PLOTS

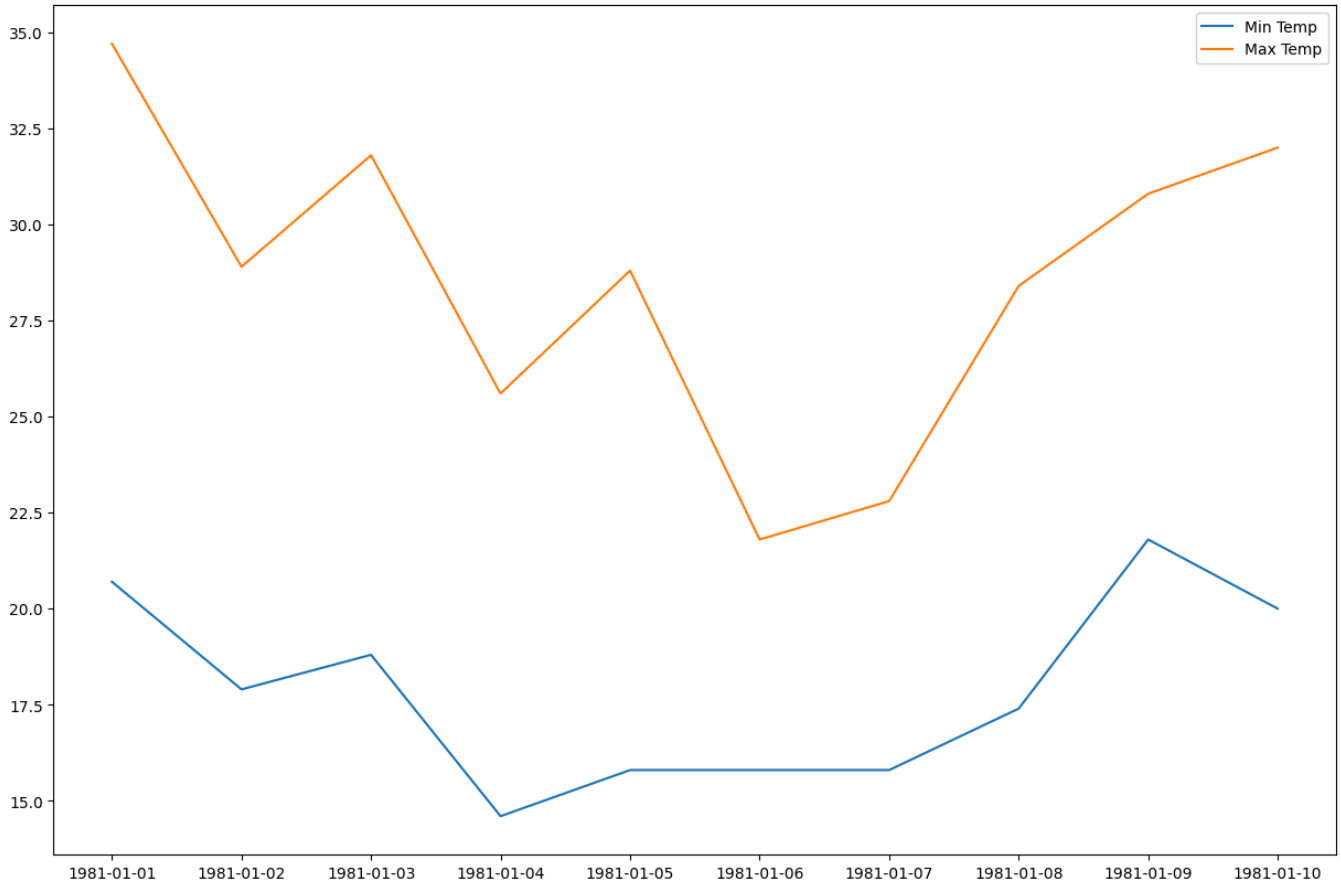
realational plots: this plot is used to understand relation b/w two varaiables categorical plots-this plot deals with categorical variables and how they can be visualized distribution plots-this plot is used for examiniung univariate and bivariate distribution matrix plots-it is an array of scatterplots regression plots-are primarily intended to add a vuisual guide that helps to emphasize patterns in dataset during exploratory data analyses

