FAO_data_Pandas

January 14, 2022

1 Land Use and Land Cover

2 LULC Analysis

2.1 1- Land Use and Land Cover:

The terms land use and land cover are often used interchangeably, but each term has its own unique meaning. Land cover refers to the surface cover on the ground like vegetation, urban infrastructure, water, bare soil etc. Identification of land cover establishes the baseline information for activities like thematic mapping and change detection analysis. Land use refers to the purpose the land serves, for example, recreation, wildlife habitat, or agriculture.

2.1.1 1.1- Importance

- LULC maps play a significant and prime role in planning, management and monitoring programmes at local, regional and national levels.
- LULC maps also help us to study the changes that are happening in our ecosystem and environment.
- Wildlife habitat protection and many more

[ ]:
```
pip install plotly
```

```
Requirement already satisfied: plotly in c:\anaconda\lib\site-packages (5.5.0)
Requirement already satisfied: six in c:\anaconda\lib\site-packages (from
plotly) (1.16.0)
Requirement already satisfied: tenacity>=6.2.0 in c:\anaconda\lib\site-packages
(from plotly) (8.0.1)
Note: you may need to restart the kernel to use updated packages.
```

[ ]:
```
#import libraries
import seaborn as sns
#canvas style
sns.set(style='whitegrid')
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

[ ]:
```
##importing dataset
## Importing LU information of 5 countries
```

```
LUG=pd.read_csv("FAOSTAT_forest_1-13-2022.csv")
LUG.head()
```

[ ]:

```
  Domain Code   Domain  Area Code      Area  Element Code  \
0          GF  Forests        165  Pakistan          5110
1          GF  Forests        165  Pakistan         72332
2          GF  Forests        351     China          5110
3          GF  Forests        351     China         72332
4          GF  Forests        351     China         72332

                                    Element  Item Code        Item  \
0                                      Area       6751  Forestland
1  Net emissions/removals (CO2) (Forest land)       6751  Forestland
2                                      Area       6751  Forestland
3  Net emissions/removals (CO2) (Forest land)       6751  Forestland
4  Net emissions/removals (CO2) (Forest land)       6751  Forestland

   Year Code  Year  Source Code      Source        Unit        Value Flag  \
0       1990  1990         3050  FAO TIER 1     1000 ha    4986.7900    E
1       1990  1990         3050  FAO TIER 1  kilotonnes       0.0000   Fc
2       1990  1990         3050  FAO TIER 1     1000 ha  157140.5900    A
3       1990  1990         3050  FAO TIER 1  kilotonnes -350983.5047    A
4       1991  1991         3050  FAO TIER 1  kilotonnes -350983.5047    A

                                  Flag Description  Note
0  Expert sources from FAO (including other divis…   NaN
1                                  Calculated data   NaN
2  Aggregate, may include official, semi-official…   NaN
3  Aggregate, may include official, semi-official…   NaN
4  Aggregate, may include official, semi-official…   NaN
```

[ ]:

