

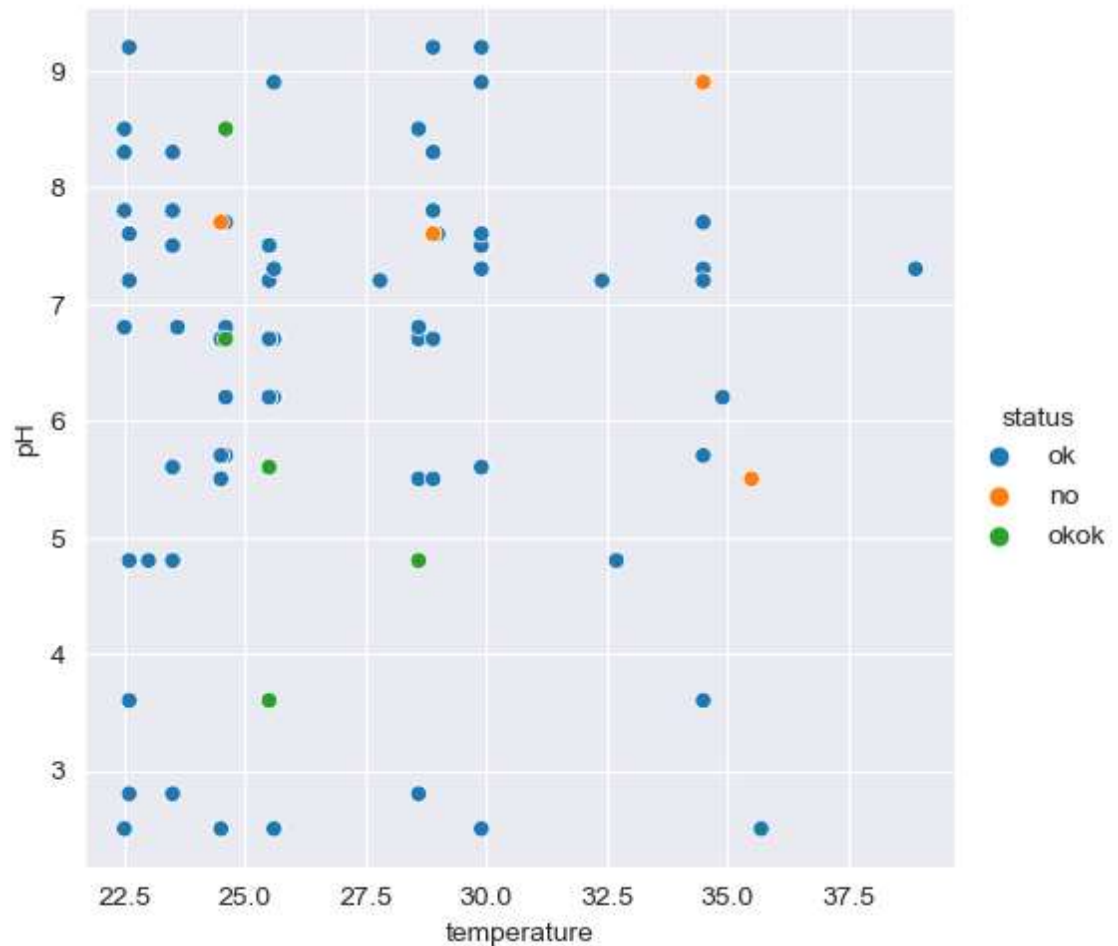
```
In [64]: import seaborn as sns
import pandas as pd
water=pd.read_csv("water_management")
water.head()
```

Out[64]:

	location	temperature	pH	acceptablerange	status
0	sample01	25.5	7.2	5-8	ok
1	sample02	24.5	2.5	18-25	ok
2	sample03	23.5	4.8	5	ok
3	sample04	34.5	7.3	12-15	no
4	sample05	22.5	6.8	15-25	ok

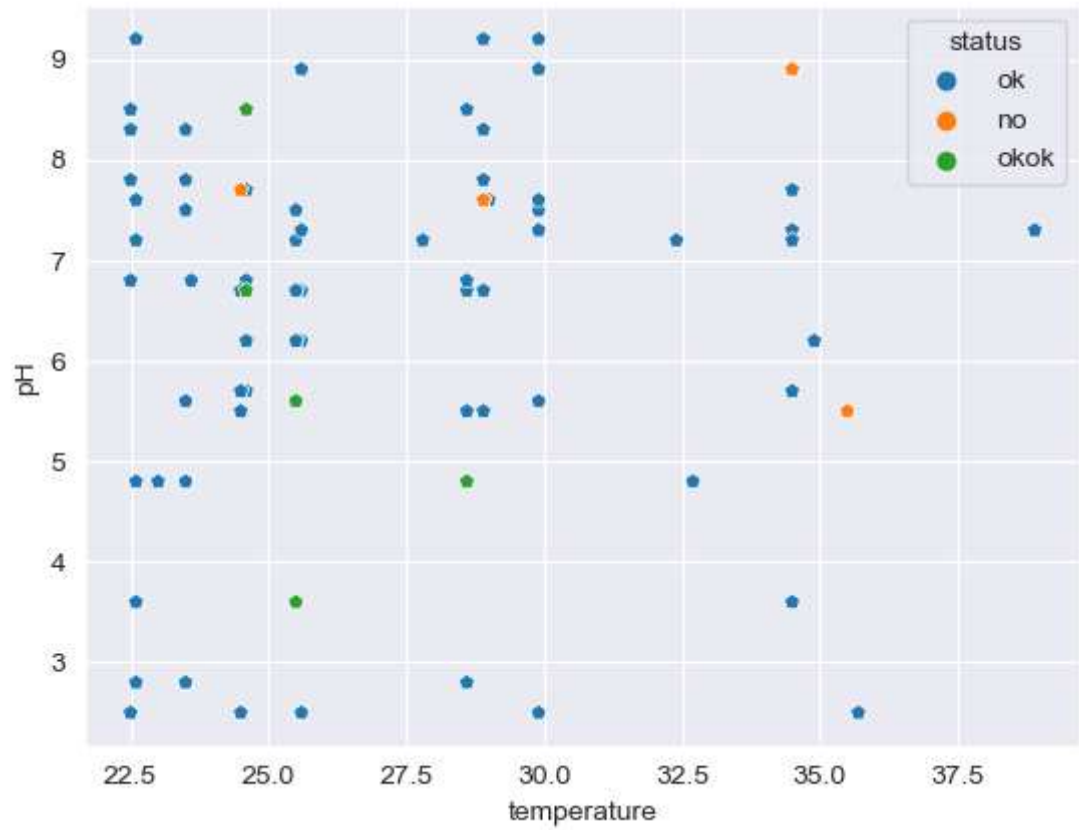
```
In [65]: sns.set_style("darkgrid")
sns.relplot(x="temperature",
            y="pH",
            hue="status",
            data=water)
```

Out[65]: <seaborn.axisgrid.FacetGrid at 0x1890f184f50>

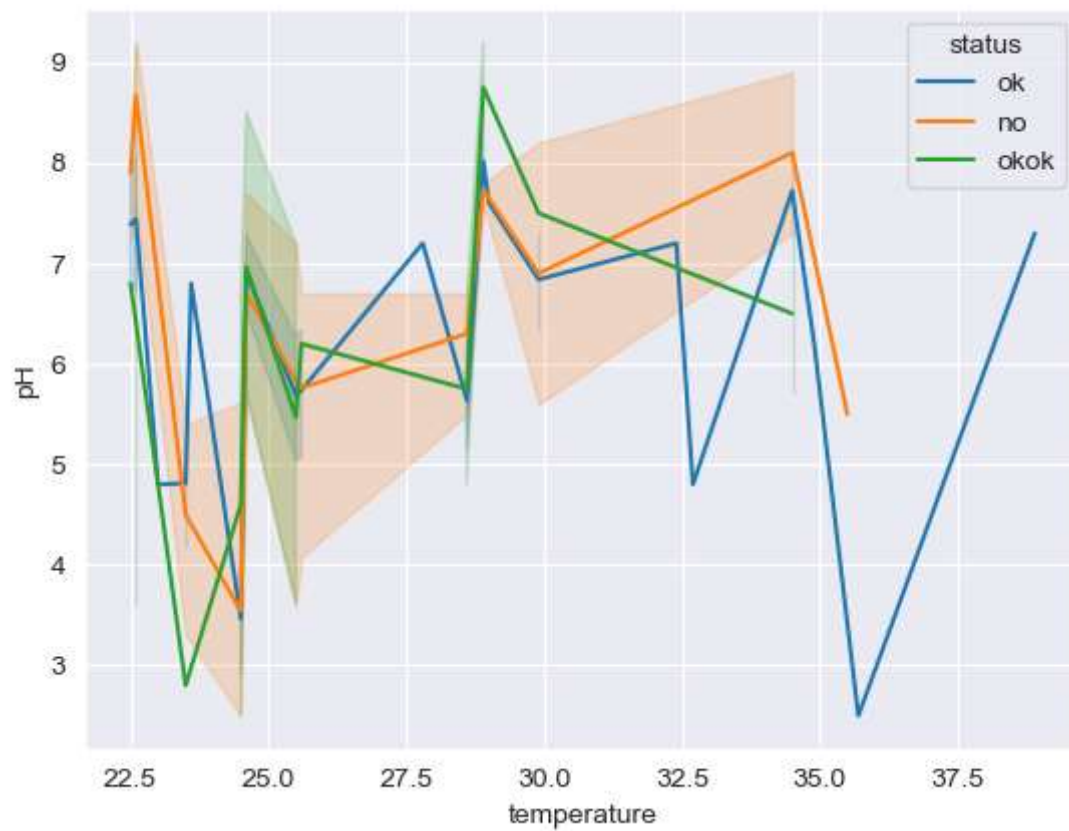


```
In [66]: sns.set_style("darkgrid")
sns.scatterplot(x="temperature",
                y="pH",
                hue="status",
                marker="p",
                data=water)
```

Out[66]: <Axes: xlabel='temperature', ylabel='pH'>

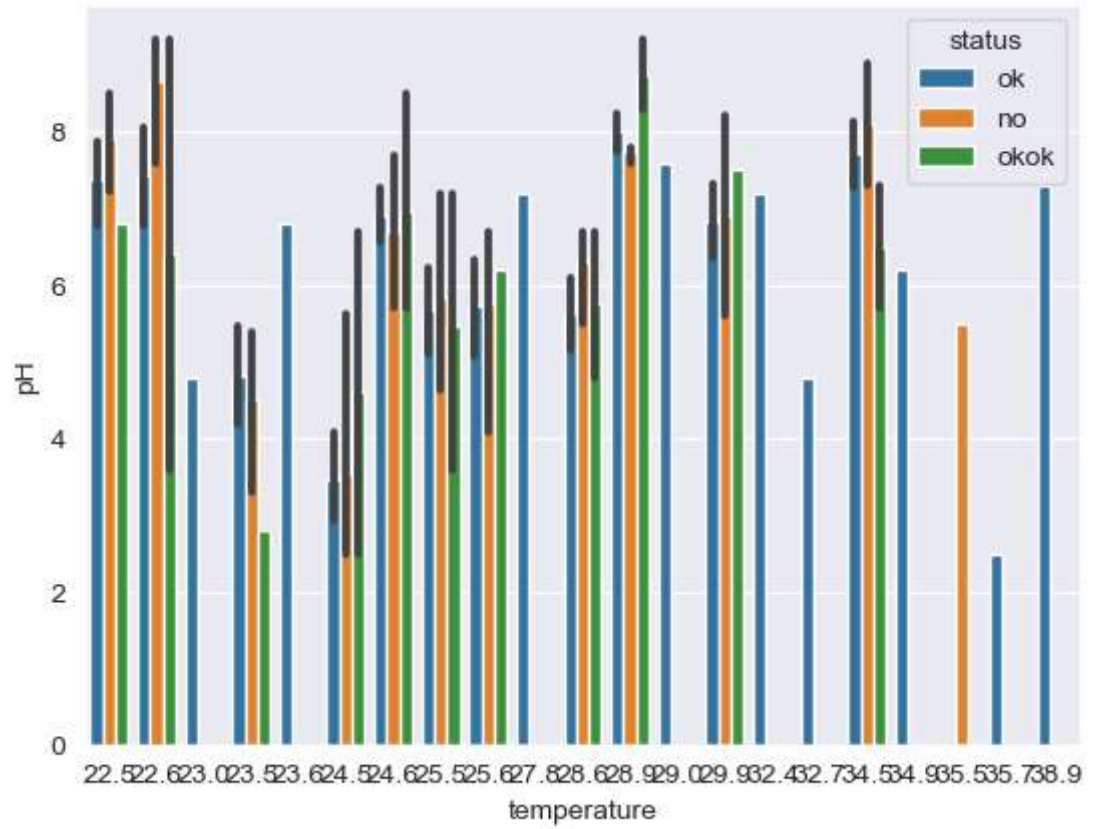


```
In [67]: ▶ import matplotlib.pyplot as plt
sns.set_style("darkgrid")
sns.lineplot(x="temperature",
             y="pH",
             hue="status",
             data=water)
plt.show()
```



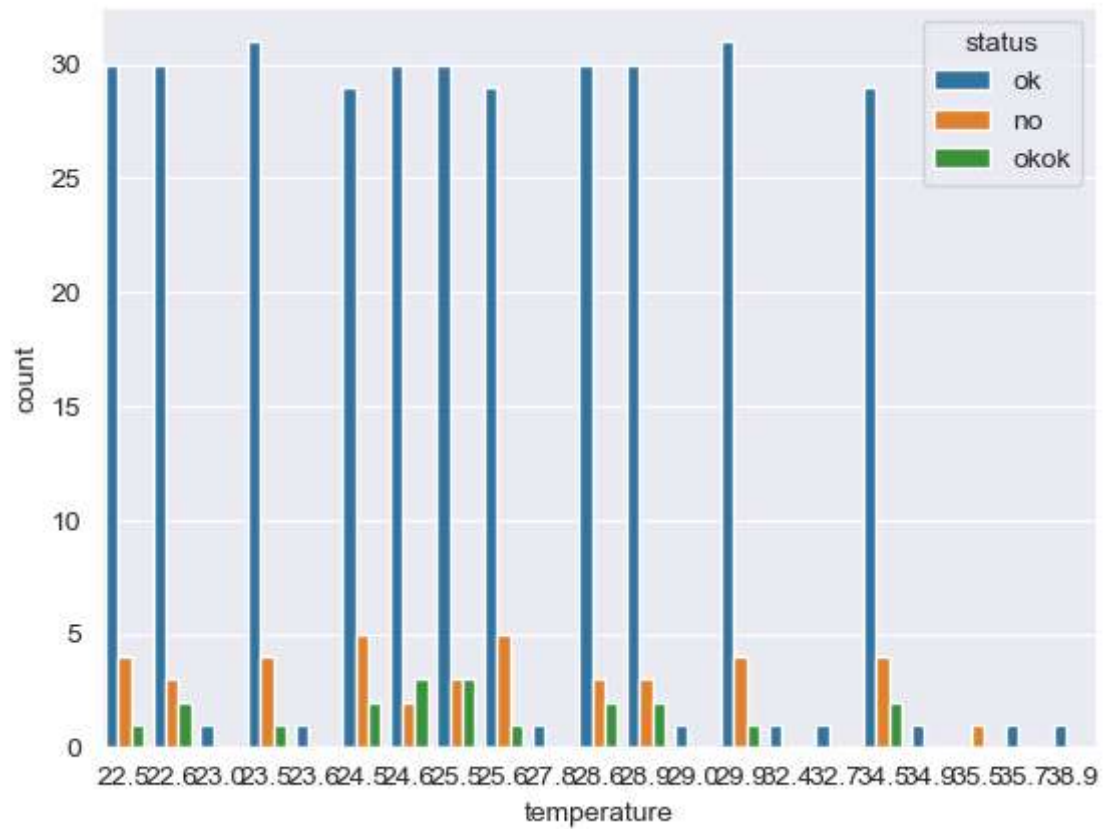
```
In [68]: sns.barplot(x="temperature",  
                    y="pH",  
                    hue="status",  
                    data=water)
```

Out[68]: <Axes: xlabel='temperature', ylabel='pH'>



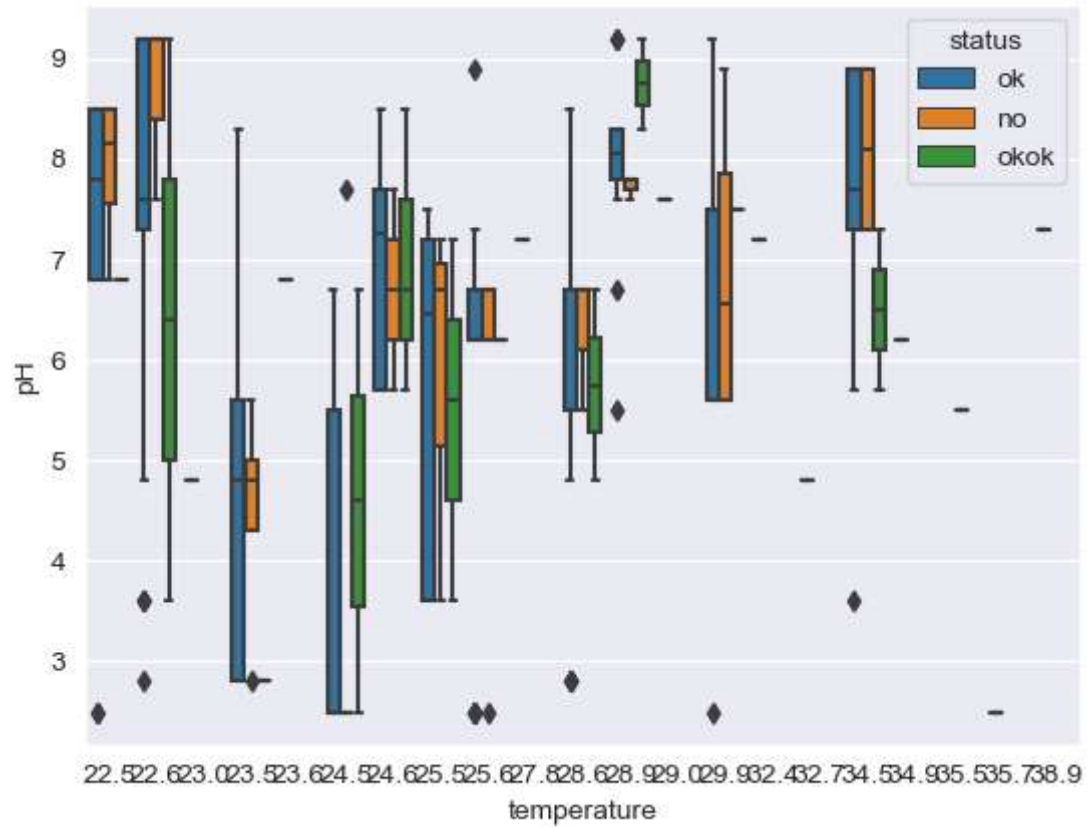
```
In [69]: sns.countplot(x="temperature",  
                        hue="status",  
                        data=water)
```

```
Out[69]: <Axes: xlabel='temperature', ylabel='count'>
```



