midterm

March 10, 2023

```
[64]: import pandas as pd
      import matplotlib.pyplot as plt
      import numpy as np
      import seaborn as sns
```

```
[65]: bitly = pd.read_fwf('bitly.txt', sep = ",")
      bitly
```

```
[65]:      { "a": "Mozilla\/5.0 (Windows NT 6.1; WOW64) AppleWebKit\/535.11 (KHTML,
      like Gecko) Chrome\/17.0.963.78 Safari\/535.11", "c": "US", "nk": 1, "tz":
      "America\/New_York", "gr": "MA", "g": "A6qOVH", "h": "wfLQtf", "l": "orofrog",
      "al": "en-US,en;q=0.8", "hh": "1.usa.gov", "r":
      "http:\/\/www.facebook.com\/l\/7AQEFzjSi\/1.usa.gov\/wfLQtf", "u":
      "http:\/\/www.ncbi.nlm.nih.gov\/pubmed\/22415991", "t": 1331923247, "hc":
      1331822918, "cy": "Danvers", "ll": [ 42.576698, -70.954903 ] }  \
      0      { "a": "GoogleMaps\/RochesterNY", "c": "US", "…
      1      { "a": "Mozilla\/4.0 (compatible; MSIE 8.0; Wi…
      2      { "a": "Mozilla\/5.0 (Macintosh; Intel Mac OS …
      3      { "a": "Mozilla\/5.0 (Windows NT 6.1; WOW64) A…
      4      { "a": "Mozilla\/5.0 (Windows NT 6.1; WOW64) A…
      …                                                   …
      3554   { "a": "Mozilla\/4.0 (compatible; MSIE 9.0; Wi…
      3555   { "a": "Mozilla\/5.0 (Windows NT 5.1) AppleWeb…
      3556   { "a": "GoogleMaps\/RochesterNY", "c": "US", "…
      3557   { "a": "GoogleProducer", "c": "US", "nk": 0, "…
      3558   { "a": "Mozilla\/4.0 (compatible; MSIE 8.0; Wi…

           Unnamed: 1 Unnamed: 2 Unnamed: 3 Unnamed: 4
      0           NaN        NaN        NaN        NaN
      1           NaN        NaN        NaN        NaN
      2           NaN        NaN        NaN        NaN
      3           NaN        NaN        NaN        NaN
      4           NaN        NaN        NaN        NaN
      …         …        …        …        …
      3554        NaN        NaN        NaN        NaN
      3555        NaN        NaN        NaN        NaN
      3556        NaN        NaN        NaN        NaN
```

```
3557         NaN         NaN         NaN         NaN
3558         NaN         NaN         NaN         NaN

[3559 rows x 5 columns]
```

[66]:

```
A = []
with open ('bitly.txt', 'r') as file:
    lines = file.readlines()
    for line in lines :
        # A.append(line.split(','))
        # break
        # print (line)
        # break
        a = line.split(',')
        print (len(a))
        print (a)
        break
# print (A)
```

```
19
['{ "a": "Mozilla\\/5.0 (Windows NT 6.1; WOW64) AppleWebKit\\/535.11 (KHTML', '
like Gecko) Chrome\\/17.0.963.78 Safari\\/535.11"', ' "c": "US"', ' "nk": 1', '
"tz": "America\\/New_York"', ' "gr": "MA"', ' "g": "A6qOVH"', ' "h": "wfLQtf"',
' "l": "orofrog"', ' "al": "en-US', 'en;q=0.8"', ' "hh": "1.usa.gov"', ' "r":
"http:\\/\\/www.facebook.com\\/l\\/7AQEFzjSi\\/1.usa.gov\\/wfLQtf"', ' "u":
"http:\\/\\/www.ncbi.nlm.nih.gov\\/pubmed\\/22415991"', ' "t": 1331923247', '
"hc": 1331822918', ' "cy": "Danvers"', ' "ll": [ 42.576698', ' -70.954903 ]
}\n']
```

