```
# IPython log file

# citi = cities
units = "imperial"
url = "http://api.openweathermap.org/data/2.5/weather?"
owm.utils.load_config
settings = {"units": units, "APPID": api_key}
query_url = f"{url}appid={api_key}&units={units}&q="

# set up lists to hold reponse info
weather_response = []
cityList = []
citNam = []
lat = []
temp = []
humi = []
eSea = []
eGnd = []
clouds = []
wind = []
code = []

# for citi in cities:
#     weather_response = owm.find_city(citi, **settings)
#     selection = ('id','coord.lat','coord.lon')
#     data = weather_response(selection)
#     cityList.append(data)

# Loop through the list of cities and perform a request for data on each
# for city in cities:
#     response = requests.get(query_url + city).json()
#     lat.append(response['coord']['lat'])
#     temp.append(response['main']['temp'])
#     cloud.append(response['clouds']['all'])
#     wind.append(response['wind']['speed'])
# logger.disabled = True
# logger.disabled = False

# cnt=1
with open('output_data/data.json', 'w', encoding='utf-8') as f:

    get_ipython().run_line_magic('logon', '-o -q')
    for city in cities:

        response = requests.get(query_url + city).json()
        json.dump(response, f, ensure_ascii=False, indent=4)
        response1=response
        try:
            print(f"For the city named {(response1['name'])}, ID Number-{(response1['id'])}: LAT={(response1['coord']['lat'])} WIND=
```

```
            response1=[]
            citNam.append(response['name']) #and citNam.append(cnt)
            code.append(response['id'])
            lat.append(response['coord']['lat'])
            temp.append(response['main']['temp_max'])
            humi.append(response['main']['humidity'])
            clouds.append(response['clouds']['all'])
            wind.append(response['wind']['speed'])
            eSea.append(response['main']['sea_level'])
            eGnd.append(response['main']['grnd_level'])

        except KeyError:
    #             print(f"The NAME is NO GOOD")
                next
#     print(f"The {code} information received is: {wind}")
#     print(f"The latitude information received is: {lat}")
#     print(f"The temperature information received is: {temp}")
#     print(f"The cloud information received is: {cloud}")
#     print(f"The wind information received is: {wind}")

#     print(f"The {city} latitude information received is: {lat}")
#     print(f"The  {city} temperature information received is: {temp}")
#     print(f"The  {city} cloud information received is: {cloud}")
#     print(f"The  {city} wind information received is: {wind}")

# # loop throught the list of units and append them to cityList list
# for citi in cities:
#     query_url = f"{url}appid={api_key}&q={citi}"
#     weather_response = requests.get(query_url)
#     response= weather_response.json()
#     lat.append(response['coord']['lat'])
#     temp.append(response['main']['temp'])
#     cloud.append(response['clouds']['all'])
#     wind.append(response['wind']['speed'])
#     cityList.append(weather_json)

#     for city in cities:
#     response = requests.get(query_url + city).json()
#     selection = ('name','id','coord.lat','coord.lon',"main.temp", 'main.temp_max',"main.humidity", "wind.speed",'clouds.all')
print(var_dic_list())
print(var_dic_list())
ls=[citNam, code,  lat, temp, humi, clouds, wind ]
for l in ls:
    print(len(l))
print(var_dic_list())
print(var_dic_list())
dictionary = {}
dictionary={
```

```
fig1=plt.scatter(latitude, temperature, marker='.', facecolors="blue", edgecolors="blue")
# Set the upper and lower limits of our y axis
plt.ylim(0,120)
# Set the upper and lower limits of our x axis
plt.xlim(-80,100)
# Create a title, x label, and y label for our chart
plt.title("Max Temperature v Latitude")

plt.xlabel("Latitude")
plt.ylabel("Temperature (Degrees F)")

# Save an image of the chart and print to screen
# NOTE: If your plot shrinks after saving an image,
# update matplotlib to 2.2 or higher,
# or simply run the above cells again.
plt.savefig("output_data/temp_v_Latitude.png")
plt.show()
print(var_dic_list())
print(var_dic_list())
fig = plt.figure()
fig1=plt.scatter(latitude, humidity, marker=".", facecolors="blue", edgecolors="blue")
# Set the upper and lower limits of our y axis
plt.ylim(0,105)
# Set the upper and lower limits of our x axis
plt.xlim(-80,100)
# Create a title, x label, and y label for our chart
plt.title("Humidity v Latitude")
plt.xlabel("Latitude")
plt.ylabel("Humidity (%)")

# Save an image of the chart and print to screen
# NOTE: If your plot shrinks after saving an image,
# update matplotlib to 2.2 or higher,
# or simply run the above cells again.
plt.savefig("output_data/humidity_v_Latitude.png")
plt.show()
print(var_dic_list())
print(var_dic_list())
fig = plt.figure()
fig1=plt.scatter(latitude, cloudiness, marker=".", facecolors="blue", edgecolors="blue")
# Set the upper and lower limits of our y axis
plt.ylim(-5,105)
# Set the upper and lower limits of our x axis
plt.xlim(-80,100)
# Create a title, x label, and y label for our chart
plt.title("Cloudiness v Latitude")
plt.xlabel("Latitude")
plt.ylabel("Cloudiness (%)")

# Save an image of the chart and print to screen
# NOTE: If your plot shrinks after saving an image,
# update matplotlib to 2.2 or higher,
# or simply run the above cells again.
plt.savefig("output_data/cloudiness_v_Latitude.png")
plt.show()
print(var_dic_list())
print(var_dic_list())
fig = plt.figure()
fig1=plt.scatter(latitude, wind_speed, marker=".", facecolors="blue", edgecolors="blue")
# Set the upper and lower limits of our y axis
plt.ylim(0,40)
# Set the upper and lower limits of our x axis
plt.xlim(-80,100)
# Create a title, x label, and y label for our chart
plt.title("Wind Speed v Latitude")
```

se1['clouds']['all'])}")