Lab Assignment 2: How to Load CSV, ASCII, and other data into Python

DS 6001: Practice and Application of Data Science

Instructions

Please answer the following questions as completely as possible using text, code, and the results of code as needed. Format your answers in a Jupyter notebook. To receive full credit, make sure you address every part of the problem, and make sure your document is formatted in a clean and professional way.

There are 11 data files attached to this lab assignment, with different extensions. First, download all of these data files, and save them in the same folder on your local machine. Your task in the following questions is to load each file into Python correctly, so that you can begin the process of data cleaning. If the variable names are included in the file, use those names to name the columns. If the variable names are not included, use these names in order:

In [1]:
```
column_names = ["Country", "Happiness score", "Whisker-high", "Whisker-low",
  "Dystopia (1.92) + residual", "Explained by: GDP per capita",
  "Explained by: Social support", "Explained by: Healthy life expectancy",
  "Explained by: Freedom to make life choices", "Explained by: Generosity",
  "Explained by: Perceptions of corruption" ]
```

If you loaded the data correctly, it will look like `data_clean.csv`, which is also attached to this lab.

Problem 0

Import the libraries you will need. Then write code to change the working directory to the folder in which you saved the data files, run the code displayed above to create the `column_names` list, load `data_clean.csv`, and display the output of the `.info()` method of `data_clean`. (1 point)

In [38]:
```
import pandas as pd
import numpy as np
import os
os.chdir("/Users/jacqu/Downloads/lab data/lab data/")
dc = pd.read_csv("data_clean.csv")
dc.info()
import pyreadstat as prs
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 156 entries, 0 to 155
Data columns (total 11 columns):
 #   Column                                      Non-Null Count  Dtype  
---  ------                                      --------------  -----  
 0   Country                                     156 non-null    object 
 1   Happiness score                             156 non-null    float64
 2   Whisker-high                                156 non-null    float64
 3   Whisker-low                                 156 non-null    float64
 4   Dystopia (1.92) + residual                  156 non-null    float64
 5   Explained by: GDP per capita                156 non-null    float64
 6   Explained by: Social support                156 non-null    float64
 7   Explained by: Healthy life expectancy       156 non-null    float64
 8   Explained by: Freedom to make life choices  156 non-null    float64
 9   Explained by: Generosity                    156 non-null    float64
 10  Explained by: Perceptions of corruption     156 non-null    float64
dtypes: float64(10), object(1)
memory usage: 13.5+ KB
```

Problem 1

Load `data1.csv`. Use the tools we discussed in class to decide whether the data file loaded correctly, and include that code in your lab report. In one or two sentences, describe how you decided on the right combination of parameters needed to load the data. (1 point)

Answer 1

In [3]:
```python
d1 = pd.read_csv("data1.csv", skiprows=[0,1])
print(d1.info())
d1.head()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 156 entries, 0 to 155
Data columns (total 11 columns):
 #   Column                                      Non-Null Count  Dtype  
---  ------                                      --------------  -----  
 0   Country                                     156 non-null    object 
 1   Happiness score                             156 non-null    float64
 2   Whisker-high                                156 non-null    float64
 3   Whisker-low                                 156 non-null    float64
 4   Dystopia (1.92) + residual                  156 non-null    float64
 5   Explained by: GDP per capita                156 non-null    float64
 6   Explained by: Social support                156 non-null    float64
 7   Explained by: Healthy life expectancy       156 non-null    float64
 8   Explained by: Freedom to make life choices  156 non-null    float64
 9   Explained by: Generosity                    156 non-null    float64
 10  Explained by: Perceptions of corruption     156 non-null    float64
dtypes: float64(10), object(1)
memory usage: 13.5+ KB
None
```

Out[3]:

	Country	Happiness score	Whisker-high	Whisker-low	Dystopia (1.92) + residual	Explained by: GDP per capita	Explained by: Social support	Explained by: Healthy life expectancy	Explained by: Freedom to make life choices
0	Finland	7.632	7.695	7.569	2.595	1.305	1.592	0.874	0.681
1	Norway	7.594	7.657	7.530	2.383	1.456	1.582	0.861	0.686

I first loaded the dataset as is with pd.read_csv(filename). I realised that the first 2 rows were just information about the data, so to get rid of it, I Googled "read_csv pandas only certain rows" and went to this SO page: https://stackoverflow.com/questions/39339142/pandas-read-csv-and-keep-only-certain-rows-python. There, I got a new parameter for read_csv that I didn't know existed until now, and utilised it to skip the first two rows. Python index starts at 0.

Problem 2

Load `data2.txt`. Use the tools we discussed in class to decide whether the data file loaded correctly, and include that code in your lab report. In one or two sentences, describe how you decided on the right combination of parameters needed to load the data. (1 point)

Answer 2

In [4]:
```
# d2 = pd.read_fwf("data2.txt")
d2 = pd.read_csv("data2.txt", sep=",", header=None, skiprows=[0,1,3])
d2.columns = d2.iloc[0]
d2 = d2.iloc[1:]
print(d2.info())
d2.head()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 158 entries, 1 to 158
Data columns (total 11 columns):
 #   Column                                 Non-Null Count  Dtype 
---  ------                                 --------------  ----- 
 0   Country                                158 non-null    object
 1   Happiness score                        156 non-null    object
 2   Whisker-high                           156 non-null    object
 3   Whisker-low                            156 non-null    object
 4   Dystopia (1.92) + residual             156 non-null    object
 5   Explained by: GDP per capita           156 non-null    object
 6   Explained by: Social support           156 non-null    object
 7   Explained by: Healthy life expectancy  156 non-null    object
 8   Explained by: Freedom to make life choices 156 non-null    object
 9   Explained by: Generosity               156 non-null    object
 10  Explained by: Perceptions of corruption    156 non-null    object
dtypes: object(11)
memory usage: 13.7+ KB
None
```

Out[4]:

	Country	Happiness score	Whisker-high	Whisker-low	Dystopia (1.92) + residual	Explained by: GDP per capita	Explained by: Social support	Explained by: Healthy life expectancy	Explained by: Freedom to make life choices
1	Finland	7.632	7.695	7.569	2.595	1.305	1.592	0.874	0.681
2	Norway	7.594	7.657	7.530	2.383	1.456	1.582	0.861	0.686

I first tried read_fwf() just to see what would happen and realised that the data was separated by commas (no space) and there were some rows similar to the problem 1 that we didn't need, so I switched to read_csv and used the documentation: https://pandas.pydata.org/docs/reference/api/pandas.read_csv.html to find the right parameters to use. Then I just identified the row with the column names and assigned that as the new column names.

Problem 3

Load `data3.txt`. Use the tools we discussed in class to decide whether the data file loaded correctly, and include that code in your lab report. In one or two sentences, describe how you decided on the right combination of parameters needed to load the data. (1 point)

Answer 3

In [5]:
```
# d3 = pd.read_csv("data3.txt")
d3 = pd.read_csv("data3.txt", sep= "\t", header=None, skiprows=[0,1])
d3.columns = d3.iloc[0]
d3 = d3.iloc[1:]
print(d3.info())
d3.head()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 156 entries, 1 to 156
Data columns (total 11 columns):
 #   Column                                    Non-Null Count  Dtype 
---  ------                                    --------------  ----- 
 0   Country                                   156 non-null    object
 1   Happiness score                           156 non-null    object
 2   Whisker-high                              156 non-null    object
 3   Whisker-low                               156 non-null    object
 4   Dystopia (1.92) + residual                156 non-null    object
 5   Explained by: GDP per capita              156 non-null    object
 6   Explained by: Social support              156 non-null    object
 7   Explained by: Healthy life expectancy     156 non-null    object
 8   Explained by: Freedom to make life choices  156 non-null    object
 9   Explained by: Generosity                  156 non-null    object
 10  Explained by: Perceptions of corruption   156 non-null    object
dtypes: object(11)
memory usage: 13.5+ KB
None
```

Out[5]:

	Country	Happiness score	Whisker-high	Whisker-low	Dystopia (1.92) + residual	Explained by: GDP per capita	Explained by: Social support	Explained by: Healthy life expectancy	Explained by: Freedom to make life choices
1	Finland	7.632	7.695	7.569	2.595	1.305	1.592	0.874	0.681
2	Norway	7.594	7.657	7.53	2.383	1.456	1.582	0.861	0.686

I used the same strategy I used in problem 2.

Problem 4

Load `data4.txt`. Use the tools we discussed in class to decide whether the data file loaded correctly, and include that code in your lab report. In one or two sentences, describe how you decided on the right combination of parameters needed to load the data. (1 point)

Answer 4

In [17]:
```
# d4 = pd.read_csv("data4.txt")
d4 = pd.read_csv("data4.txt", header=None, sep="$")
d4.columns = column_names
print(d4.info())
d4.head()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 156 entries, 0 to 155
Data columns (total 11 columns):
 #   Column                                     Non-Null Count  Dtype  
---  ------                                     --------------  -----  
 0   Country                                    156 non-null    object 
 1   Happiness score                            156 non-null    float64
 2   Whisker-high                               156 non-null    float64
 3   Whisker-low                                156 non-null    float64
 4   Dystopia (1.92) + residual                 156 non-null    float64
 5   Explained by: GDP per capita               156 non-null    float64
 6   Explained by: Social support               156 non-null    float64
 7   Explained by: Healthy life expectancy      156 non-null    float64
 8   Explained by: Freedom to make life choices 156 non-null    float64
 9   Explained by: Generosity                   156 non-null    float64
 10  Explained by: Perceptions of corruption    156 non-null    float64
dtypes: float64(10), object(1)
memory usage: 13.5+ KB
None
```

Out[17]:

	Country	Happiness score	Whisker-high	Whisker-low	Dystopia (1.92) + residual	Explained by: GDP per capita	Explained by: Social support	Explained by: Healthy life expectancy	Explained by: Freedom to make life choices
0	Finland	7.632	7.695	7.569	2.595	1.305	1.592	0.874	0.681
1	Norway	7.594	7.657	7.530	2.383	1.456	1.582	0.861	0.686
2	Denmark	7.555	7.623	7.487	2.370	1.351	1.590	0.868	0.683
3	Iceland	7.495	7.593	7.398	2.426	1.343	1.644	0.914	0.677
4	Switzerland	7.487	7.570	7.405	2.320	1.420	1.549	0.927	0.660

When I viewed the data table with pd.read_csv("data4.txt"), everything was just smushed together. It took me awhile to figure out that I should open the txt file in a notepad app on my laptop. From there, I saw that the data was separated with dollar sign, so I just followed the coding parameters I used in problem 2 and 3, but this time with a separator of '$'.

Problem 5

Load `data5.csv`. Use the tools we discussed in class to decide whether the data file loaded correctly, and include that code in your lab report. In one or two sentences, describe how you decided on the right combination of parameters needed to load the data. (1 point)

Answer 5

In [19]:
```
d5 = pd.read_csv("data5.csv")
print(d5.info())
d5.head()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 158 entries, 0 to 157
Data columns (total 11 columns):
 #   Column                                      Non-Null Count  Dtype
---  ------                                      --------------  -----
 0   Country                                     158 non-null    object
 1   Happiness score                             156 non-null    float64
 2   Whisker-high                                156 non-null    float64
 3   Whisker-low                                 156 non-null    float64
 4   Dystopia (1.92) + residual                  156 non-null    float64
 5   Explained by: GDP per capita                156 non-null    float64
 6   Explained by: Social support                156 non-null    float64
 7   Explained by: Healthy life expectancy       156 non-null    float64
 8   Explained by: Freedom to make life choices  156 non-null    float64
 9   Explained by: Generosity                    156 non-null    float64
 10  Explained by: Perceptions of corruption     156 non-null    float64
dtypes: float64(10), object(1)
memory usage: 13.7+ KB
None
```

Out[19]:

	Country	Happiness score	Whisker-high	Whisker-low	Dystopia (1.92) + residual	Explained by: GDP per capita	Explained by: Social support	Explained by: Healthy life expectancy	Explained by: Freedom to make life choices
0	Finland	7.632	7.695	7.569	2.595	1.305	1.592	0.874	0.681
1	Norway	7.594	7.657	7.530	2.383	1.456	1.582	0.861	0.686
2	Denmark	7.555	7.623	7.487	2.370	1.351	1.590	0.868	0.683
3	Iceland	7.495	7.593	7.398	2.426	1.343	1.644	0.914	0.677
4	Switzerland	7.487	7.570	7.405	2.320	1.420	1.549	0.927	0.660

I didn't find anything wrong with the format of the data table after running the code through read_csv, so I didn't make any modifications.

Problem 6

Load `data6.dat`. Use the tools we discussed in class to decide whether the data file loaded correctly, and include that code in your lab report. In one or two sentences, describe how you decided on the right combination of parameters needed to load the data. (1 point)

Answer 6

In [50]:

```python
d6 = pd.read_csv("data6.dat")
d6 = d6.replace(999.000, np.NaN)
print(d6.info())
d6.head()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 156 entries, 0 to 155
Data columns (total 11 columns):
 #   Column                                      Non-Null Count  Dtype  
---  ------                                      --------------  -----  
 0   Country                                     156 non-null    object 
 1   Happiness score                             142 non-null    float64
 2   Whisker-high                                135 non-null    float64
 3   Whisker-low                                 136 non-null    float64
 4   Dystopia (1.92) + residual                  145 non-null    float64
 5   Explained by: GDP per capita                137 non-null    float64
 6   Explained by: Social support                134 non-null    float64
 7   Explained by: Healthy life expectancy       142 non-null    float64
 8   Explained by: Freedom to make life choices  140 non-null    float64
 9   Explained by: Generosity                    145 non-null    float64
 10  Explained by: Perceptions of corruption     143 non-null    float64
dtypes: float64(10), object(1)
memory usage: 13.5+ KB
None
```

Out[50]:

	Country	Happiness score	Whisker-high	Whisker-low	Dystopia (1.92) + residual	Explained by: GDP per capita	Explained by: Social support	Explained by: Healthy life expectancy	Explained by: Freedom to make life choices
0	Finland	7.632	7.695	7.569	2.595	NaN	NaN	NaN	0.681
1	Norway	7.594	7.657	7.530	NaN	NaN	1.582	NaN	0.686
2	Denmark	7.555	7.623	7.487	2.370	1.351	1.590	NaN	0.683
3	Iceland	7.495	7.593	NaN	2.426	1.343	1.644	0.914	0.677
4	Switzerland	7.487	7.570	7.405	2.320	1.420	1.549	0.927	0.660

I didn't find anything wrong with the format of the data table after running the code through read_csv, so I didn't make any modifications. I was a little alarmed at the 999.000 data points, so I just changed it to NaN, but other than that, I just did everything like normal.

Problem 7

Load `data7.xlsx`, which is an Excel file. Keep only the sheet named "Data". Use the tools we discussed in class to decide whether the data file loaded correctly, and include that code in your lab report. In one or two sentences, describe how you decided on the right combination of parameters needed to load the data. (2 points)

Answer 7

In [24]:
```
d7 = pd.read_excel("data7.xlsx",sheet_name=1)
print(d7.info())
d7.head()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 156 entries, 0 to 155
Data columns (total 11 columns):
 #   Column                                      Non-Null Count  Dtype
---  ------                                      --------------  -----
 0   Country                                     156 non-null    object
 1   Happiness score                             156 non-null    float64
 2   Whisker-high                                156 non-null    float64
 3   Whisker-low                                 156 non-null    float64
 4   Dystopia (1.92) + residual                  156 non-null    float64
 5   Explained by: GDP per capita                156 non-null    float64
 6   Explained by: Social support                156 non-null    float64
 7   Explained by: Healthy life expectancy       156 non-null    float64
 8   Explained by: Freedom to make life choices  156 non-null    float64
 9   Explained by: Generosity                    156 non-null    float64
 10  Explained by: Perceptions of corruption     156 non-null    float64
dtypes: float64(10), object(1)
memory usage: 13.5+ KB
None
```

Out[24]:

	Country	Happiness score	Whisker-high	Whisker-low	Dystopia (1.92) + residual	Explained by: GDP per capita	Explained by: Social support	Explained by: Healthy life expectancy	Explained by: Freedom to make life choices
0	Finland	7.632	7.695	7.569	2.595	1.305	1.592	0.874	0.681
1	Norway	7.594	7.657	7.530	2.383	1.456	1.582	0.861	0.686
2	Denmark	7.555	7.623	7.487	2.370	1.351	1.590	0.868	0.683
3	Iceland	7.495	7.593	7.398	2.426	1.343	1.644	0.914	0.677
4	Switzerland	7.487	7.570	7.405	2.320	1.420	1.549	0.927	0.660

I tried read_csv first, and when that didn't work, I Googled if there was a way to load an excel file, resulting in me finding this documentation: https://pandas.pydata.org/docs/reference/api/pandas.read_excel.html. I then opened up the excel file in MS Excel to find the sheet name, realised that I could use the index of the sheet (in this case, 1), so I used that as my parameter.

Problem 8

Load `data8.dta`, which is a Stata 13 file. Use the tools we discussed in class to decide whether the data file loaded correctly, and include that code in your lab report. In one or two sentences, describe how you decided on the right combination of parameters needed to load the data. (2 points)

Answer 8

In [51]:
```
d8 = pd.read_stata("data8.dta", columns=None)
d8.columns = column_names
print(d8.info())
d8.head()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 156 entries, 0 to 155
Data columns (total 11 columns):
 #   Column                                      Non-Null Count  Dtype  
---  ------                                      --------------  -----  
 0   Country                                     156 non-null    object 
 1   Happiness score                             156 non-null    float32
 2   Whisker-high                                156 non-null    float32
 3   Whisker-low                                 156 non-null    float32
 4   Dystopia (1.92) + residual                  156 non-null    float32
 5   Explained by: GDP per capita                156 non-null    float32
 6   Explained by: Social support                156 non-null    float32
 7   Explained by: Healthy life expectancy       156 non-null    float32
 8   Explained by: Freedom to make life choices  156 non-null    float32
 9   Explained by: Generosity                    156 non-null    float32
 10  Explained by: Perceptions of corruption     156 non-null    float32
dtypes: float32(10), object(1)
memory usage: 8.5+ KB
None
```

Out[51]:

	Country	Happiness score	Whisker-high	Whisker-low	Dystopia (1.92) + residual	Explained by: GDP per capita	Explained by: Social support	Explained by: Healthy life expectancy	Explained by: Freedom to make life choices
0	Finland	7.632	7.695	7.569	2.595	1.305	1.592	0.874	0.681
1	Norway	7.594	7.657	7.530	2.383	1.456	1.582	0.861	0.686
2	Denmark	7.555	7.623	7.487	2.370	1.351	1.590	0.868	0.683
3	Iceland	7.495	7.593	7.398	2.426	1.343	1.644	0.914	0.677
4	Switzerland	7.487	7.570	7.405	2.320	1.420	1.549	0.927	0.660

I remember that we went over read_stata in class, so I used that. And then I Googled the documentation found here: https://pandas.pydata.org/docs/reference/api/pandas.read_stata.html and got the parameter columns to make sure I didn't have any column names (so that I could declare the names later).

Problem 9

Load `data9.sav`, which is an SPSS file. Use the tools we discussed in class to decide whether the data file loaded correctly, and include that code in your lab report. In one or two sentences, describe how you decided on the right combination of parameters needed to load the data. (2 points)

Answer 9

In [52]:
```python
d9 = pd.read_spss("data9.sav")
d9.columns = column_names
```

```
print(d9.info())
d9.head()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 156 entries, 0 to 155
Data columns (total 11 columns):
 #   Column                                      Non-Null Count  Dtype  
---  ------                                      --------------  -----  
 0   Country                                     156 non-null    object 
 1   Happiness score                             156 non-null    float64
 2   Whisker-high                                156 non-null    float64
 3   Whisker-low                                 156 non-null    float64
 4   Dystopia (1.92) + residual                  156 non-null    float64
 5   Explained by: GDP per capita                156 non-null    float64
 6   Explained by: Social support                156 non-null    float64
 7   Explained by: Healthy life expectancy       156 non-null    float64
 8   Explained by: Freedom to make life choices  156 non-null    float64
 9   Explained by: Generosity                    156 non-null    float64
 10  Explained by: Perceptions of corruption     156 non-null    float64
dtypes: float64(10), object(1)
memory usage: 13.5+ KB
None
```

Out[52]:

	Country	Happiness score	Whisker-high	Whisker-low	Dystopia (1.92) + residual	Explained by: GDP per capita	Explained by: Social support	Explained by: Healthy life expectancy	Explained by: Freedom to make life choices
0	Finland	7.632	7.695	7.569	2.595	1.305	1.592	0.874	0.681
1	Norway	7.594	7.657	7.530	2.383	1.456	1.582	0.861	0.686
2	Denmark	7.555	7.623	7.487	2.370	1.351	1.590	0.868	0.683
3	Iceland	7.495	7.593	7.398	2.426	1.343	1.644	0.914	0.677
4	Switzerland	7.487	7.570	7.405	2.320	1.420	1.549	0.927	0.660

First I tried read_csv just because, then Googled how to import spss files, leading me to the documentation: https://pandas.pydata.org/docs/reference/api/pandas.read_spss.html. Then, I got an error code, so I read the error code, realised I didn't have pyreadstat installed, so I went ahead and installed it. Then I just loaded it like normal with read_spss and set the column names with column_names.

Problem 10

Load `data10.xpt`, which is a SAS file. Use the tools we discussed in class to decide whether the data file loaded correctly, and include that code in your lab report. In one or two sentences, describe how you decided on the right combination of parameters needed to load the data. (If some of the country names display as `b'Finland'`, don't worry aout that.) (2 points)

Answer 10

In [54]:
```
d10 = pd.read_sas("data10.xpt")
d10.columns = column_names
print(d10.info())
d10.head()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 156 entries, 0 to 155
Data columns (total 11 columns):
 #   Column                                    Non-Null Count  Dtype  
---  ------                                    --------------  -----  
 0   Country                                   156 non-null    object 
 1   Happiness score                           156 non-null    float64
 2   Whisker-high                              156 non-null    float64
 3   Whisker-low                               156 non-null    float64
 4   Dystopia (1.92) + residual                156 non-null    float64
 5   Explained by: GDP per capita              156 non-null    float64
 6   Explained by: Social support              156 non-null    float64
 7   Explained by: Healthy life expectancy     156 non-null    float64
 8   Explained by: Freedom to make life choices 156 non-null    float64
 9   Explained by: Generosity                  156 non-null    float64
 10  Explained by: Perceptions of corruption   156 non-null    float64
dtypes: float64(10), object(1)
memory usage: 13.5+ KB
None
```

Out[54]:

	Country	Happiness score	Whisker-high	Whisker-low	Dystopia (1.92) + residual	Explained by: GDP per capita	Explained by: Social support	Explained by: Healthy life expectancy	Explain b Freedo to ma li choic
0	b'Finland'	7.632	7.695	7.569	2.595	1.305	1.592	0.874	0.6
1	b'Norway'	7.594	7.657	7.530	2.383	1.456	1.582	0.861	0.6
2	b'Denmark'	7.555	7.623	7.487	2.370	1.351	1.590	0.868	0.6
3	b'Iceland'	7.495	7.593	7.398	2.426	1.343	1.644	0.914	0.6
4	b'Switzerland'	7.487	7.570	7.405	2.320	1.420	1.549	0.927	0.6

I ran into read_sas while looking at the spss documentation: https://pandas.pydata.org/docs/reference/api/pandas.read_sas.html. So I just utilized that and the column_names to format it correctly, no extra parameters needed.

Problem 11

Please load the `data11.txt` file, which is a fixed width file. The columns are defined as follows:

Variable	Width	Start	End
Country	24	1	24
Happiness score	5	25	29
Whisker-high	5	30	34
Whisker-low	5	35	39
Dystopia (1.92) + residual	5	40	44
Explained by: GDP per capita	5	45	49
Explained by: Social support	5	50	54
Explained by: Healthy life expectancy	5	55	59
Explained by: Freedom to make life choices	5	60	64
Explained by: Generosity	5	65	69
Explained by: Perceptions of corruption	5	70	74

Then save the this loaded data frame as a CSV file on your local machine. Be sure to use a unique filename so as not to overwrite any existing files. (5 points)

Answer 11

In [58]:
```
widths = [24,5,5,5,5,5,5,5,5,5,5]
d11 = pd.read_fwf("data11.txt", widths=widths, header=None)
d11.columns = column_names
print(d11.info())
d11.head()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 156 entries, 0 to 155
Data columns (total 11 columns):
 #   Column                                      Non-Null Count  Dtype  
---  ------                                      --------------  -----  
 0   Country                                     156 non-null    object 
 1   Happiness score                             156 non-null    float64
 2   Whisker-high                                156 non-null    float64
 3   Whisker-low                                 156 non-null    float64
 4   Dystopia (1.92) + residual                  156 non-null    float64
 5   Explained by: GDP per capita                156 non-null    float64
 6   Explained by: Social support                156 non-null    float64
 7   Explained by: Healthy life expectancy       156 non-null    float64
 8   Explained by: Freedom to make life choices  156 non-null    float64
 9   Explained by: Generosity                    156 non-null    float64
 10  Explained by: Perceptions of corruption     156 non-null    float64
dtypes: float64(10), object(1)
memory usage: 13.5+ KB
None
```

Out[58]:

	Country	Happiness score	Whisker-high	Whisker-low	Dystopia (1.92) + residual	Explained by: GDP per capita	Explained by: Social support	Explained by: Healthy life expectancy	Explained by: Freedom to make life choices
0	Finland	7.632	7.695	7.569	2.595	1.305	1.592	0.874	0.681
1	Norway	7.594	7.657	7.530	2.383	1.456	1.582	0.861	0.686
2	Denmark	7.555	7.623	7.487	2.370	1.351	1.590	0.868	0.683
3	Iceland	7.495	7.593	7.398	2.426	1.343	1.644	0.914	0.677
4	Switzerland	7.487	7.570	7.405	2.320	1.420	1.549	0.927	0.660

In [62]:
```
d11.to_csv("data11_change.csv")
```

In [ ]: