

Data vizualization with Seaborn

```
In [ ]: import warnings  
warnings.filterwarnings('ignore')
```

```
In [ ]: import seaborn as sns  
import matplotlib as plt  
plt.style.use('ggplot')
```

```
In [ ]: tips= sns.load_dataset('tips')  
iris= sns.load_dataset('iris')
```

```
In [ ]: tips.sample(2)
```

```
Out[ ]:   total_bill  tip  sex  smoker  day  time  size  
    229      22.12  2.88  Female     Yes  Sat  Dinner     2  
    140      17.47  3.50  Female      No Thur  Lunch     2
```

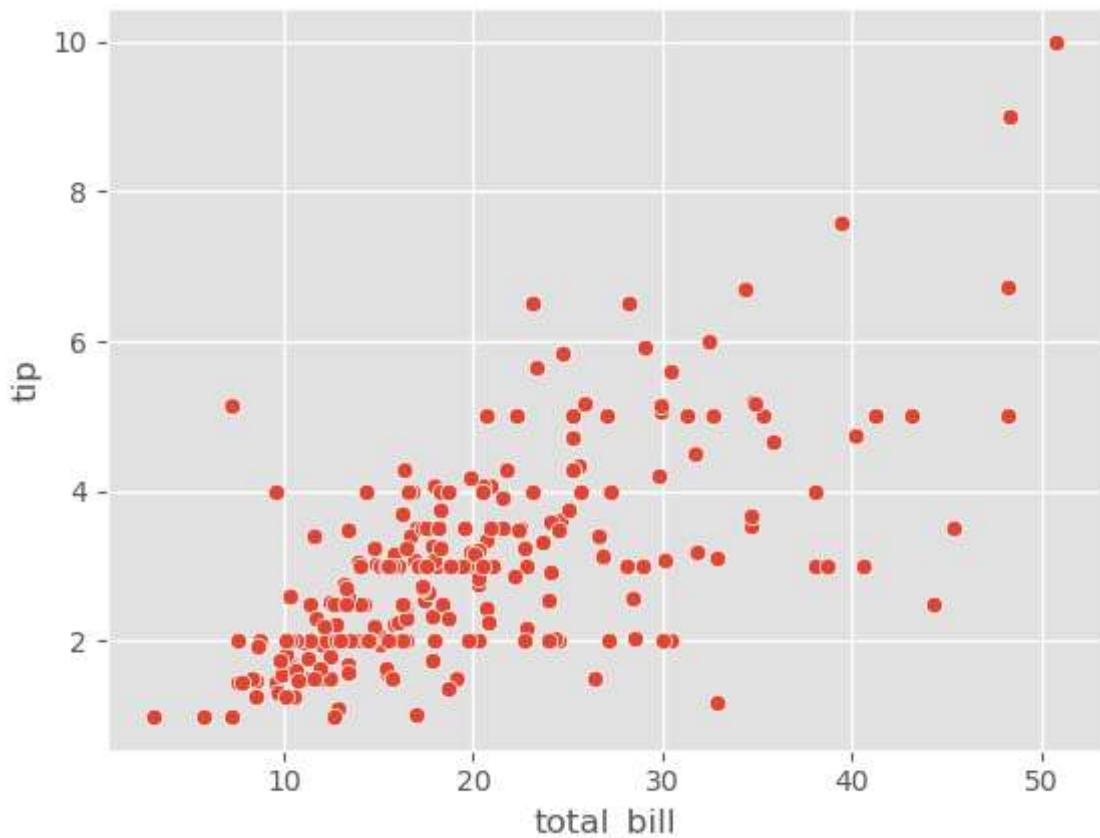
```
In [ ]: iris.sample(2)
```

```
Out[ ]:   sepal_length  sepal_width  petal_length  petal_width  species  
    27          5.2           3.5           1.5          0.2  setosa  
    21          5.1           3.7           1.5          0.4  setosa
```

Categorical Scatter plots

```
In [ ]: # on numerical data  
sns.scatterplot(data=tips,x='total_bill',y='tip')
```

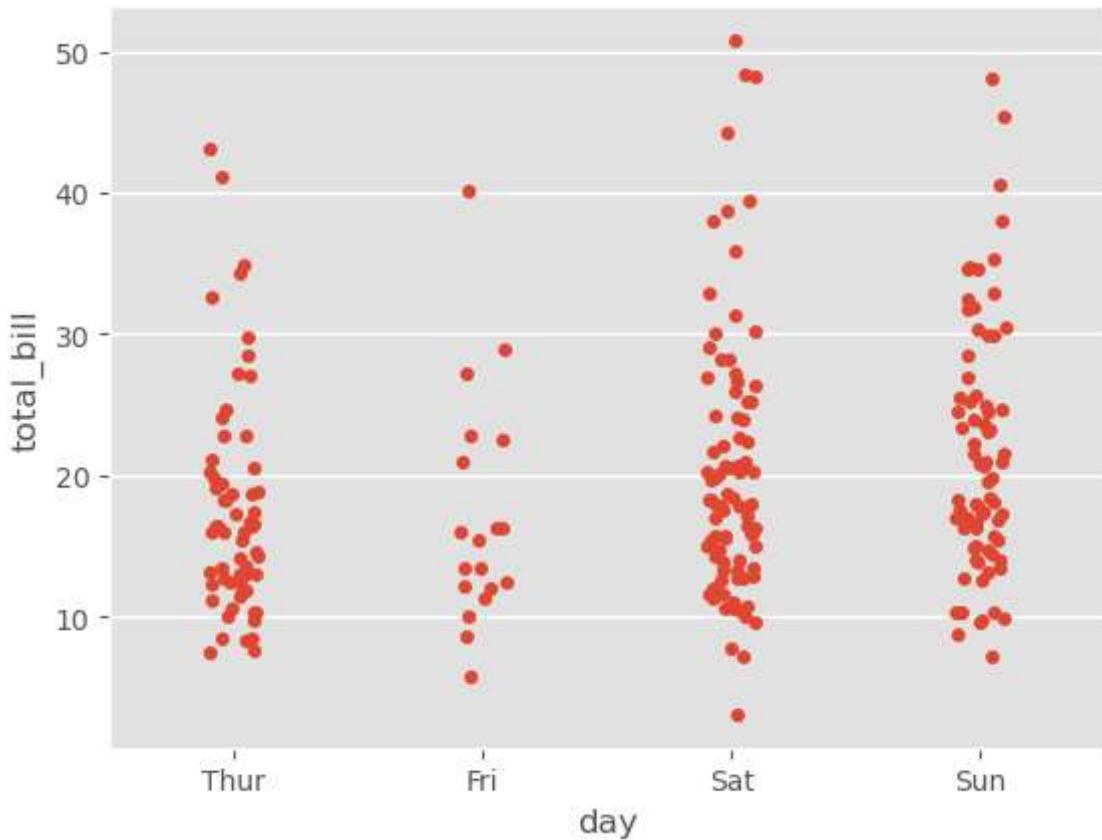
```
Out[ ]: <Axes: xlabel='total_bill', ylabel='tip'>
```



1. Strip plot

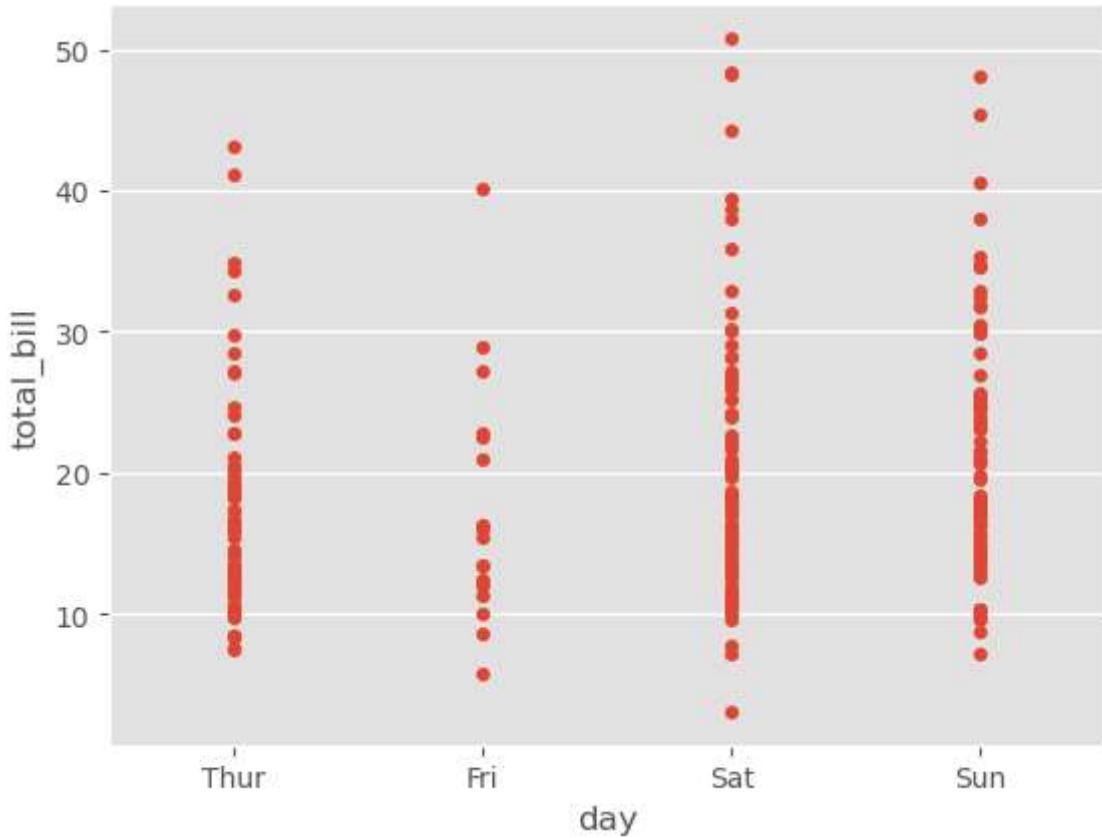
```
In [ ]: # axes Level function  
# on categorical data  
sns.stripplot(data=tips,x='day',y='total_bill')
```

```
Out[ ]: <Axes: xlabel='day', ylabel='total_bill'>
```



```
In [ ]: # jitter
sns.stripplot(data=tips,x='day',y='total_bill',jitter=False)
```

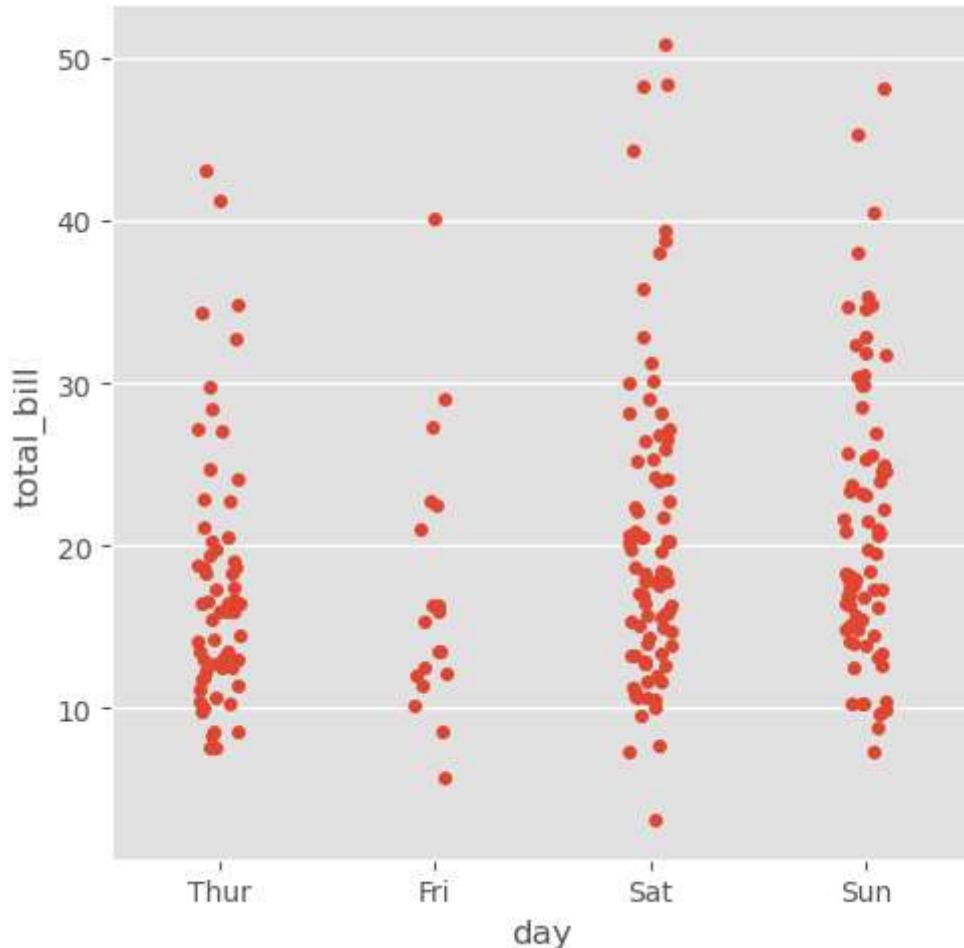
```
Out[ ]: <Axes: xlabel='day', ylabel='total_bill'>
```



Catplot

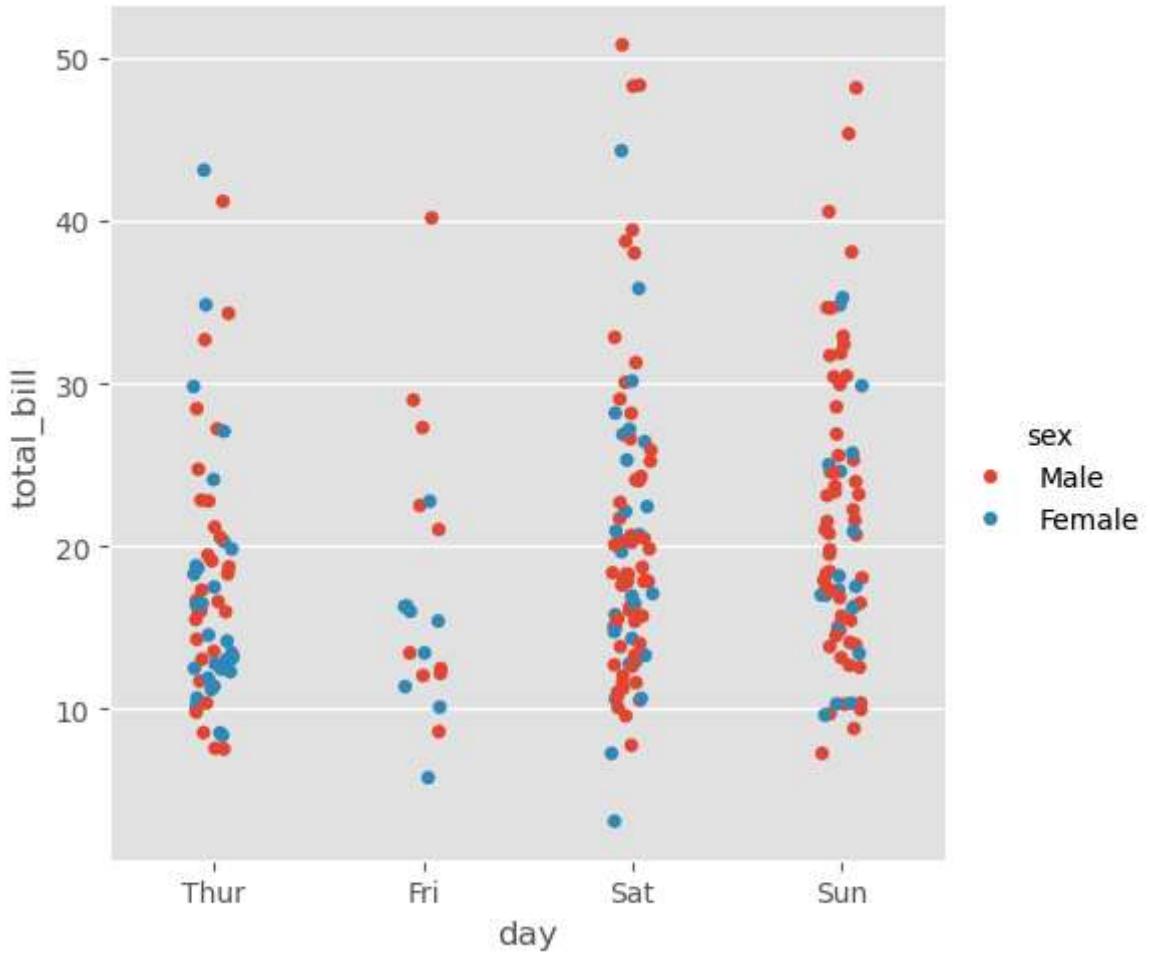
```
In [ ]: # catplot -categorical plot
# figure level functions
sns.catplot(data=tips,x='day',y='total_bill',kind='strip')
```

```
Out[ ]: <seaborn.axisgrid.FacetGrid at 0x1ec29b0dad0>
```



```
In [ ]: # hue
sns.catplot(data=tips,x='day',y='total_bill',kind='strip',hue='sex')
```

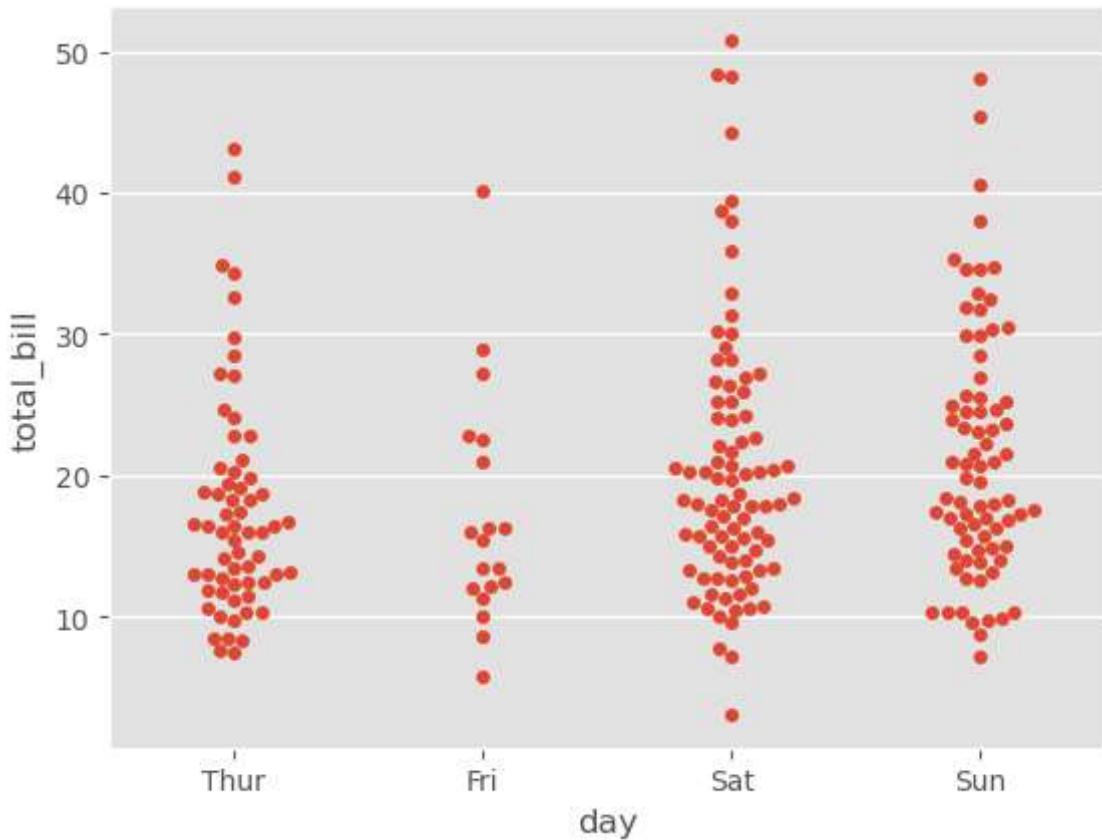
```
Out[ ]: <seaborn.axisgrid.FacetGrid at 0x1ec29b0c810>
```



2. swarm plot

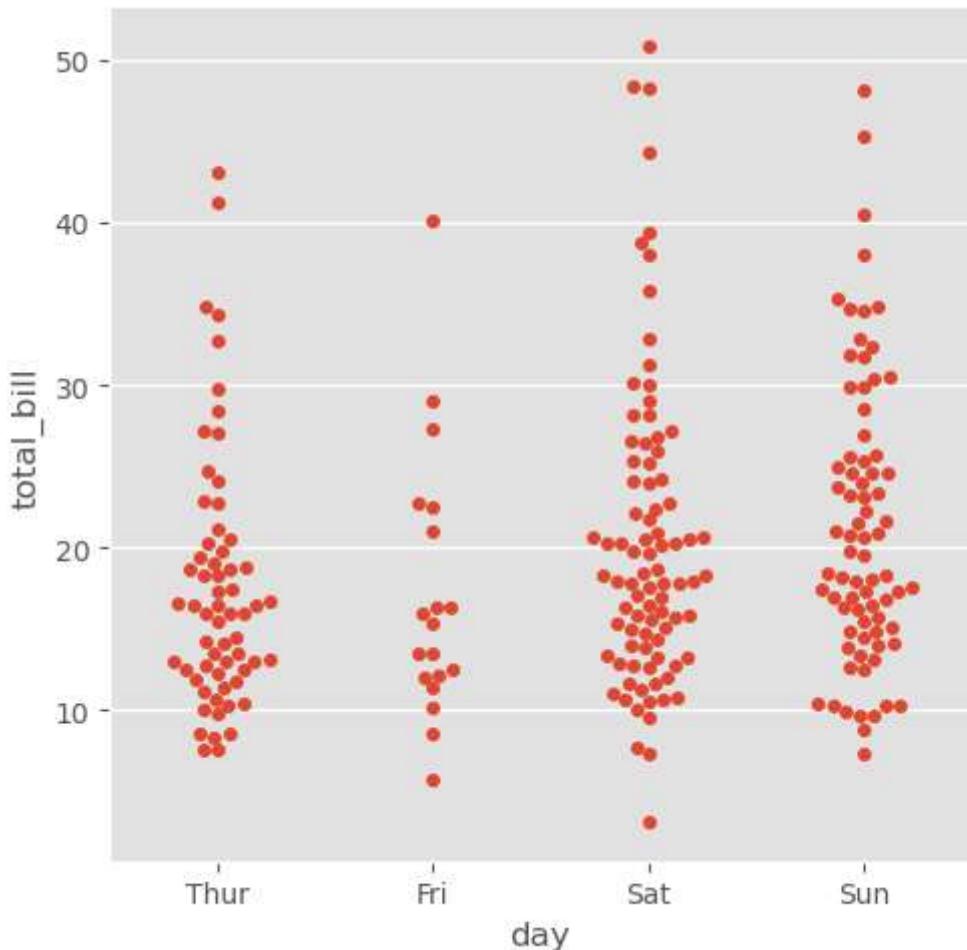
```
In [ ]: sns.swarmplot(data=tips,x='day',y='total_bill')
```

```
Out[ ]: <Axes: xlabel='day', ylabel='total_bill'>
```



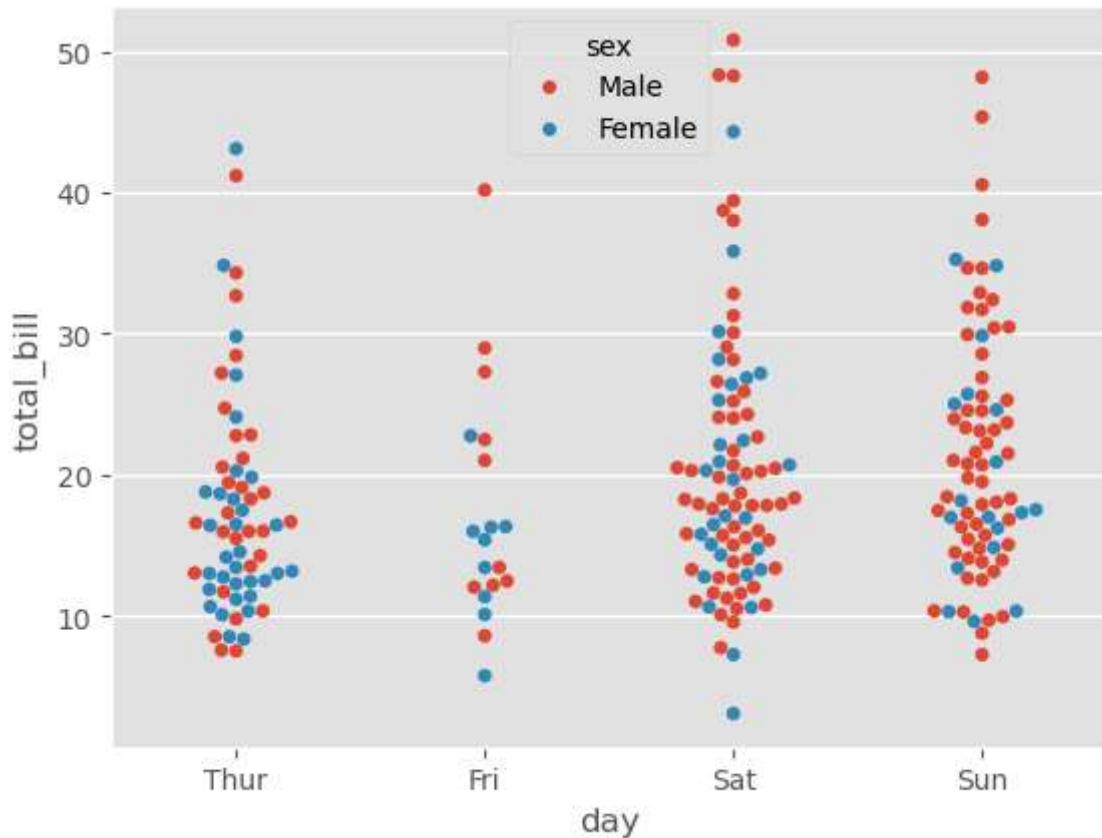
```
In [ ]: # use by catplot
sns.catplot(data=tips,x='day',y='total_bill',kind='swarm')
```

```
Out[ ]: <seaborn.axisgrid.FacetGrid at 0x1ec2f1f62d0>
```



```
In [ ]: # hue  
sns.swarmplot(data=tips,x ='day', y='total_bill',hue='sex')
```

```
Out[ ]: <Axes: xlabel='day', ylabel='total_bill'>
```

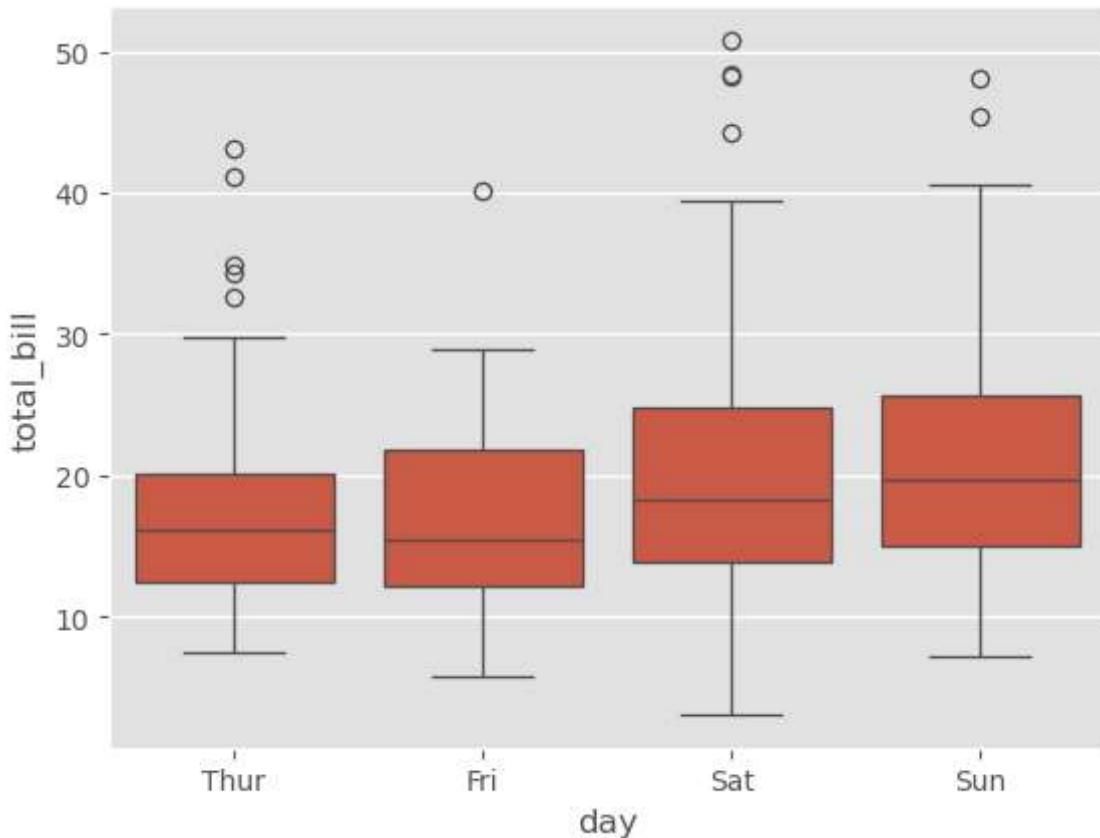


Categorical Distribution plots

1. Boxplot

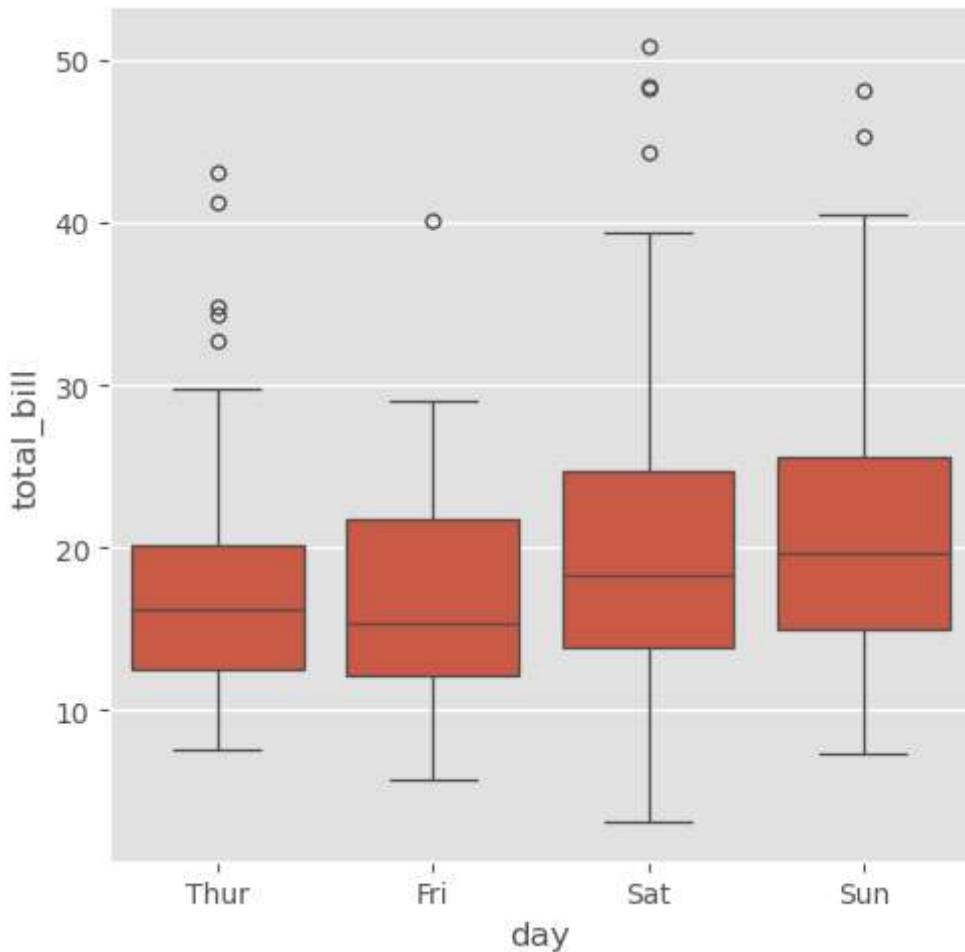
```
In [ ]: #Box plot  
sns.boxplot(data=tips,x='day',y='total_bill')
```

```
Out[ ]: <Axes: xlabel='day', ylabel='total_bill'>
```



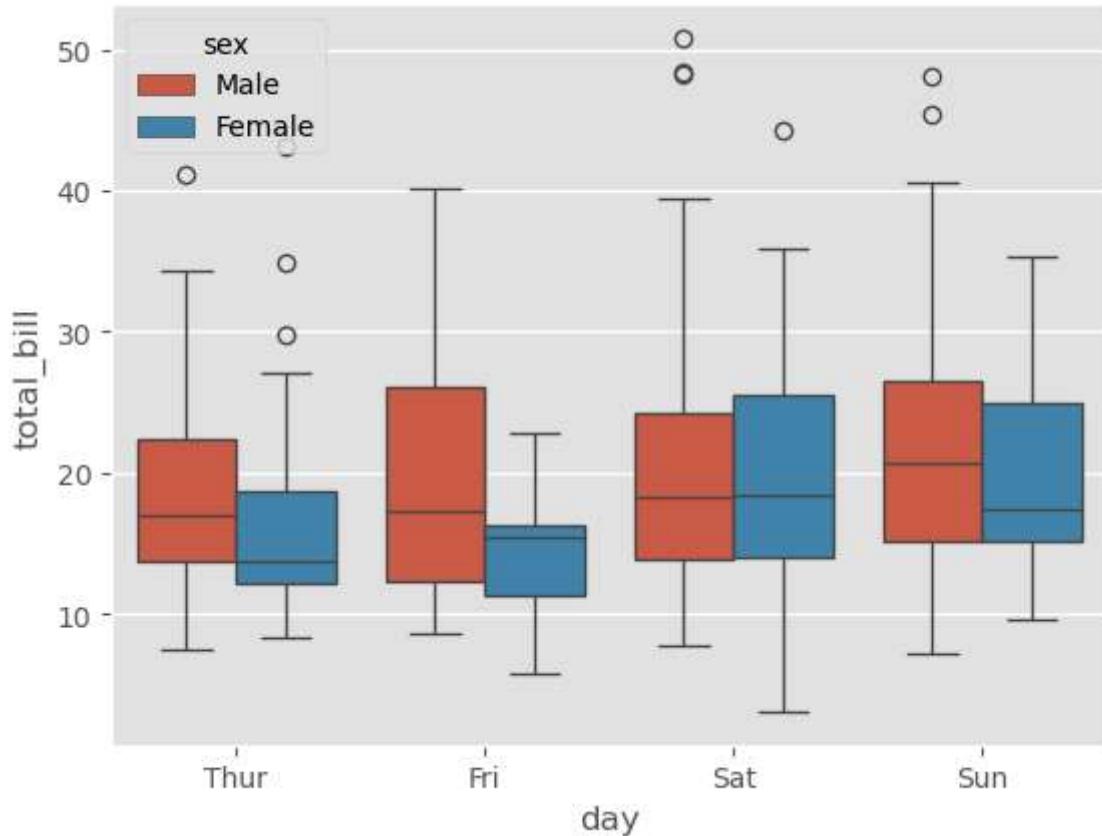
```
In [ ]: # Using catplot - Figure Level  
sns.catplot(data=tips,x='day',y='total_bill',kind='box')
```

```
Out[ ]: <seaborn.axisgrid.FacetGrid at 0x1ec2f469a10>
```



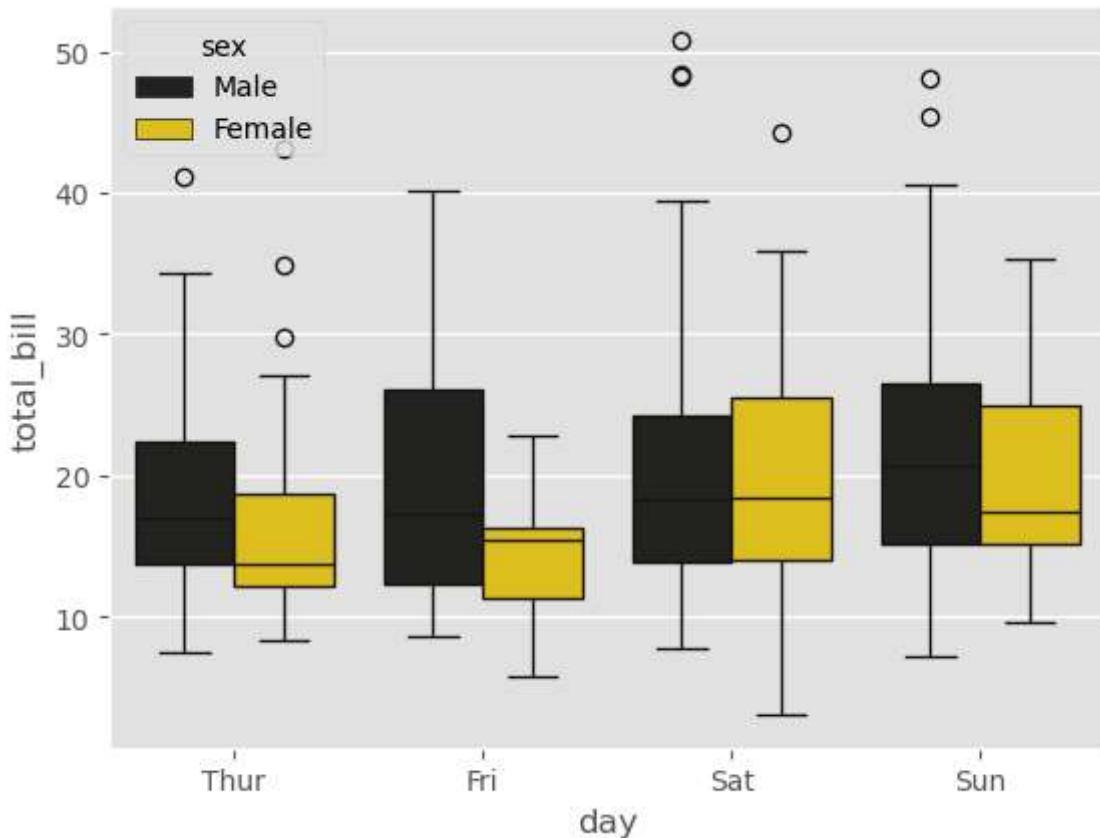
```
In [ ]: # hue
sns.boxplot(data=tips,x='day',y='total_bill',hue='sex')
```

```
Out[ ]: <Axes: xlabel='day', ylabel='total_bill'>
```



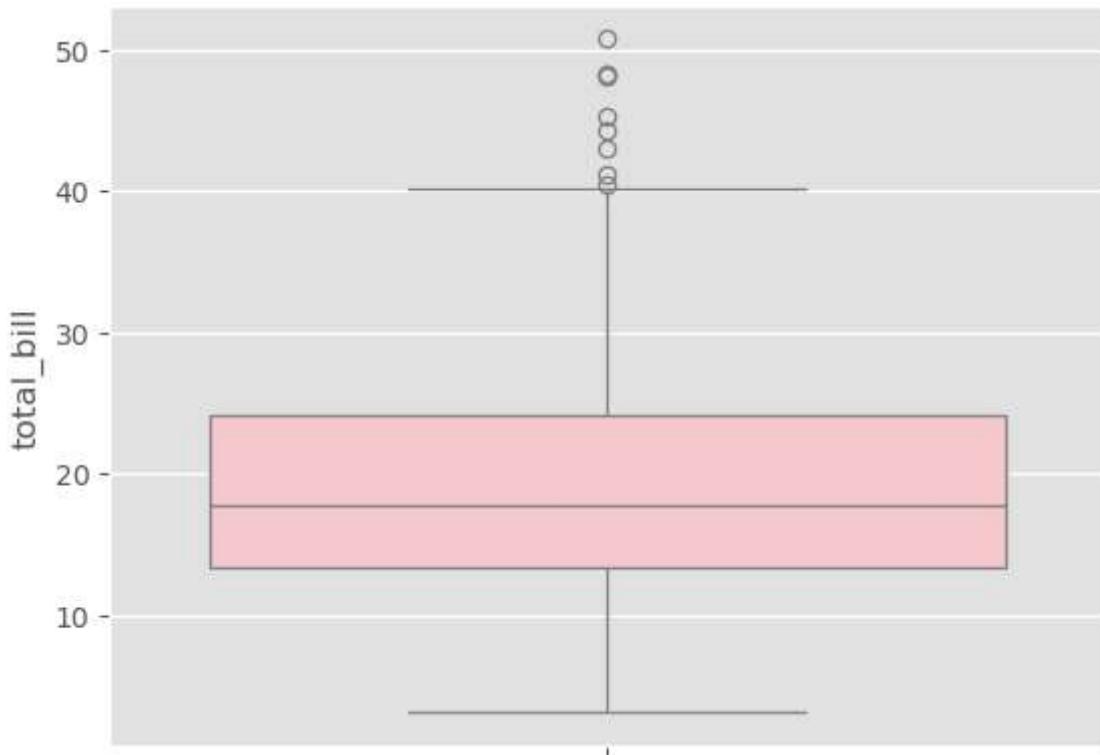
```
In [ ]: # color
sns.boxplot(data=tips,x='day',y='total_bill',hue='sex', color = 'gold')
```

```
Out[ ]: <Axes: xlabel='day', ylabel='total_bill'>
```



```
In [ ]: # single boxplot -> numerical col
sns.boxplot(data=tips , y='total_bill', color ='pink')
```

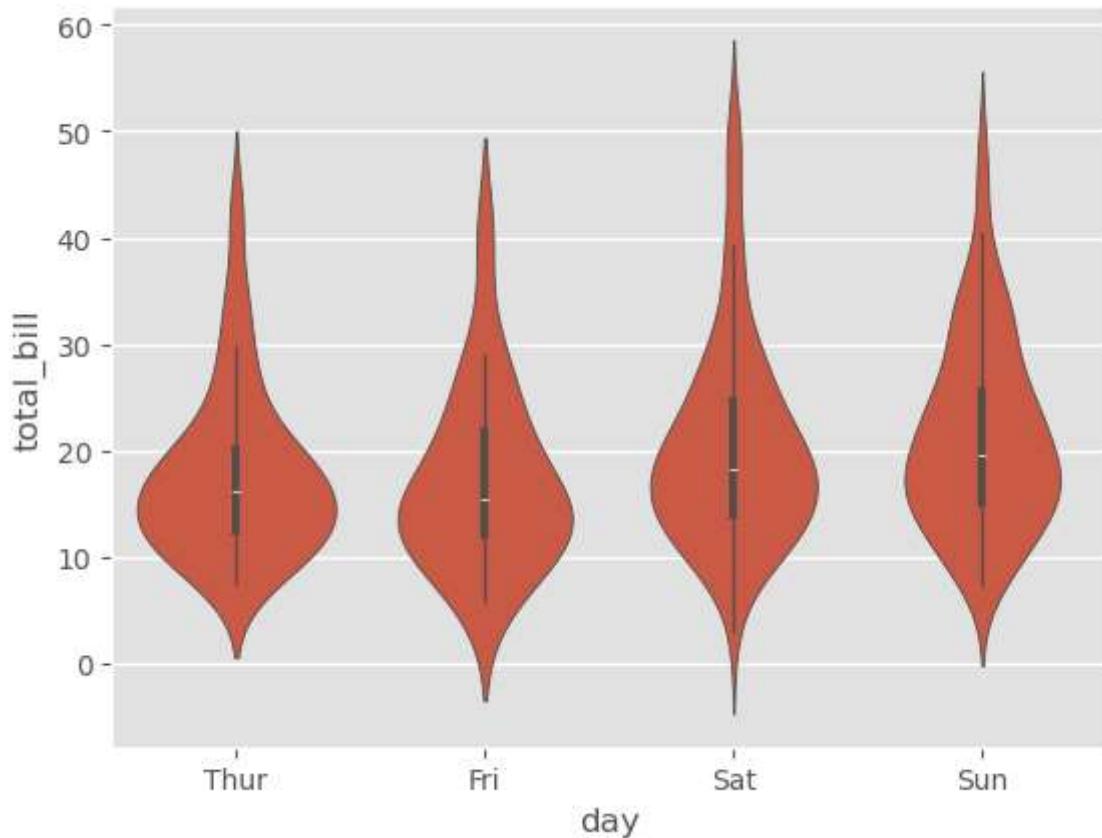
```
Out[ ]: <Axes: ylabel='total_bill'>
```



2. Violin Plot

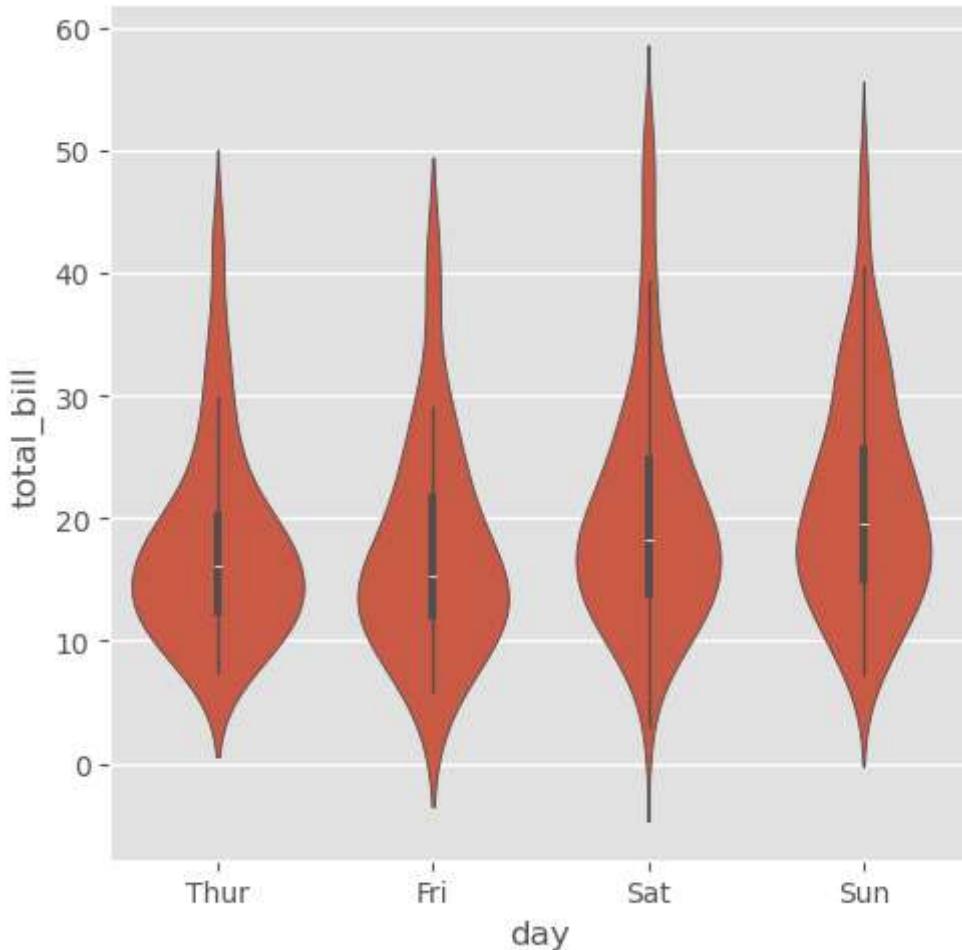
```
In [ ]: # Violinplot = (Boxplot + KDEplot)
sns.violinplot(data=tips,x='day',y='total_bill')
```

```
Out[ ]: <Axes: xlabel='day', ylabel='total_bill'>
```

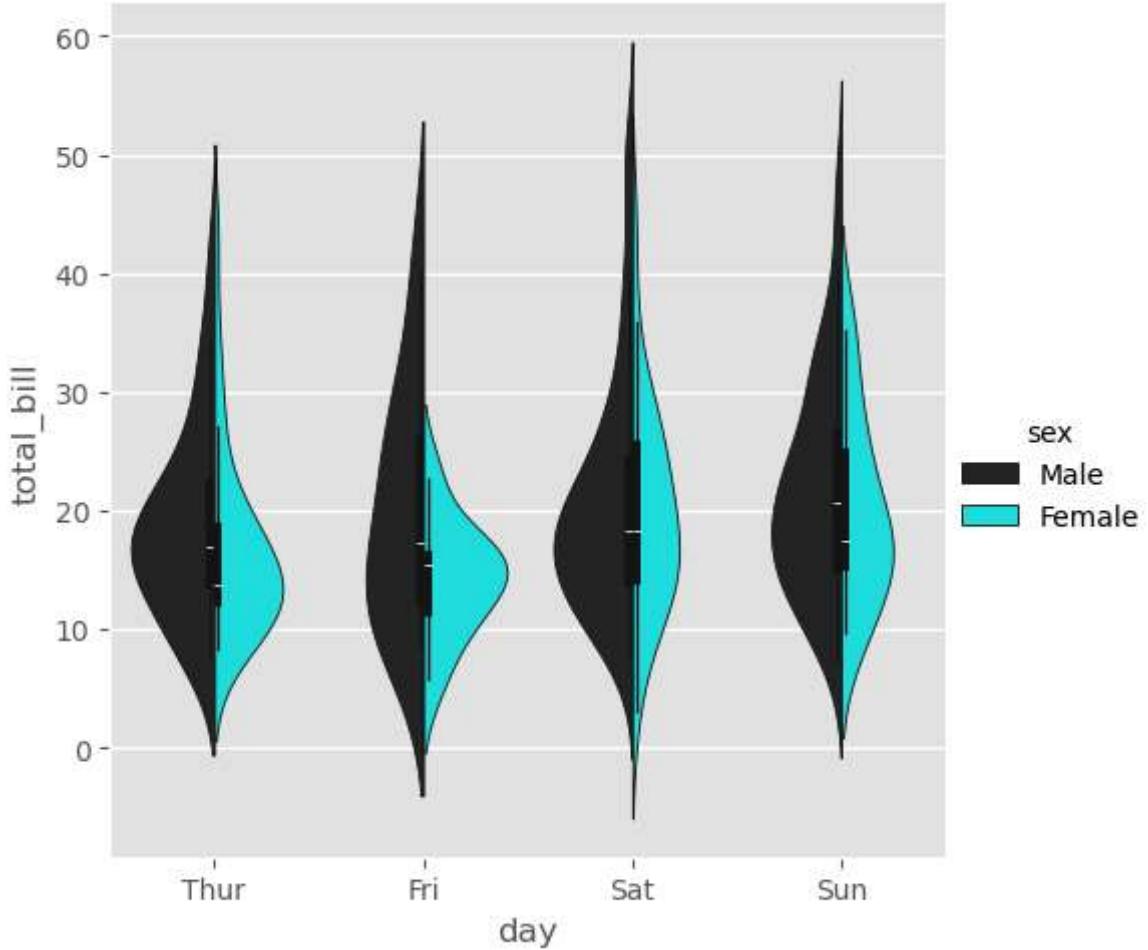


```
In [ ]: # catplot -figure level function
sns.catplot(data=tips,x='day',y='total_bill',kind ='violin')
```

```
Out[ ]: <seaborn.axisgrid.FacetGrid at 0x1ec30803950>
```

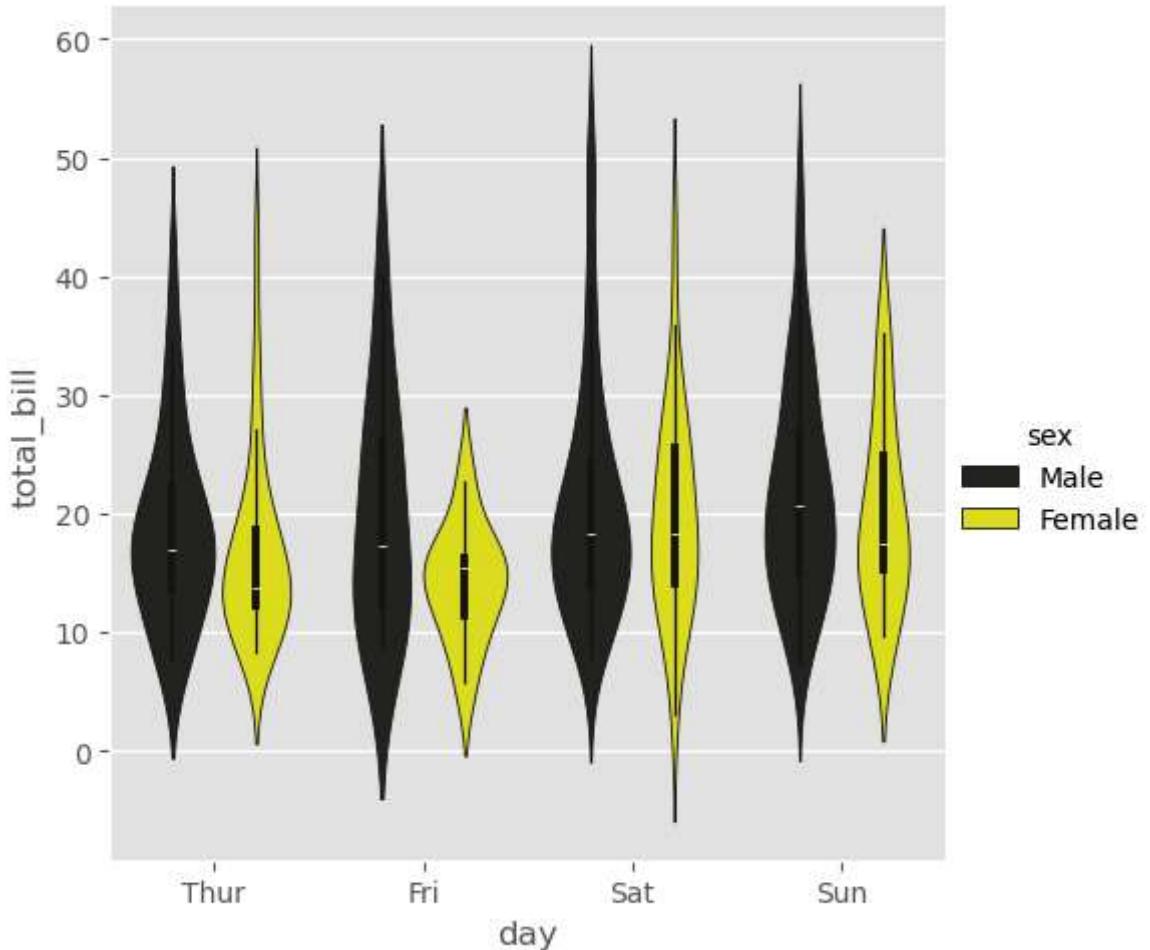


```
In [ ]: # Color
sns.catplot(data=tips,x='day',y='total_bill',kind ='violin', hue ='sex', split =
Out[ ]: <seaborn.axisgrid.FacetGrid at 0x1ec308b4390>
```

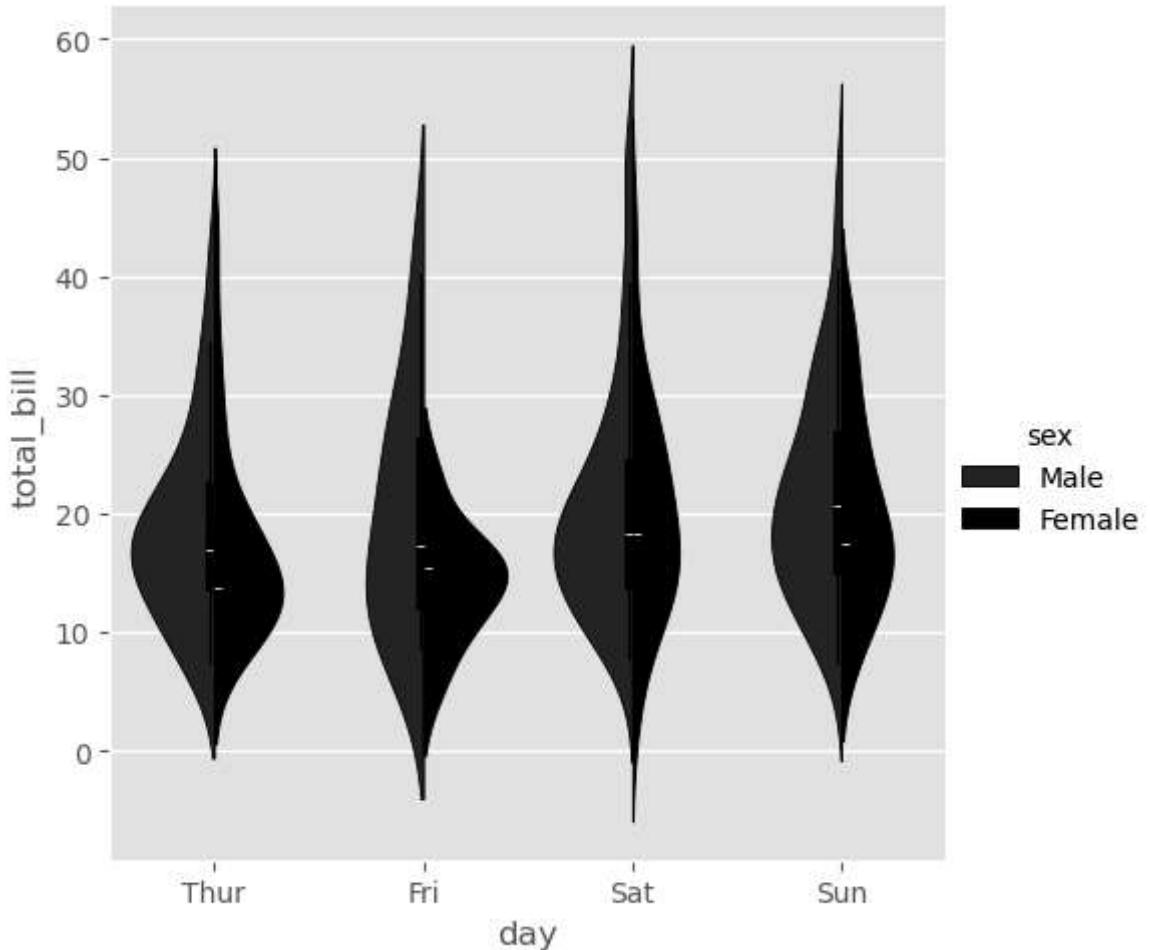


```
In [ ]: # hue
sns.catplot(data=tips,x='day',y='total_bill',kind ='violin',hue ='sex' , color =
```

```
Out[ ]: <seaborn.axisgrid.FacetGrid at 0x1ec31211390>
```



```
In [ ]: # split
sns.catplot(data=tips,x='day',y='total_bill',kind ='violin', hue ='sex',color ='
Out[ ]: <seaborn.axisgrid.FacetGrid at 0x1ec308dd910>
```

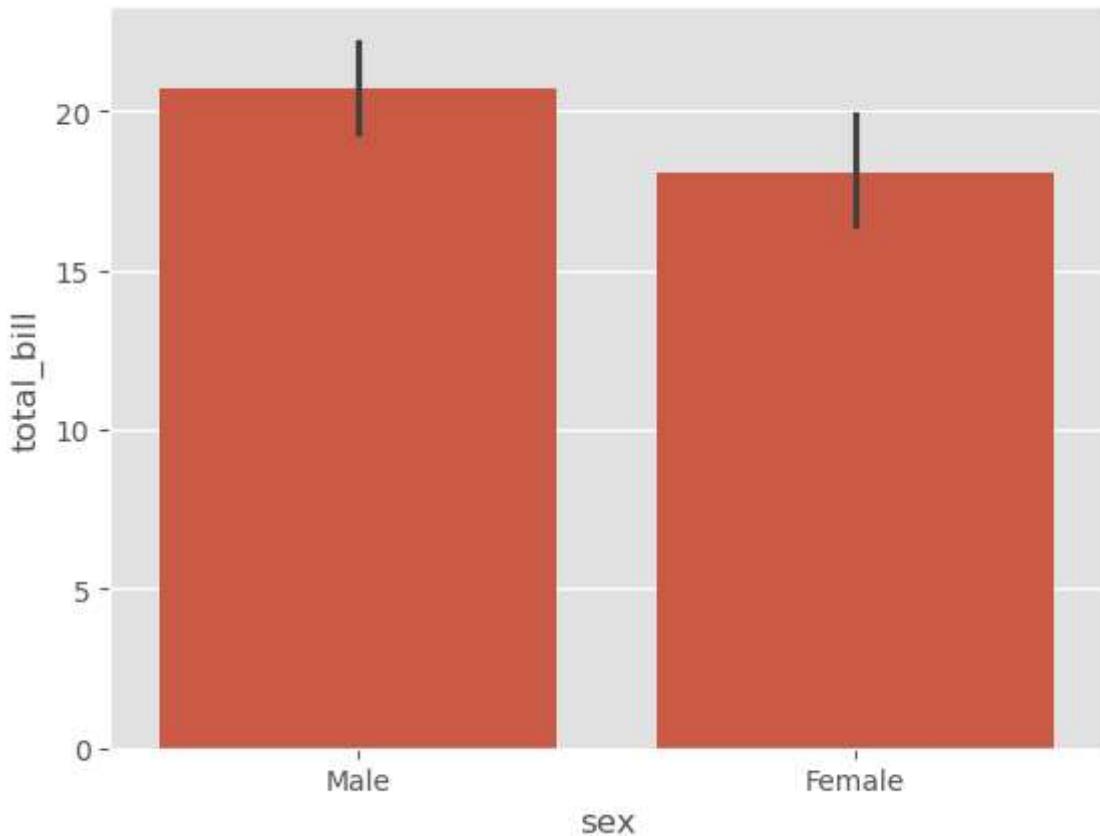


Categorical Estimate plot

1.Barplot

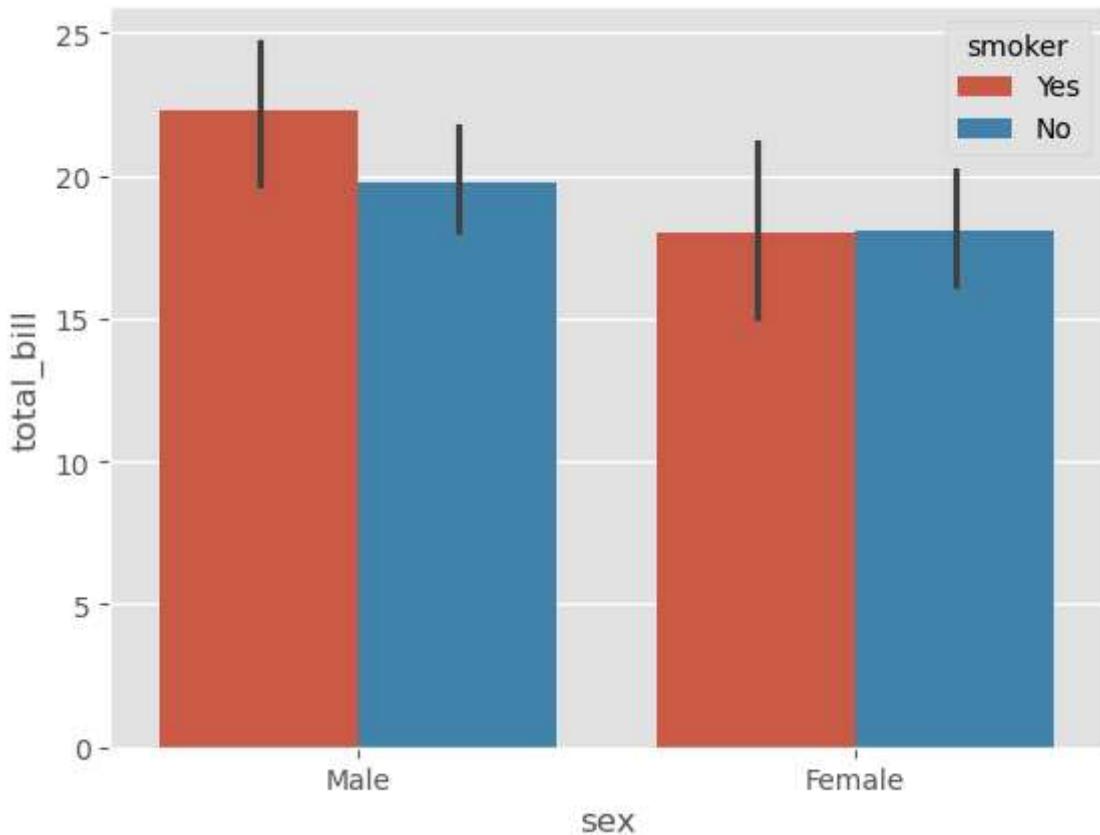
```
In [ ]: sns.barplot(data=tips, x='sex', y='total_bill')
```

```
Out[ ]: <Axes: xlabel='sex', ylabel='total_bill'>
```



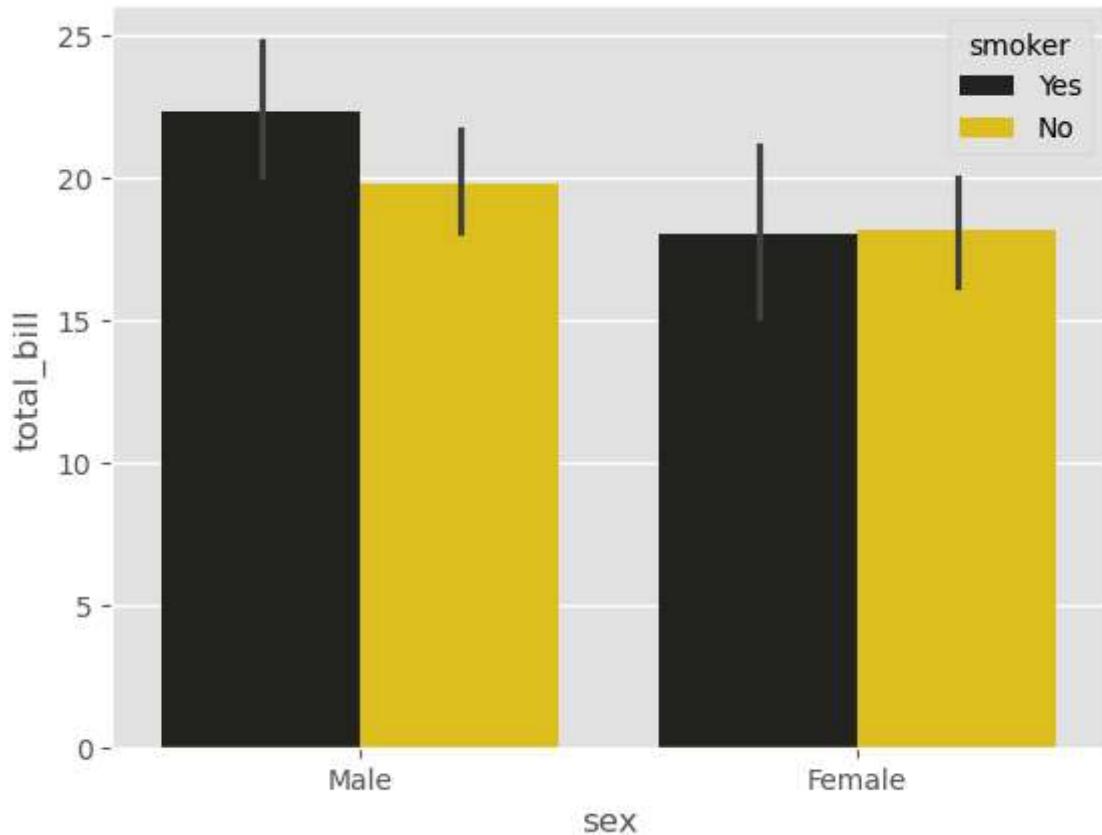
```
In [ ]: # hue
sns.barplot(data=tips, x='sex', y='total_bill',hue='smoker')
```

```
Out[ ]: <Axes: xlabel='sex', ylabel='total_bill'>
```



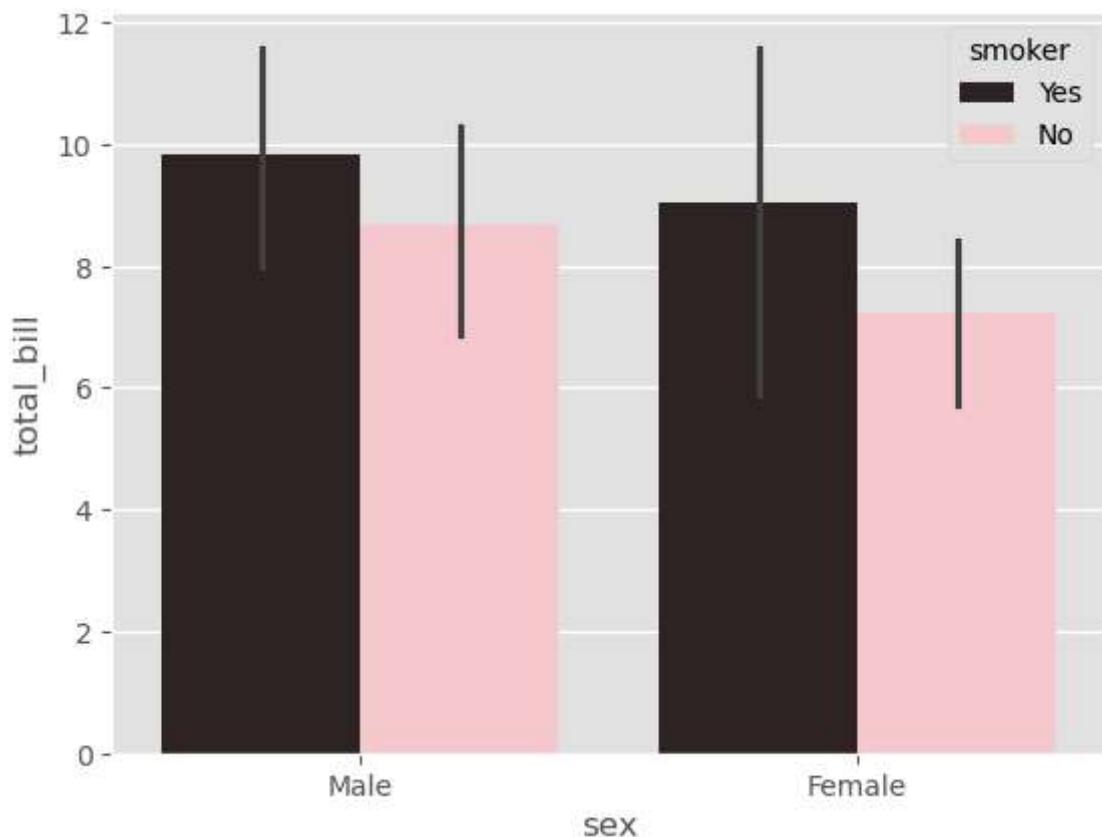
```
In [ ]: # color
sns.barplot(data=tips, x='sex', y='total_bill',hue='smoker',color = 'gold')
```

```
Out[ ]: <Axes: xlabel='sex', ylabel='total_bill'>
```



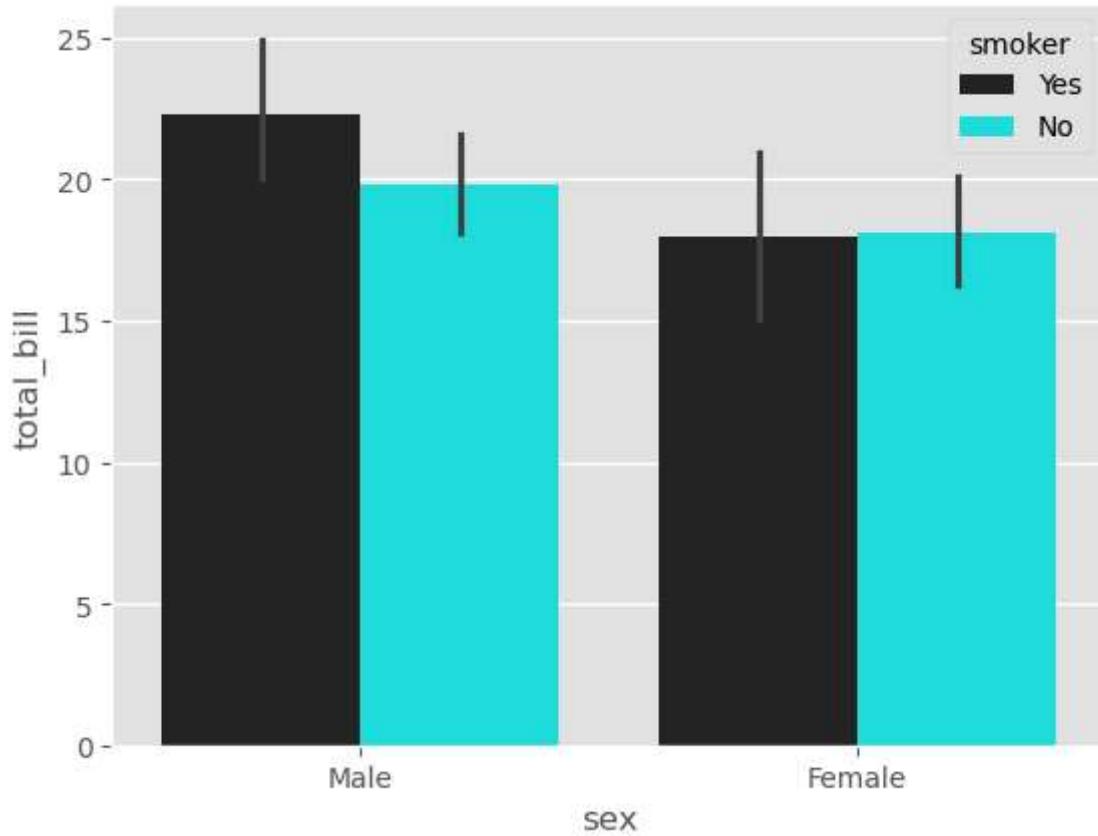
```
In [ ]: # estimator
import numpy as np
sns.barplot(data=tips, x='sex', y='total_bill', hue='smoker', color ='pink', esti
```

```
Out[ ]: <Axes: xlabel='sex', ylabel='total_bill'>
```



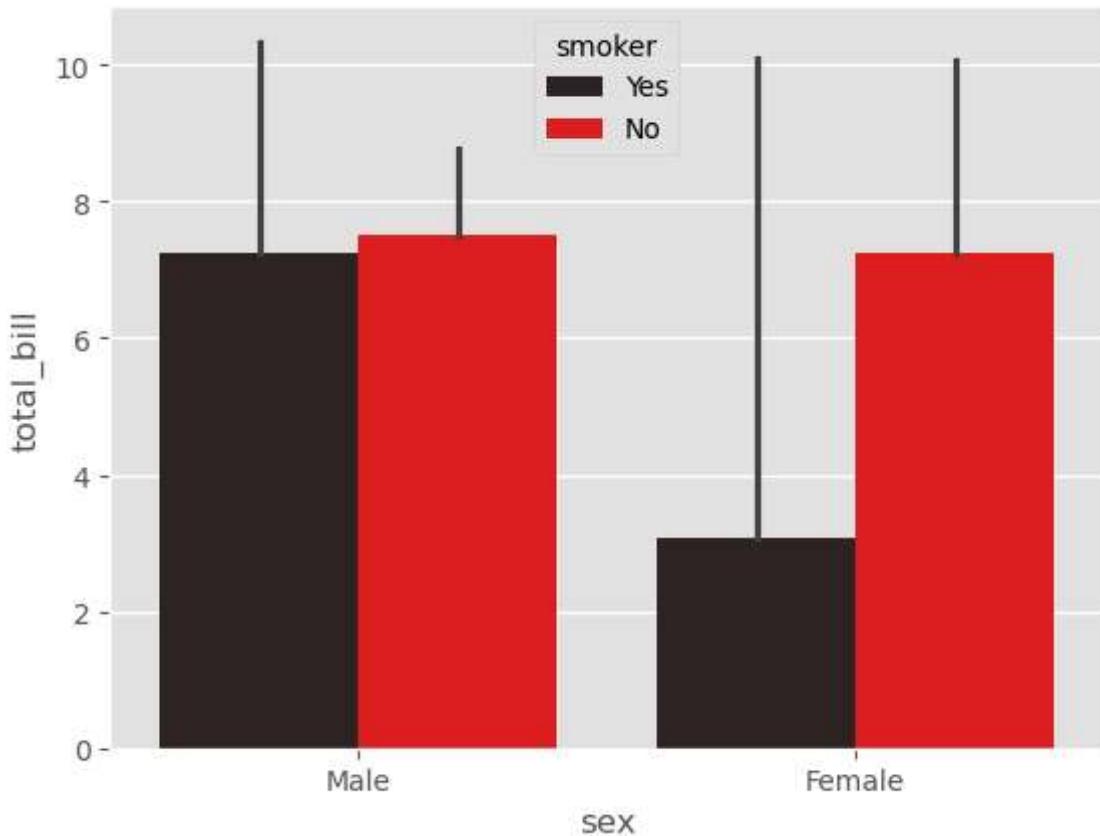
```
In [ ]: # estimator ( mean)
sns.barplot(data=tips, x='sex', y='total_bill',hue='smoker',color ='cyan',estima
```

```
Out[ ]: <Axes: xlabel='sex', ylabel='total_bill'>
```



```
In [ ]: # estimator ( min)
sns.barplot(data=tips, x='sex', y='total_bill',hue='smoker',color ='red',estimat
```

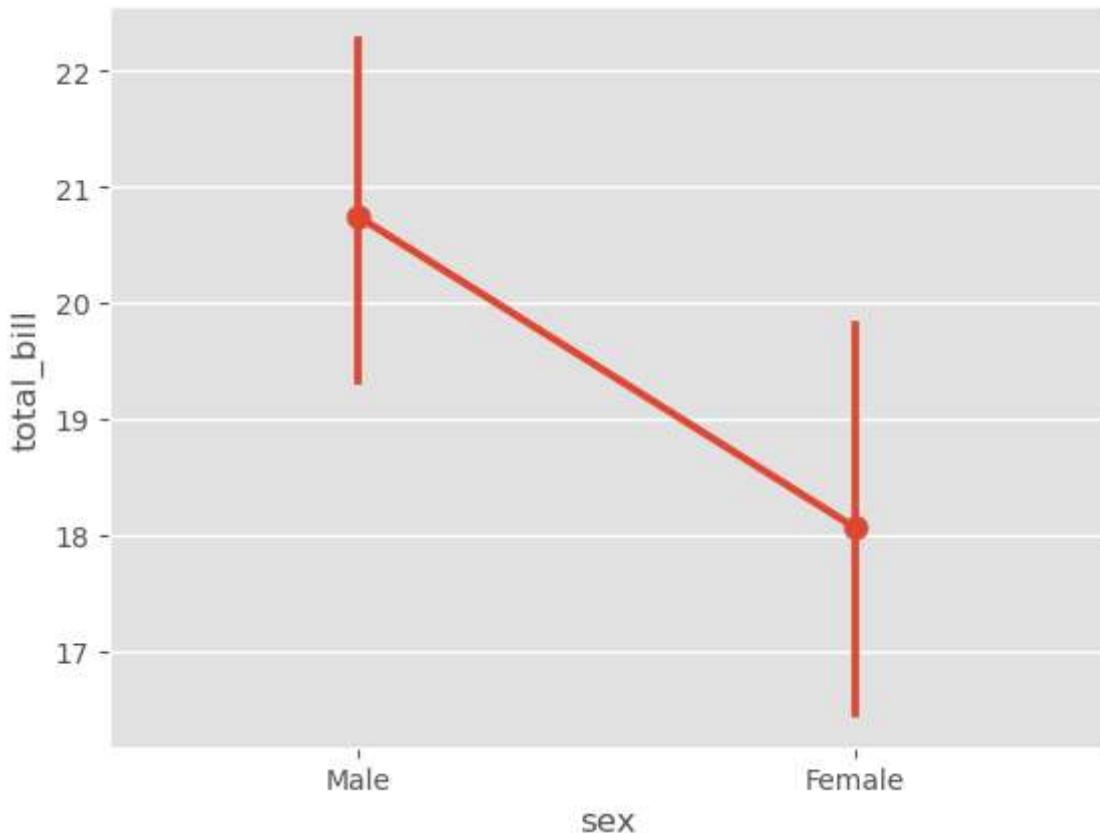
```
Out[ ]: <Axes: xlabel='sex', ylabel='total_bill'>
```



2. Point plot

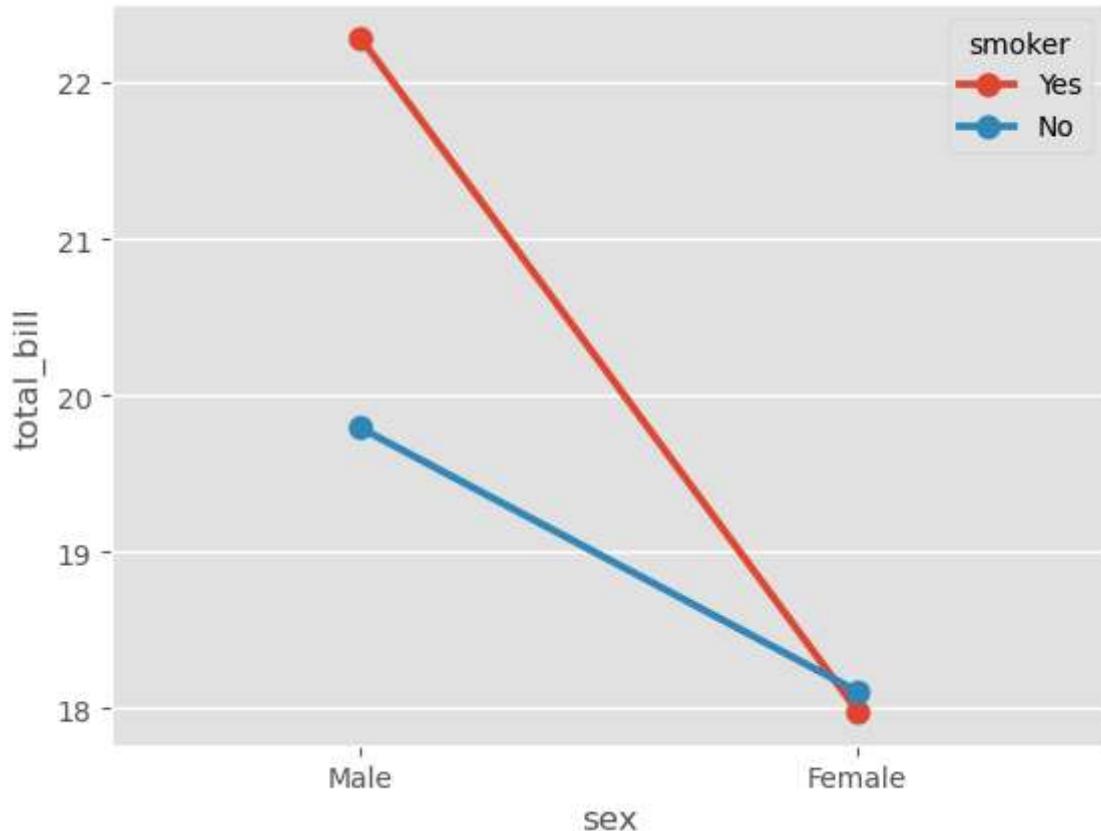
```
In [ ]: sns.pointplot(data=tips, x='sex', y='total_bill')
```

```
Out[ ]: <Axes: xlabel='sex', ylabel='total_bill'>
```



```
In [ ]: # CI ---> for removing error bars  
sns.pointplot(data=tips, x='sex', y='total_bill' , hue ='smoker', ci =None)
```

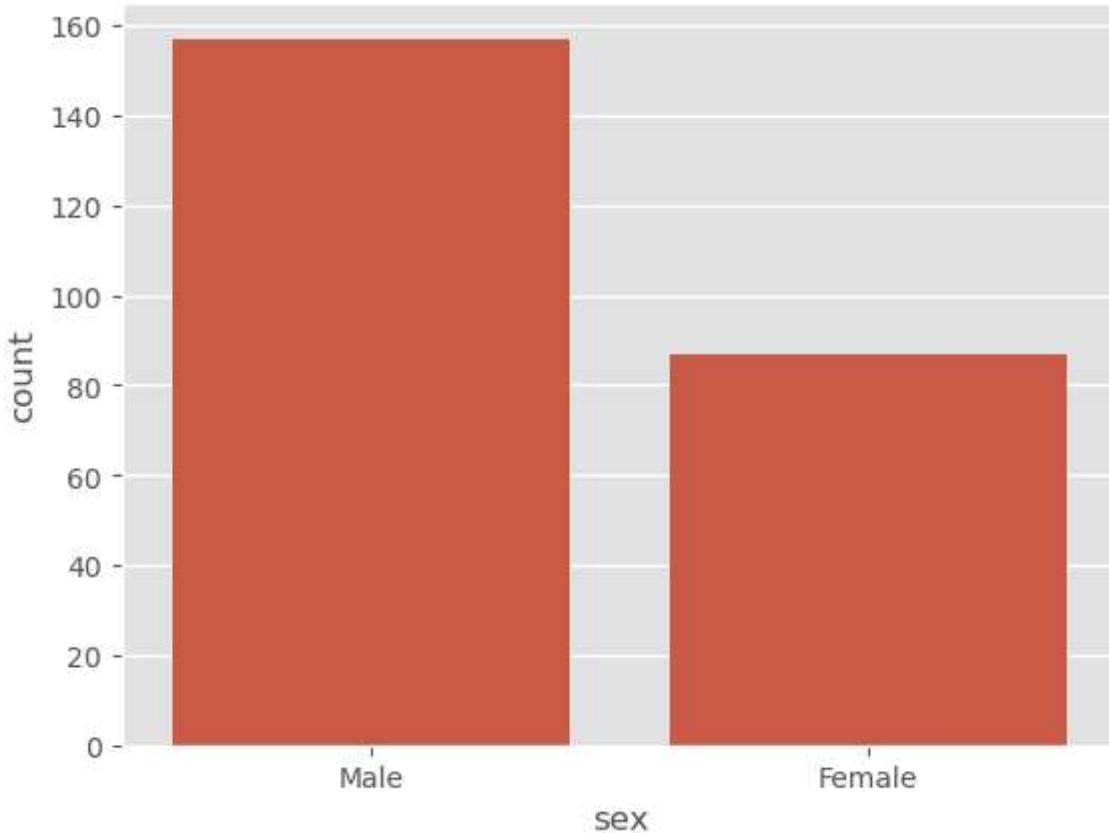
```
Out[ ]: <Axes: xlabel='sex', ylabel='total_bill'>
```



3. Count Plot

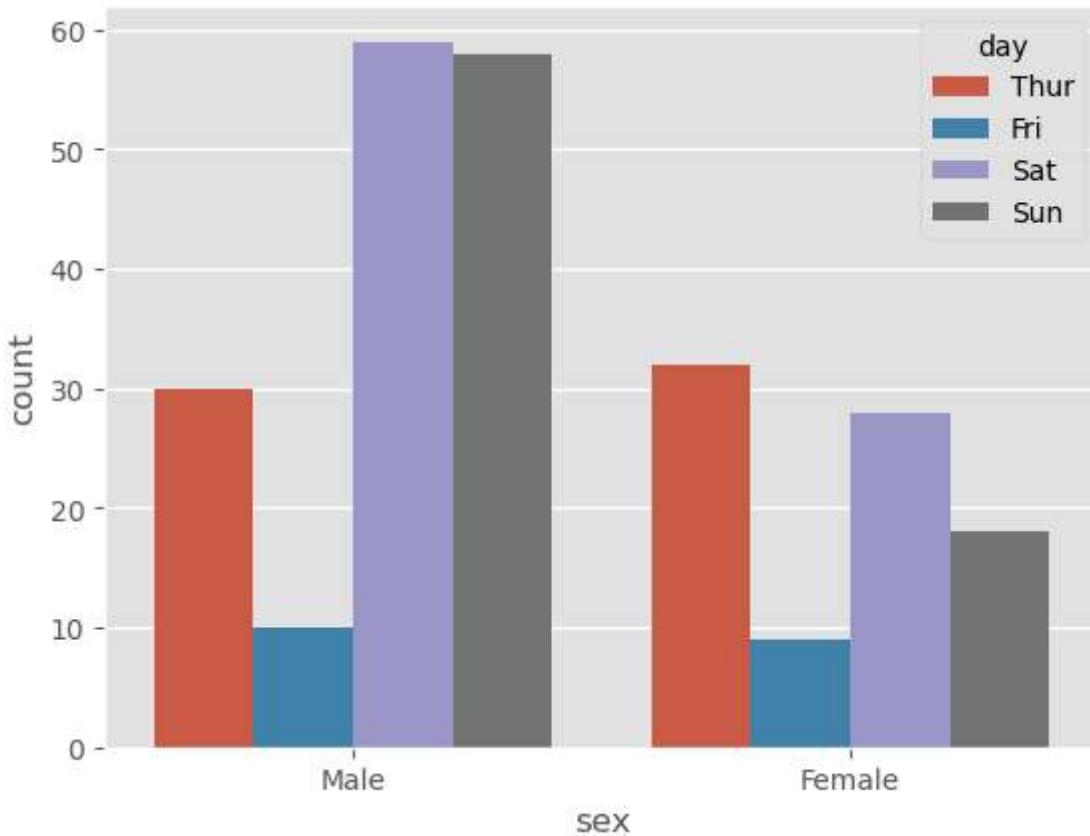
```
In [ ]: sns.countplot(data=tips,x='sex')
```

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



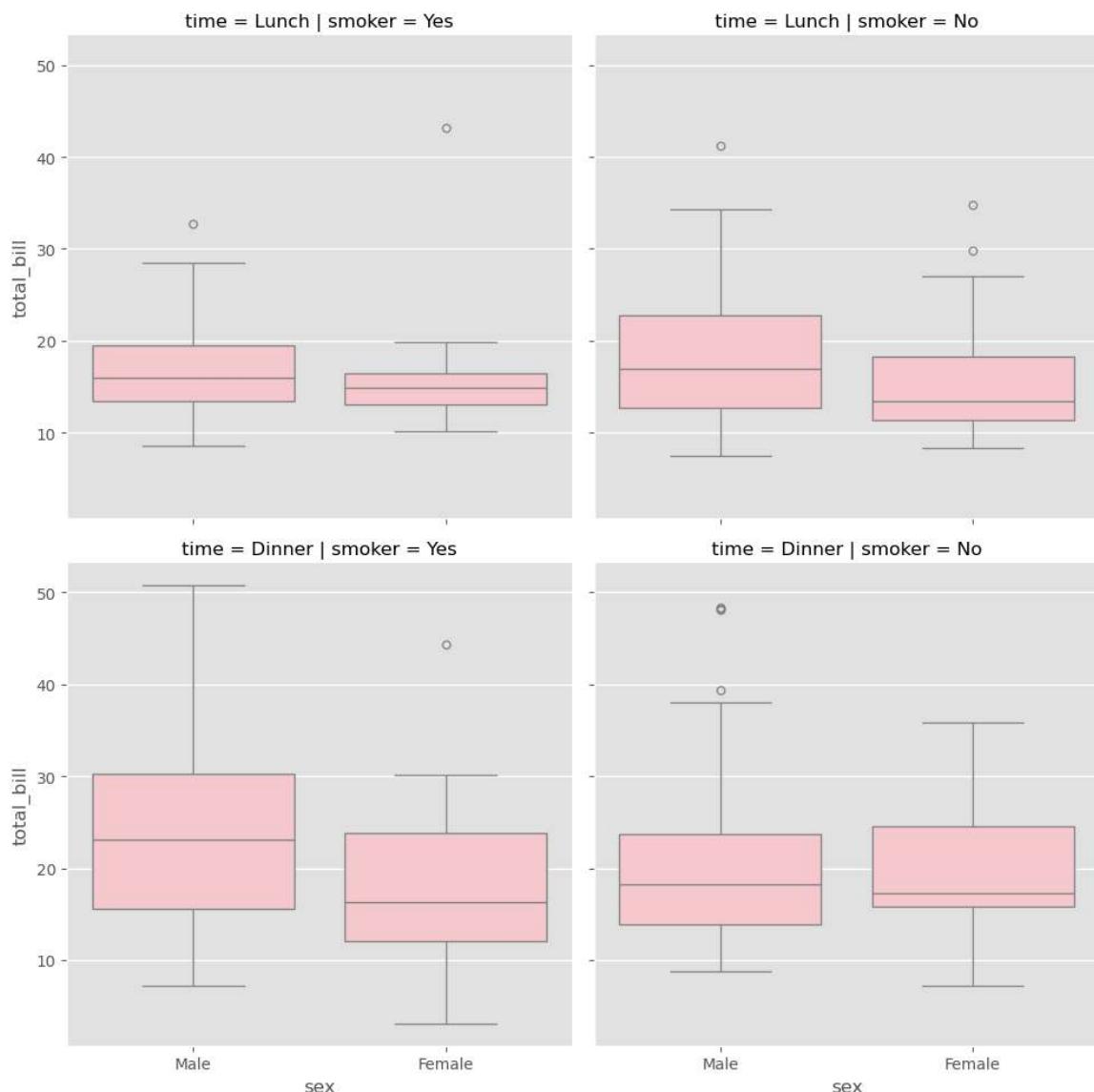
```
In [ ]: sns.countplot(data=tips,x='sex',hue='day')
```

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



```
In [ ]: # facetting using catplot
sns.catplot(data=tips, x='sex',y='total_bill',col='smoker',kind='box',row='time')
```

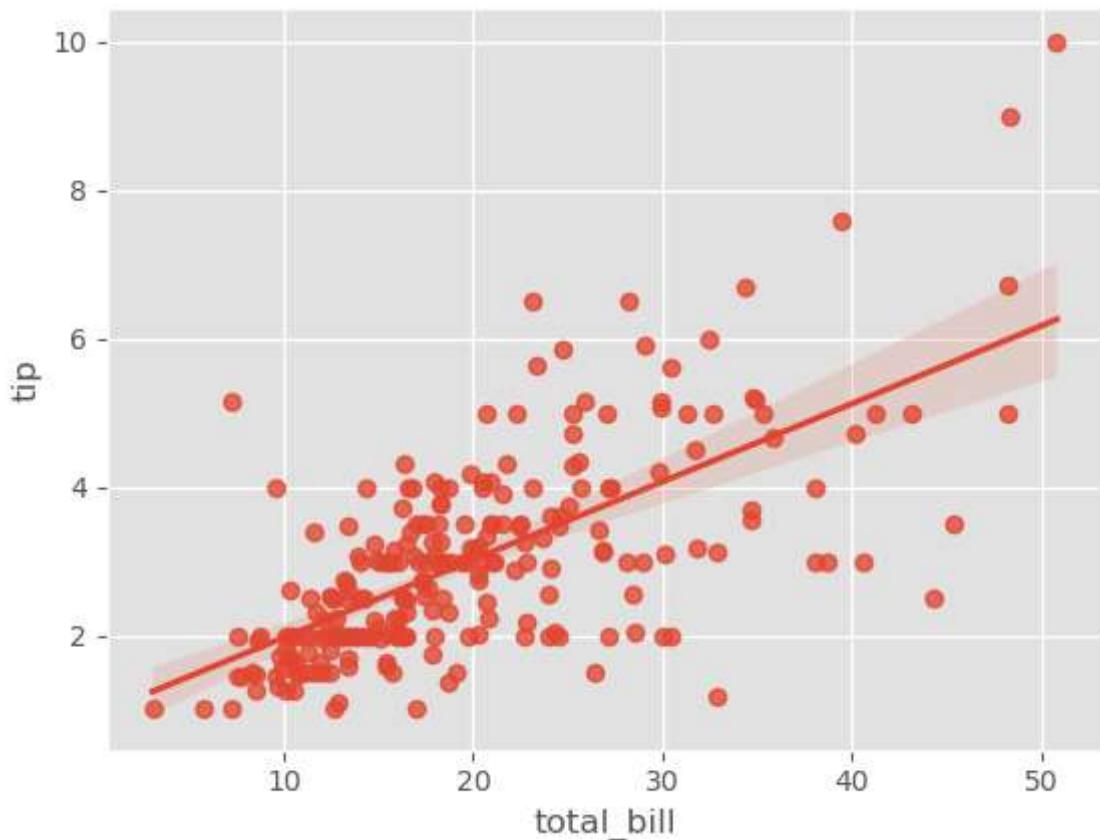
```
Out[ ]: <seaborn.axisgrid.FacetGrid at 0x1ec32b8a2d0>
```



Regression Plots

```
In [ ]: # axes_level  
# hue parameter is not available  
sns.regplot(data=tips,x='total_bill',y='tip')
```

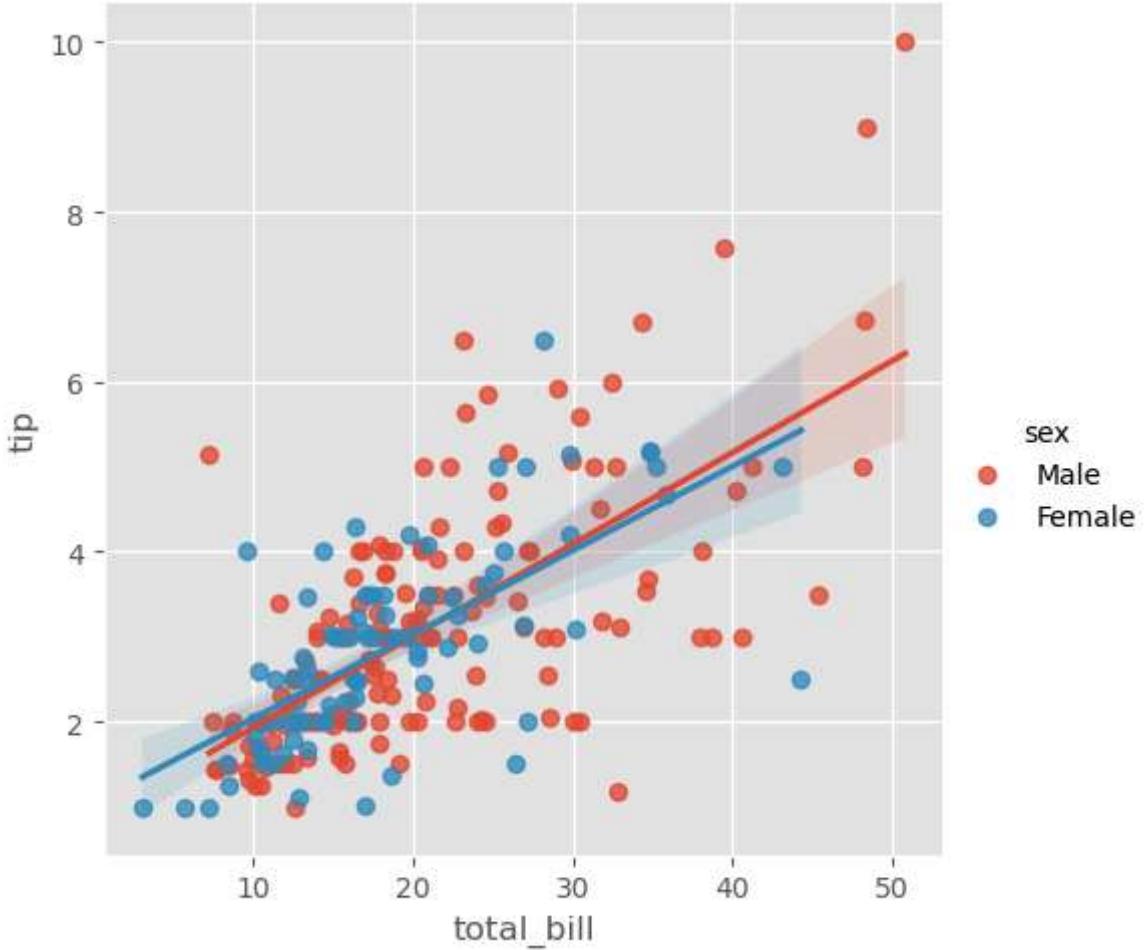
```
Out[ ]: <Axes: xlabel='total_bill', ylabel='tip'>
```



LM plot

```
In [ ]: sns.lmplot(data=tips,x='total_bill',y='tip', hue ='sex')
```

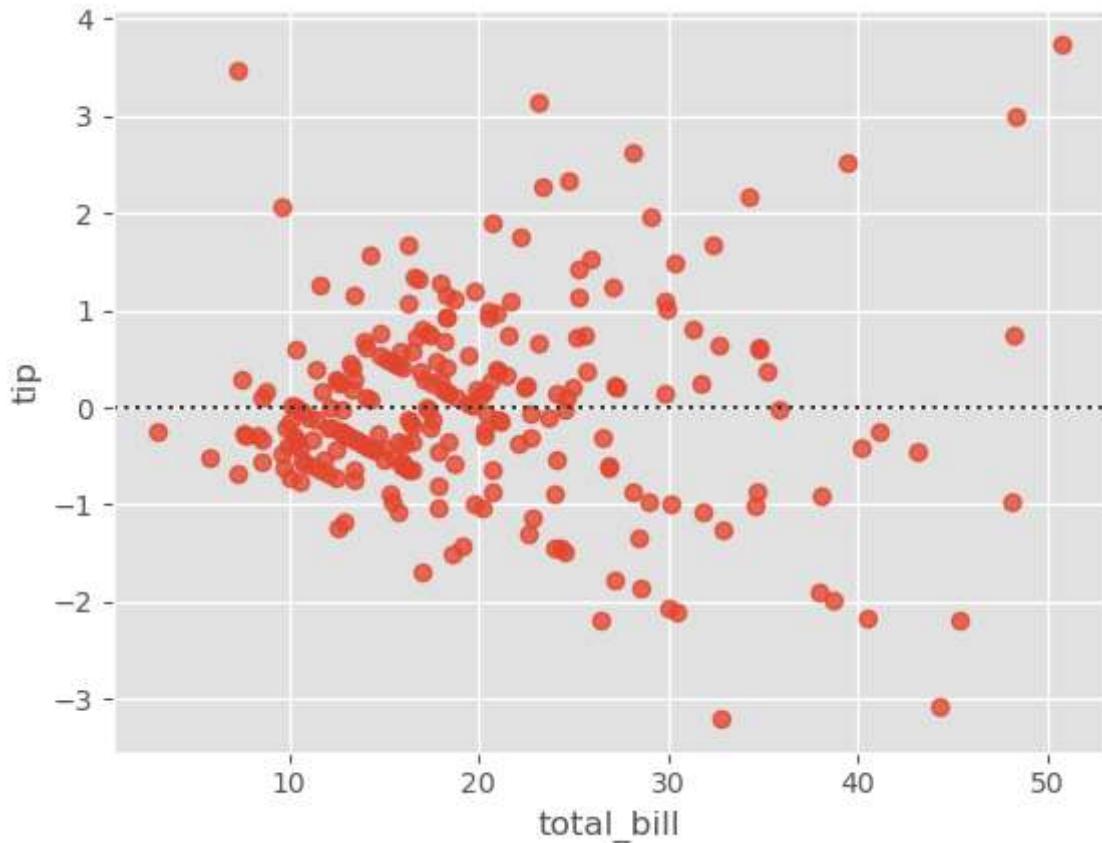
```
Out[ ]: <seaborn.axisgrid.FacetGrid at 0x1ec3298bad0>
```



residplot

```
In [ ]: # residual plot
sns.residplot(data=tips,x='total_bill',y='tip')
```

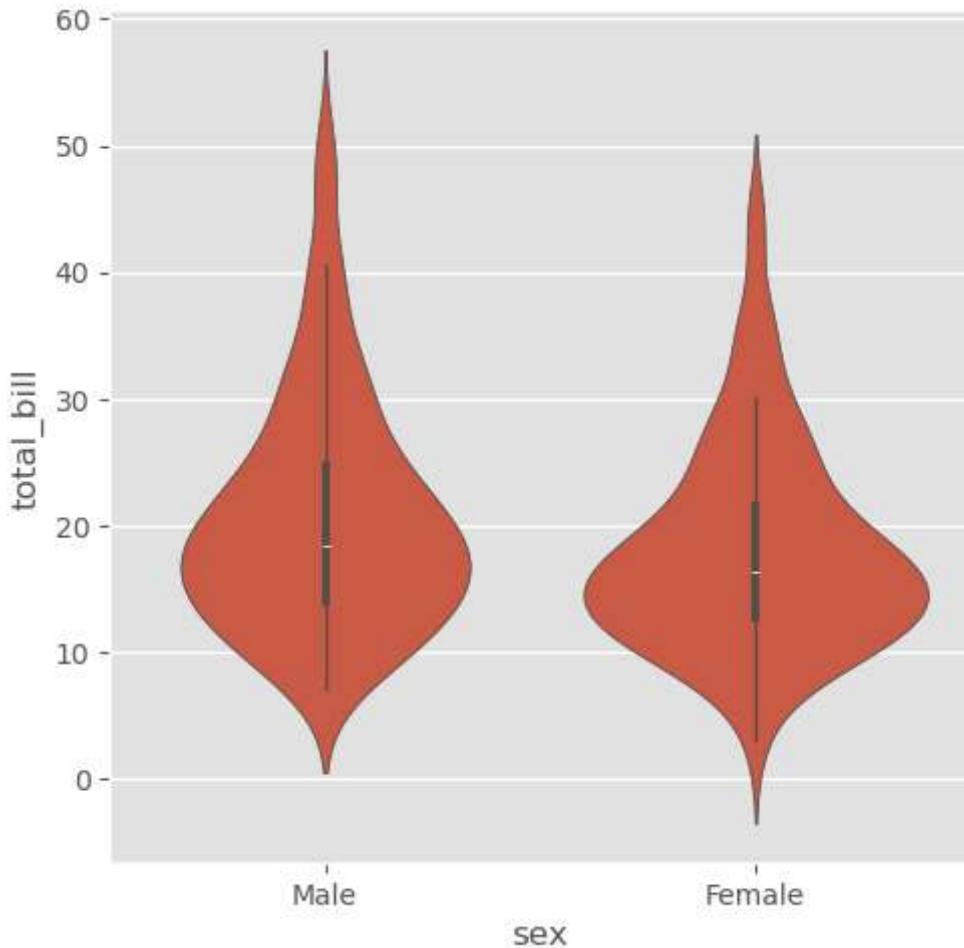
```
Out[ ]: <Axes: xlabel='total_bill', ylabel='tip'>
```



Second way to plot Facet plots -> FacetGrid

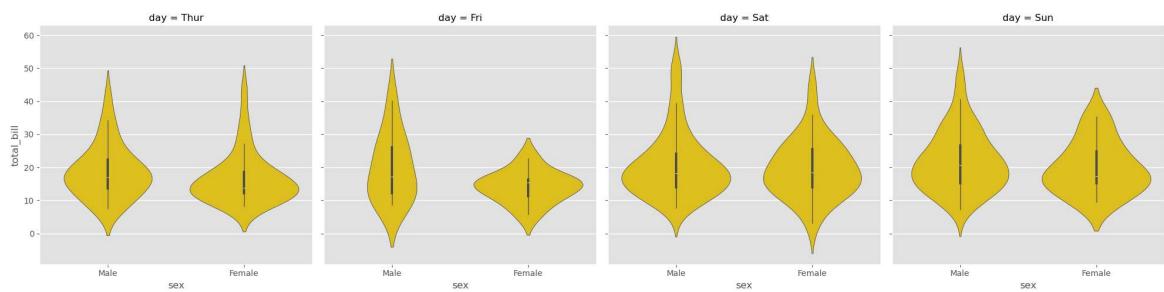
```
In [ ]: # figure Level -> relplot -> displot -> catplot -> Lmplot  
sns.catplot(data=tips,x='sex',y='total_bill',kind='violin')
```

```
Out[ ]: <seaborn.axisgrid.FacetGrid at 0x1ec34352ed0>
```



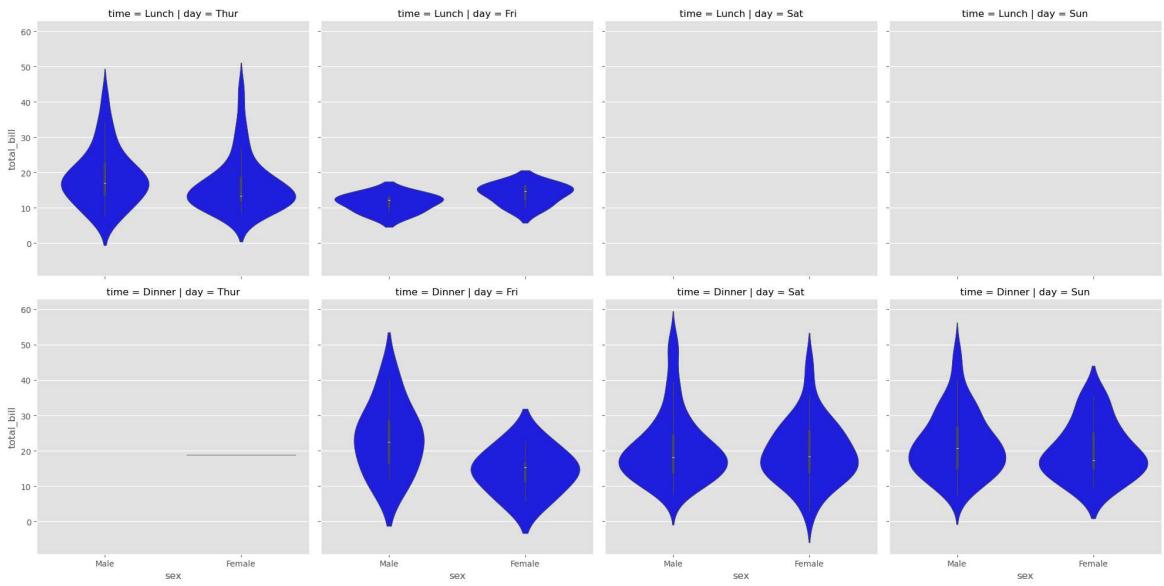
```
In [ ]: # Facet Plot - columns
sns.catplot(data=tips,x='sex',y='total_bill', color = 'gold',kind='violin',col='
```

```
Out[ ]: <seaborn.axisgrid.FacetGrid at 0x1ec34412b50>
```



```
In [ ]: # Facet Plot - column + rows
sns.catplot(data=tips,x='sex',y='total_bill',color ='blue' ,kind='violin',col='c
```

```
Out[ ]: <seaborn.axisgrid.FacetGrid at 0x1ec3427dfd0>
```



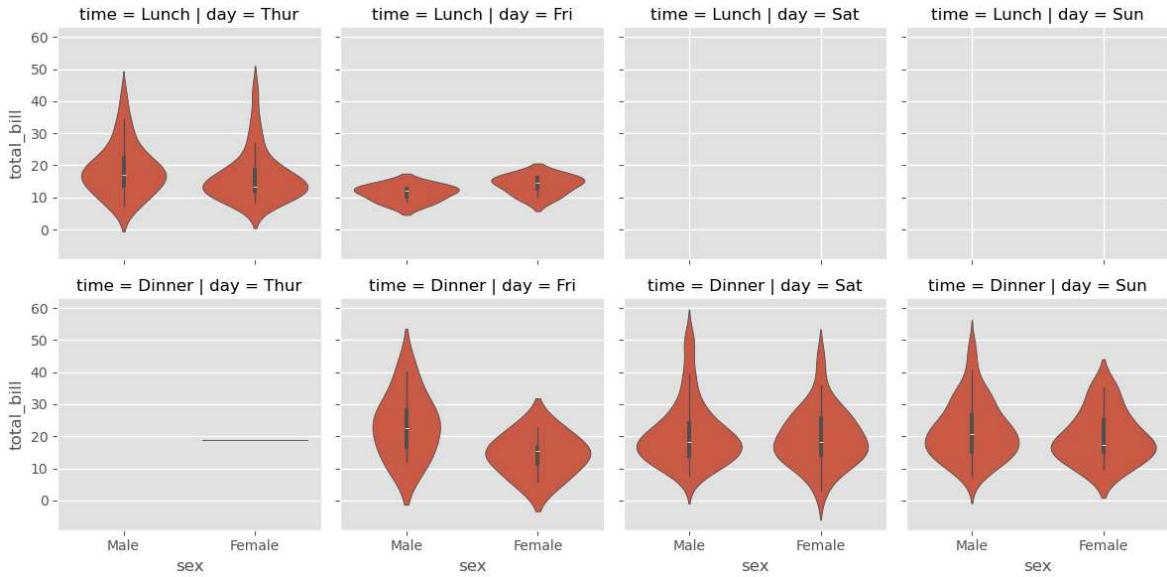
Facetgrid

```
In [ ]: # Second Method - Facetgrid
# it is the Lower Level code of catplot
secondmethod = sns.FacetGrid(data=tips, col='day', row='time')

# Define a plotting function to map to the FacetGrid
def violin_plot(x, y, **kwargs):
    sns.violinplot(x=x, y=y, **kwargs)

# Map the violinplot to the grid
secondmethod.map_dataframe(violin_plot, x='sex', y='total_bill')
```

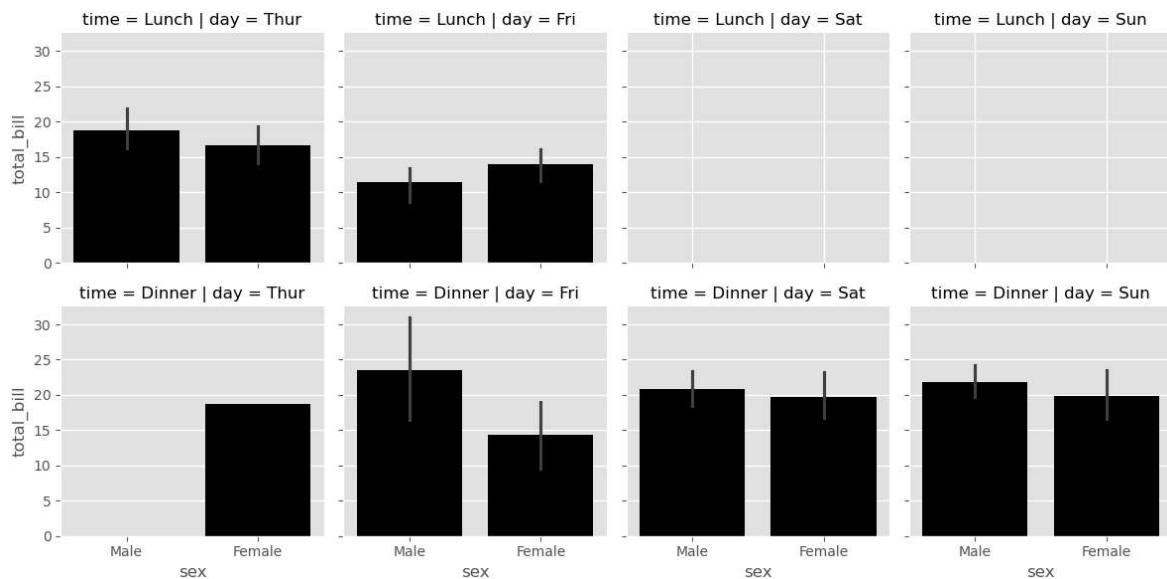
Out[]: <seaborn.axisgrid.FacetGrid at 0x1ec329b54d0>



```
In [ ]: secondmethod = sns.FacetGrid(data=tips, col='day', row='time')
#secondmethod.map(sns.barplot, 'sex', 'total_bill', color='black')
def bar_plot(x, y, **kwargs):
    sns.barplot(x=x, y=y, **kwargs)

# Map the barplot to the grid
secondmethod.map_dataframe(bar_plot, x='sex', y='total_bill', color='black')
```

```
Out[ ]: <seaborn.axisgrid.FacetGrid at 0x1ec351e5850>
```

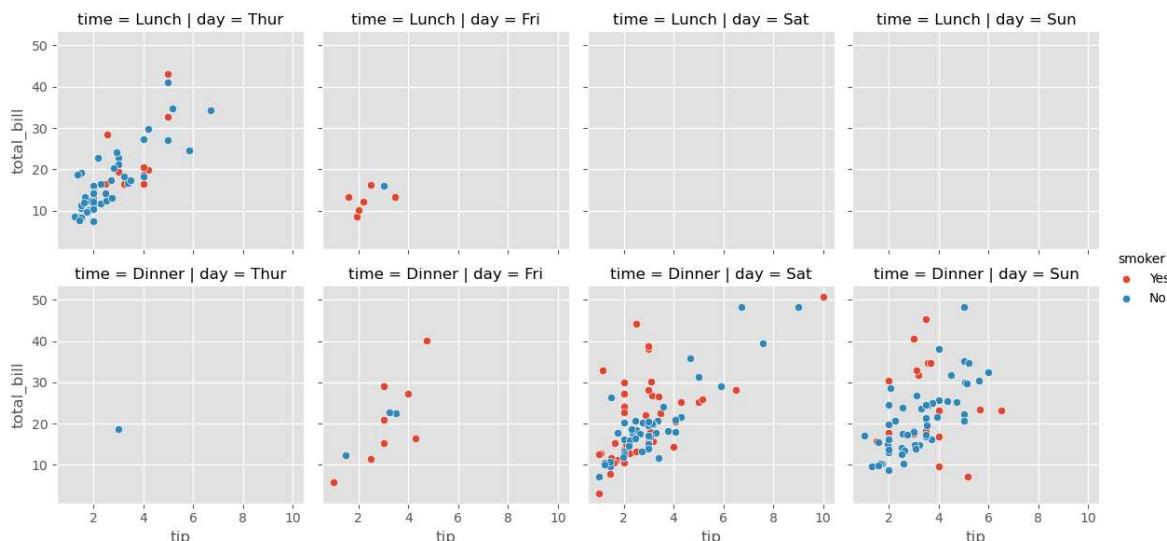


```
In [ ]: # Adding Legend
secondmethod = sns.FacetGrid(data=tips,col='day',row='time',hue='smoker')

def scatterplot(x, y, **kwargs):
    sns.scatterplot(x=x, y=y, **kwargs)

secondmethod.map_dataframe(scatterplot, x='tip', y='total_bill')
secondmethod.add_legend()
```

```
Out[ ]: <seaborn.axisgrid.FacetGrid at 0x1ec34f72a90>
```



Plotting Pairwise Relationship (PairGrid Vs Pairplot)

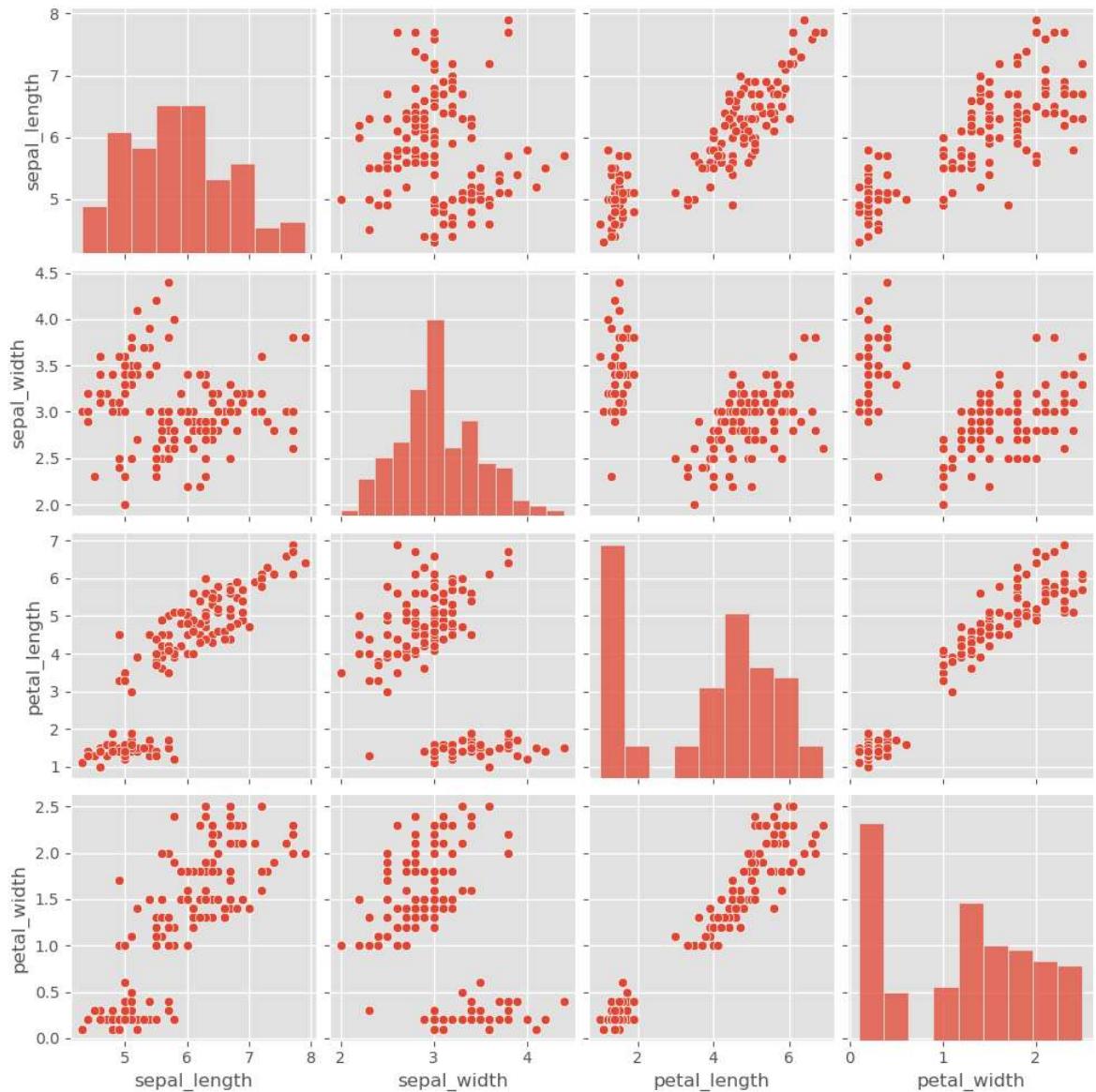
```
In [ ]: iris.sample()
```

```
Out[ ]:  sepal_length  sepal_width  petal_length  petal_width  species
```

| | sepal_length | sepal_width | petal_length | petal_width | species |
|----|--------------|-------------|--------------|-------------|------------|
| 67 | 5.8 | 2.7 | 4.1 | 1.0 | versicolor |

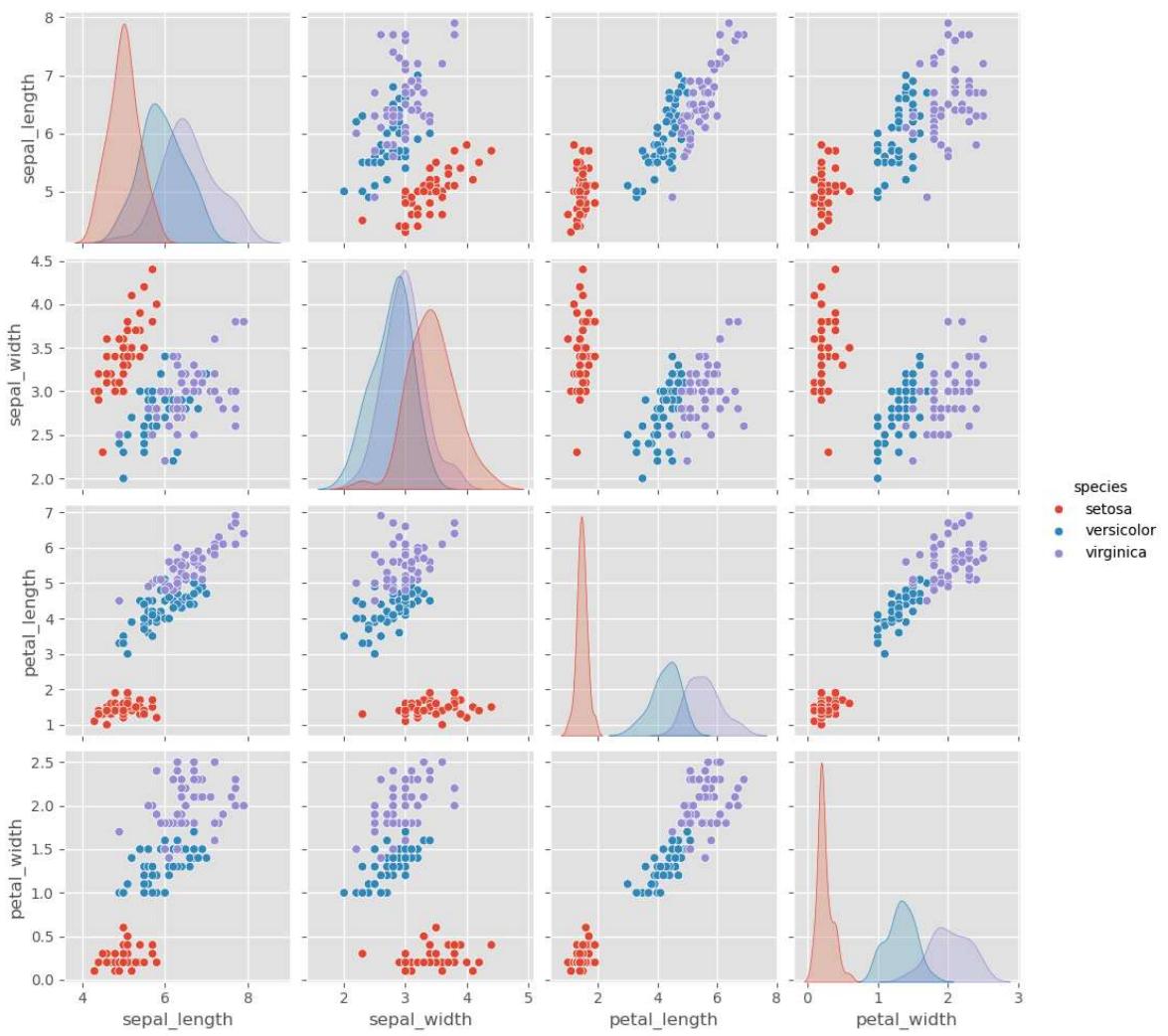
```
In [ ]: # Pairplot
sns.pairplot(iris)
```

```
Out[ ]: <seaborn.axisgrid.PairGrid at 0x1ec373572d0>
```



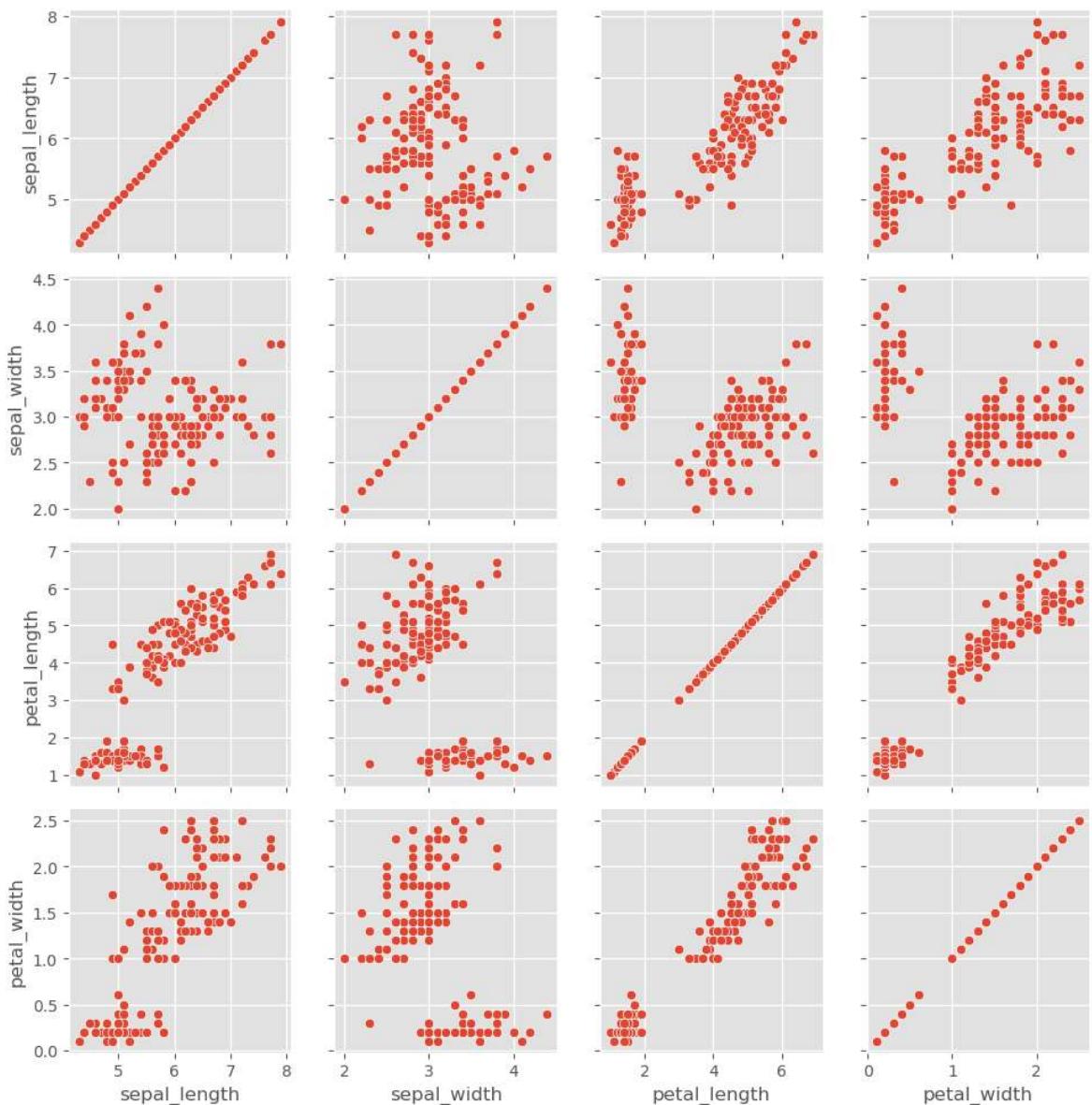
```
In [ ]: #hue  
sns.pairplot(iris,hue='species')
```

```
Out[ ]: <seaborn.axisgrid.PairGrid at 0x1ec38e98e90>
```

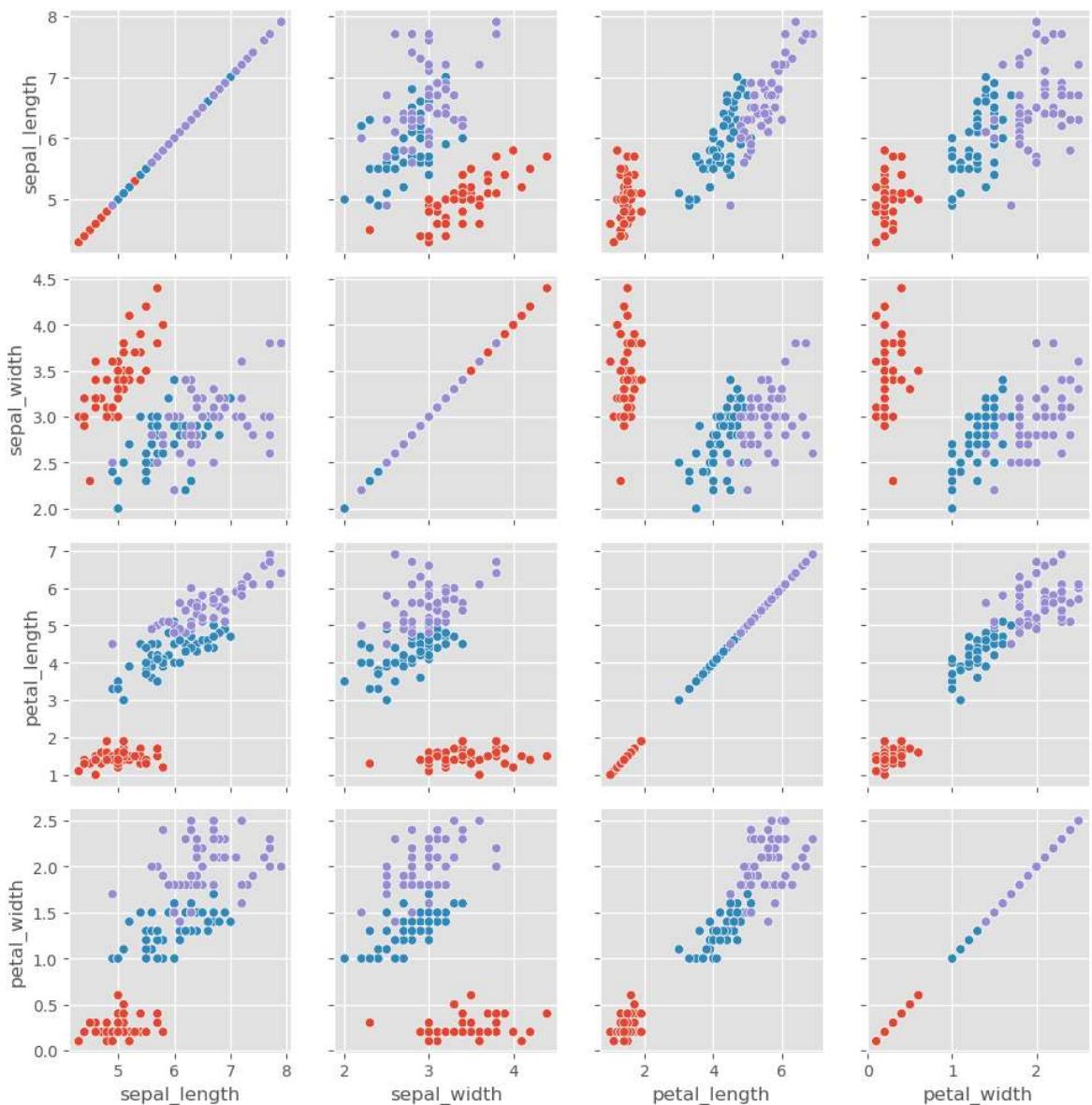


```
In [ ]: # pair grid
g = sns.PairGrid(data=iris )
# g.map
g.map(sns.scatterplot)
```

```
Out[ ]: <seaborn.axisgrid.PairGrid at 0x1ec3a4eab50>
```

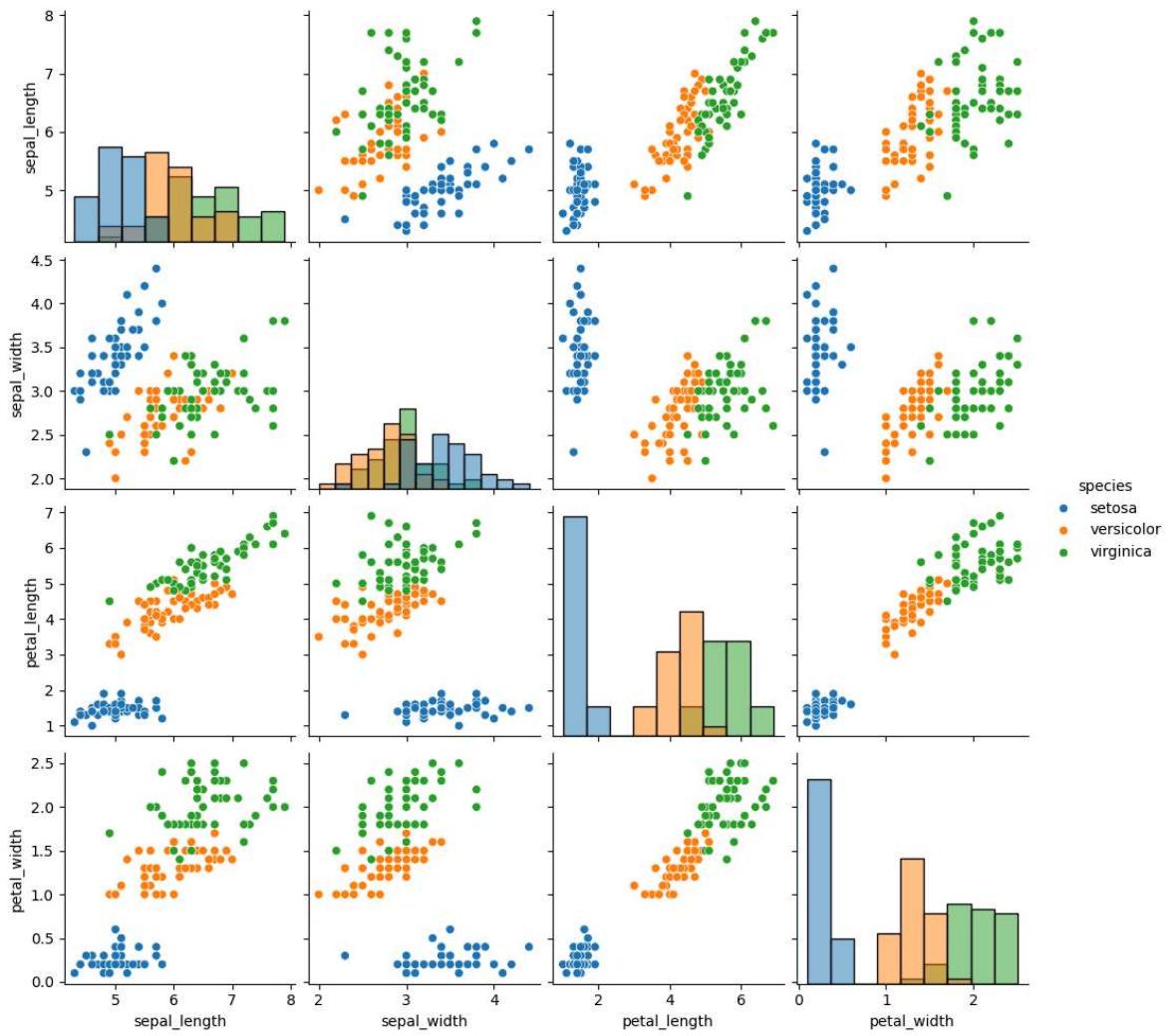


```
In [ ]: # hue
g = sns.PairGrid(data=iris,hue='species')
# g.map
g.map(sns.scatterplot)
# style
plt.style.use('default')
```



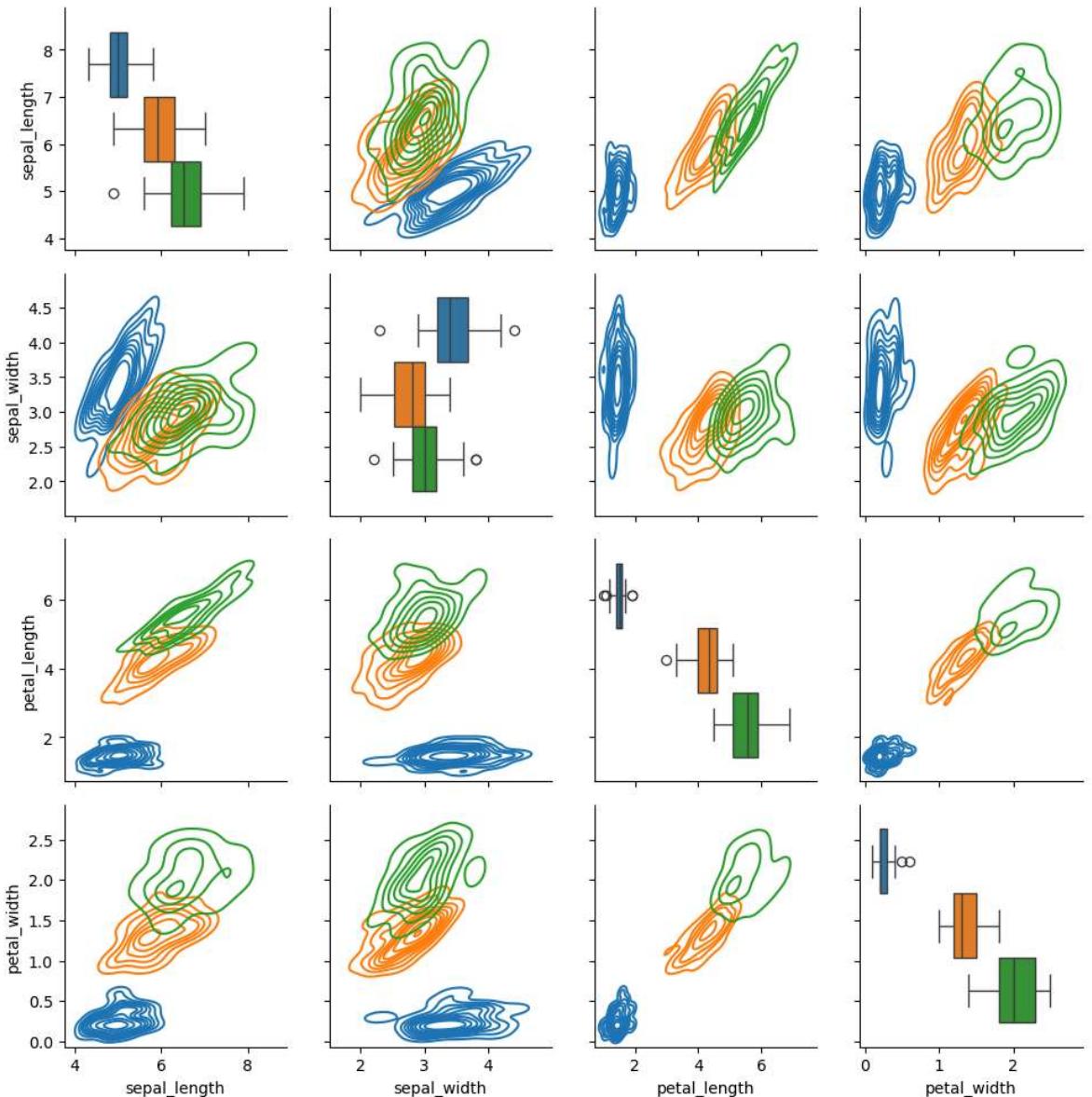
```
In [ ]: sns.pairplot(data=iris, hue='species', diag_kind='hist')
```

```
Out[ ]: <seaborn.axisgrid.PairGrid at 0x1ec3b075850>
```



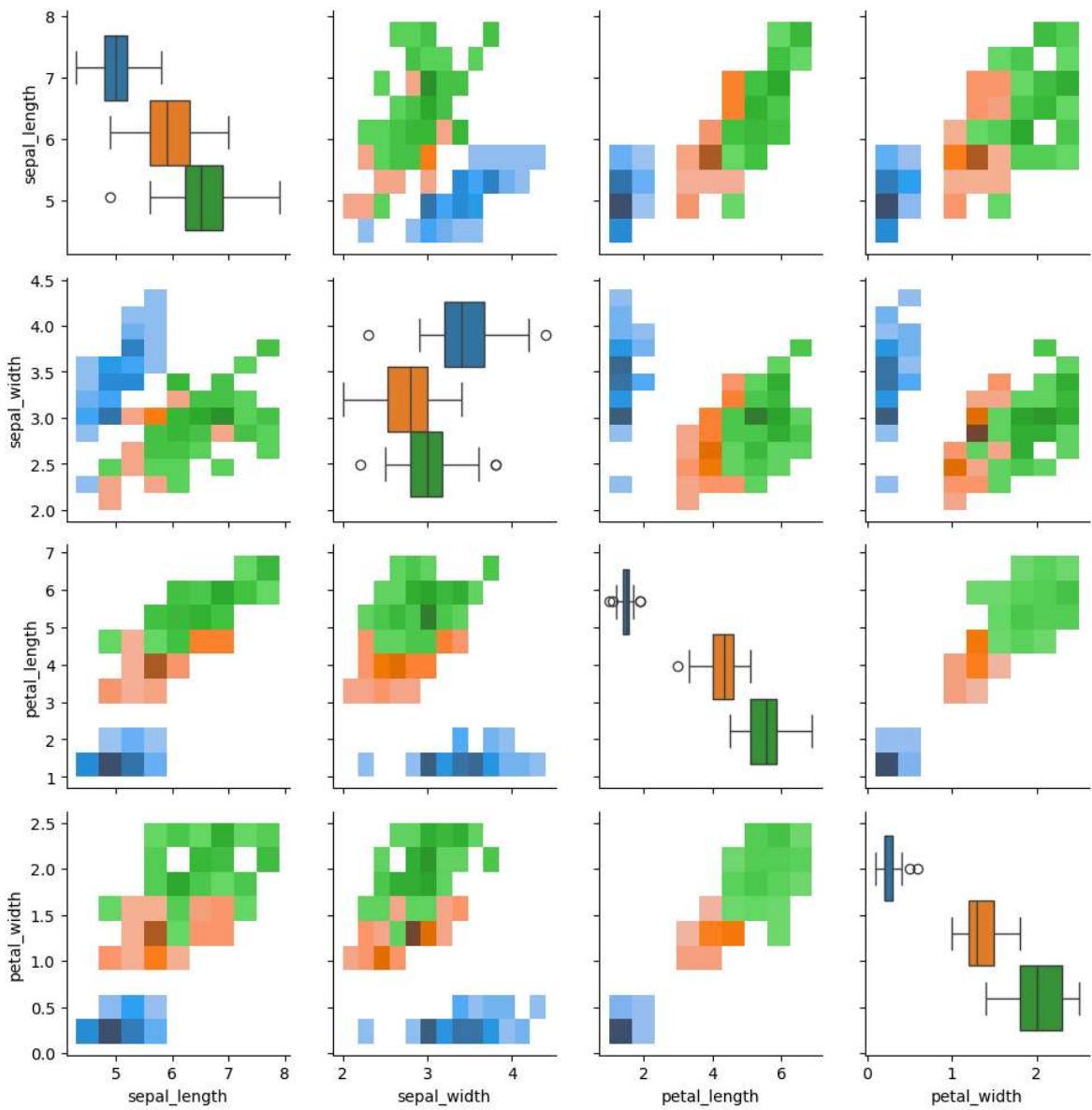
```
In [ ]: g = sns.PairGrid(data=iris,hue='species')
g.map_diag(sns.boxplot)
g.map_offdiag(sns.kdeplot)
```

```
Out[ ]: <seaborn.axisgrid.PairGrid at 0x1ec3e796f50>
```



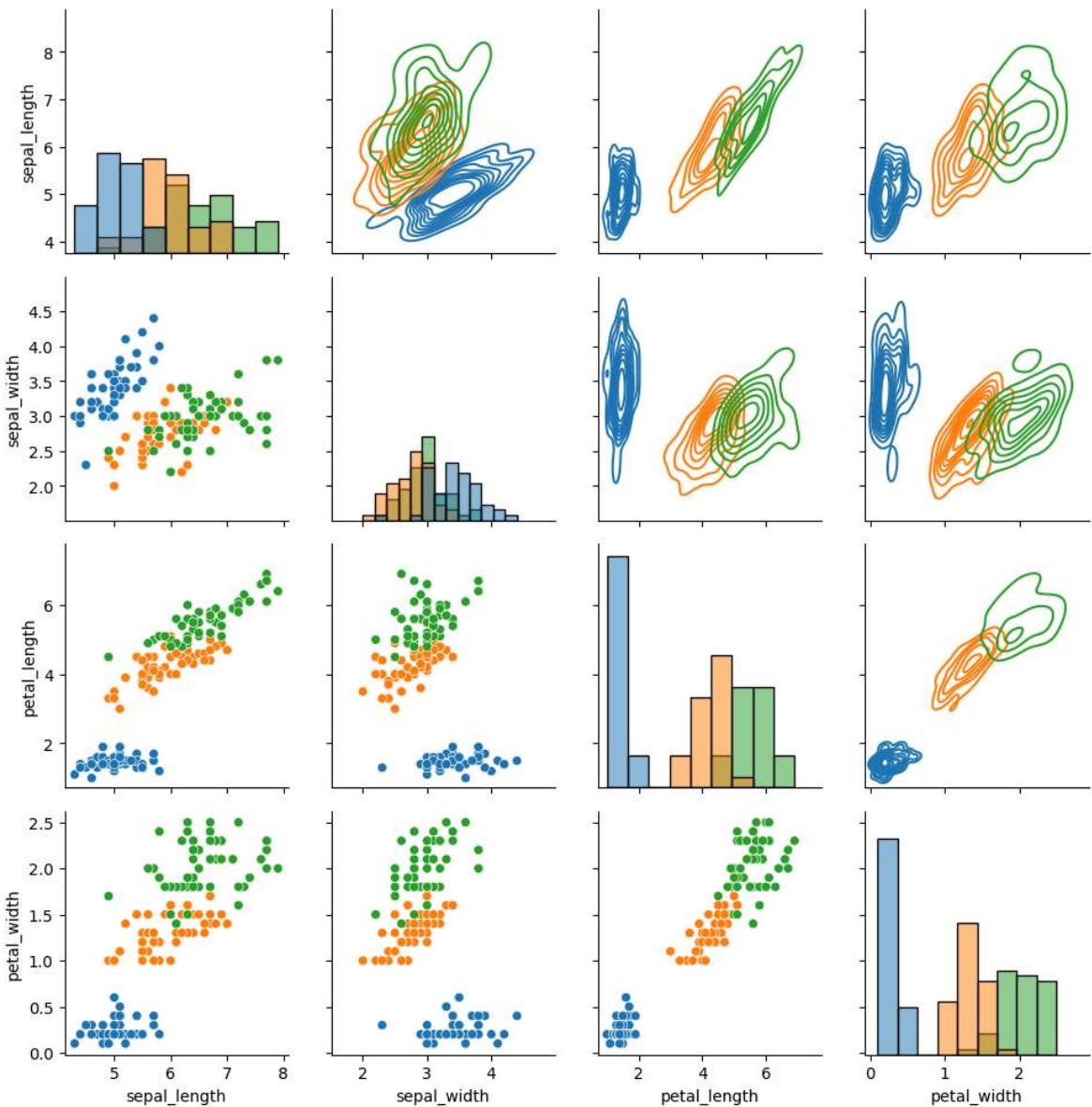
```
In [ ]: g = sns.PairGrid(data=iris,hue='species')
g.map_diag(sns.boxplot)
g.map_offdiag(sns.histplot)
```

```
Out[ ]: <seaborn.axisgrid.PairGrid at 0x1ec3f74dd10>
```



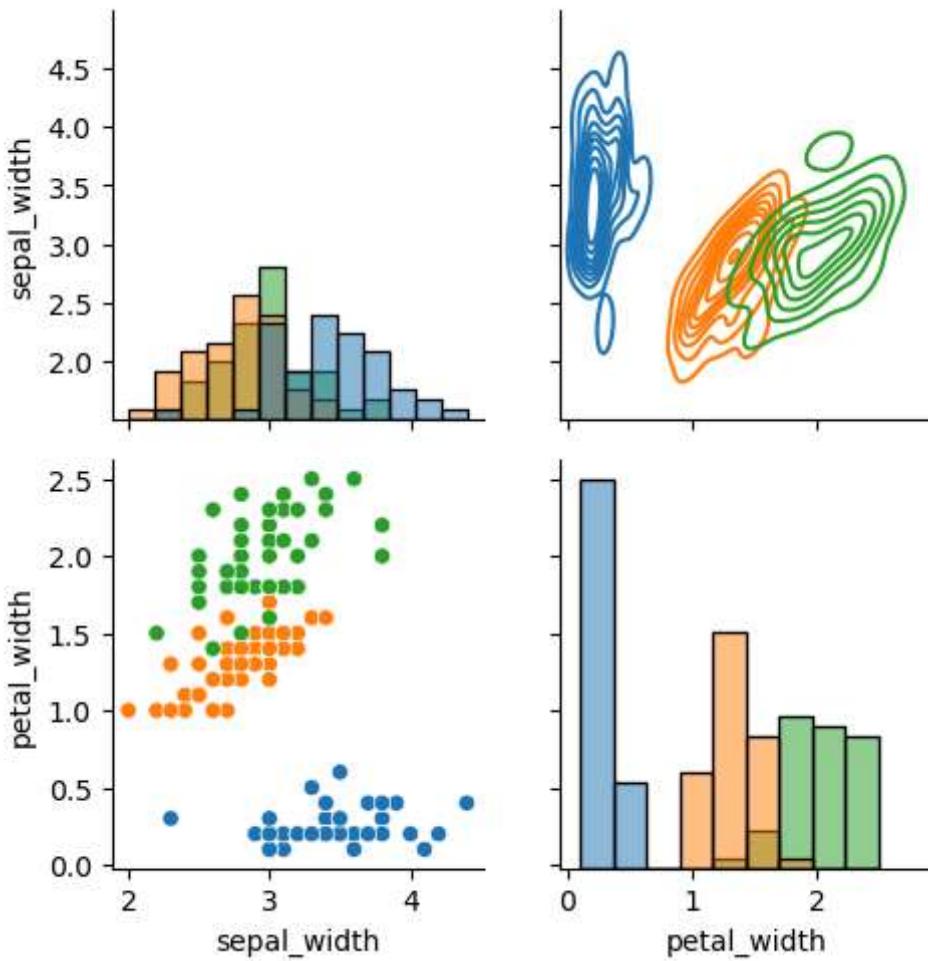
```
In [ ]: # map_diag -> map_upper -> map_lower
g = sns.PairGrid(data=iris,hue='species')
g.map_diag(sns.histplot) # diagonal
g.map_upper(sns.kdeplot) # upper
g.map_lower(sns.scatterplot) # Lower
```

```
Out[ ]: <seaborn.axisgrid.PairGrid at 0x1ec43683650>
```



```
In [ ]: # vars - working upon the desired columns
g = sns.PairGrid(data=iris,hue='species',vars=['sepal_width','petal_width'])
g.map_diag(sns.histplot)
g.map_upper(sns.kdeplot)
g.map_lower(sns.scatterplot)
```

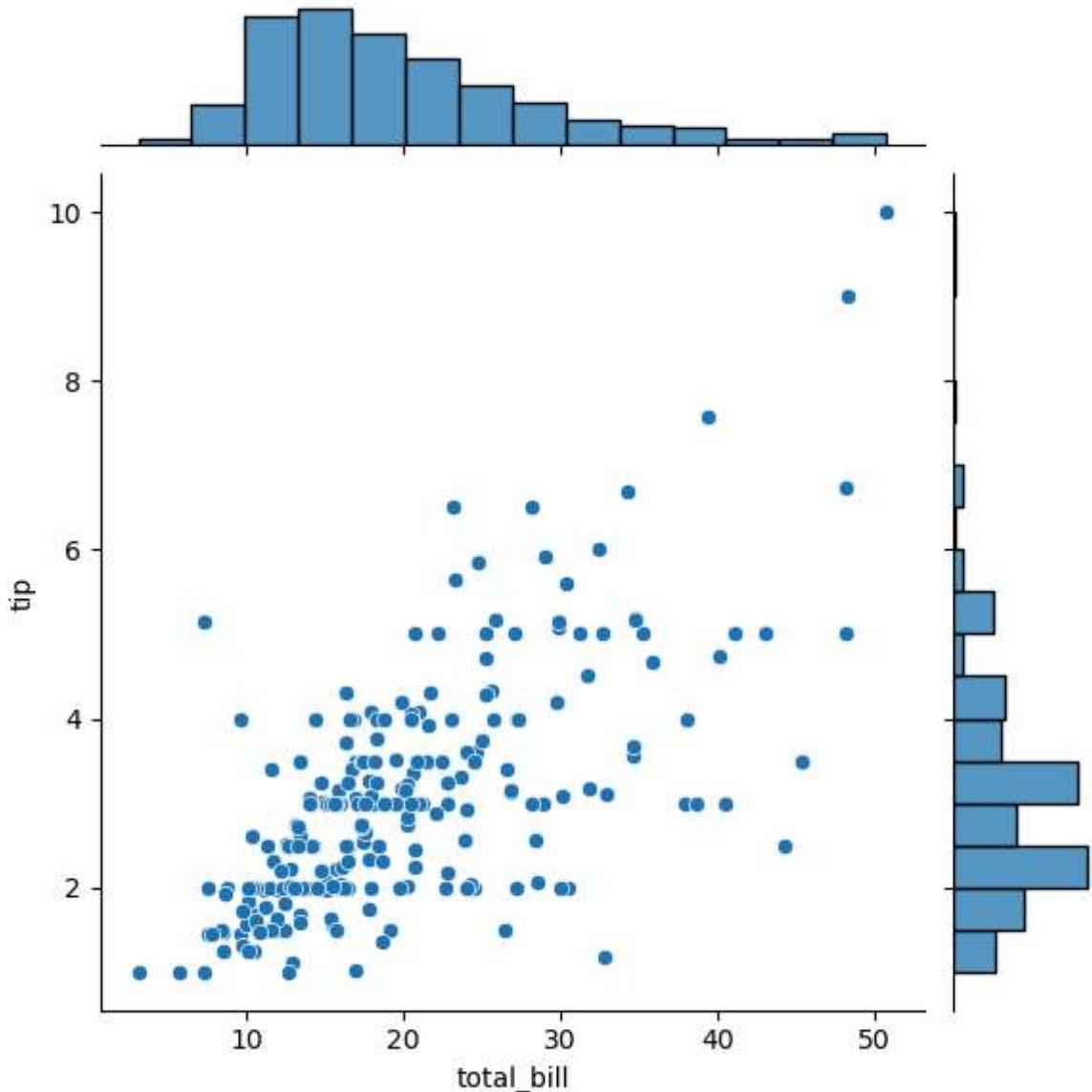
```
Out[ ]: <seaborn.axisgrid.PairGrid at 0x1ec408d37d0>
```



JointGrid Vs Jointplot

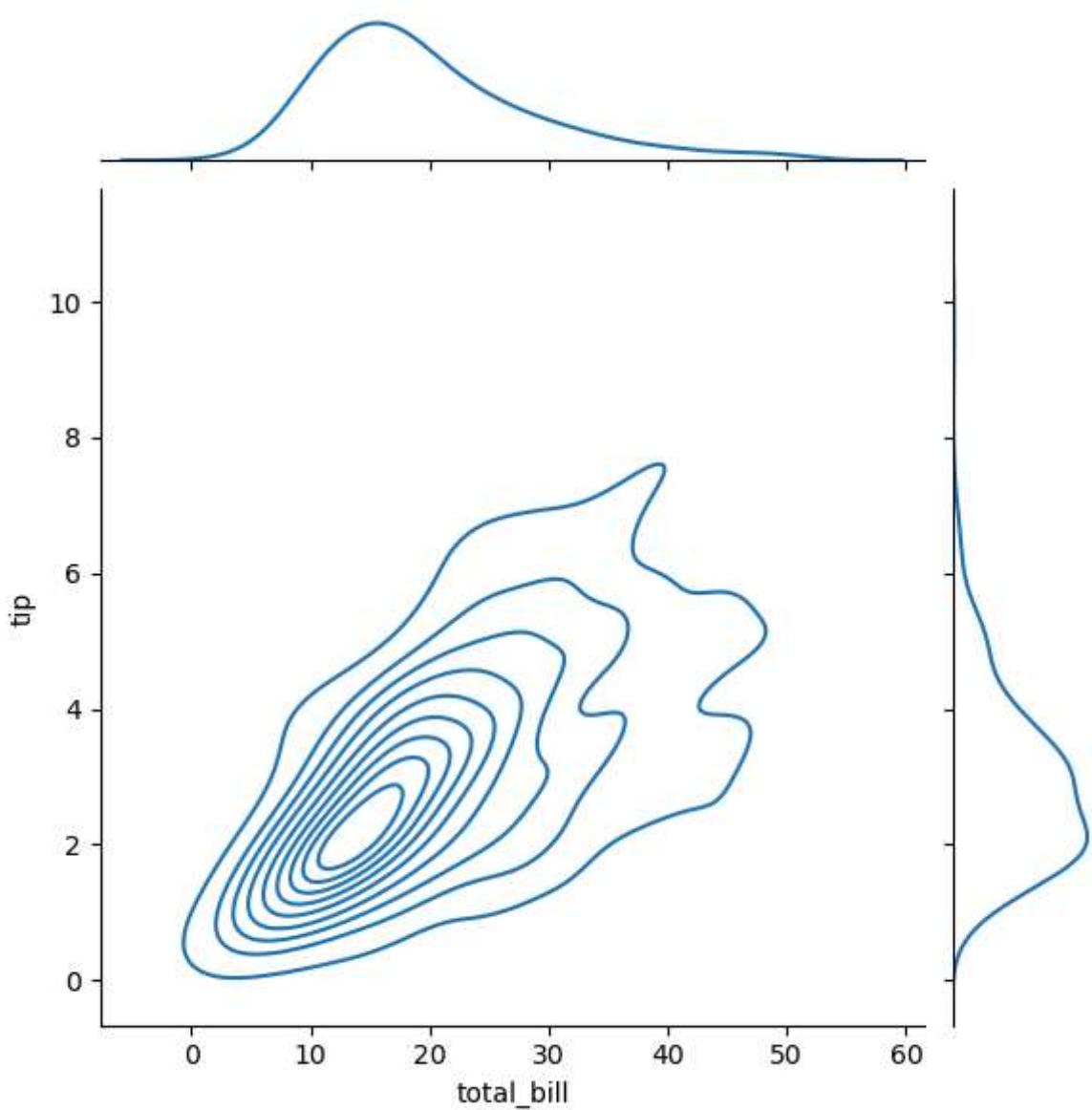
```
In [ ]: sns.jointplot(data=tips,x='total_bill',y='tip')
```

```
Out[ ]: <seaborn.axisgrid.JointGrid at 0x1ec4382ee50>
```



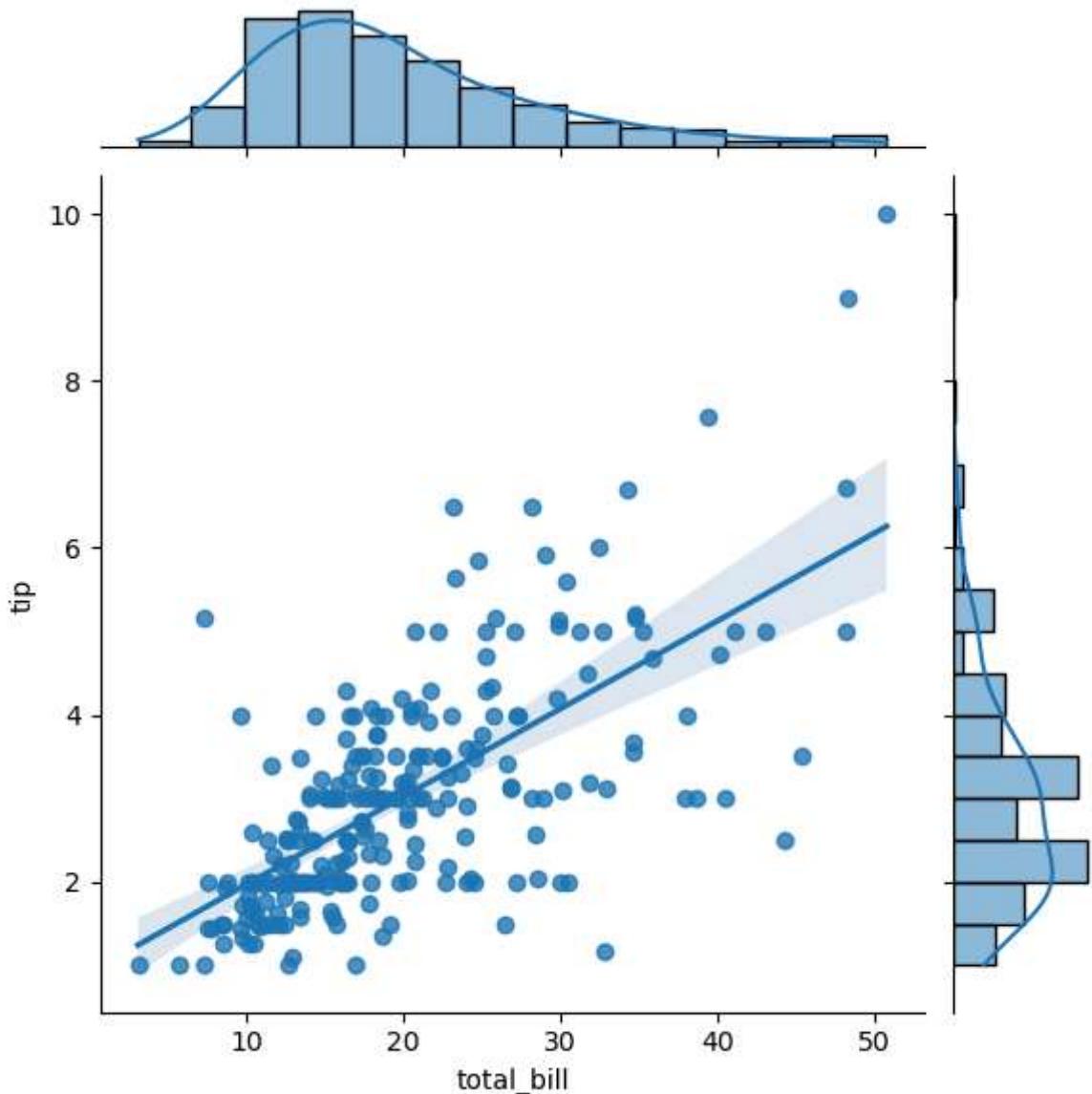
```
In [ ]: # kind
sns.jointplot(data=tips,x='total_bill',y='tip',kind= 'kde' )
```

```
Out[ ]: <seaborn.axisgrid.JointGrid at 0x1ec4612d610>
```



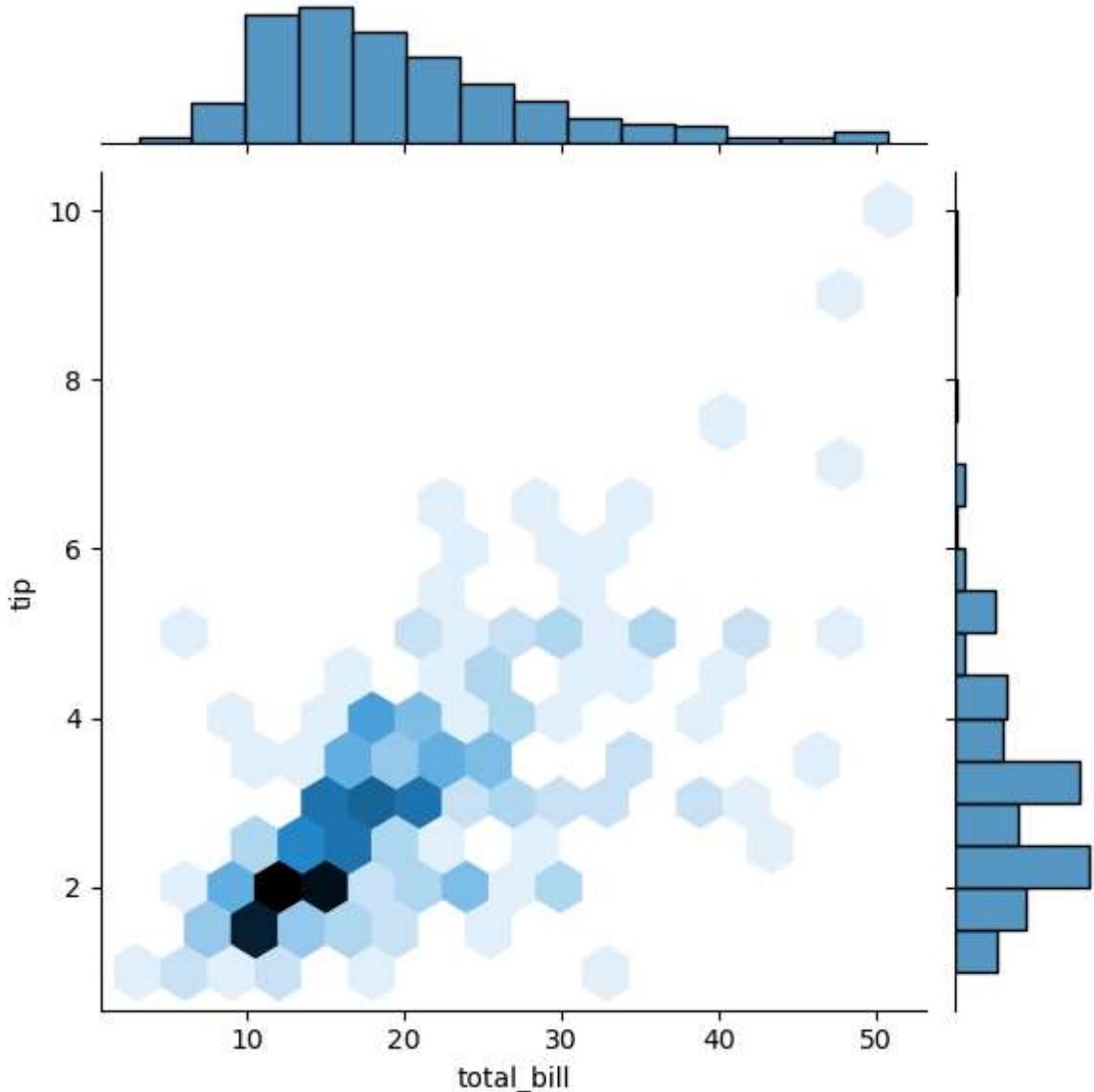
```
In [ ]: #regression Line  
sns.jointplot(data=tips,x='total_bill',y='tip',kind= 'reg' )
```

```
Out[ ]: <seaborn.axisgrid.JointGrid at 0x1ec462eeb90>
```



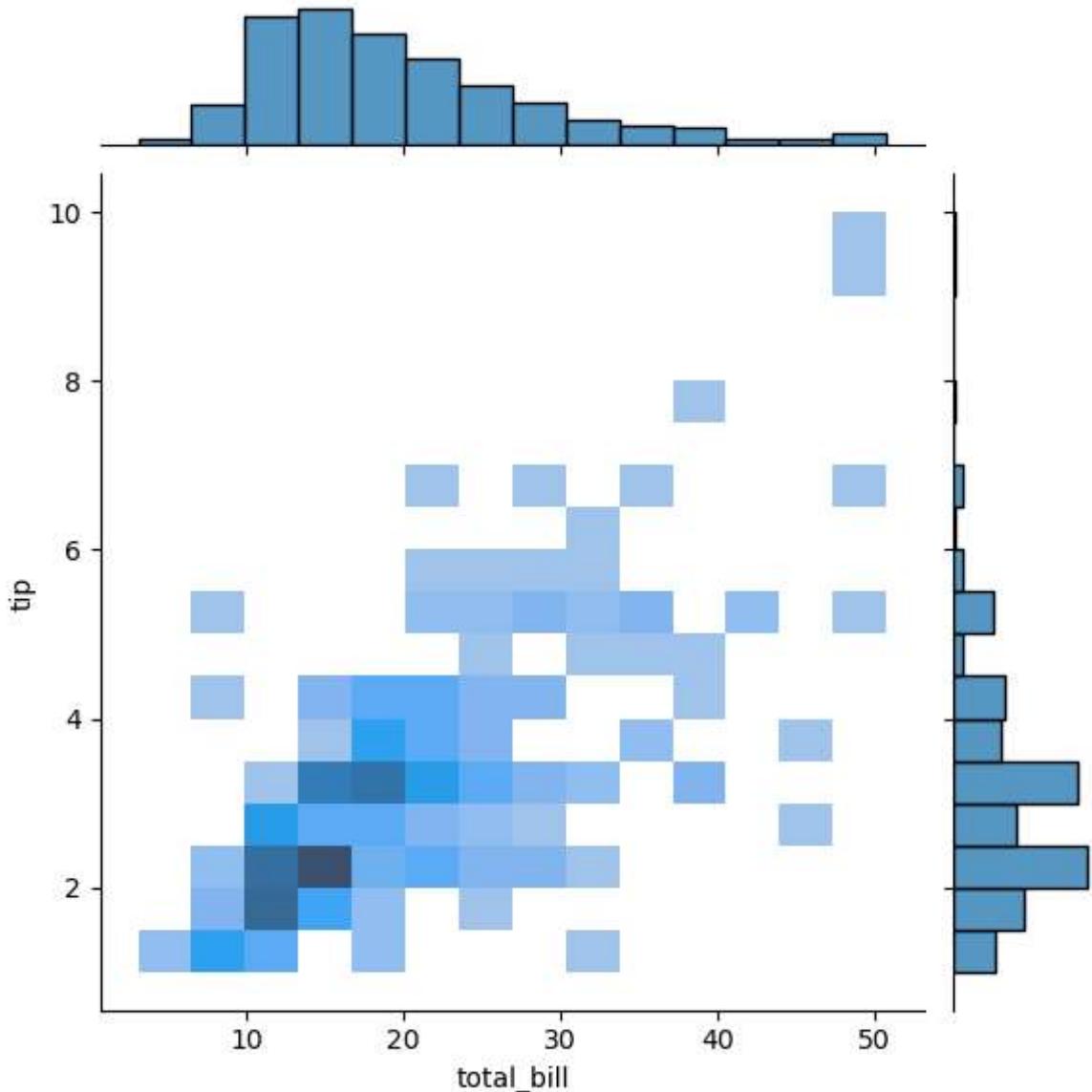
```
In [ ]: # hex  
sns.jointplot(data=tips,x='total_bill',y='tip',kind= 'hex' )
```

```
Out[ ]: <seaborn.axisgrid.JointGrid at 0x1ec43596ed0>
```



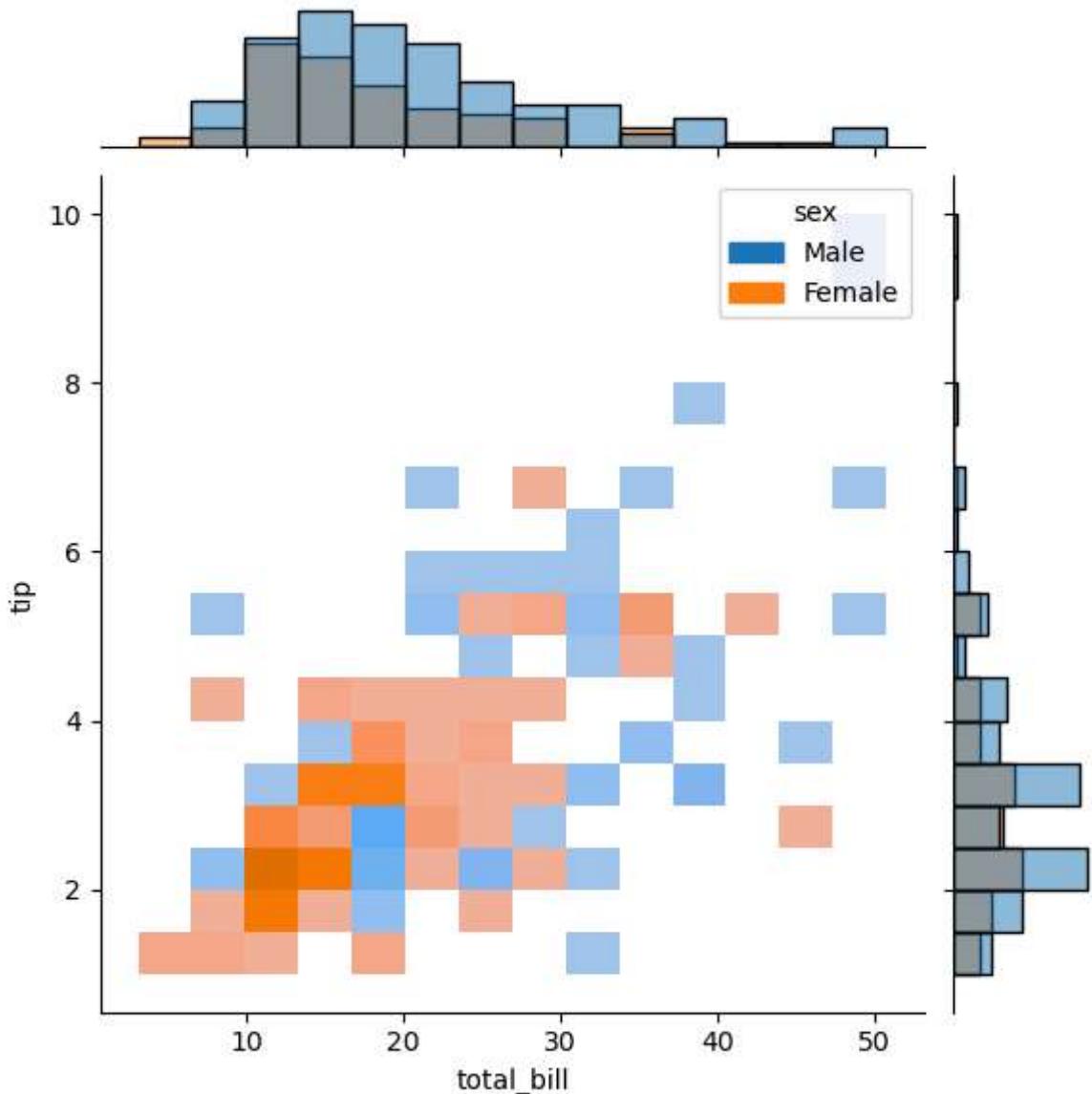
```
In [ ]: # kind -2d histogram + 1 histogram
sns.jointplot(data=tips,x='total_bill',y='tip',kind='hist' )
```

```
Out[ ]: <seaborn.axisgrid.JointGrid at 0x1ec46d0c810>
```



```
In [ ]: # hue
sns.jointplot(data=tips,x='total_bill',y='tip',kind='hist',hue='sex')
```

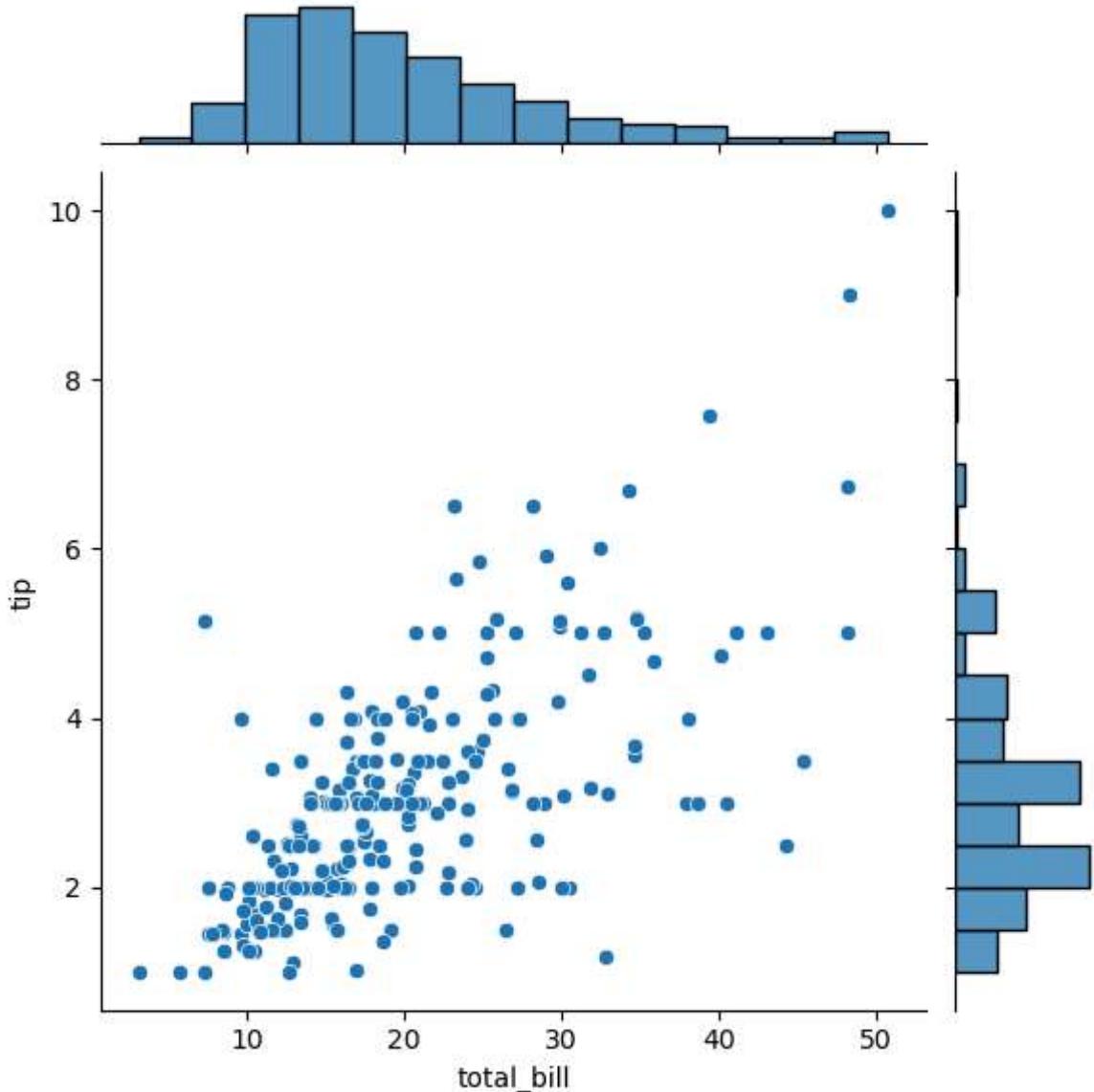
```
Out[ ]: <seaborn.axisgrid.JointGrid at 0x1ec46d3ee10>
```



Joint Grid

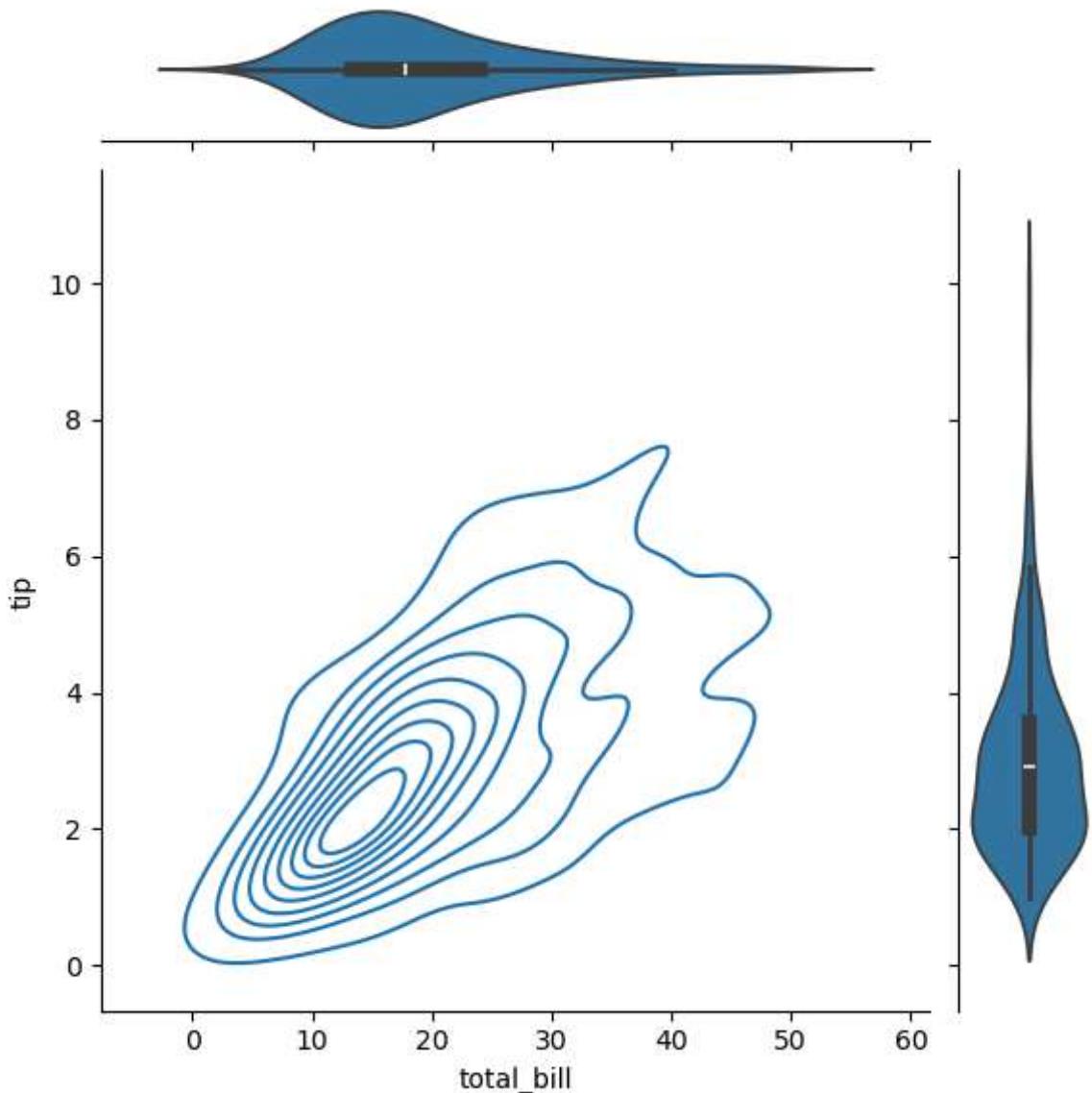
```
In [ ]: # we can change plots as per our requirements
g = sns.JointGrid(data=tips,x='total_bill',y='tip')
g.plot(sns.scatterplot,sns.histplot)
```

```
Out[ ]: <seaborn.axisgrid.JointGrid at 0x1ec473a8850>
```



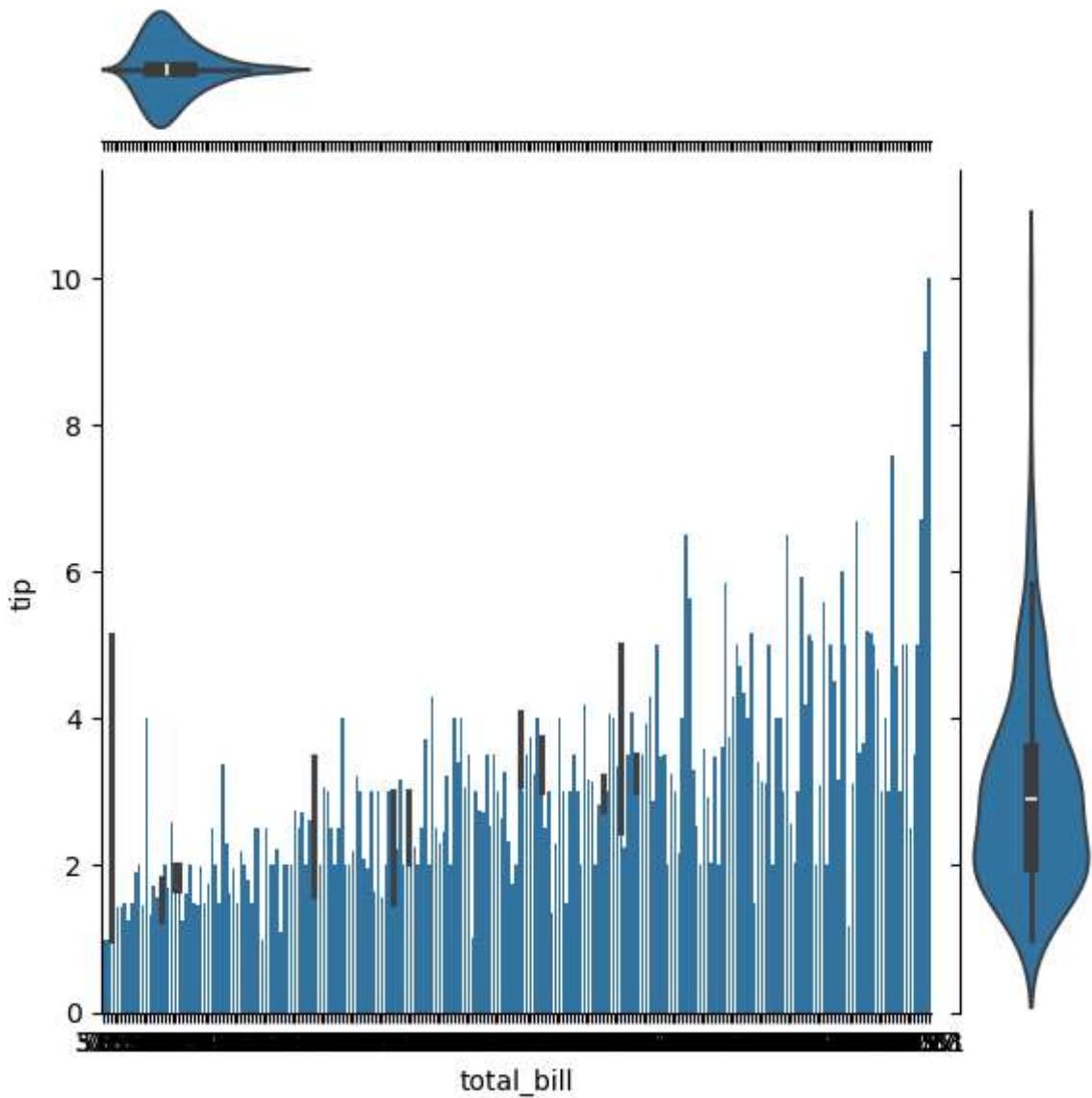
```
In [ ]: g = sns.JointGrid(data=tips,x='total_bill',y='tip')
g.plot(sns.kdeplot,sns.violinplot)
```

```
Out[ ]: <seaborn.axisgrid.JointGrid at 0x1ec473a9850>
```



```
In [ ]: g = sns.JointGrid(data=tips,x='total_bill',y='tip')
g.plot(sns.barplot,sns.violinplot)
```

```
Out[ ]: <seaborn.axisgrid.JointGrid at 0x1ec47c59ad0>
```



In []: