Handout 11

Data visualization with Matplotlib. Streamlit plotting function.

Matplotlib is a *multiplatform* visualization library built on NumPy arrays. More modern packages and tools have been developed on top of it (e.g. Seaborn, ggplot)

```
import matplotlib.pyplot as plt
```

DISPLAYING THE PLOT WITH MATPLOTLIB

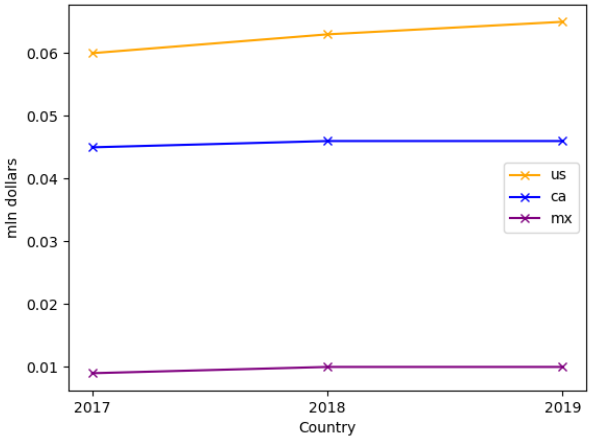
```
plt.figure() – starts a new figure
plt.plot(x-values, y-values, other-plot-params)
plt.show() – displays the figure in a window
```

	country	year	population	gdp
0	us	2017	325.1	19.485
1	us	2018	327.1	20.612
2	us	2019	329.1	21.433
3	ca	2017	36.7	1.650
4	ca	2018	37.1	1.716
5	ca	2019	37.4	1.736
6	mx	2017	124.8	1.159
7	mx	2018	126.2	1.222
8	mx	2019	127.6	1.269

Example: assuming **the required imports** are made

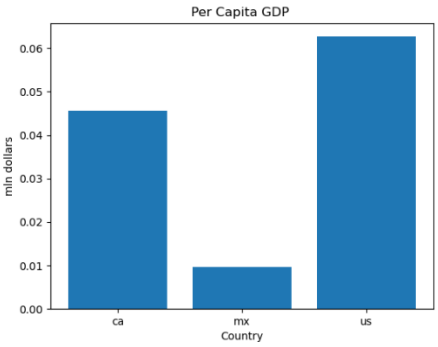
plottingDemo.py

```
def verySimplePLot( df ):
    fig =  plt.figure()  # must get the fig to return
it later
    usOnlydf = df [df.country == 'us']
    caOnlydf = df [df.country == 'ca']
    mxOnlydf = df [df.country == 'mx']

    print("Plotting each country's GDP over years")
    plt.plot(usOnlydf.year, usOnlydf.perCapitaGDP, marker ='x',color = 'orange', label =
"us")
    plt.plot(caOnlydf.year, caOnlydf.perCapitaGDP, marker='x', color='blue', label = "ca")
    plt.plot(mxOnlydf.year, mxOnlydf.perCapitaGDP, marker='x', color='purple', label =
"mx")

    plt.xlabel("Country")
    plt.ylabel("mLn dollars")

    plt.xticks(range(df.year.min(),
df.year.max()+1))
    plt.legend()
    plt.show()
    return fig
```

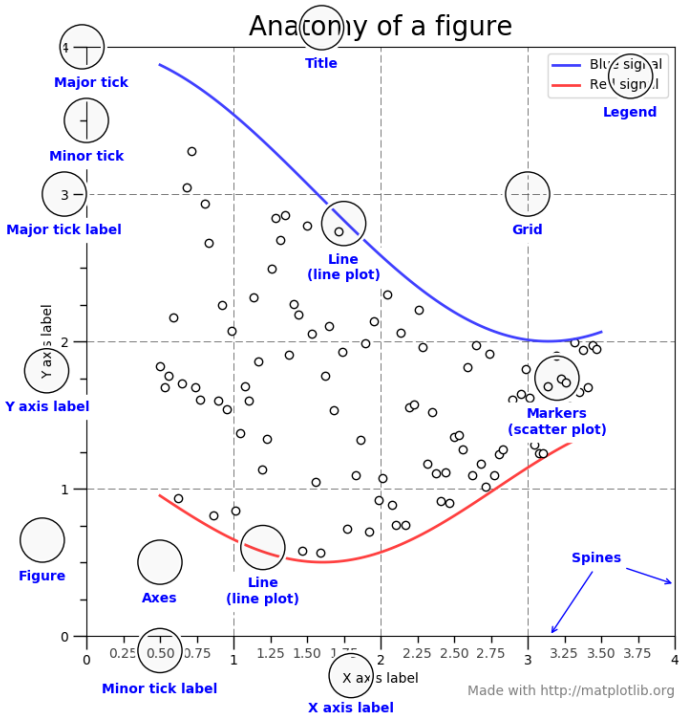


DISPLAYING A BARCHART:

```
#plottingDemo.py
def barChart(df):
    fig =  plt.figure()  # must get the fig to return it
later
    aveByCountry =
df.groupby(by=['country']).mean()['perCapitaGDP']
    print(aveByCountry)
    plt.bar( aveByCountry.index, aveByCountry )
    plt.title("Per Capita GDP ")
    plt.xlabel("Country")
    plt.ylabel("mln dollars")
    plt.show()
    return fig
```



CONTROLLING PARAMETERS OF A PLOT



The following parameters can be specified as a part of the argument list of the plot function:

- `color='blue', color = 'g'`
- `linestyle = 'solid' # or '-', 'dashed', '–' 'dashdot', '-.', 'dotted', ':'`
- the above two may be combined in the single third parameter

```
plt.plot(x, y, '--r')  # -- is line type, r  is red
```

- scatter plots – instead of lines, can display just the points. Marker parameter indicates the point type: o,x,.o+,>

```
plt.plot(x, y, 'ro')
```
or
```
plt.scatter(x, y,  marker='o', color = 'red')
```

- adjust axis limits
  ```
  plt.xlim(-5,10);        plt.ylim(-5,10)
  ```
- include a grid
  ```
  plt.grid(True)
  ```
- set the ticks
  ```
  plt.yticks(range(20, 100, 10) )
  ```
- get the ticks, change tick label display angle
  ```
  locs, labels = plt.xticks()
  plt.setp(labels, rotation=45) # rotate the tick labels
  ```
- adjust the grid parameters
  ```
  plt.grid(color = 'lightgrey', linestyle = '--', linewidth = .5)
  ```
- adjust the style
  plt.style.use('ggplot') # other styles available as

INCLUDING A PLOT IN STREAMLIT

PlottingWithinStreamlit1.py

```
'''plottingDemoWithStreamlit.py
'''
def verySimplePLot( df ):
    fig =  plt.figure()  # must get the fig to return it later
  . . .  code as before
    #plt.show()
    return fig

def main():
    df = pd.read_csv("popgdp.csv")
    df["perCapitaGDP"] = df.gdp / df.population
    df.perCapitaGDP = df.perCapitaGDP.round(3)
    print(df)

    fig1 = verySimplePLot(df)

    #Display the figure within Streamlit
    st.write ("Country per-capita GDP over time")
    st.pyplot(fig1)

main()
```

COMBINE PLOTTING WITH USER INPUT

Include a Streamlit **multiselect** user input control for selecting a country; in **makePlot**, add a parameter specifying which countries to include.

```
'''PlottingWithinStreamlit3.py
 Plotting using input from Streamlit widget
 '''
import pandas as pd
```

```
import matplotlib.pyplot as plt
import streamlit as st

def makePlot (df, countries):
    fig = plt.figure()

    for oneCountry  in countries:
            countryData = df [df.country == oneCountry]
            plt.plot(countryData.year, countryData.perCapitaGDP,
                     marker = 'x', label = oneCountry)

    plt.xticks(range(df.year.min(), df.year.max()+1))
    plt.legend()
    plt.title("Per Capita GDP over years")
    plt.xlabel("Year")
    plt.ylabel("mln dollars")
    #plt.show()
    return fig

def main():
    df = pd.read_csv("popgdp.csv")

    #change country name to uppercase
    df.country = df.country.str.upper()

    # add a new column
    df["perCapitaGDP"] = df.gdp / df.population
    df.perCapitaGDP = df.perCapitaGDP.round(3)

    print(df)

    st.write("Plotting per capita GDP")

    #grab countries listed in the df
    countriesLst =  sorted( set(df.country))

    #enable user selection of countries in Streamlit multiselect
    countrySelection = st.multiselect("Select an option: ",  countriesLst)
    print(countrySelection)

    #call plotting function to produce plot for the specified countries
    fig1 = makePlot (df, countrySelection )
    # line chart
    st.pyplot(fig1)

main()
```

PRACTICE

- Create a plot and a bar-chart for data in **`djia.csv, djia-prices.csv`**
- Display the created plots in a web page using Streamlit