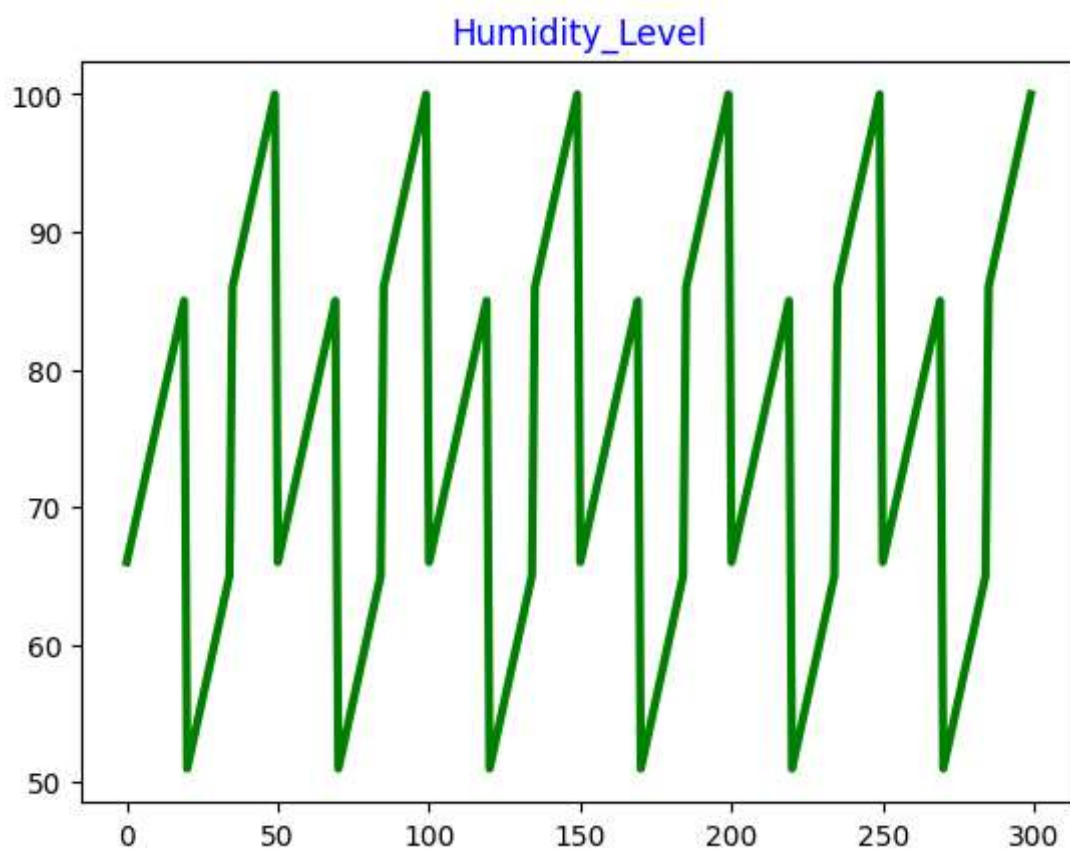


In [30]: 1 *#Graphs*

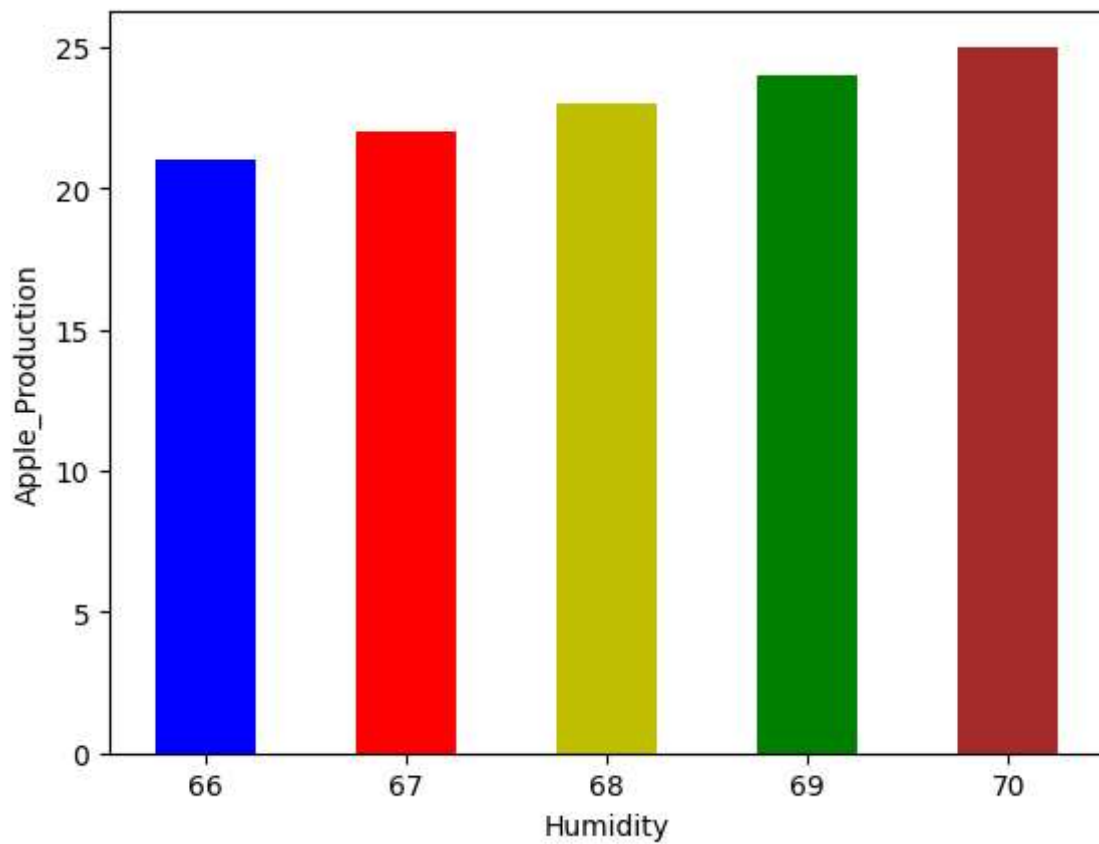
```
In [2]: 1 import pandas as pd
2 apple=pd.read_csv("apple.csv")
3 x=apple.drop(["Apple", "Temperature", "Rainfall"],axis=1)
4 y=apple['Apple']
```

```
In [7]: 1 #plot
2 import matplotlib.pyplot as plt
3 plt.plot(x,color='green',linewidth=3,markerfacecolor='palegreen',markersiz
4 plt.title("Humidity_Level",color="b")
5 plt.show()
```



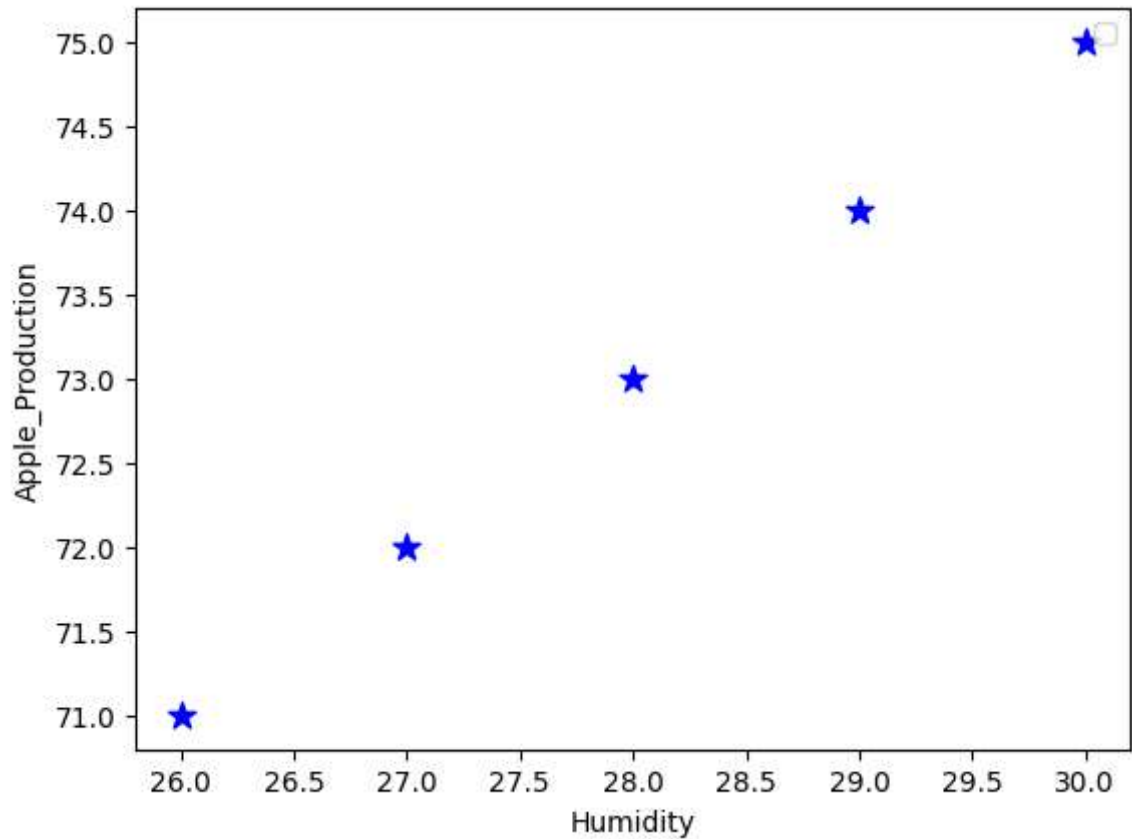
```
In [13]: 1 #bar chart
2 v=[21,22,23,24,25]
3 n=[66,67,68,69,70]
4 color=['b','r','y','g','brown']
5 plt.bar(n,v,color=color,width=0.5)
6 plt.xlabel("Humidity")
7 plt.ylabel("Apple_Production")
8
9 plt.show
```

```
Out[13]: <function matplotlib.pyplot.show(close=None, block=None)>
```

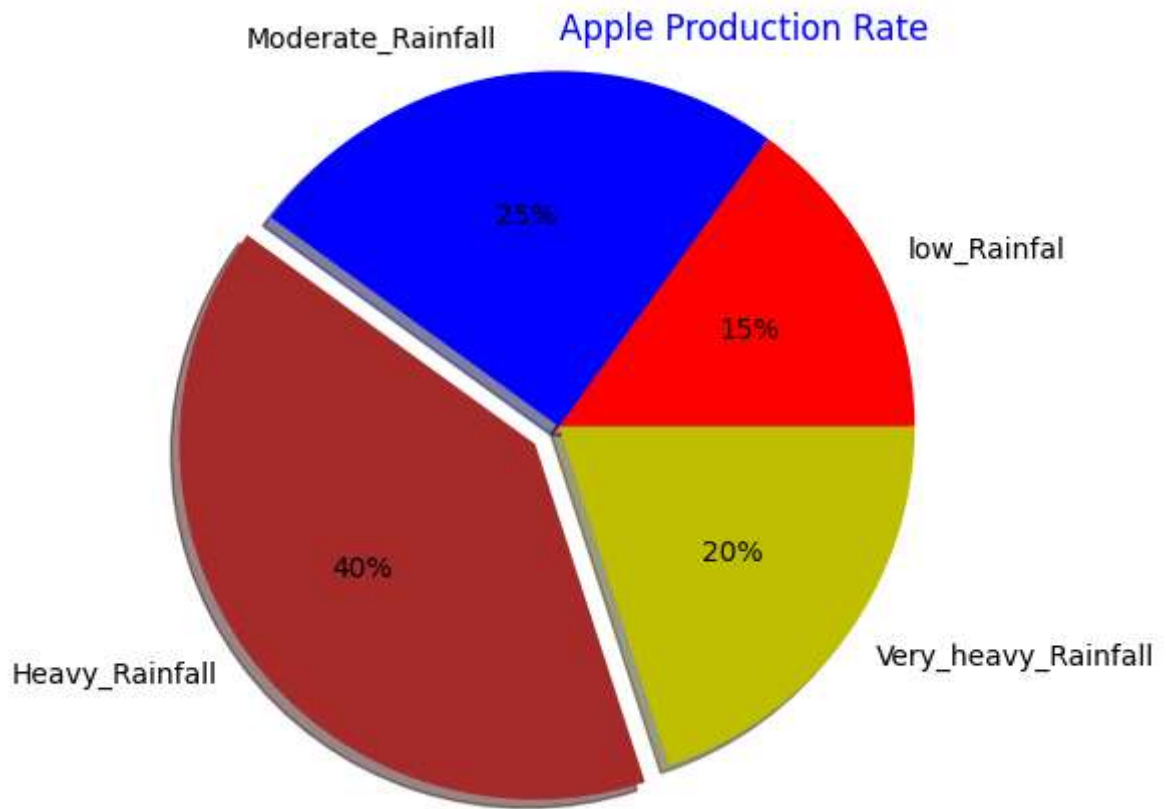


```
In [15]: 1 #scatter plot
2 x=[26,27,28,29,30]
3 y=[71,72,73,74,75]
4 plt.xlabel("Humidity")
5 plt.ylabel("Apple_Production")
6 plt.scatter(x,y,s=100,color="blue",marker="*") #s=size
7 plt.legend()
8 plt.show()
```

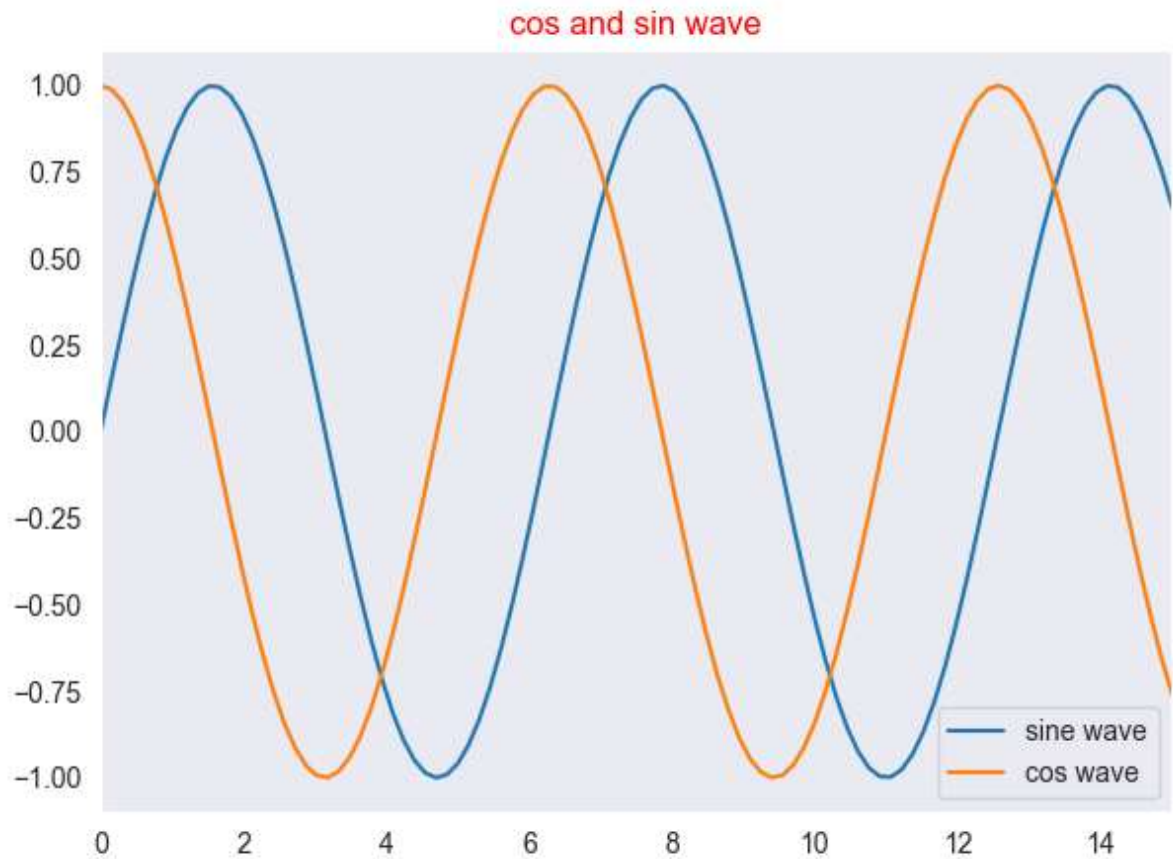
No artists with labels found to put in legend. Note that artists whose label start with an underscore are ignored when legend() is called with no argument.



```
In [21]: 1 #pie chart
2 x=['low_Rainfal','Moderate_Rainfall','Heavy_Rainfall','Very_heavy_Rainfall']
3
4 slices=[3,5,8,4]
5
6 colors=['r','b','brown','y']
7 plt.pie(slices,labels=x,colors=colors,shadow=True,explode=(0,0,0.1,0),radi
8 plt.title("Apple Production Rate",color="b",loc="right")
9 plt.show()
```



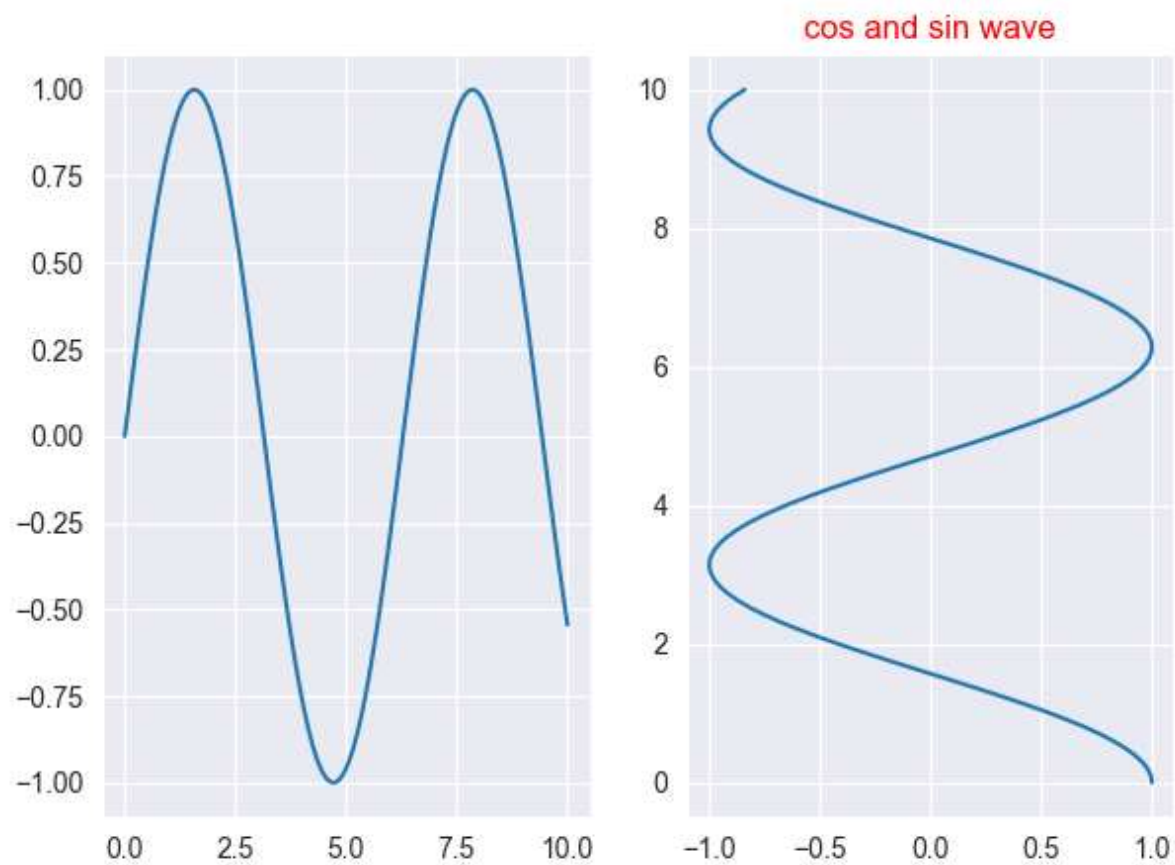
```
In [27]: 1 #sin and cos wave on same plot
2 import numpy as np
3 import seaborn as sns
4 import matplotlib.pyplot as plt
5 #creating a style
6 sns.set_style("dark")
7 #creating a subplots
8 fig, ax=plt.subplots(figsize=(7,5))
9 #range of cos and sin
10 x=np.linspace(0,15,100)
11 #creating a sin
12 y=np.sin(x)
13 ax.plot(x,y,label="sine wave")
14 #creating a cos
15 y1=np.cos(x)
16 ax.plot(x,y1,label="cos wave")
17 ax.set_xlim(0,15)
18 ax.set_ylim(-1,1)
19 plt.title('cos and sin wave',color='r')
20 #creating a legend
21 ax.legend()
22 plt.show()
23
```



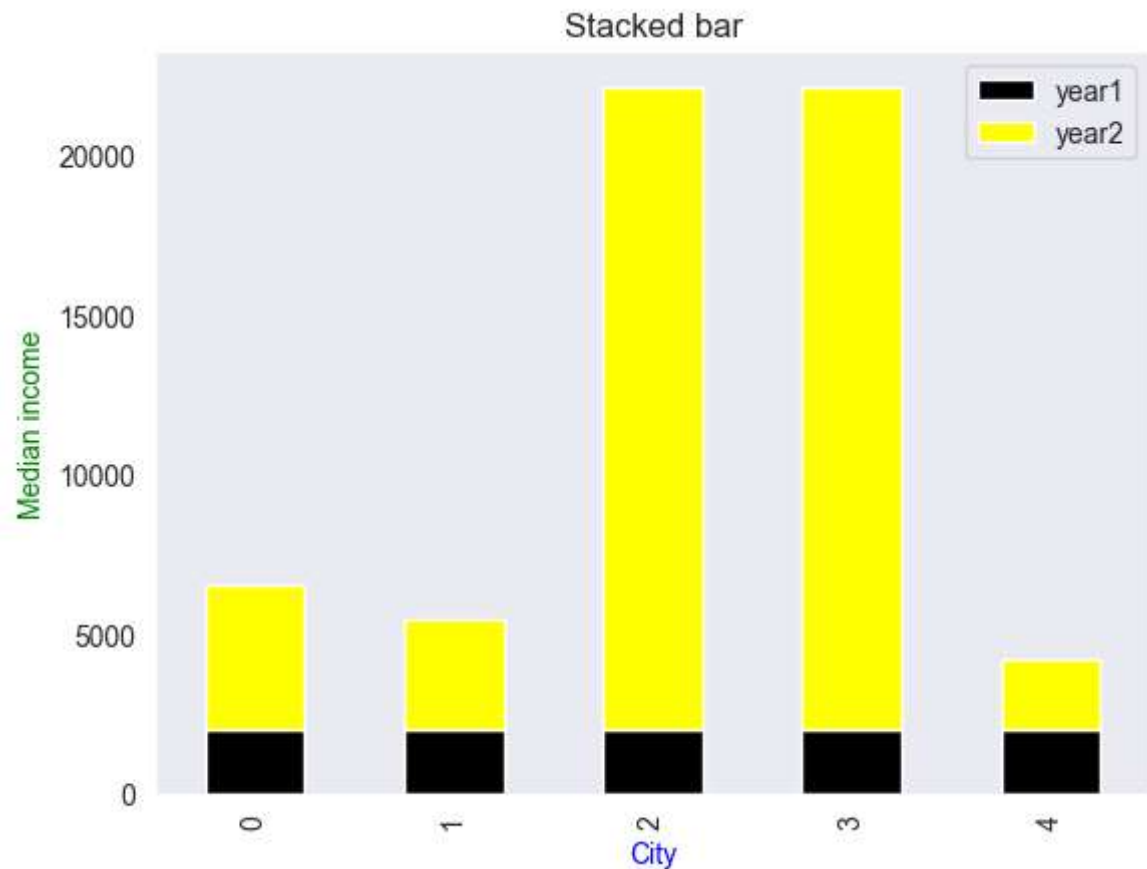
```
In [28]: 1 #sin and cos wave on different plots
2 import numpy as np
3 import seaborn as sns
4 from matplotlib import pyplot as plt
5 #creating a style
6 sns.set_style('darkgrid')
7 #creating a subplots
8 fig, ax=plt.subplots(figsize=(7, 5))
9 #range of cos and sin
10 x=np.linspace(0,10,100)
11 #creating a sin
12 y=np.sin(x)
13 ax.plot(x,y,label="sine wave",color="black")
14 #creating a cos
15 y1=np.cos(x)
16 ax.plot(x,y1,label="cos wave")
17 plt.subplot(1,2,1)
18 plt.plot(x,y)
19 plt.subplot(1,2,2)
20 plt.plot(y1,x)
21 ax.set_xlim(0,20)
22 ax.set_ylim(0,20)
23 plt.title('cos and sin wave',color='r')
24 #creating a legend
25 ax.legend()
26 plt.show()
```

C:\Users\Nikhil\AppData\Local\Temp\ipykernel_3484\3648385.py:17: MatplotlibDeprecationWarning: Auto-removal of overlapping axes is deprecated since 3.6 and will be removed two minor releases later; explicitly call ax.remove() as needed.

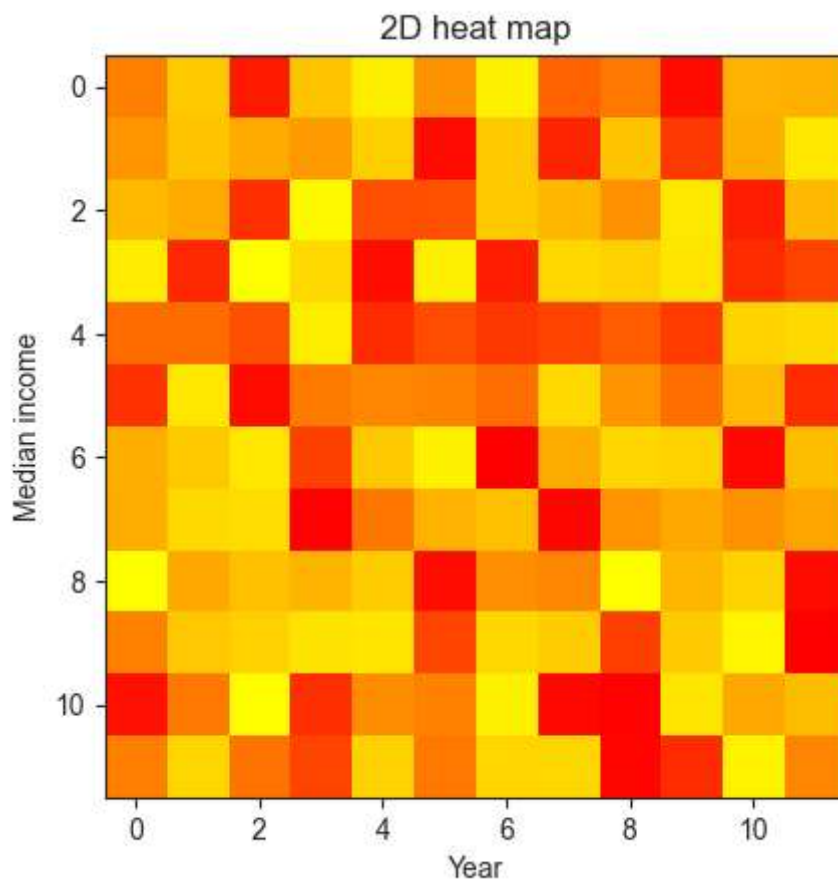
```
plt.subplot(1,2,1)
```



```
In [31]: 1 #stacked bar
2 import pandas as pd
3 import seaborn as sns
4 import matplotlib.pyplot as plt
5 #creating
6 sns.set_style('dark')
7 y1=pd.Series([2005,2012,2014,2006,2018])
8 y2=pd.Series([4506,3441,20103,20133,2206])
9 data=pd.DataFrame({"year1":y1,"year2":y2})
10 index=['Karnataka','Delhi','Hyderabad','TamilNadu','Maharashtra']#index is
11 data.plot(kind='bar',stacked=1,color=['black','yellow'])
12 plt.title('Stacked bar')
13 plt.xlabel("City",color="b")
14 plt.ylabel("Median income",color="g")
15 plt.show()
16
```

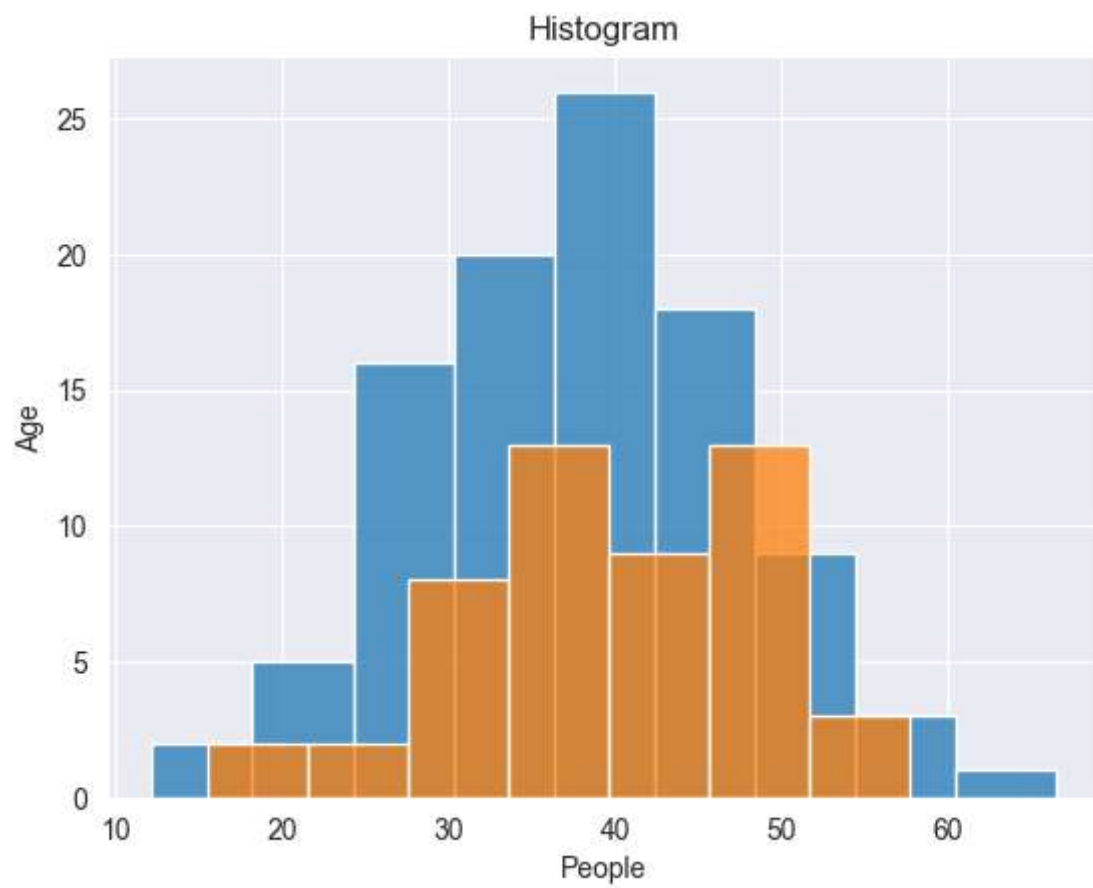



```
In [33]: 1 #heat map
2 import pandas as pd
3 import seaborn as sns
4 import matplotlib.pyplot as plt
5 import numpy as np
6 #creating heat map
7 data=np.random.random((12,12))
8 plt.imshow(data,cmap='autumn',interpolation='nearest')
9 plt.title("2D heat map")
10 plt.xlabel("Year")
11 sns.set_style("darkgrid")
12 plt.ylabel("Median income")
13 plt.show()
14
```



```
In [34]: 1 #histogram
2 import numpy as np
3 import seaborn as sb
4 import matplotlib.pyplot as plt
5 sb.set_style("darkgrid")
6 x=np.random.normal(40,10,100)
7 y=np.random.normal(40,10,50)
8 sb.histplot(x)
9 sb.histplot(y)
10 plt.xlabel("People")
11 plt.ylabel("Age")
12 plt.title("Histogram")
13 plt.show()
```

```
C:\Users\Nikhil\anaconda3\anaconda\envs\bb\Lib\site-packages\seaborn\_oldcor
e.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be remo
ved in a future version. Use isinstance(dtype, CategoricalDtype) instead
    if pd.api.types.is_categorical_dtype(vector):
C:\Users\Nikhil\anaconda3\anaconda\envs\bb\Lib\site-packages\seaborn\_oldcor
e.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be remo
ved in a future version. Convert inf values to NaN before operating instead.
    with pd.option_context('mode.use_inf_as_na', True):
C:\Users\Nikhil\anaconda3\anaconda\envs\bb\Lib\site-packages\seaborn\_oldcor
e.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be remo
ved in a future version. Use isinstance(dtype, CategoricalDtype) instead
    if pd.api.types.is_categorical_dtype(vector):
C:\Users\Nikhil\anaconda3\anaconda\envs\bb\Lib\site-packages\seaborn\_oldcor
e.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be remo
ved in a future version. Convert inf values to NaN before operating instead.
    with pd.option_context('mode.use_inf_as_na', True):
```



In []:

1