```
##importing dataset
## Importing LU information of 5 countries
LU=pd.read_csv("FAO_ALL_LU.csv")
#LU.head()
```

[ ]:

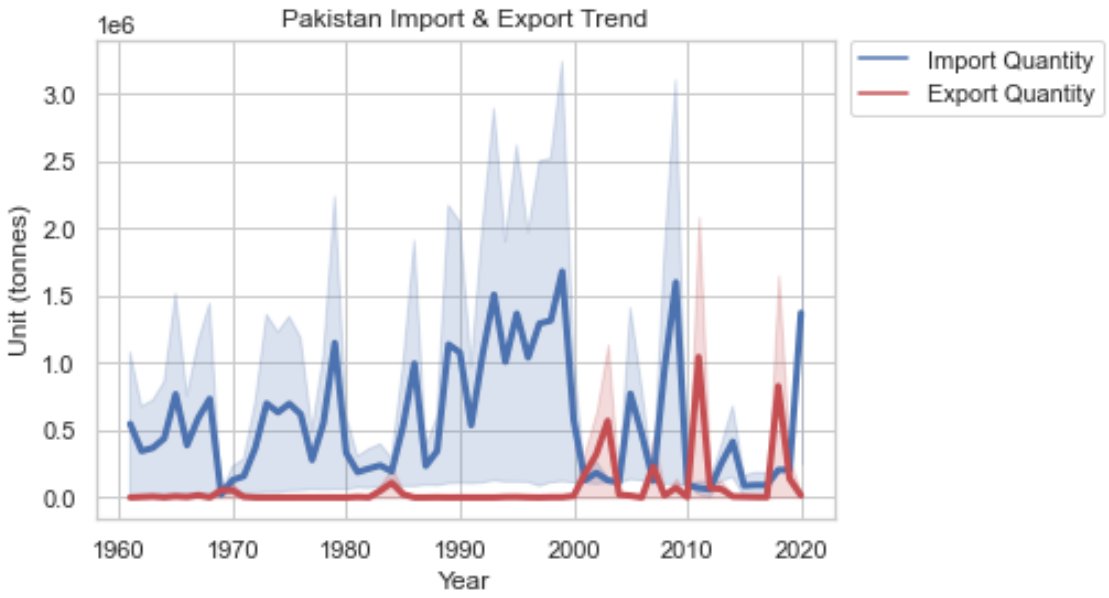
```
IE=pd.read_csv("FAOSTAT_import_1-13-2022.csv")
IE.head()
```

[ ]:

```
  Domain Code                      Domain  Area Code      Area  \
0         TCL  Crops and livestock products        165  Pakistan
1         TCL  Crops and livestock products        165  Pakistan
2         TCL  Crops and livestock products        165  Pakistan
3         TCL  Crops and livestock products        165  Pakistan
4         TCL  Crops and livestock products        165  Pakistan

   Element Code          Element  Item Code   Item  Year Code  Year    Unit  \
```

```
0          5610  Import Quantity          15  Wheat       1961  1961  tonnes
1          5910  Export Quantity          15  Wheat       1961  1961  tonnes
2          5610  Import Quantity         667    Tea       1961  1961  tonnes
3          5910  Export Quantity         667    Tea       1961  1961  tonnes
4          5610  Import Quantity          15  Wheat       1962  1962  tonnes

     Value Flag Flag Description
0  1078899  NaN    Official data
1      432  NaN    Official data
2    16056  NaN    Official data
3        0  NaN    Official data
4   677219  NaN    Official data
```

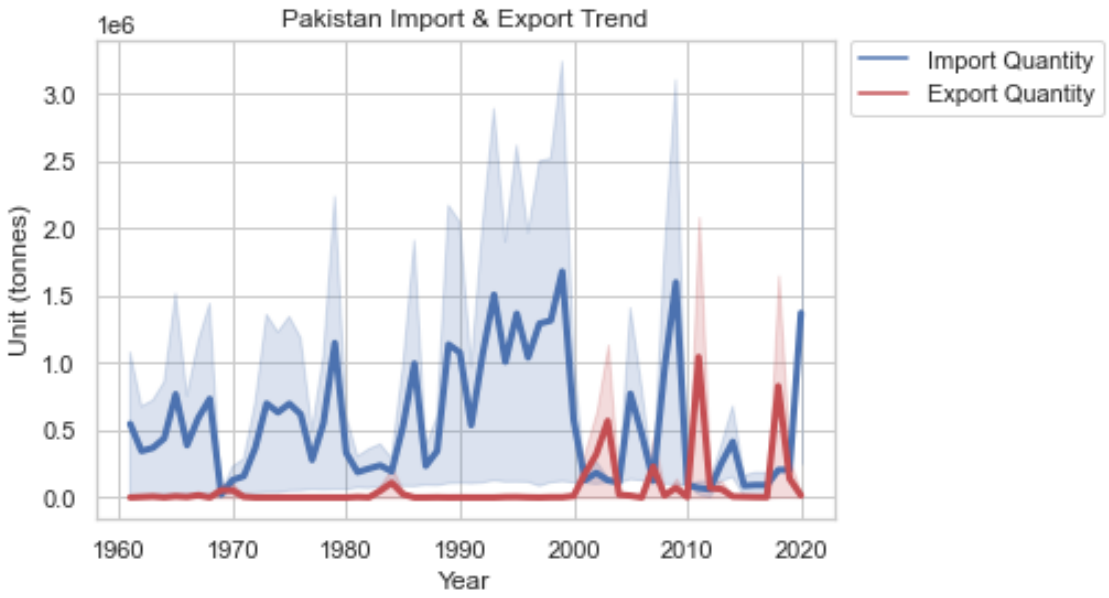
[ ]:
```
sns.lineplot(data=IE, x="Year", y="Value", hue="Element", linewidth=3,
             palette={"Import Quantity": "b", "Export Quantity": "r"}
              ).set(title='Pakistan Import & Export Trend')
plt.xlabel("Year")
plt.ylabel("Unit (tonnes)")
plt.legend(bbox_to_anchor=(1.02, 1), loc='upper left', borderaxespad=0)
plt.show()
```

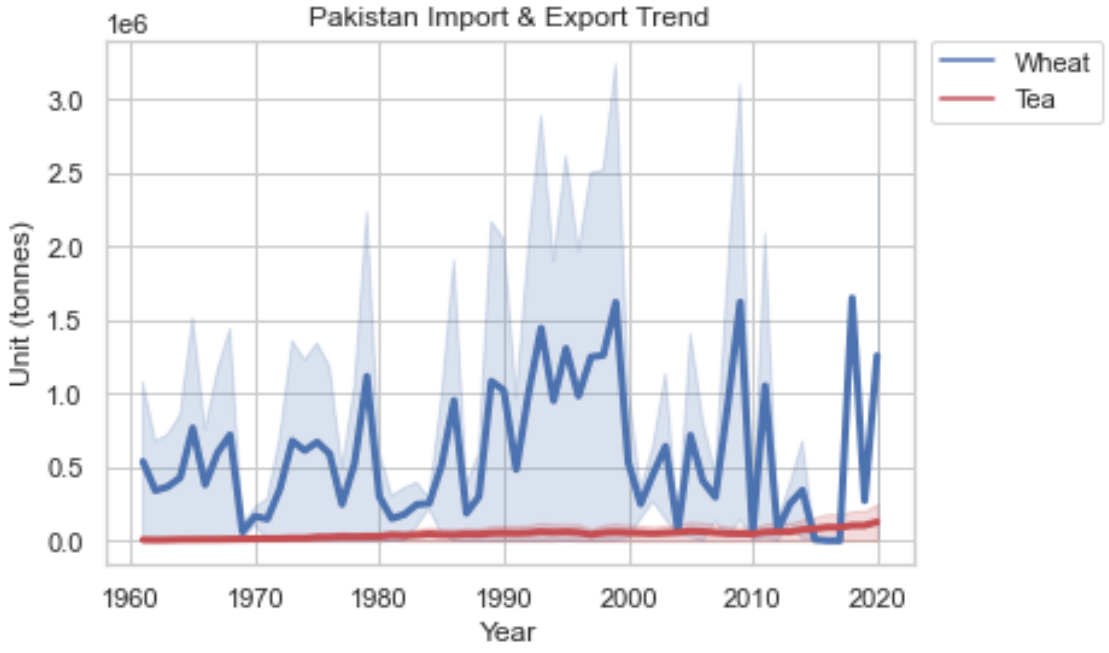


[ ]:
```
sns.lineplot(data=IE, x="Year", y="Value", hue="Item", linewidth=3,
             palette={"Wheat": "b", "Tea": "r"}
              ).set(title='Pakistan Import & Export Trend')
plt.xlabel("Year")
plt.ylabel("Unit (tonnes)")
plt.ylim(0, 4000000)
```