0.0.1 Part 2

[67]:

```
data = pd.read_fwf('ex10.22.txt', sep = ' ')
data.head()
```

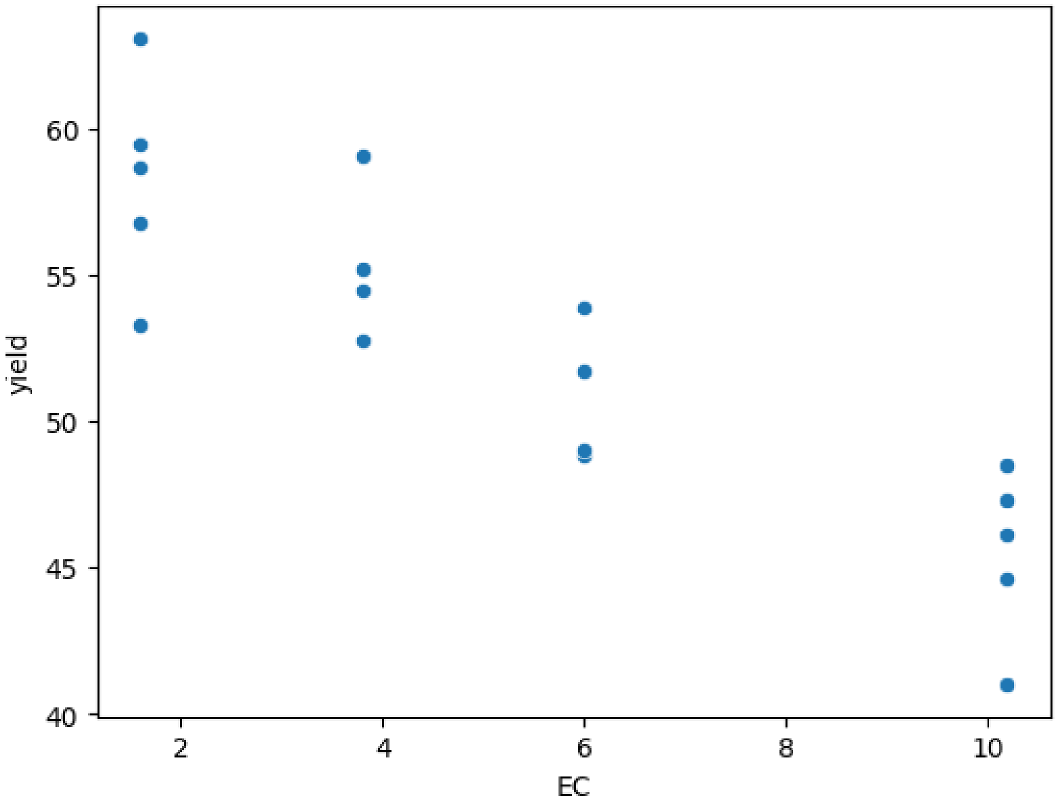
[67]:

```
   yield   EC  ECf
0   59.5  1.6    A
1   53.3  1.6    A
2   56.8  1.6    A
3   63.1  1.6    A
4   58.7  1.6    A
```

[68]:

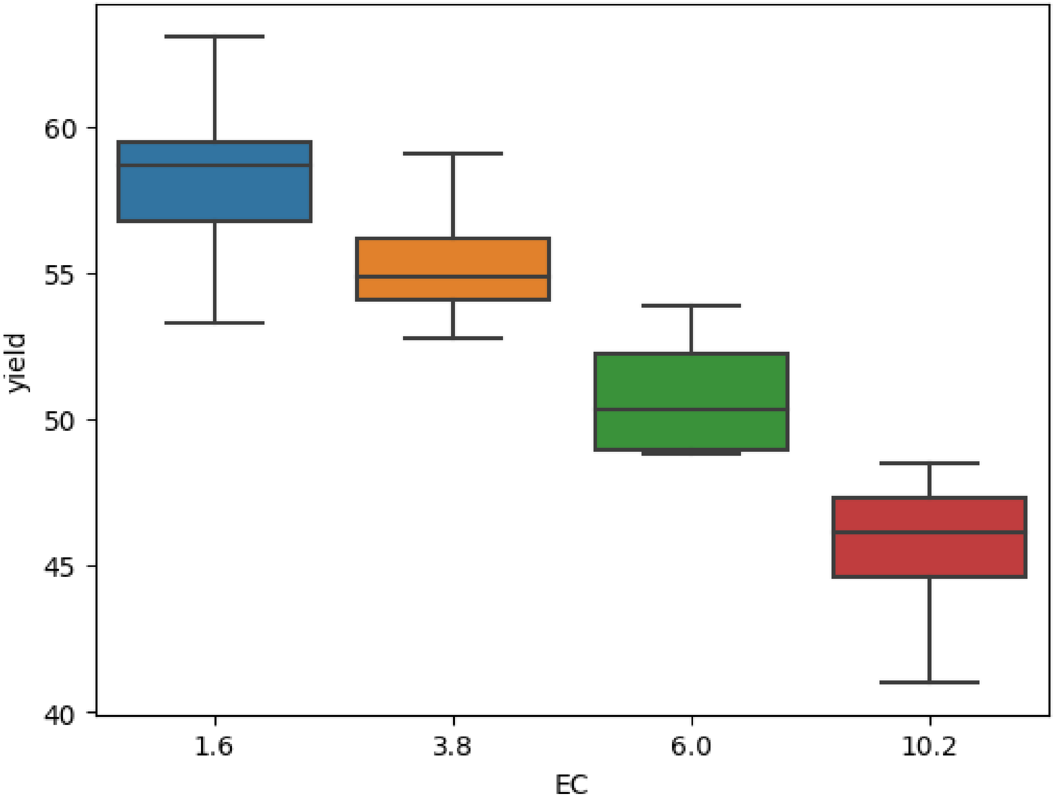
```
'''
Sử dụng biểu đồ scatter để trực quan mối tương quan giữa sản lượng (yeild) và EC
'''
sns.scatterplot(data, x = 'EC', y = 'yield')
```

[68]: `<AxesSubplot: xlabel='EC', ylabel='yield'>`



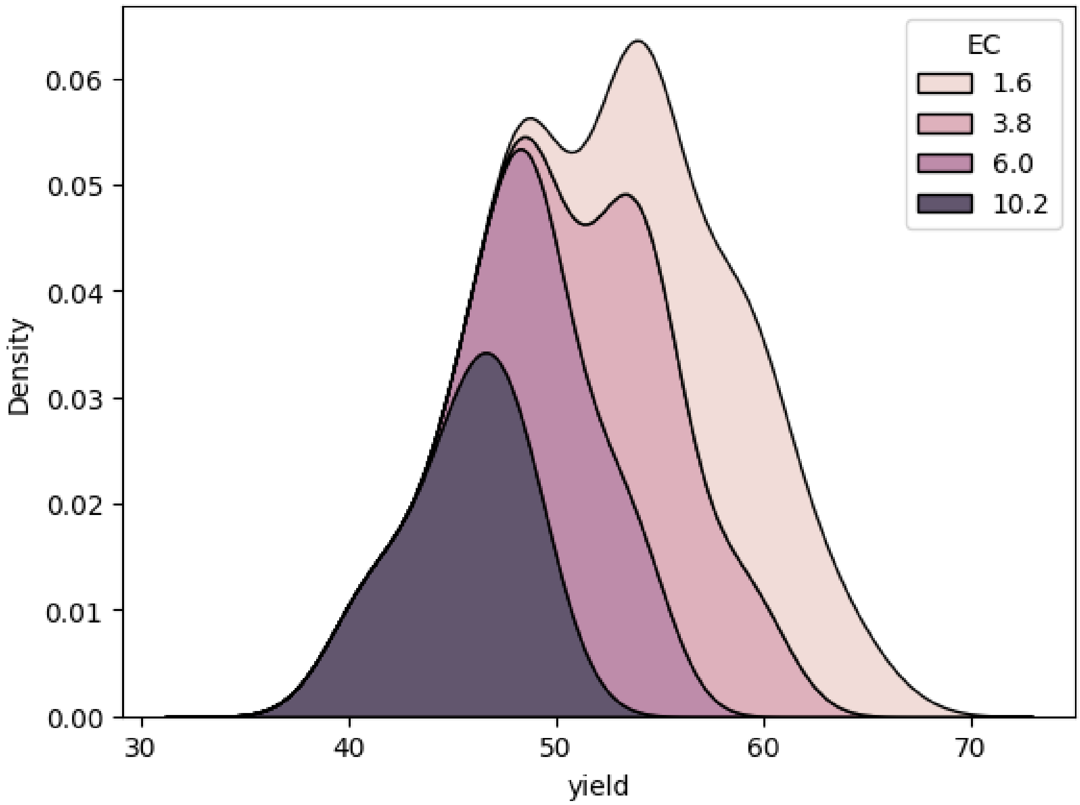
[69]:
```
'''
Tổng hợp thông tin, sử dụng boxplot để trực quan phân phối của sản lượng cà chua
theo từng cấp độ của EC
'''

sns.boxplot(x = data['EC'], y = data['yield'])
```