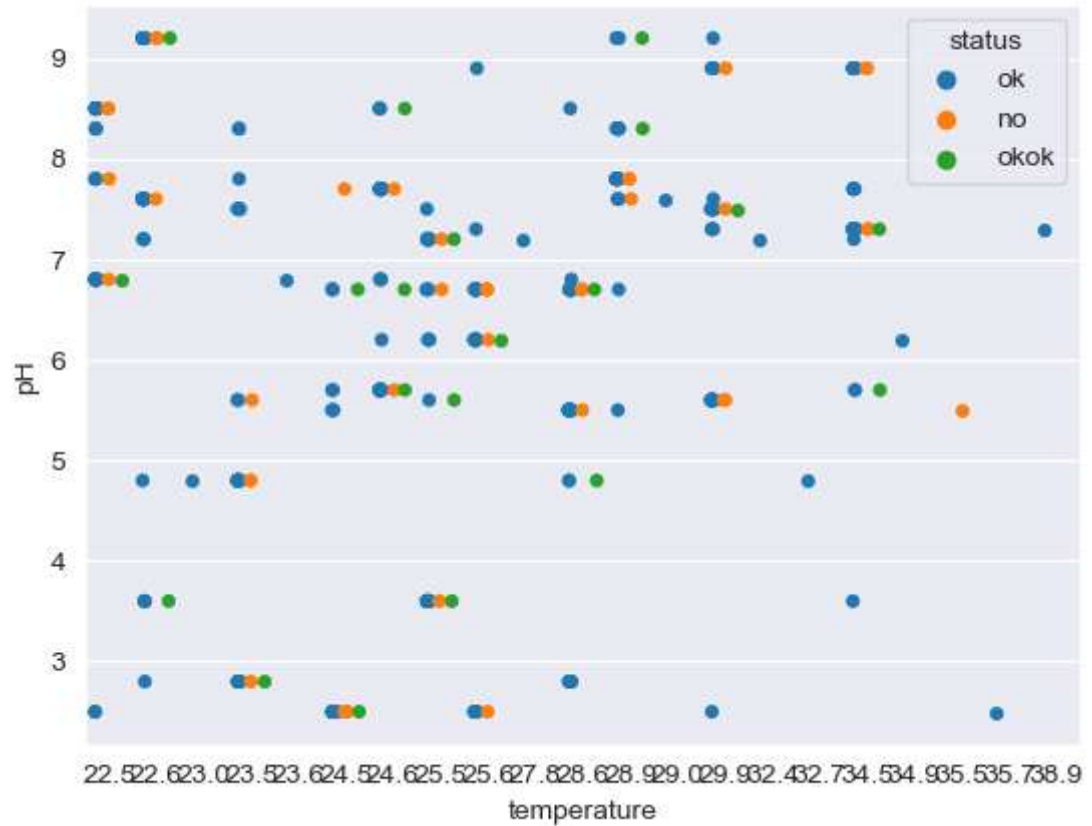
```
In [70]: sns.boxplot(x="temperature",  
                    y="pH",  
                    hue="status",  
                    data=water)
```

```
Out[70]: <Axes: xlabel='temperature', ylabel='pH'>
```



```
In [71]: sns.stripplot(x="temperature",  
                        y="pH",  
                        hue="status",  
                        jitter=True,  
                        dodge=True,  
                        data=water)
```

Out[71]: <Axes: xlabel='temperature', ylabel='pH'>



```
In [72]: sns.swarmplot(x="temperature",  
                      y="pH",  
                      hue="status",  
                      data=water)
```

E:\new\Lib\site-packages\seaborn\categorical.py:3544: UserWarning: 60.0% of the points cannot be placed; you may want to decrease the size of the markers or use stripplot.

warnings.warn(msg, UserWarning)

E:\new\Lib\site-packages\seaborn\categorical.py:3544: UserWarning: 58.3% of the points cannot be placed; you may want to decrease the size of the markers or use stripplot.

warnings.warn(msg, UserWarning)

E:\new\Lib\site-packages\seaborn\categorical.py:3544: UserWarning: 66.7% of the points cannot be placed; you may want to decrease the size of the markers or use stripplot.

warnings.warn(msg, UserWarning)

E:\new\Lib\site-packages\seaborn\categorical.py:3544: UserWarning: 62.9% of the points cannot be placed; you may want to decrease the size of the markers or use stripplot.

warnings.warn(msg, UserWarning)

E:\new\Lib\site-packages\seaborn\categorical.py:3544: UserWarning: 68.6% of the points cannot be placed; you may want to decrease the size of the markers or use stripplot.

warnings.warn(msg, UserWarning)

E:\new\Lib\site-packages\seaborn\categorical.py:3544: UserWarning: 61.1% of the points cannot be placed; you may want to decrease the size of the markers or use stripplot.

warnings.warn(msg, UserWarning)

Out[72]: <Axes: xlabel='temperature', ylabel='pH'>





```
In [73]: sns.violinplot(x="temperature",
                        y="pH",
                        hue="status",
                        data=water)
sns.swarmplot(x="temperature",
              y="pH",
              hue="status",
              color="black",
              data=water)
```

E:\new\Lib\site-packages\seaborn\categorical.py:166: FutureWarning: Setting a gradient palette using color= is deprecated and will be removed in version 0.13. Set `palette='dark:black'` for same effect.

warnings.warn(msg, FutureWarning)

E:\new\Lib\site-packages\seaborn\categorical.py:3544: UserWarning: 60.0% of the points cannot be placed; you may want to decrease the size of the markers or use stripplot.

warnings.warn(msg, UserWarning)

E:\new\Lib\site-packages\seaborn\categorical.py:3544: UserWarning: 58.3% of the points cannot be placed; you may want to decrease the size of the markers or use stripplot.

warnings.warn(msg, UserWarning)

E:\new\Lib\site-packages\seaborn\categorical.py:3544: UserWarning: 66.7% of the points cannot be placed; you may want to decrease the size of the markers or use stripplot.

warnings.warn(msg, UserWarning)

E:\new\Lib\site-packages\seaborn\categorical.py:3544: UserWarning: 62.9% of the points cannot be placed; you may want to decrease the size of the markers or use stripplot.

warnings.warn(msg, UserWarning)

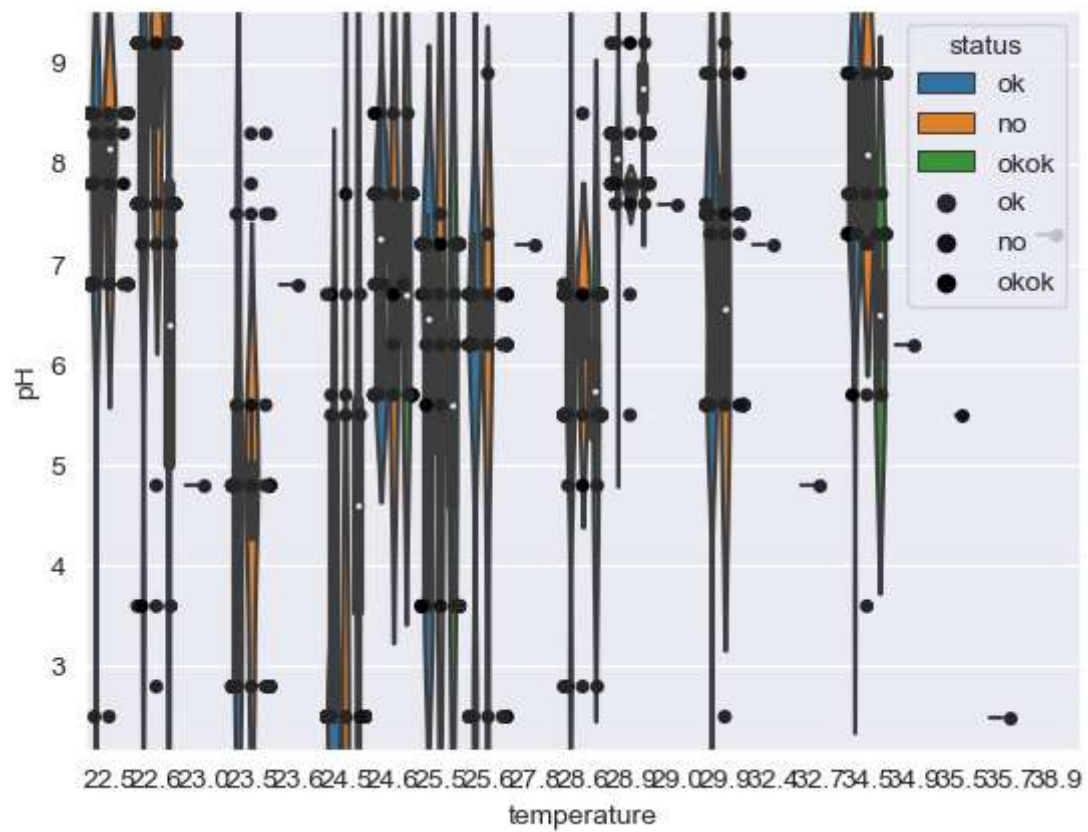
E:\new\Lib\site-packages\seaborn\categorical.py:3544: UserWarning: 68.6% of the points cannot be placed; you may want to decrease the size of the markers or use stripplot.

warnings.warn(msg, UserWarning)

E:\new\Lib\site-packages\seaborn\categorical.py:3544: UserWarning: 61.1% of the points cannot be placed; you may want to decrease the size of the markers or use stripplot.

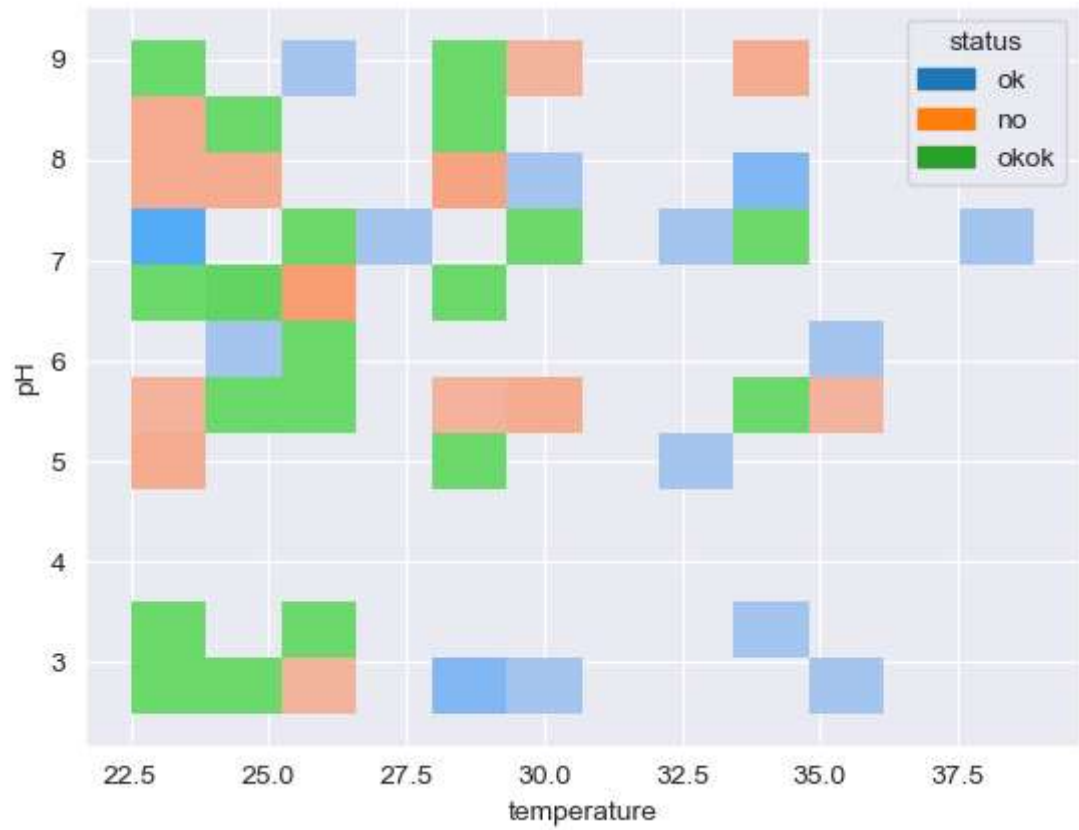
warnings.warn(msg, UserWarning)