```
plt.legend(bbox_to_anchor=(1.02, 1), loc='upper left', borderaxespad=0)
plt.show()
```

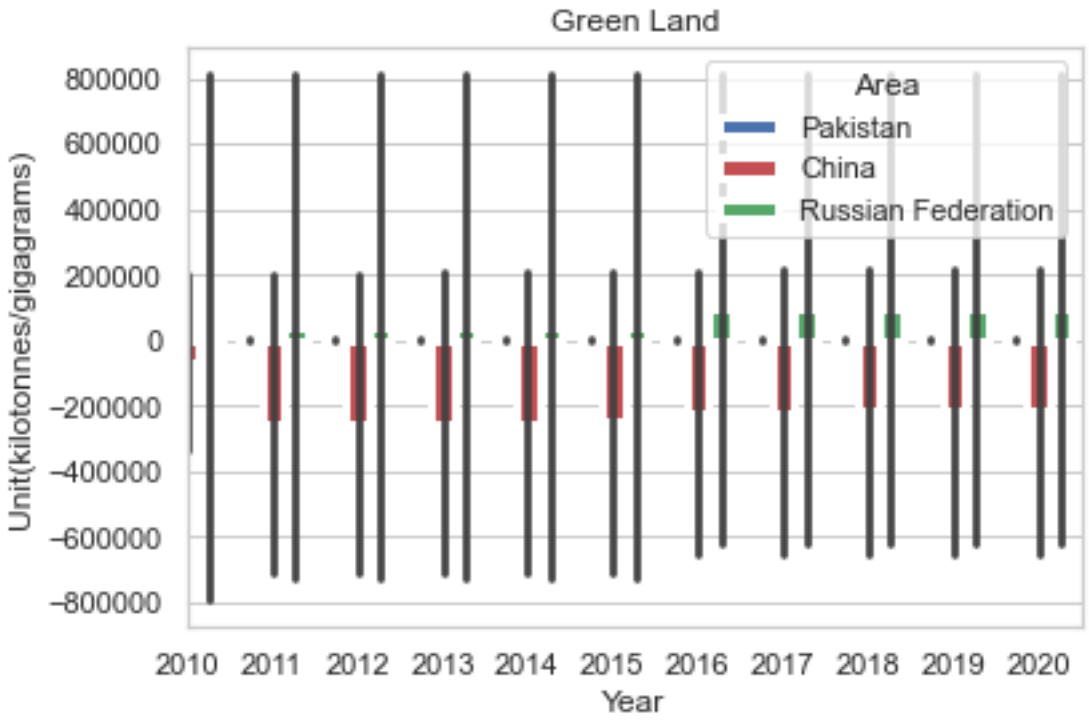
[ ]:
```
sns.lineplot(data=IE, x="Year", y="Value", hue="Element", linewidth=3,
             palette={"Import Quantity": "b", "Export Quantity": "r"}
              ).set(title='Pakistan Import & Export Trend')
plt.xlabel("Year")
plt.ylabel("Unit (tonnes)")
plt.legend(bbox_to_anchor=(1.02, 1), loc='upper left', borderaxespad=0)
fig, axs = plt.subplots(ncols=1)

sns.lineplot(data=IE, x="Year", y="Value", hue="Item", linewidth=3,
             palette={"Wheat": "b", "Tea": "r"}
              ).set(title='Pakistan Import & Export Trend')
plt.xlabel("Year")
plt.ylabel("Unit (tonnes)")
plt.legend(bbox_to_anchor=(1.02, 1), loc='upper left', borderaxespad=0)
plt.show()
```





[ ]:
```
sns.barplot(data=LUG, x="Year", y="Value",saturation=1, hue="Area",
            linewidth=3,
            palette={"Pakistan": "b", "China": "r","Russian Federation":"g"}
             ).set(title='Green Land')
```

