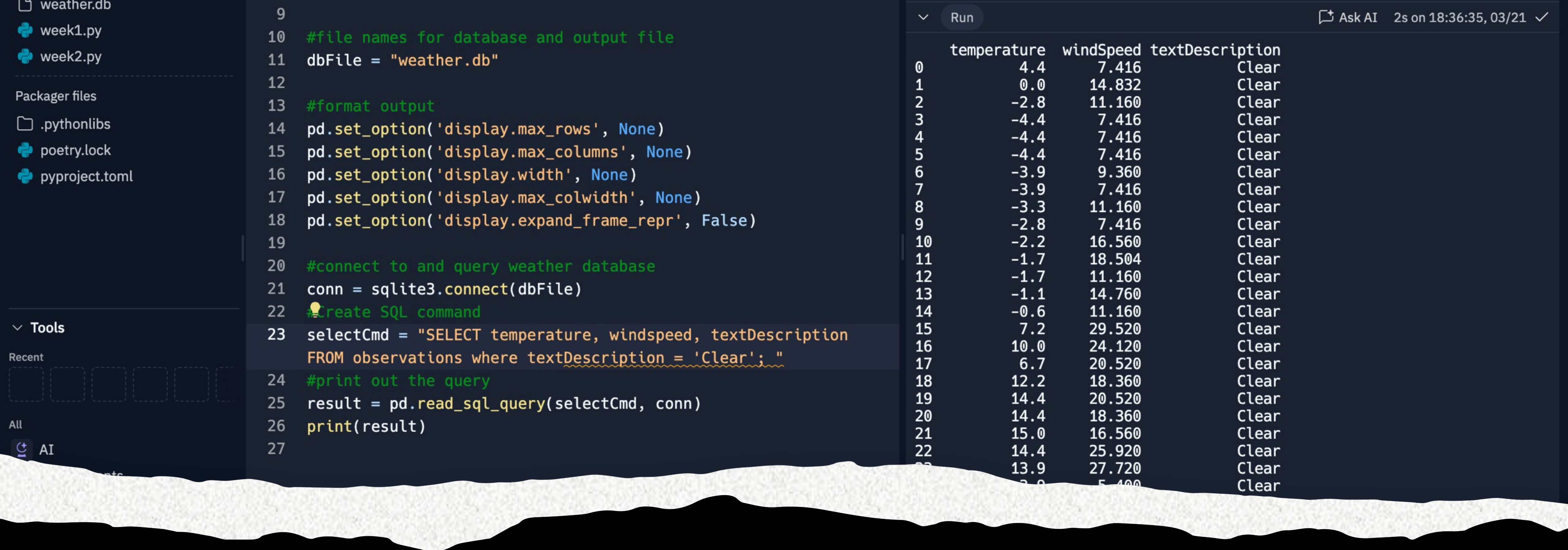
```python
#Purpose: Query database using SQL
#Name: Andrea Barber
#Date: 03/21/2024
#   Run BuildWeatherDB.py to build weather database before running this program

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 * FROM observations ORDER BY timestamp; "


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

```
Package operations: 6 installs, 0 updates, 0 removals

  • Installing six (1.16.0)
  • Installing numpy (1.26.4)
  • Installing python-dateutil (2.9.0.post0)
  • Installing pytz (2024.1)
  • Installing tzdata (2024.1)
  • Installing pandas (2.2.1)

Writing lock file
```

```
                     timestamp  windSpeed  temperature  relativeHumidity
  windDirection  barometricPressure  visibility                     text
Description
0    2024-03-14T16:53:00+00:00     14.760         20.6               NaN
          200.0           101660.00    16090.00
     Clear
1    2024-03-14T17:53:00+00:00      7.560         22.2         32.631634
            NaN           101520.00    16090.00                        M
ostly Clear
2    2024-03-14T18:53:00+00:00     16.560         22.2         34.020260
          200.0           101390.00    16090.00
     Clear
3    2024-03-14T19:53:00+00:00     22.320         22.8         31.463624
          190.0           101290.00    16090.00
     Clear
4    2024-03-14T20:53:00+00:00     16.560         22.2         32.631634