Out[73]: <Axes: xlabel='temperature', ylabel='pH'>



```
In [74]: import seaborn as sns
import pandas as pd
heart=pd.read_csv("water_management")
sns.histplot(x="temperature",
             y="pH",
             hue="status",
             data=water)
```

Out[74]: <Axes: xlabel='temperature', ylabel='pH'>



```
In [75]: sns.distplot(water["temperature"],kde=True,color="red",bins=30)
```

```
C:\Users\Chandana.M\AppData\Local\Temp\ipykernel_12720\2324261578.py:
1: UserWarning:
```

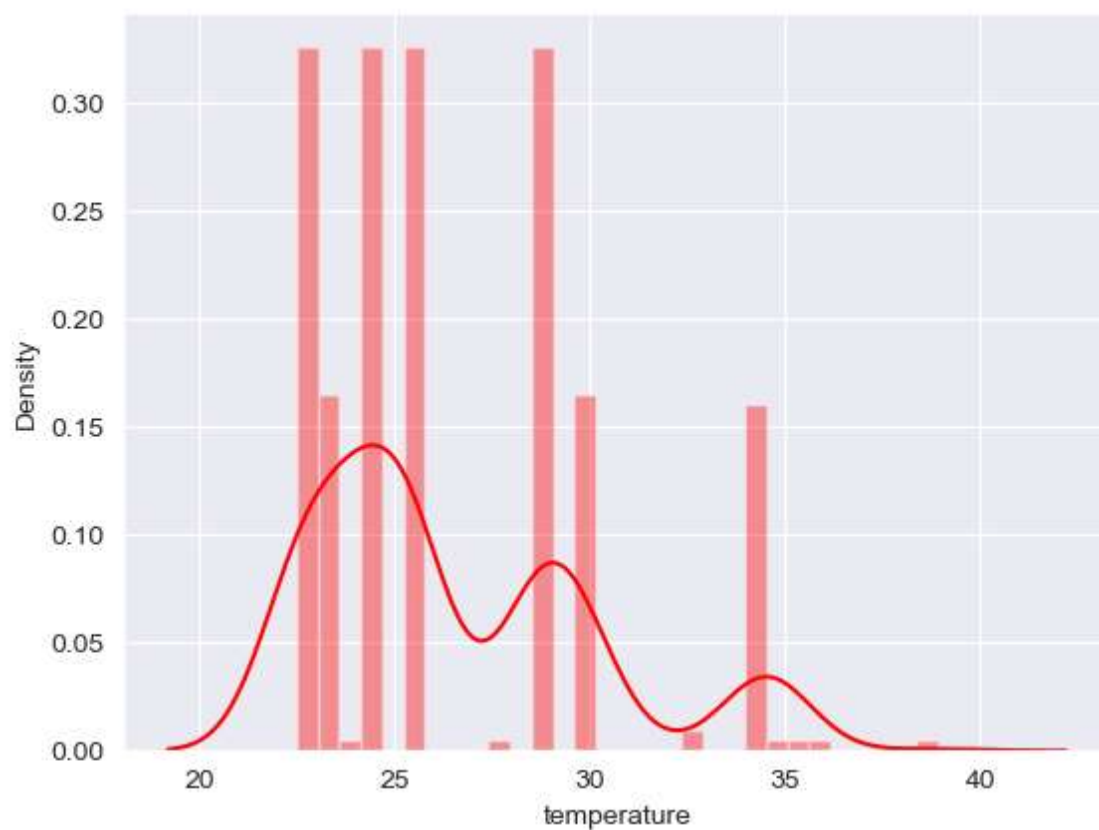
```
`distplot` is a deprecated function and will be removed in seaborn v0.
14.0.
```

Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see <https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751> (<https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751>)

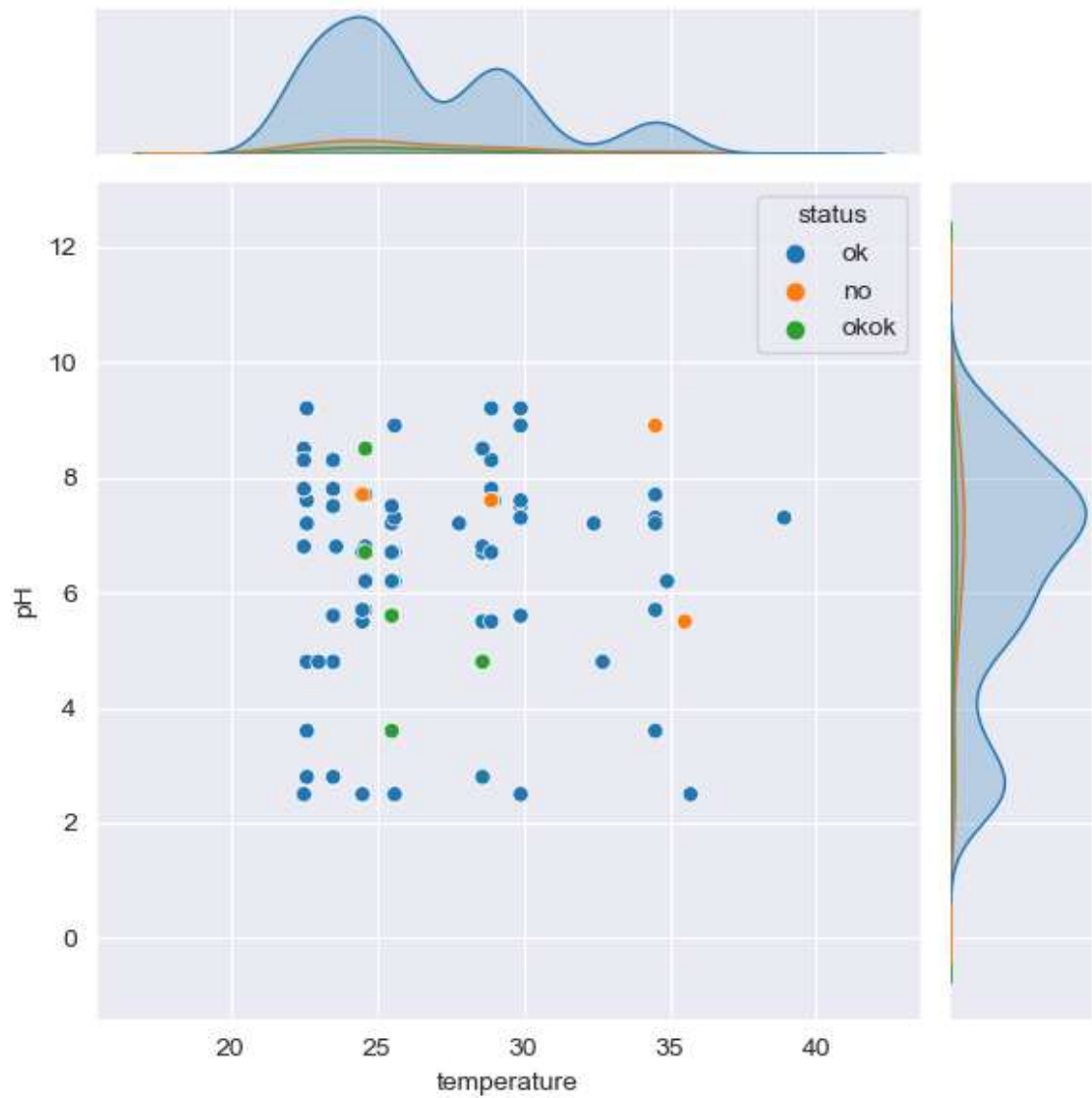
```
sns.distplot(water["temperature"],kde=True,color="red",bins=30)
```

```
Out[75]: <Axes: xlabel='temperature', ylabel='Density'>
```



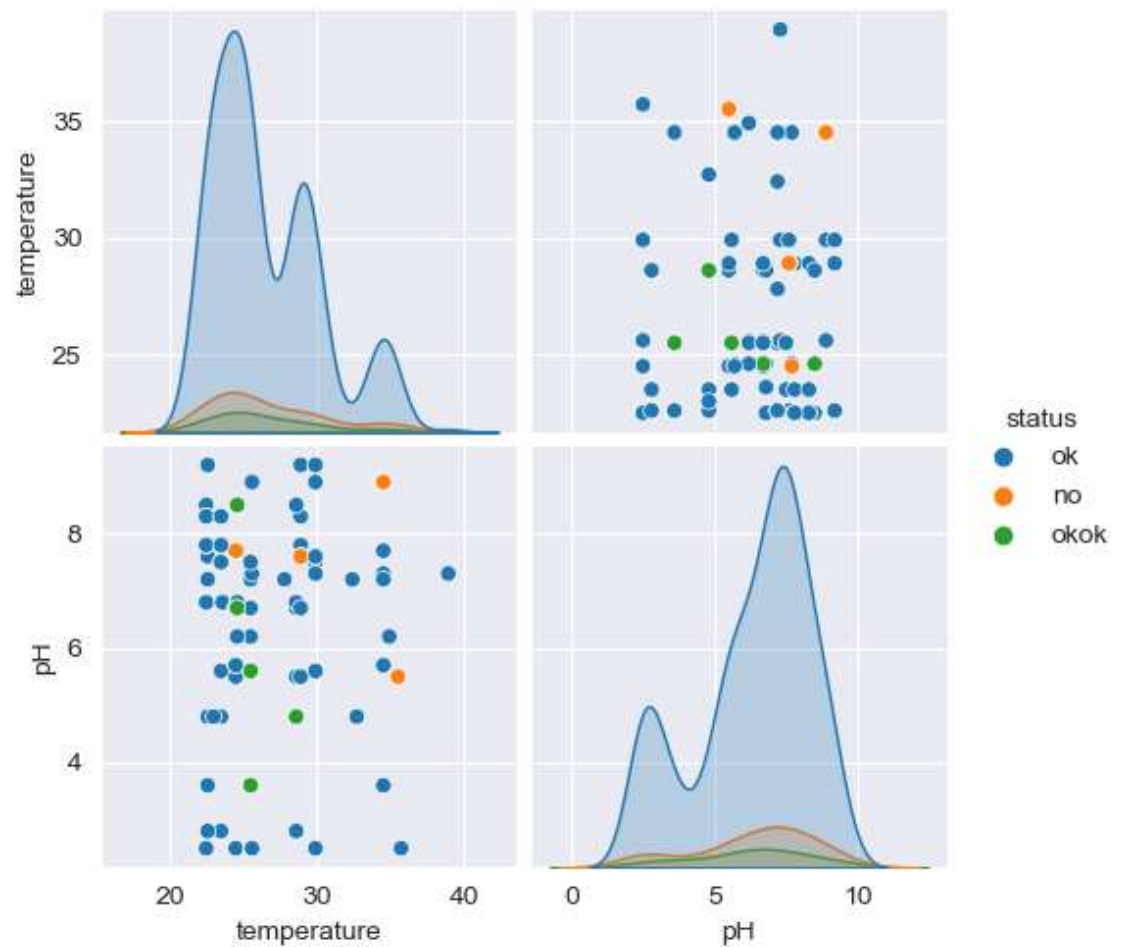
```
In [76]: sns.jointplot(x="temperature",  
                      y="pH",  
                      hue="status",  
                      data=water)
```

Out[76]: <seaborn.axisgrid.JointGrid at 0x18913107210>



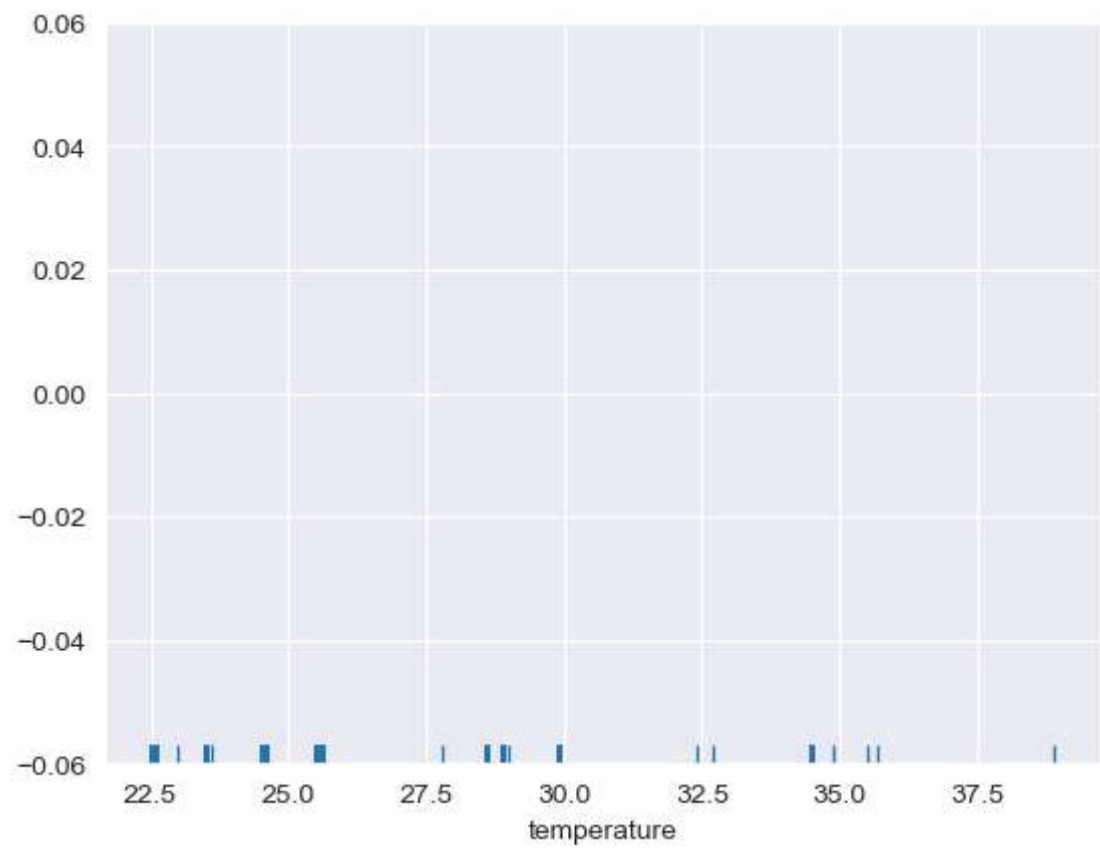
```
In [77]: sns.pairplot(data=water, hue="status")
```

```
Out[77]: <seaborn.axisgrid.PairGrid at 0x18913422350>
```



```
In [78]: sns.rugplot(data=water,x="temperature")
```

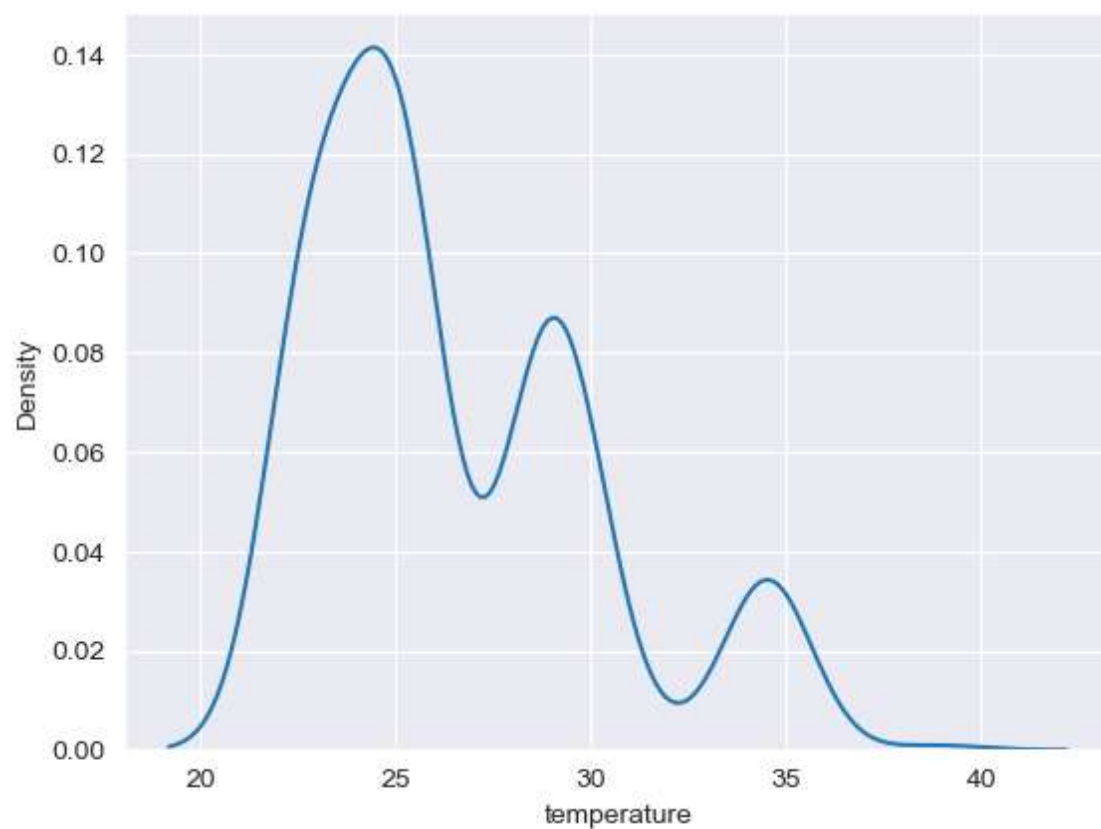
```
Out[78]: <Axes: xlabel='temperature'>
```





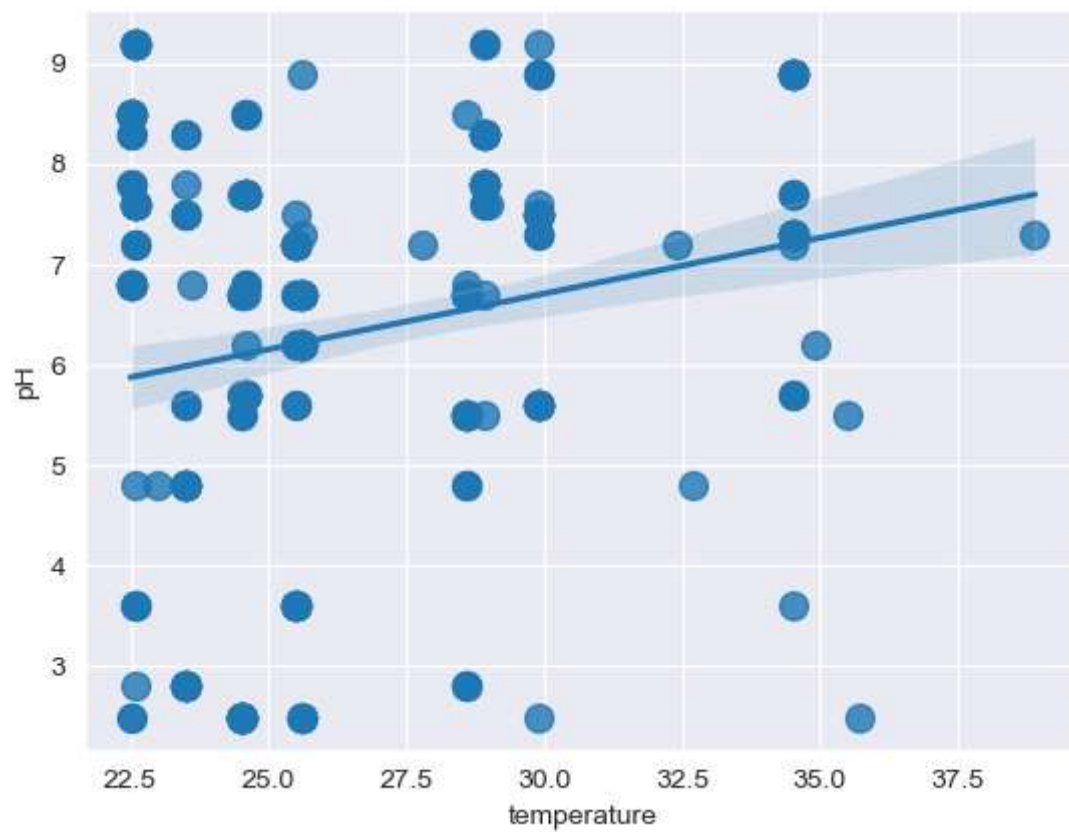
```
In [79]: sns.kdeplot(data=water,x="temperature")
```

```
Out[79]: <Axes: xlabel='temperature', ylabel='Density'>
```

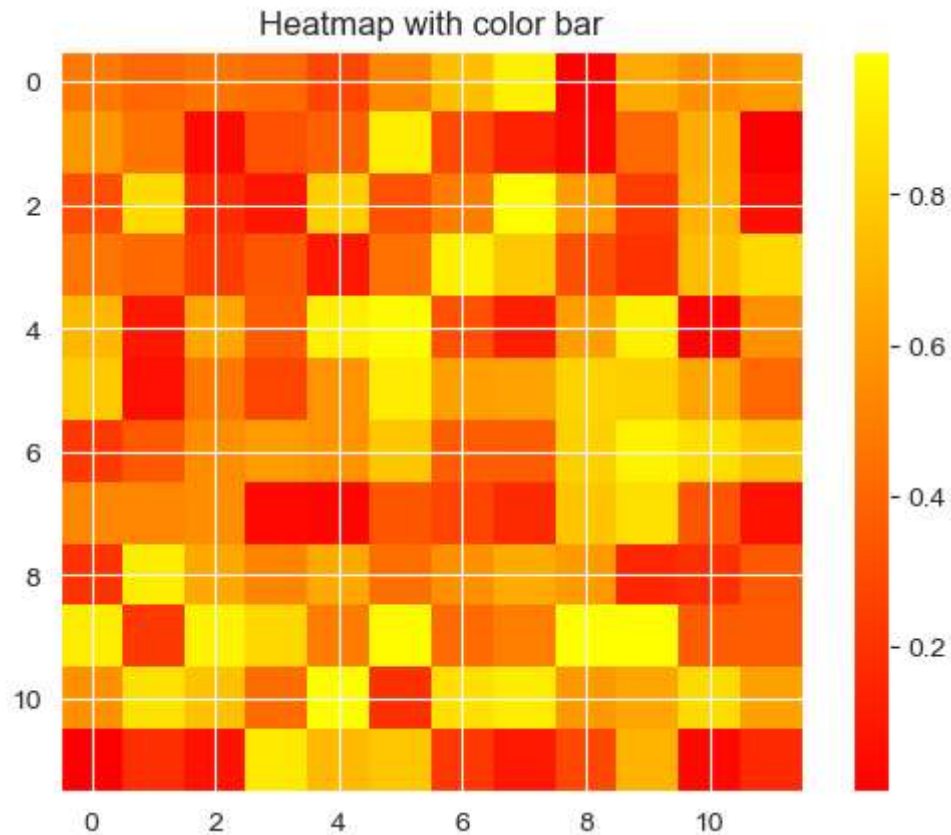


```
In [82]: sns.regplot(x="temperature",  
                    y="pH",  
                    data=water,  
                    scatter_kws={"s":100})
```

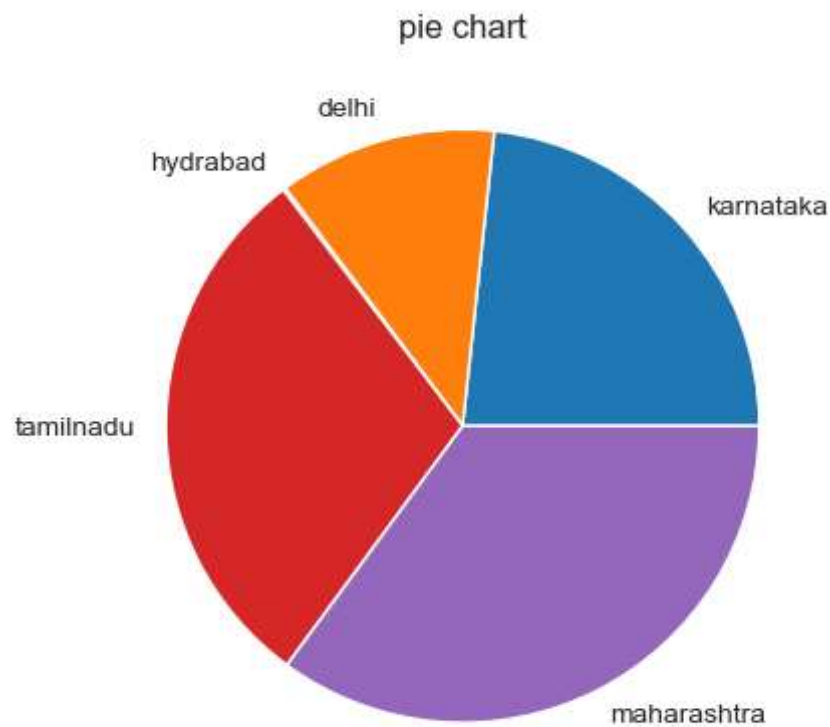
Out[82]: <Axes: xlabel='temperature', ylabel='pH'>



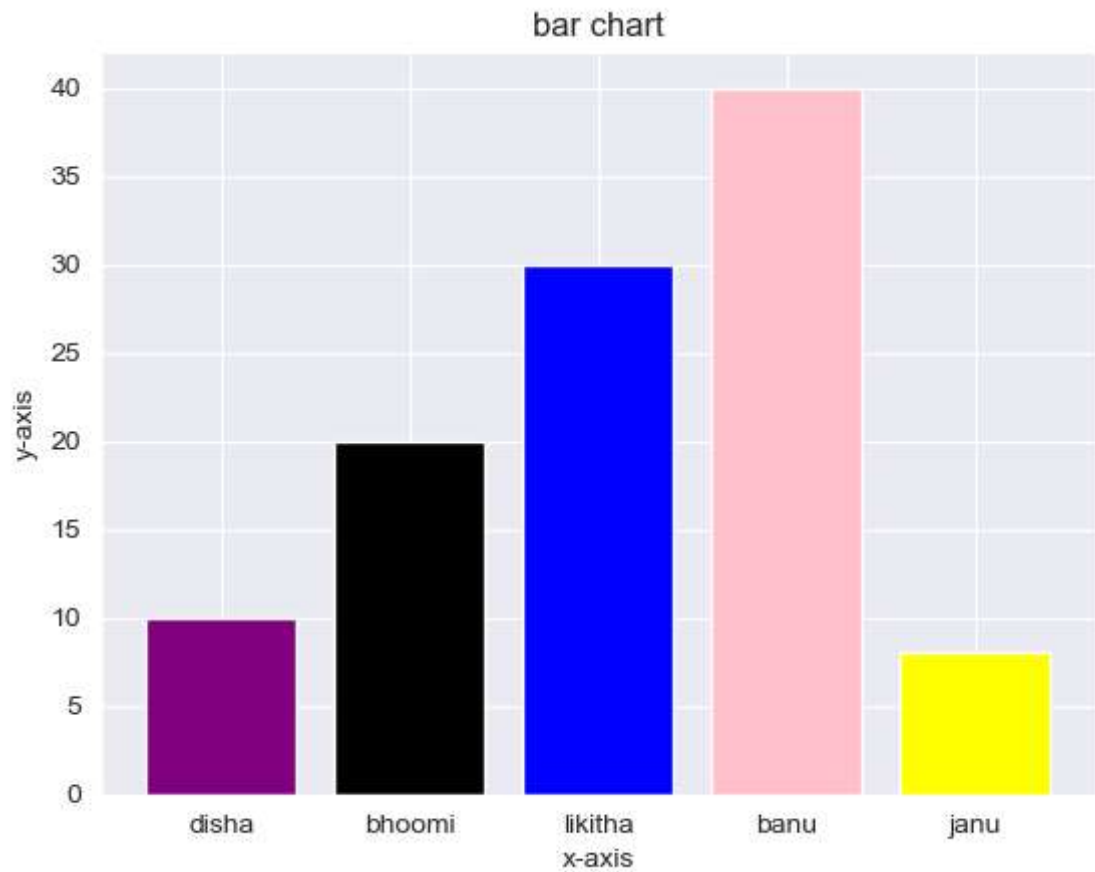
```
In [81]: ▶ #heatmap
import numpy as np
import matplotlib.pyplot as plt
data = np.random.random((12, 12))
plt.imshow(data, cmap='autumn', interpolation='nearest')
plt.colorbar()
plt.title("Heatmap with color bar")
plt.show()
```



```
In [83]: # create a pie chart using proportions visualization  
import matplotlib.pyplot as plt  
import numpy as np  
import seaborn as sns  
# creating a style  
sns.set_style("darkgrid")  
# assigning a values to pie chart  
x=["karnataka", "delhi", "hydrabad", "tamilnadu", "maharashtra"]  
e=[200211, 102344, 1220, 254103, 301552]  
# creating a pie chart using pie  
plt.pie(e, labels=x)  
plt.title("pie chart")  
plt.show()
```

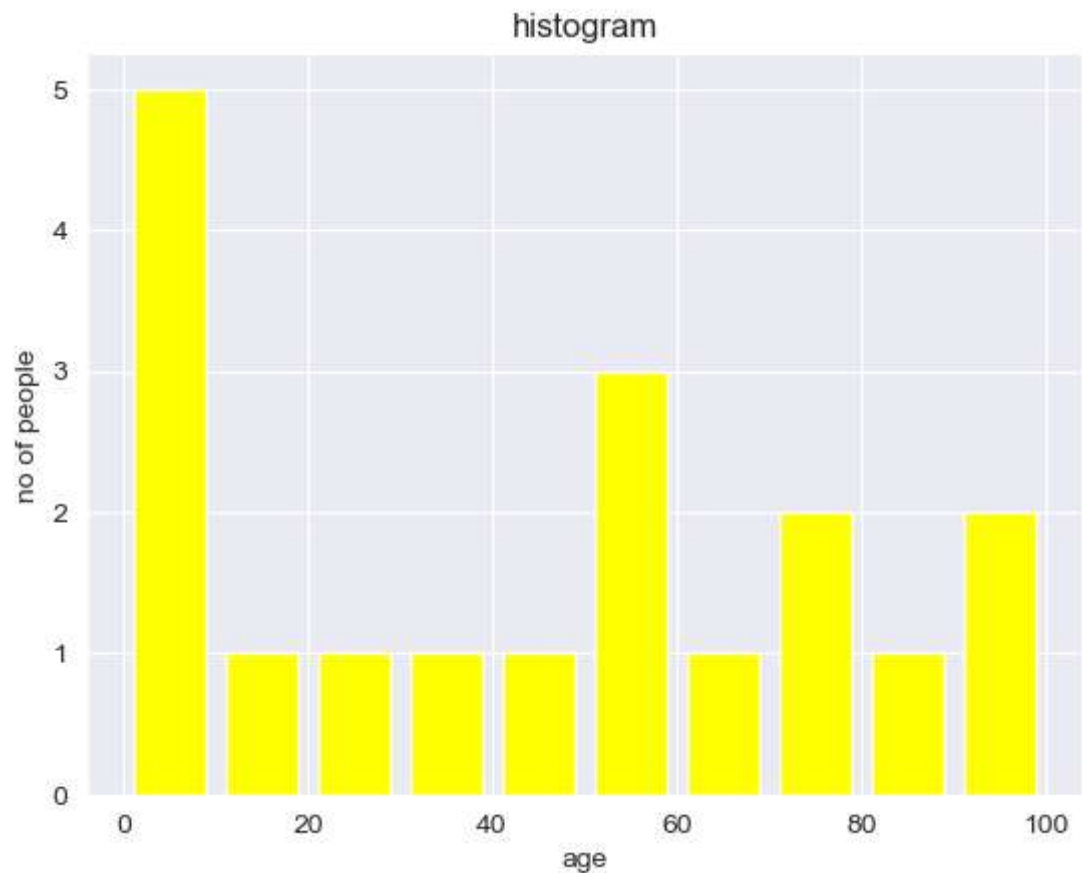


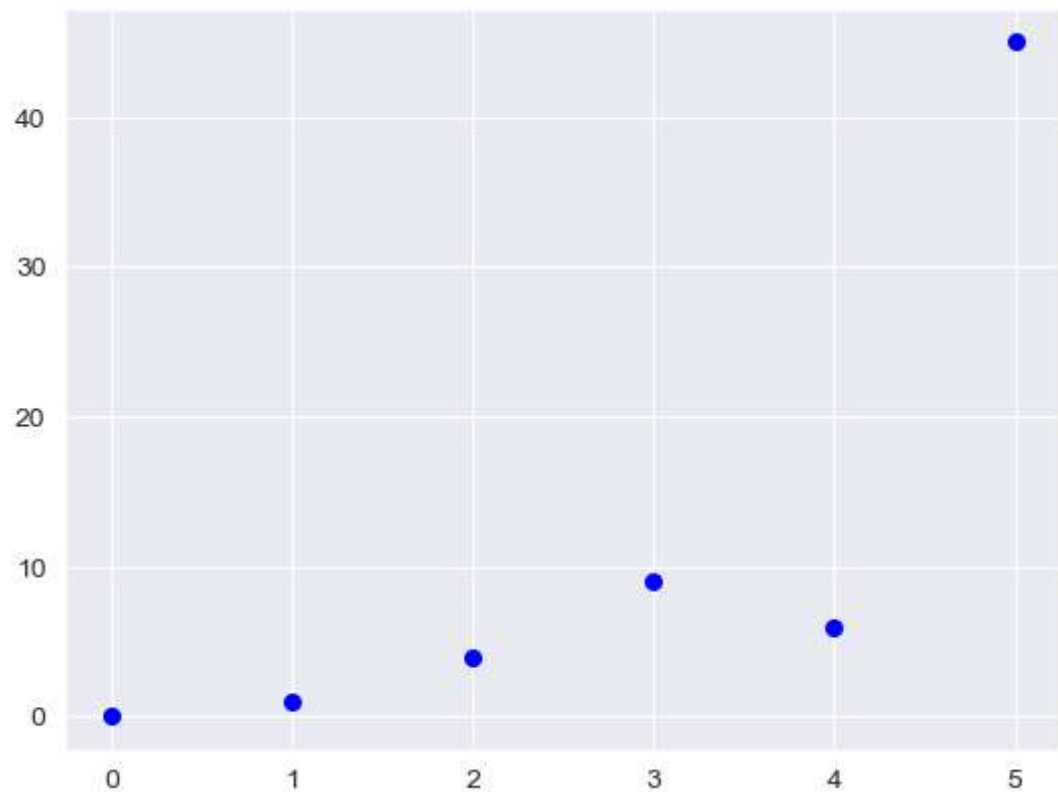
```
In [86]: ▶ #bar chart
height=[10,20,30,40,8]
names=["disha","bhoomi","likitha","banu","janu"]
c1=["purple","black","blue","pink","yellow"]
plt.bar(names,height,width=0.8,color=c1)
plt.xlabel("x-axis")
plt.ylabel("y-axis")
plt.title("bar chart")
plt.show()
```



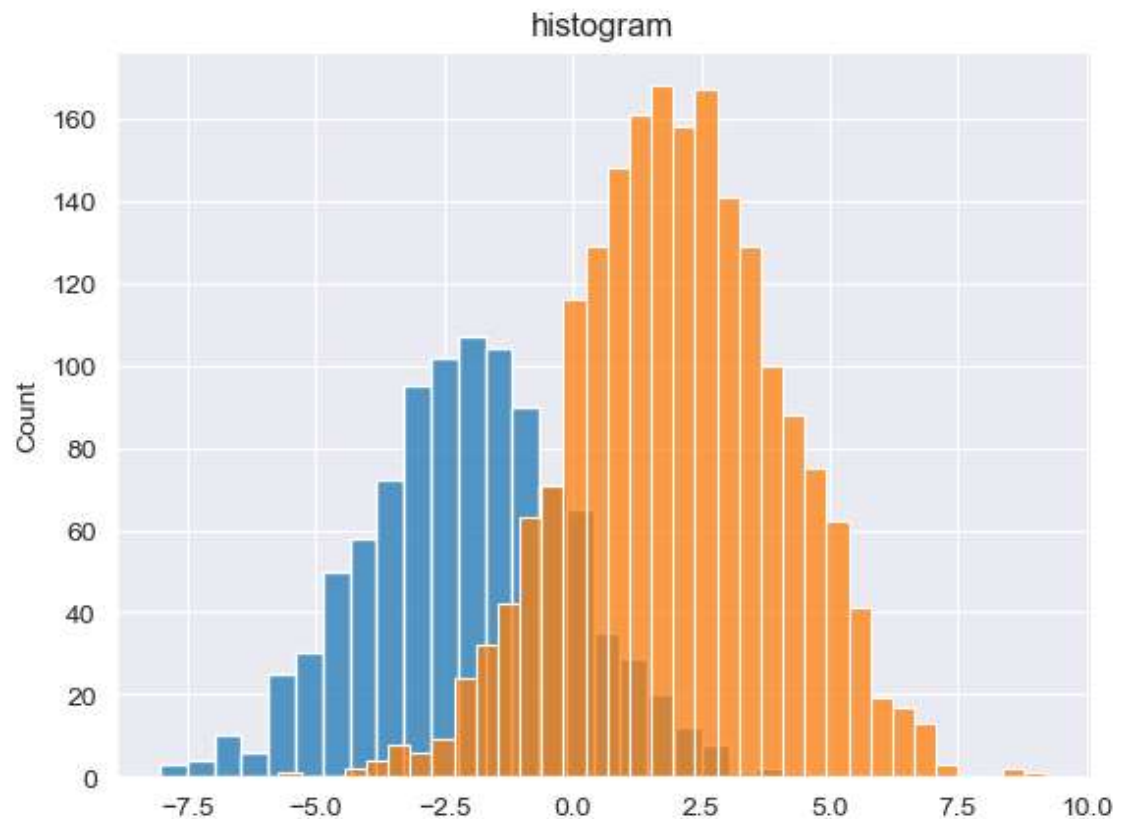
```
In [89]: ▶ #histogram
ages=[2,52,41,74,84,52,74,20,62,32,10,52,5,4,1,6,95,100]
range=(0,100)
bins=10
plt.hist(ages,bins,range,color="yellow",histtype="bar",rwidth=0.8)
plt.xlabel("age")
plt.ylabel("no of people")
plt.title("histogram")
plt.show()

#scatter plot
x_values=[0,1,2,3,4,5]
y_values=[0,1,4,9,6,45]
plt.scatter(x_values,y_values,s=30,color="blue")
plt.show()
```



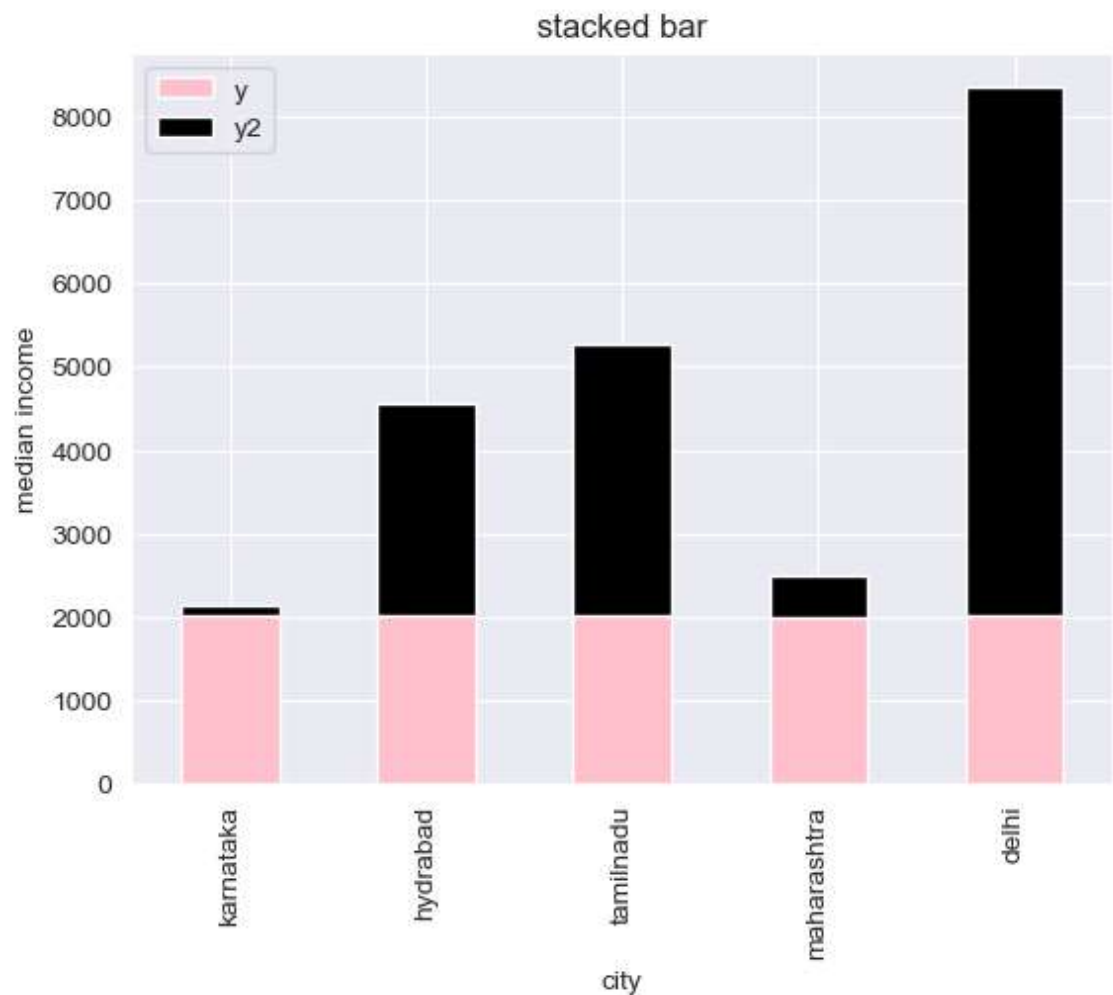


```
In [97]: #double plot histogram  
import pandas as pd  
import seaborn as sns  
import matplotlib.pyplot as plt  
import numpy as np  
#creating a style  
sns.set_style('darkgrid')  
#declaring a random and assignning a range to it  
y1=np.random.normal(-2,2,1000)  
y2=np.random.normal(2,2,2000)  
# creating a histogram graph using hisplot  
sns.histplot(y1)  
sns.histplot(y2)  
plt.title('histogram')  
plt.show()
```

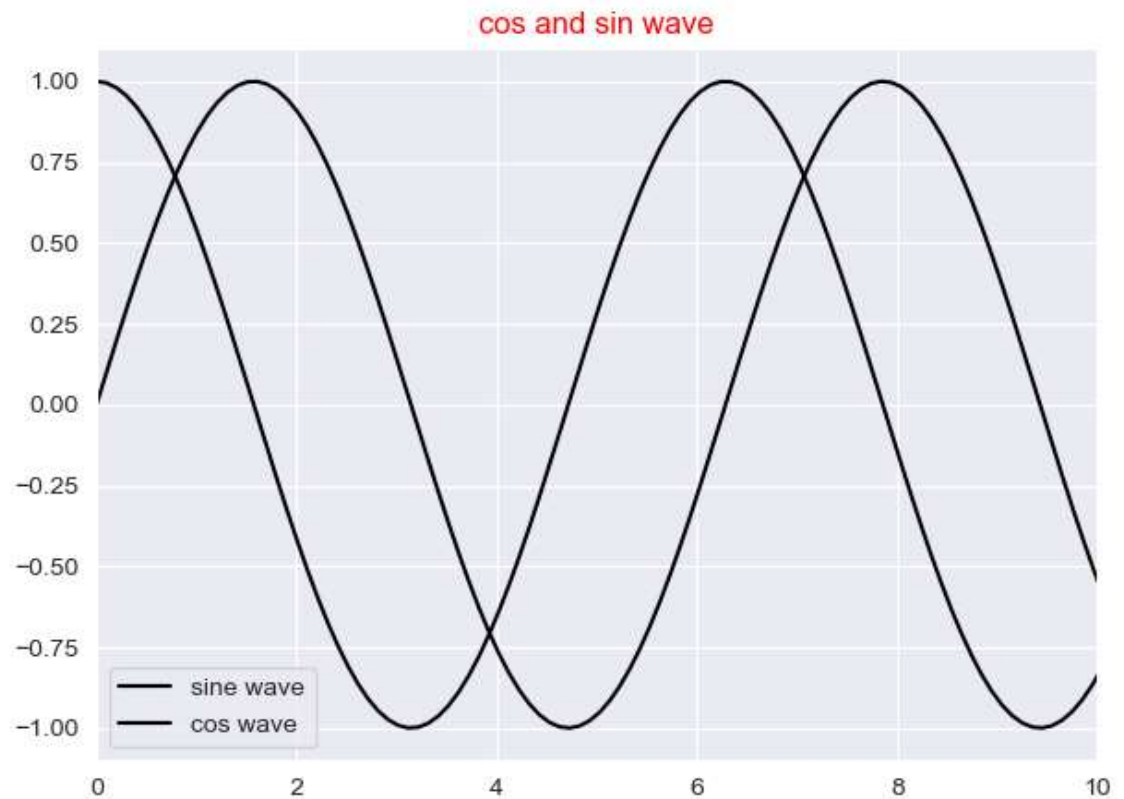




```
In [92]: #stacked  
import seaborn as sns  
import pandas as pd  
import matplotlib.pyplot as plt  
sns.set_style("darkgrid")  
data=pd.DataFrame({'y':[2012,2015,2014,2006,2018] ,  
'y2':[120,2549,3254,487,6325]} ,  
index=["karnataka","hydrabad","tamilnadu","maharashtra","delhi"])  
data.plot(kind="bar",stacked=True,color=["pink","black"])  
plt.title("stacked bar")  
plt.xlabel("city")  
plt.ylabel("median income")  
plt.show()
```



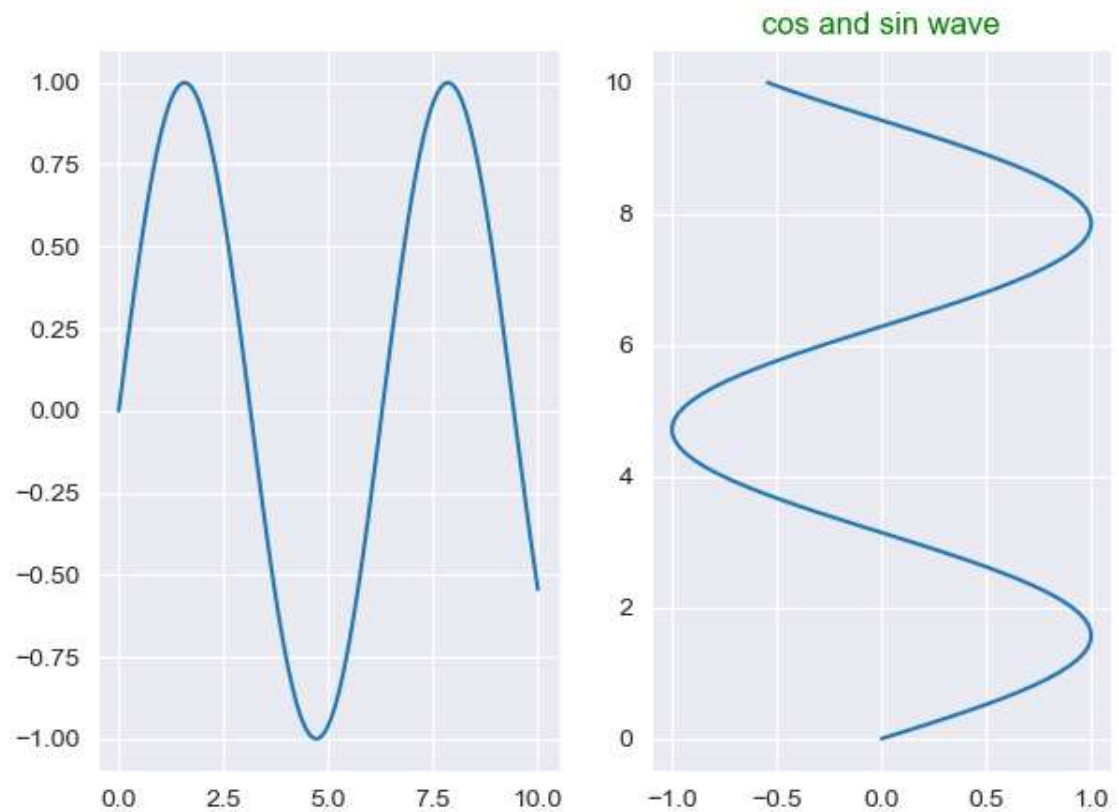
```
In [96]: #cos and sin  
import pandas as pd  
import seaborn as sns  
import matplotlib.pyplot as plt  
import numpy as np  
sns.set_style("darkgrid")  
fig, ax = plt.subplots(figsize=(7,5))  
x = np.linspace(0,20,200)  
y = np.sin(x)  
ax.plot(x,y, label="sine wave",color="black")  
y1=np.cos(x)  
ax.plot(x,y1, label="cos wave",color="black")  
ax.set_xlim(0,10)  
ax.set_ylim(-1,1)  
plt.title('cos and sin wave',color='r')  
ax.legend()  
plt.show()
```



```
In [95]: #cos and sin sub plots
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
sns.set_style("darkgrid")
fig, ax = plt.subplots(figsize=(7,5))
x = np.linspace(0,10,100)
y = np.sin(x)
ax.plot(x,y, label="sine wave")
plt.subplot(121)
plt.plot(x,y)
plt.subplot(122)
plt.plot(y,x)
ax.set_xlim(0,10)
ax.set_ylim(0,10)
plt.title('cos and sin wave',color='green')
ax.legend()
plt.show
```

C:\Users\Chandana.M\AppData\Local\Temp\ipykernel\_12720\725055847.py:1  
 1: 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(121)

Out[95]: <function matplotlib.pyplot.show(close=None, block=None)>



In [ ]: ▶