```
plt.xlabel("Year")
plt.ylabel("Unit(kilotonnes/gigagrams)")
plt.xlim(20)
plt.show()
```



3 Pandas Library practice

[ ]:
```
##importing dataset
## Importing LU information of 5 countries
LU=pd.read_csv("FAO_LU_comparison.csv")
#LU.head()
```

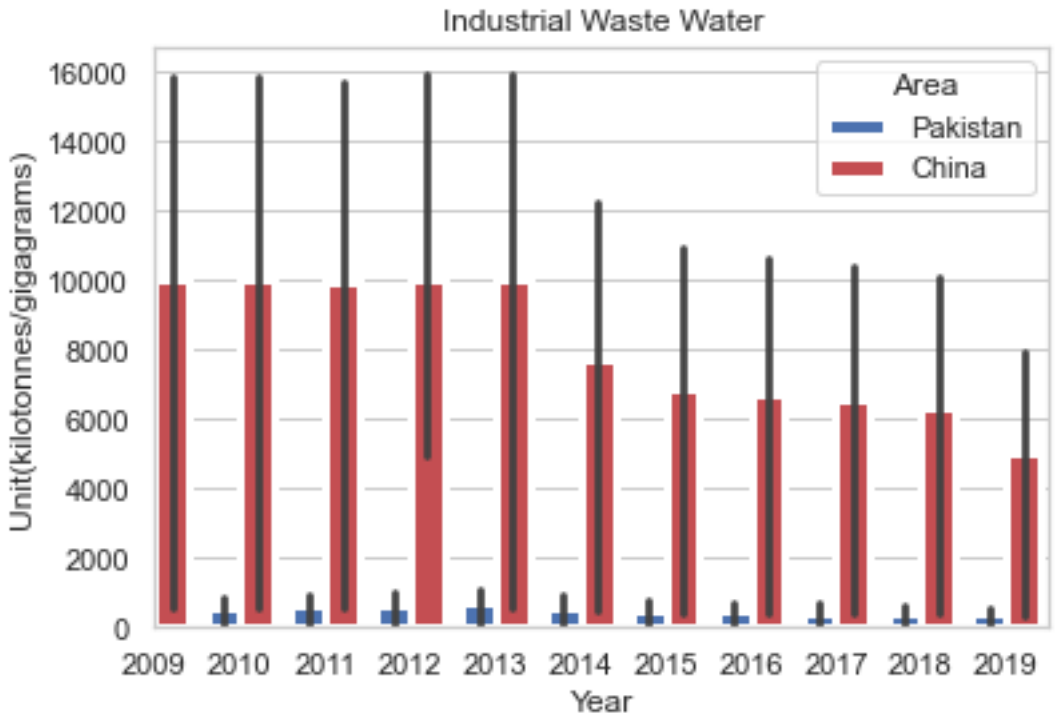
[ ]:
```
LUW=pd.read_csv("FAOSTAT_wastewater_1-13-2022.csv")
LUW.head()
```

[ ]:
```
  Domain Code          Domain  Area Code      Area  Element Code  \
0          GW  Waste Disposal        165  Pakistan          7225
1          GW  Waste Disposal        165  Pakistan        723112
2          GW  Waste Disposal        165  Pakistan        723114
3          GW  Waste Disposal        351     China          7225
4          GW  Waste Disposal        351     China        723112
```

```
                   Element  Item Code                   Item  Year Code  Year  \
0          Emissions (CH4)       6989  Industrial wastewater       1990  1990
1  Emissions (CO2eq) (SAR)       6989  Industrial wastewater       1990  1990
2  Emissions (CO2eq) (AR4)       6989  Industrial wastewater       1990  1990
3          Emissions (CH4)       6989  Industrial wastewater       1990  1990
4  Emissions (CO2eq) (SAR)       6989  Industrial wastewater       1990  1990

         Unit        Value Flag  \
0  kilotonnes    30.170089   Fc
1   gigagrams   771.065186   Fc
2   gigagrams   886.423221   Fc
3  kilotonnes   272.908754    A
4   gigagrams  6687.476069    A