[69]:
```
<AxesSubplot: xlabel='EC', ylabel='yield'>
```



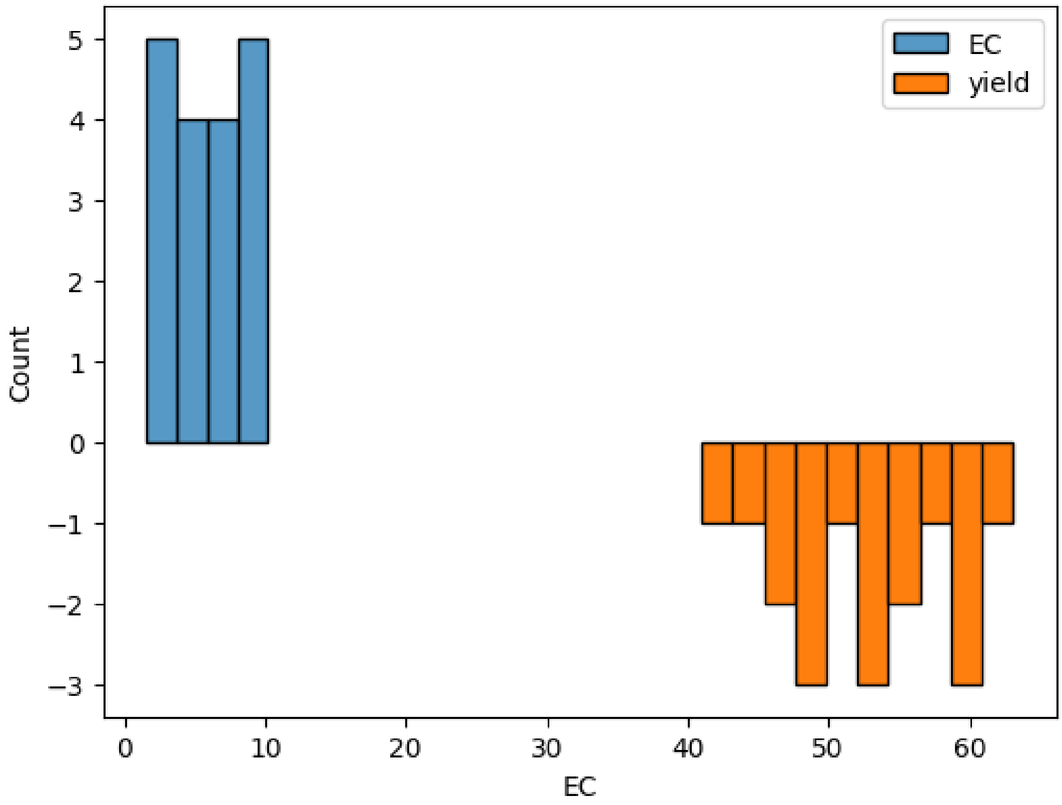
```
[70]: ''' Sử dụng ridgeline để trực quan phân phối của sản lượng cà chua theo từng␣
       ↪cấp độ của
      EC'''
      sns.kdeplot(data, x = data['yield'], hue = data['EC'], multiple='stack')
```

```
[70]: <AxesSubplot: xlabel='yield', ylabel='Density'>
```



```
[110]: sns.histplot(x = data['EC'] , bins=4, label = 'EC')

       height, bins = np.histogram(data['yield'], bins = 10)
       height *= -1
       bins_width = np.diff(bins)[0]
       bins_pos = (bins[:-1] + bins_width / 2)
       plt.bar(bins_pos, height, width=bins_width, label = 'yield', edgecolor =␣
        ↪'black')
       plt.legend()
```

```
[110]: <matplotlib.legend.Legend at 0x7f458e275450>
```



0.0.2 Part 3

```
[72]: tomato = pd.read_csv('tomato-yields.csv')
      tomato
```

```
[72]:          Entity Code  Year  \
      0        Africa  NaN  1961
      1        Africa  NaN  1962
      2        Africa  NaN  1963
      3        Africa  NaN  1964
      4        Africa  NaN  1965
      …           …    …     …
      11277  Zimbabwe  ZWE  2016
      11278  Zimbabwe  ZWE  2017
      11279  Zimbabwe  ZWE  2018
      11280  Zimbabwe  ZWE  2019
      11281  Zimbabwe  ZWE  2020

             Tomatoes | 00000388 || Yield | 005419 || tonnes per hectare
      0                                                   12.320172
      1                                                   12.976988
```

```
2                                                 12.867894
3                                                 13.189582
4                                                 13.492712
…                                                       …
11277                                              7.237900
11278                                              7.219100
11279                                              7.225900
11280                                              7.226900
11281                                              7.224700

[11282 rows x 4 columns]
```

[73]:
```
tomato.isna().sum()
```

[73]:
```
Entity                                                    0
Code                                                   2489
Year                                                      0
Tomatoes | 00000388 || Yield | 005419 || tonnes per hectare       0
dtype: int64
```

[74]:
```
tomato.isna().sum()
tomato = tomato.rename(columns={"Tomatoes | 00000388 || Yield | 005419 ||␣
 ↪tonnes per hectare" : "Yields"})
tomato.head(5)
```

[74]:
```
   Entity Code  Year     Yields
0  Africa  NaN  1961  12.320172
1  Africa  NaN  1962  12.976988
2  Africa  NaN  1963  12.867894
3  Africa  NaN  1964  13.189582
4  Africa  NaN  1965  13.492712
```

[75]:
```
North_America = ['Belize', 'Costa Rica', 'ElSalvador', 'Guatemala', 'Honduras',␣
 ↪'Mexico',
'Nicaragua', 'Panama']
```

[76]:
```
'Đọc và chuyển thể dữ liệu, sử dụng pivot, sao cho mỗi trường thông tin (cột)␣
 ↪là một năm tương ứng'
new_tomato = tomato.copy()
new_tomato = new_tomato.pivot(index = 'Entity',columns='Year', values='Yields')
# new_tomato.isna().sum()
new_tomato.head(10)
```

[76]:
```
Year                      1961       1962       1963       1964       1965  \
Entity
Africa               12.320172  12.976988  12.867894  13.189582  13.492712
Africa (FAO)         12.336499  12.962899  12.888400  13.195300  13.452499
```

```
Albania              12.000000  12.000000  12.400000  12.799999  12.799999
Algeria              16.456999  17.500000  17.500000  13.644899  12.285299
Americas (FAO)       18.990599  21.422100  19.558899  20.495800  22.032900
Angola                2.500000   2.500000   2.500000   2.500000   2.500000
Antigua and Barbuda   2.500000   2.500000   3.333300   3.000000   3.333300
Argentina            15.814899  16.476299  13.880799  17.329800  16.843199
Armenia                    NaN        NaN        NaN        NaN        NaN
Asia                 14.197464  14.080638  14.460519  14.681830  15.145983

Year                      1966       1967       1968       1969       1970  \
Entity
Africa               13.327377  12.466840  12.748196  13.281709  12.926590
Africa (FAO)         13.269099  12.445000  12.705600  13.228399  12.890900
Albania              12.500000  13.214299  12.857100  12.000000  12.333300
Algeria              11.177899   9.095799  10.047800  11.190700   9.449600
Americas (FAO)       21.345299  21.999800  24.566200  22.137699  23.054100
Angola                2.500000   2.500000   3.000000   3.000000   3.076900
Antigua and Barbuda   3.750000   3.571400   3.500000   3.333300   3.437500
Argentina            16.843199  16.410099  15.835000  17.413500  17.969799
Armenia                    NaN        NaN        NaN        NaN        NaN
Asia                 15.810604  15.971325  16.678328  17.489529  17.756638

Year                 …       2011       2012       2013       2014  \
Entity               …
Africa               …  19.040854  16.706995  15.755281  17.457846
Africa (FAO)         …  18.850000  16.706999  15.755300  17.457800
Albania              …  32.786900  31.538500  36.022301  37.184399
Algeria              …  37.502098  36.995800  43.342400  47.055099
Americas (FAO)       …  53.996197  56.599800  56.908497  60.573200
Angola               …   2.981900   2.682900   2.741900   2.682100
Antigua and Barbuda  …   9.565200  10.000000   9.787200   9.729199
Argentina            …  38.545998  38.627998  38.717697  39.327999
Armenia              …  40.291100  42.360100  43.965698  45.882198
Asia                 …  35.808250  35.564072  36.698162  37.637932

Year                      2015       2016       2017       2018       2019  \
Entity
Africa               17.360165  14.999870  14.249650  13.247025  13.902744
Africa (FAO)         17.360199  14.999900  14.249599  13.247000  13.902699
Albania              41.082298  44.014198  44.370499  43.817497  44.975098
Algeria              48.359299  56.772900  53.646698  58.672398  59.124599
Americas (FAO)       59.540798  58.476498  57.541199  62.518597  65.266701
Angola                2.594600   2.625400   2.638200   2.657600   2.678700
Antigua and Barbuda   9.740000   9.260900   8.727300   8.390200   8.081100
Argentina            39.631298  39.697899  39.821400  39.299000  39.317799
Armenia              43.314800  39.074699  37.657200  32.010201  37.012600
Asia                 39.567120  41.075172  42.274757  42.611004  43.077641
```

```
Year                     2020
Entity
Africa              14.087778
Africa (FAO)        14.087800
Albania             45.649399
Algeria             62.164700
Americas (FAO)      67.644997
Angola               2.696100
Antigua and Barbuda  5.833300
Argentina           39.336700
Armenia             38.819901
Asia                43.594265

[10 rows x 60 columns]
```

```
[77]: 'Sử dụng biểu đồ đường để trực quan sản lượng thu hoạch được của các quốc gia␣
       ↪bắc My'
      mask = tomato['Entity'].isin(North_America)
      import plotly.express as px

      fig = px.line(tomato[mask], x = 'Year', y = 'Yields', color = 'Entity')
      fig.show()
```

```
[78]: new_tomato.columns
```

```
[78]: Int64Index([1961, 1962, 1963, 1964, 1965, 1966, 1967, 1968, 1969, 1970, 1971,
                  1972, 1973, 1974, 1975, 1976, 1977, 1978, 1979, 1980, 1981, 1982,
                  1983, 1984, 1985, 1986, 1987, 1988, 1989, 1990, 1991, 1992, 1993,
                  1994, 1995, 1996, 1997, 1998, 1999, 2000, 2001, 2002, 2003, 2004,
                  2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015,
                  2016, 2017, 2018, 2019, 2020],
                 dtype='int64', name='Year')
```

```
[89]: year_2000 = tomato.loc[tomato['Year'] == 2000]
      df = year_2000.sort_values(by = ['Yields'], ascending=False)
      df.head(5)
```

```
[89]:                Entity Code  Year      Yields
      6762      Netherlands  NLD  2000  433.333282
      2641          Denmark  DNK  2000  392.592590
      10555  United Kingdom  GBR  2000  377.000000
      9626           Sweden  SWE  2000  353.061188
      7391           Norway  NOR  2000  328.032288
```

```
[94]: df[:5]
      country = df['Entity'][0:5]
```

```
country
```

[94]:
```
6762        Netherlands
2641            Denmark
10555    United Kingdom
9626             Sweden
7391             Norway
Name: Entity, dtype: object
```

[95]:
```
mask = tomato['Entity'].isin(country)

fig = px.line(tomato[mask], x = 'Year', y = 'Yields', color = 'Entity')
fig.show()
```

[97]:
```
# from osgeo import gdal
import geopandas as gpd

import geoplot as gplt
geoFile = gpd.read_file('data.shx/ne_10m_admin_0_countries.shp')
geoFile.head(10)
```

```
---------------------------------------------------------------------------
CPLE_OpenFailedError                      Traceback (most recent call last)
File fiona/ogrext.pyx:136, in fiona.ogrext.gdal_open_vector()

File fiona/_err.pyx:291, in fiona._err.exc_wrap_pointer()

CPLE_OpenFailedError: Unable to open data.shx/ne_10m_admin_0_countries.shx or␣
  ↪data.shx/ne_10m_admin_0_countries.SHX. Set SHAPE_RESTORE_SHX config option to
  ↪YES to restore or create it.

During handling of the above exception, another exception occurred:

DriverError                               Traceback (most recent call last)
Cell In[97], line 5
      2 import geopandas as gpd
      4 import geoplot as gplt
----> 5 geoFile = gpd.read_file('data.shx/ne_10m_admin_0_countries.shp')
      6 geoFile.head(10)

File ~/.local/lib/python3.10/site-packages/geopandas/io/file.py:259, in␣
  ↪_read_file(filename, bbox, mask, rows, engine, **kwargs)
    256     path_or_bytes = filename
    258 if engine == "fiona":
--> 259     return _read_file_fiona(
    260         path_or_bytes, from_bytes, bbox=bbox, mask=mask, rows=rows,␣
  ↪**kwargs
```

```
    261     )
    262 elif engine == "pyogrio":
    263     return _read_file_pyogrio(
    264         path_or_bytes, bbox=bbox, mask=mask, rows=rows, **kwargs
    265     )

File ~/.local/lib/python3.10/site-packages/geopandas/io/file.py:303, in␣
 ↪_read_file_fiona(path_or_bytes, from_bytes, bbox, mask, rows, where, **kwargs
    300     reader = fiona.open
    302 with fiona_env():
--> 303     with reader(path_or_bytes, **kwargs) as features:
    304         crs = features.crs_wkt
    305         # attempt to get EPSG code

File ~/.local/lib/python3.10/site-packages/fiona/env.py:457, in␣
 ↪ensure_env_with_credentials.<locals>.wrapper(*args, **kwds)
    454     session = DummySession()
    456 with env_ctor(session=session):
--> 457     return f(*args, **kwds)

File ~/.local/lib/python3.10/site-packages/fiona/__init__.py:335, in open(fp,␣
 ↪mode, driver, schema, crs, encoding, layer, vfs, enabled_drivers, crs_wkt,␣
 ↪allow_unsupported_drivers, **kwargs)
    332     path = parse_path(fp)
    334 if mode in ("a", "r"):
--> 335     colxn = Collection(
    336         path,
    337         mode,
    338         driver=driver,
    339         encoding=encoding,
    340         layer=layer,
    341         enabled_drivers=enabled_drivers,
    342         allow_unsupported_drivers=allow_unsupported_drivers,
    343         **kwargs
    344     )
    345 elif mode == "w":
    346     colxn = Collection(
    347         path,
    348         mode,
   (…)
    357         **kwargs
    358     )

File ~/.local/lib/python3.10/site-packages/fiona/collection.py:234, in␣
 ↪Collection.__init__(self, path, mode, driver, schema, crs, encoding, layer,␣
 ↪vsi, archive, enabled_drivers, crs_wkt, ignore_fields, ignore_geometry,␣
 ↪include_fields, wkt_version, allow_unsupported_drivers, **kwargs)
    232 if self.mode == "r":
```

```
    233     self.session = Session()
--> 234     self.session.start(self, **kwargs)
    235 elif self.mode in ("a", "w"):
    236     self.session = WritingSession()

File fiona/ogrext.pyx:587, in fiona.ogrext.Session.start()

File fiona/ogrext.pyx:143, in fiona.ogrext.gdal_open_vector()

DriverError: Unable to open data.shx/ne_10m_admin_0_countries.shx or data.shx/
  ↪ne_10m_admin_0_countries.SHX. Set SHAPE_RESTORE_SHX config option to YES to␣
  ↪restore or create it.
```