+ Code + Text

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

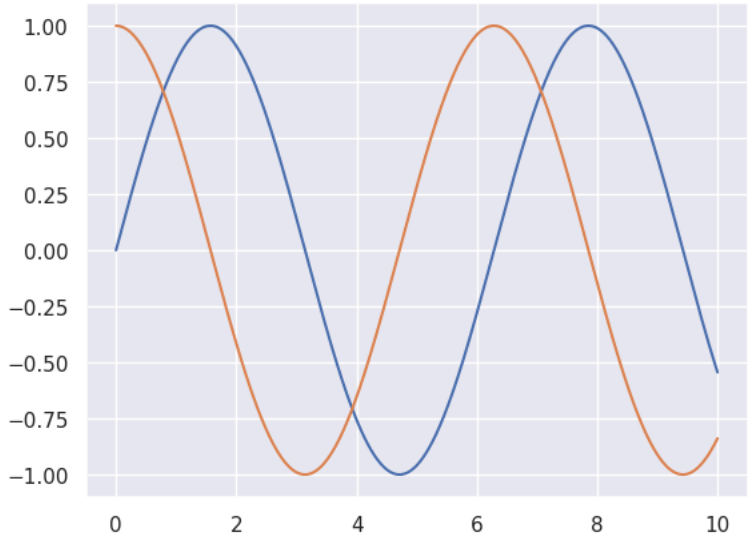
```
#simple plotting with seaborn
d=['1981-01-01','1981-01-02','1981-01-03','1981-01-04','1981-01-05','1981-01-06','1981-01-07','1981-01-08','1981-01-09','1981-01-10']
min_t=[20.7,17.9,18.8,14.6,15.8,15.8,15.8,17.4,21.8,20.0]
max_t=[34.7,28.9,31.8,25.6,28.8,21.8,22.8,28.4,30.8,32.0]
#plotting
fig,axes=plt.subplots(nrows=1,ncols=1,figsize=(15,10))
axes.plot(d,min_t,label='Min Temp')
axes.plot(d,max_t,label='Max Temp')
axes.legend()
```

```
<matplotlib.legend.Legend at 0x79554d1b26e0>
```



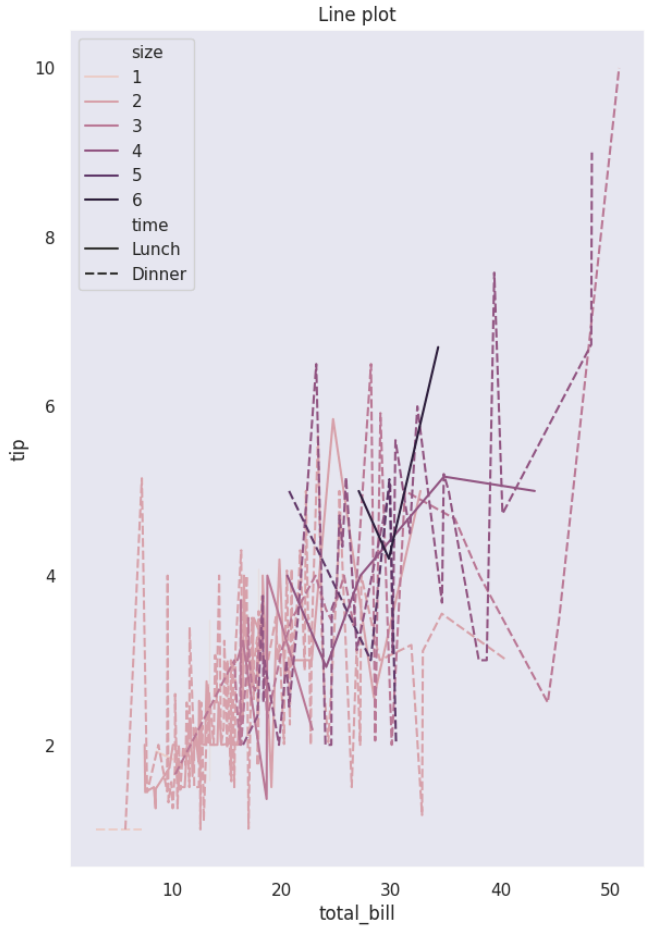

```
#seaborn style as the default matplotlib style
sns.set()
```

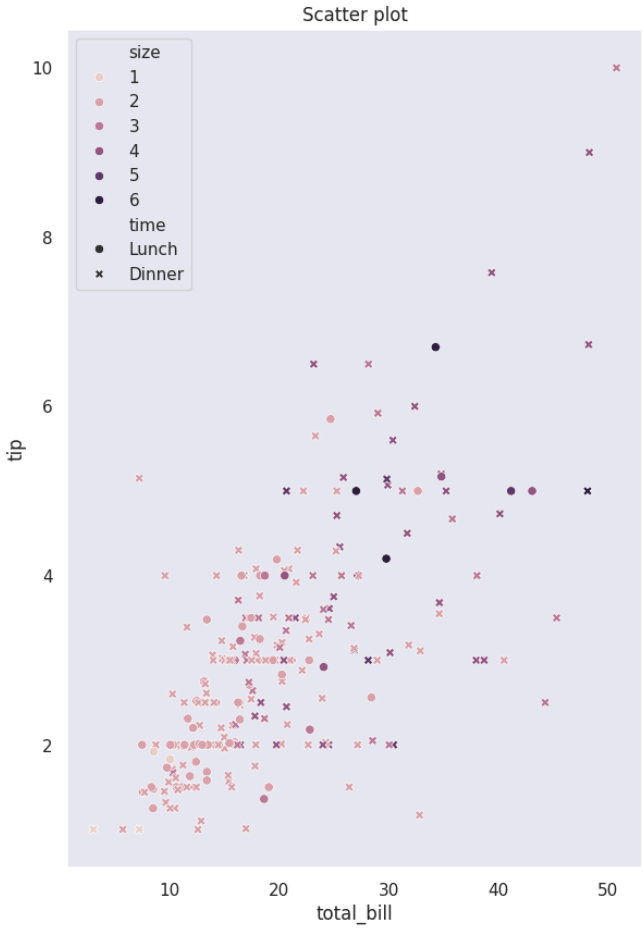
```
#simple sine plot
x=np.linspace(0,10,1000)
plt.plot(x,np.sin(x),x,np.cos(x));
```




```
#Relationbal plot
#Line plot:the line plot is one of the most basic plot in seaborn libraray.
#this plot is mainly used to visualize the data in form of some time series,i.e. in
sns.set(style="dark")
fig,ax=plt.subplots(ncols=2,nrows=1,figsize=(15,10))
#loading data with seaborn
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")
#scatter 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')
```

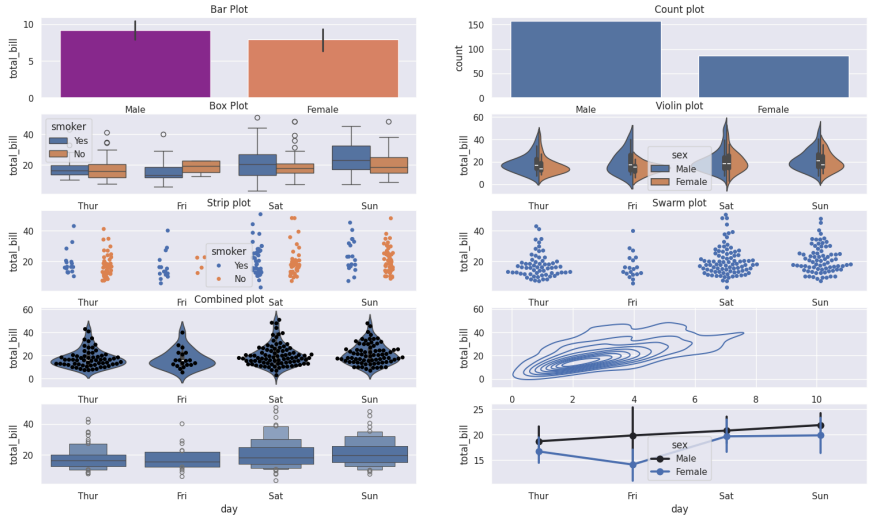
```
   total_bill   tip     sex smoker  day    time  size
0       16.99  1.01  Female     No  Sun  Dinner     2
1       10.34  1.66    Male     No  Sun  Dinner     3
2       21.01  3.50    Male     No  Sun  Dinner     3
3       23.68  3.31    Male     No  Sun  Dinner     2
4       24.59  3.61  Female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






```
#categorical plots
#plots are used for visualizing the relationship b/w varaibles
#variables can be either be completely numerical or category luike group,class
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_title('Bar Plot')
#countplot : counts the categories and returns a count of their occurences
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')
#violinplot
sns.violinplot(x='day',y='total_bill',data=df,hue='sex',split=True,ax=ax[1,1]).set_title('Violin plot')
#Stripplot
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
sns.swarmplot(x='day',y='total_bill',data=df,ax=ax[2,1]).set_title('Swarm plot')
#combine violin and 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 plot')
#density plot
sns.kdeplot(x="tip",y="total_bill",data=df,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])
```

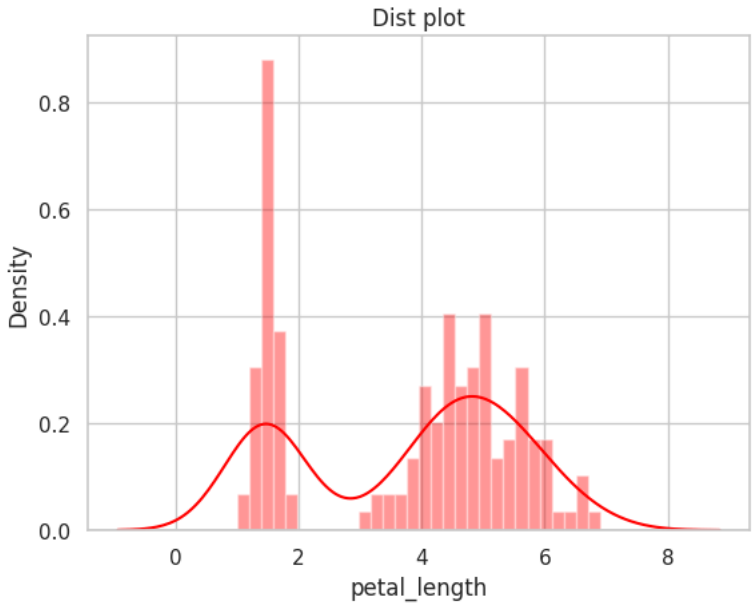
```
<ipython-input-7-a1eb2e497a2d>:9: FutureWarning:

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-7-a1eb2e497a2d>:26: 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-7-a1eb2e497a2d>:28: 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])
<Axes: xlabel='day', ylabel='total_bill'>
```



Distribution plots-in seaborn is used for examining univariate and bivariate distribution 4 main types: joinplot distplot pairplot rugplot

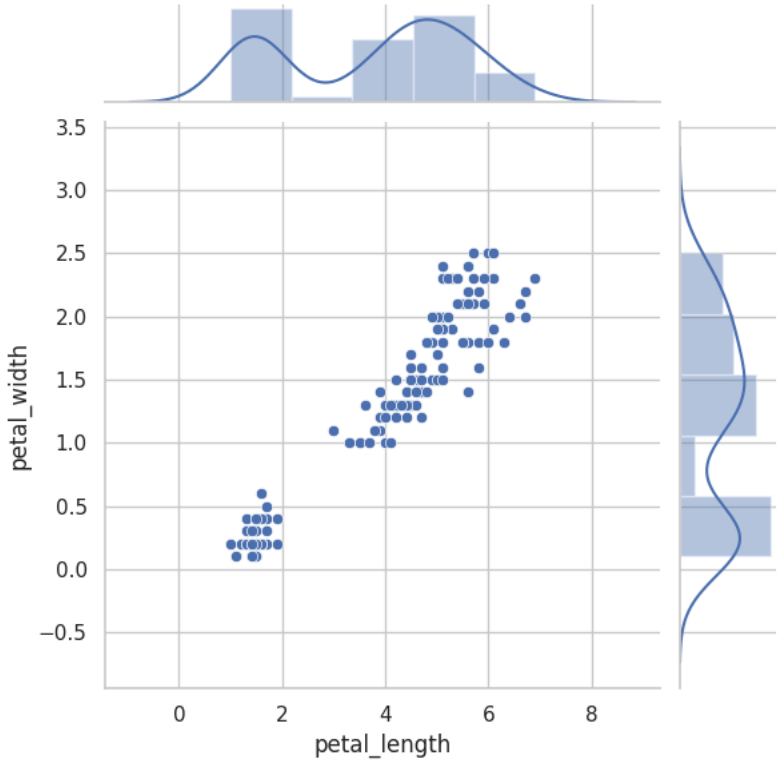
```
sns.set_style('whitegrid')
#data-'iris'
df=sns.load_dataset('iris')
print(df.head())
#Displot-used for univariant
#kde-is a way to estimate probability density function(pdf)
sns.distplot(df['petal_length'],kde=True,color='red',bins=30).set_title('Dist plot')
```

```
   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-8-b2521e15fee6>:7: 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')
```



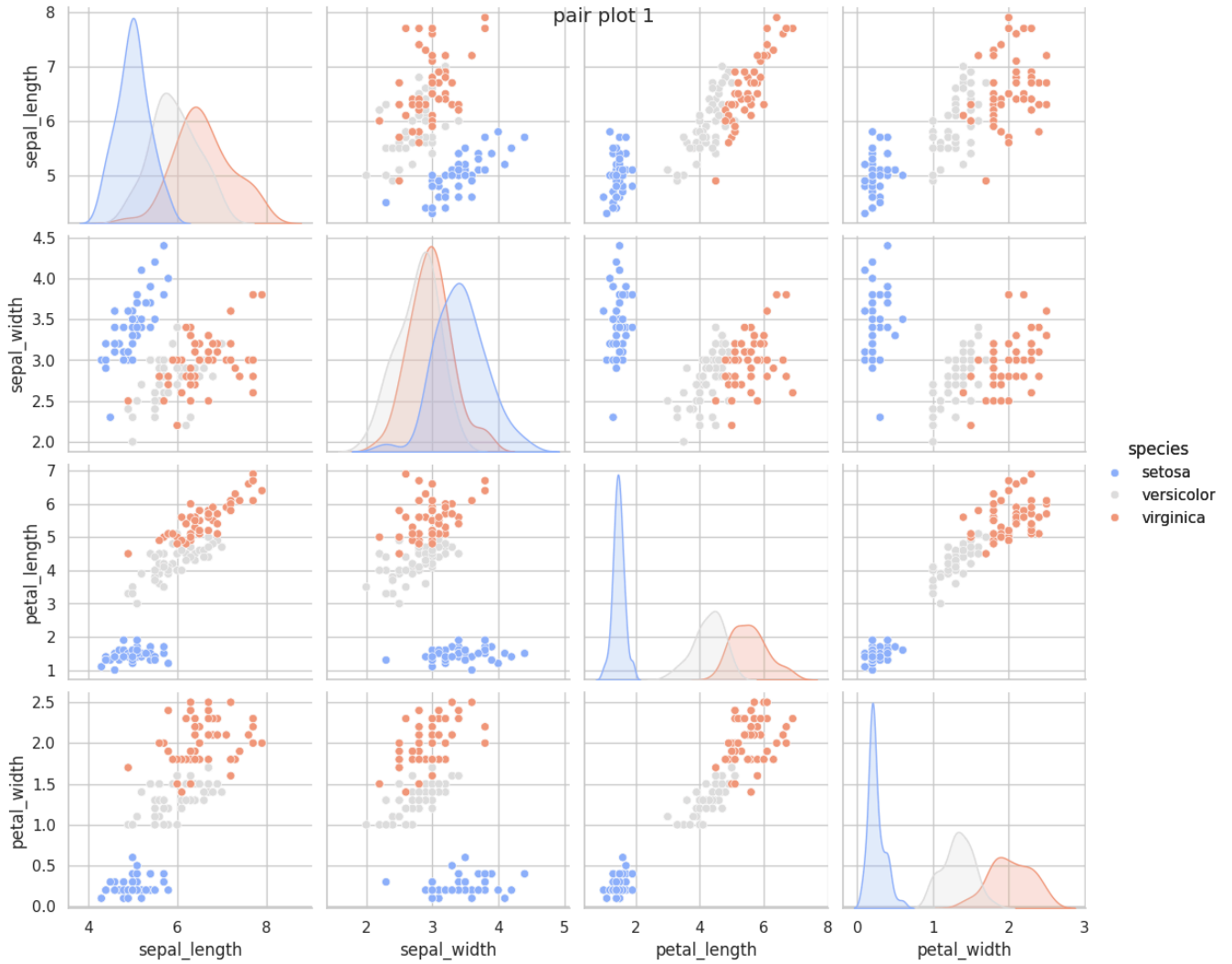

```
#joint grid
jointgrid=sns.JointGrid(x='petal_length',y='petal_width',data=df)
jointgrid.plot_joint(sns.scatterplot)
jointgrid.plot_marginals(sns.distplot)
g=sns.jointplot(x='petal_length',y='petal_width',data=df,kind='hex')
g.fig.suptitle('joint plot')
```

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



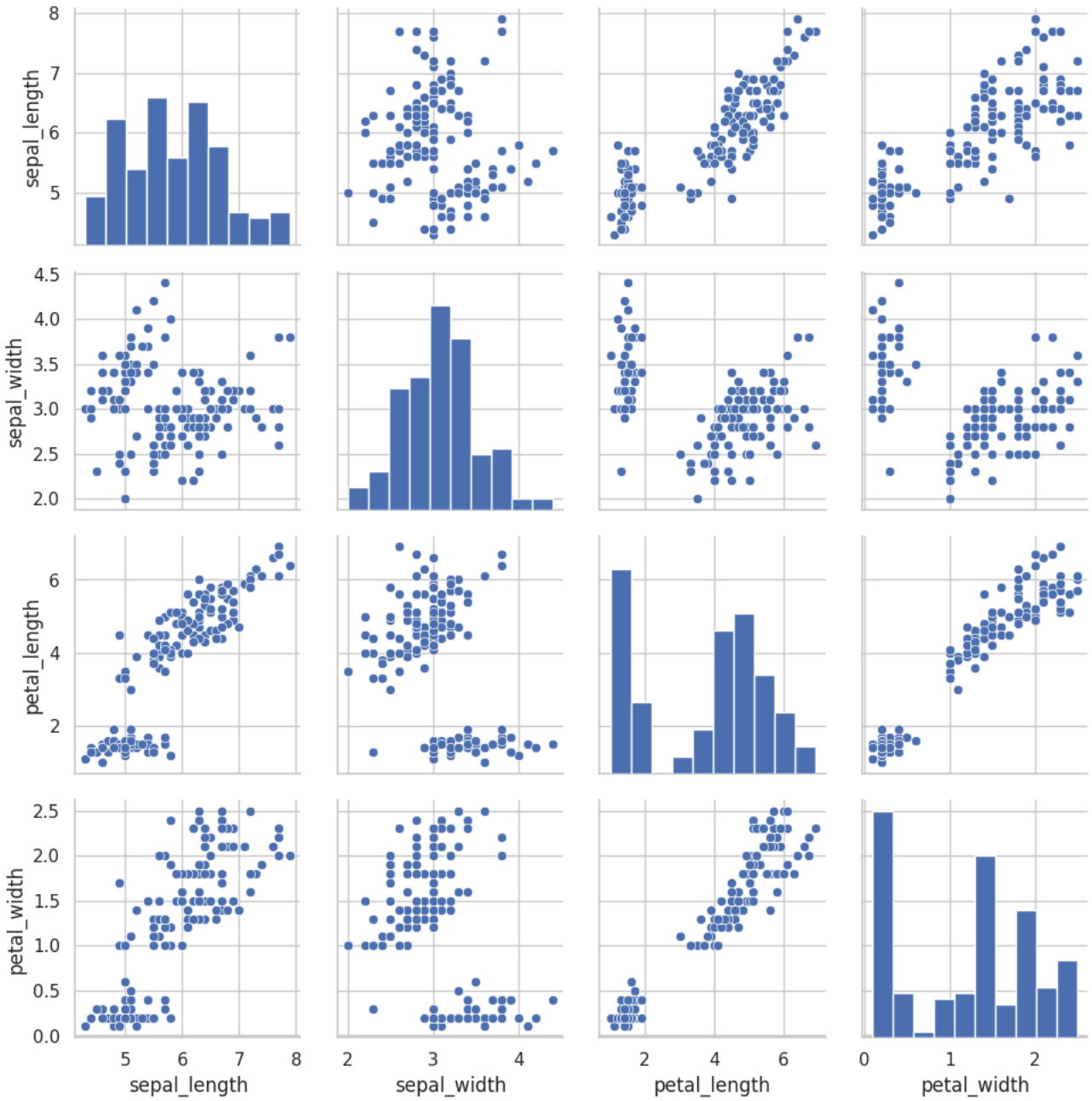

```
#pairplot
g=sns.pairplot(df,hue="species",palette='coolwarm')
g.fig.suptitle("pair plot 1")
g.add_legend()
```

```
<seaborn.axisgrid.PairGrid at 0x7955461a5ba0>
```

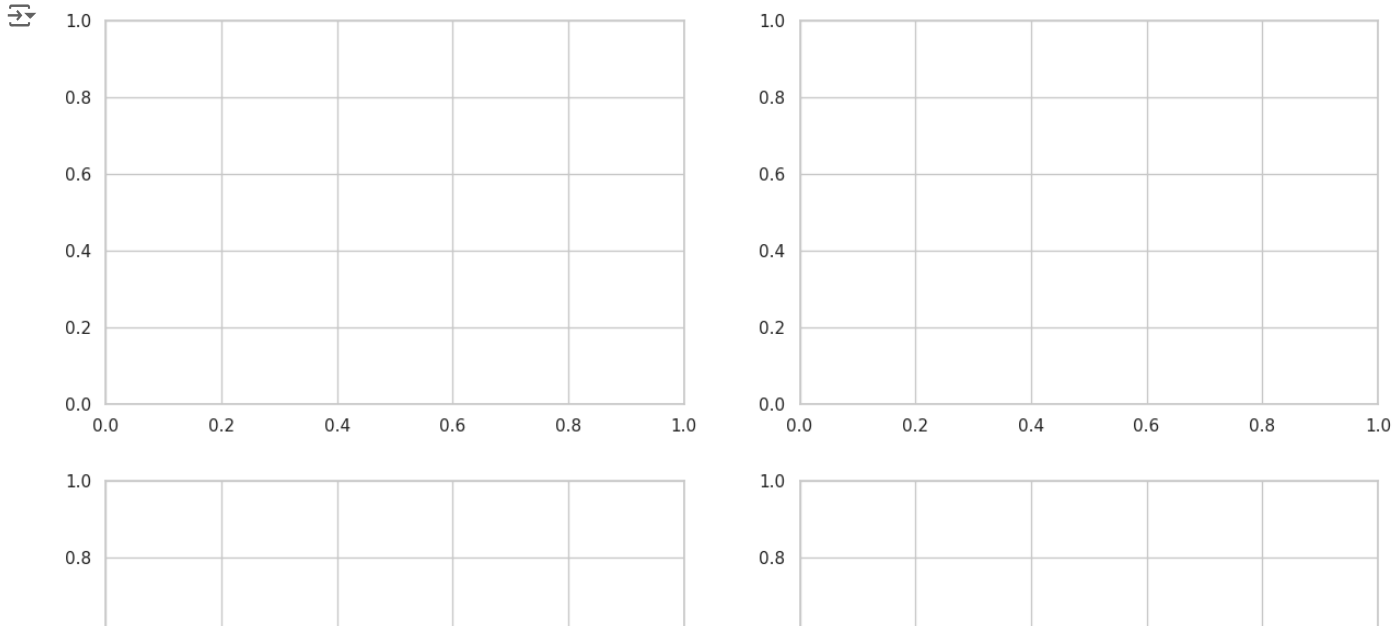



```
#pairGrid
pairgrid=sns.PairGrid(data=df)
pairgrid=pairgrid.map_offdiag(sns.scatterplot)
pairgrid=pairgrid.map_diag(plt.hist)
#diff kinds of plots on upper triangular axes
pairgrid=sns.PairGrid(data=df)
pairgrid=pairgrid.map_upper(sns.scatterplot)
pairgrid=pairgrid.map_diag(plt.hist)
pairgrid=pairgrid.map_lower(sns.kdeplot)
#avoid redundancy
g=sns.PairGrid(df,diag_sharey=False,corner=True)
g.map_lower(sns.scatterplot)
g.map_diag(sns.kdeplot)
```

```
<seaborn.axisgrid.PairGrid at 0x7955454c55a0>
```




```
#matrix plots
fig,ax=plt.subplots(nrows=2,ncols=2,figsize=(15,10))
#data
df1=sns.load_dataset('flights')
df2=sns.load_dataset('iris')
df11=pd.pivot_table(values='passengers',index='month',columns='year',data=df1)
#calculates correlation b/w cols in dataframe
dfc1=df1.corr(numeric_only=True)
dfc2=df2.corr(numeric_only=True)
```




```
#heatmaps
sns.heatmap(df11,cmap='YlGnBu',linecolor='r',linewidths=0.5,annot=True,fmt='d',square=True,ax=ax[0,0]).set_title('heat map flights')
sns.heatmap(dfc2,cmap='coolwarm',linecolor='black',linewidths=1,annot=True,ax=ax[0,1]).set_title('heat maps Iris')
#lower triangle display
mask1=np.triu(dfc2)
sns.heatmap(dfc2,annot=True,mask=mask1,ax=ax[1,0],cmap='coolwarm').set_title('heat maps lower triangle')
#upper triangle
mask2=np.tril(dfc2)
sns.heatmap(dfc2,annot=True,cmap='YlGnBu',mask=mask2,ax=ax[1,1],).set_title('heat maps upper triangle')
#cluster maps
sns.clustermap(df11,cmap='RdYlGn')
#standard_scale
sns.clustermap(df11,cmap='plasma',standard_scale=1)
```

```
<seaborn.matrix.ClusterGrid at 0x79554486e740>
```