                                    Flag Description
0                                   Calculated data
1                                   Calculated data
2                                   Calculated data
3  Aggregate, may include official, semi-official…
4  Aggregate, may include official, semi-official…
```

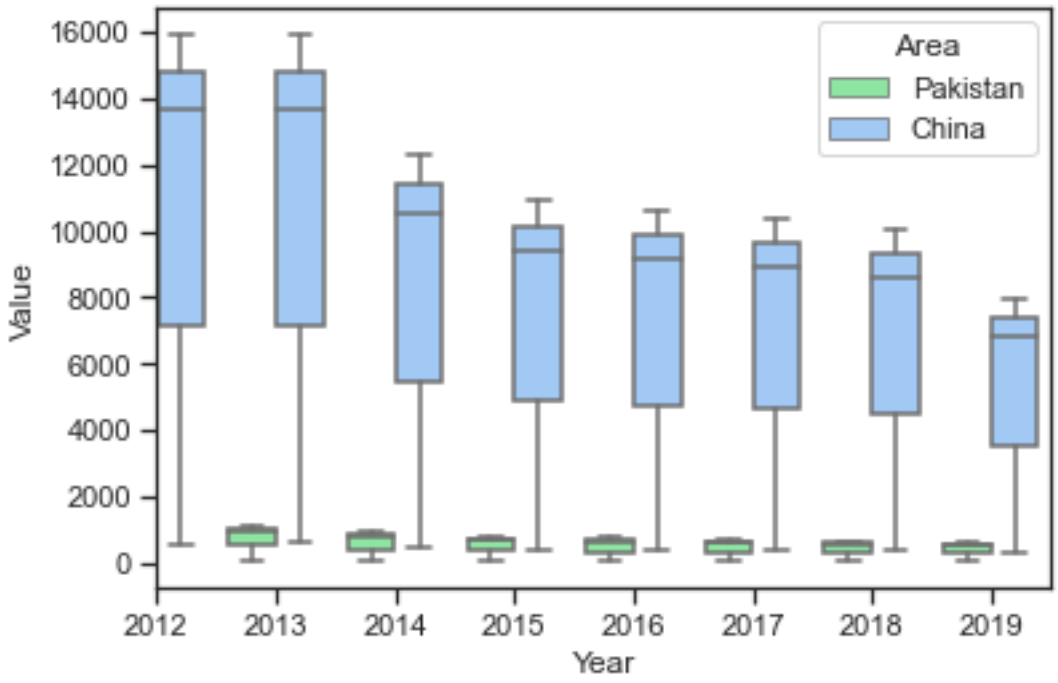
[ ]:
```
import plotly.express as px
#df = px.data.tips()
fig = px.histogram(LUW, x="Year", y="Value", color="Area",
                   title="Industrial Waste_Water")
fig.show()
```

[ ]:
```
sns.barplot(data=LUW, x="Year", y="Value",saturation=1, hue="Area",
            linewidth=3,
            palette={"Pakistan": "b", "China": "r"}
             ).set(title='Industrial Waste Water')

plt.xlabel("Year")
plt.ylabel("Unit(kilotonnes/gigagrams)")
plt.xlim(19)
plt.show()
```



[ ]:
```
import seaborn as sns
sns.set_theme(style="ticks", palette="pastel")