          210.0           101320.00    16090.00
     Clear
5    2024-03-14T21:53:00+00:00     22.320         21.7         36.307798
          220.0           101320.00    16090.00
    Cloudy
6    2024-03-14T22:53:00+00:00     11.160         20.6         41.893108
          240.0           101250.00    16090.00                       Mo
stly Cloudy
7    2024-03-14T23:53:00+00:00      9.360         20.0         42.010163
          210.0           101150.00    16090.00


8    2024-03-15T01:53:00+00:00     18.360         19.4         41.839273
```

Query to retrieve all columns and all rows (Screenshot)

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

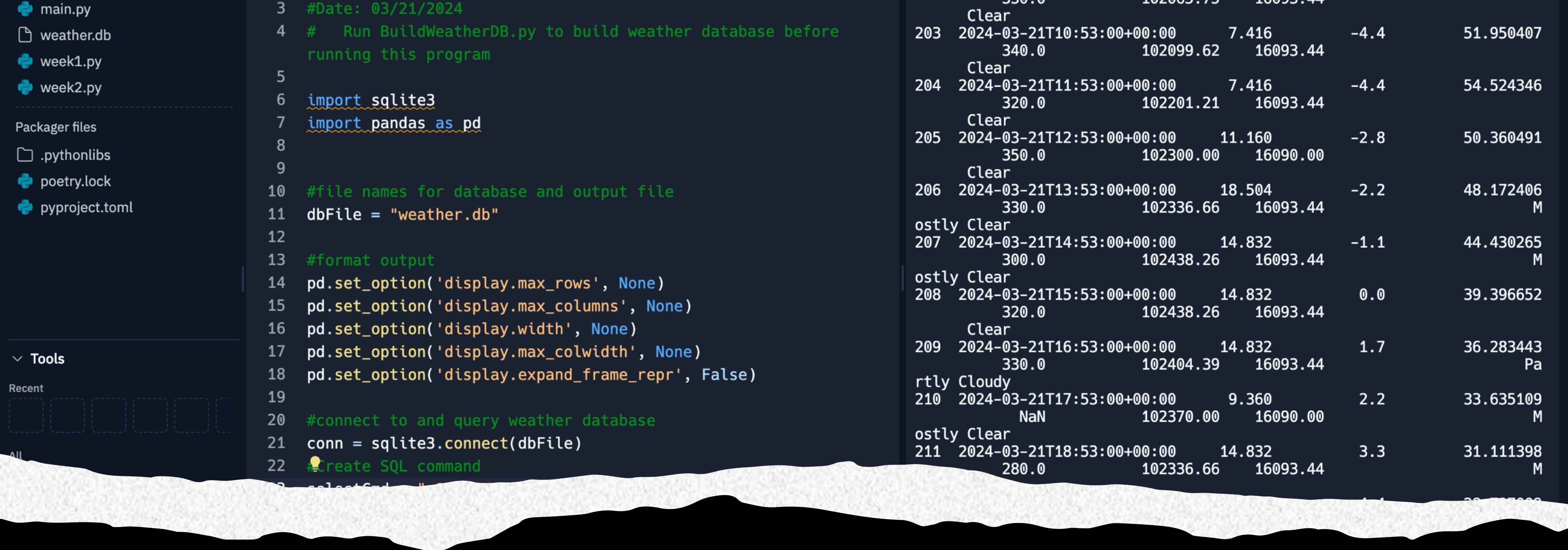


```
#file names for database and output file
dbFile = "weather.db"

#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 temperature, windspeed, textDescription FROM observations where textDescription = 'Clear'; "
#print out the query
result = pd.read_sql_query(selectCmd, conn)
print(result)
```

```
    temperature  windSpeed textDescription
0           4.4      7.416           Clear
1           0.0     14.832           Clear
2          -2.8     11.160           Clear
3          -4.4      7.416           Clear
4          -4.4      7.416           Clear
5          -4.4      7.416           Clear
6          -3.9      9.360           Clear
7          -3.9      7.416           Clear
8          -3.3     11.160           Clear
9          -2.8      7.416           Clear
10         -2.2     16.560           Clear
11         -1.7     18.504           Clear
12         -1.7     11.160           Clear
13         -1.1     14.760           Clear
14         -0.6     11.160           Clear
15          7.2     29.520           Clear
16         10.0     24.120           Clear
17          6.7     20.520           Clear
18         12.2     18.360           Clear
19         14.4     20.520           Clear
20         14.4     18.360           Clear
21         15.0     16.560           Clear
22         14.4     25.920           Clear
           13.9     27.720           Clear
                                     Clear
```

Query to retrieve lowest and highest temperatures (Screenshot)

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



```python
#Date: 03/21/2024
#   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
```

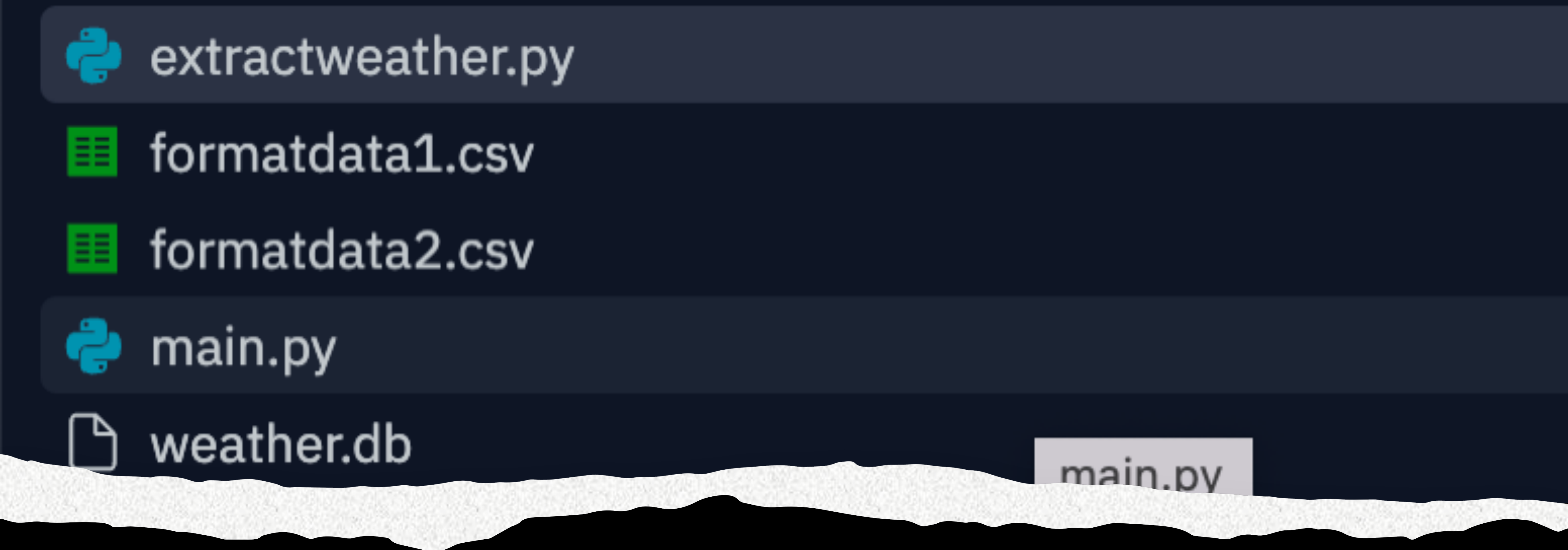
```
 Clear
203  2024-03-21T10:53:00+00:00      7.416        -4.4        51.950407
        340.0          102099.62    16093.44
 Clear
204  2024-03-21T11:53:00+00:00      7.416        -4.4        54.524346
        320.0          102201.21    16093.44
 Clear
205  2024-03-21T12:53:00+00:00     11.160        -2.8        50.360491
        350.0          102300.00    16090.00
 Clear
206  2024-03-21T13:53:00+00:00     18.504        -2.2        48.172406
        330.0          102336.66    16093.44                        M
ostly Clear
207  2024-03-21T14:53:00+00:00     14.832        -1.1        44.430265
        300.0          102438.26    16093.44                        M
ostly Clear
208  2024-03-21T15:53:00+00:00     14.832         0.0        39.396652
        320.0          102438.26    16093.44
 Clear
209  2024-03-21T16:53:00+00:00     14.832         1.7        36.283443
        330.0          102404.39    16093.44                       Pa
rtly Cloudy
210  2024-03-21T17:53:00+00:00      9.360         2.2        33.635109
          NaN          102370.00    16090.00                        M
ostly Clear
211  2024-03-21T18:53:00+00:00     14.832         3.3        31.111398
        280.0          102336.66    16093.44                        M
```

Query to retrieve the data when the weather is clear (Screenshot)

A different request was made to obtain the temperature, wind speed, and text description when the weather is clear

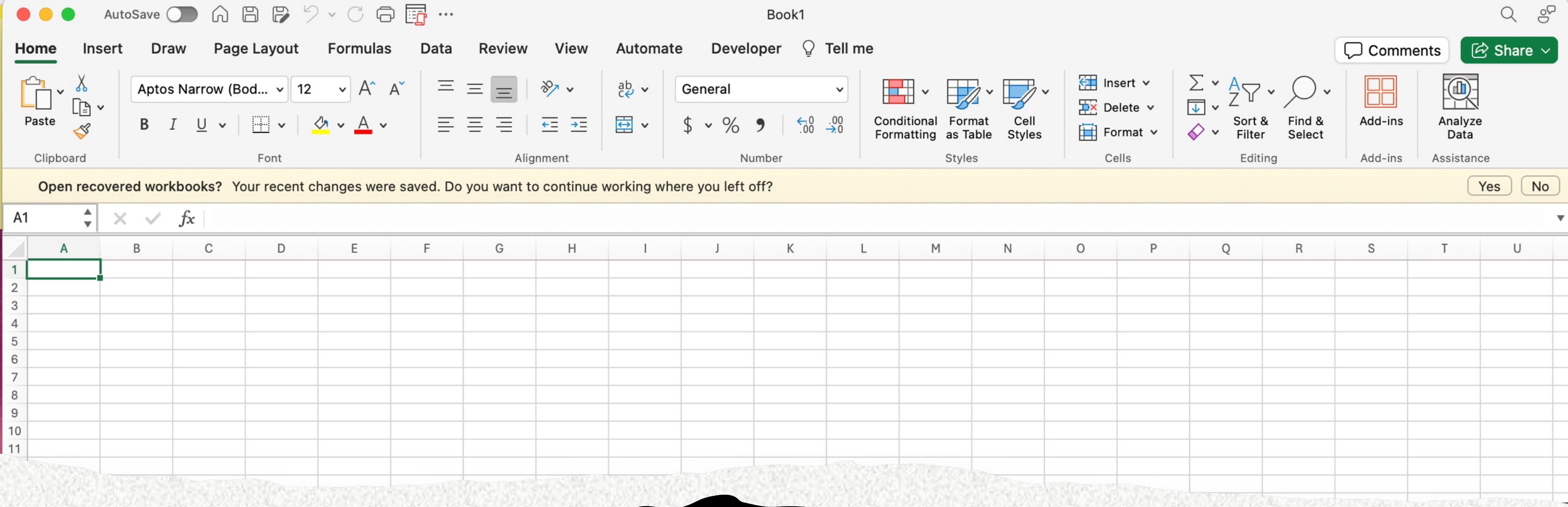
Data cleansing

- When processing data from machines, inaccuracies or irrelevant information may occur. Data-cleaning tools can automatically convert the information into the correct format for other applications to use.
- A Python script is processing the data generated by the code and storing it in a CSV file for Excel compatibility.
- Data frequently requires cleaning to remove errors or fill in gaps. It must be accurate, consistent, and uniform.


extractweather.py
formatdata1.csv
formatdata2.csv
main.py
weather.db

Extracting Temperature and Humidity using Python code

- The weather.db data repository might include empty or absent records. The script extracts just the temperature and humidity readings and exports them to a CSV file. Two separate files are generated, formatdata1.csv and formatdata2.csv, each holding half of the rows. Absent and incorrect values are also logged in the output.



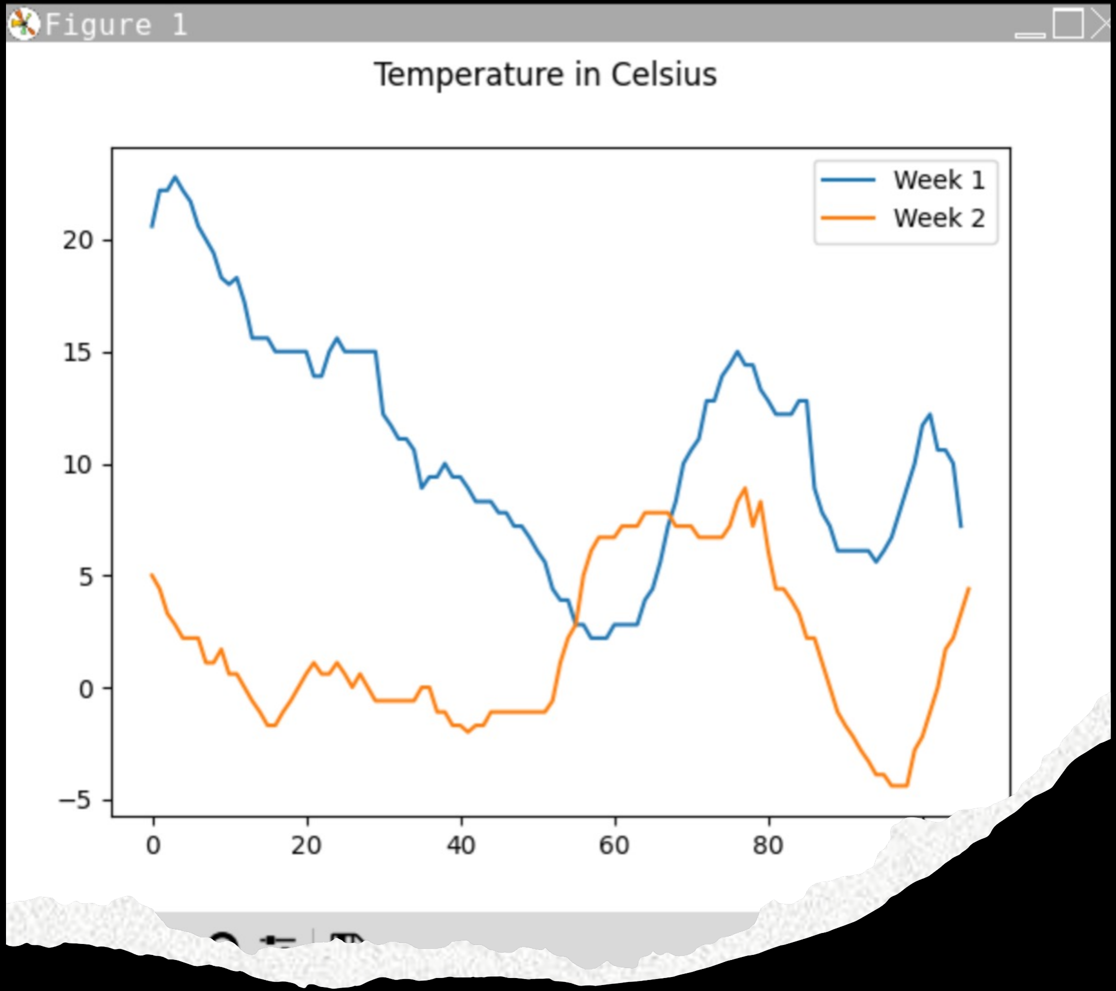
Open recovered workbooks? Your recent changes were saved. Do you want to continue working where you left off?

Data Formatted in an Microsoft Excel Spreadsheet

- The Python program created a formatdata1.csv file
- This contains 3 columns: Celsius, Fahrenheit, and Humidity
- Statistics can be performed on this spreadsheet

Data Visualization

A Line chart was created in Microsoft Excel showing the Temperature and Humidity over Period 1.

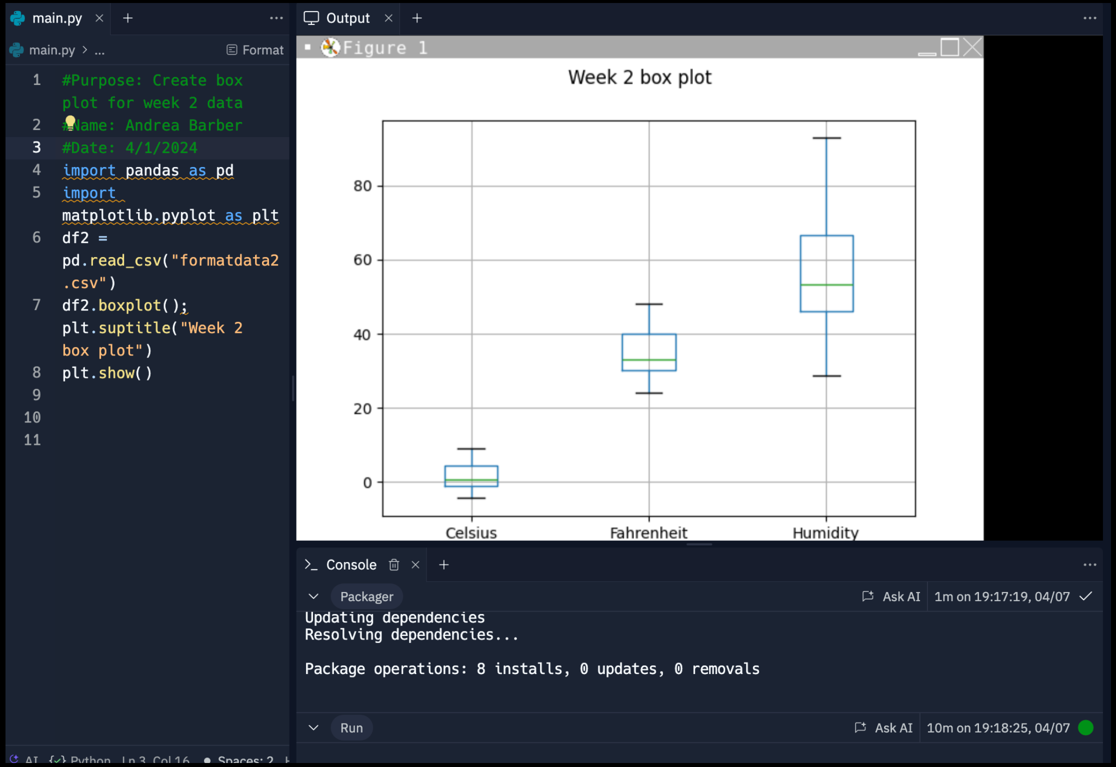


Data Analytics

- Python data analysis packages enable users to create visual representations, like charts and graphs, to illustrate data
- The dataset can also be modified and stored in a table-like structure
- The data analysis components are accessible through Replit
- Multiple graphs were created to examine temperature and humidity
- Subsequent forecasts were derived from the collected information

Histogram of Humidity

- #Purpose: Create a histogram of humidity data from the second period
- #Name: Andrea Barber
- #Date 4/7/2024
- 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()
- Print(df2.info())
- Print(df2.describe())
- Print(“The Median is”, df2.median())

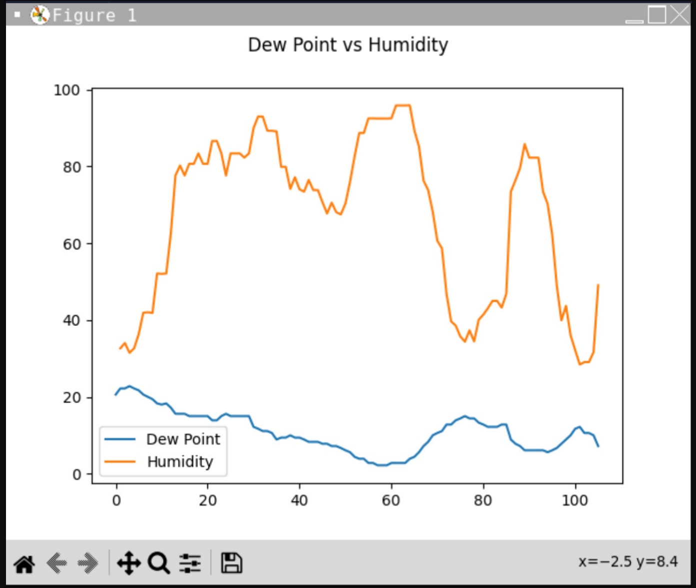


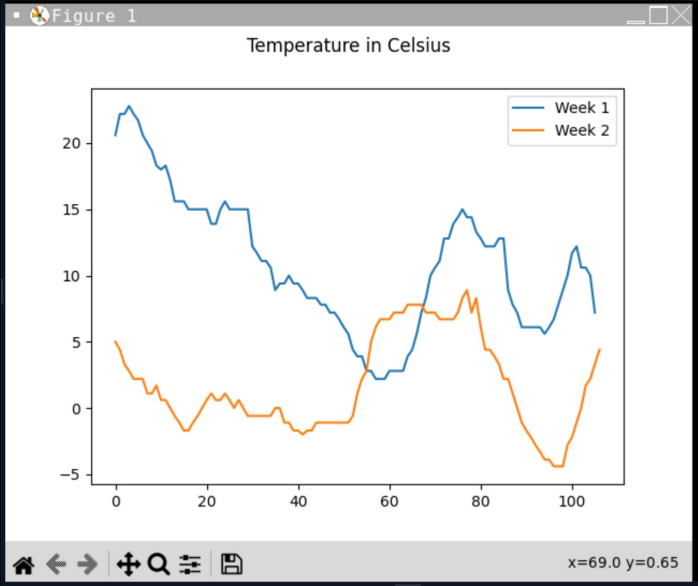
```
#Purpose: Create box plot for week 2 data
#Name: Andrea Barber
#Date: 4/1/2024
import pandas as pd
import matplotlib.pyplot as plt
df2 = pd.read_csv("formatdata2.csv")
df2.boxplot(); plt.suptitle("Week 2 box plot")
plt.show()
```

```
Updating dependencies
Resolving dependencies...

Package operations: 8 installs, 0 updates, 0 removals
```

Analysis

- The following phase in the examination involved formulating a query and utilizing the information presented in the graphs to address it.
- My Question: What is the impact of rising temperatures on the dew point and relative humidity?
- Two graphs were generated to display the temperature in Fahrenheit and humidity levels, using information collected during the first two weeks.
- Answer supported by Chart:
  - As humidity peaks, the dew point does not necessarily increase, which could be attributed to temperature changes affecting the air's capacity to hold moisture.





Code

The code to create the Fahrenheit plot is below:

Import pandas as pd

Import matplotlin.pyplot as plt

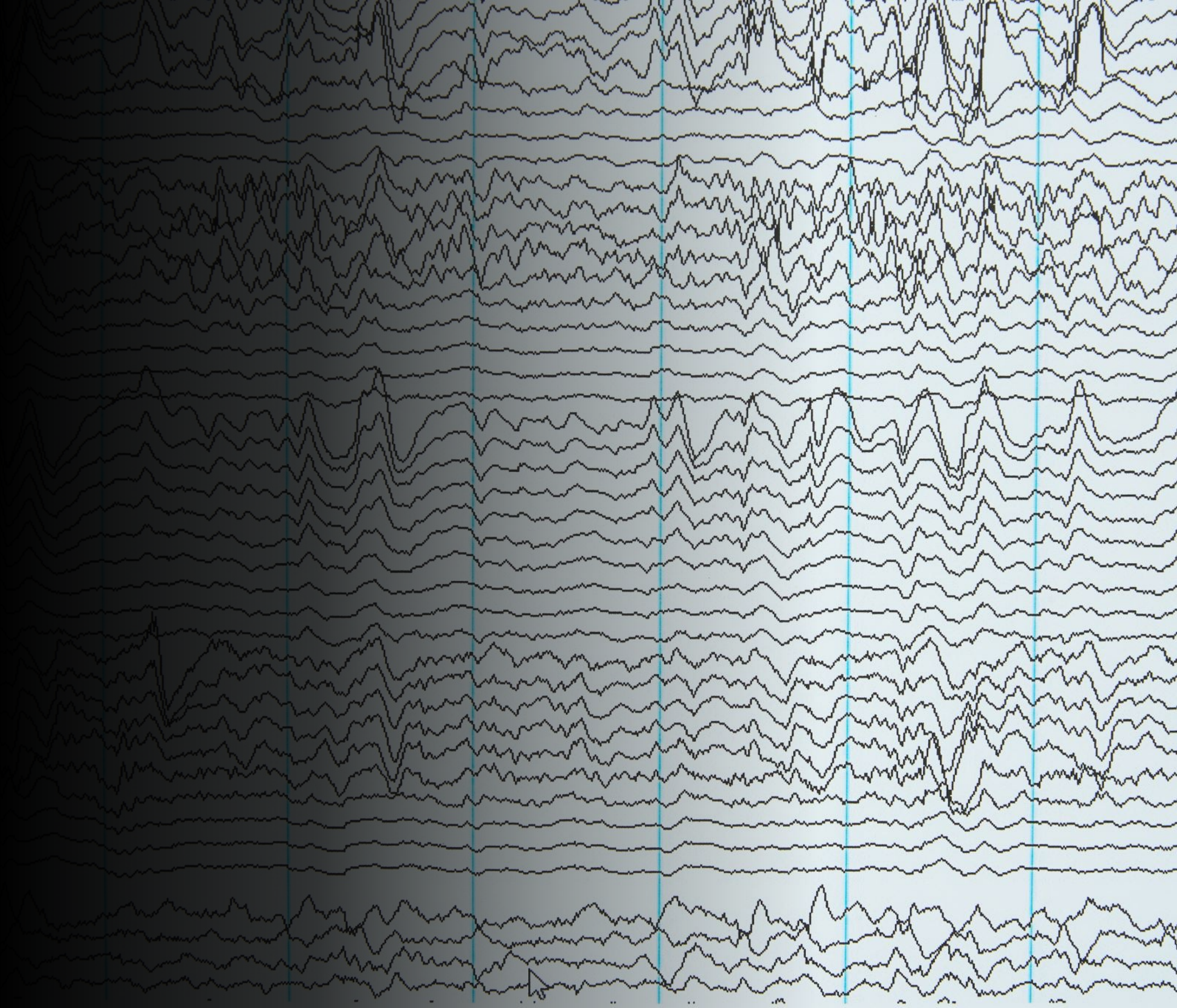
Df2= pd.read_csv(”formatdata1.csv”)

Df2= pd.read_csv((”formatdata2.csv”)

Plt.figure();df2.Fahrenheit.plot(label=’Farenheit”);df2.Humidity.plot(label = “Humidity”),

Plt.legend(loc=‘best’);plt.subptitle(‘Trail 2 – Temperature vs Humidity’)

Plt.show()



Prediction

if the temperature starts to climb, expect the humidity to drop, since warm air can hold more water. But if it cools down, the humidity will go up, and we might even see some dew on the grass or fog over the fields over the next few weeks.

Challenges

- When creating the Python program it needed to be in the same repository.
- Misspelling certain words in for the library install

Career Skills

- Using flowcharts to plan project
- Database development
- Troubleshooting errors in the code and data cleansing
- Programming using Python
- Analysis by creating and reviewing charts and graphs to make predictions using data acquired

Conclusion

- This project explored the core concepts of coding with data by utilizing information collected from the cloud platform to facilitate data analysis tasks
- Creating this project offered a practical learning experience to apply the concepts and abilities explored this semester