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

DS 6001: Practice and Application of Data Science

Brian Nolton

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 [18]:
```
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 [1]:
```
import pandas as pd
import numpy as np
import os

os.chdir("C:/Users/brian/OneDrive/Documents/DS 6001/lab2 data/lab data")
```

```
data_clean = pd.read_csv("data_clean.csv")
data_clean.info()
```

```
<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
```

For the sake of brevity I didn't show everything I did to check to see if the data file loaded correctly. I show the read command with all of its fine tunings and the head command so you can see how it works.

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)

In [104…

```
data1 = pd.read_csv("data1.csv", header=2)
data1.head()
```

Out[104…

	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
0	Finland	7.632	7.695	7.569	2.595	1.305	1.592	0.874
1	Norway	7.594	7.657	7.530	2.383	1.456	1.582	0.861
2	Denmark	7.555	7.623	7.487	2.370	1.351	1.590	0.868
3	Iceland	7.495	7.593	7.398	2.426	1.343	1.644	0.914
4	Switzerland	7.487	7.570	7.405	2.320	1.420	1.549	0.927

Initially, there were rows above where the data was, so I used header=2 to eliminate them.

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)

In [103…
```
data2 = pd.read_csv("data2.txt", skiprows=4, names=column_names, comment='/')
data2.head()
```

Out[103…

	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
0	Finland	7.632	7.695	7.569	2.595	1.305	1.592	0.874
1	Norway	7.594	7.657	7.530	2.383	1.456	1.582	0.861
2	Denmark	7.555	7.623	7.487	2.370	1.351	1.590	0.868
3	Iceland	7.495	7.593	7.398	2.426	1.343	1.644	0.914
4	Switzerland	7.487	7.570	7.405	2.320	1.420	1.549	0.927

The data file had rows above that needed to be skipped, but I couldn't use header even though the names I wanted as the column names were there, I skipped the first 4 rows and used the column names given at the begginning of the document. I also noticed there were comments in the data preceeded by a '/', so I eliminated those as well.

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)

In [102…
```
data3 = pd.read_csv("data3.txt", header=2, sep='\t')
data3.head()
```

Out[102…

	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
0	Finland	7.632	7.695	7.569	2.595	1.305	1.592	0.874
1	Norway	7.594	7.657	7.530	2.383	1.456	1.582	0.861
2	Denmark	7.555	7.623	7.487	2.370	1.351	1.590	0.868
3	Iceland	7.495	7.593	7.398	2.426	1.343	1.644	0.914
4	Switzerland	7.487	7.570	7.405	2.320	1.420	1.549	0.927

This data set also had text above, but this time everything was separated by tab instead of a comma. I used header=3 for the first part and changed the sep to '/t' for the tabs.

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)

In [101…

```
data4 = pd.read_csv("data4.txt", sep='$', header=None, names=column_names)
data4.head()
```

Out[101…

	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
0	Finland	7.632	7.695	7.569	2.595	1.305	1.592	0.874
1	Norway	7.594	7.657	7.530	2.383	1.456	1.582	0.861
2	Denmark	7.555	7.623	7.487	2.370	1.351	1.590	0.868
3	Iceland	7.495	7.593	7.398	2.426	1.343	1.644	0.914
4	Switzerland	7.487	7.570	7.405	2.320	1.420	1.549	0.927

This data set had no text or headers and was separated by a '
'. $I used header = None and the column names provided for the first part, and changed the s$
' for the second part.

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)

In [106…
```
data5 = pd.read_csv("data5.csv", skiprows=(157,158))
data5.tail()
```

Out[106…

	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
151	Yemen	3.355	3.448	3.262	1.106	0.442	1.073	0.343
152	Tanzania	3.303	3.414	3.193	0.628	0.455	0.991	0.381
153	South Sudan	3.254	3.385	3.123	1.691	0.337	0.608	0.177
154	Central African Republic	3.083	3.227	2.939	2.487	0.024	0.000	0.010
155	Burundi	2.905	3.074	2.735	1.752	0.091	0.627	0.145

This dataset had a footnote on the bottom 2 rows. I removed them with skiprows and called out the 2 specific rows.

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)

In [99]:
```
data6 = pd.read_csv("data6.dat", na_values=999)
data6.head()
```

Out[99]:

	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
0	Finland	7.632	7.695	7.569	2.595	NaN	NaN	NaN
1	Norway	7.594	7.657	7.530	NaN	NaN	1.582	NaN
2	Denmark	7.555	7.623	7.487	2.370	1.351	1.590	NaN
3	Iceland	7.495	7.593	NaN	2.426	1.343	1.644	0.914
4	Switzerland	7.487	7.570	7.405	2.320	1.420	1.549	0.927

In this dataset there were a bunch of 999s in spots that don't make sense. I set all these values to NaNs.

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)

In [127…

```
data7 = pd.read_excel("data7.xlsx", sheet_name="Data")
data7.head()
```

Out[127…

	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
0	Finland	7.632	7.695	7.569	2.595	1.305	1.592	0.874
1	Norway	7.594	7.657	7.530	2.383	1.456	1.582	0.861
2	Denmark	7.555	7.623	7.487	2.370	1.351	1.590	0.868
3	Iceland	7.495	7.593	7.398	2.426	1.343	1.644	0.914
4	Switzerland	7.487	7.570	7.405	2.320	1.420	1.549	0.927

This dataset was an Excel workbook so I had to use read_excel and specify the "Data" sheet. I had to do a pip install to be able to read it.

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)

In [131…
```
data8 = pd.read_stata("data8.dta")
data8.columns=column_names
data8.head()
```

Out[131…

	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
0	Finland	7.632	7.695	7.569	2.595	1.305	1.592	0.874
1	Norway	7.594	7.657	7.530	2.383	1.456	1.582	0.861
2	Denmark	7.555	7.623	7.487	2.370	1.351	1.590	0.868
3	Iceland	7.495	7.593	7.398	2.426	1.343	1.644	0.914
4	Switzerland	7.487	7.570	7.405	2.320	1.420	1.549	0.927

This dataset was a Stata file so I had to use read_stata. The column names had no spaces so I decided to change them to the predefined names.

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)

In [124…
```
data9 = pd.read_spss("data9.sav")
data9.head()
```

Out[124...

	country	happiness	whiskerhigh	whiskerlow	dystopia	gdpPC	socsupport	lifeexp
0	Finland	7.632	7.695	7.569	2.595	1.305	1.592	0.874
1	Norway	7.594	7.657	7.530	2.383	1.456	1.582	0.861
2	Denmark	7.555	7.623	7.487	2.370	1.351	1.590	0.868
3	Iceland	7.495	7.593	7.398	2.426	1.343	1.644	0.914
4	Switzerland	7.487	7.570	7.405	2.320	1.420	1.549	0.927

This data file was a SPSS file so i used read_spss. I had to do a pip install to be able to read it. The column names are a little different but I decided not to change them.

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)

In [133...

```
data10= pd.read_sas("data10.xpt", encoding="utf-8")
data10.columns=column_names
data10.head()
```

Out[133...

	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
0	Finland	7.632	7.695	7.569	2.595	1.305	1.592	0.874
1	Norway	7.594	7.657	7.530	2.383	1.456	1.582	0.861
2	Denmark	7.555	7.623	7.487	2.370	1.351	1.590	0.868
3	Iceland	7.495	7.593	7.398	2.426	1.343	1.644	0.914
4	Switzerland	7.487	7.570	7.405	2.320	1.420	1.549	0.927

This dataset required read_sas, however when doing that standalone there was a 'b' in front of each country, I used the utf-8 encoder to solve this. I also had to fix the column names but had to use the .columns command instead of the names parameter.

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)

In [77]:
```
colwidths = pd.read_csv("colwidths.csv")
data11 = pd.read_fwf("data11.txt", widths=colwidths["Widths"], names=column_names)
data11.head()
```

Out[77]:

	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
0	Finland	7.632	7.695	7.569	2.595	1.305	1.592	0.874
1	Norway	7.594	7.657	7.530	2.383	1.456	1.582	0.861
2	Denmark	7.555	7.623	7.487	2.370	1.351	1.590	0.868
3	Iceland	7.495	7.593	7.398	2.426	1.343	1.644	0.914
4	Switzerland	7.487	7.570	7.405	2.320	1.420	1.549	0.927

I originally wrote the one below, and the reread the directions and saw the CSV thing...

In [73]:
```
widths = [24,5,5,5,5,5,5,5,5,5,5]
data11 = pd.read_fwf("data11.txt", widths=widths, names=column_names)
data11.head()
```

Out[73]:

	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
0	Finland	7.632	7.695	7.569	2.595	1.305	1.592	0.874
1	Norway	7.594	7.657	7.530	2.383	1.456	1.582	0.861
2	Denmark	7.555	7.623	7.487	2.370	1.351	1.590	0.868
3	Iceland	7.495	7.593	7.398	2.426	1.343	1.644	0.914
4	Switzerland	7.487	7.570	7.405	2.320	1.420	1.549	0.927