# Load the example tips datase

# Draw a nested boxplot to show bills by day and time
sns.boxplot(x="Year", y="Value",
            hue="Area", palette=["g","b"],saturation=1,
            data=LUW)
#sns.despine(offset=20, trim=True)
plt.xlim(22)
```

[ ]:
```
(22.0, 29.5)
```



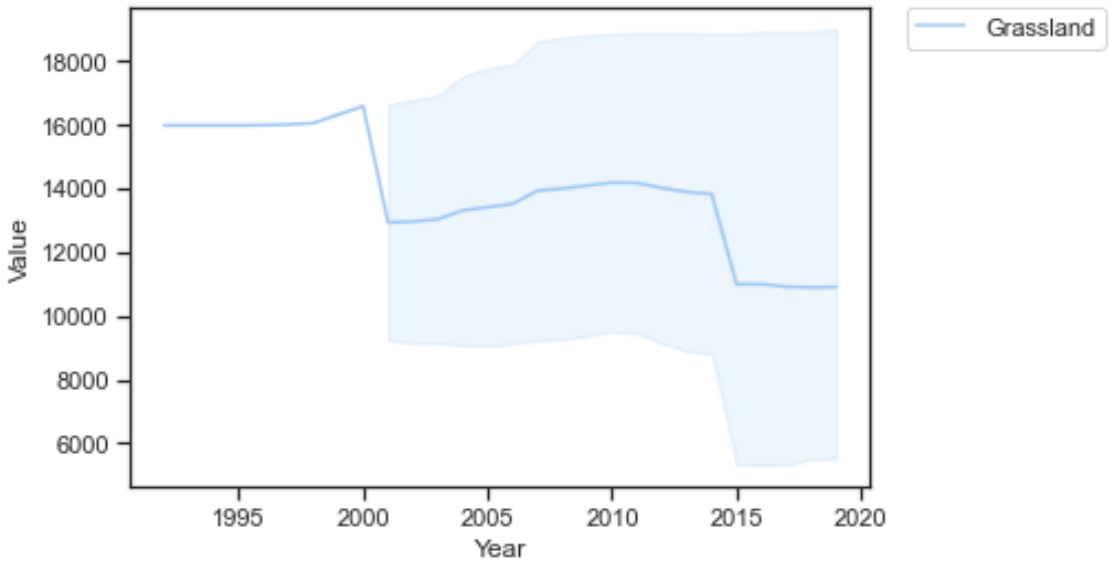
[ ]:
```
##importing dataset
# Pakistan LC data
LC=pd.read_csv("LC.csv")
#LC
```

[ ]:
```
##importing grassland dataset of Pakistan
LG=pd.read_csv("Grassland_Pak.csv")
#LG.head()
```

3.1 Grassland of Pakistan area is getting decrease with the time

3.1.1 Timeseries map of grassland

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

#plot multiple time series
sns.lineplot(x='Year', y='Value', hue='Item', data=LG)
plt.legend(bbox_to_anchor=(1.05, 1), loc='upper left', borderaxespad=0)
plt.show()
```

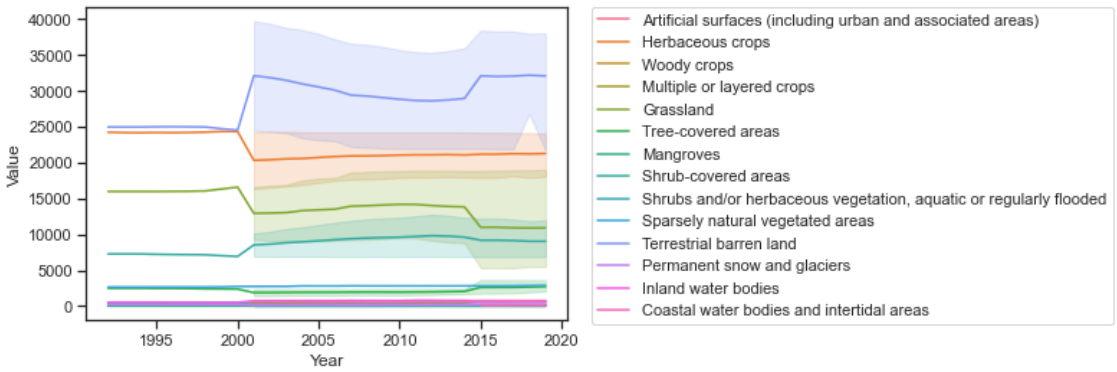


[ ]:
```
##importing grassland dataset
LUA=pd.read_csv("FAO_ALL_LU.csv")
#LUA.head()
```

3.2 Land Use all calsses trend in Pakistan

3.2.1 * Time, types of classes and value analysis

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

#plot multiple time series
sns.lineplot(x='Year', y='Value', hue='Item', data=LUA)
plt.legend(bbox_to_anchor=(1.05, 1), loc='upper left', borderaxespad=0)
plt.show()
```



[ ]:
```
import plotly.express as px #Some selected items
LC=pd.read_csv("LC.csv")
fig = px.histogram(LC, x="Item", y="Value", color="Area",
                   marginal="box", # or violin, rug
                   hover_data=LC.columns)
fig.show()
```

3.3 Compasrion with other countries

3.3.1 each country has specific value for every LU class

3.4 How to run plotly graph in VS code

[ ]:
```
import plotly.io as pio
#pio.renderers.default = "vscode"
pio.renderers.default = "notebook_connected"
```

[ ]:
```
import plotly.express as px
LU=pd.read_csv("FAO_LU_comparison.csv")
fig = px.histogram(LU, x="Item", y="Value", color="Area",
                   marginal="box", # or violin, rug
                   hover_data=LU.columns)
fig.show()
```

3.5 Pie chart analysis for each Land use class

- High value for Terrestrial barren land
- Less for Mangroves, woody crops, inland water bodies etc

[ ]:
```
import plotly.express as px
#df = px.data.gapminder().query("year == 2007").query("Area ==␣
↪'Pakistan','Italy','Chile','Cameroon'")
#df.loc[df['Value'] < 2.e6, 'country'] = 'Other countries' # Represent only␣
↪large countries
fig = px.pie(LUA, values='Value', names='Item', title='Total Land Use of␣
↪Pakistan')
fig.show()
```

3.5.1 Other contries Land use

- Pakistan and Chile have more value

[ ]:
```
import plotly.express as px
#df = px.data.gapminder().query("year == 2007").query("Area ==␣
↪'Pakistan','Italy','Chile','Cameroon'")
```

```
#df.loc[df['Value'] < 2.e6, 'country'] = 'Other countries' # Represent only␣
↪large countries
fig = px.pie(LU, values='Value', names='Area', title='Total Area for Land Use')
fig.show()
```

4 Some pandas practice on FAO data

[ ]:
```
##importing dataset
LU=pd.read_csv("FAO_LU_comparison.csv")
#LU.head()
```

[ ]:
```
mean1 = LU['Value'].mean() #mean function
mean1
```

[ ]: 19435.188111914635

[ ]:
```
LU_PAK=LU[LU["Area"]=="Pakistan"]
#LU_PAK
```

[ ]:
```
mean4 = LU_PAK['Value'].mean() #mean function
mean4
```

[ ]: 34583.710480561575

[ ]:
```
LU_ITA=LU[LU["Area"]=="Italy"]
```

[ ]:
```
mean2 = LU_ITA['Value'].mean() #mean function
mean2
```

[ ]: 14436.351932372472

[ ]:
```
LU_Chile=LU[LU["Area"]=="Chile"]
```

[ ]:
```
mean3 = LU_Chile['Value'].mean() #mean function
mean3
```

[ ]: 26012.186008539386

[ ]:
```
LU_Ca=LU[LU["Area"]=="Cameroon"]
```

[ ]:
```
mean5 = LU_Ca['Value'].mean() #mean function
mean5
```

[ ]: 17587.760381981974

[ ]:
```
import numpy as np
import pandas as pd
```

```
import matplotlib.pyplot as plt
#create values for table
td=[
    ["Country", "Mean"],
    ["Pak", 34583.710480561575],
    ["Italy", 14436.351932372472],
    ["Chile", 26012.186008539386],
    ["Cameroon", 17587.760381981974]
]
td
#create table
#table = ax.table(cellText=table_data, loc='center')
```

[ ]:
```
[['Country', 'Mean'],
 ['Pak', 34583.710480561575],
 ['Italy', 14436.351932372472],
 ['Chile', 26012.186008539386],
 ['Cameroon', 17587.760381981974]]
```

[ ]:
```
type(td)
```

[ ]:
```
list
```

[ ]:
```
x=pd.DataFrame(td)
x
```

[ ]:
```
           0             1
0    Country          Mean
1        Pak  34583.710481
2      Italy  14436.351932
3      Chile  26012.186009
4   Cameroon  17587.760382
```

[ ]:
```
type(x)
```

[ ]:
```
pandas.core.frame.DataFrame
```

[ ]:
```
fig = px.bar(LU, x="Item", y="Year", color="Value", range_y=[1970, 2022],
             pattern_shape="Area", pattern_shape_sequence=[".", "x", "+"])
fig.update_layout(legend=dict(
    yanchor="top",
    y=0.99,
    xanchor="left",
    x=-0.5
))
fig.show()
```

[ ]: