Zeru-Zhou-project5

February 13, 2022

1 Project 5 – Zeru Zhou

TA Help: NA

Collaboration: NA

- Got help from Dr. Ward's videos

1.1 Question 1

[2]:
```
import pandas as pd
```

[3]:
```
stations = pd.read_csv("/depot/datamine/data/whin/190/stations.csv")
obs = pd.read_csv("/depot/datamine/data/whin/190/observations.csv")
```

[4]:
```
stations.head()
```

[4]:
```
    id            name   latitude  longitude
0    1  WHIN001-PULA001  40.938940 -86.474180
1  142  WHIN052-MONT004  40.104830 -86.866190
2  143  WHIN053-PULA005  40.982240 -86.385420
3  151  WHIN059-CASS006  40.844360 -86.181730
4   20  WHIN020-FOUN001  40.270957 -87.148604
```

[5]:
```
obs.head()
```

[5]:
```
                                id  station_id  temperature_average  \
0  obs_1NnyYGMtAHBFDYWOBlsDlqppzVI           1                 70.0
1  obs_1NoONHuqV4VjOK8p8FguPT02T5B           1                 69.0
2  obs_1NqnftCklLZHBCHyykvcuc8QvE9           1                 76.0
3  obs_1NqpV058q10hGNBNvYOBzzwpqOx           1                 76.0
4  obs_1NqrK3mraUzaj2j7hg6VcB23RjJ           1                 76.0

   temperature_high  temperature_low  humidity_average  barometric_pressure  \
0              71.0             70.0              83.0                30.05
1              70.0             69.0              84.0                30.04
2              77.0             76.0              76.0                29.89
3              76.0             76.0              77.0                29.88
4              76.0             76.0              77.0                29.88
```

```
   wind_speed_average  wind_speed_high  wind_direction_high  …  rain  \
0                 0.0              3.0                247.5  …   0.0
1                 1.0              3.0                247.5  …   0.0
2                 2.0              4.0                202.5  …   0.0
3                 2.0              4.0                202.5  …   0.0
4                 2.0              4.0                202.5  …   0.0

   rain_last_hour  temperature_soil_2  temperature_soil_5  \
0             0.0                77.0                78.0
1             0.0                76.0                78.0
2             0.0                80.0                80.0
3             0.0                80.0                80.0
4             0.0                80.0                80.0

   temperature_soil_10  temperature_soil_15  moisture_soil_2  moisture_soil_5  \
0                 76.0                 74.0             24.0             24.0
1                 76.0                 74.0             24.0             25.0
2                 78.0                 75.0             31.0             30.0
3                 78.0                 75.0             31.0             31.0
4                 78.0                 75.0             32.0             31.0

   moisture_soil_10  moisture_soil_15
0              10.0               9.0
1              10.0               9.0
2              12.0              10.0
3              12.0              10.0
4              12.0              10.0

[5 rows x 23 columns]
```

[4]:
```
dat = obs.merge(stations, how = "left", left_on = "station_id", right_on = "id")
```

[5]:
```
dat = dat.drop(columns = "id_y")
dat = dat.rename(columns = {"id_x": "id", "name": "station_name"})
```

[11]:
```
dat.head()
```

[11]:
```
                             id  station_id  temperature_average  \
0  obs_1NnyYGMtAHBFDYWOBlsDlqppzVI           1                 70.0
1  obs_1No0NHuqV4VjOK8p8FguPT02T5B           1                 69.0
2  obs_1NqnftCklLZHBCHyykvcuc8QvE9           1                 76.0
3  obs_1NqpV058q10hGNBNvYOBzzwpqOx           1                 76.0
4  obs_1NqrK3mraUzaj2j7hg6VcB23RjJ           1                 76.0

   temperature_high  temperature_low  humidity_average  barometric_pressure  \
0              71.0             70.0              83.0                30.05
```

```
1              70.0            69.0             84.0                30.04
2              77.0            76.0             76.0                29.89
3              76.0            76.0             77.0                29.88
4              76.0            76.0             77.0                29.88

   wind_speed_average  wind_speed_high  wind_direction_high  …  \
0                 0.0              3.0                247.5  …
1                 1.0              3.0                247.5  …
2                 2.0              4.0                202.5  …
3                 2.0              4.0                202.5  …
4                 2.0              4.0                202.5  …

   temperature_soil_5  temperature_soil_10  temperature_soil_15  \
0                78.0                 76.0                 74.0
1                78.0                 76.0                 74.0
2                80.0                 78.0                 75.0
3                80.0                 78.0                 75.0
4                80.0                 78.0                 75.0

   moisture_soil_2  moisture_soil_5  moisture_soil_10  moisture_soil_15  \
0             24.0             24.0              10.0               9.0
1             24.0             25.0              10.0               9.0
2             31.0             30.0              12.0              10.0
3             31.0             31.0              12.0              10.0
4             32.0             31.0              12.0              10.0

      station_name  latitude  longitude
0  WHIN001-PULA001  40.93894  -86.47418
1  WHIN001-PULA001  40.93894  -86.47418
2  WHIN001-PULA001  40.93894  -86.47418
3  WHIN001-PULA001  40.93894  -86.47418
4  WHIN001-PULA001  40.93894  -86.47418

[5 rows x 26 columns]
```

All the operations are done above.

1.2 Question 2

[6]:
```
from cyksuid import ksuid
```

[14]:
```
dat["id"].sample(10).tolist()
```

[14]:
```
['obs_1izAMnnF8k7VBNsBlXllp7rgLf9',
 'obs_1k0ezkKh0QDD1bUjpKOpCrFNeKz',
 'obs_1pne5pItAB1Bq29ABdjMPWATmY1',
 'obs_1nVMFQrskDOqKT7QOwVsGO5pqHj',
```

```
 'obs_1TCORE68ruVagTZot8v4NXKANJK',
 'obs_1hpEBs0d8obYurOG4ZJSOMITuQd',
 'obs_1kOYFOuwR7G5pEGlnnYTHa6m3C5',
 'obs_1dZxEOiWUczjqLnE6QHksgPHnKT',
 'obs_1jKaOO2vPlGUrIOrsfxtPTDqIQh',
 'obs_1sggMnmGLtOwU5JEUqkGmRRLavP']
```

[7]:
```
def get_datetime(Str):
    return ksuid.parse(Str.replace("obs_" , "")).datetime
```

[18]:
```
Value = sorted(dat["id"].sample(10).tolist())
```

[19]:
```
for val in Value:
    print(get_datetime(val))
```

```
2020-04-05 01:30:00
2020-05-08 13:45:00
2020-09-10 01:30:00
2020-12-03 16:00:00
2020-12-28 02:45:00
2021-04-22 15:00:00
2021-05-07 14:45:00
2021-05-23 13:15:00
2021-05-25 03:15:00
2021-05-29 18:45:00
```

As we can see, the time is sorted and from the earliest to the latest.

1.3 Question 3

[1]:
```
import numpy as np
```

[8]:
```
def degrees_to_radians(value):
    return float(value * np.arctan2(0,-1)/180)
```

[26]:
```
degrees_to_radians(88.0)
```

[26]:
```
1.53588974175501
```

Result is listed above.

1.4 Question 4

[9]:
```
def degrees_to_radians(value):
    return float(value * np.arctan2(0,-1)/180)
```

[12]:
```
def get_distance(Ser1, Ser2):
    lat1 = degrees_to_radians(dat.loc[dat["id"] == Ser1, "latitude"])
    lat2 = degrees_to_radians(dat.loc[dat["id"] == Ser2, "latitude"])
    lon1 = degrees_to_radians(dat.loc[dat["id"] == Ser1, "longitude"])
    lon2 = degrees_to_radians(dat.loc[dat["id"] == Ser2, "longitude"])
    return 2*6367.4447*np.arcsin(np.sqrt(np.sin((lat2-lat1)/2)**2+np.
↪cos(lat1)*np.cos(lat2)*np.sin((lon2-lon1)/2)**2))
```

[13]:
```
get_distance("obs_1amnn4xst3O9VOawmUHFiqBVnCK",␣
↪"obs_1fwlznMZXXS8WBkmyTHRgWnHYYf")
```

[13]:
```
37.896692299010574
```

[14]:
```
location1 = dat.loc[dat['id']=="obs_1amnn4xst3O9VOawmUHFiqBVnCK", :]
location2 = dat.loc[dat['id']=="obs_1fwlznMZXXS8WBkmyTHRgWnHYYf", :]
```

[15]:
```
def get_distance(Ser1, Ser2):
    lat1 = degrees_to_radians(Ser1["latitude"])
    lat2 = degrees_to_radians(Ser2["latitude"])
    lon1 = degrees_to_radians(Ser1["longitude"])
    lon2 = degrees_to_radians(Ser2["longitude"])
    return 2*6367.4447*np.arcsin(np.sqrt(np.sin((lat2-lat1)/2)**2+np.
↪cos(lat1)*np.cos(lat2)*np.sin((lon2-lon1)/2)**2))
```

[16]:
```
get_distance(location1, location2)
```

[16]:
```
37.896692299010574
```

Here are 2 ways to do the problem. Same idea but different processes.

1.5 Question 5

[1]:
```
import plotly.express as px
```

[8]:
```
def plot_stations(dat):
    dat['position'] = dat['latitude'].astype(str) + dat['longitude'].astype(str)
    figure = px.scatter_geo(dat.groupby(['position']).head(1), lat =␣
↪'latitude', lon = 'longitude', hover_name = "station_id", scope = 'usa')
    figure.update_layout(title = 'World Map', title_x = 0.5)
    return figure.show()
```

[9]:
```
plot_stations(dat)
```

Function is created and plot was zoomed in.

1.6 Pledge

By submitting this work I hereby pledge that this is my own, personal work. I've acknowledged in the designated place at the top of this file all sources that I used to complete said work, including but not limited to: online resources, books, and electronic communications. I've noted all collaboration with fellow students and/or TA's. I did not copy or plagiarize another's work.

> As a Boilermaker pursuing academic excellence, I pledge to be honest and true in all that I do. Accountable together – We are Purdue.