

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from matplotlib.pyplot import figure
```

```
import seaborn as sns
```

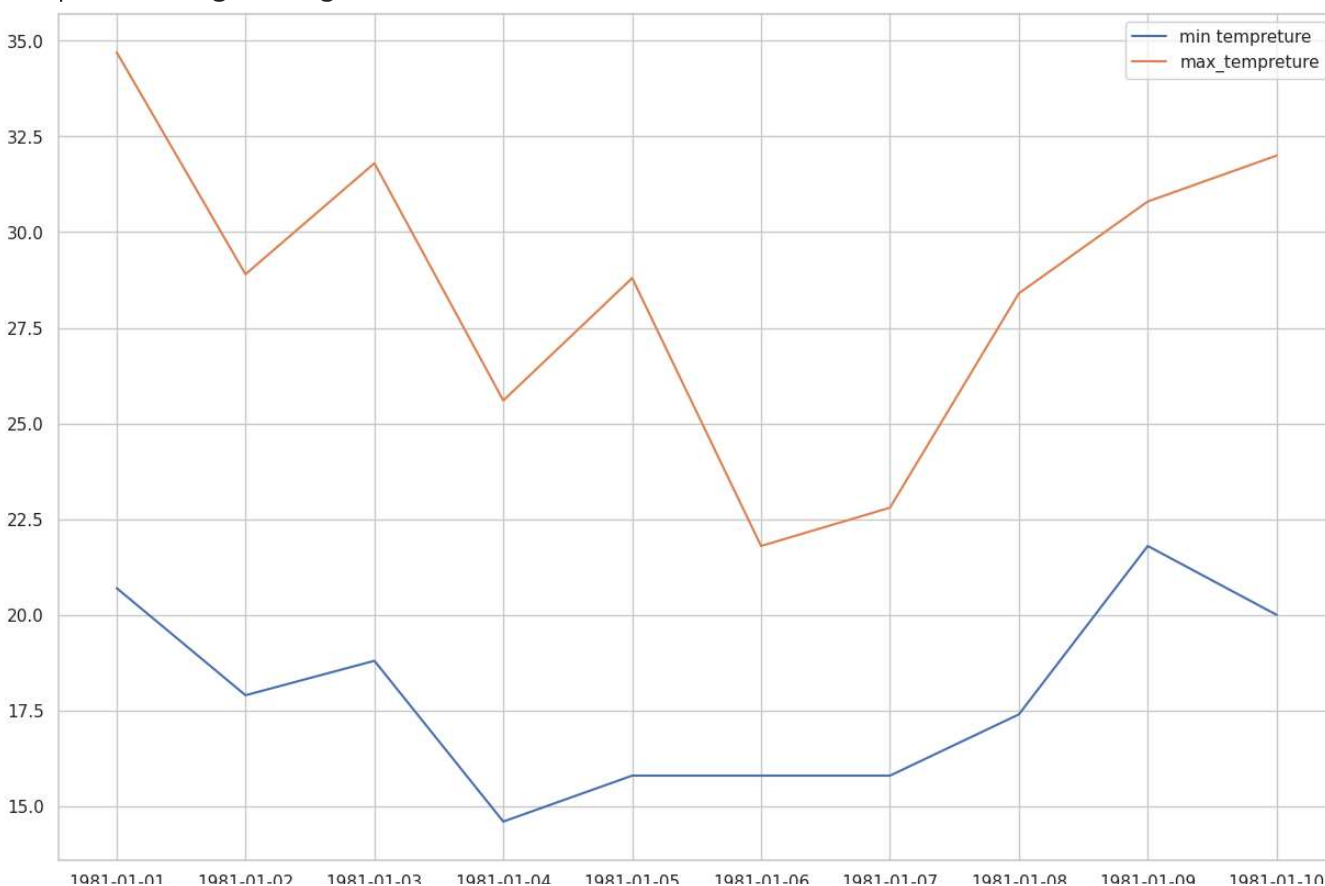
```
%matplotlib inline
```

```
dates = [ '1981-01-01', '1981-01-02', '1981-01-03', '1981-01-04', '1981-01-05',
          '1981-01-06', '1981-01-07', '1981-01-08', '1981-01-09', '1981-01-10']
min_temperature = [20.7,17.9, 18.8, 14.6, 15.8, 15.8,15.8, 17.4, 21.8, 20.0]
max_temperature = [34.7, 28.9, 31.8,25.6, 28.8, 21.8, 22.8, 28.4, 30.8, 32.0]
```

```
fig,axes = plt.subplots(nrows= 1, ncols= 1, figsize=(15,10))
axes.plot (dates, min_temperature, label = 'min temperture')
axes.plot (dates, max_temperature, label = 'max_tempreture')
axes.legend()
```

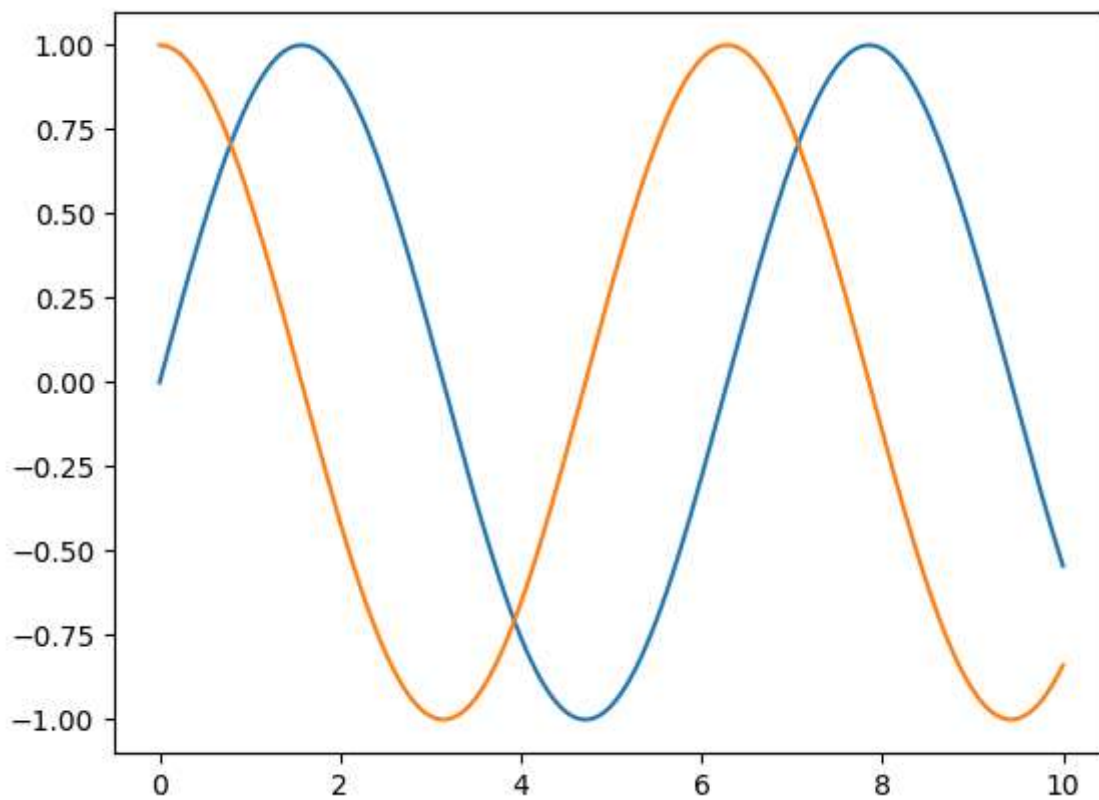


<matplotlib.legend.Legend at 0x7e73399568f0>



```
x=np.linspace(0,10,1000)
plt.plot(x,np.sin(x),x,np.cos(x))
```

↗ [<matplotlib.lines.Line2D at 0x7e733e5fb940>,
<matplotlib.lines.Line2D at 0x7e733e5fb9a0>]



`sns.set()`

#1.Relational Plots

#Line plot: The line plot is one of the most basic plot in the seaborn library

#mainly used to visualize data in time series,

sns.set(Style="dark")

fig, ax=plt.subplots(ncols=2,nrows=1,figsize=(10,5))

#load dataset

df=sns.load_dataset("tips")

print(df.head())

sns.lineplot(x="total_bill",y="tip",hue="size",style="time",data=df,ax=ax[0]).set_title("L

sct_plt=sns.scatterplot(x="total_bill", y = "tip", hue = "size", style= "time", data=df,ax=

sct_plt.figure.savefig('scatter_plt')

print("plot Saved")

import seaborn as sns

import matplotlib.pyplot as plt

sns.set(style="dark")

fig, ax = plt.subplots(ncols=2, nrows=1, figsize=(10, 5))

Load dataset

df = sns.load_dataset("tips")

print(df.head())

sns.lineplot(x="total_bill", y="tip", hue="size", style="time", data=df, ax=ax[0]).set_title

sns.scatterplot(x="total_bill", y="tip", hue="size", style="time", data=df, ax=ax[1]).set_ti

plt.savefig('scatter_plot.png') # Save the scatter plot

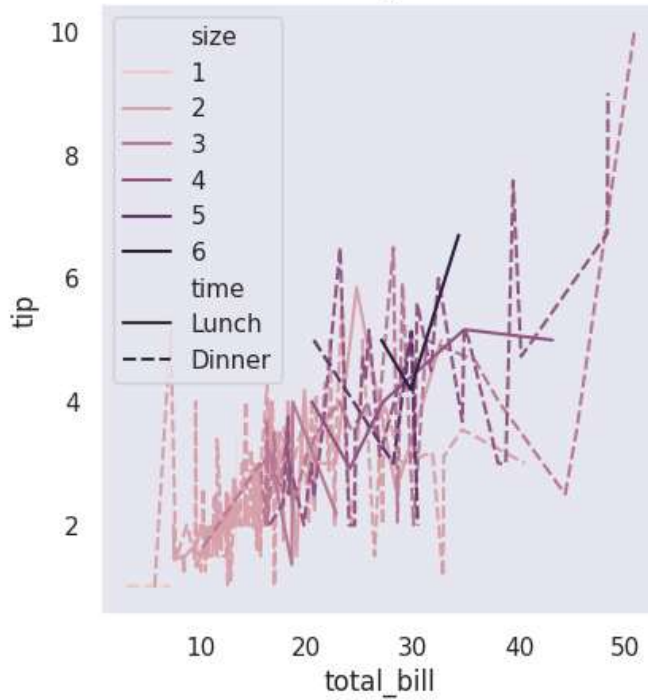
plt.show()

print("Plot saved as 'scatter_plot.png'")

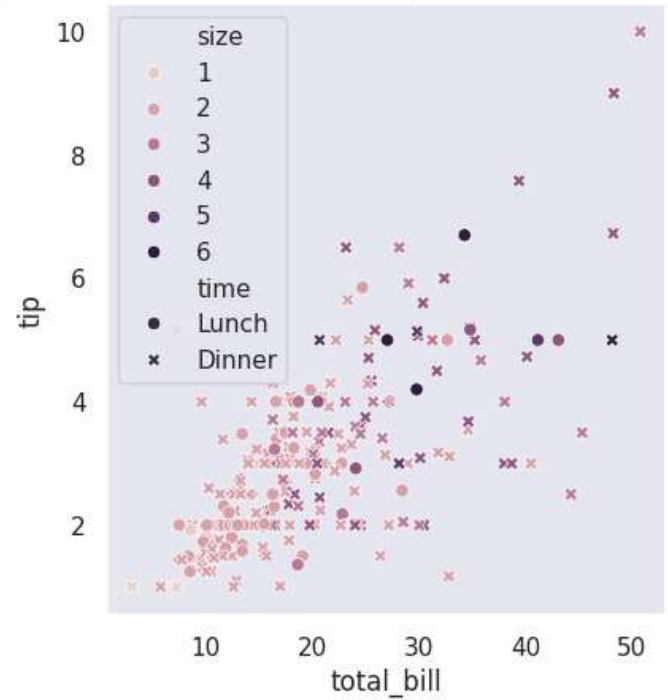


	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4

Line plot



Scatter plot



Plot saved as 'scatter plot.png'

```
sns.set_style('darkgrid')
fig,ax=plt.subplots(nrows=5,ncols=2)
fig.set_size_inches(18.5,10.5)

df=sns.load_dataset('tips')

sns.barplot(x='sex',y='total_bill',data=df,palette='plasma',estimator=np.std,ax=ax[0,0]).set
sns.countplot(x='sex',data=df,ax=ax[0,1]).set_title('count plot')

sns.boxplot(x='day',y='total_bill',data=df,hue='smoker',ax=ax[1,0]).set_title('box plot')

sns.violinplot(x='day',y='total_bill',data=df,hue='sex',split=True,ax=ax[1,1]).set_title('vi
sns.stripplot(x='day',y='total_bill',data=df,jitter=True,hue='smoker',dodge=True,ax=ax[2,0])

sns.swarmplot(x='day',y='total_bill',data=df,ax=ax[2,1]).set_title('swarm plot')

sns.violinplot(x='day',y='total_bill',data=df,ax=ax[3,0])
sns.swarmplot(x='day',y='total_bill',data=df,color='black',ax=ax[3,0]).set_title('combined p
sns.kdeplot(df[['tip','total_bill']],ax=ax[3,1])

sns.boxenplot(x='day',y='total_bill',color="b",scale="linear",data=df,ax=ax[4,0])

sns.pointplot(x='day',y='total_bill',color="b",hue="sex",data=df,ax=ax[4,1])

sns.catplot(x='day',y='total_bill',data=df,kind='bar')
```



<ipython-input-15-a96510233cae>:7: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.

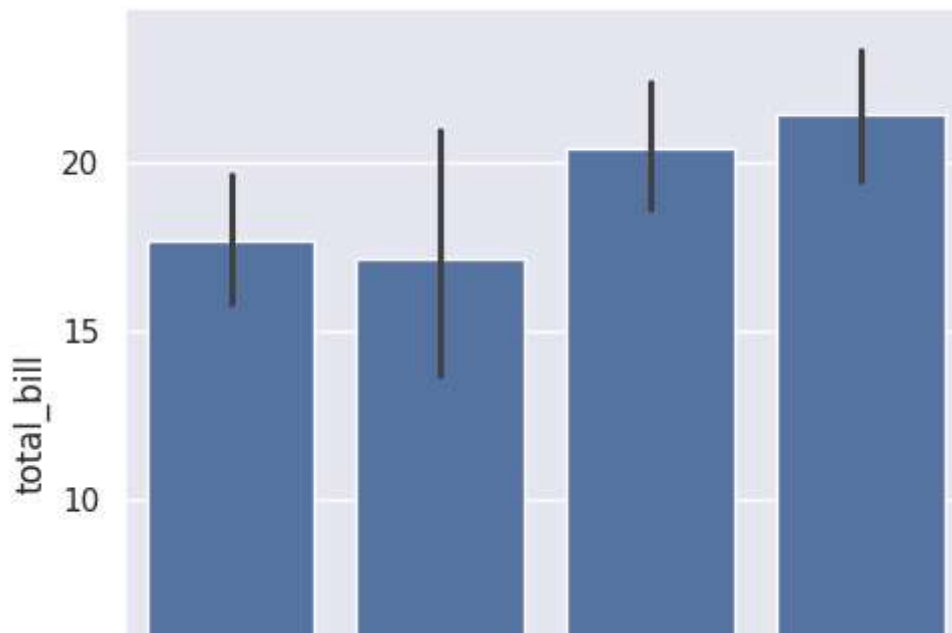
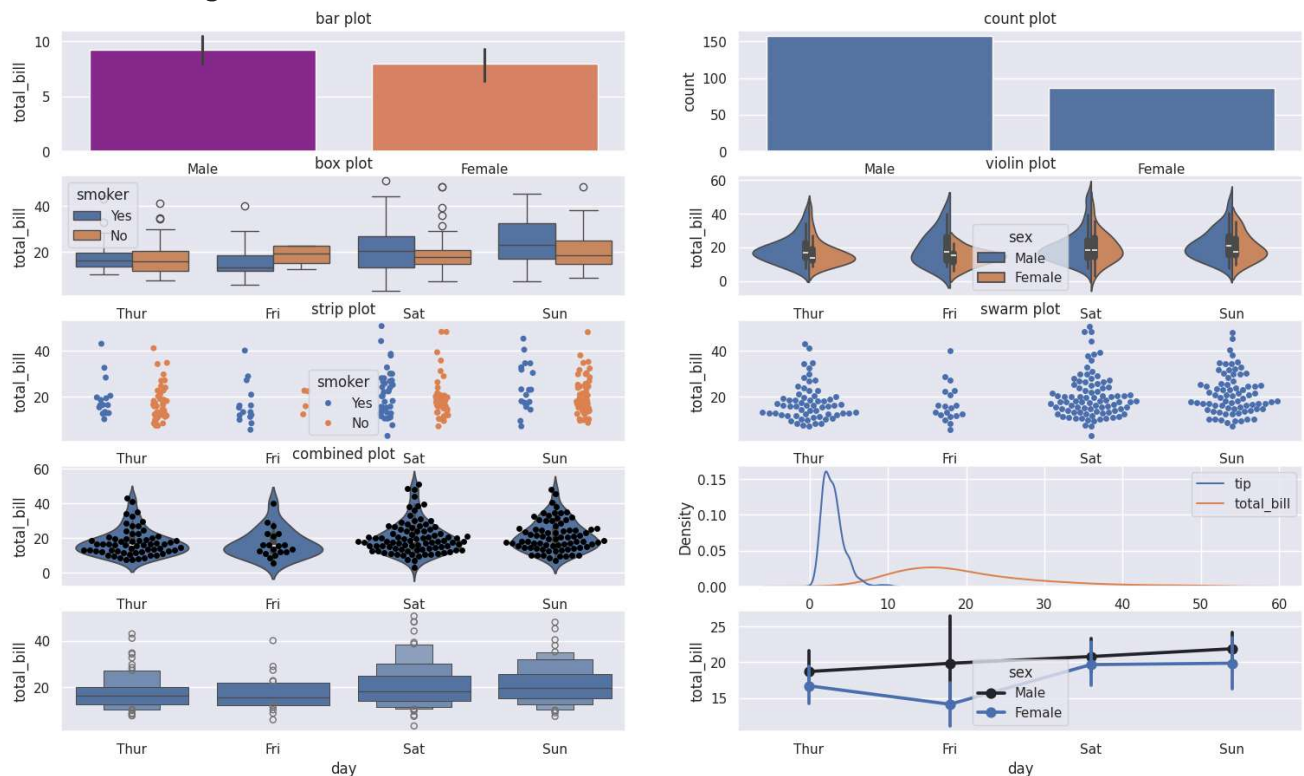
```
sns.barplot(x='sex',y='total_bill',data=df,palette='plasma',estimator=np.std,ax=ax[
<ipython-input-15-a96510233cae>:24: FutureWarning:
```

The `scale` parameter has been renamed to `width_method` and will be removed in v0.15

