Following are three trends based on the data:

1. Cities, with latitude closer to the equator, get very hot. Maximum temperature is recorded in cities, whose latitude is close to the equator.
2. Cities, with latitude closer to the equator, are less windy. Wind speed is < 8 mps for most cities closer to the equator.
3. Cities, with latitude closer to the equator, gets very humid. Humidity is > 50% for most cities closer to the equator.

Following is the Code (same as Jupyter notebook). Each section is separated by page break

# Dependencies and Setup

import matplotlib.pyplot as plt

import pandas as pd

import numpy as np

import requests

import time

import json

import csv

# Import API key, Jupyter is working funky with multiple open files, so renamed the api key file

from my\_api\_keys import api\_key

# Incorporated citipy to determine city based on latitude and longitude

from citipy import citipy

# Output File (CSV)

output\_data\_file = "output\_data/cities.csv"

# Range of latitudes and longitudes

lat\_range = (-90, 90)

lng\_range = (-180, 180)

# List for holding lat\_lngs and cities

lat\_lngs = []

cities = []

# Create a set of random lat and lng combinations

lats = np.random.uniform(low=-90.000, high=90.000, size=1500)

lngs = np.random.uniform(low=-180.000, high=180.000, size=1500)

lat\_lngs = zip(lats, lngs)

# Identify nearest city for each lat, lng combination

for lat\_lng in lat\_lngs:

city = citipy.nearest\_city(lat\_lng[0], lat\_lng[1]).city\_name

# If the city is unique, then add it to a our cities list

if city not in cities:

cities.append(city)

# Print the city count to confirm sufficient count

len(cities)

# Create an empty list to store the responses

response\_json = []

# Save config information

url = "http://api.openweathermap.org/data/2.5/weather?"

# Make a request for each of the indices

for x in range(len(cities)):

query\_url = url + "appid=" + api\_key + "&q=" + str(cities[x])

print(f"Processing Record {x} | city {cities[x]}")

# Get one of the posts

post\_response = requests.get(query\_url)

# Save post's JSON

response\_json.append(post\_response.json())

if (x != 0 and x%60 == 0) :

time.sleep(60)

# write response\_json to csv file

df = pd.DataFrame(response\_json)

df.to\_csv(output\_data\_file, encoding="utf-8", index=False)

city\_list = []

country\_list = []

cloud\_list = []

date\_list = []

humid\_list = []

lat\_list = []

lng\_list = []

max\_temp\_list = []

wind\_speed\_list = []

for index in range(len(response\_json)):

if (response\_json[index]["cod"] == 200) :

city\_list.append(response\_json[index]["name"])

country\_list.append(response\_json[index]["sys"]["country"])

cloud\_list.append(response\_json[index]["clouds"]["all"])

date\_list.append(response\_json[index]["dt"])

humid\_list.append(response\_json[index]["main"]["humidity"])

lat\_list.append(response\_json[index]["coord"]["lat"])

lng\_list.append(response\_json[index]["coord"]["lon"])

max\_temp\_list.append(response\_json[index]["main"]["temp\_max"])

wind\_speed\_list.append(response\_json[index]["wind"]["speed"])

my\_df = pd.DataFrame({"City" : city\_list, "Cloudiness" : cloud\_list, "Country" : country\_list, "Date" : date\_list,

"Humidity" : humid\_list, "Lat" : lat\_list, "Lng" : lng\_list,

"Max Temp" : max\_temp\_list, "Wind Speed" : wind\_speed\_list})

my\_df.head()

#Plot Latitude vs. Max Temperature

plt.scatter(my\_df["Lat"], my\_df["Max Temp"], marker="o", facecolors="red", edgecolors="black", s=np.pi\*5, alpha=0.5)

plt.title('City Latitude vs. Max Temperature (07/04/2019)')

plt.xlabel('Latitude')

plt.ylabel('Max Temperature')

# Save Figure

plt.savefig("Images/AshokWeatherScatter1.png")

plt.show()

#Plot Latitude vs. Humidity

plt.scatter(my\_df["Lat"], my\_df["Humidity"], marker="o", facecolors="aqua", edgecolors="black", s=np.pi\*5, alpha=0.5)

plt.title('City Latitude vs. Humidity (07/04/2019)')

plt.xlabel('Latitude')

plt.ylabel('Humidity (%)')

# Save Figure

plt.savefig("Images/AshokWeatherScatter2.png")

plt.show()

#Plot Latitude vs. Cloudiness

plt.scatter(my\_df["Lat"], my\_df["Cloudiness"], marker="o", facecolors="blue", edgecolors="black", s=np.pi\*5, alpha=0.5)

plt.title('City Latitude vs. Cloudiness (07/04/2019)')

plt.xlabel('Latitude')

plt.ylabel('Cloudiness (%)')

# Save Figure

plt.savefig("Images/AshokWeatherScatter3.png")

plt.show()

#Plot Latitude vs. Wind Speed

plt.scatter(my\_df["Lat"], my\_df["Wind Speed"], marker="o", facecolors="green", edgecolors="black", s=np.pi\*5, alpha=0.5)

plt.title('City Latitude vs. Wind Speed (07/04/2019)')

plt.xlabel('Latitude')

plt.ylabel('Wind Speed (mph)')

# Save Figure

plt.savefig("Images/AshokWeatherScatter4.png")

plt.show()