

Onions Yield with Time

Pakistan - India - Bangladesh

1. Importing Libraries and Datafile

```
In [ ]: # Importing Libraries

import seaborn as sns
import pandas as pd
import matplotlib.pyplot as plt
import plotly.express as px

# Loading csv file
df = pd.read_csv('Onion_Yield.csv')           #< Write the name of the csv file to import as filename.csv
```

2. Analyzing Dataset

```
In [ ]: #Pulling the header of the dataset
df.head(10)
```

	Domain Code	Domain	Area Code (FAO)	Country	Element Code	Element	Item Code (FAO)	Item	Year Code	Year	Unit	Value	Flag	Flag Description
0	QCL	Crops and livestock products	16	Bangladesh	5419	Yield	403	Onions, dry	1961	1961	hg/ha	61211	Fc	Calculated data
1	QCL	Crops and livestock products	16	Bangladesh	5419	Yield	403	Onions, dry	1962	1962	hg/ha	55574	Fc	Calculated data
2	QCL	Crops and livestock products	16	Bangladesh	5419	Yield	403	Onions, dry	1963	1963	hg/ha	55020	Fc	Calculated data
3	QCL	Crops and livestock products	16	Bangladesh	5419	Yield	403	Onions, dry	1964	1964	hg/ha	51804	Fc	Calculated data
4	QCL	Crops and livestock products	16	Bangladesh	5419	Yield	403	Onions, dry	1965	1965	hg/ha	52800	Fc	Calculated data
5	QCL	Crops and livestock products	16	Bangladesh	5419	Yield	403	Onions, dry	1966	1966	hg/ha	53145	Fc	Calculated data
6	QCL	Crops and livestock products	16	Bangladesh	5419	Yield	403	Onions, dry	1967	1967	hg/ha	57587	Fc	Calculated data
7	QCL	Crops and livestock products	16	Bangladesh	5419	Yield	403	Onions, dry	1968	1968	hg/ha	57587	Fc	Calculated data
8	QCL	Crops and livestock products	16	Bangladesh	5419	Yield	403	Onions, dry	1969	1969	hg/ha	55994	Fc	Calculated data
9	QCL	Crops and livestock products	16	Bangladesh	5419	Yield	403	Onions, dry	1970	1970	hg/ha	55457	Fc	Calculated data

```
In [ ]: # Quick Stats of the Dataset
df.describe()
```

	Area Code (FAO)	Element Code	Item Code (FAO)	Year Code	Year	Value
count	180.000000	180.0	180.0	180.000000	180.000000	180.000000
mean	93.666667	5419.0	403.0	1990.500000	1990.500000	98622.583333
std	61.163760	0.0	0.0	17.366409	17.366409	37180.036189
min	16.000000	5419.0	403.0	1961.000000	1961.000000	31592.000000
25%	16.000000	5419.0	403.0	1975.750000	1975.750000	66570.250000
50%	100.000000	5419.0	403.0	1990.500000	1990.500000	104329.000000
75%	165.000000	5419.0	403.0	2005.250000	2005.250000	126332.500000
max	165.000000	5419.0	403.0	2020.000000	2020.000000	187041.000000

3. Defining Components of the Graph

```
In [ ]: # Defining the main data axis and components of the graph
xAx = 'Year'           #< Select the data column for your x-axis
yAx = 'Value'          #< Select the data column for your y-axis
sns hue = 'Country'    #< Select the data column for your hue option

# Defining Graph Title, axis lables
LTitle = 'Onion Yield wrt Countries (Calculated)'
xLable = 'Year'        #< Write your Chart Title
yLable = 'Yield (hg/ha)' #< Write your x-axis lable
LegTitle = 'Countries' #< Write your y-axis lable
                        #< Write your Legend Title
```

4. Plotting Varioius Graphs as Options

```
In [ ]: #Line Plot of the Data using plotly
fig = px.line(df, x = xAx, y = yAx, color =sns hue, title= LTitle)

# Defining Titles and lables
fig.update_layout(
    title=LTitle,
    xaxis_title=xLable,
    yaxis_title=yLable,
    legend_title=LegTitle,
)
#Slider
fig.update_xaxes(rangeslider_visible=True)

fig.show()
```

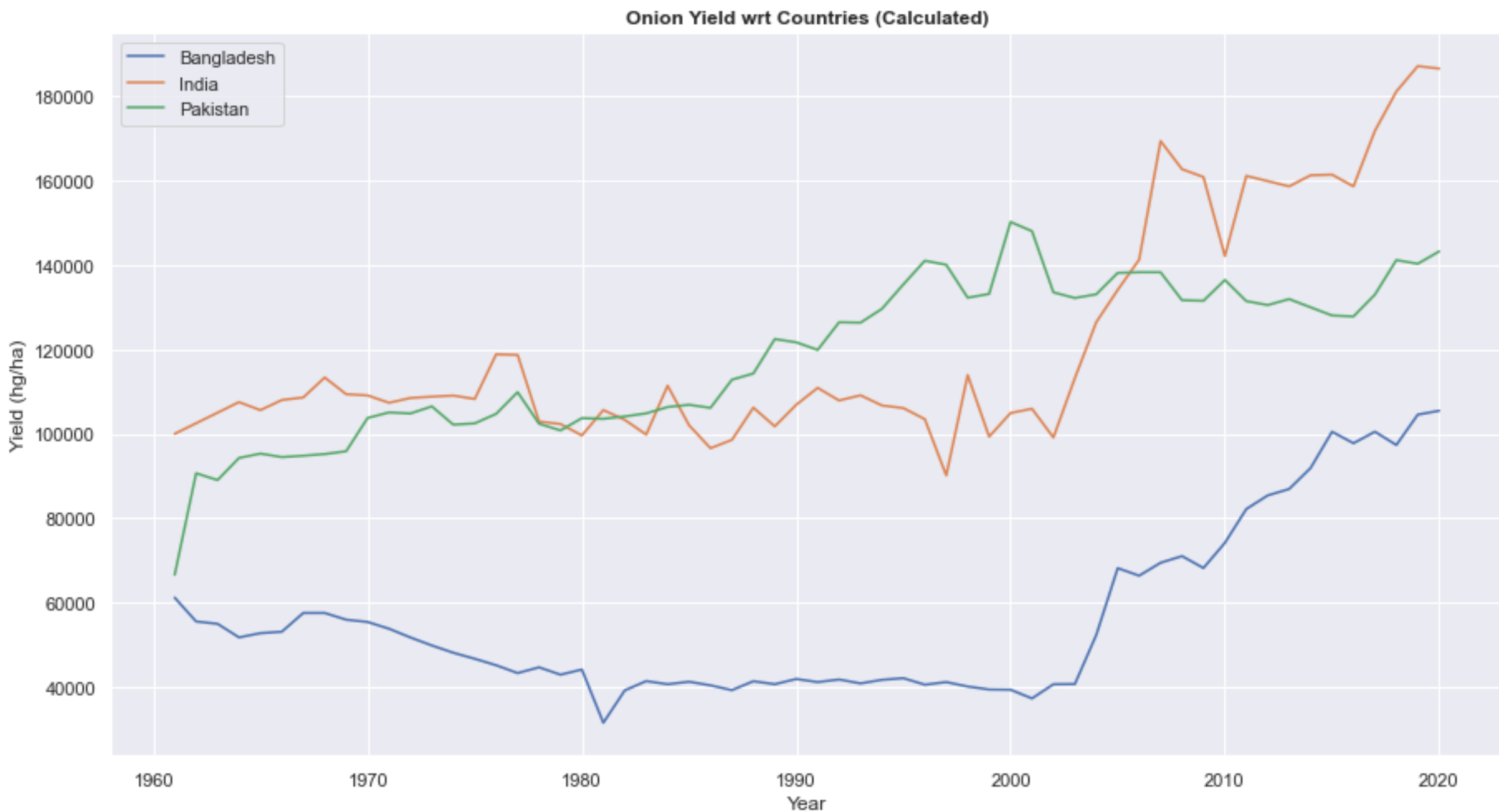
```
In [ ]: #Draww a line plot using Seaborn

#Defining Graph size
sns.set(rc = {'figure.figsize':(15,8)})

#Plot the graph
sns.lineplot(x=xAx, y=yAx, hue=sns hue, data=df, ci=95)

# Defining Titles and lables
plt.title(LTitle, size="12", weight="bold")
plt.xlabel(xLable)
plt.ylabel(yLable)
plt.legend()

plt.show()
```



```
In [ ]: #Draww a lm plot using Seaborn

# Setting Theme
sns.set_theme(style="darkgrid")

# Make a custom palette with Country colors
pal = dict(Pakistan="green", Bangladesh="blue", India="orange")

# time plot based on Value(Area) and Country
g = sns.lmplot(x=xAx, y=yAx, hue=sns hue, data=df, palette=pal,y_jitter=10, logistic=False, truncate=False, height=10)

# Defining Titles and lables
plt.title(LTitle, size="12", weight="bold")
plt.xlabel(xLable)
plt.ylabel(yLable)
plt.legend()
plt.ylim()

plt.show()
```