```
sns.boxenplot(x='day',y='total_bill',color="b",scale="linear",data=df,ax=ax[4,0])
<ipython-input-15-a96510233cae>:26: FutureWarning:
```

Setting a gradient palette using color= is deprecated and will be removed in v0.14.0.

```
sns.pointplot(x='day',y='total_bill',color="b",hue="sex",data=df,ax=ax[4,1])
<seaborn.axisgrid.FacetGrid at 0x7e7339c41de0>
```



5


```
sns.set_style("darkgrid")
fig,ax=plt.subplots(nrows=5,ncols=2)
fig.set_size_inches(18.5,10,5)

df=sns.load_dataset("tips")

#barplot

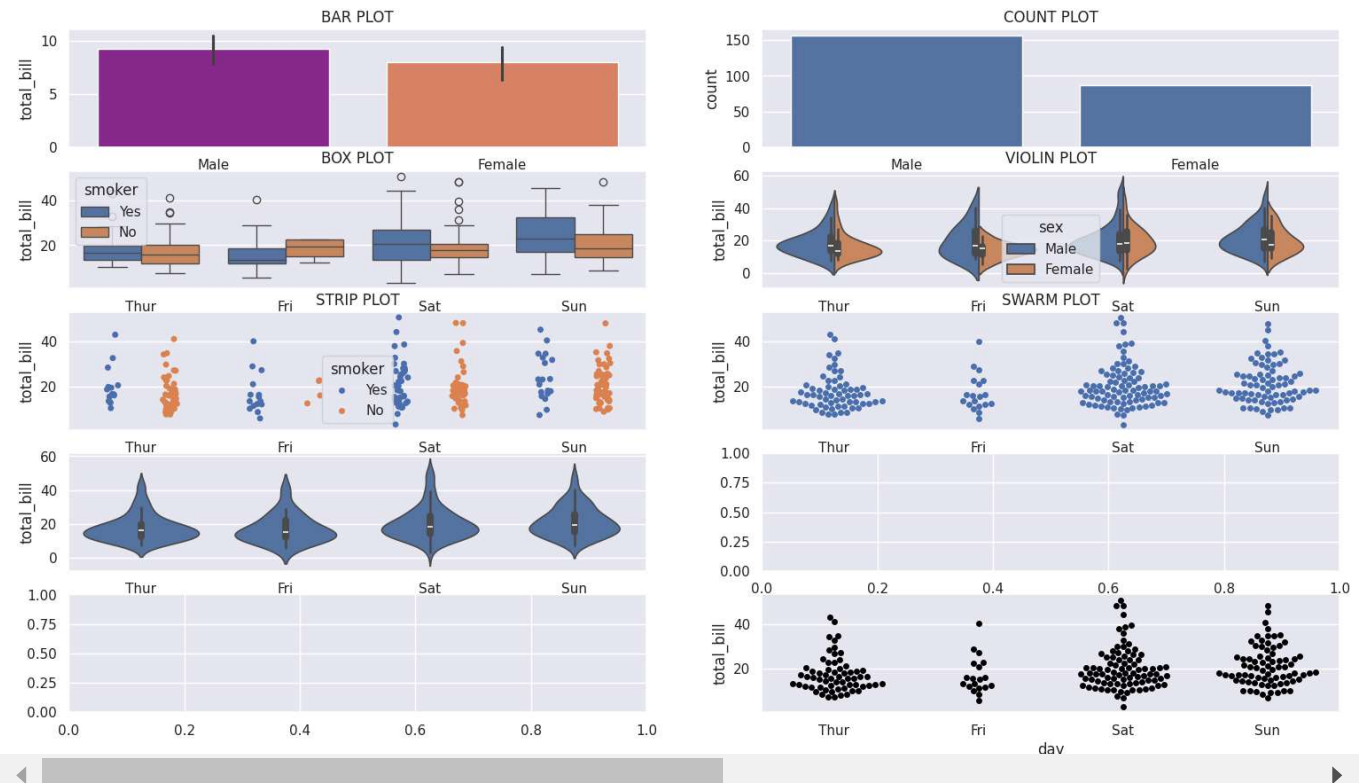
sns.barplot(x="sex",y="total_bill",data=df,palette="plasma",estimator=np.std,ax=ax[0,0]).set

#COUNTPLOT
sns.countplot(x="sex",data=df,ax=ax[0,1]).set_title("COUNT PLOT")
#BOXPLOT
sns.boxplot(x="day",y="total_bill",data=df,hue="smoker",ax=ax[1,0]).set_title("BOX PLOT")
#VIOLIN PLOT
sns.violinplot(x="day",y="total_bill",data=df,hue="sex",split=True,ax=ax[1,1]).set_title("VI
#Stripplot
sns.stripplot(x="day",y="total_bill",data=df,jitter=True, hue="smoker",dodge=True,ax=ax[2,0]
#SWARM PLOT
sns.swarmplot(x="day",y="total_bill",data=df,ax=ax[2,1]).set_title("SWARM PLOT")
#COMBINING SWARM AND VIOLIN
sns.violinplot(x="day",y="total_bill",data=df,ax=ax[3,0])
sns.swarmplot(x="day",y="total_bill",data=df,color="black")
```

 <ipython-input-9-0e1b20d111bc>:9: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0.

