In [1]:

```
# Dependencies
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import requests
import json

# OpenWeather API Key
api_key = "d9cc2b43f4569d1bd791efaeac304dc7"
units = "metric"
```

In [2]:

```
# Import cities file as DataFrame
cities_pd = pd.read_csv("worldcities.csv")
cities_pd.head()
```

Out[2]:

	city	city_ascii	lat	lng	pop	country	iso2	iso3	province
0	Qal eh-ye Now	Qal eh-ye	34.983000	63.133300	2997.0	Afghanistan	AF	AFG	Badghis
1	Chaghcharan	Chaghcharan	34.516701	65.250001	15000.0	Afghanistan	AF	AFG	Ghor
2	Lashkar Gah	Lashkar Gah	31.582998	64.360000	201546.0	Afghanistan	AF	AFG	Hilmand
3	Zaranj	Zaranj	31.112001	61.886998	49851.0	Afghanistan	AF	AFG	Nimroz
4	Tarin Kowt	Tarin Kowt	32.633298	65.866699	10000.0	Afghanistan	AF	AFG	Uruzgan

In [3]:

```
# Add columns for Temperature, Humidity, Cloudiness, Wind Speed
# Note that we used "" to specify initial entry.
cities_pd["Temperature_C"] = ""
cities_pd["Temperature_F"] = ""
cities_pd["Humidity"] = ""
cities_pd["Cloudiness"] = ""
cities_pd["Wind_Speed"] = ""

cities_pd.head()
```

Out[3]:

	city	city_ascii	lat	lng	pop	country	iso2	iso3	province	Temperature_C	Temperature_F	Humidity
0	Qal eh-ye Now	Qal eh-ye	34.983000	63.133300	2997.0	Afghanistan	AF	AFG	Badghis			
1	Chaghcharan	Chaghcharan	34.516701	65.250001	15000.0	Afghanistan	AF	AFG	Ghor			
2	Lashkar Gah	Lashkar Gah	31.582998	64.360000	201546.0	Afghanistan	AF	AFG	Hilmand			
3	Zaranj	Zaranj	31.112001	61.886998	49851.0	Afghanistan	AF	AFG	Nimroz			
4	Tarin Kowt	Tarin Kowt	32.633298	65.866699	10000.0	Afghanistan	AF	AFG	Uruzgan			

In [4]:

```
# Random selection of 500 cities with a population >1000
selected_cities = cities_pd.sample(n=500)
selected_cities = selected_cities[selected_cities["pop"].astype(int) > 1000]

# View selected_cities
selected_cities.head()
```

Out[4]:

	city	city_ascii	lat	lng	pop	country	iso2	iso3	province	Temperature_C	Temperature_F	Humid
4706	Roxas	Roxas	11.585273	122.751101	91880.5	Philippines	PH	PHL	Capiz			
3575	Ayakoz	Ayakoz	47.964732	80.429705	39670.0	Kazakhstan	KZ	KAZ	East Kazakhstan			
1047	Obidos	Obidos	-1.910027	-55.520007	26278.5	Brazil	BR	BRA	Pará			
4505	Gujranwala	Gujranwala	32.160426	74.185022	1448735.5	Pakistan	PK	PAK	Punjab			
3863	Majuro	Majuro	7.103004	171.380000	22950.0	Marshall Islands	MH	MHL	NaN			

In [5]:
```
# Loop through the cities_pd and run a temp search for each city
for index, row in cities_pd.iterrows():
    target_url = "http://api.openweathermap.org/data/2.5/weather?"

    # Build query URL
    query_url = target_url + "appid=" + api_key + "&q=" + (row["city"]
    cities_data = requests.get(query_url).json()
    try:
        selected_cities.set_value(index, "Temperature_C", cities_data["list"][0]["main"]["temp"])
        selected_cities.set_value(index, "Humidity", cities_data["list"][0]["main"]["humidity"])
        selected_cities.set_value(index, "Cloudiness", cities_data["list"][0]["main"]["clouds"])
        selected_cities.set_value(index, "Wind_Speed", cities_data["list"][0]["main"]["wind"]["speed"])
        print(target_url)

    except:
        #print("Error with data. Skipping")
        continue

# View selected_cities for added data
selected_cities.head()
```

```
  File "<ipython-input-5-62b40c529d41>", line 7
    cities_data = requests.get(query_url).json()
              ^
SyntaxError: invalid syntax
```

In [ ]:
```
# Convert Celcius to Fahrenheit
        cel=int("Temperature_C")
        far=(9/5*(cel))+32
        selected_cities.set_value(index, "Temperature_F", far)
selected_cities.head()
```

In [ ]:
```
# Build a scatter plot for Temperature and Latitude
plt.scatter(selected_cities["Temperature_F"],
            selected_cities["Latitude"],
            edgecolor="black", linewidths=1, marker="o",
            alpha=0.8, label="City")

# Incorporate the other graph properties
plt.title("Temperature vs. Latitude by City")
plt.ylabel("Temperature_F")
plt.xlabel("Latitude")
plt.grid(True)
plt.xlim([-2.5, 150])
plt.ylim([-2.5, 110000])

# Save the figure
plt.savefig("output_analysis/Temperature_Latitude.png")

# Show plot
plt.show()
```

In [ ]:
```
# Build a scatter plot for Humidity and Latitude
plt.scatter(selected_cities["Humidity"],
            selected_cities["Latitude"],
            edgecolor="black", linewidths=1, marker="o",
            alpha=0.8, label="City")

# Incorporate the other graph properties
plt.title("Humidity vs. Latitude by City")
plt.ylabel("Humidity")
plt.xlabel("Latitude")
plt.grid(True)
plt.xlim([-2.5, 150])
plt.ylim([-2.5, 110000])

# Save the figure
plt.savefig("output_analysis/Humidity_Latitude.png")

# Show plot
plt.show()
```

In [ ]:

```
# Build a scatter plot for Cloudiness and Latitude
plt.scatter(selected_cities["Cloudiness"],
            selected_cities["Latitude"],
            edgecolor="black", linewidths=1, marker="o",
            alpha=0.8, label="City")

# Incorporate the other graph properties
plt.title("Cloudiness vs. Latitude by City")
plt.ylabel("Cloudiness")
plt.xlabel("Latitude")
plt.grid(True)
plt.xlim([-2.5, 150])
plt.ylim([-2.5, 110000])

# Save the figure
plt.savefig("output_analysis/Cloudiness_Latitude.png")

# Show plot
plt.show()
```

In [ ]:

```
# Build a scatter plot for Wind_Speed and Latitude
plt.scatter(selected_cities["Wind_Speed"],
            selected_cities["Latitude"],
            edgecolor="black", linewidths=1, marker="o",
            alpha=0.8, label="City")

# Incorporate the other graph properties
plt.title("Wind_Speed vs. Latitude by City")
plt.ylabel("Wind_Speed")
plt.xlabel("Latitude")
plt.grid(True)
plt.xlim([-2.5, 150])
plt.ylim([-2.5, 110000])

# Save the figure
plt.savefig("output_analysis/Wind_Speed_Latitude.png")

# Show plot
plt.show()
```