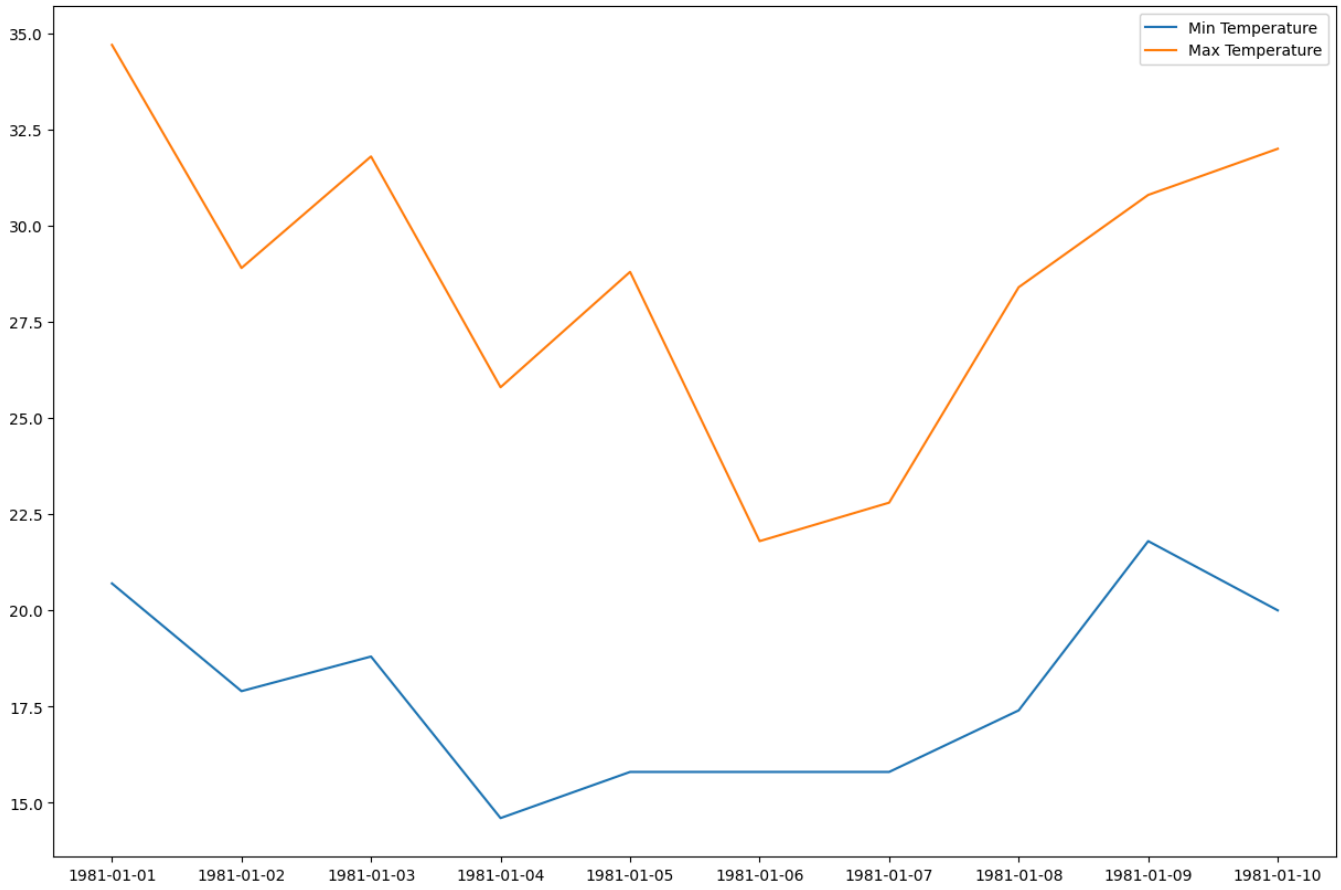
```
 import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from matplotlib.pyplot import figure
import seaborn as sns
%matplotlib inline
```

```
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', '1
min_temperature = [20.7,17.9,18.8,14.6,15.8,15.8,15.8,17.4,21.8,20.0]
max_temperature = [34.7,28.9,31.8,25.8,28.8,21.8,22.8,28.4,30.8,32.0]
fig,axes=plt.subplots(nrows=1,ncols=1,figsize=(15,10))
axes.plot(dates,min_temperature, label='Min Temperature')
axes.plot(dates,max_temperature, label='Max Temperature')
axes.legend()
```

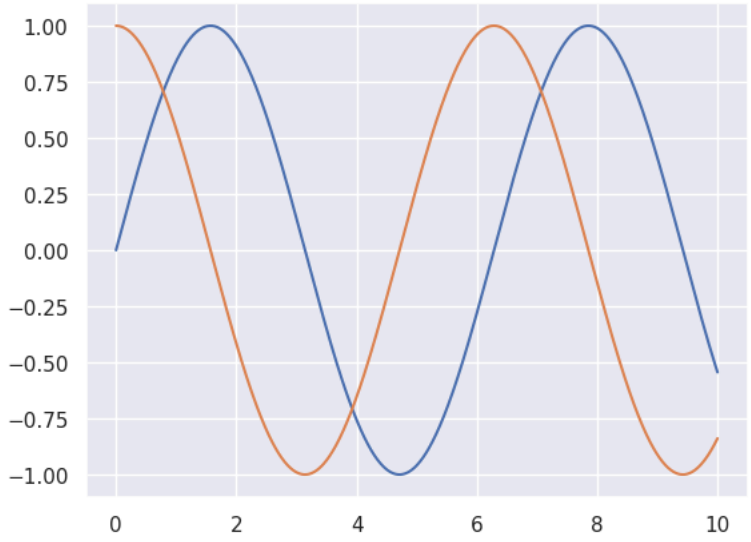
```
<matplotlib.legend.Legend at 0x7bacab2a6590>
```




```
sns.set()
```

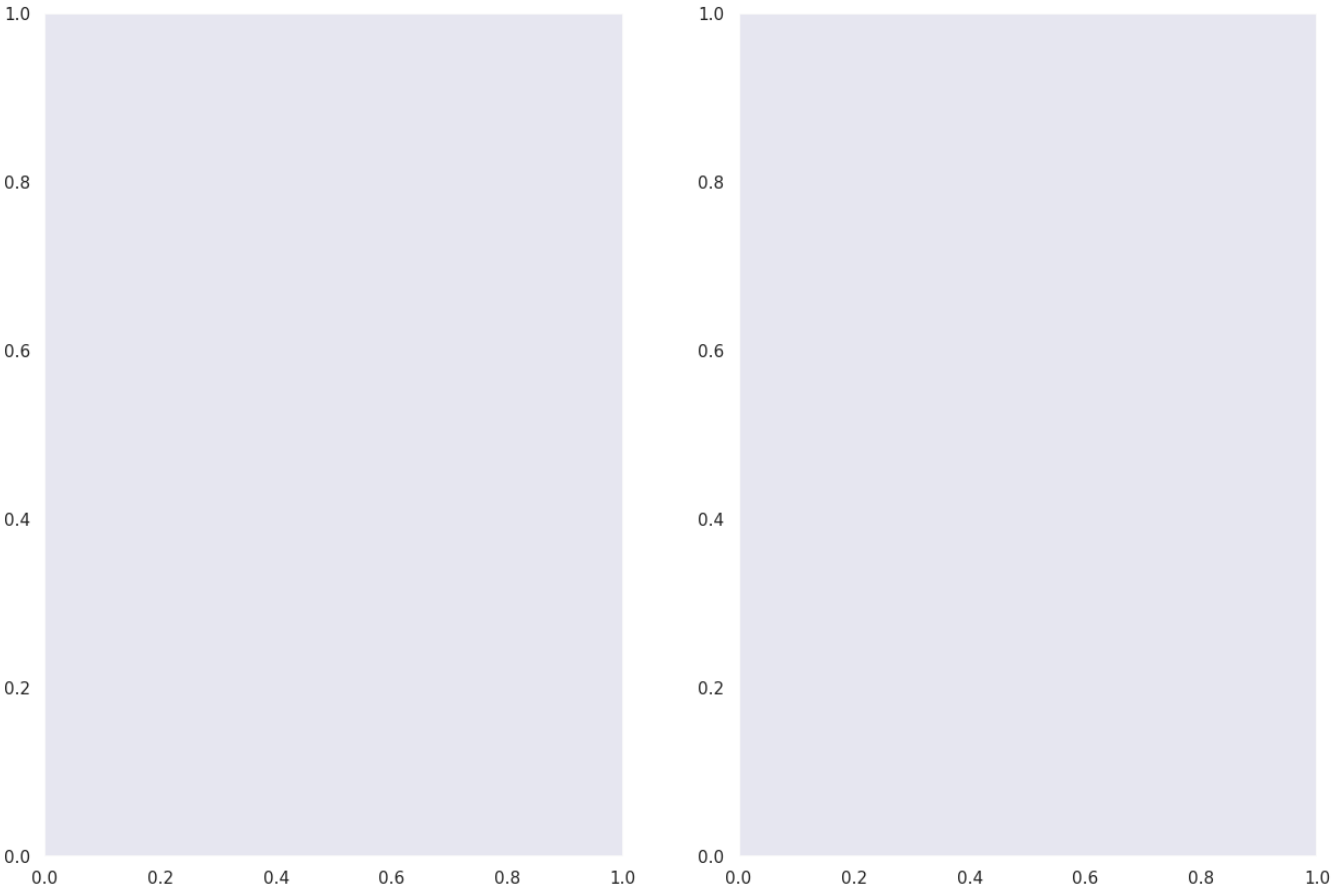
```
x = np.linspace(0,10,1000)
plt.plot(x,np.sin(x),x,np.cos(x))
```

```
[<matplotlib.lines.Line2D at 0x7baca82f7b80>,
 <matplotlib.lines.Line2D at 0x7baca82f7be0>]
```



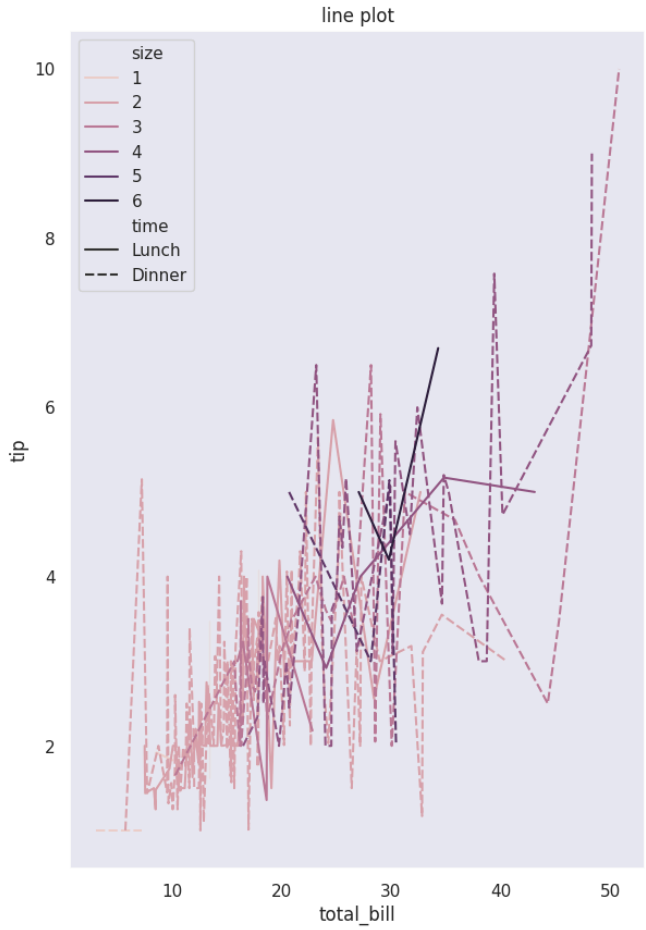

```
sns.set(style="dark")
fig,ax=plt.subplots(ncols=2,nrows=1,figsize=(15,10))
df=sns.load_dataset("tips")
print(df.head())
```

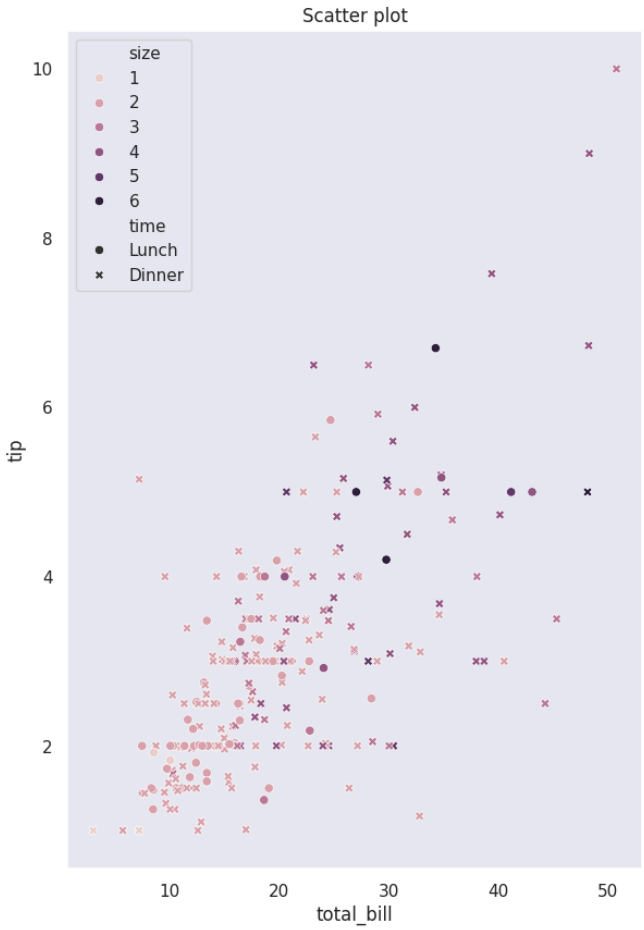
```
   total_bill   tip     sex smoker  day    time  size
0       16.99  1.01  Female     No  Sun  Dinner     2
1       10.34  1.66    Male     No  Sun  Dinner     3
2       21.01  3.50    Male     No  Sun  Dinner     3
3       23.68  3.31    Male     No  Sun  Dinner     2
4       24.59  3.61  Female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



