https://www.youtube.com/watch?v=kLWeKeqc9Ms&list=PLKnIA16_RmvbAlyx4_rdtR66B7EHX5k3z&index=8 (https://www.youtube.com/watch?v=kLWeKeqc9Ms&list=PLKnIA16_RmvbAlyx4_rdtR66B7EHX5k3z&index=8

In [2]:
```
import seaborn as sns
tips=sns.load_dataset('tips')
iris=sns.load_dataset('iris')
```

Categorical data

In [3]:
```
# Categorical Plots
#     Categorical Scatter Plot
#     Stripplot
#     Swarmplot
# Categorical Distribution Plots
#     Boxplot
#     Violinplot
# Categorical Estimate Plot -> for central tendency
#     Barplot
#     Pointplot
#     Countplot

# Figure level function -> catplot
```

strip plot

In [4]:
```
tips
```

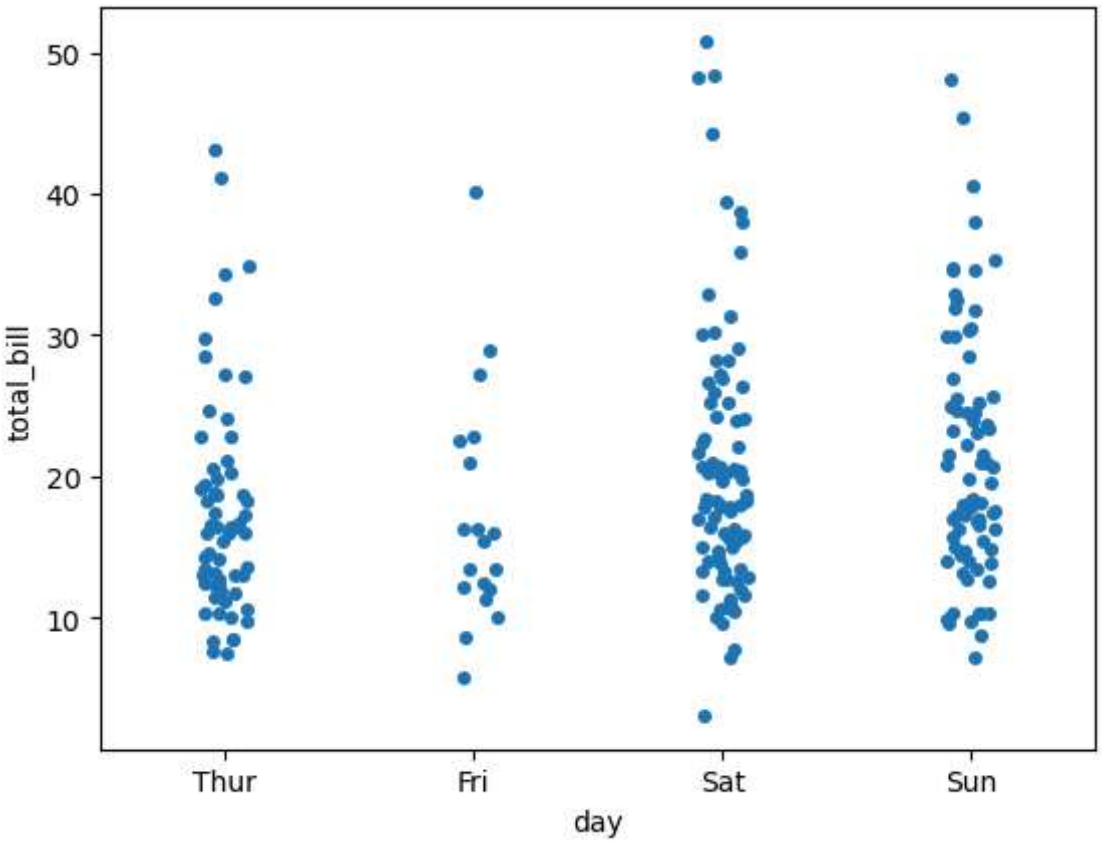
Out[4]:

	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
...	...	...	...	...	...	...	...
239	29.03	5.92	Male	No	Sat	Dinner	3
240	27.18	2.00	Female	Yes	Sat	Dinner	2
241	22.67	2.00	Male	Yes	Sat	Dinner	2
242	17.82	1.75	Male	No	Sat	Dinner	2
243	18.78	3.00	Female	No	Thur	Dinner	2

244 rows × 7 columns

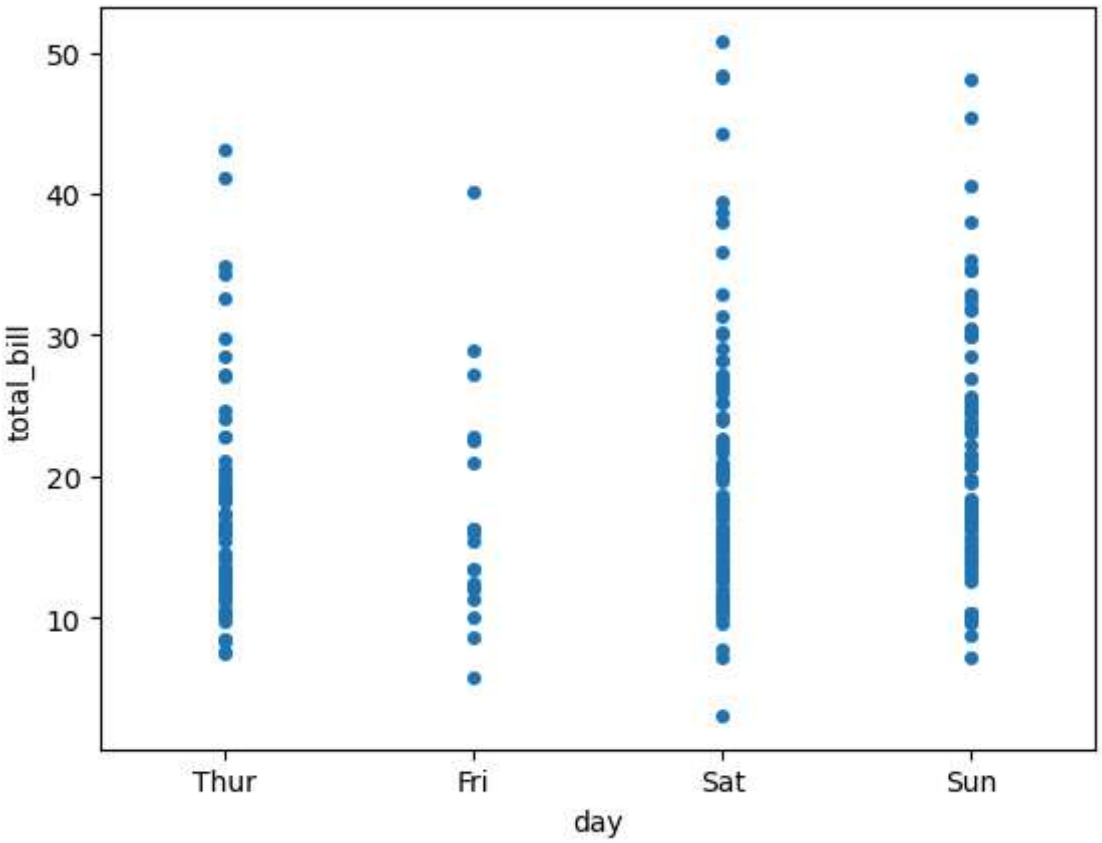
In [5]:
```
sns.stripplot(data=tips,x='day',y='total_bill')       #axis level function
```

Out[5]:
```
<Axes: xlabel='day', ylabel='total_bill'>
```



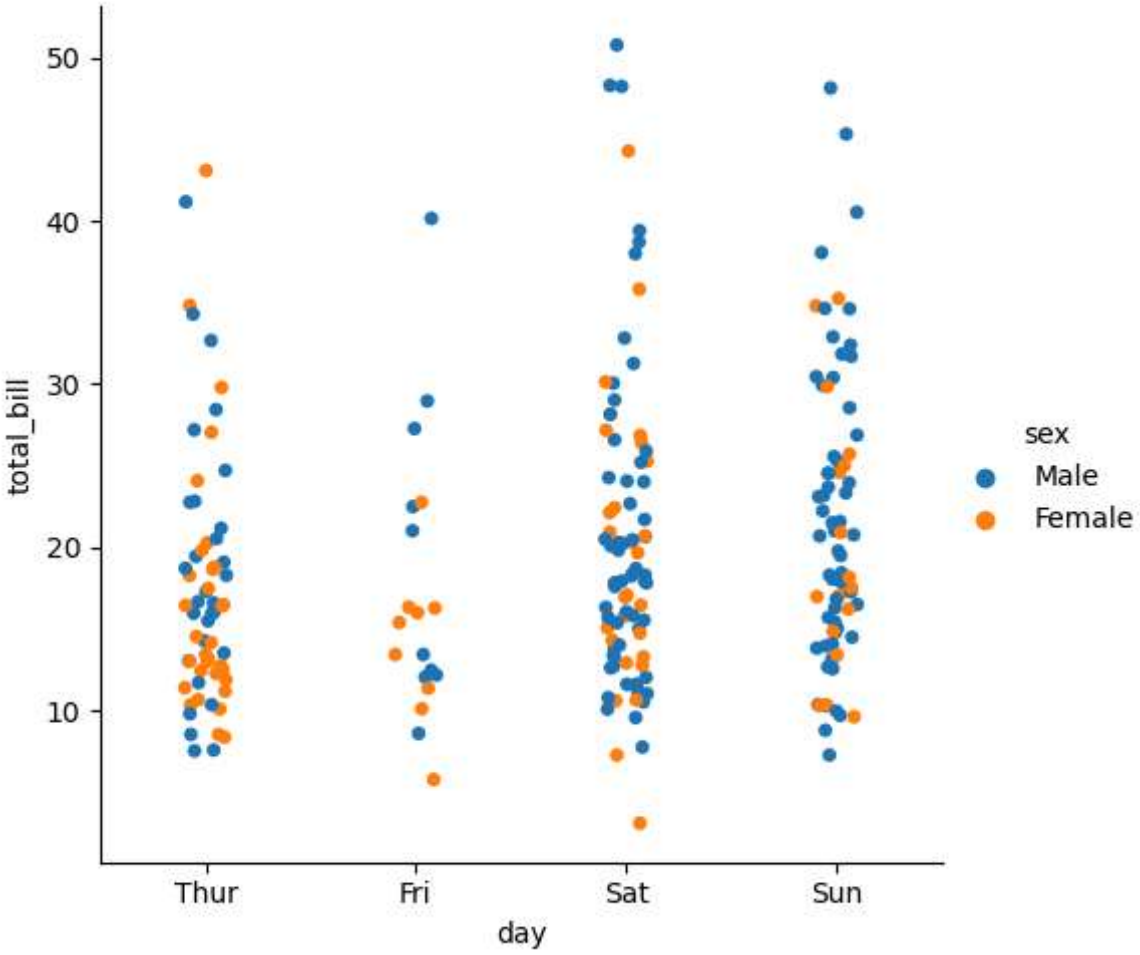
In [6]:
```
sns.stripplot(sns.stripplot(data=tips,x='day',y='total_bill'),jitter=False)
```

Out[6]:
```
<Axes: xlabel='day', ylabel='total_bill'>
```



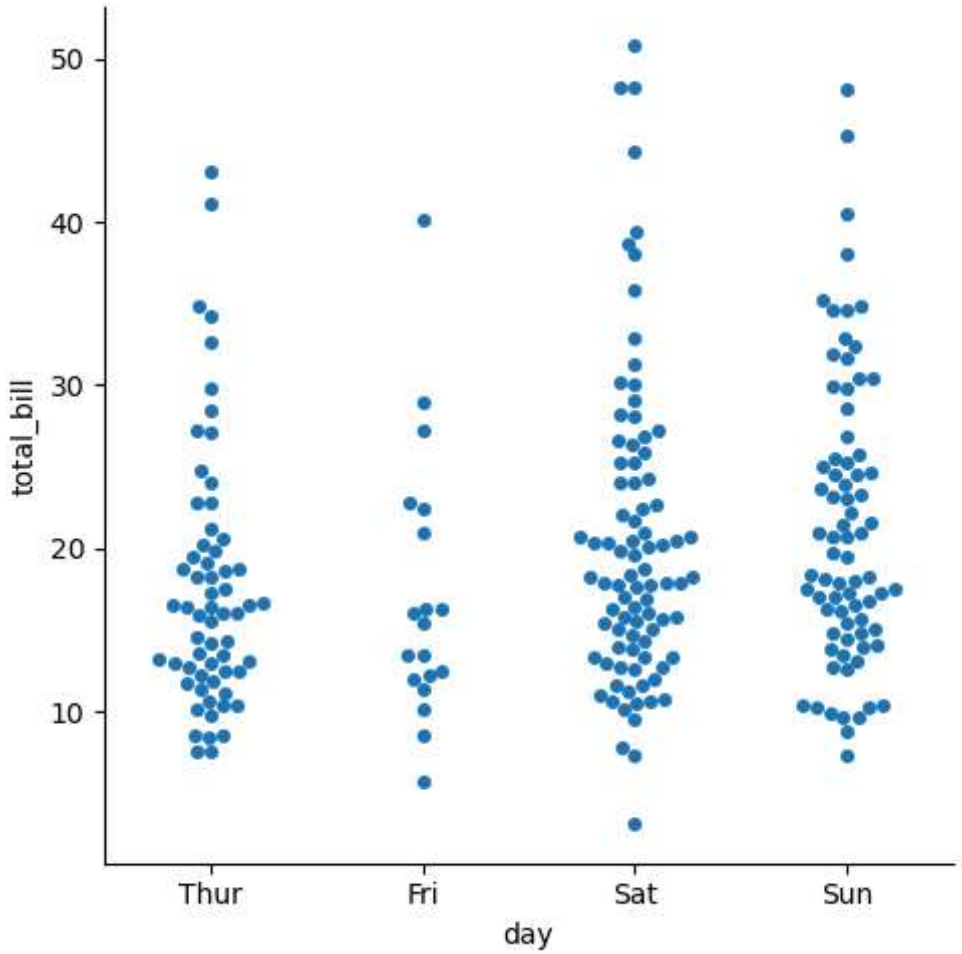
In [11]:
```
sns.catplot(kind='strip',data=tips,x='day',y='total_bill',jitter=0.1,hue='sex')
```

Out[11]:
```
<seaborn.axisgrid.FacetGrid at 0x23ce4f57c10>
```



In [13]:
```
sns.catplot(kind='swarm',data=tips,x='day',y='total_bill')       #swarm plot gives better distribution but works better
```

Out[13]:
```
<seaborn.axisgrid.FacetGrid at 0x23ce50221d0>
```



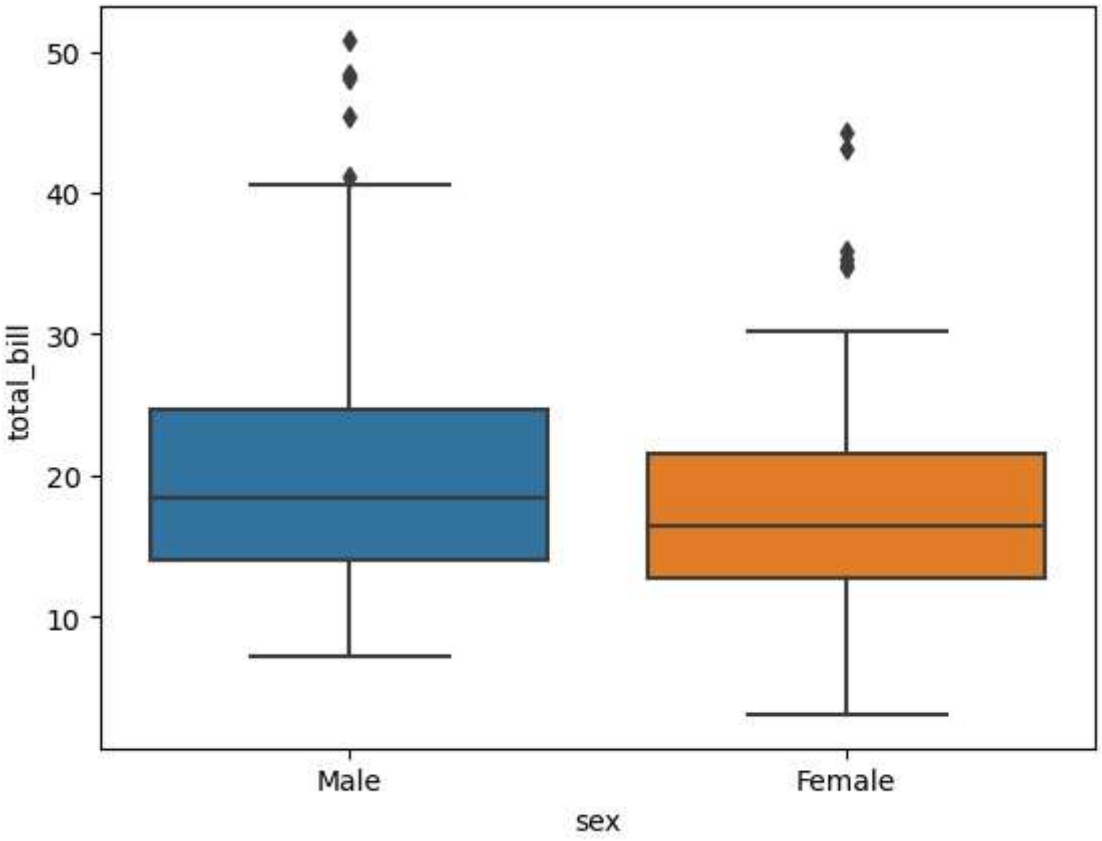
categorical distribution plots

box plot

A boxplot is a standardized way of displaying the distribution of data based on a five number summary (“minimum”, first quartile [Q1], median, third quartile [Q3] and “maximum”). It can tell you about your outliers and what their values are. Boxplots can also tell you if your data is symmetrical, how tightly your data is grouped and if and how your data is skewed.

In [14]:
```
sns.boxplot(data=tips,x='sex',y='total_bill')
```

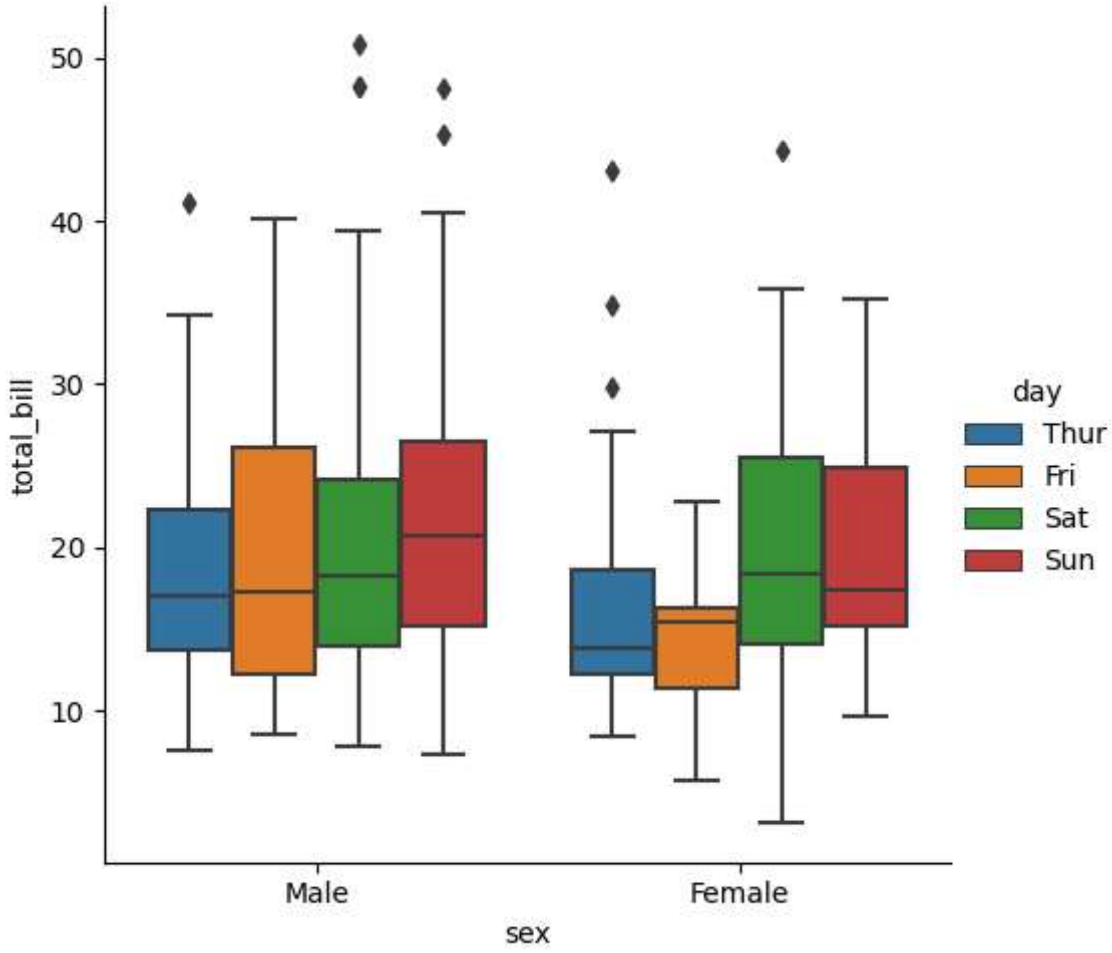
Out[14]:
```
<Axes: xlabel='sex', ylabel='total_bill'>
```



In [15]:
```
#figure level
```

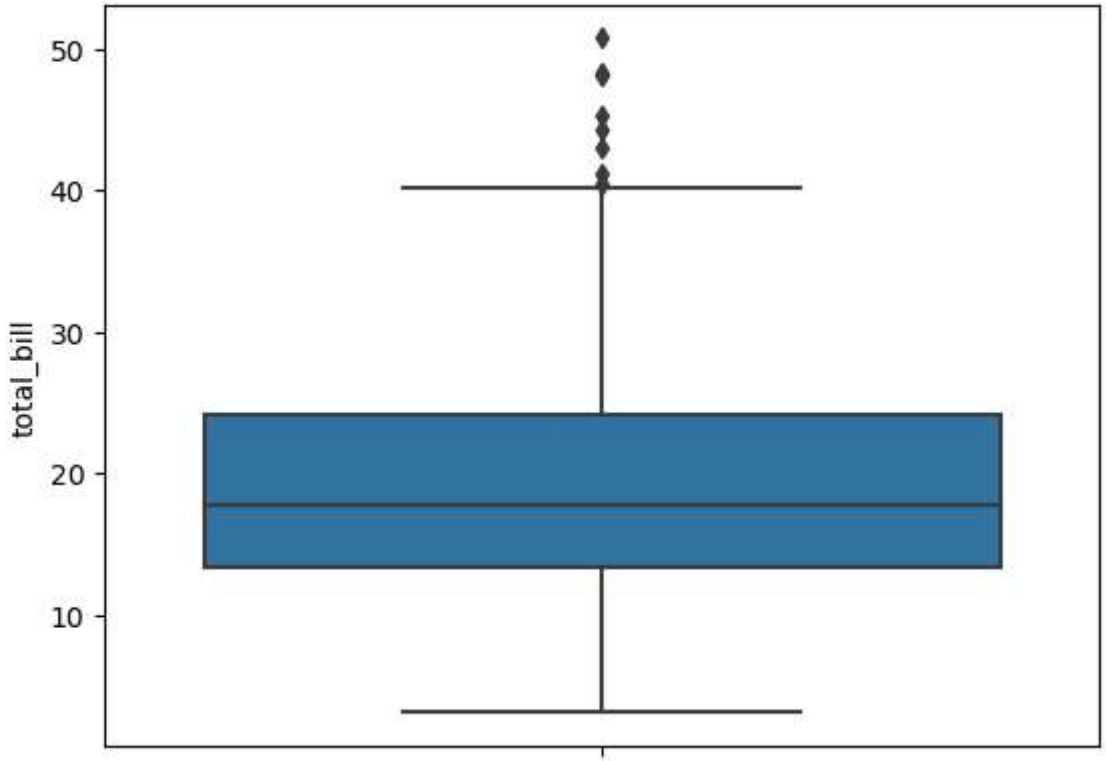
In [18]:
```
sns.catplot(kind='box',data=tips,x='sex',y='total_bill',hue='day')
```

Out[18]:
```
<seaborn.axisgrid.FacetGrid at 0x23ce63175d0>
```



In [19]:
```
#single box plot
sns.boxplot(data=tips,y='total_bill')
```

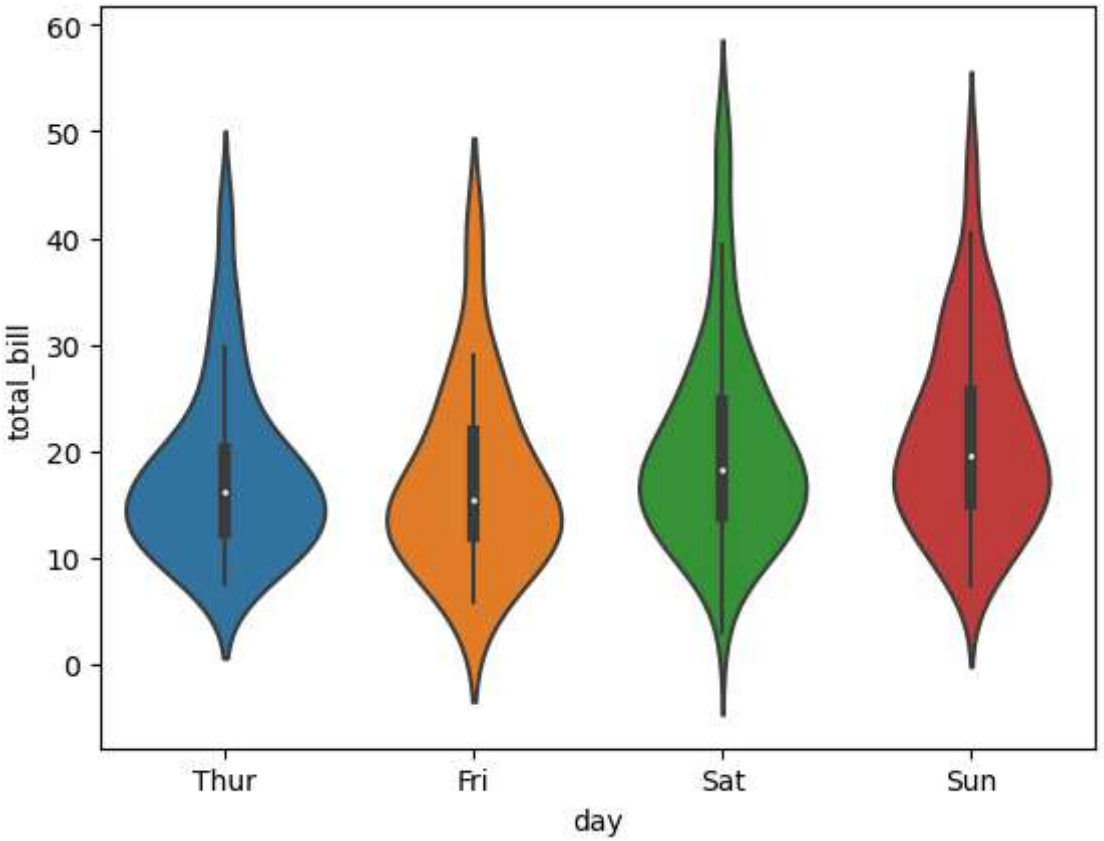
Out[19]:
```
<Axes: ylabel='total_bill'>
```



violin plot : box+kde

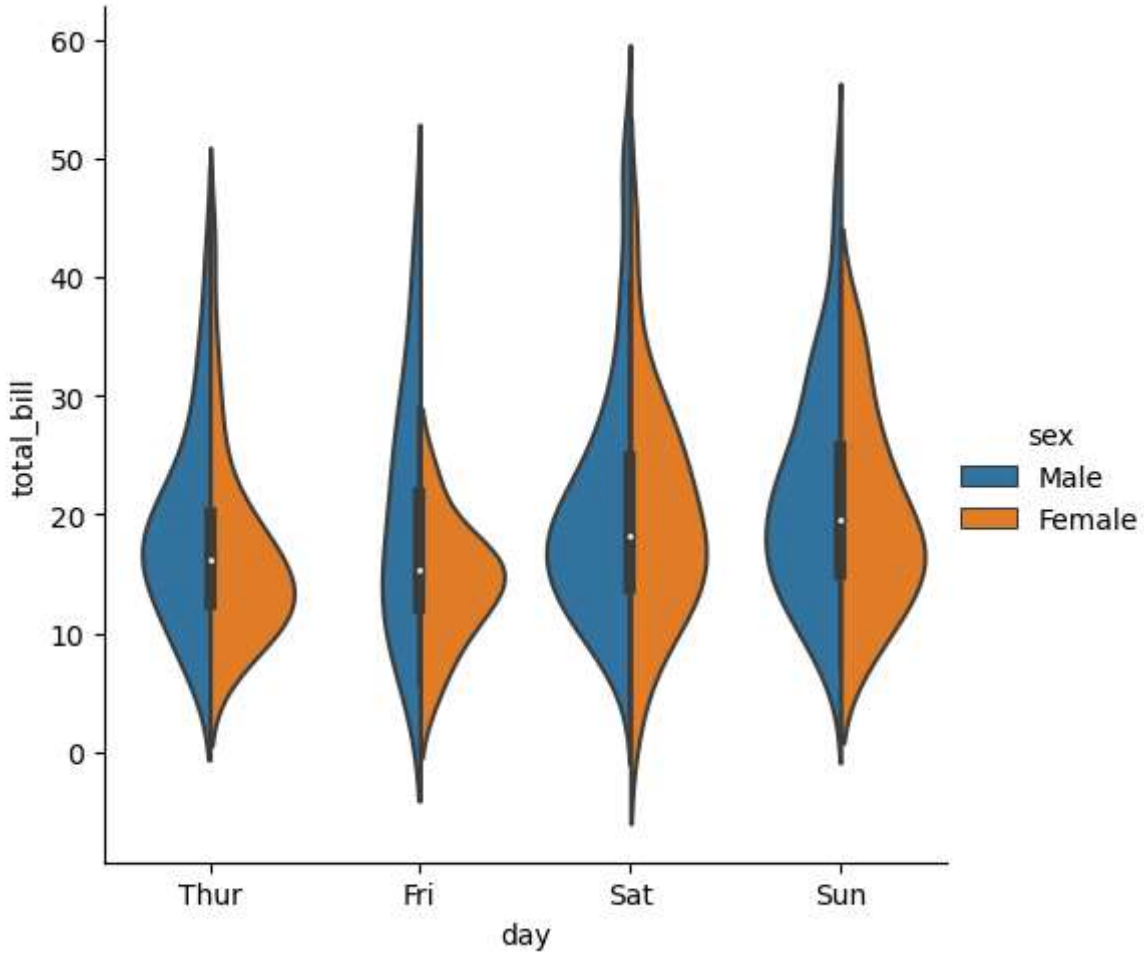
In [20]:
```
sns.violinplot(data=tips,x='day',y='total_bill')
```

Out[20]:
```
<Axes: xlabel='day', ylabel='total_bill'>
```



In [22]:
```
sns.catplot(kind='violin',data=tips,x='day',y='total_bill',hue='sex',split=True)
```

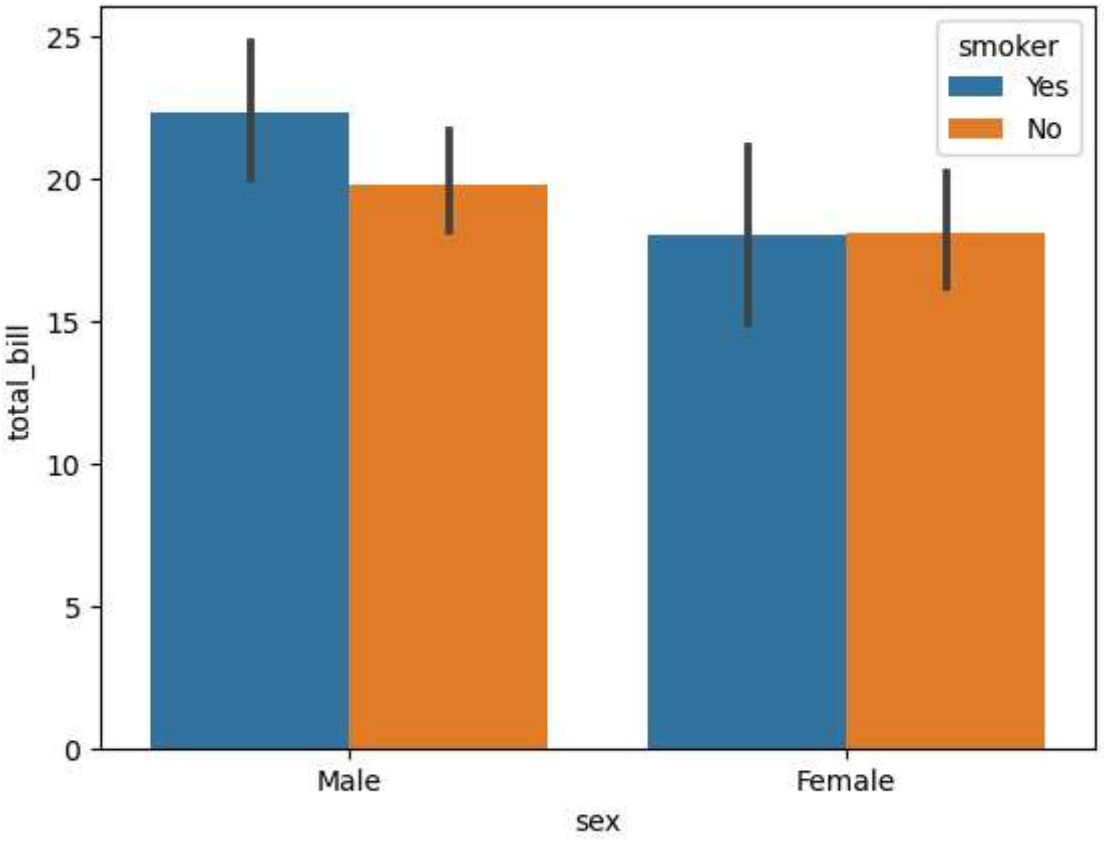
Out[22]:
```
<seaborn.axisgrid.FacetGrid at 0x23ce6f62390>
```



Bar plot

In [25]:
```
sns.barplot(data=tips,x='sex',y='total_bill',hue='smoker')   #the wicks are the error bar
```

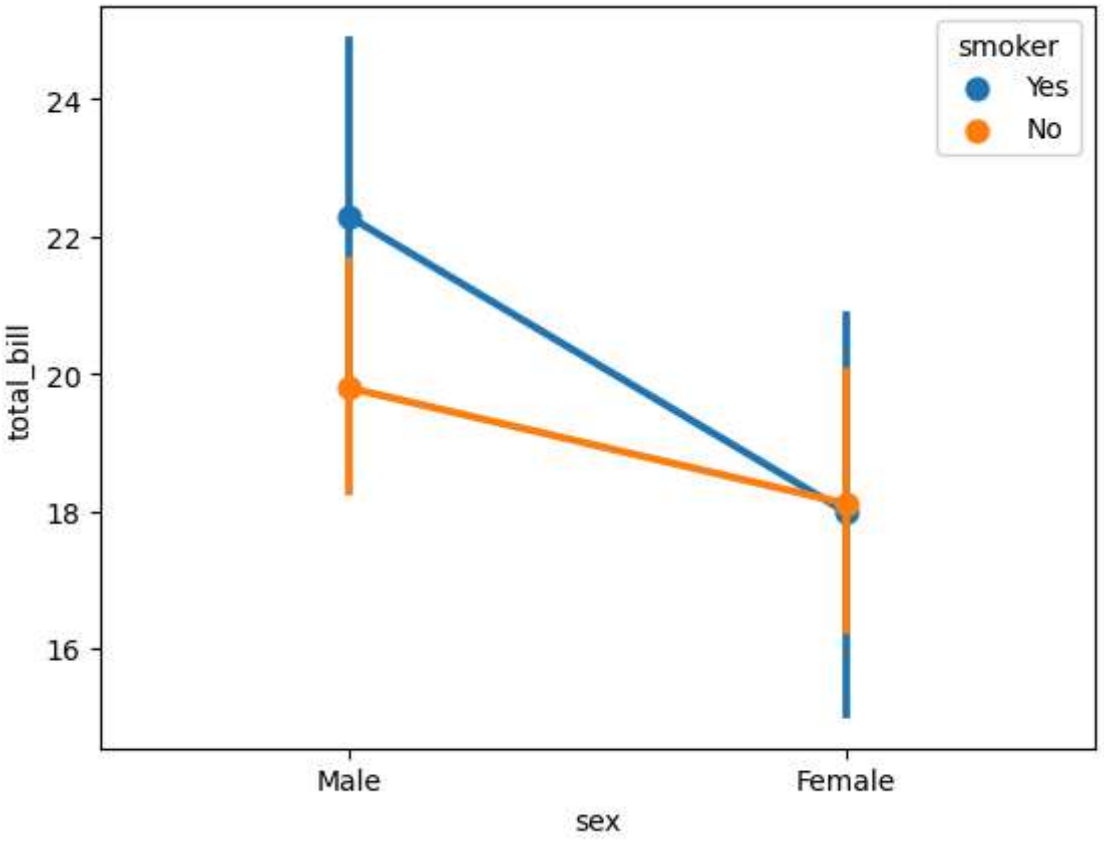
Out[25]:
```
<Axes: xlabel='sex', ylabel='total_bill'>
```



point plot

In [26]:
```
sns.pointplot(data=tips,x='sex',y='total_bill',hue='smoker')
```

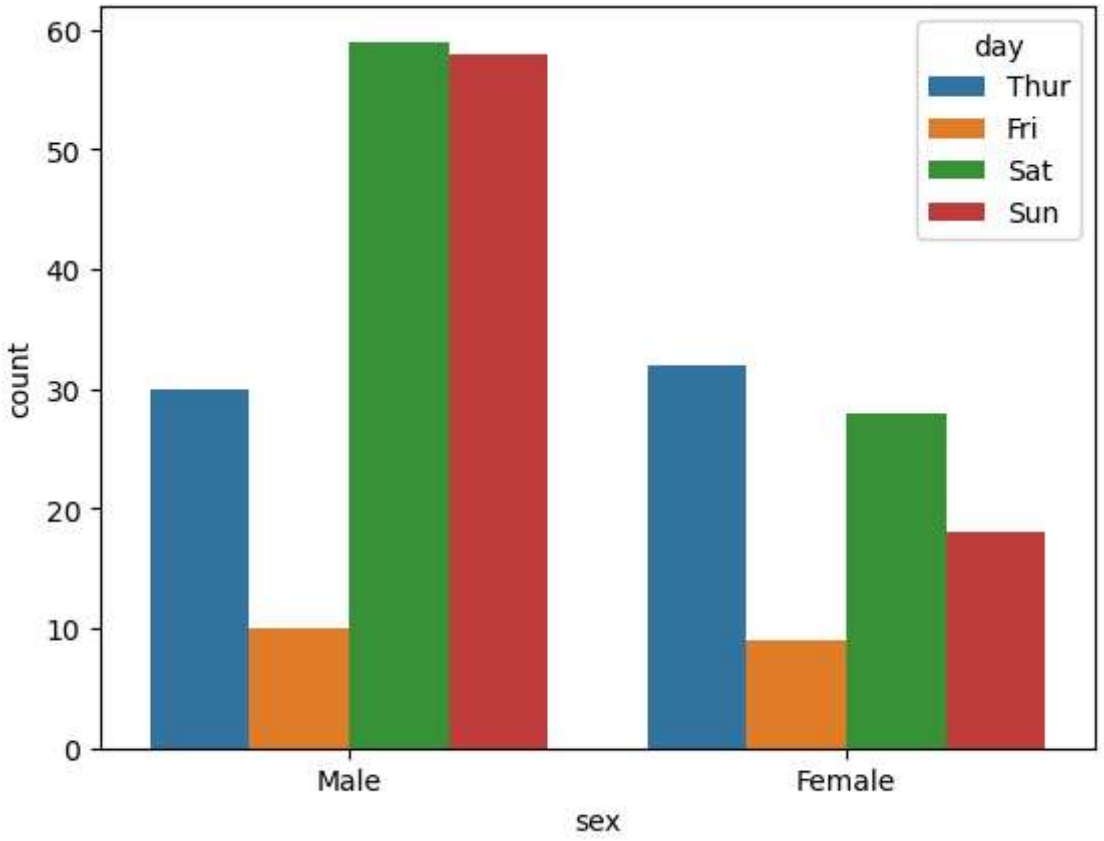
Out[26]:
```
<Axes: xlabel='sex', ylabel='total_bill'>
```



count plot

In [27]:
```
sns.countplot(data=tips,x='sex',hue='day')
```

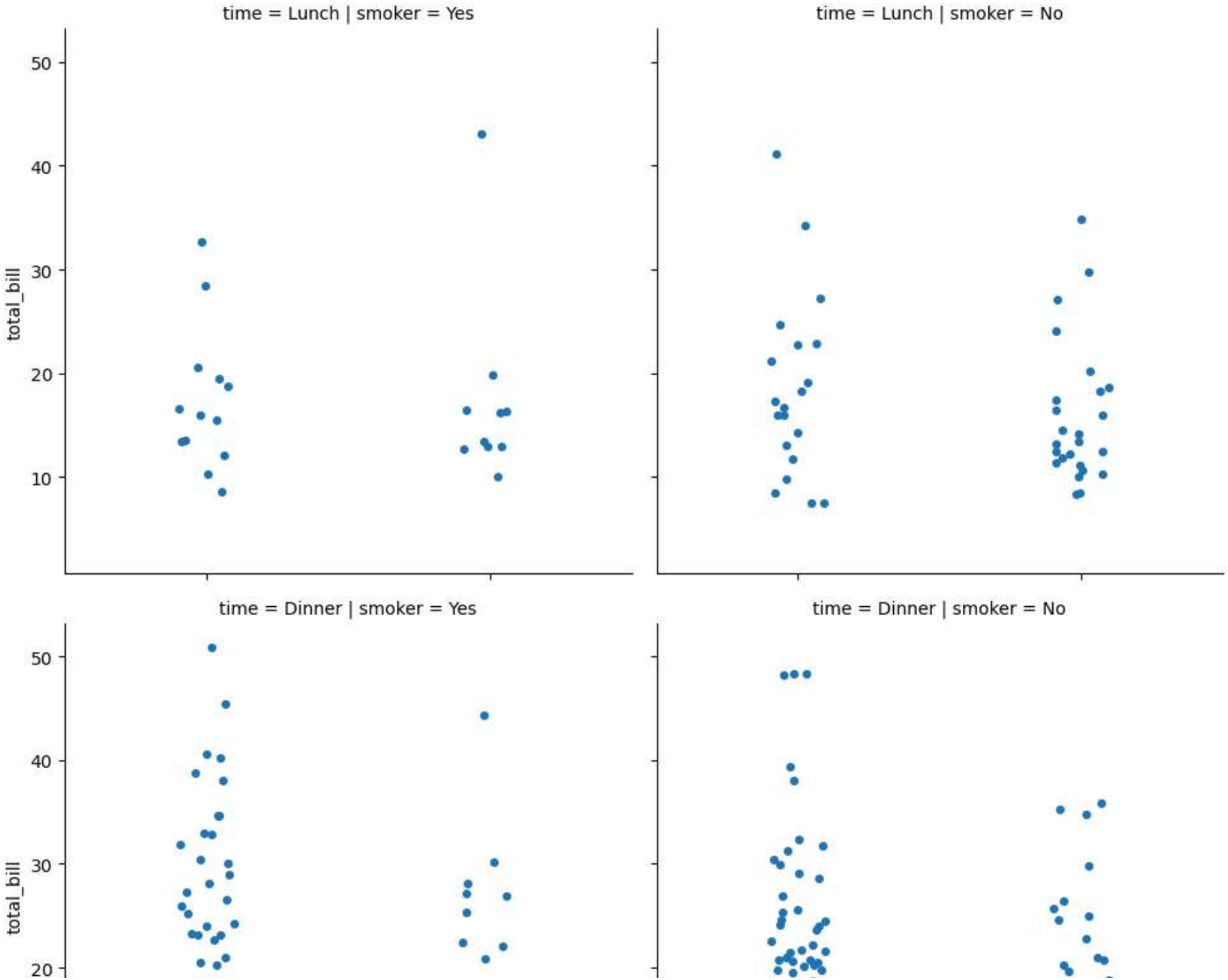
Out[27]:
```
<Axes: xlabel='sex', ylabel='count'>
```

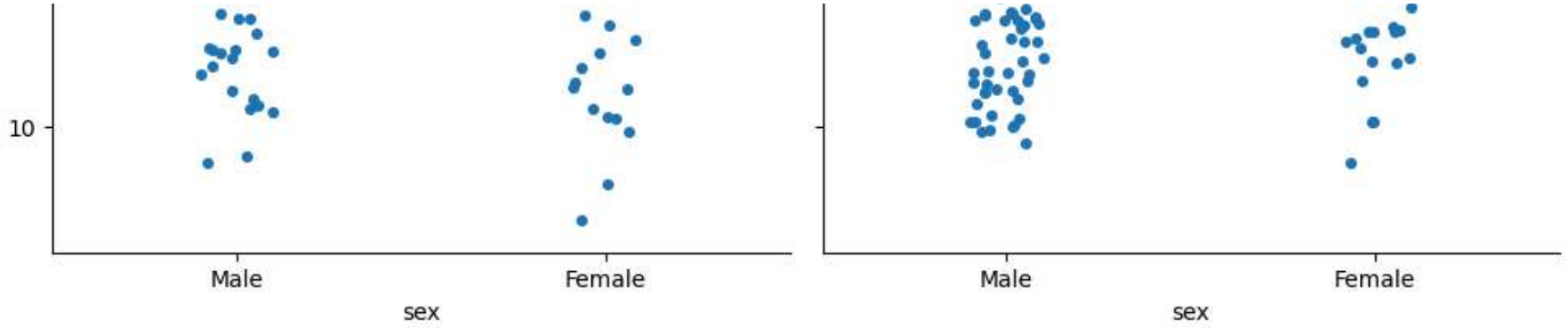


facet using catplot

In [29]:
```
sns.catplot(data=tips,x='sex',y='total_bill',col='smoker',row='time')
```

Out[29]:
```
<seaborn.axisgrid.FacetGrid at 0x23ce7346dd0>
```


time = Lunch | smoker = Yes
time = Lunch | smoker = No
time = Dinner | smoker = Yes
time = Dinner | smoker = No
total_bill
50
40
30
20
10



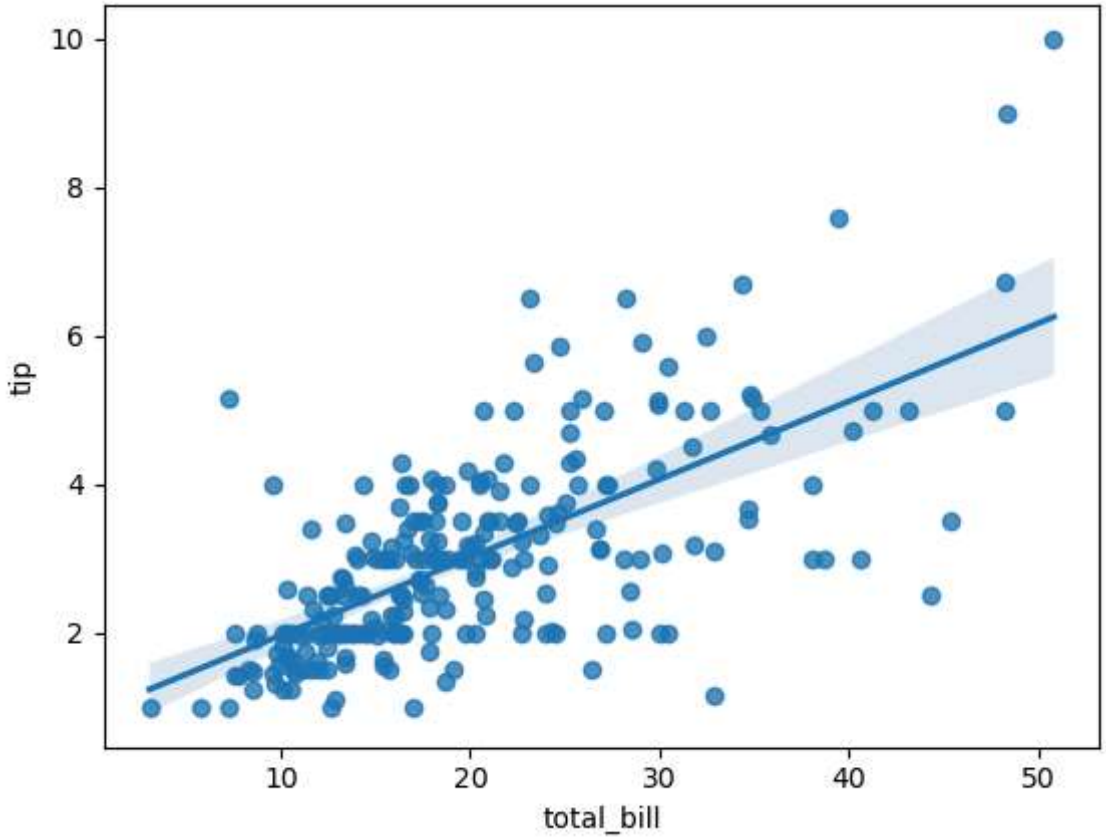
Reg plot or regression plot

1.regplot

2.lmplot

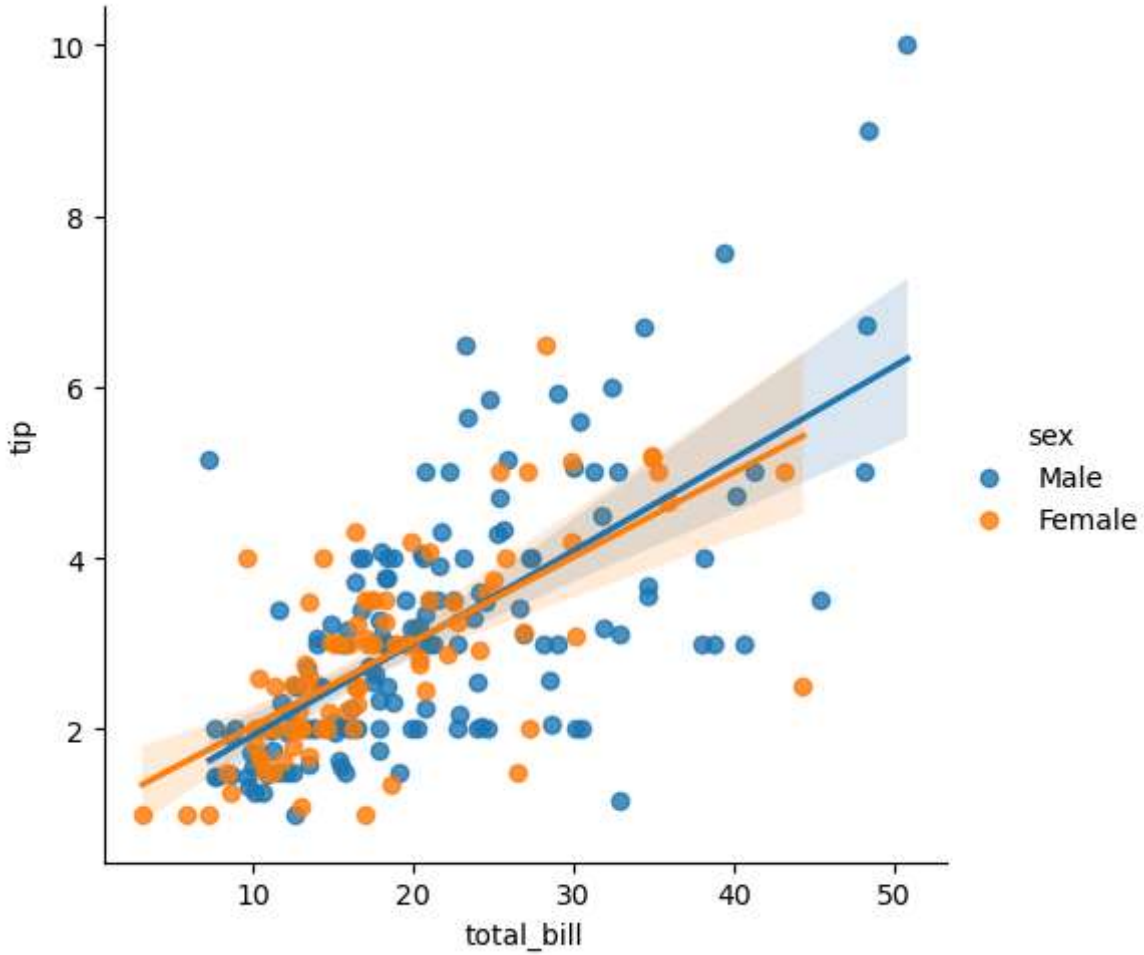
In [31]:
```
sns.regplot(data=tips,x='total_bill',y='tip')      #95% confidence level
```

Out[31]:
```
<Axes: xlabel='total_bill', ylabel='tip'>
```



In [33]:
```
sns.lmplot(data=tips,x='total_bill',y='tip',hue='sex')     #95% confidence level
```

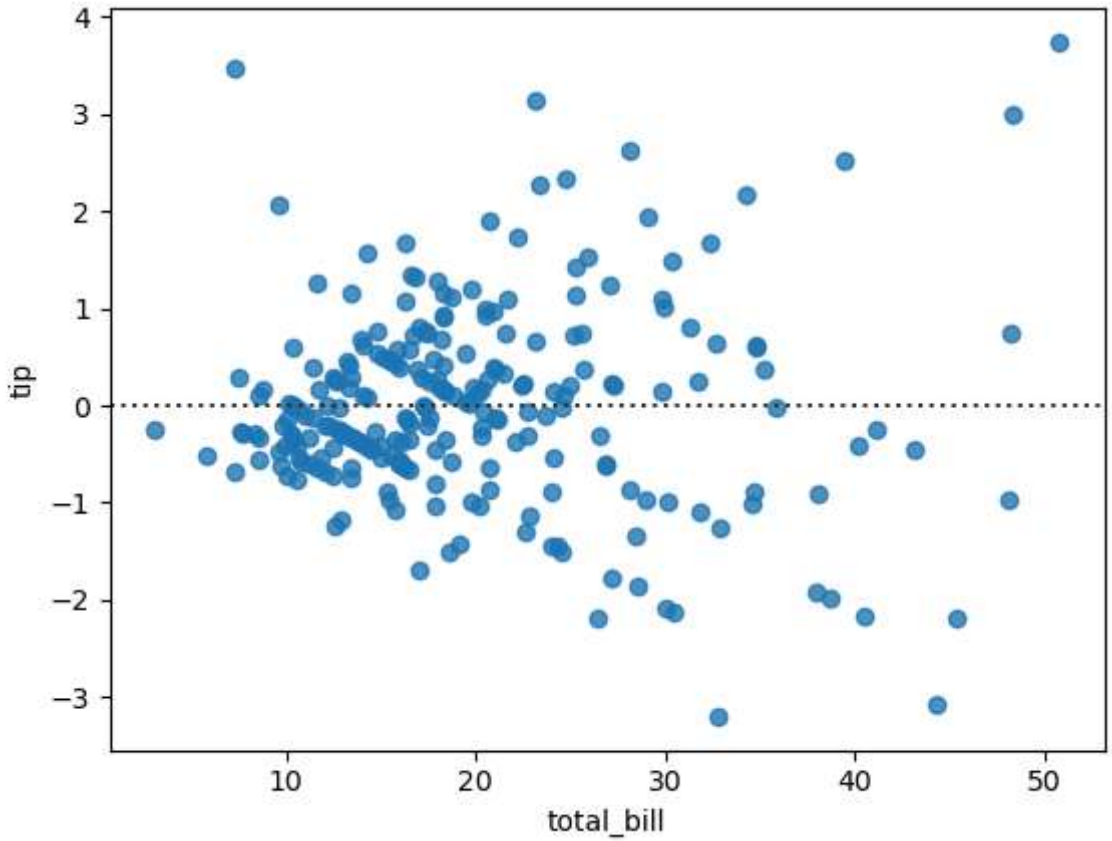
Out[33]: <seaborn.axisgrid.FacetGrid at 0x23ce8ee4590>



In [34]:
```
#resid plot
```

In [35]:
```
sns.residplot(data=tips,x='total_bill',y='tip')
```

Out[35]:
```
<Axes: xlabel='total_bill', ylabel='tip'>
```



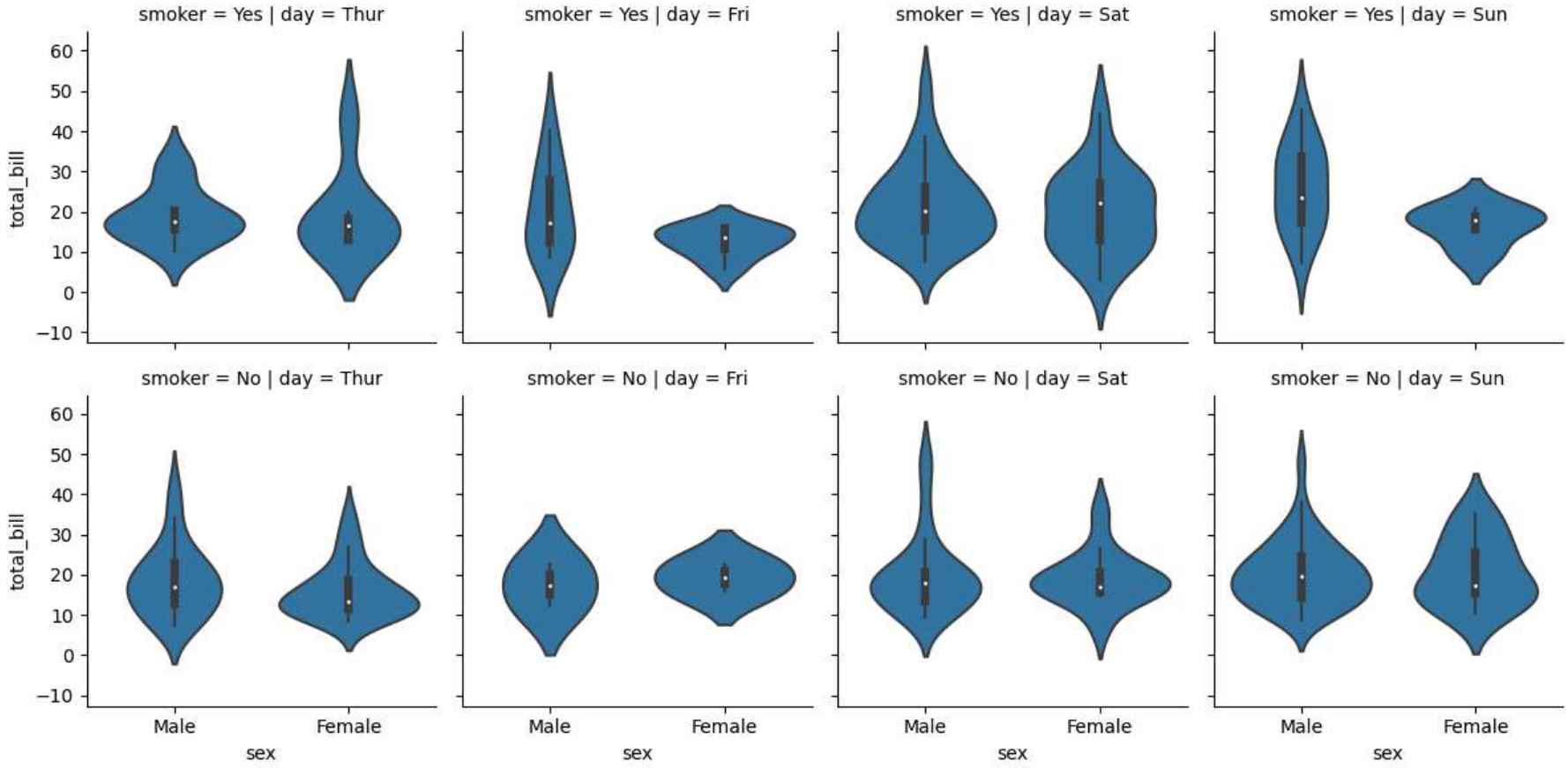
Multi grid plot

Facet grid

In [38]:
```
g=sns.FacetGrid(data=tips,col='day',row='smoker')
g.map(sns.violinplot,'sex','total_bill')
g.add_legend()
```

```
C:\Users\mrchi\anaconda3\Lib\site-packages\seaborn\axisgrid.py:712: UserWarning: Using the violinplot function withou
t specifying `order` is likely to produce an incorrect plot.
  warnings.warn(warning)
```

Out[38]:
```
<seaborn.axisgrid.FacetGrid at 0x23ceb472790>
```



pairplot (imp)

In [39]:
```
iris
```

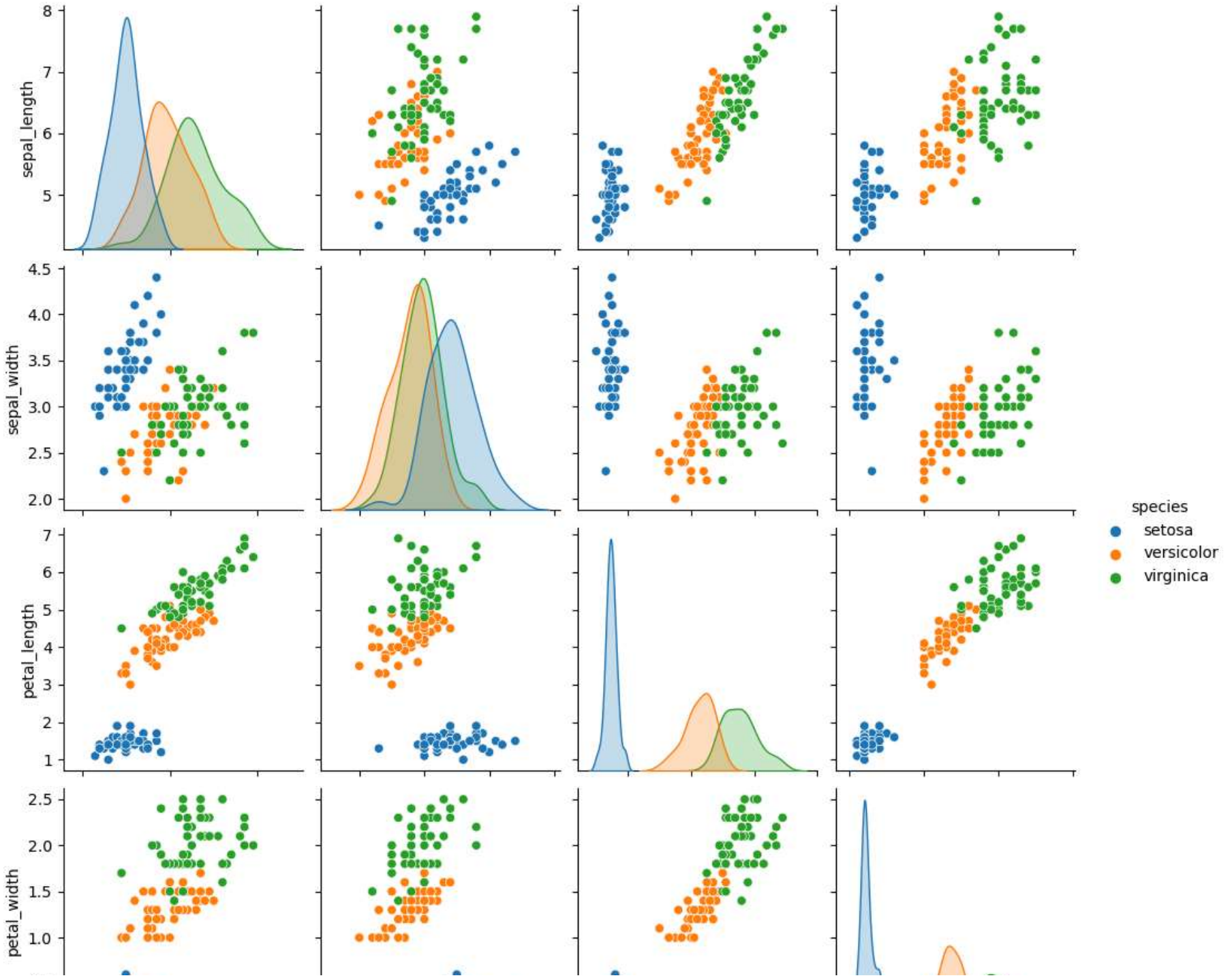
Out[39]:

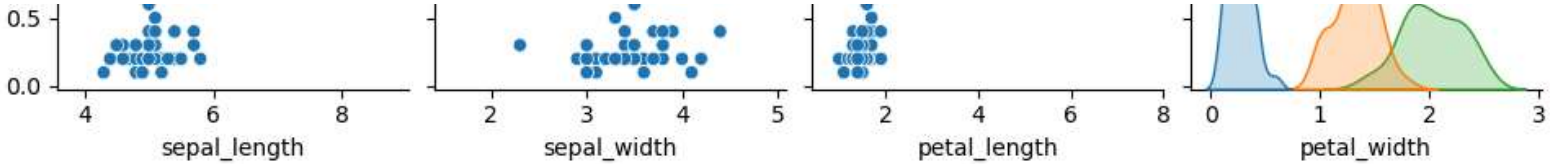
	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	virginica
146	6.3	2.5	5.0	1.9	virginica
147	6.5	3.0	5.2	2.0	virginica
148	6.2	3.4	5.4	2.3	virginica
149	5.9	3.0	5.1	1.8	virginica

150 rows × 5 columns

In [41]:
```
sns.pairplot(iris,hue='species')
```

Out[41]:
```
<seaborn.axisgrid.PairGrid at 0x23ceb25df50>
```

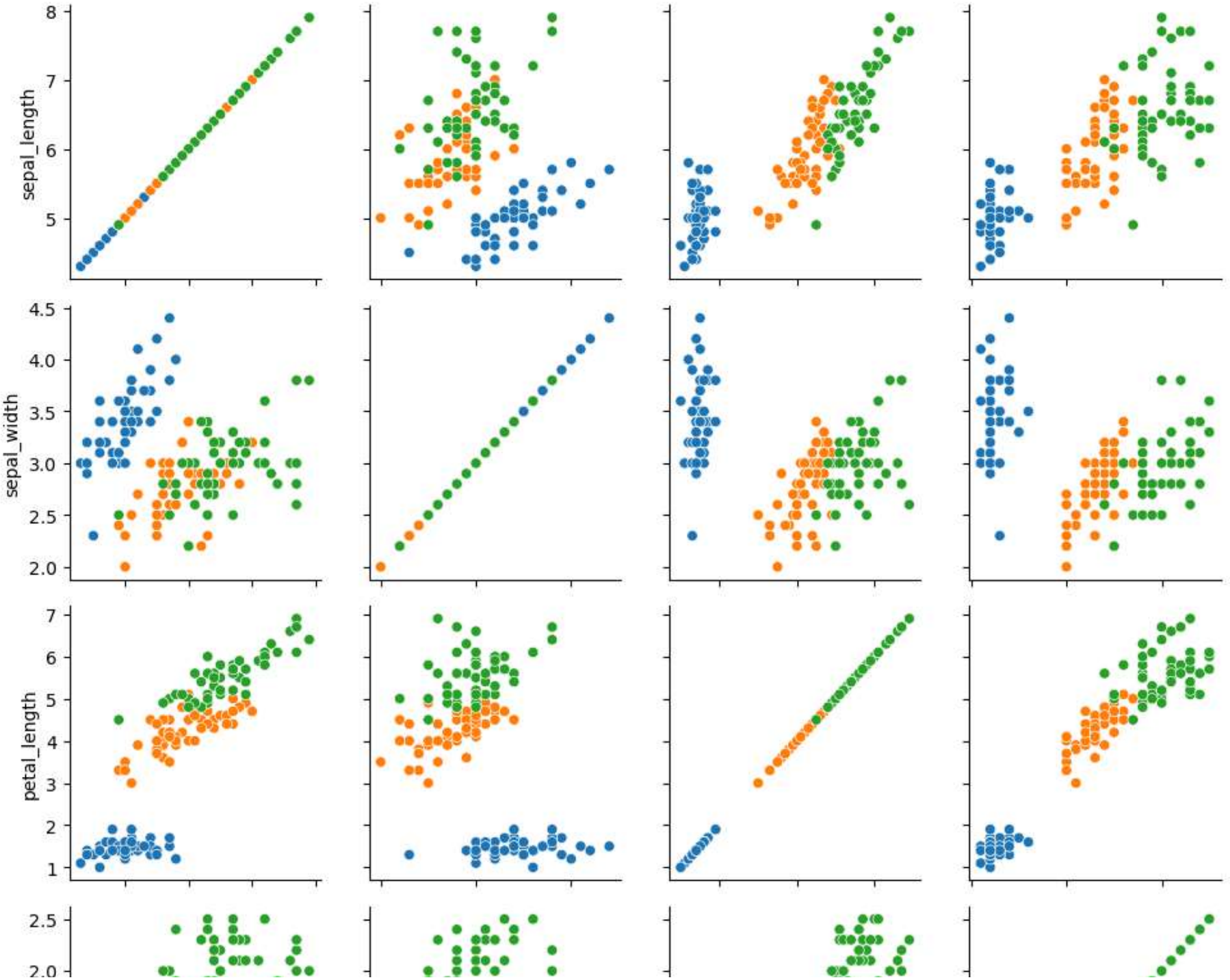

sepal_length
sepal_width
petal_length
petal_width
species
setosa
versicolor
virginica

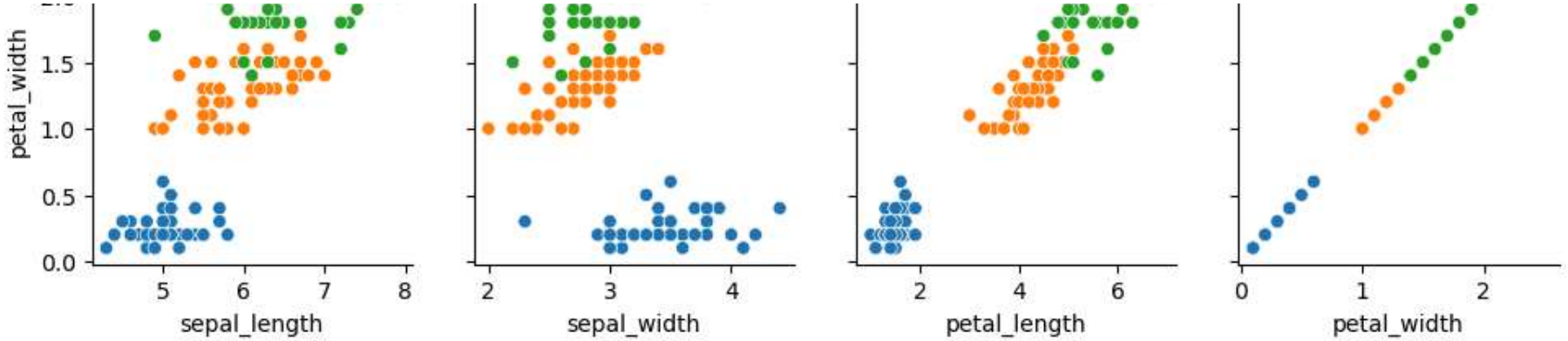

0.5
0.0
4
6
8
sepal_length
2
3
4
5
sepal_width
2
4
6
8
petal_length
0
1
2
3
petal_width

PairGrid (pairplot ka baap)

In [47]:
```
g1=sns.PairGrid(data=iris,hue='species')
g1.map(sns.scatterplot)
```

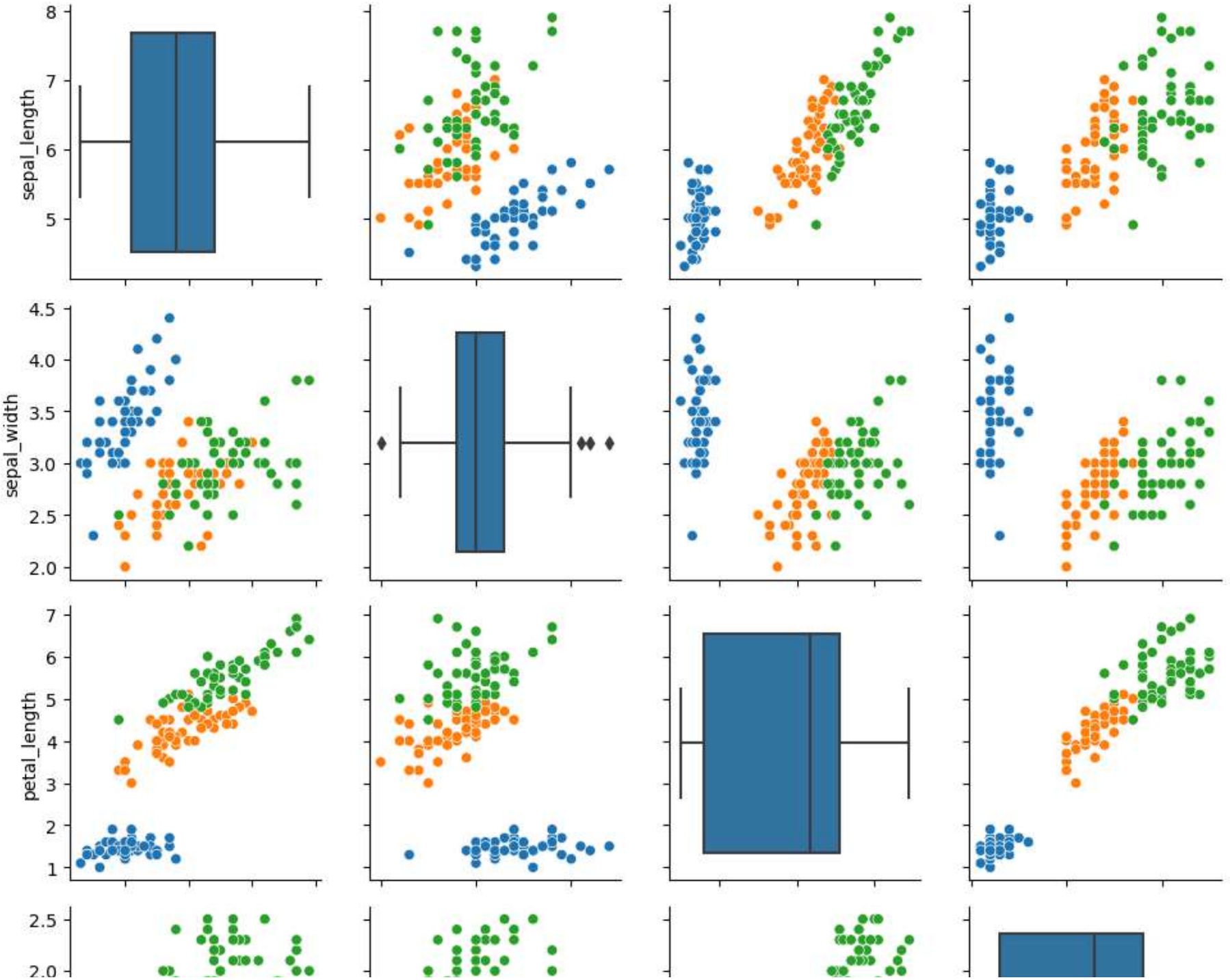
Out[47]:
```
<seaborn.axisgrid.PairGrid at 0x23cf1fedf50>
```

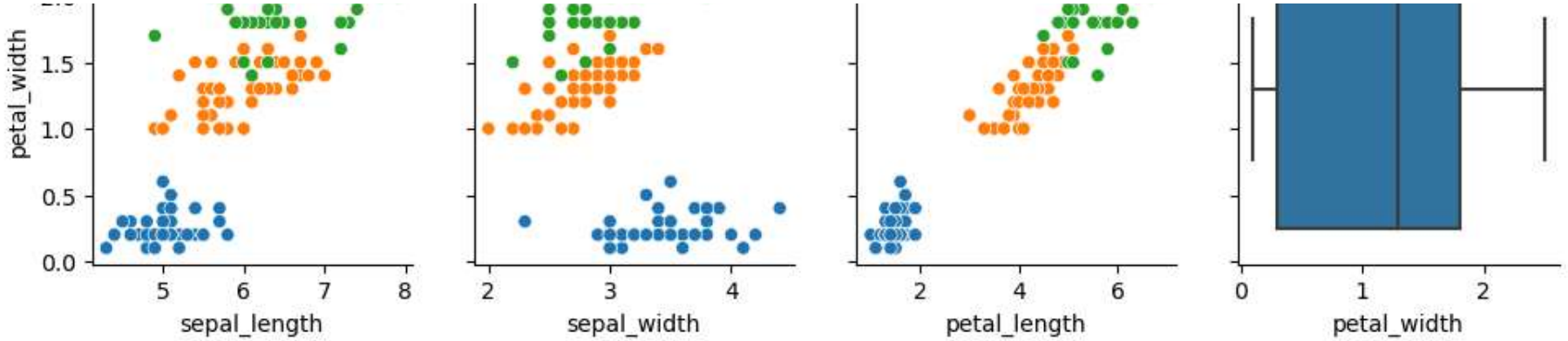

sepal_length
sepal_width
petal_length
8
7
6
5
4.5
4.0
3.5
3.0
2.5
2.0
7
6
5
4
3
2
1
2.5
2.0


petal_width
1.5
1.0
0.5
0.0
5
6
7
8
sepal_length
2
3
4
sepal_width
2
4
6
petal_length
0
1
2
petal_width

In [54]:
```
g2=sns.PairGrid(data=iris,hue='species')
g2.map_diag(sns.boxplot)
g2.map_offdiag(sns.scatterplot)
```

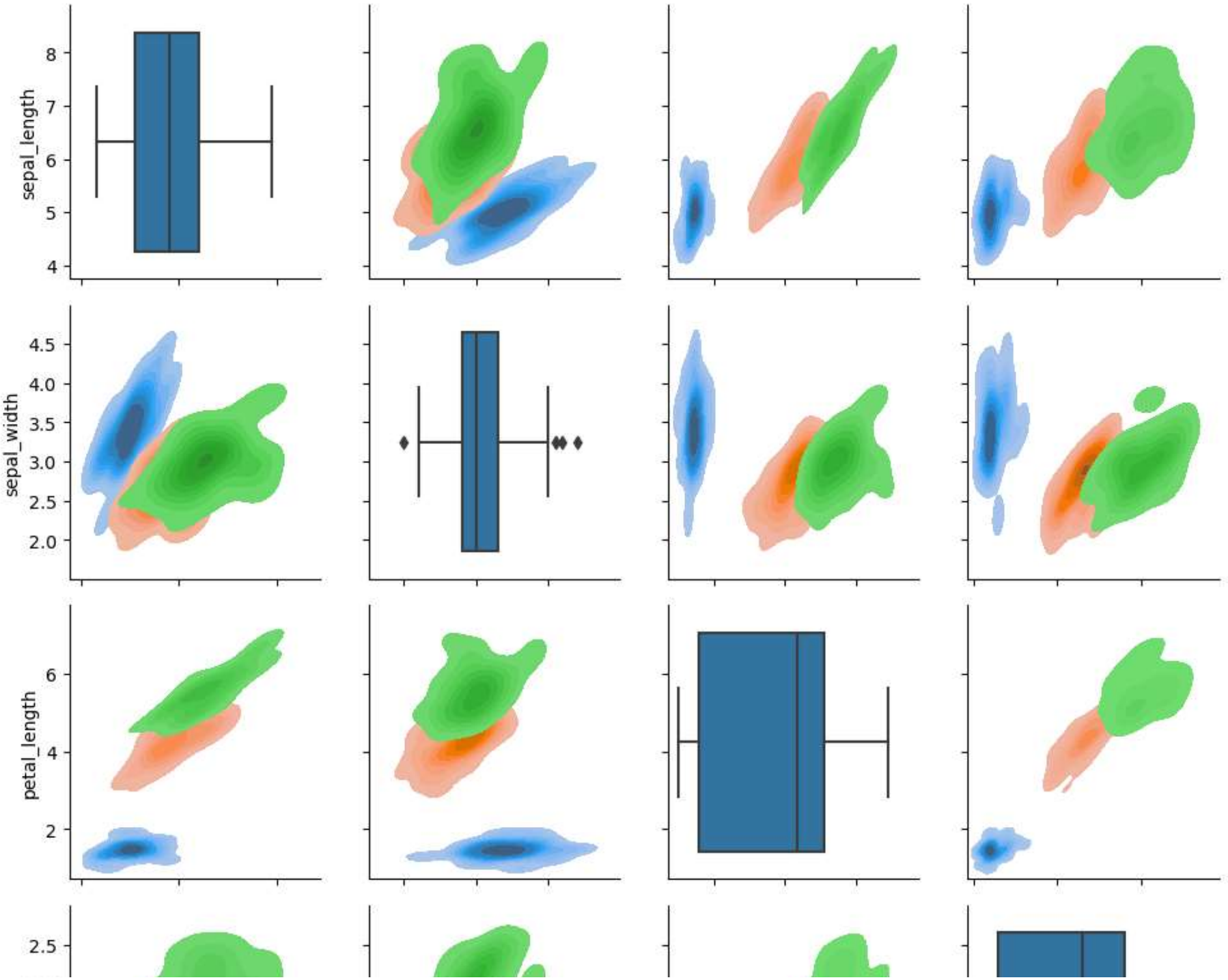
Out[54]:
```
<seaborn.axisgrid.PairGrid at 0x23cf8890590>
```

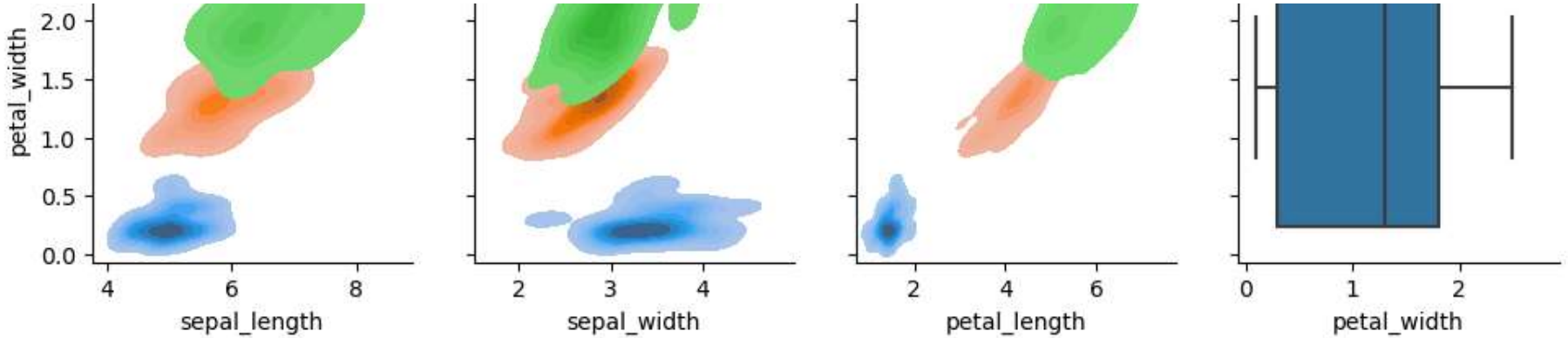

sepal_length
sepal_width
petal_length
8
7
6
5
4.5
4.0
3.5
3.0
2.5
2.0
7
6
5
4
3
2
1
2.5
2.0


petal_width
1.5
1.0
0.5
0.0
5
6
7
8
sepal_length
2
3
4
sepal_width
2
4
6
petal_length
0
1
2
petal_width

In [56]:
```
g3=sns.PairGrid(data=iris,hue='species')
g3.map_diag(sns.boxplot)
g3.map_offdiag(sns.kdeplot,fill=True)
```

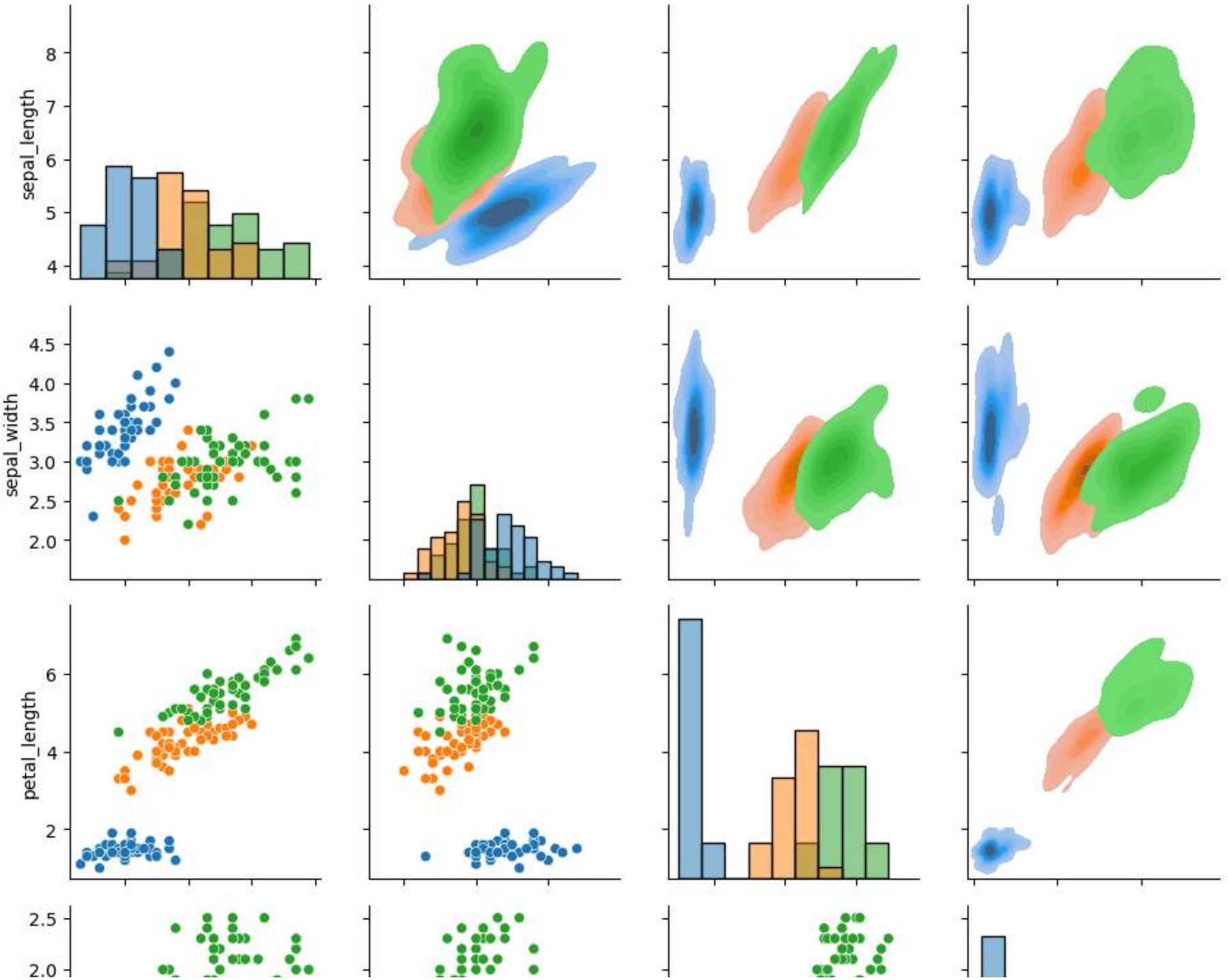
Out[56]:
```
<seaborn.axisgrid.PairGrid at 0x23ceb485110>
```

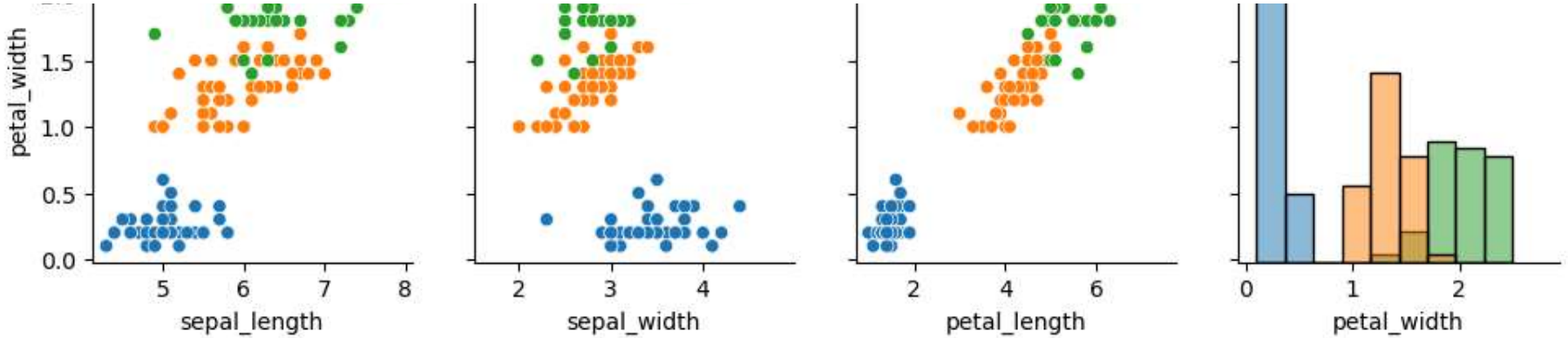

sepal_length
sepal_width
petal_length
8
7
6
5
4
4.5
4.0
3.5
3.0
2.5
2.0
6
4
2
2.5


petal_width
2.0
1.5
1.0
0.5
0.0
4
6
8
sepal_length
2
3
4
sepal_width
2
4
6
petal_length
0
1
2
petal_width

In [57]:
```
g3=sns.PairGrid(data=iris,hue='species')
g3.map_diag(sns.histplot)
g3.map_upper(sns.kdeplot,fill=True)
g3.map_lower(sns.scatterplot)
```

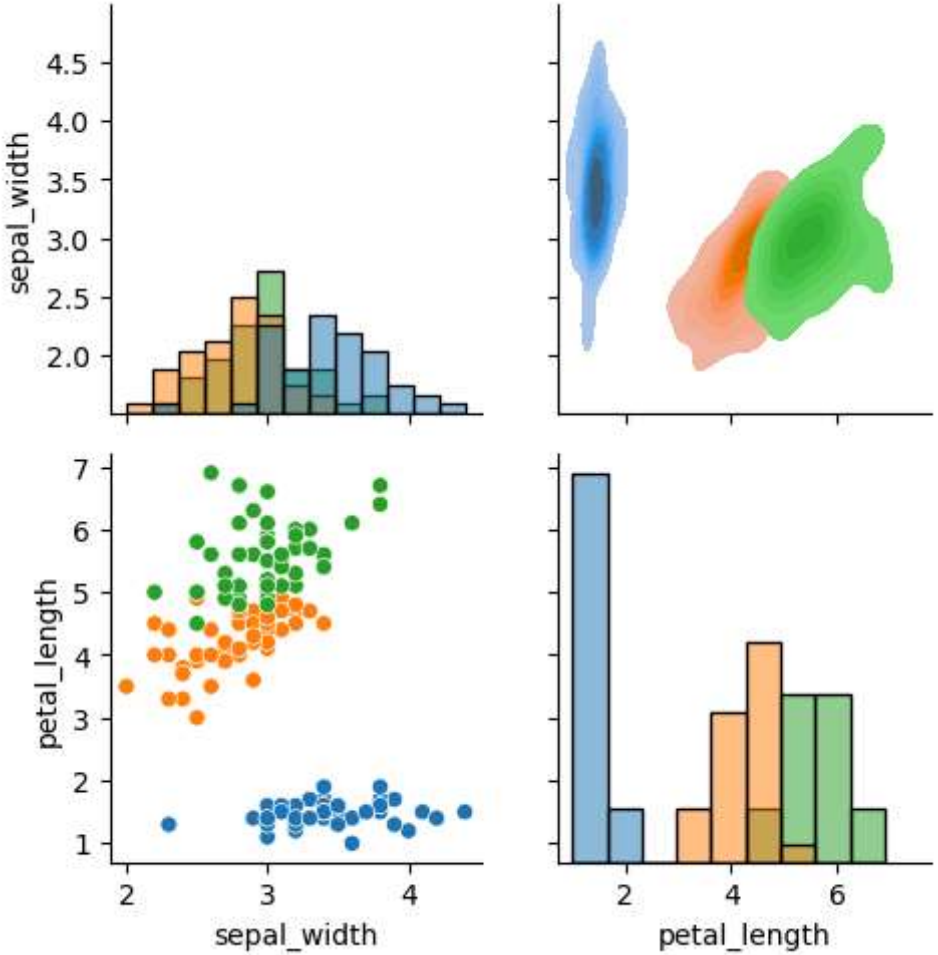
Out[57]:
```
<seaborn.axisgrid.PairGrid at 0x23cfbcd5110>
```


sepal_length
sepal_width
petal_length
8
7
6
5
4
4.5
4.0
3.5
3.0
2.5
2.0
6
4
2
2.5
2.0


petal_width
0.0
0.5
1.0
1.5
5
6
7
8
sepal_length
2
3
4
sepal_width
2
4
6
petal_length
0
1
2
petal_width

In [58]:
```
g4=sns.PairGrid(data=iris,hue='species',vars=['sepal_width','petal_length'])
g4.map_diag(sns.histplot)
g4.map_upper(sns.kdeplot,fill=True)
g4.map_lower(sns.scatterplot)
```

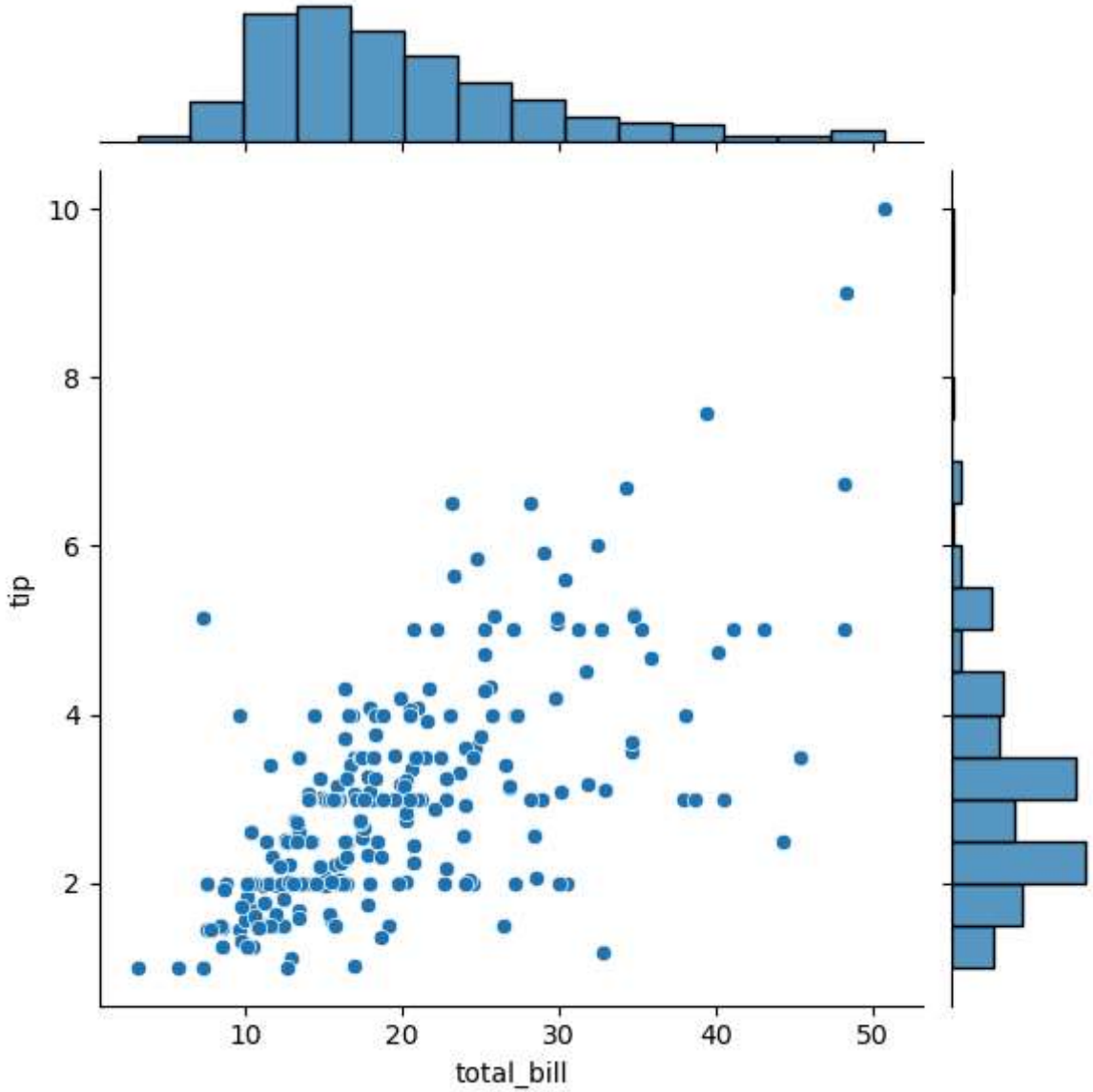
Out[58]: <seaborn.axisgrid.PairGrid at 0x23cfe5ed950>



Joint plot and joint grid

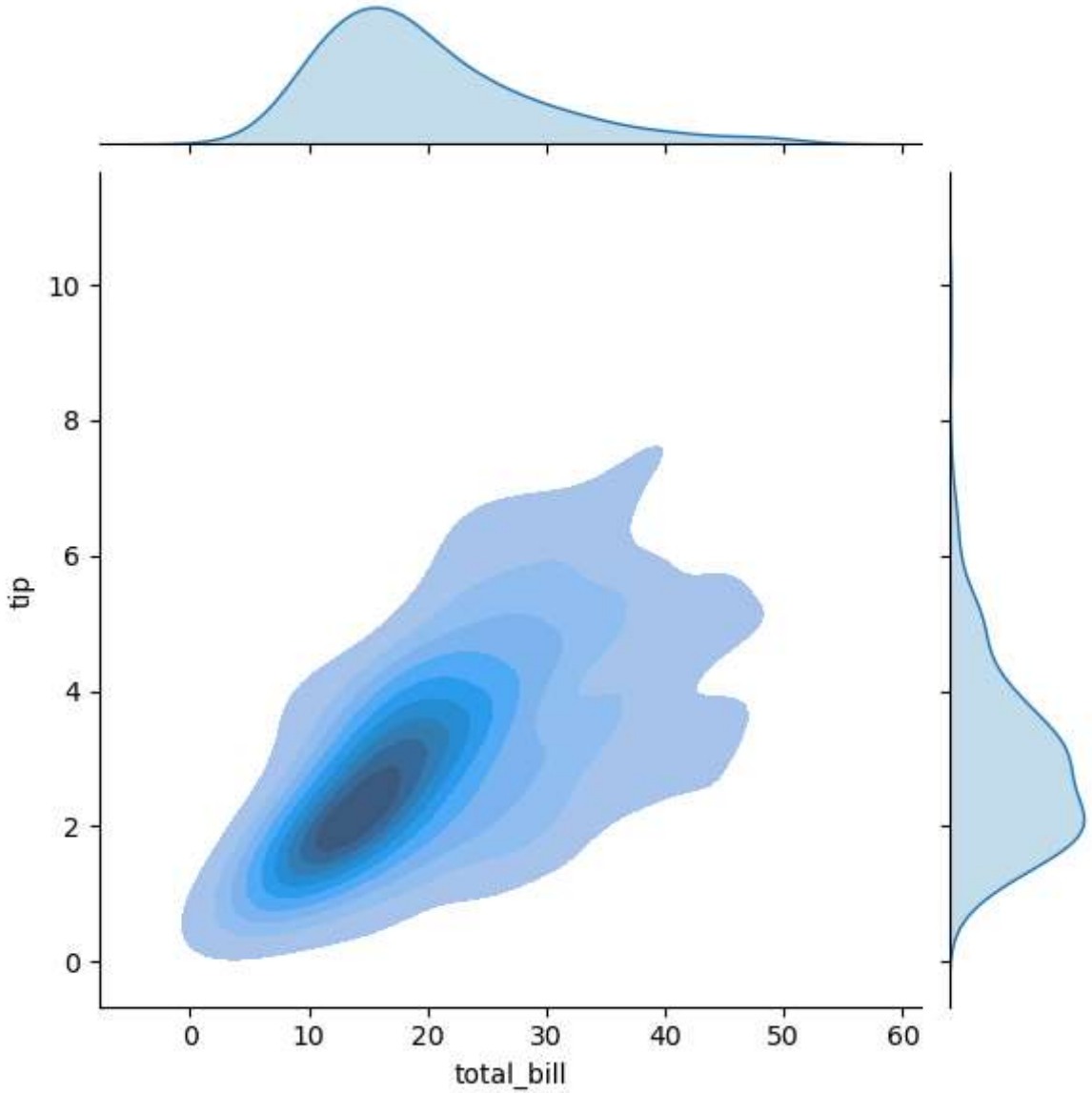
In [59]:
```
sns.jointplot(data=tips,x='total_bill',y='tip')
```

Out[59]:
```
<seaborn.axisgrid.JointGrid at 0x23cfd43e0d0>
```


10
8
6
4
2
tip
10
20
30
40
50
total_bill

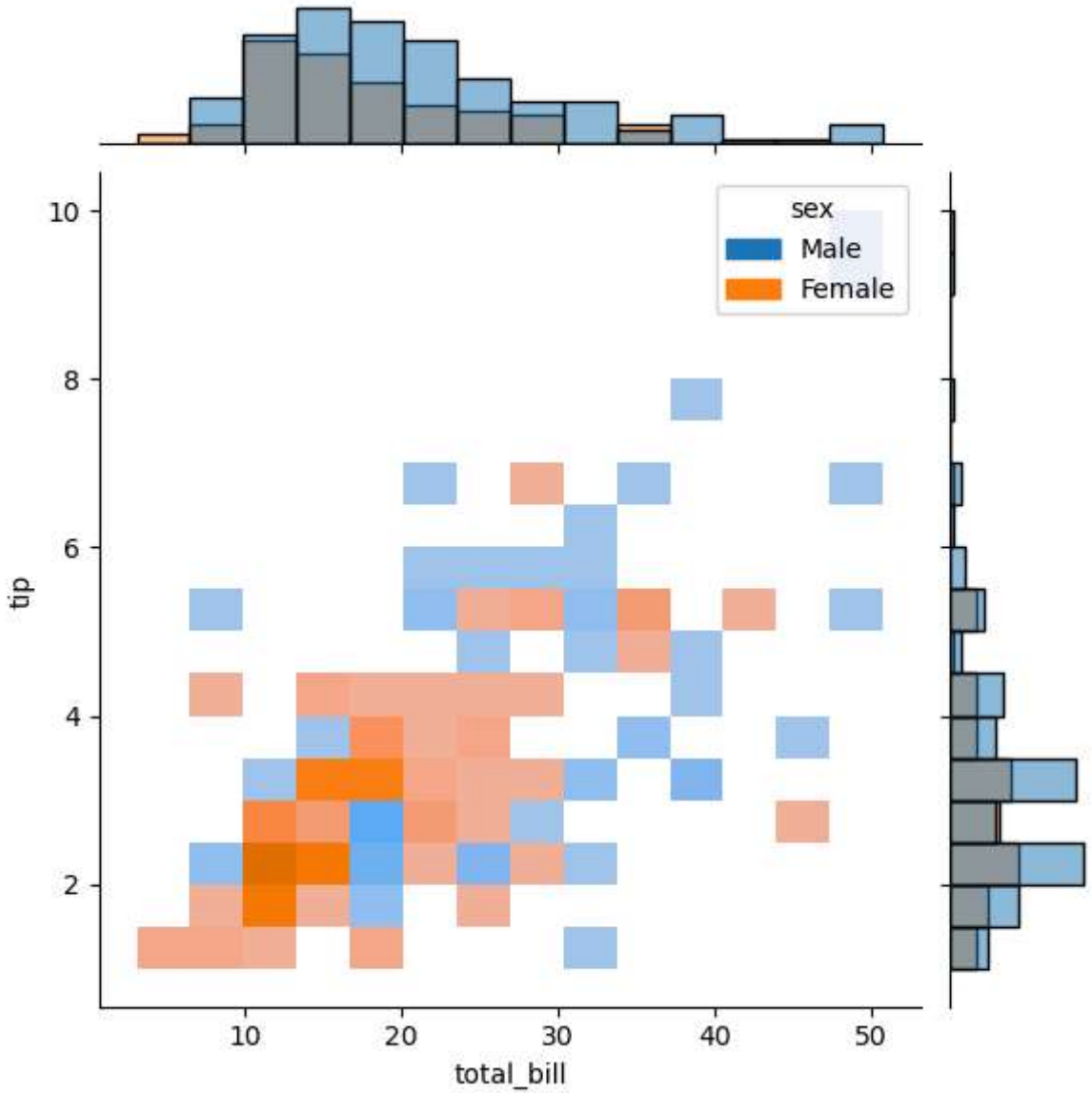
In [62]:
```
sns.jointplot(data=tips,x='total_bill',y='tip',kind='kde',fill=True)
```

Out[62]:
```
<seaborn.axisgrid.JointGrid at 0x23cff9af710>
```



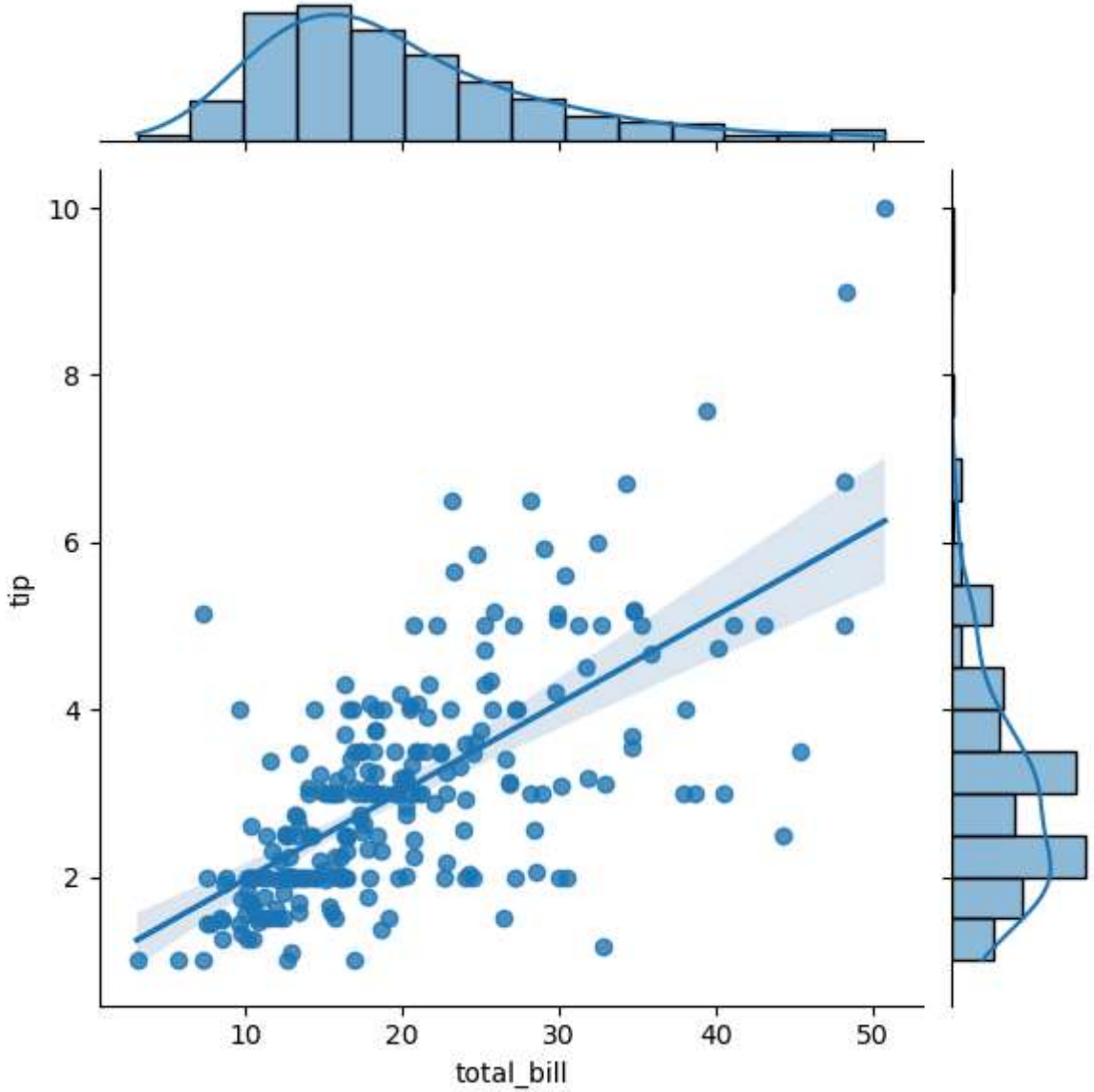
In [67]:
```
sns.jointplot(data=tips,x='total_bill',y='tip',kind='hist',hue='sex')
```

Out[67]:
```
<seaborn.axisgrid.JointGrid at 0x23c80727550>
```



In [70]:
```
sns.jointplot(data=tips,x='total_bill',y='tip',kind='reg')
```

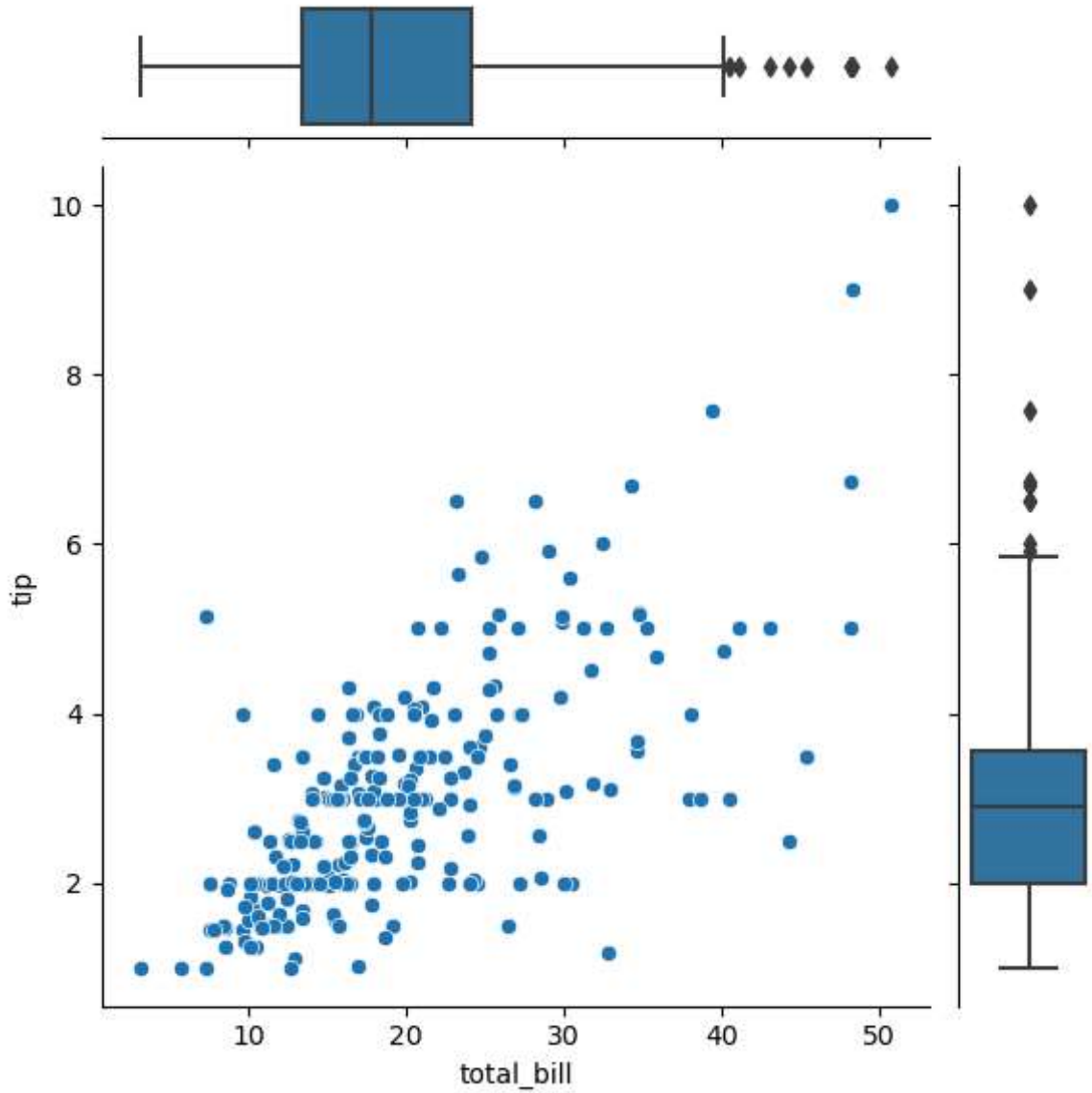
Out[70]:
```
<seaborn.axisgrid.JointGrid at 0x23c821075d0>
```



joint Grid

In [71]:
```
g6=sns.JointGrid(data=tips,x='total_bill',y='tip')
g6.plot(sns.scatterplot,sns.boxplot)
```

Out[71]:
```
<seaborn.axisgrid.JointGrid at 0x23c80e9ba50>
```



utility functions

In [72]:
```
sns.get_dataset_names()
```

Out[72]:
```
['anagrams',
 'anscombe',
 'attention',
 'brain_networks',
 'car_crashes',
 'diamonds',
 'dots',
 'dowjones',
 'exercise',
 'flights',
 'fmri',
 'geyser',
 'glue',
 'healthexp',
 'iris',
 'mpg',
 'penguins',
 'planets',
 'seaice',
 'taxis',
 'tips',
 'titanic',
 'anagrams',
 'anagrams',
 'anscombe',
 'anscombe',
 'attention',
 'attention',
 'brain_networks',
 'brain_networks',
 'car_crashes',
 'car_crashes',
 'diamonds',
 'diamonds',
 'dots',
 'dots',
 'dowjones',
 'dowjones',
 'exercise',
 'exercise',
 'flights',
```

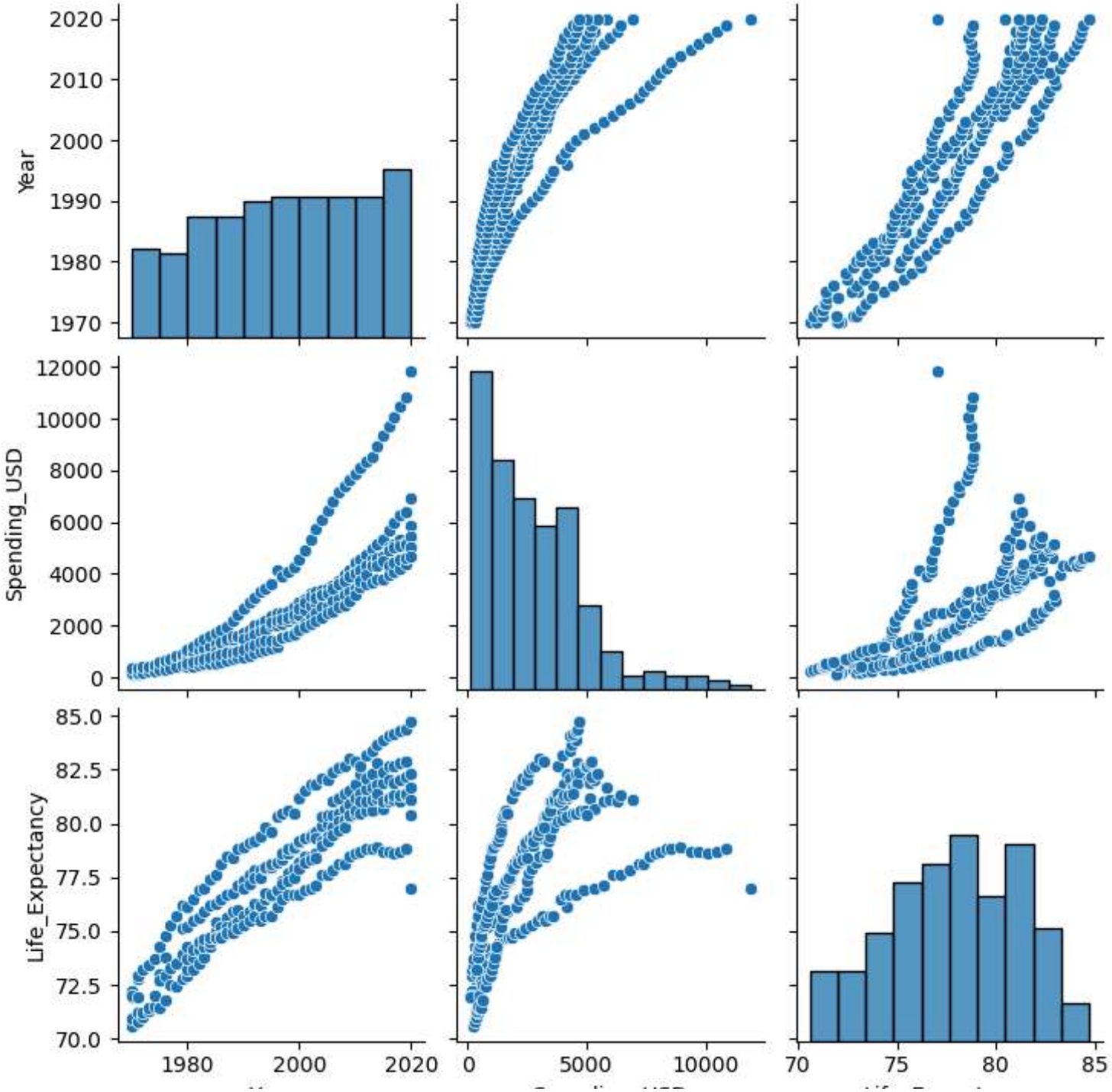
```
'flights',
'fmri',
'fmri',
'geyser',
'geyser',
'glue',
'glue',
'healthexp',
'healthexp',
'iris',
'iris',
'mpg',
'mpg',
'penguins',
'penguins',
'planets',
'planets',
'seaice',
'seaice',
'taxis',
'taxis',
'tips',
'tips',
'titanic',
'titanic',
'anagrams',
'anscombe',
'attention',
'brain_networks',
'car_crashes',
'diamonds',
'dots',
'dowjones',
'exercise',
'flights',
'fmri',
'geyser',
'glue',
'healthexp',
'iris',
'mpg',
'penguins',
```

```
 'planets',
 'seaice',
 'taxis',
 'tips',
 'titanic']
```

In [75]:
```
health=sns.load_dataset('healthexp')
```

In [76]:
```
sns.pairplot(health)
```

Out[76]:
```
<seaborn.axisgrid.PairGrid at 0x23c826c1c50>
```


Year
Spending_USD
Life_Expectancy
2020
2010
2000
1990
1980
1970
12000
10000
8000
6000
4000
2000
0
85.0
82.5
80.0
77.5
75.0
72.5
70.0
1980
2000
2020
0
5000
10000
70
75
80
85

Year	Spending_USD	Life_Expectancy

In [ ]: