Question

December 10, 2021

0.0.1 Instructions

```
- The data required for this task has been provided in the file 'data.csv'
- Read the questions provided for each cell and assign your answers to respective variables pro
- If answers are floating point numbers round of updo two floating point after the decimal
  - for example 10.546 should be read as 10.55, 10.544 as 10.54 and 10.1 as 10.10
- pandas and numpy packages are preinstalled for this task which should be sufficient to comple
- If you need any other additional package run !pip3 install <package_name> --user in a new cel
- Please dont change variable name meant to assign your answers.
```

NOTE: Run the last cell to save your answers in pickle file.

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

[2]:
```
### Read the data (this will not be graded)
!wget https://hr-projects-assets-prod.s3.amazonaws.com/c3pde3c3lgm/
 ↪963fbab228e2896e79fc09e385ab377d/data.csv
```

```
--2021-12-10 10:41:57--  https://hr-projects-assets-
prod.s3.amazonaws.com/c3pde3c3lgm/963fbab228e2896e79fc09e385ab377d/data.csv
Resolving hr-projects-assets-prod.s3.amazonaws.com (hr-projects-assets-
prod.s3.amazonaws.com)… 52.217.13.52
Connecting to hr-projects-assets-prod.s3.amazonaws.com (hr-projects-assets-
prod.s3.amazonaws.com)|52.217.13.52|:443… connected.
HTTP request sent, awaiting response… 200 OK
Length: 332846 (325K) [binary/octet-stream]
Saving to: ‘data.csv.1’

data.csv.1          100%[===================>] 325.04K   311KB/s    in 1.0s

2021-12-10 10:41:59 (311 KB/s) - ‘data.csv.1’ saved [332846/332846]
```

[3]:
```
df = pd.read_csv('data.csv')
df.columns
```

```
[3]: Index(['Day', 'Average temperature (°F)', 'Average humidity (%)',
            'Average dewpoint (°F)', 'Average barometer (in)',
            'Average windspeed (mph)', 'Average gustspeed (mph)',
            'Average direction (°deg)', 'Rainfall for month (in)',
            'Rainfall for year (in)', 'Maximum rain per minute ',
            'Maximum temperature (°F)', 'Minimum temperature (°F)',
            'Maximum humidity (%)', 'Minimum humidity (%)', 'Maximum pressure ',
            'Minimum pressure ', 'Maximum windspeed (mph)',
            'Maximum gust speed (mph)', 'Maximum heat index (°F)'],
           dtype='object')
```

0.0.2 What is the standard deviation of maximum windspeed across all the days

```
[4]: q1 = df['Maximum windspeed (mph)'].std().round(2)
```

0.0.3 What is the difference between 50th percentile and 75th percentile of average temperature

```
[5]: q2 = (-np.percentile(df['Average temperature (°F)'],50)+np.
      ↪percentile(df['Average temperature (°F)'],75)).round(2)
```

```
[6]: from scipy.stats import pearsonr
```

0.0.4 What is the pearson correlation between average dew point and average temperature

```
[7]: q3 = pearsonr(df['Average dewpoint (°F)'], df['Average temperature (°F)'])[0].
      ↪round(2)
```

0.0.5 Out of all the available records which month has the lowest average humidity.

```
- Assign your answer as month index, for example if its July index is 7
```

```
[8]: from datetime import datetime
     a=df.index[df['Average humidity (%)'] == df['Average humidity (%)'].min()].
      ↪tolist()
     stri=df.loc[a[0]].Day
     q4=datetime.strptime(stri, '%d/%m/%Y').date().month
```

0.0.6 Which month has the highest median for maximum_gust_speed out of all the available records. Also find the repective value

```
- hint: group by month
```

[9]:
```
df['new_data']=pd.to_datetime(df['Day'],format='%d/%m/%Y').dt.month
simpli = df[['new_data','Maximum gust speed (mph)']]
aa=df.groupby(by="new_data").median()['Maximum gust speed (mph)']

q6 = aa[aa==aa.max()].index.values[0]
q5 = aa[aa==aa.max()].values[0]
```

0.0.7 Determine the average temperature between the months of March 2010 to May 2012 (including both the months)

[10]:
```
df['Date']=pd.to_datetime(df['Day'],format='%d/%m/%Y')
#df[['new_data','Average temperature (°F)']]
q7 = df[(df['Date'] >= '2010-03-01') & (df['Date'] <= '2012-05-31')]['Average␣
↪temperature (°F)'].mean().round(2)
```

0.0.8 Find the range of averange temperature on Dec 2010

[11]:
```
q88 = df[(df['Date'] >= '2010-12-01') & (df['Date'] <= '2010-12-31')]['Average␣
↪temperature (°F)']
q8=round(q88.max()-q88.min(),2)
```

0.0.9 Out of all available records which day has the highest difference between maximum_pressure and minimum_pressure

```
- assign the date in string format as 'yyyy-mm-dd'. Make sure you enclose it with single quote
```

[12]:
```
df["pressure_diff"] = df['Maximum pressure '] - df['Minimum pressure ']
q9 = df.Date[df["pressure_diff"]== df["pressure_diff"].max()].values[0]
q9 = np.datetime_as_string(q9, unit='D')
```

0.0.10 How many days falls under median (i.e equal to median value) of barrometer reading.

[14]:
```
q10 = len(df[df['Average barometer (in)']==df['Average barometer (in)'].
↪median()])
```

0.0.11 Out of all the available records how many days are within one standard deviation of average temperaturem

[15]:
```
mean_avg=df['Average temperature (°F)'].mean()
std_avg=df['Average temperature (°F)'].std()

q11= len(df[(df["Average temperature (°F)"] >= mean_avg-std_avg) & (df["Average␣
↪temperature (°F)"]<= mean_avg + std_avg)])
```

0.1 Run this cell, to save your answers. Donot Modify this cell.

[16]:
```
import pickle
import hashlib






def make_pickle2(file_name, obj):
    with open(file_name, 'wb') as f:
        pickle.dump(geth(obj), f, pickle.HIGHEST_PROTOCOL)

def geth(obj):
    obj = str(obj).encode()
    m = hashlib.md5()
    m.update( bytes(obj) )
    return m.hexdigest()

def pickling():

    try:
        make_pickle2('q1.pickle', q1)
    except:
        print('q1 not defined')
    try:
        make_pickle2('q2.pickle', q2)
    except:
        print('q2 not defined')
    try:
        make_pickle2('q3.pickle', q3)
    except:
        print('q3 not defined')
    try:
        make_pickle2('q4.pickle', q4)
    except:
        print('q4 not defined')
    try:
        make_pickle2('q5.pickle', q5)
    except:
        print('q5 not defined')

    try:
        make_pickle2('q6.pickle', q6)
    except:
        print('q6 not defined')

    try:
        make_pickle2('q7.pickle', q7)
    except:
        print('q7 not defined')
    try:
        make_pickle2('q8.pickle', q8)
    except:
        print('q8 not defined')
    try:
        make_pickle2('q9.pickle', q9)
    except:
        print('q9 not defined')
    try:
        make_pickle2('q10.pickle', q10)
    except:
        print('q10 not defined')
    try:
        make_pickle2('q11.pickle', q11)
    except:
        print('q11 not defined')



pickling()
```

[ ]: