WeatherPy

Note

- Instructions have been included for each segment. You do not have to follow them exactly, but they are included to help you think through the steps.

In [1]:
```
import sys
sys.path.append("/anaconda3/lib/python3.7/site-packages")
from citipy import citipy
```

In [2]:
```
# Dependencies and Setup
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
import requests
import time


# Import API key
from api_keys import api_key
# !pip install citipy

# 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)
```

In [ ]:
```

```

In [ ]:
```

```

Generate Cities List

In [3]:
```
# 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)
```

Out[3]:
```
604
```

Perform API Calls

- Perform a weather check on each city using a series of successive API calls.
- Include a print log of each city as it'sbeing processed (with the city number and city name).

In [4]:
```
new_cities = []
cloudiness = []
country = []
date = []
humidity = []
temp = []
lat = []
lng = []
wind = []
```

In [5]:
```
record_counter = 0
set_counter = 0
# Starting URL for Weather Map API Call
url = "http://api.openweathermap.org/data/2.5/weather?units=Imperial&APPID=
print('-------------------------')
print('Beginning Data Retrieval')
print('-------------------------')

for city in cities:
    query_url = url + "&q=" + city
    # Get weather data
    response = requests.get(query_url).json()
    if record_counter < 50:
        record_counter += 1
    else:
        set_counter += 1
        record_counter = 0

    print('Processing record {} of set {} | {}'.format(record_counter, set_
    print(url)
    try:
        cloudiness.append(response['clouds']['all'])
        country.append(response['sys']['country'])
        date.append(response['dt'])
        humidity.append(response['main']['humidity'])
        temp.append(response['main']['temp_max'])
        lat.append(response['coord']['lat'])
        lng.append(response['coord']['lon'])
        wind.append(response['wind']['speed'])
        new_cities.append(city)
    except:
        print("City not found!")
        pass

print('-------------------------')
print('Data Retrieval Complete')
print('-------------------------')
```

```
Processing record 16 of set 7 | tarudant
http://api.openweathermap.org/data/2.5/weather?units=Imperial&APPID=60d7b
29002415654b126178e48b80cc5 (http://api.openweathermap.org/data/2.5/weath
er?units=Imperial&APPID=60d7b29002415654b126178e48b80cc5)
City not found!
Processing record 17 of set 7 | tapaua
http://api.openweathermap.org/data/2.5/weather?units=Imperial&APPID=60d7b
29002415654b126178e48b80cc5 (http://api.openweathermap.org/data/2.5/weath
er?units=Imperial&APPID=60d7b29002415654b126178e48b80cc5)
City not found!

Processing record 18 of set 7 | dodoma
http://api.openweathermap.org/data/2.5/weather?units=Imperial&APPID=60d7b
29002415654b126178e48b80cc5 (http://api.openweathermap.org/data/2.5/weath
er?units=Imperial&APPID=60d7b29002415654b126178e48b80cc5)
Processing record 19 of set 7 | clyde river
http://api.openweathermap.org/data/2.5/weather?units=Imperial&APPID=60d7b
29002415654b126178e48b80cc5 (http://api.openweathermap.org/data/2.5/weath
er?units=Imperial&APPID=60d7b29002415654b126178e48b80cc5)
Processing record 20 of set 7 | springbok
```

Processing Record 20 of Set 7 | Storfjorden

Convert Raw Data to DataFrame

- Export the city data into a .csv.
- Display the DataFrame

In [8]:
```
city_data_df = pd.DataFrame({'City': new_cities,
                             'Latitude': lat,
                             'Longitude': lng,
                             'Temperature': temp,
                             'Humidity': humidity,
                             'Cloudiness': cloudiness,
                             'Wind Speed': wind,
                             'Date' : date,
                             'Country' : country})
```

In [9]:
```
pd.DataFrame.to_csv(city_data_df, 'city_data.csv')

city_data_df.head()
```

Out[9]:

	City	Latitude	Longitude	Temperature	Humidity	Cloudiness	Wind Speed	Date	Country
0	mizan teferi	6.99	35.58	60.21	91	0	3.78	1571957095	ET
1	kruisfontein	-34.00	24.73	63.68	44	100	7.29	1571957096	ZA
2	vaini	15.34	74.49	68.36	98	88	5.93	1571957096	IN
3	teya	21.05	-89.07	93.20	52	40	16.11	1571957096	MX
4	rikitea	-23.12	-134.97	71.19	83	80	28.81	1571957096	PF

Plotting the Data

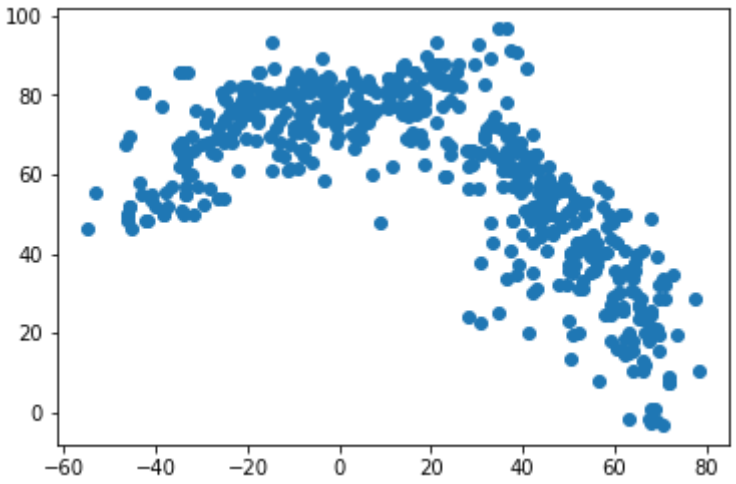
- Use proper labeling of the plots using plot titles (including date of analysis) and axes labels.
- Save the plotted figures as .pngs.

Latitude vs. Temperature Plot

In [10]:
```
plt.scatter(city_data_df['Latitude'], city_data_df['Temperature'])
plt.title(f'City Latitude vs. Temperature {date.today()}')
plt.xlabel('Latitude')
plt.ylabel('Temperature (F)')
plt.grid(True)
plt.savefig('lat_temp.png', bbox_inches='tight')
```

```
---------------------------------------------------------------------------
--
AttributeError                            Traceback (most recent call las
t)
<ipython-input-10-b964c37cb178> in <module>
      1 plt.scatter(city_data_df['Latitude'], city_data_df['Temperature']
)
----> 2 plt.title(f'City Latitude vs. Temperature {date.today()}')
      3 plt.xlabel('Latitude')
      4 plt.ylabel('Temperature (F)')
      5 plt.grid(True)

AttributeError: 'list' object has no attribute 'today'
```

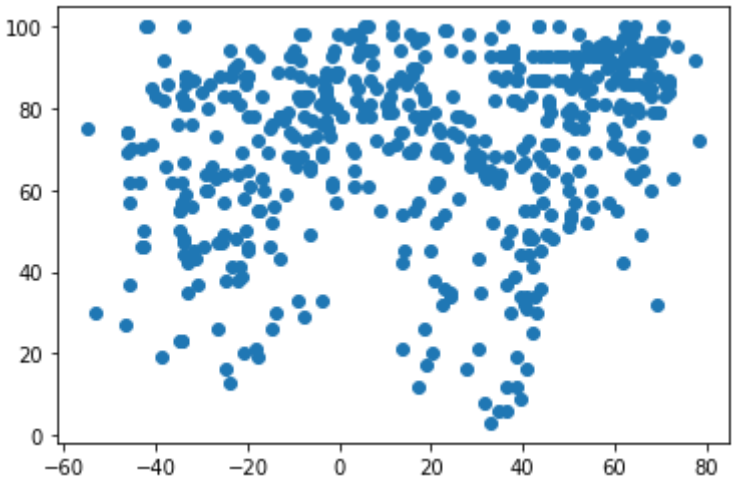


Latitude vs. Humidity Plot

In [11]:
```
plt.scatter(city_data_df['Latitude'], city_data_df['Humidity'])
plt.title(f'City Latitude vs. Humidity {date.today()}')
plt.xlabel('Latitude')
plt.ylabel('Humidity (%)')
plt.grid(True)
plt.savefig('lat_humid.png', bbox_inches='tight')
```

```
---------------------------------------------------------------------------
--
AttributeError                            Traceback (most recent call las
t)
<ipython-input-11-5d308bda9191> in <module>
      1 plt.scatter(city_data_df['Latitude'], city_data_df['Humidity'])
----> 2 plt.title(f'City Latitude vs. Humidity {date.today()}')
      3 plt.xlabel('Latitude')
      4 plt.ylabel('Humidity (%)')
      5 plt.grid(True)

AttributeError: 'list' object has no attribute 'today'
```

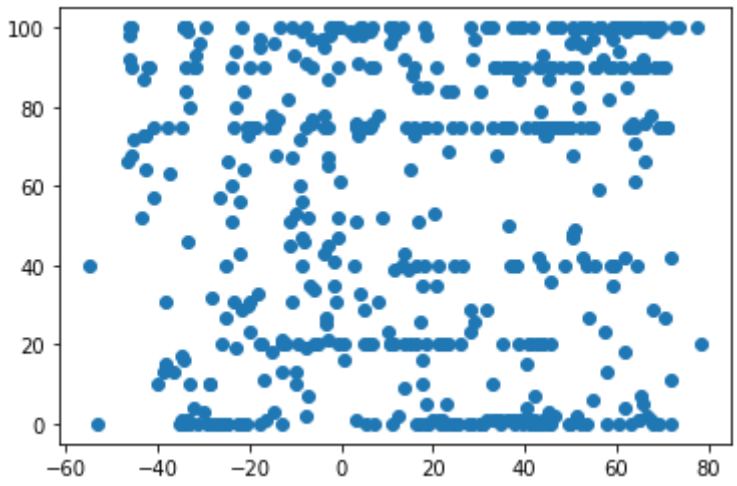


Latitude vs. Cloudiness Plot

In [12]:
```
plt.scatter(city_data_df['Latitude'], city_data_df['Cloudiness'])
plt.title(f'City Latitude vs. Cloudiness {date.today()}')
plt.xlabel('Latitude')
plt.ylabel('Cloudiness (%)')
plt.grid(True)
plt.savefig('lat_cloud.png', bbox_inches='tight')
```

```
---------------------------------------------------------------------------
--
AttributeError                            Traceback (most recent call las
t)
<ipython-input-12-d9f3c6f89d28> in <module>
      1 plt.scatter(city_data_df['Latitude'], city_data_df['Cloudiness'])
----> 2 plt.title(f'City Latitude vs. Cloudiness {date.today()}')
      3 plt.xlabel('Latitude')
      4 plt.ylabel('Cloudiness (%)')
      5 plt.grid(True)

AttributeError: 'list' object has no attribute 'today'
```

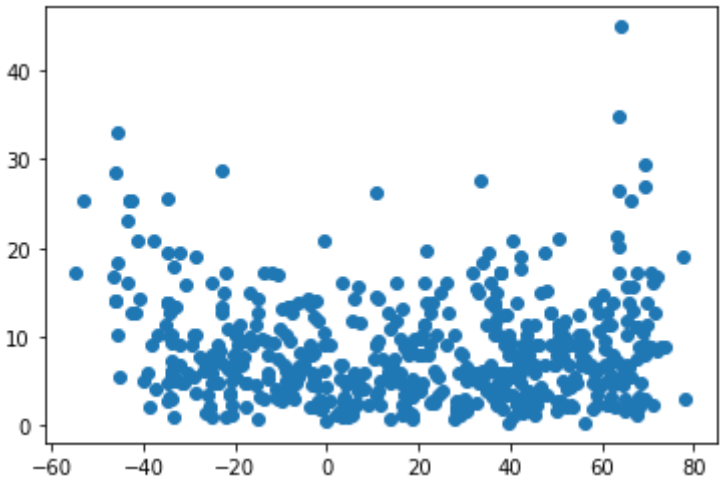


Latitude vs. Wind Speed Plot

In [13]:
```
plt.scatter(city_data_df['Latitude'], city_data_df['Wind Speed'])
plt.title(f'City Latitude vs. Wind Speed {date.today()}')
plt.xlabel('Latitude')
plt.ylabel('Wind Speed (mph)')
plt.grid(True)
plt.savefig('lat_wind.png', bbox_inches='tight')
```

```
---------------------------------------------------------------------------
--
AttributeError                            Traceback (most recent call las
t)
<ipython-input-13-b6dfeb2be9a1> in <module>
      1 plt.scatter(city_data_df['Latitude'], city_data_df['Wind Speed'])
----> 2 plt.title(f'City Latitude vs. Wind Speed {date.today()}')
      3 plt.xlabel('Latitude')
      4 plt.ylabel('Wind Speed (mph)')
      5 plt.grid(True)

AttributeError: 'list' object has no attribute 'today'
```



In [ ]: