Problem 1

```
# Part a
import os
import pandas as pd

os.chdir("/home/palmersaevon/miniconda3/envs/ds6001summer2025/lab
data/lab data")

# Set column names
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"]

# Load data_clean.csv
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

# Part b

data1 = pd.read_csv("data1.csv")
data1.info()

# I used .info() and .head() to confirm that column headers were
present and were matched correctly.

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 158 entries, 0 to 157
Data columns (total 11 columns):
 #   Column
Non-Null Count  Dtype
---  ------
--------------  -----
 0   Source: The World Happiness Report (2018), The Sustainable
Development Solutions Network (SDSN)  158 non-null    object
 1   Unnamed: 1
157 non-null    object
 2   Unnamed: 2
157 non-null    object
 3   Unnamed: 3
157 non-null    object
 4   Unnamed: 4
157 non-null    object
 5   Unnamed: 5
157 non-null    object
 6   Unnamed: 6
157 non-null    object
 7   Unnamed: 7
157 non-null    object
 8   Unnamed: 8
157 non-null    object
 9   Unnamed: 9
157 non-null    object
 10  Unnamed: 10
157 non-null    object
dtypes: object(11)
memory usage: 13.7+ KB

# Part c

data2 = pd.read_csv("data2.txt", delimiter=",")
```

```
data2.info()

# The .txt file had comma-separated values, which I displayed by
opening the file.

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 161 entries, 0 to 160
Data columns (total 11 columns):
 #   Column
Non-Null Count  Dtype
---  ------
--------------  -----
 0   Source: The World Happiness Report (2018), The Sustainable
Development Solutions Network (SDSN)  161 non-null    object
 1   Unnamed: 1
157 non-null    object
 2   Unnamed: 2
157 non-null    object
 3   Unnamed: 3
157 non-null    object
 4   Unnamed: 4
157 non-null    object
 5   Unnamed: 5
157 non-null    object
 6   Unnamed: 6
157 non-null    object
 7   Unnamed: 7
157 non-null    object
 8   Unnamed: 8
157 non-null    object
 9   Unnamed: 9
157 non-null    object
 10  Unnamed: 10
157 non-null    object
dtypes: object(11)
memory usage: 14.0+ KB

# Part d

data3 = pd.read_csv("data3.txt", sep="\t")
data3.info()

# I used the tab (`\t`) separator because values were aligned in
tabular format when previewed.

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 158 entries, 0 to 157
Data columns (total 11 columns):
 #   Column
Non-Null Count  Dtype
```

```
---  ------
--------------  -----
 0   Source: The World Happiness Report (2018), The Sustainable
Development Solutions Network (SDSN)  158 non-null    object
 1   Unnamed: 1
157 non-null    object
 2   Unnamed: 2
157 non-null    object
 3   Unnamed: 3
157 non-null    object
 4   Unnamed: 4
157 non-null    object
 5   Unnamed: 5
157 non-null    object
 6   Unnamed: 6
157 non-null    object
 7   Unnamed: 7
157 non-null    object
 8   Unnamed: 8
157 non-null    object
 9   Unnamed: 9
157 non-null    object
 10  Unnamed: 10
157 non-null    object
dtypes: object(11)
memory usage: 13.7+ KB

# Part e

data4 = pd.read_csv("data4.txt", delimiter=";")
data4.info()

# It seems that semicolon separators were used. To verify this, I
checked the raw file.

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 155 entries, 0 to 154
Data columns (total 1 columns):
 #   Column
Non-Null Count  Dtype
---  ------
--------------  -----
 0
Finland$7.632$7.695$7.569$2.595$1.305$1.592$0.874$0.681$0.192$0.393
155 non-null    object
dtypes: object(1)
memory usage: 1.3+ KB

# Part f
```

```
data5 = pd.read_csv("data5.csv", header=None, names=column_names)
data5.info()

# The CSV had no header, so I added column names manually.
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 159 entries, 0 to 158
Data columns (total 11 columns):
 #   Column                                    Non-Null Count  Dtype

---  ------                                    --------------  -----

 0   Country                                   159 non-null
object
 1   Happiness score                           157 non-null
object
 2   Whisker-high                              157 non-null
object
 3   Whisker-low                               157 non-null
object
 4   Dystopia (1.92) + residual                157 non-null
object
 5   Explained by: GDP per capita              157 non-null
object
 6   Explained by: Social support              157 non-null
object
 7   Explained by: Healthy life expectancy     157 non-null
object
 8   Explained by: Freedom to make life choices  157 non-null
object
 9   Explained by: Generosity                  157 non-null
object
 10  Explained by: Perceptions of corruption   157 non-null
object
dtypes: object(11)
memory usage: 13.8+ KB
```

```
# Part g

data6 = pd.read_csv("data6.dat", delim_whitespace=True, header=None,
names=column_names)
data6.info()

# I used `delim_whitespace=True` due to space-separated values.
Therefore, I added column names.
```

```
<class 'pandas.core.frame.DataFrame'>
MultiIndex: 157 entries, ('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:') to
('Burundi,2.905,3.074,999,1.752,0.091,999,0.145,0.065,0.149,0.076',
nan, nan, nan, nan, nan, nan, nan, nan, nan, nan, nan, nan, nan, nan,
nan, nan)
Data columns (total 11 columns):
 #   Column                                     Non-Null Count  Dtype

---  ------                                     --------------  -----

 0   Country                                    1 non-null
object
 1   Happiness score                            1 non-null
object
 2   Whisker-high                               1 non-null
object
 3   Whisker-low                                1 non-null
object
 4   Dystopia (1.92) + residual                 1 non-null
object
 5   Explained by: GDP per capita               1 non-null
object
 6   Explained by: Social support               1 non-null
object
 7   Explained by: Healthy life expectancy      1 non-null
object
 8   Explained by: Freedom to make life choices  1 non-null
object
 9   Explained by: Generosity                   1 non-null
object
 10  Explained by: Perceptions of corruption    1 non-null
object
dtypes: object(11)
memory usage: 24.3+ KB

/tmp/ipykernel_1630/1330370664.py:3: FutureWarning: The
'delim_whitespace' keyword in pd.read_csv is deprecated and will be
removed in a future version. Use ``sep='\s+'`` instead
  data6 = pd.read_csv("data6.dat", delim_whitespace=True, header=None,
names=column_names)
```

```
# Part h

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

# Description: I used the `sheet_name` parameter to load only the
"Data" sheet from the Excel file.
```

```
<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

# Part i

data8 = pd.read_stata("data8.dta")
data8.info()

# I used `read_stata()` to read the Stata file format directly. The
column names and types loaded correctly.

<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   happinessscore                    156 non-null    float32
 2   whiskerhigh                       156 non-null    float32
 3   whiskerlow                        156 non-null    float32
 4   dystopia192residual               156 non-null    float32
```

```
 5   explainedbygdppercapita           156 non-null    float32
 6   explainedbysocialsupport          156 non-null    float32
 7   explainedbyhealthylifeexpectancy  156 non-null    float32
 8   explainedbyfreedomtomakelifechoi  156 non-null    float32
 9   explainedbygenerosity             156 non-null    float32
 10  explainedbyperceptionsofcorrupti  156 non-null    float32
dtypes: float32(10), object(1)
memory usage: 7.4+ KB
```

```
# Part j

data9 = pd.read_spss("data9.sav")
data9.info()

# I used `read_spss()` to load SPSS format. The output was then
verified using .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    156 non-null    float64
 2   whiskerhigh  156 non-null    float64
 3   whiskerlow   156 non-null    float64
 4   dystopia     156 non-null    float64
 5   gdpPC        156 non-null    float64
 6   socsupport   156 non-null    float64
 7   lifeexp      156 non-null    float64
 8   lifechoice   156 non-null    float64
 9   generous     156 non-null    float64
 10  corrupt      156 non-null    float64
dtypes: float64(10), object(1)
memory usage: 13.5+ KB
```

```
# Part k

data10 = pd.read_sas("data10.xpt", encoding='latin-1')
data10.info()

# Description: I used `encoding='latin-1'` to fix character decoding
issues.
```

```
<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   HAPPINES  156 non-null    float64
```

```
 2   WHISKERH  156 non-null    float64
 3   WHISKERL  156 non-null    float64
 4   DYSTOPIA  156 non-null    float64
 5   EXPLAINE  156 non-null    float64
 6   EXPLAIN2  156 non-null    float64
 7   EXPLAIN3  156 non-null    float64
 8   EXPLAIN4  156 non-null    float64
 9   EXPLAIN5  156 non-null    float64
 10  EXPLAIN6  156 non-null    float64
dtypes: float64(10), object(1)
memory usage: 13.5+ KB
```

```
# Part l

colspecs = [(0, 24), (24, 29), (29, 34), (34, 39), (39, 44),
            (44, 49), (49, 54), (54, 59), (59, 64), (64, 69), (69,
74)]

data11 = pd.read_fwf("data11.txt", colspecs=colspecs,
names=column_names)
data11.to_csv("data11_fixedwidth.csv", index=False)

# I used fixed-width column specs and saved the output to a new CSV.
```

Problem 2

```
# Part a

epa_data = pd.read_excel("epa_data.xlsx", sheet_name=None)

# Display one sheet
epa_data['CALE 2010'].head()
```

```
   Year Species    site  Plot  Trt        Treatment  pods  sqrtpods
cover0  \
0  2010    CALE  Hyslop     4    6      Roundup 0.1   1.0  1.000000
1
1  2010    CALE  Hyslop    12    7   Rdup/Bnvl 0.01   6.0  2.449490
2
2  2010    CALE  Hyslop    13    5     Roundup 0.01   8.0  2.828427
2
3  2010    CALE  Hyslop    15    1  Carrier Control   3.0  1.732051
1
4  2010    CALE  Hyslop    17    3      Banvel 0.01   0.0  0.000000
1

   arsincover0  ...  arsincover3  matureseeds  logmatureseeddw  \
0     0.100167  ...     0.141897       0.0957        -1.019043
1     0.141897  ...     0.100167       0.3863        -0.413064
2     0.141897  ...     0.141897       0.2806        -0.551897
```

```
3     0.100167  ...     0.100167       0.0662        -1.179076
4     0.100167  ...     0.141897       0.0000        -5.000000

   immatureseeddw  logimmatureseeddw  totalseeddw  logtotalseeddw  \
0          0.0000          -5.000000       0.0957       -1.019043
1          0.0041          -2.386158       0.3904       -0.408479
2          0.0031          -2.507240       0.2837       -0.547125
3          0.0001          -3.958607       0.0663       -1.178421
4          0.0000          -5.000000       0.0000       -5.000000

   percentimmatureseed  arsinpercentimmatureseed  Final Comments
0             0.000000                  0.000000             NaN
1             1.050205                  0.102660             NaN
2             1.092704                  0.104724             NaN
3             0.150830                  0.038847             NaN
4                  NaN                       NaN             NaN

[5 rows x 21 columns]
```

```
# Part b

# Remove unwanted sheets
del epa_data['Solution Chemistry']
del epa_data['ReadMe']

# Verify keys
epa_data.keys()
```

```
dict_keys(['CALE 2010', 'CALE 2011', 'ELGL 2010', 'ELGL 2011', 'ERLA
2010', 'ERLA 2011', 'FEID 2010', 'FEID 2011', 'FRVI 2010', 'FRVI
2011', 'IRTE 2010', 'IRTE 2011', 'POGR 2010', 'POGR 2011', 'PRVU
2010', 'PRVU 2011', 'RAOC 2010', 'RAOC 2011'])
```

```
# Part c

# Standardize column names
for df in epa_data.keys():
    epa_data[df].columns = epa_data[df].columns.str.lower()
    epa_data[df].columns = epa_data[df].columns.str.strip()

# Part d

combined = pd.concat(epa_data.values(), ignore_index=True)
combined.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2304 entries, 0 to 2303
Data columns (total 71 columns):
 #   Column                                Non-Null Count  Dtype
---  ------                                --------------  -----
 0   year                                  2304 non-null   int64
```

```
1   species                              2304 non-null   object
2   site                                 2304 non-null   object
3   plot                                 2304 non-null   int64
4   trt                                  2304 non-null   int64
5   treatment                            2304 non-null   object
6   pods                                 506 non-null    float64
7   sqrtpods                             633 non-null    float64
8   cover0                               2128 non-null   float64
9   arsincover0                          2011 non-null   float64
10  cover3                               537 non-null    float64
11  arsincover3                          537 non-null    float64
12  matureseeds                          369 non-null    float64
13  logmatureseeddw                      978 non-null    float64
14  immatureseeddw                       739 non-null    float64
15  logimmatureseeddw                    975 non-null    float64
16  totalseeddw                          1467 non-null   float64
17  logtotalseeddw                       1433 non-null   float64
18  percentimmatureseed                  361 non-null    float64
19  arsinpercentimmatureseed             361 non-null    float64
20  final comments                       377 non-null    object
21  matureseedw                          616 non-null    float64
22  logmatureseed+0.00001                128 non-null    float64
23  immatureseedw                        616 non-null    float64
24  logimmatureseed+0.00001              128 non-null    float64
25  totalseeddw+0.00001                  128 non-null    float64
26  biomass                              853 non-null    float64
27  logbiomass                           853 non-null    float64
28  panicles                             475 non-null    float64
29  sqrtpanicles                         475 non-null    float64
30  cover5                               1700 non-null   float64
31  arsincover5                          1700 non-null   float64
32  matureseeddw                         359 non-null    float64
33  percentimmatureseeds                 496 non-null    float64
34  arsinpercentimmatureseeds            496 non-null    float64
35  logmaturelseeddw                     124 non-null    float64
36  logimmaturelseeddw                   124 non-null    float64
37  previous treatment comments          224 non-null    object
38  seedheads                            373 non-null    float64
39  sqrtseedheads                        373 non-null    float64
40  previous year treatment comments     11 non-null     object
41  logmatureseeddwtruncated             112 non-null    float64
42  immartureseedw                       112 non-null    float64
43  logimmartureseedwtruncated           112 non-null    float64
44  revharvestbiomass                    122 non-null    float64
45  log10revharvestbiomass               122 non-null    float64
46  previous treatment                   128 non-null    object
47  arsincover1                          117 non-null    float64
48  unnamed: 11                          2 non-null      object
49  unnamed: 12                          1 non-null      object
```

```
 50  unnamed: 13                           2 non-null      object
 51  unnamed: 14                           1 non-null      object
 52  mature                                126 non-null    float64
 53  nopods                                127 non-null    float64
 54  logmatureseeds                        128 non-null    float64
 55  logimmatureseeds                      128 non-null    float64
 56  logtotalseeds                         128 non-null    float64
 57  extra comments                        0 non-null      float64
 58  previous treatment (2010) comments    64 non-null     object
 59  inflorescences                        116 non-null    float64
 60  sqrtinfloroescence                    116 non-null    float64
 61  totseeddw                             222 non-null    float64
 62  infloroescences                       122 non-null    float64
 63  sqrtinfluorescences                   122 non-null    float64
 64  atypicalinfluorescences               122 non-null    float64
 65  sqrtatypicalinfluorescences           122 non-null    float64
 66  totalinflorescence                    122 non-null    float64
 67  sqrttotalinfluorescences              122 non-null    float64
 68  percentatypical                       107 non-null    float64
 69  arsinpercentatypical                  107 non-null    float64
 70  treatment #                           128 non-null    float64
dtypes: float64(56), int64(3), object(12)
memory usage: 1.2+ MB

# Part e

cols_to_keep = ['year', 'species', 'site', 'plot', 'trt', 'treatment',
                'pods', 'cover0', 'arsincover0', 'cover3',
'arsincover3',
                'cover5', 'arsincover5']

final_data = combined[cols_to_keep]
final_data.to_csv("epa_combined_clean.csv", index=False)
```