```
sns.set(style='dark')
fig,ax=plt.subplots(ncols=2,nrows=1,figsize=(15,10))
df=sns.load_dataset('tips')
print(df.head())
#lineplot
sns.lineplot(x='total_bill',y='tip',hue='size',style='time',data=df,ax=ax[0]).set_title('line plot')
#scatterplot
Sct_plt=sns.scatterplot(x='total_bill',y='tip',hue='size',style='time',data=df,ax=ax[1]).set_title('Scatter plot')
#saving plot
Sct_plt.figure.savefig('scatter_plot1.png')
print('plot saved')
```

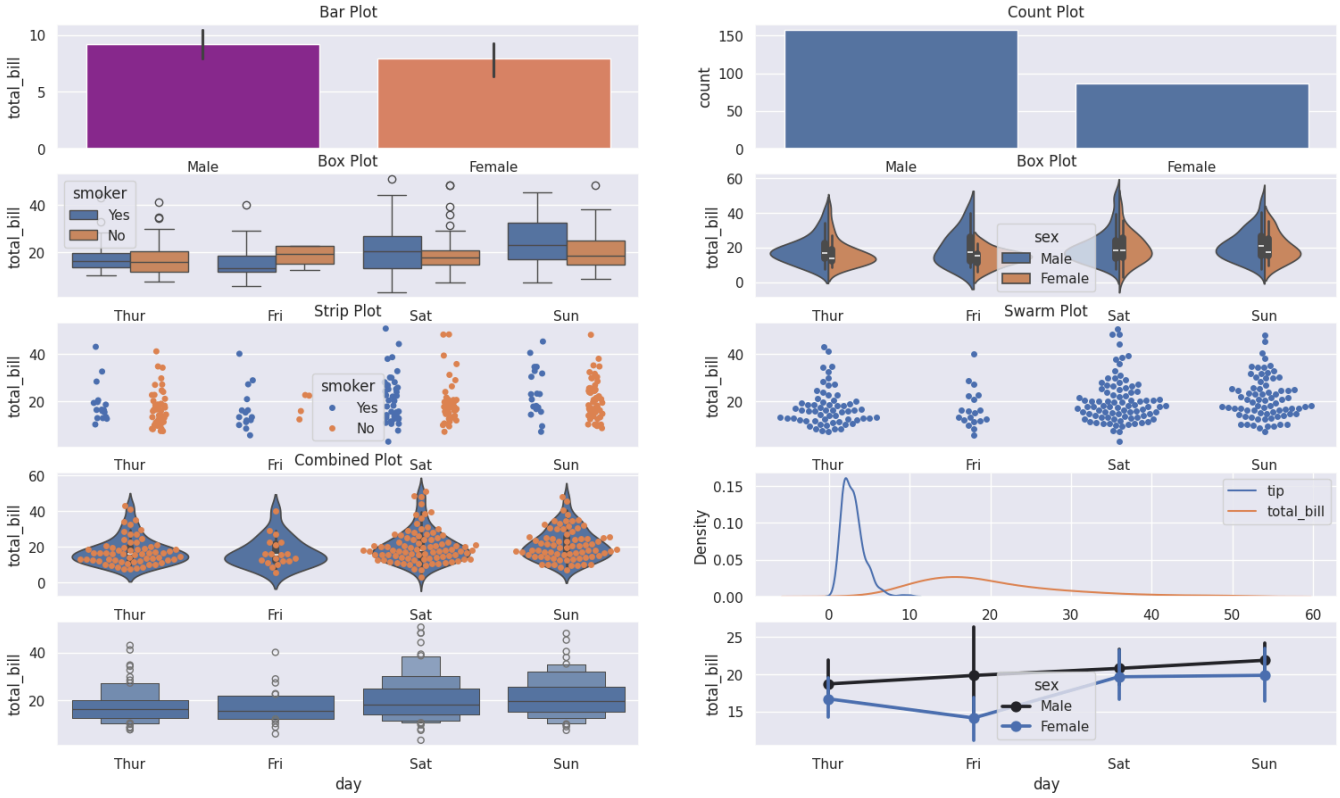
```
   total_bill   tip     sex smoker  day    time  size
0       16.99  1.01  Female     No  Sun  Dinner     2
1       10.34  1.66    Male     No  Sun  Dinner     3
2       21.01  3.50    Male     No  Sun  Dinner     3
3       23.68  3.31    Male     No  Sun  Dinner     2
4       24.59  3.61  Female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





```
#Categorical plots

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_title('Bar Plot')
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('Box Plot')
sns.stripplot(x='day',y='total_bill',data=df,jitter=True,hue='smoker',dodge=True,ax=ax[2,0]).set_title('Strip Plot')
sns.swarmplot(x='day',y='total_bill',data=df,ax=ax[2,1]).set_title('Swarm Plot')
#Combining violin and swarmplot
sns.violinplot(x='day',y='total_bill',data=df,ax=ax[3,0])
sns.swarmplot(x='day',y='total_bill',data=df,ax=ax[3,0]).set_title('Combined Plot')
#Density plot
sns.kdeplot(data=df[['tip','total_bill']],ax=ax[3,1])
#boxenplot
sns.boxenplot(x="day",y="total_bill",color="b",scale="linear",data=df,ax=ax[4,0])
#RidgePlot
sns.pointplot(x="day",y="total_bill",color="b",hue="sex",data=df,ax=ax[4,1])
#Catplot or genral plot provides a parameter called 'kind' to choose
```

```
<ipython-input-8-747d3d3f7022>:7: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `le

  sns.barplot(x='sex',y='total_bill',data=df,palette='plasma',estimator=np.std,ax=ax[0,0]).set_title('Bar Plot')
<ipython-input-8-747d3d3f7022>:19: FutureWarning:

The `scale` parameter has been renamed to `width_method` and will be removed in v0.15. Pass `width_method='linear' for the same effe
  sns.boxenplot(x="day",y="total_bill",color="b",scale="linear",data=df,ax=ax[4,0])
<ipython-input-8-747d3d3f7022>:21: FutureWarning:

Setting a gradient palette using color= is deprecated and will be removed in v0.14.0. Set `palette='dark:b'` for the same effect.

  sns.pointplot(x="day",y="total_bill",color="b",hue="sex",data=df,ax=ax[4,1])
<Axes: xlabel='day', ylabel='total_bill'>
/usr/local/lib/python3.10/dist-packages/IPython/core/pylabtools.py:151: UserWarning: Creating legend with loc="best" can be slow wit
  fig.canvas.print_figure(bytes_io, **kw)
```



Distribution Plots in seaborn it is used for examining univariate and bivariate distributions.

There are mainly 4 types of plots

*Joint plot

*Distplot

*Pairplot

*Rugplot

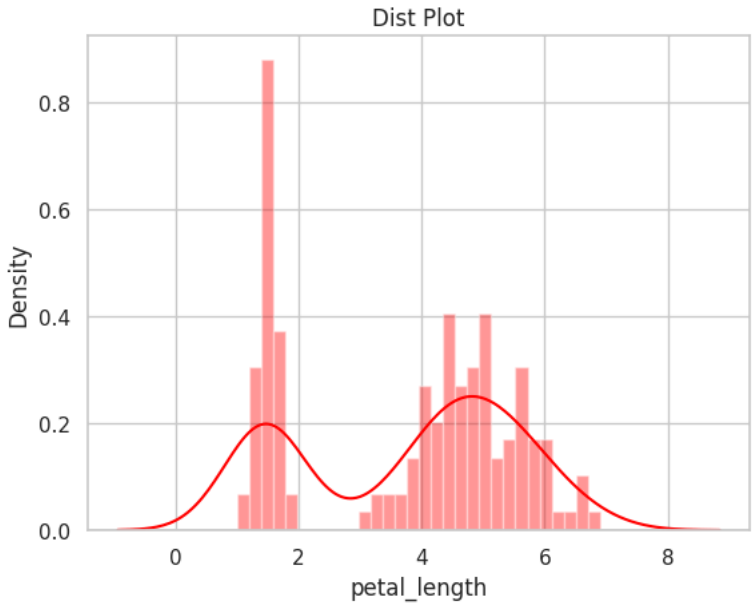
```
df=sns.load_dataset('iris')
```

```
sns.set_style('whitegrid')
print(df.head())
```

```
   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
```

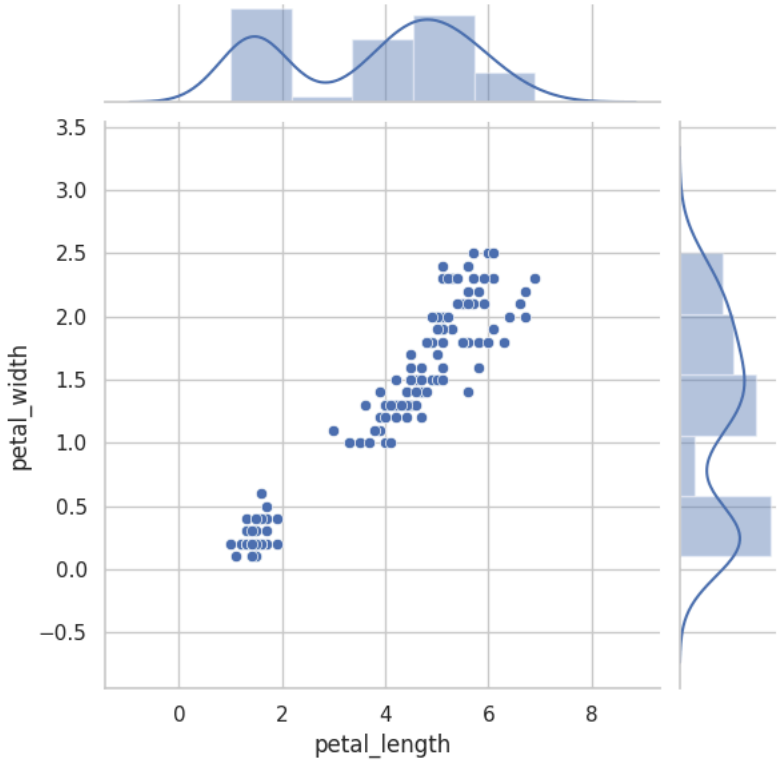
```
sns.distplot(df['petal_length'],kde=True,color='red',bins=30).set_title('Dist Plot')
```

```
<ipython-input-11-d9311dab8e33>: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

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



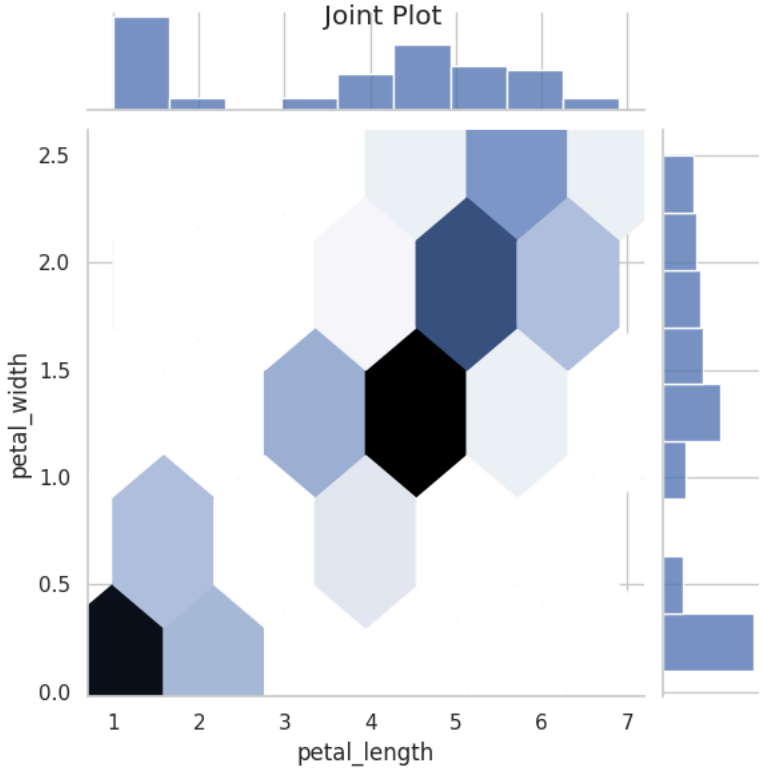

```
jointgrid =sns.JointGrid(x='petal_length',y='petal_width',data=df)
jointgrid.plot_joint(sns.scatterplot)
jointgrid.plot_marginals(sns.distplot)
```

```
/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 0x7bacaaeee080>
```



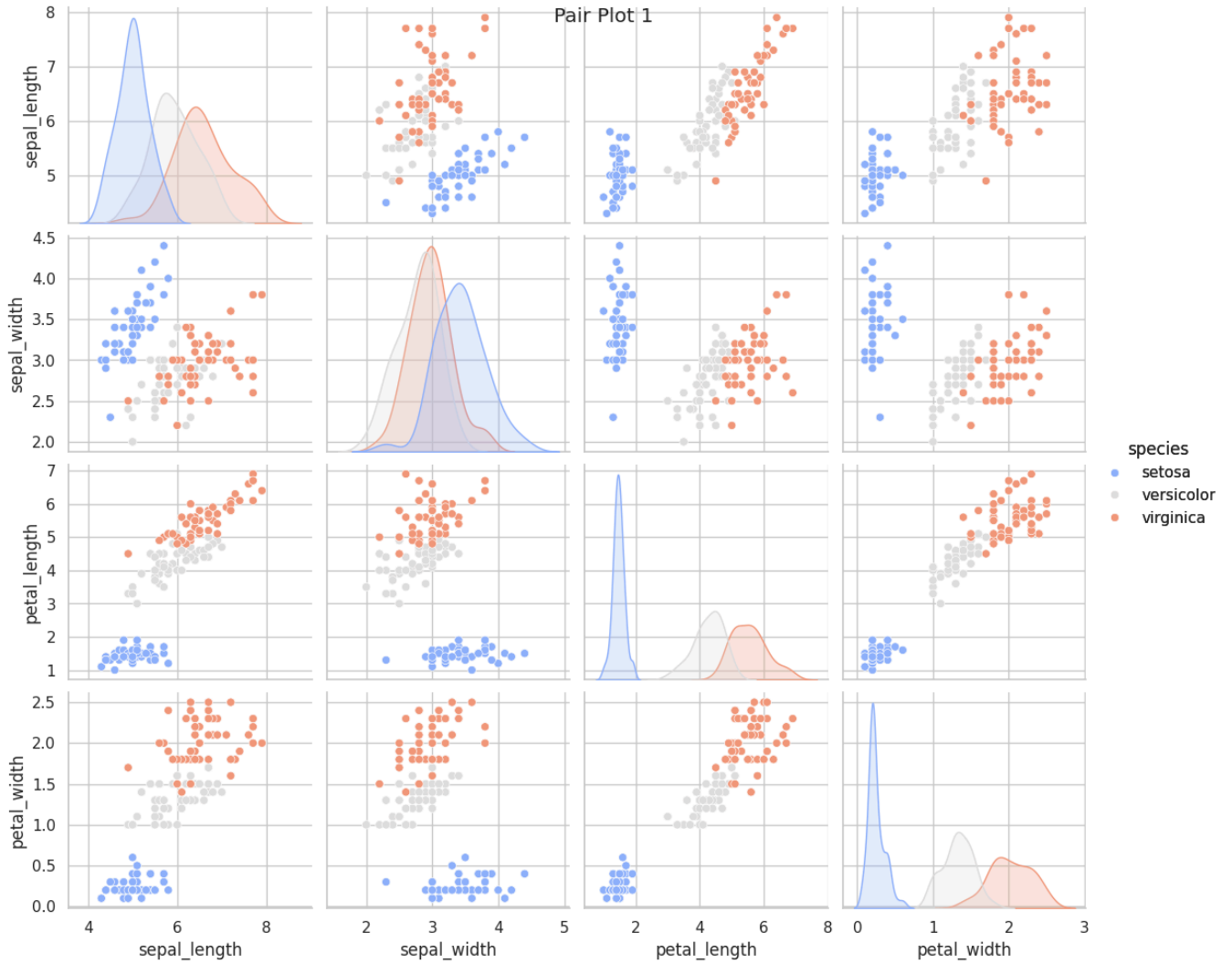

```
g=sns.jointplot(x ='petal_length',y='petal_width',data=df,kind='hex')
g.fig.suptitle('Joint Plot')
```

```
Text(0.5, 0.98, 'Joint Plot')
```

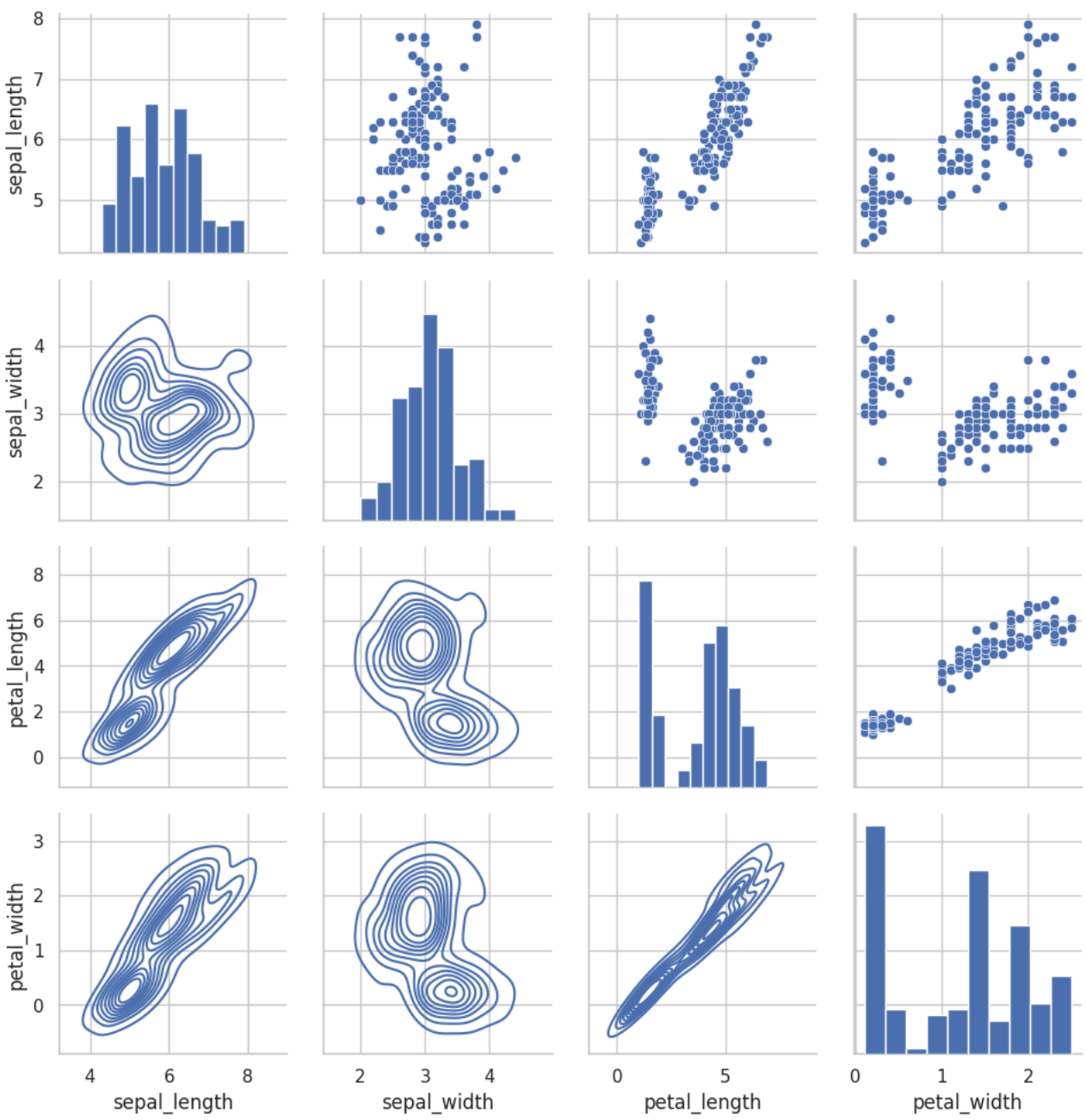



```
g=sns.pairplot(df,hue='species',palette='coolwarm')
g.fig.suptitle('Pair Plot 1')
g.add_legend()
```

```
<seaborn.axisgrid.PairGrid at 0x7baca3f81c90>
```

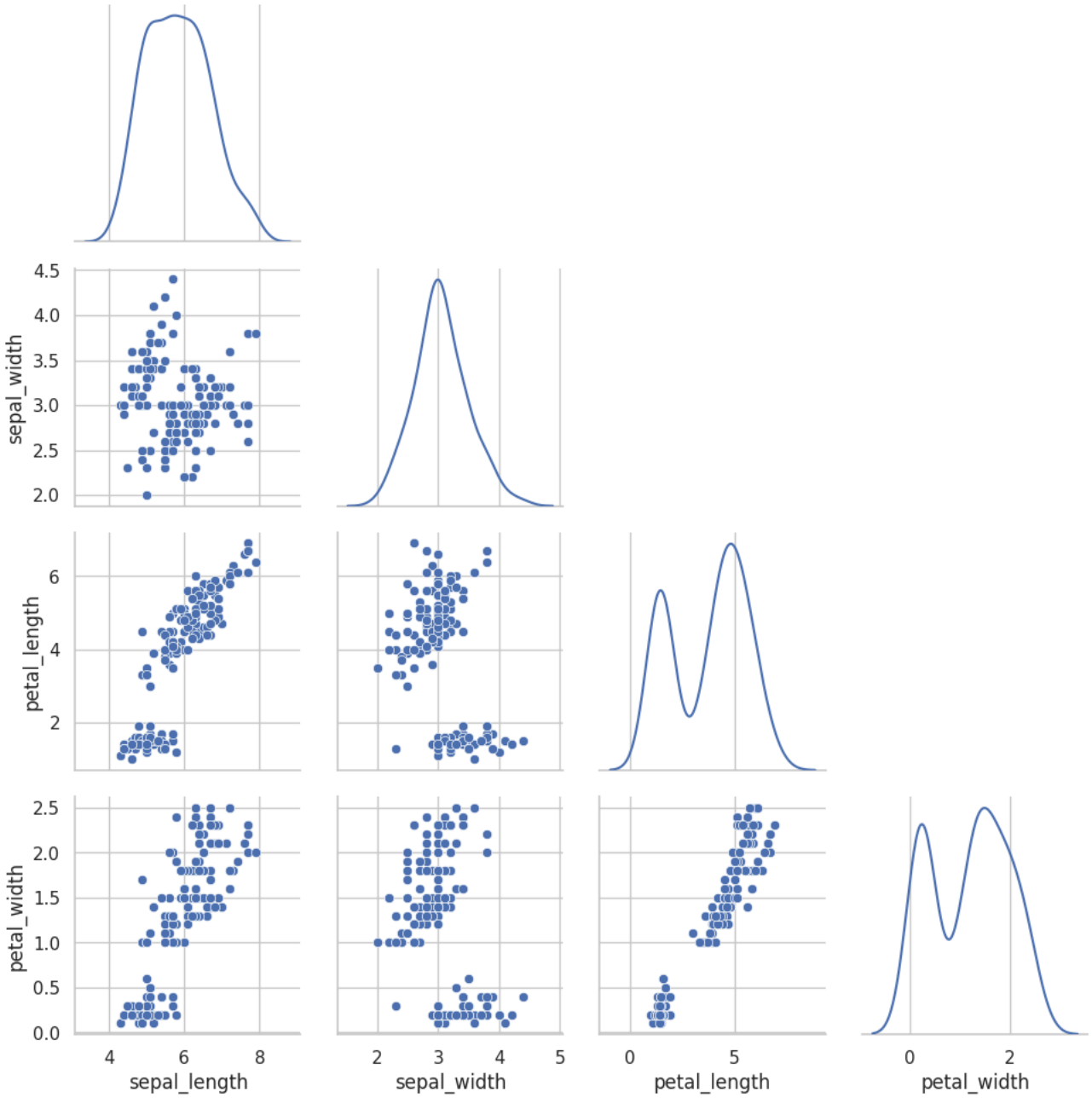



```
pairgrid = sns.PairGrid(data=df)
pairgrid = pairgrid.map_upper(sns.scatterplot)
pairgrid = pairgrid.map_diag(plt.hist)
pairgrid = pairgrid.map_lower(sns.kdeplot)
```




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

```
<seaborn.axisgrid.PairGrid at 0x7baca2a346d0>
```




```
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)
k=df1.drop('month',axis=1)
l=df2.drop('species',axis=1)
dfc1=k.corr()
dfc2=l.corr()
sns.heatmap(df11,cmap='viridis',linecolor='r',linewidth=0.5,annot=True,fmt='d',square=True,ax=ax[0,0]).set_title('Heat Map Flights')
sns.heatmap(dfc2,cmap='coolwarm',linecolor='black',linewidth=1,annot=True,ax=ax[0,1]).set_title('Heat Map iris')
mask1=np.triu(dfc2)
sns.heatmap(dfc2,annot=True,mask=mask1,ax=ax[1,0],cmap='coolwarm').set_title('Heat Map Lower Traingle')
mask2=np.tril(dfc2)
sns.heatmap(dfc2,annot=True,cmap='viridis',mask=mask2,ax=ax[1,1]).set_title('Heat Upper Triangle')
sns.clustermap(df11,cmap='plasma',standard_scale=1)
```

```
<seaborn.matrix.ClusterGrid at 0x7baca2bc5d20>
```