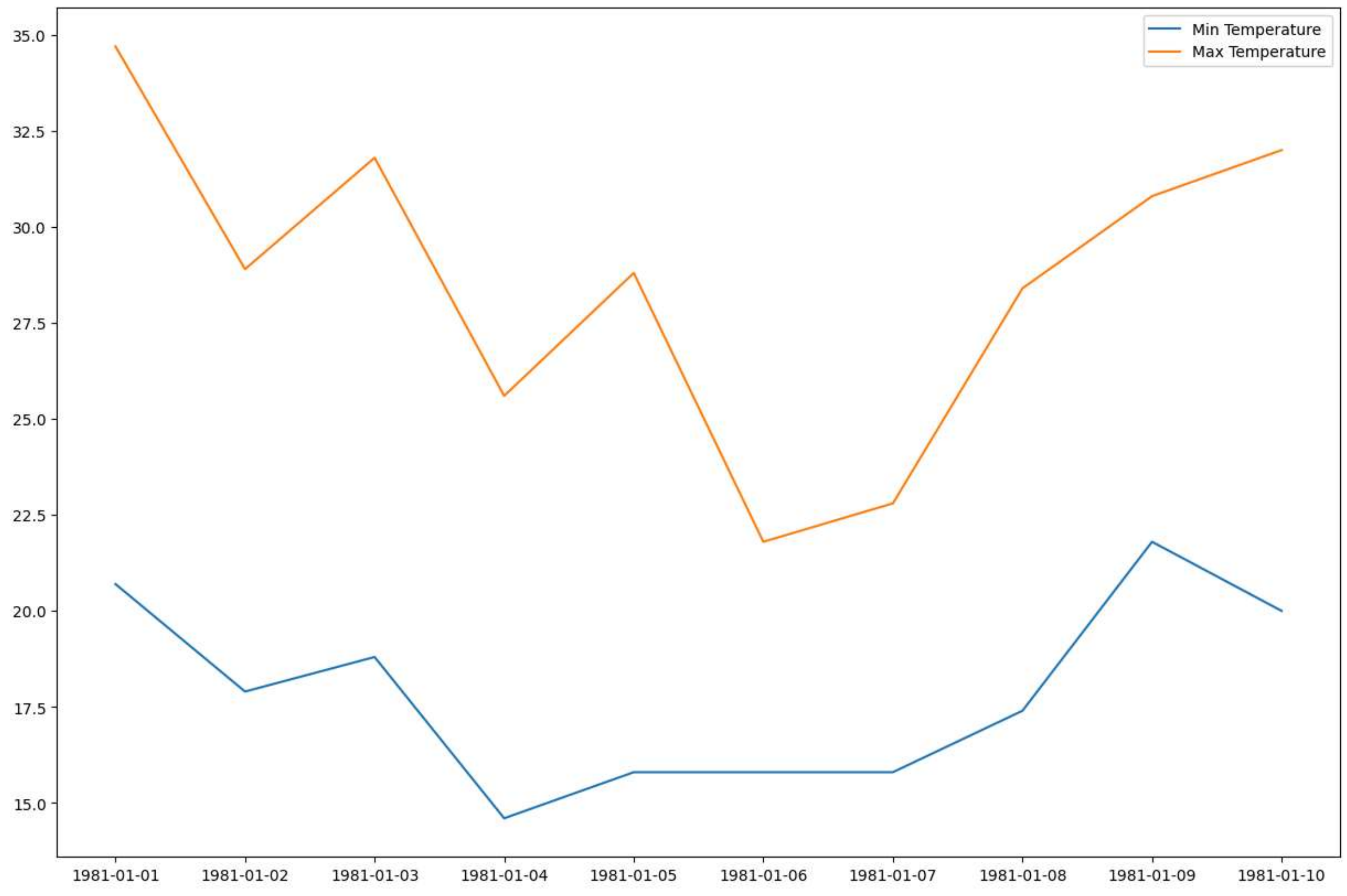
1. Relational plots: this plot is used to understand the relation between two variables.
2. Categorical plots: this plot deals with categorical variables and how they can be visualised.
3. Distribution plots: this plot is used to examine univariate and bivariate distributuions.
4. Matrix plot: A matrix plot is an array of scatterplot.
5. Regression plot: the regression plots in seaborn are primarily intended to add a visual guide that helps to emphasize patterns in a dataset during exploratory data analysis.

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

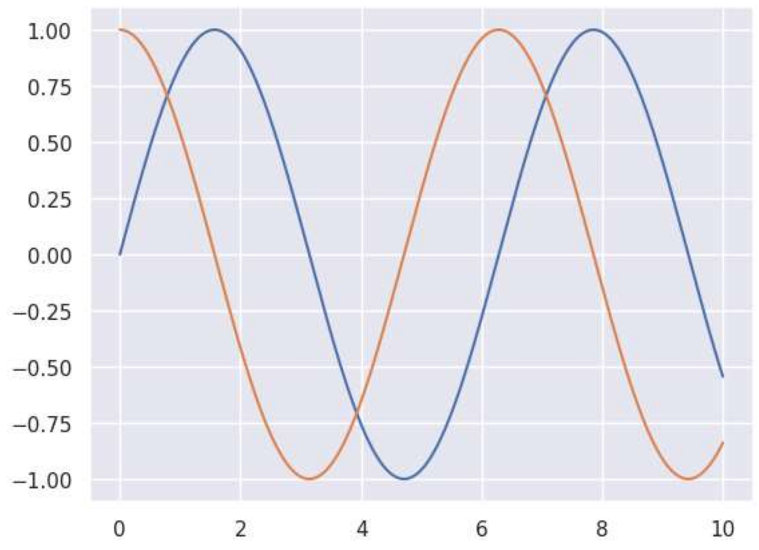
```
#Simple plotting with Seaborn #Data
dates=['1981-01-01','1981-01-02','1981-01-03','1981-01-04','1981-01-05','1981-01-06','1981-01-07','1981-01-08','1981-01-09','1981-01-10']
min_temperature =[20.7,17.9,18.8,14.6,15.8,15.8,15.8,17.4,21.8,20.0]
max_temperature =[34.7,28.9,31.8,25.6,28.8,21.8,22.8,28.4,30.8,32.0]
#plotting
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 0x7ad5123be6e0>
```



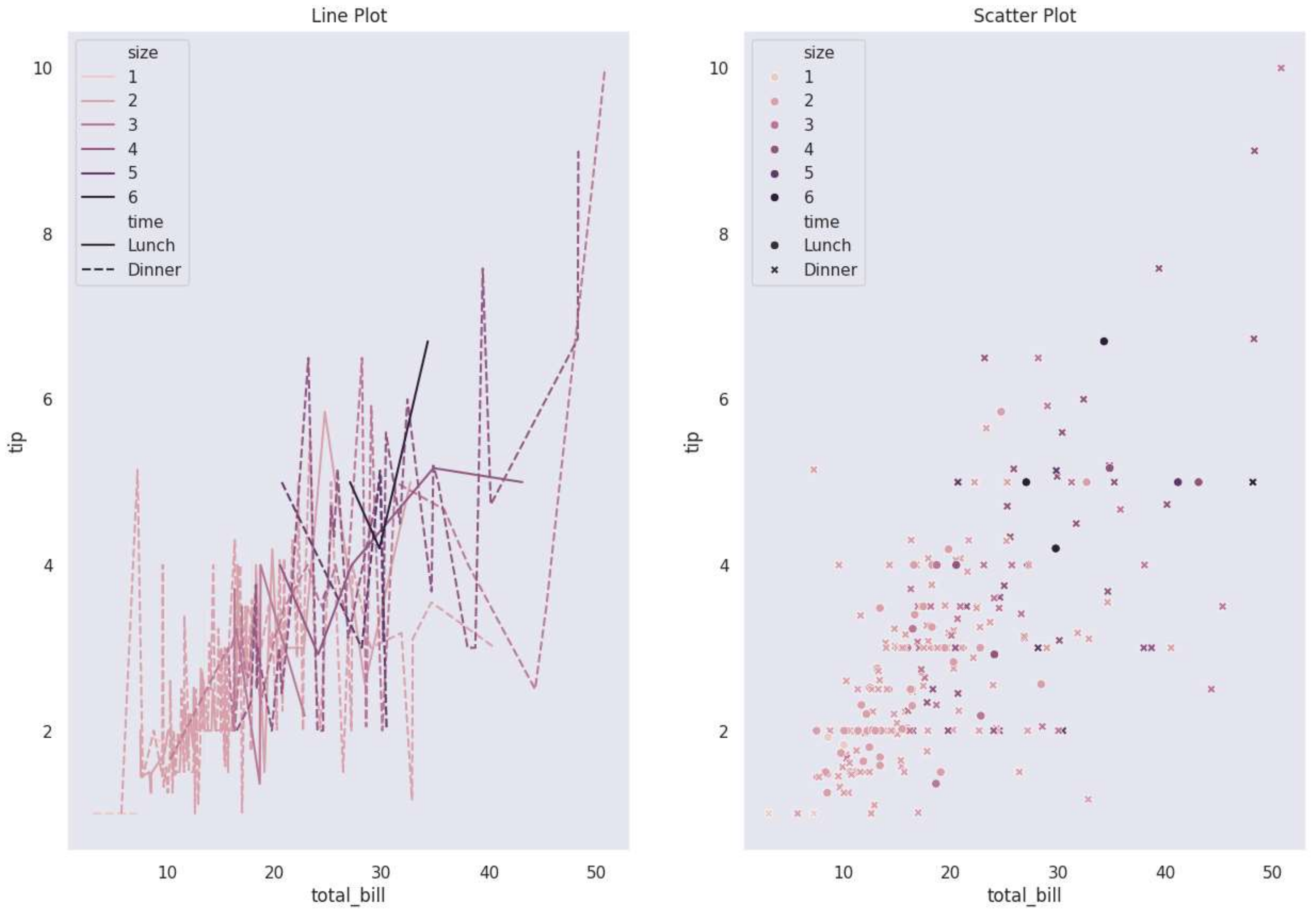
```
#seaborn style as default matplotlib style
sns.set()
```

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



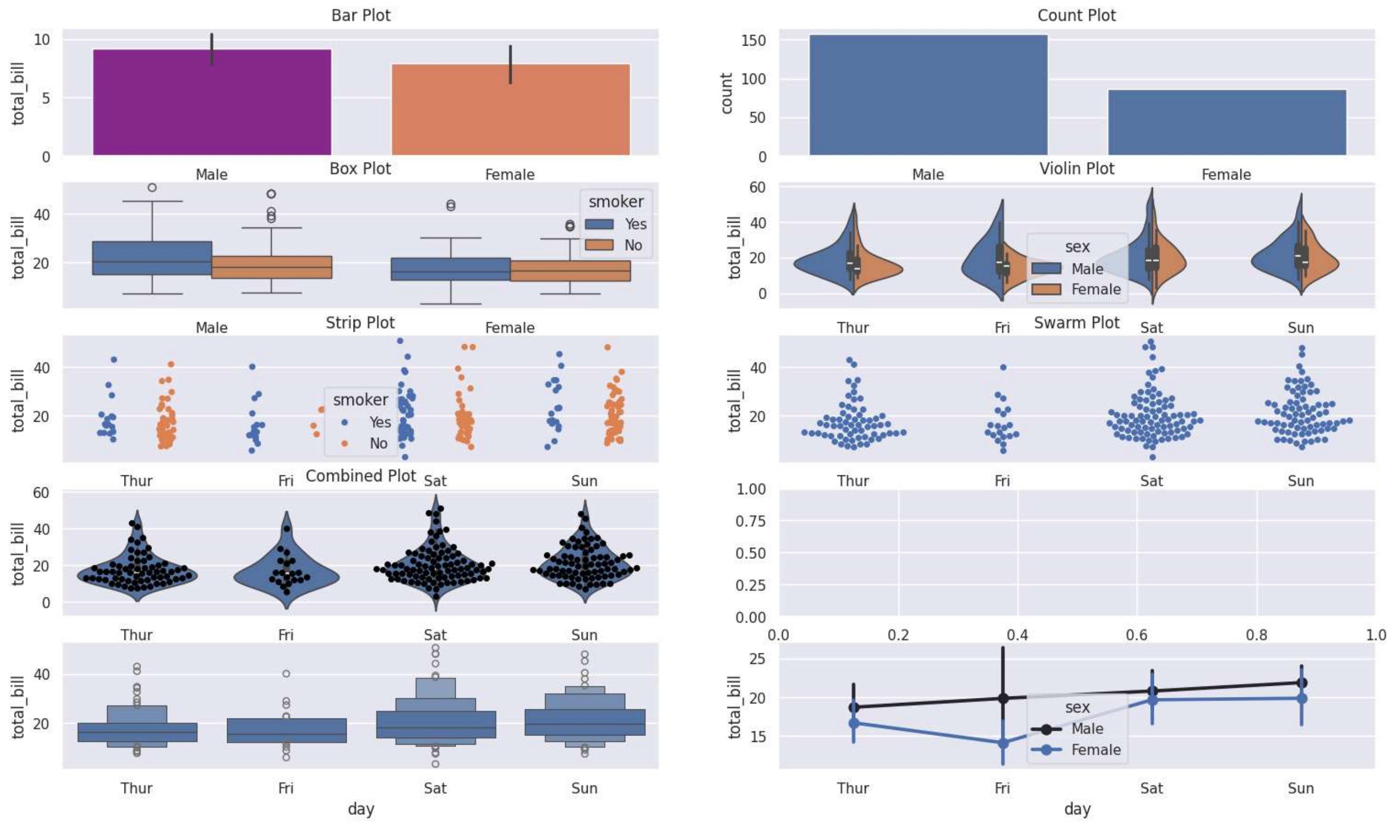
```
#I.Realtional plots
#lineplot: the line plot is one of the most basic plot in seaborn library.
#This plot is mainly used to visualize the data in form of sometime series
sns.set(style="dark")
fig,ax=plt.subplots(ncols=2,nrows=1,figsize=(15,10))
df=sns.load_dataset("tips")
print(df.head())
sns.lineplot(x="total_bill",y="tip",hue="size",style="time",data=df,ax=ax[0]).set_title("Line Plot")
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')
```

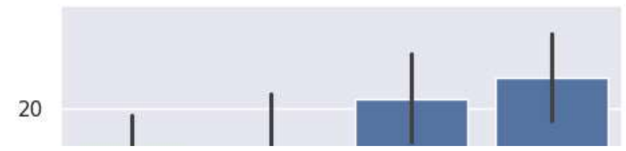
```
   total_bill   tip     sex smoker  day    time  size
0       16.99  1.01  Female     No  Sun  Dinner     2
1       10.34  1.66    Male     No  Sun  Dinner     3
2       21.01  3.50    Male     No  Sun  Dinner     3
3       23.68  3.31    Male     No  Sun  Dinner     2
4       24.59  3.61  Female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



```
#II. Categorical Plots(barplot,countplot,boxplot)
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='sex',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('Violin Plot')
sns.stripplot(x='day',y='total_bill',data=df, jitter=True,hue='smoker',dodge= True,ax=ax[2,0]).set_title('Strip Plot')
#Swarm plot similar to strip plot except the fact
sns.swarmplot(x='day',y='total_bill',data=df, ax=ax[2,1]).set_title('Swarm Plot')
#Combining the idea of a violin plot and a strip plot to form this 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 Plot')
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-7-4e5dd2e738ba>:6: FutureWarning:

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

  sns.barplot(x='sex',y='total_bill',data=df,palette='plasma',estimator=np.std,ax=ax[0,0]).set_title('Bar Plot')
<ipython-input-7-4e5dd2e738ba>:16: FutureWarning:

The `scale` parameter has been renamed to `width_method` and will be removed in v0.15. Pass `width_method='linear'` for the same effect.
  sns.boxenplot(x="day",y="total_bill",color="b",scale="linear",data=df,ax=ax[4,0])
<ipython-input-7-4e5dd2e738ba>:17: 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])
<seaborn.axisgrid.FacetGrid at 0x7ad50bf07010>
```





III.Distribution plots in seaborn is used for examining univariate and bivariate distributions. 4main types of distribution plots

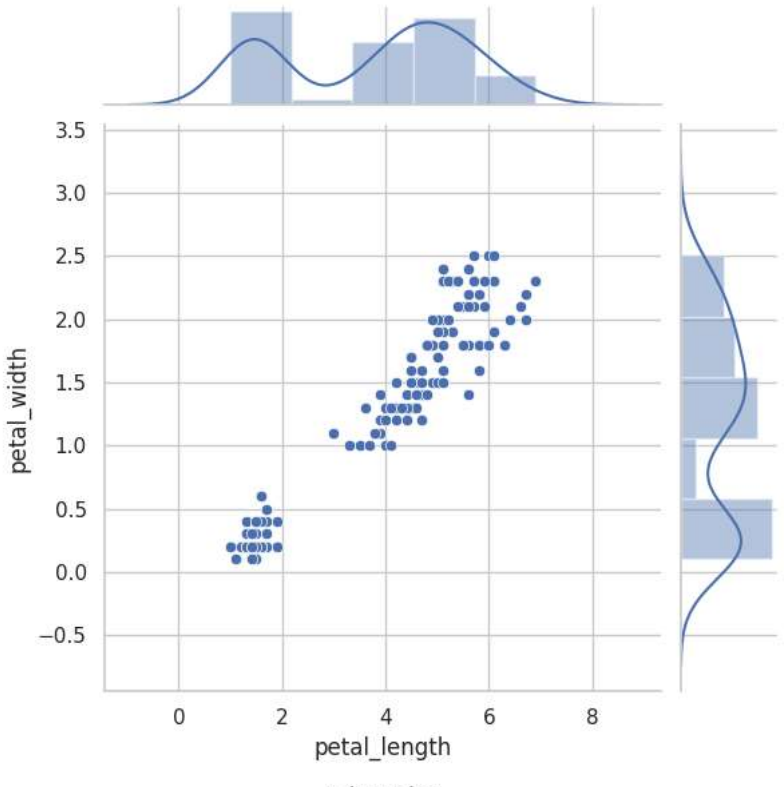
- Join plot
- Dist plot
- Pair plot
- rug plot

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

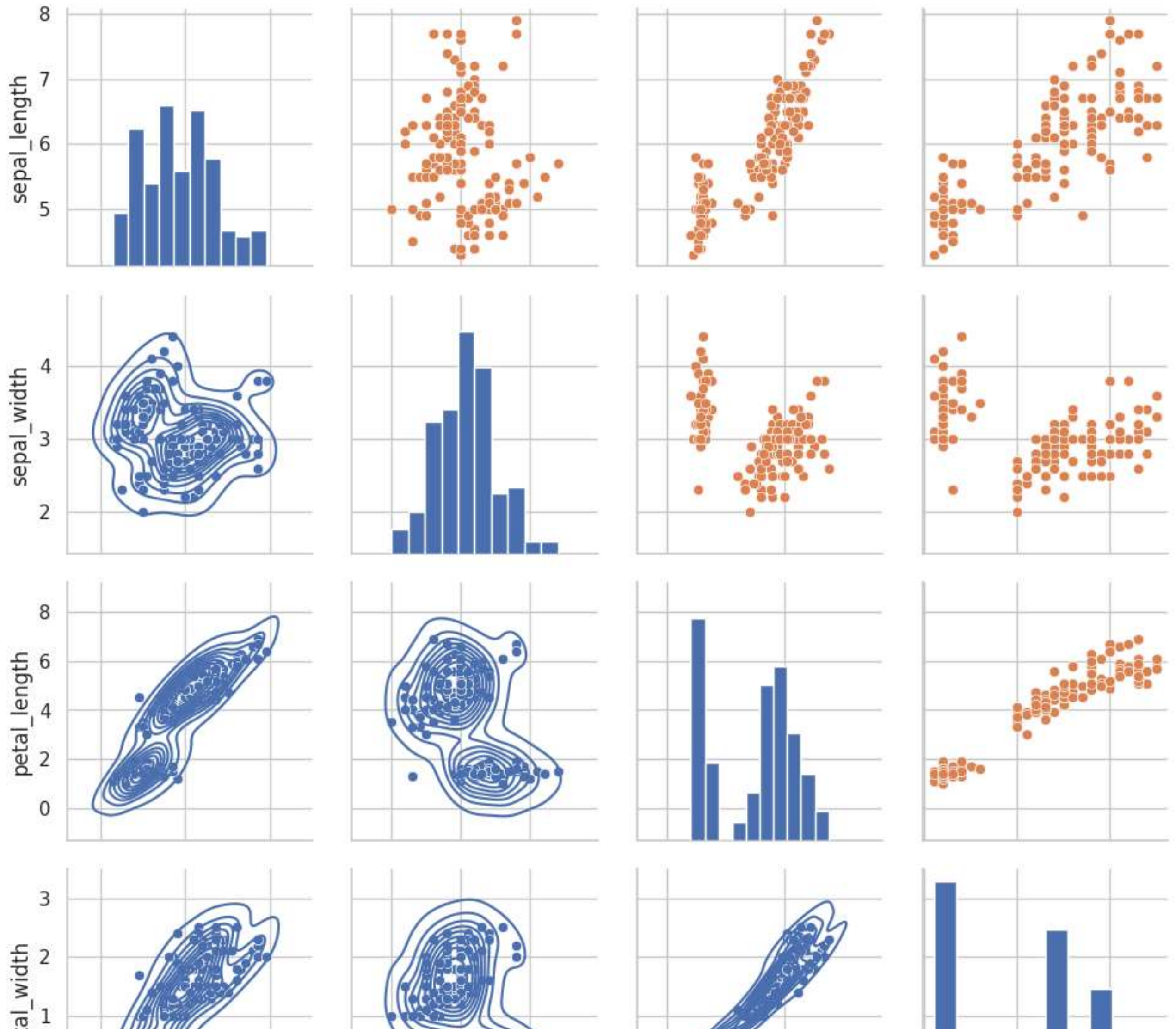
```
jointgrid = sns.JointGrid(x='petal_length',y='petal_width',data=df)
jointgrid.plot_joint(sns.scatterplot)
jointgrid.plot_marginals(sns.distplot)
g=sns.jointplot(x='petal_length',y='petal_width',data=df,kind='hex')
g.fig.suptitle('Joint Plot')
g=sns.pairplot(df,hue="species",palette='coolwarm')
g.fig.suptitle("Pair Plot 1")
g.add_legend()
```

```
/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.PairGrid at 0x7ad511fd1cc0>
```



```
pairgrid=sns.PairGrid(data=df)
pairgrid=pairgrid.map_offdiag(sns.scatterplot)
pairgrid=pairgrid.map_diag(plt.hist)
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 0x7ad50e0764d0>
```


sepal_length
sepal_width
petal_length
al_width
8
7
6
5
4
3
2
8
6
4
2
0
3
2
1