The instructions and the csv data file for the fourth homework are attached. Please read the instructions carefully, including the Appendices.

These are the variables assigned to each group based on last names:

- Group 1 [Last names A-G]: Assigned variable is social support (family)
- Group 2 [Last names H-R]: Assigned variable is healthy life expectancy at birth (health)
- Group 3 [Last names S-Z]: Assigned variable is freedom to make life choices (freedom)

Dataset: For this assignment, you will use the happiness dataset that contains panel data with country and years related to the happiness index. The dataset is described in **Appendix 1**.

Requirements: Using the variable assigned to you (based on your last name group) your Python code should do the following (See sample output on **Appendix 2**).

1. Reduce your dataset to the last four years (2017-2020) and keep all the variables. Add code to answer how many rows and columns are in the reduced dataset, and whether your variable has any missing values.
2. Calculate the average of your assigned variable for all countries across the four-year period. Sort the dataset to show first the countries with the highest values.
3. Calculate the median of your assigned variable by region.
4. Calculate the mean of the variable by region and year and graphically show how the variable has changed.

Grading Rubric: 20% independent verification of program run; 50% required and correct output; 20% file submission compliance; 10% authorship, code, and printout documentation.

Appendix 1: Data Set Description

Six key variables are combined to form the happiness score index calculated for each country:

- Economy: Log GDP per capita
- Family: social support
- Health: healthy life expectancy at birth
- Freedom: freedom to make life choices
- Generosity: perceptions of generosity
- Trust: perceptions of corruption

The “Life Ladder” is the main life evaluation question, where 10 is the best possible life and 0 is the worst. You can read more about this at: https://worldhappiness.report/ed/2021/

Appendix 2: Sample Output (for the Life Ladder Variable)

Q1: Characteristics of reduced dataset and missing values on Life Ladder

Country name	Life Ladder
Finland	7.828750
Denmark	7.612750
Switzerland	7.546250
Iceland	7.528000
Netherlands	7.462750
...	...
Central African Republic	3.476000
Rwanda	3.312333
Zimbabwe	3.277000
South Sudan	2.817000
Afghanistan	2.577000

```
The data has 528 rows and 12 columns
Are there missing values in the Life Ladder Column? False
```

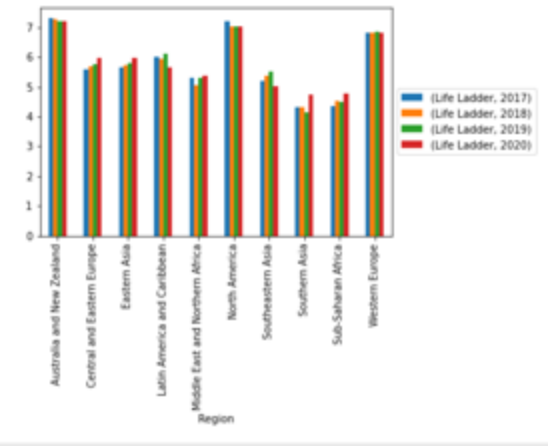
Q2: Average by country with sorted results

	Life Ladder			
year	2017	2018	2019	2020
Region				
Australia and New Zealand	7.2920	7.2735	7.2195	7.1970
Central and Eastern Europe	5.6225	5.7845	5.8210	6.0905
Eastern Asia	5.6180	5.7940	5.7810	5.9020
Latin America and Caribbean	6.1570	6.0560	6.0860	5.7090
Middle East and Northern Africa	5.2490	4.8970	4.9390	4.8620
North America	7.2035	7.0290	7.0265	7.0265
Southeastern Asia	5.1365	5.3175	5.4280	5.0800
Southern Asia	4.3205	4.4670	4.3280	4.7525
Sub-Saharan Africa	4.4325	4.4860	4.6190	4.7395
Western Europe	7.0610	6.9620	7.0960	6.9370

Q3: Median by Region and Year

Q4: Mean by Region and Year in Table and Graph

	Life Ladder			
year	2017	2018	2019	2020
Region				
Australia and New Zealand	7.292000	7.273500	7.219500	7.197000
Central and Eastern Europe	5.591964	5.683214	5.759815	5.984727
Eastern Asia	5.656500	5.739400	5.785667	5.956500
Latin America and Caribbean	6.002143	5.951167	6.117579	5.660182
Middle East and Northern Africa	5.286824	5.055667	5.288625	5.363455
North America	7.203500	7.029000	7.026500	7.026500
Southeastern Asia	5.193375	5.382100	5.504333	5.011400
Southern Asia	4.319500	4.304667	4.140500	4.752500
Sub-Saharan Africa	4.353278	4.533800	4.499314	4.761571
Western Europe	6.812952	6.819095	6.853364	6.826150




```
#Ex. 1: Pivot Tables
#Source: VanderPlas, 2017 (Chp. V3, p.170-172)
# https://www.analyticsvidhya.com/blog/2020/03/pivot-table-pandas-python/

import numpy as np
import pandas as pd
import seaborn as sns
titanic = sns.load_dataset('titanic')
titanic.head()

# getting an overview of our data
print("The dataset has {0} rows and {1} columns".format(titanic.shape[0], titanic.shape[1]))
# checking for missing values
print("Are there missing values? {}".format(titanic.isnull().any().any()))
# general information about column data types and number of values
titanic.info()

#Q1: What is the percentage of people who survived by gender and class?

#Pivot Tables by hand with Aggregate
titanic.groupby(['sex', 'class'])['survived'].aggregate('mean').unstack()

#Pivot Tables with the method
titanic.pivot_table('survived', index='sex', columns='class')

#Using the margins parameter to calculate subtotals
titanic.pivot_table('survived', index='sex', columns='class', margins=True)

#Q2: In absolute values, how many people survived per gender and class?
```

```
table =  titanic.pivot_table('survived', index='sex',columns='class', aggfunc='sum' )
table
#Show table above graphically in a bar chart
table.plot(kind='bar')

#Q3: How many people boarded the titanic regardless of how many survived?
titanic.pivot_table('who',index='sex',columns='class', aggfunc='count', margins=True)

#Q4: How many survived and how much they paid on average?
titanic.pivot_table(index='sex', columns='class',
                    aggfunc={'survived':sum, 'fare':'mean'})
#Ex: Visualization Tutorial
#Source: https://realpython.com/pandas-plot-python/
import pandas as pd

d_url =
("https://raw.githubusercontent.com/fivethirtyeight/data/master/college-majors/recent-grads.csv")
df = pd.read_csv(d_url)
pd.set_option("display.max.columns", None)
df.head()

%matplotlib inline

#Show the distribution of earnings graphically
df.plot(x="Rank", y=["P25th", "Median", "P75th"])

#Survey your data by creating a histogram for the median column
median_column = df["Median"]
median_column.plot(kind="hist")

#Analysis of outliers
top_5 = df.sort_values(by="Median", ascending=False).head()
top_5.plot(x="Major", y="Median", kind="bar", rot=5, fontsize=4)

#Which are the majors whose median salary is above $60,000
top_medians = df[df["Median"] > 60000].sort_values("Median")
top_medians.plot(x="Major", y=["P25th", "Median", "P75th"], kind="bar")

#Correlation checks if two columns are connected (move together)
#Compare the median salary with unemployement-rate
df.plot(x="Median", y="Unemployment_rate", kind="scatter")

#Analyze categorical data by grouping
#With .groupby(), you create a DataFrameGroupBy object and
```

```
#with .sum(), you create a Series.
cat_totals = df.groupby("Major_category")["Total"].sum().sort_values()
cat_totals
cat_totals.plot(kind="barh", fontsize=4)


#Then create a pie chart to visualize ratios
#Lump smaller categories (with a total under 100K) into "Other"
small_cat_totals = cat_totals[cat_totals < 100_000]
big_cat_totals = cat_totals[cat_totals > 100_000]
small_sums = pd.Series([small_cat_totals.sum()], index=["Other"])
big_cat_totals = big_cat_totals.append(small_sums)
big_cat_totals.plot(kind="pie", label="")

#Analyze the distribution of data within a category
df[df["Major_category"] == "Engineering"]["Median"].plot(kind="hist")
```