```
sns.barplot(x="sex",y="total_bill",data=df,palette="plasma",estimator=np.std,ax=ax[0,0])
<Axes: xlabel='day', ylabel='total_bill'>
```



distribution plots

joinplot distplot pairplot rugplot

```
import seaborn as sns
import matplotlib.pyplot as plt
```

```
sns.set_style('whitegrid')
df=sns.load_dataset('iris')
print(df.head())
```

```
sns.distplot(df['petal_length'],kde=True,color='red',bins=30).set_title('dist plot')
```



```

↗
sepal_length  sepal_width  petal_length  petal_width  species
0             5.1         3.5         1.4         0.2   setosa
1             4.9         3.0         1.4         0.2   setosa
2             4.7         3.2         1.3         0.2   setosa
3             4.6         3.1         1.5         0.2   setosa
4             5.0         3.6         1.4         0.2   setosa

```

<ipython-input-39-2012b622620a>:8: 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).

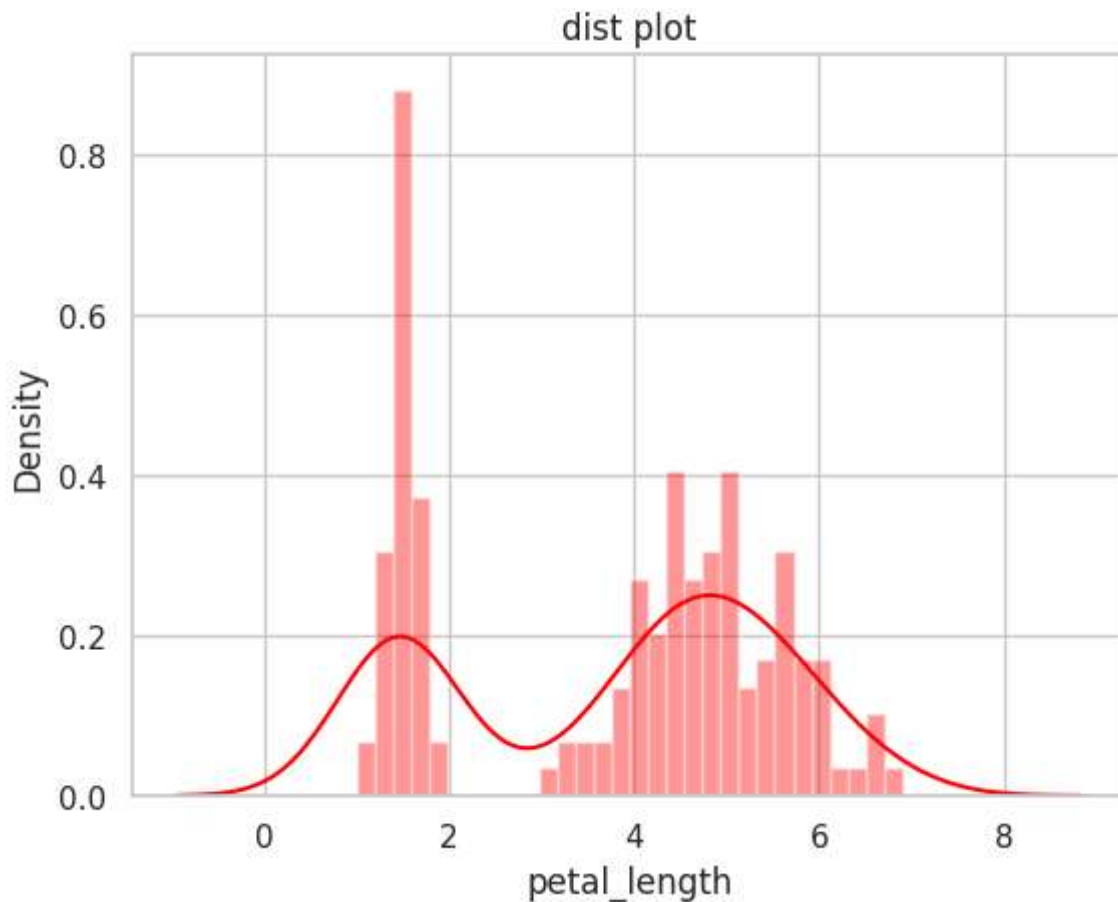
For a guide to updating your code to use the new functions, please see

<https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751>

```

sns.distplot(df['petal_length'],kde=True,color='red',bins=30).set_title('dist plot')
Text(0.5, 1.0, 'dist plot')

```



```

jointgrid=sns.JointGrid(x='petal_length',y='petal_width',data=df)
jointgrid.plot_joint(sns.scatterplot)
jointgrid.plot_marginals(sns.distplot)

```

➔ /usr/local/lib/python3.10/dist-packages/seaborn/axisgrid.py:1886: 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>

```
func(self.x, **orient_kw_x, **kwargs)
```

/usr/local/lib/python3.10/dist-packages/seaborn/axisgrid.py:1892: UserWarning:

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

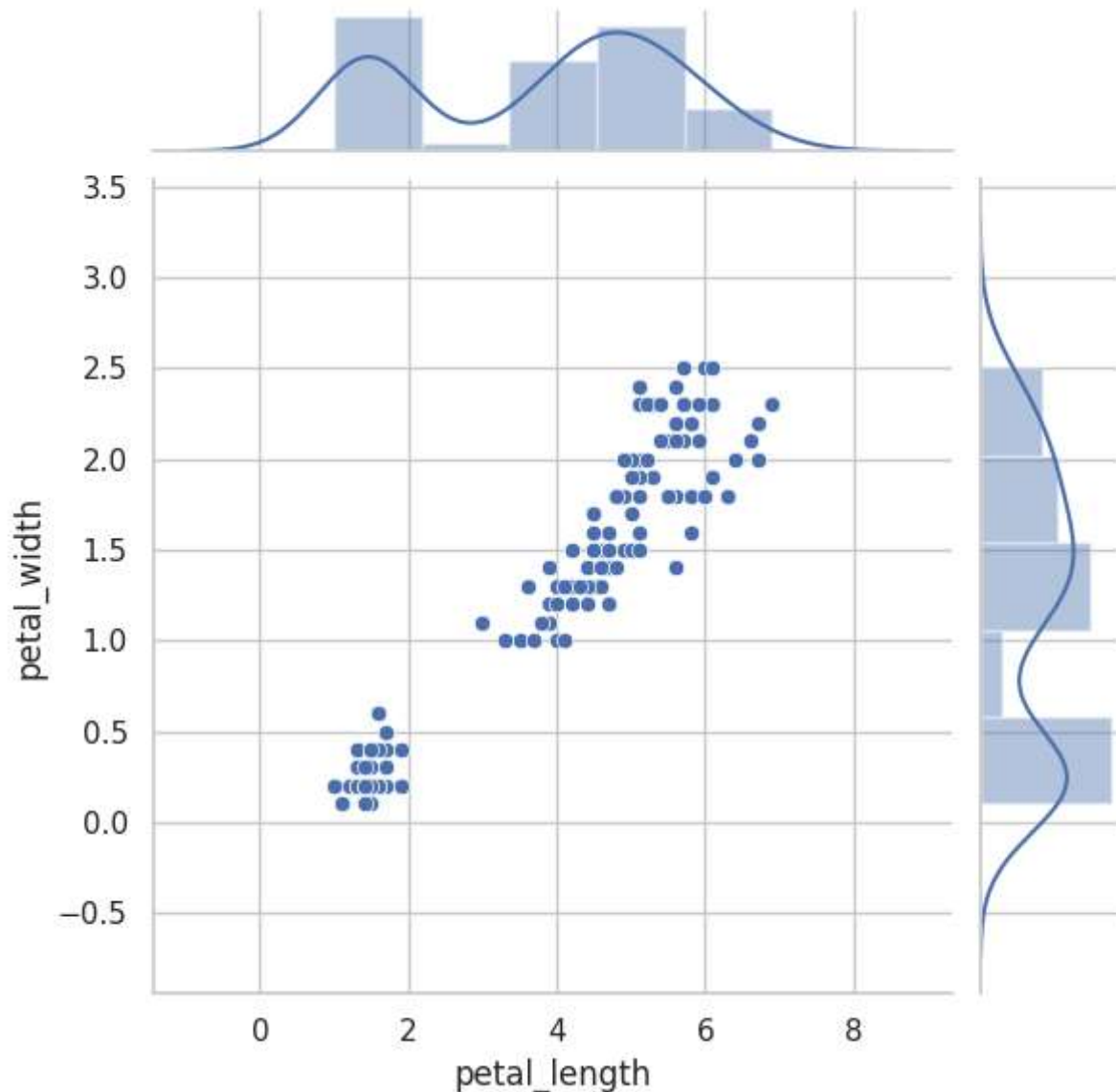
Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see

<https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751>

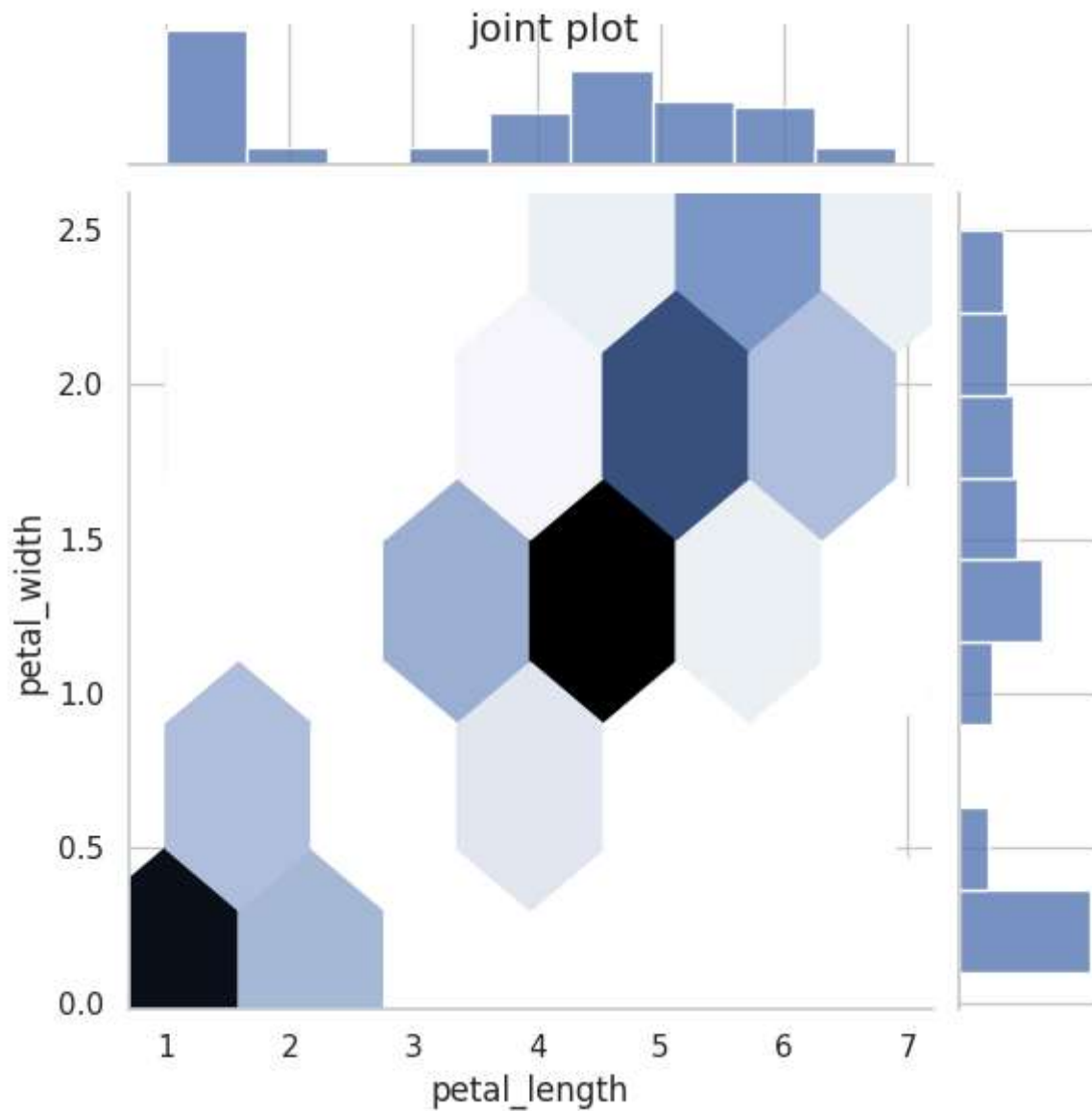
```
func(self.y, **orient_kw_y, **kwargs)
```

<seaborn.axisgrid.JointGrid at 0x7e7339912f50>




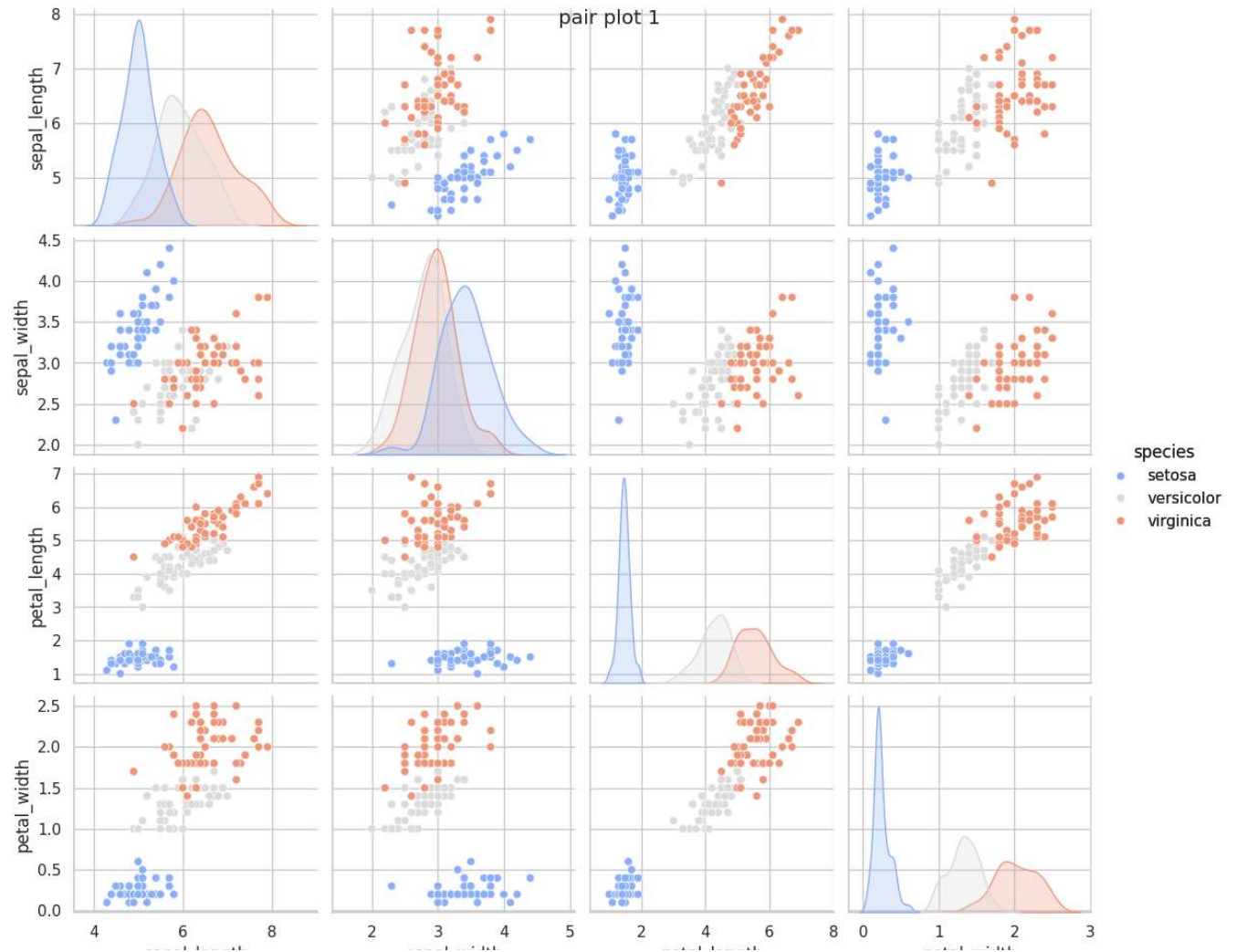
```
g=sns.jointplot(x='petal_length',y='petal_width',data=df,kind='hex')
g.fig.suptitle('joint plot')
```

```
Text(0.5, 0.98, 'joint plot')
```

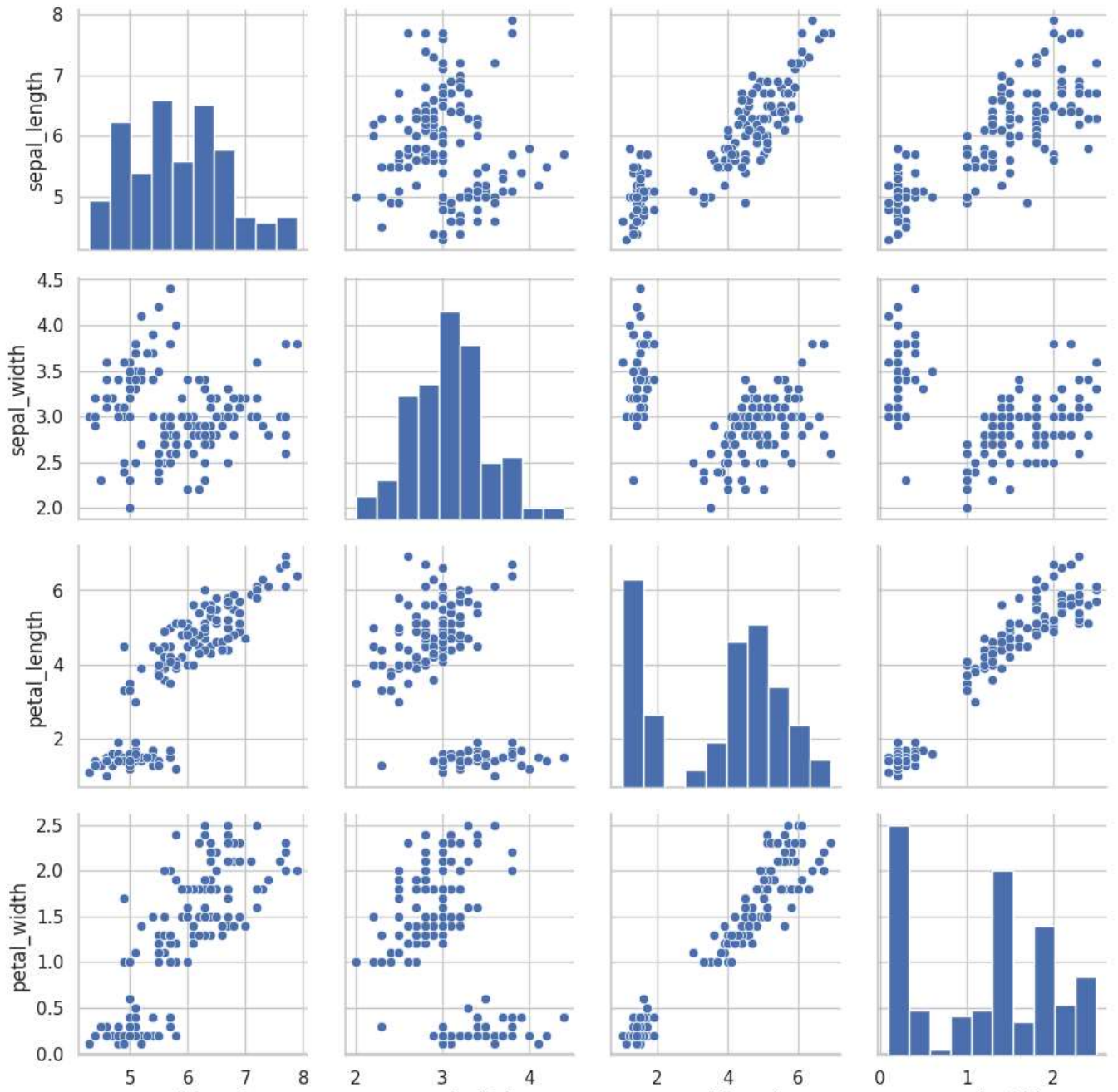


```
g=sns.pairplot(df,hue="species",palette='coolwarm')
g.fig.suptitle("pair plot 1")
g.add_legend()
```

 <seaborn.axisgrid.PairGrid at 0x7e733435b9d0>



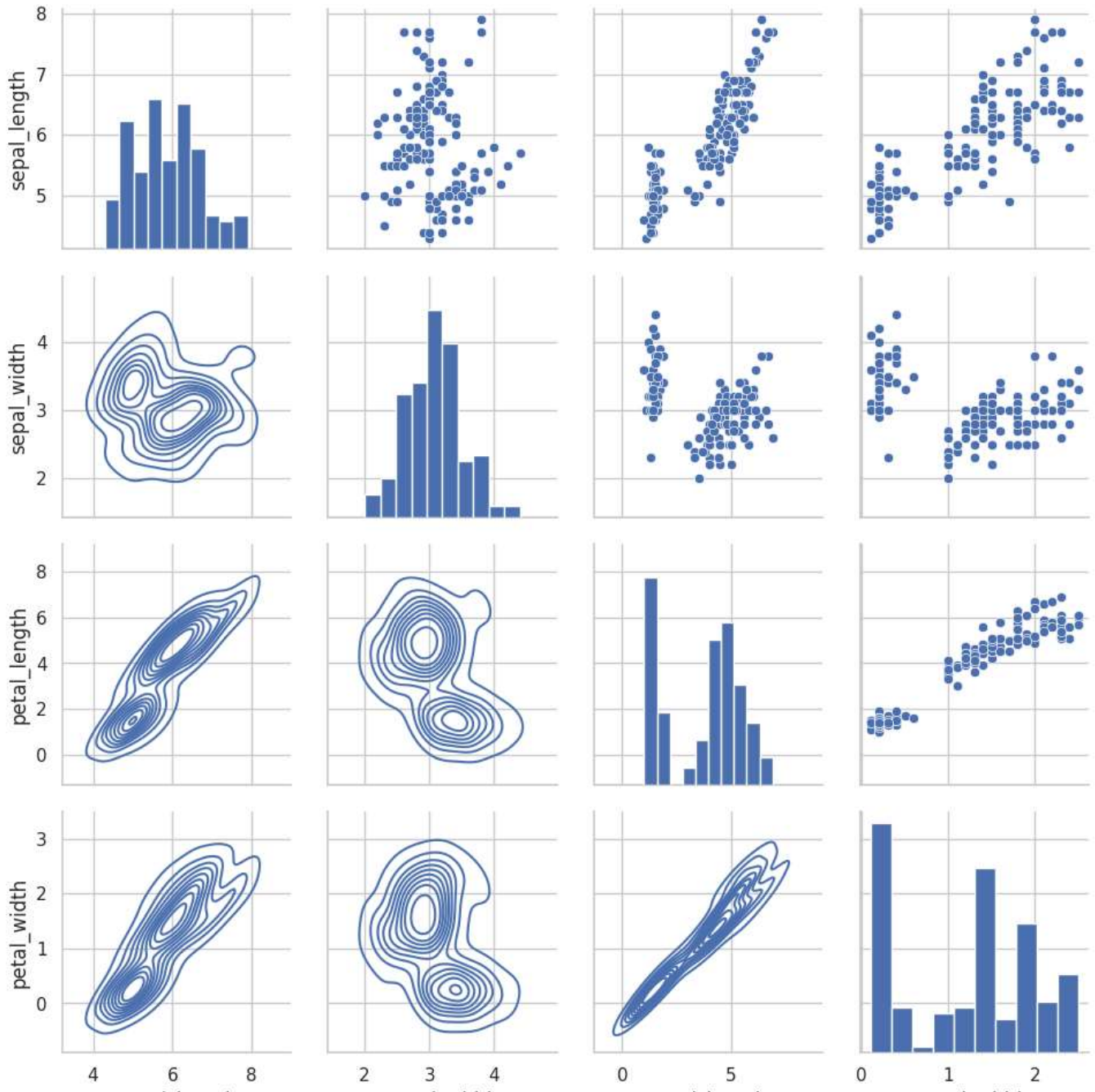
```
pairgrid = sns.PairGrid(data=df)
pairgrid = pairgrid.map_offdiag(sns.scatterplot)
pairgrid = pairgrid.map_diag(plt.hist)
```



```

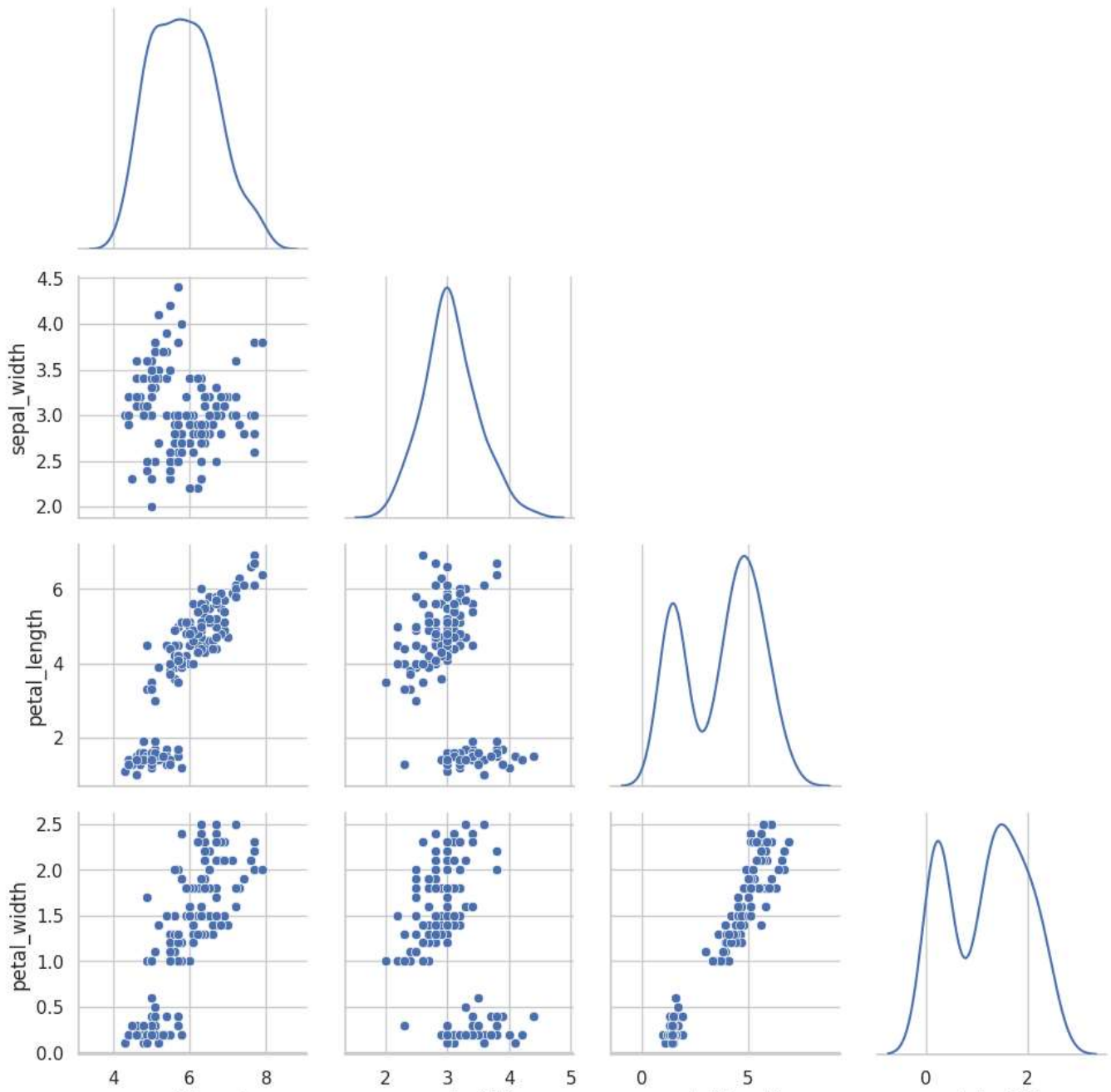
pairgrid = sns.PairGrid(data=df)
pairgrid = pairgrid.map_upper(sns.scatterplot)
pairgrid = pairgrid.map_diag(plt.hist)
pairgrid = pairgrid.map_lower(sns.kdeplot)

```



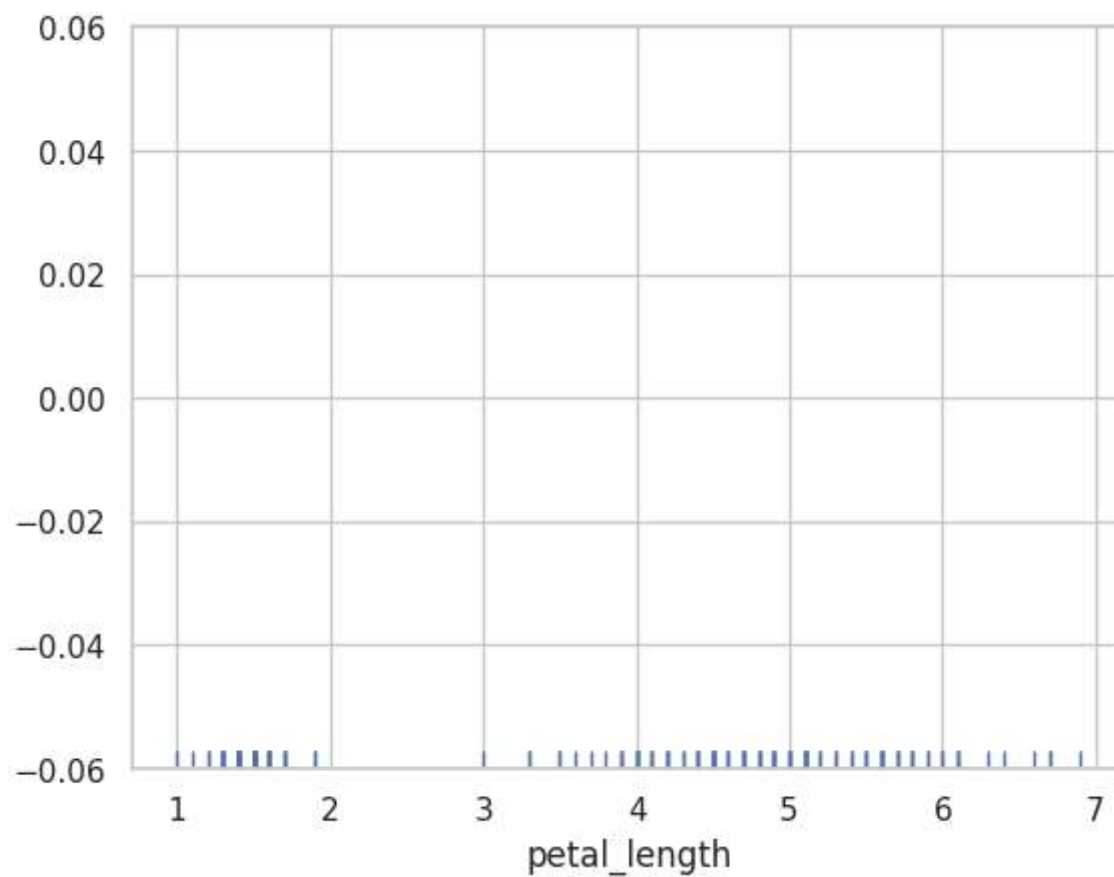

```
g = sns.PairGrid(df, diag_sharey=False, corner=True)
g.map_lower(sns.scatterplot)
g.map_diag(sns.kdeplot)
```

↗ <seaborn.axisgrid.PairGrid at 0x7e7336516440>



```
sns.rugplot(df['petal_length'])
```

↗ <Axes: xlabel='petal_length'>



matrix plot


```
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
import numpy as np

fig, ax = plt.subplots(nrows=2, ncols=2, figsize=(15, 10))

df1 = sns.load_dataset('flights')
df2 = sns.load_dataset('iris')
df11 = pd.pivot_table(values='passengers', index='month', columns='year', data=df1)

# Exclude non-numeric columns from correlation computation for both datasets
dfc1 = df1.select_dtypes(include=[np.number]).corr()
dfc2 = df2.select_dtypes(include=[np.number]).corr()

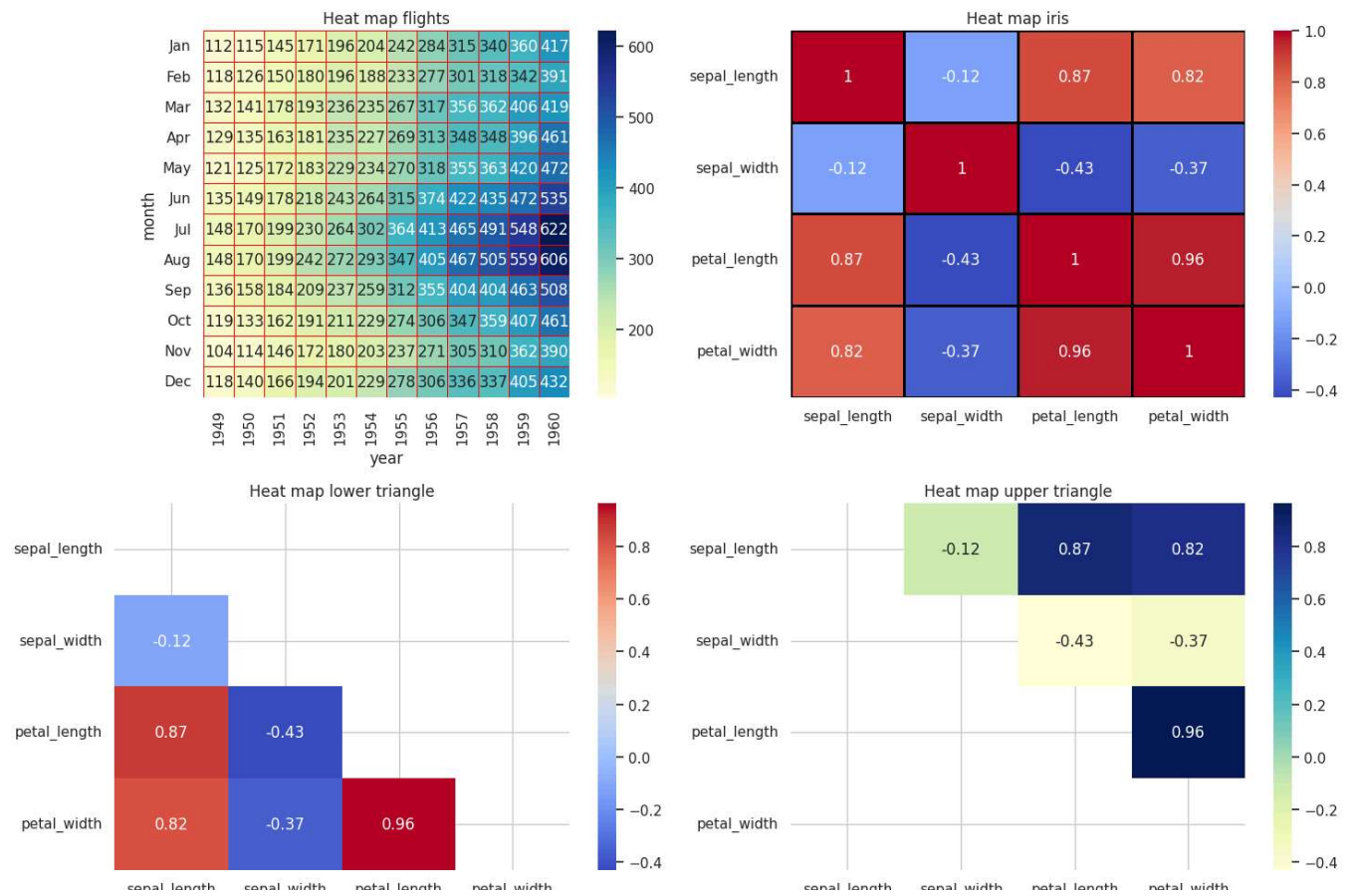
sns.heatmap(df11, cmap='YlGnBu', linecolor='red', linewidths=0.5, annot=True, fmt='d', squar

sns.heatmap(dfc2, cmap='coolwarm', linecolor='black', linewidths=1, annot=True, ax=ax[0, 1])

mask1 = np.triu(np.ones_like(dfc2, dtype=bool))
sns.heatmap(dfc2, annot=True, mask=mask1, ax=ax[1, 0], cmap='coolwarm').set_title('Heat map

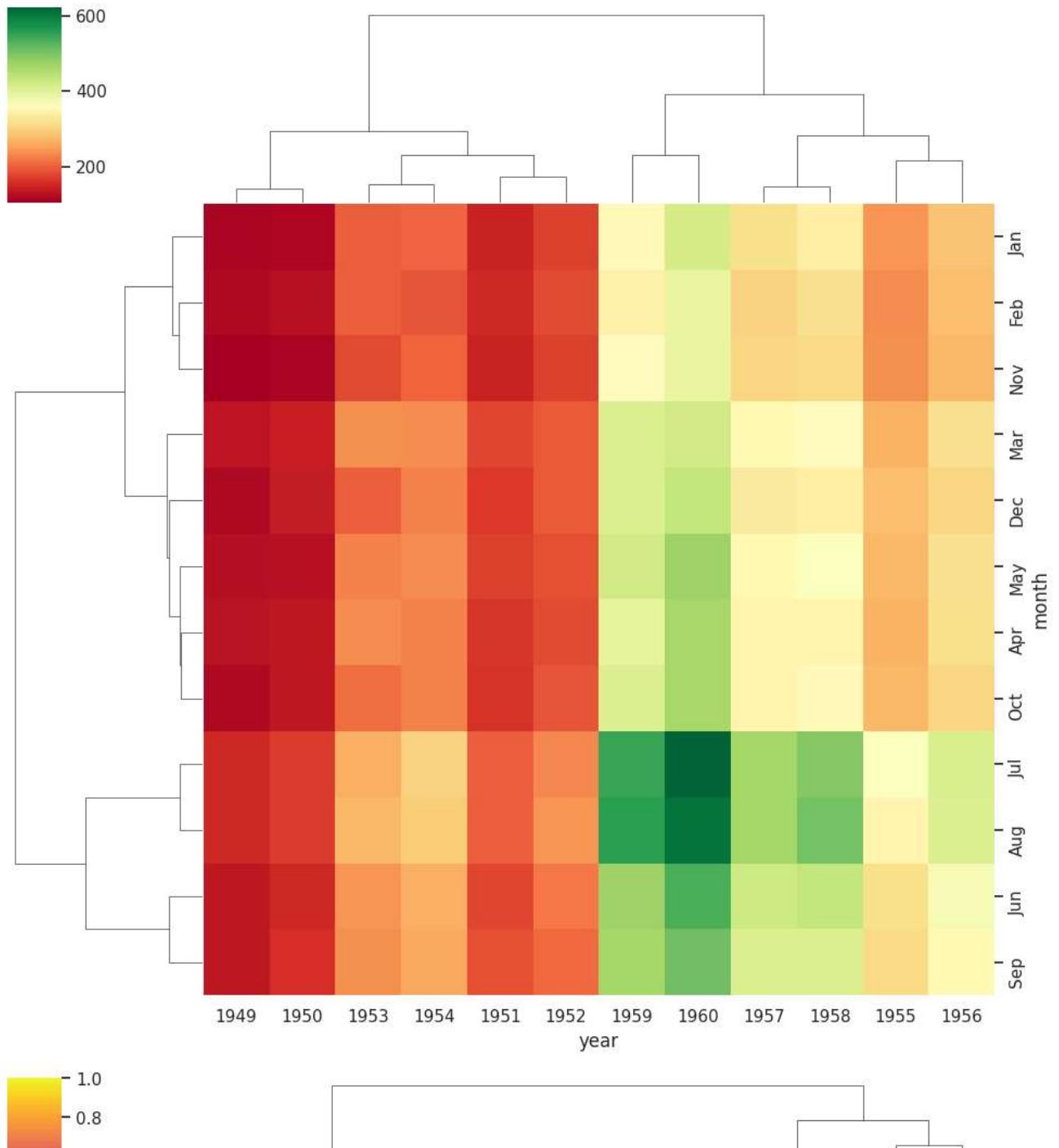
mask2 = np.tril(np.ones_like(dfc2, dtype=bool))
sns.heatmap(dfc2, annot=True, cmap='YlGnBu', mask=mask2, ax=ax[1, 1]).set_title('Heat map up

plt.tight_layout()
plt.show()
```



```
sns.clustermap(df11,cmap='RdYlGn')
sns.clustermap(df11,cmap='plasma',standard_scale=1)
```

 <seaborn.matrix.ClusterGrid at 0x7e73363915a0>



exercise(titanic Dataset)

1 plot histogram for the #of every class of passengers who