Geog0111_Coursework_PartA

November 14, 2022

1 Geog0111 Coursework Part A

[1]:

```python
## create a directory for to save files in following code

from pathlib import Path

CW1 = Path('CW1')
CW1.mkdir(parents=True,exist_ok=True)
```

[2]:

```python
## define a function that can derive stream discharge from USGS

def stream_discharge(year:int):
    """
    Access datasets from url of USGS and write it as a text file
    Read this text file into pandas dataframe
    Convert units of cubic feet per second to million litres per day
    Calculate and output the stream discharge in ML/day

    """
    from urlpath import URL
    from pathlib import Path
    import pandas as pd

    # create a directory for stream discharge
    streamdir = Path('CW1/stream')
    streamdir.mkdir(parents=True,exist_ok=True)

    # for the URL
    site = 'https://waterservices.usgs.gov'
    site_dir = 'nwis/dv'
    site_file = f'?
 ↪sites=08220000&format=rdb&startDT={year}-01-01&endDT={year}-12-31&parameterCd=00060'

    url = URL(site,site_dir,site_file)

    # get stream data from url
    year_stream = url.get()

    # write stream date to a text file
    if year_stream.status_code == 200:

        # setup Path object for output file
        s = Path(streamdir,f'stream_{year}')

        # write text data
        s.write_text(year_stream.text)

    # read the file into pandas dataframe
    df = pd.read_csv(s,delimiter="\t",skiprows=30,header=None)

    # convert units of cubic feet per second to million litres per day
    unit_convert_coefficient = 28.316846592 * 1e-6 * 60 * 60 * 24

    # calculate the stream discharge in ML/day and output
    return df[3] * unit_convert_coefficient
```

[3]:

```python
help (stream_discharge)
```

```
Help on function stream_discharge in module __main__:

stream_discharge(year: int)
    Access datasets from url of USGS and write it as a text file
    Read this text file into pandas dataframe
    Convert units of cubic feet per second to million litres per day
    Calculate and output the stream discharge in ML/day
```

[4]:

```python
## define a function that can derive mean temperature from CSC

def mean_temperature(year:int):
    """
    Access datasets from url of CSC and write it as a text file
    Read this text file into pandas dataframe
    Create a datetime list in dataframe for filtering
    Convert the units of Fahrenheit to Celcius
    Calculate and output the mean temperature in Celcius

    """
    from urlpath import URL
    from pathlib import Path
    import pandas as pd
    import numpy as np

    # create a directory for temparature
    tempdir = Path('CW1/temp')
    tempdir.mkdir(parents=True,exist_ok=True)

    # for the URL
    site = 'https://raw.githubusercontent.com'
    site_dir = 'UCL-EO/geog0111/master/notebooks/data'
    site_file = 'delNorteT.dat'

    url = URL(site,site_dir,site_file)

    # get temperature data from url
    tempdata = url.get()

    # write stream date to a text file
    if tempdata.status_code == 200:

        # setup Path object for output file
        t = Path(tempdir,'tempdata')
        # write text data
        t.write_text(tempdata.text)

    # read the file into pandas
    df = pd.read_csv(t,delimiter="\t")

    # use datetime to filter date description
    df['DATE'] =  pd.to_datetime(df["DEL NORTE 2E"])

    # replace M and T to avoid their impacts on calculation
    df = df.replace('M',np.nan,regex=True).replace('T',np.nan,regex=True)

    # convert data type of maxt and mint to float, and calculate their mean
    df['meant'] = (df['maxt'].astype(float) + df['mint'].astype(float))/2

    # convert the units of Fahrenheit to Celcius
    df['meant'] = (df['meant']-32)*5/9

    # select the data set for specific year and output
    return df['meant'][df['DATE'].dt.year == year]
```

[5]:

```
help (mean_temperature)
```

```
Help on function mean_temperature in module __main__:

mean_temperature(year: int)
    Access datasets from url of CSC and write it as a text file
    Read this text file into pandas dataframe
    Create a datetime list in dataframe for filtering
    Convert the units of Fahrenheit to Celcius
    Calculate and output the mean temperature in Celcius
```

[6]:

```python
## define a function that can create dataframe for stream,temperature and day

def day_meant_stream_2csv(year:int):
    '''
    Access datasets from url of USGS and write it as a text file
    Read this text file into pandas dataframe
    Convert units of cubic feet per second to million litres per day
    Calculate and output the stream discharge in ML/day

    Access datasets from url of CSC and write it as a text file
    Read this text file into pandas dataframe
    Create a datetime list in dataframe for filtering
    Convert the units of Fahrenheit to Celcius
    Calculate and output the mean temperature in Celcius

    create dataframe for stream discharge, mean temperature and day of year
    save this dataframe as a dataset in a CSV-format file

    '''
    import pandas as pd
    from pathlib import Path

    # set each data list for dataframe
    y = list(stream_discharge(year))
    z = list(mean_temperature(year))
    x = list(range(1,len(y)+1))

    # load into pandas
    df = pd.DataFrame({'day_of_year':x,'stream_discharge':y,'mean_temperature':
 ↪z})

    # save this dataframe as a dataset to CSV file without the index
    df.to_csv(Path(f'work/delNorte{year}.csv'),index=False)

    # return a pandas dataframe containing the dataset
    return pd.read_csv(Path(f'work/delNorte{year}.csv'))
```

[7]:

```python
help (day_meant_stream_2csv)
```

```
Help on function day_meant_stream_2csv in module __main__:

day_meant_stream_2csv(year: int)
    Access datasets from url of USGS and write it as a text file
    Read this text file into pandas dataframe
```

```
Convert units of cubic feet per second to million litres per day
Calculate and output the stream discharge in ML/day

Access datasets from url of CSC and write it as a text file
Read this text file into pandas dataframe
Create a datetime list in dataframe for filtering
Convert the units of Fahrenheit to Celcius
Calculate and output the mean temperature in Celcius

create dataframe for stream discharge, mean temperature and day of year
save this dataframe as a dataset in a CSV-format file
```

[8]:

```python
## demonstrate running the function to access and save the data

from datetime import datetime
from pathlib import Path
import pandas as pd

# create a dictionary to save dataframe of each year, preparing for plotting
dfdic = {}

# use a for loop to run the code year by year
for year in range(2016,2019+1):

    # run the defined function
    df = day_meant_stream_2csv(year)

    # save dataframe to the dictionary
    dfdic[year] = df

    # locate the files
    csvfiles = Path('work',f'delNorte{year}.csv')

    # show the year and CSV file name
    print(f'Year: {year}','\t','\t',f'CSV File name: delNorte{year}','\n')

    # demonstrate the pandas dataframe
    df1=pd.read_csv(csvfiles)
    print(df1)

    # show the file size of csvfiles
    print(f'File size              : {csvfiles.stat().st_size} bytes')

    # show the modification date of csvfiles
    modification_date = csvfiles.stat().st_mtime
    print(f'Modification date      : {datetime.
↪fromtimestamp(modification_date)}')

    # create a split line between each year
    print('\n',"{:-^50s}".format("Split line"),'\n')
```

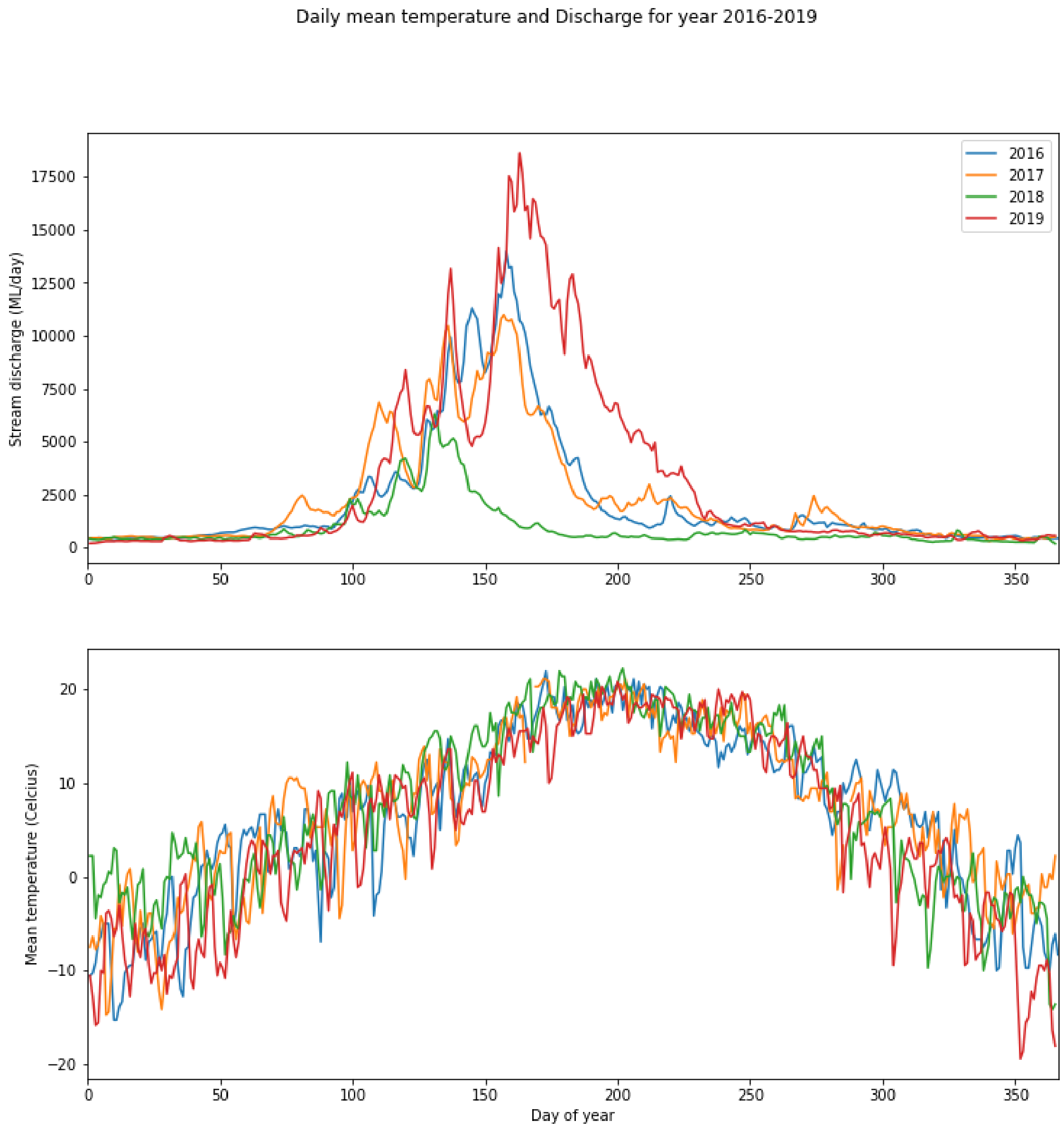
```
Year: 2016               CSV File name: delNorte2016

     day_of_year  stream_discharge  mean_temperature
0              1        403.684965        -10.555556
1              2        415.917843        -10.277778
2              3        440.383598         -9.166667
3              4        464.849354         -6.944444
4              5        452.616476         -6.388889
..           …               …               …
361          362        428.150720         -8.888889
362          363        428.150720        -10.000000
363          364        440.383598         -7.222222
364          365        440.383598         -6.111111
365          366        440.383598         -8.333333

[366 rows x 3 columns]
File size              : 14124 bytes
Modification date      : 2022-11-14 01:35:08.285469

 --------------------Split line--------------------

Year: 2017               CSV File name: delNorte2017

     day_of_year  stream_discharge  mean_temperature
0              1        452.616476         -7.500000
1              2        452.616476         -6.388889
2              3        452.616476         -7.777778
3              4        452.616476         -7.222222
4              5        464.849354         -4.166667
..           …               …               …
360          361        440.383598         -1.111111
361          362        452.616476         -1.111111
362          363        464.849354          0.277778
363          364        489.315109         -0.277778
364          365        501.547987          2.222222

[365 rows x 3 columns]
File size              : 14052 bytes
Modification date      : 2022-11-14 01:35:08.828464

 --------------------Split line--------------------
```

```
Year: 2018               CSV File name: delNorte2018

     day_of_year  stream_discharge  mean_temperature
0              1        428.150720          2.222222
1              2        391.452087          2.222222
2              3        379.219210         -4.444444
3              4        366.986332         -1.944444
4              5        379.219210         -2.222222
..           …               …                 …
360          361        513.780865         -3.055556
361          362        489.315109         -4.444444
362          363        391.452087        -13.611111
363          364        281.356188        -14.166667
364          365        195.726044        -13.611111

[365 rows x 3 columns]
File size                : 14215 bytes
Modification date        : 2022-11-14 01:35:09.660473

 --------------------Split line--------------------

Year: 2019               CSV File name: delNorte2019

     day_of_year  stream_discharge  mean_temperature
0              1        210.405497        -10.555556
1              2        215.298648        -12.777778
2              3        225.084950        -15.833333
3              4        244.657555        -15.555556
4              5        256.890432        -10.000000
..           …               …                 …
360          361        538.246620        -10.000000
361          362        587.178131         -8.888889
362          363        587.178131        -10.277778
363          364        562.712375        -16.388889
364          365        562.712375        -18.055556

[365 rows x 3 columns]
File size                : 14313 bytes
Modification date        : 2022-11-14 01:35:10.483500

 --------------------Split line--------------------
```

[9]:

```
## visualise the data

import matplotlib.pyplot as plt
```

```
# set the plot size and type
x_size,y_size = 12,12
fig, axs = plt.subplots(2,1,figsize=(x_size,y_size))
axs = axs.flatten()

# set the plot title
fig.suptitle('Daily mean temperature and Discharge for year 2016-2019')

# use for loop to plot 4 dataframes in one chart with two panels
for year,df in dfdic.items():

    axs[0].plot(df["day_of_year"],df['stream_discharge'],label=year)
    axs[1].plot(df["day_of_year"],df['mean_temperature'],label=year)

# set legend for the plot
axs[0].legend(loc='best')

# set limit for x axis
axs[0].set_xlim(0,366)
axs[1].set_xlim(0,366)

# set label of y axis for both two panels
axs[0].set_ylabel('Stream discharge (ML/day)')
axs[1].set_ylabel('Mean temperature (Celcius)')

# set label of x axis (two panels share one x-label)
_=axs[1].set_xlabel('Day of year')

# save plot to PNG file
ofile = 'CW1/Daily mean temperature and Discharge for year 2016-2019.png'
plt.savefig(ofile)
```


Daily mean temperature and Discharge for year 2016-2019
Stream discharge (ML/day)
17500
15000
12500
10000
7500
5000
2500
0
0
50
100
150
200
250
300
350
2016
2017
2018
2019
Mean temperature (Celcius)
20
10
0
−10
−20
0
50
100
150
200
250
300
350
Day of year