

Joinplot - Pariplot

```
In [ ]: import seaborn as sns
import matplotlib.pyplot as plt
```

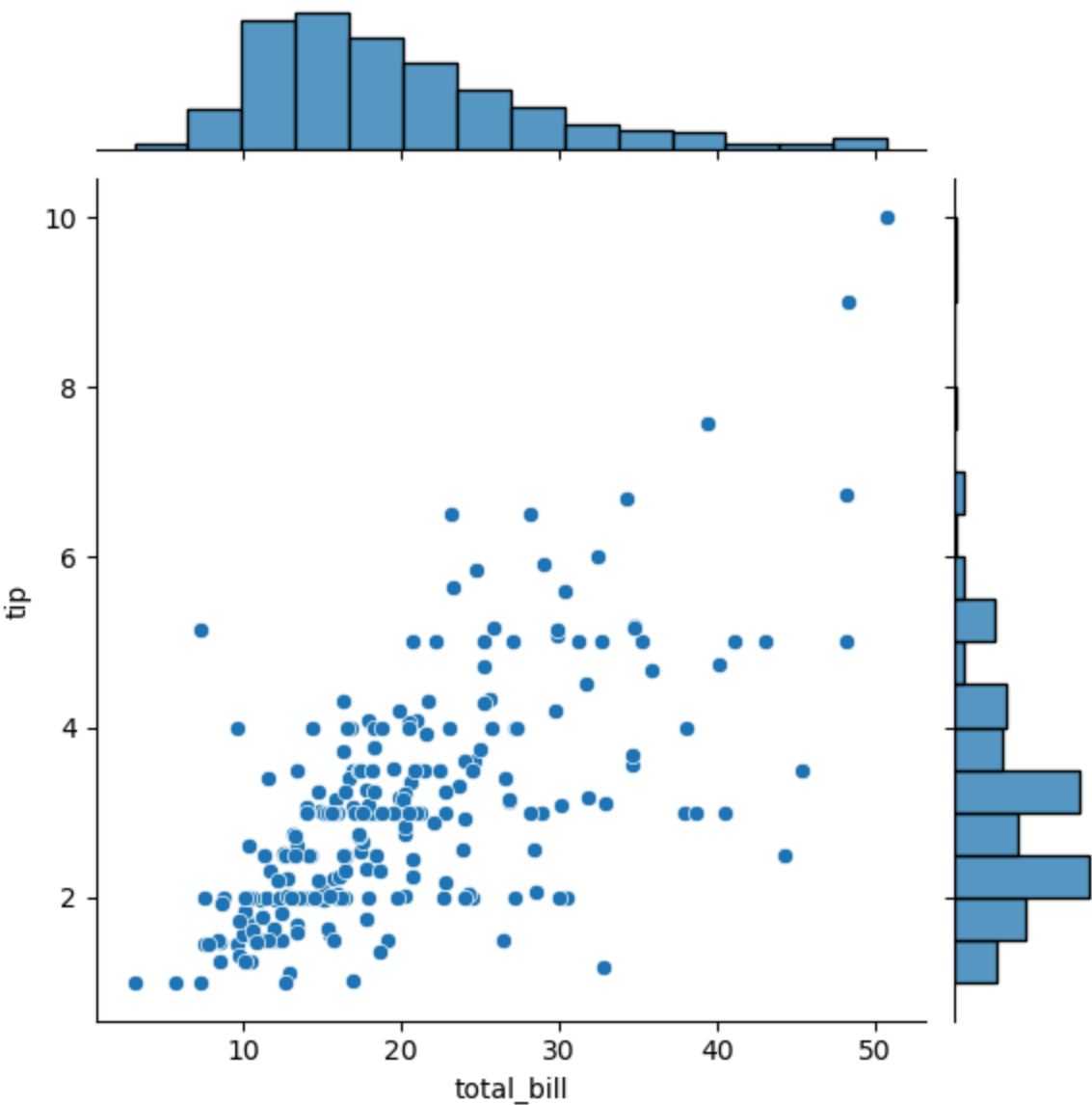
```
In [ ]: tips = sns.load_dataset('tips')
tips.head(3)
```

Out[]:

	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

Joinplot

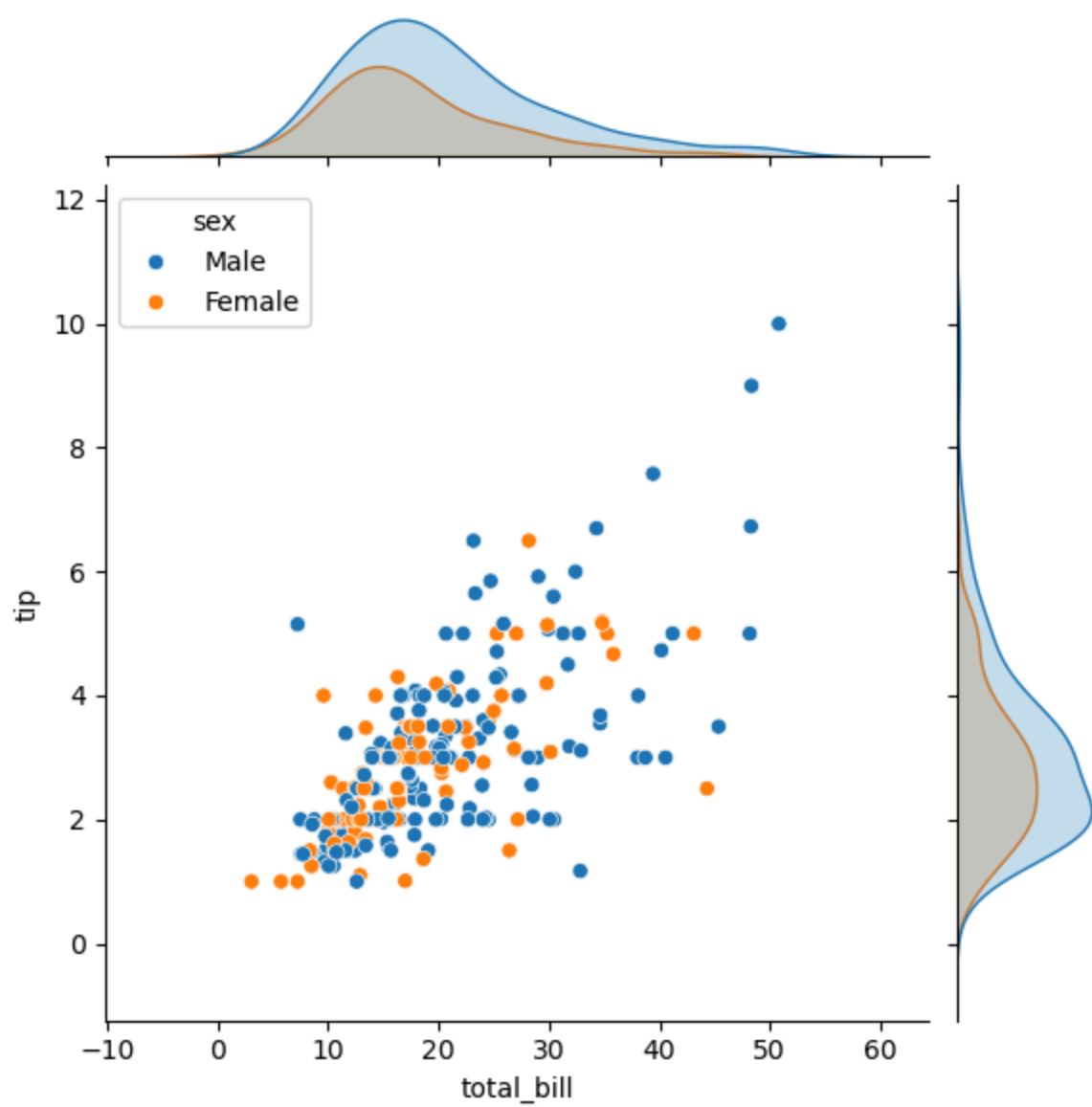
```
In [ ]: #Joinplot
sns.jointplot(data=tips,x='total_bill',y='tip')
plt.show()
```



Cómo se puede observar, junta un gráfico `scatter` y un `histograma` , que relaciona 2 variables. Nos ayuda a visualizar que hay 2 histogramas una en el eje `x` y otro en el eje `y` .

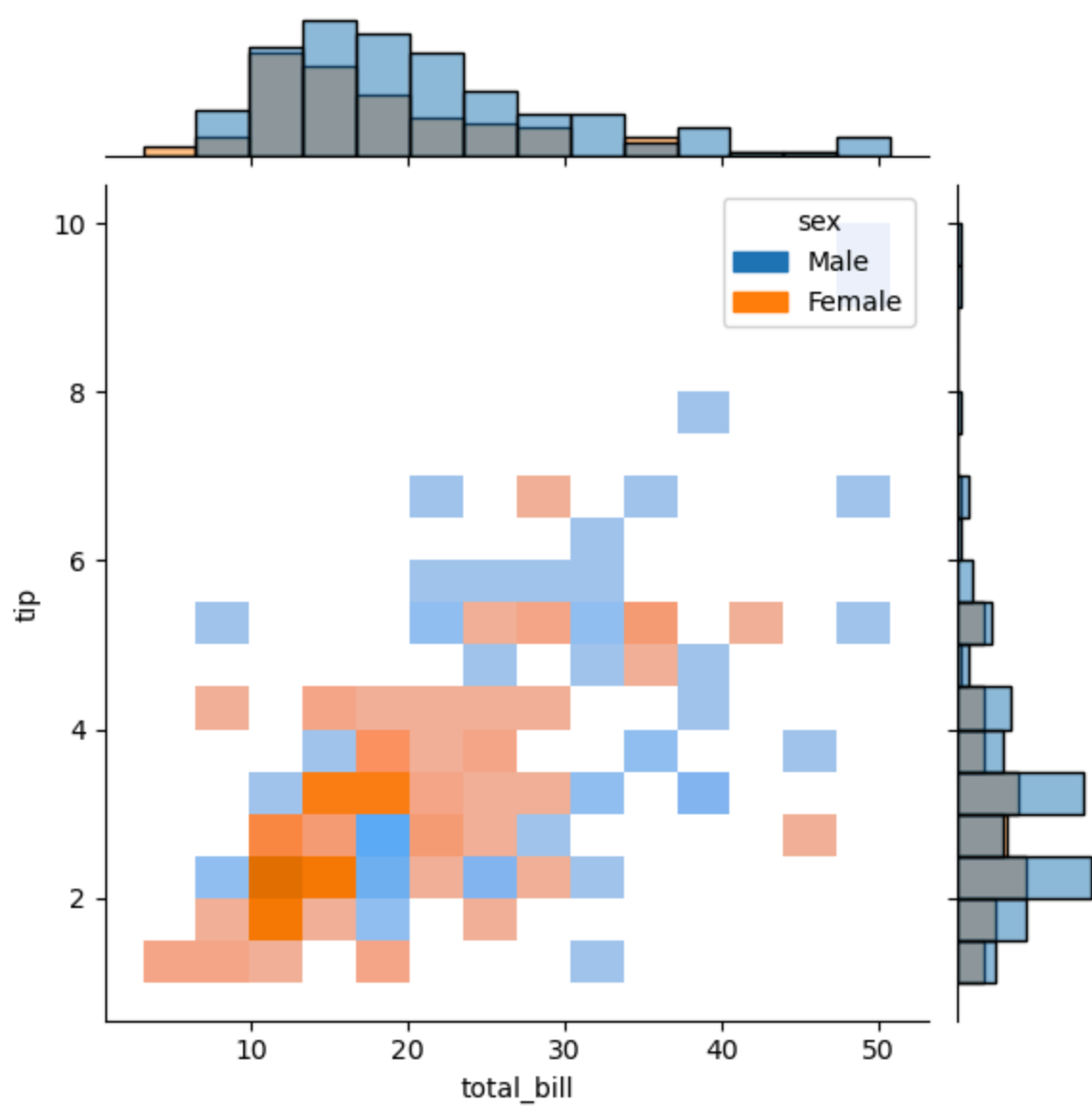
Personalizando

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

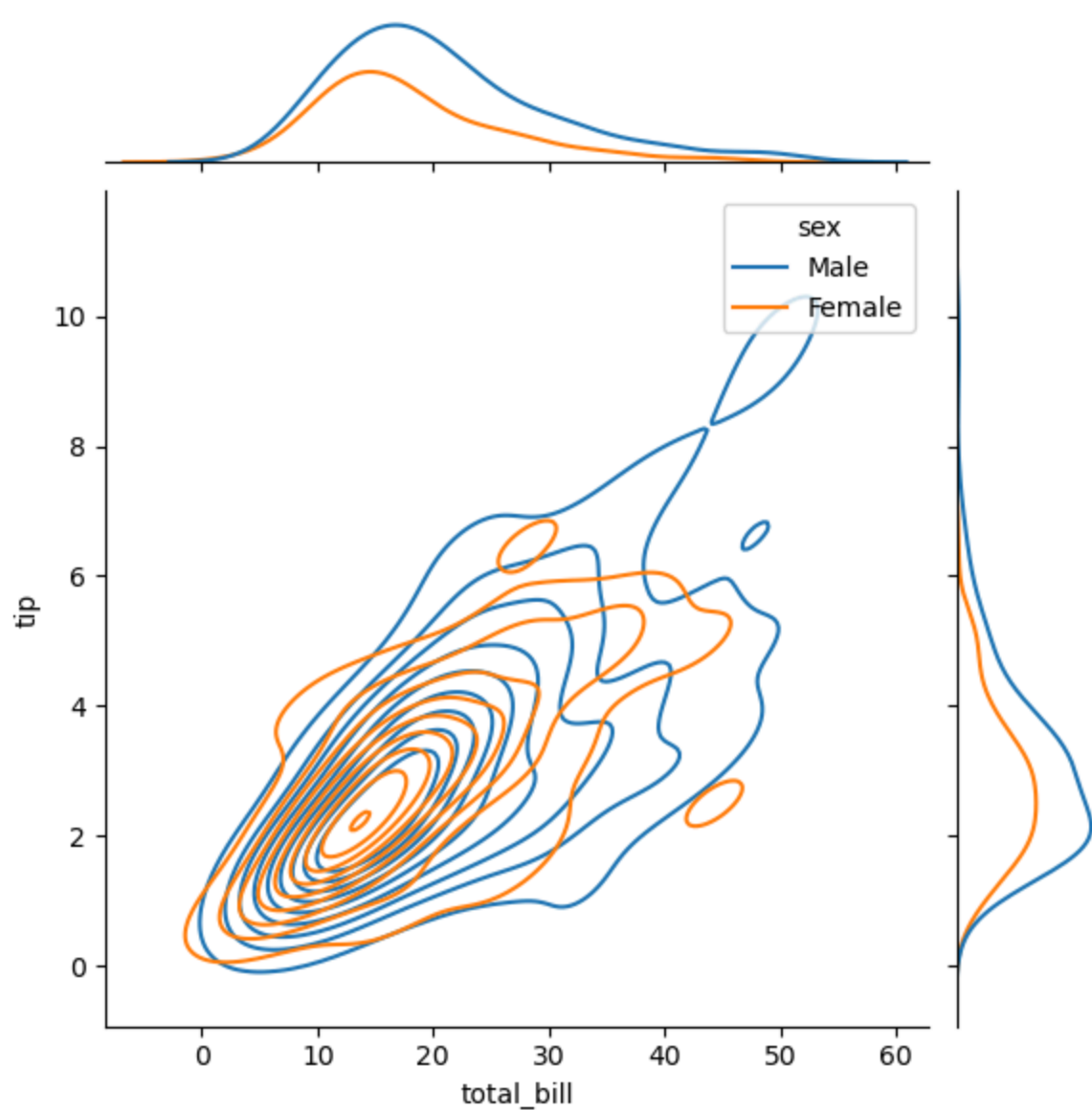


Aquí automáticamente, cambia dependiendo del diagrama más adecuado. 2 de tipo KDE y uno scatter .

```
In [ ]: #Especificando que hue='sex' se muestre como hist
#kind='hist'
sns.jointplot(data=tips,x='total_bill',y='tip',hue='sex',kind='hist')
plt.show()
```

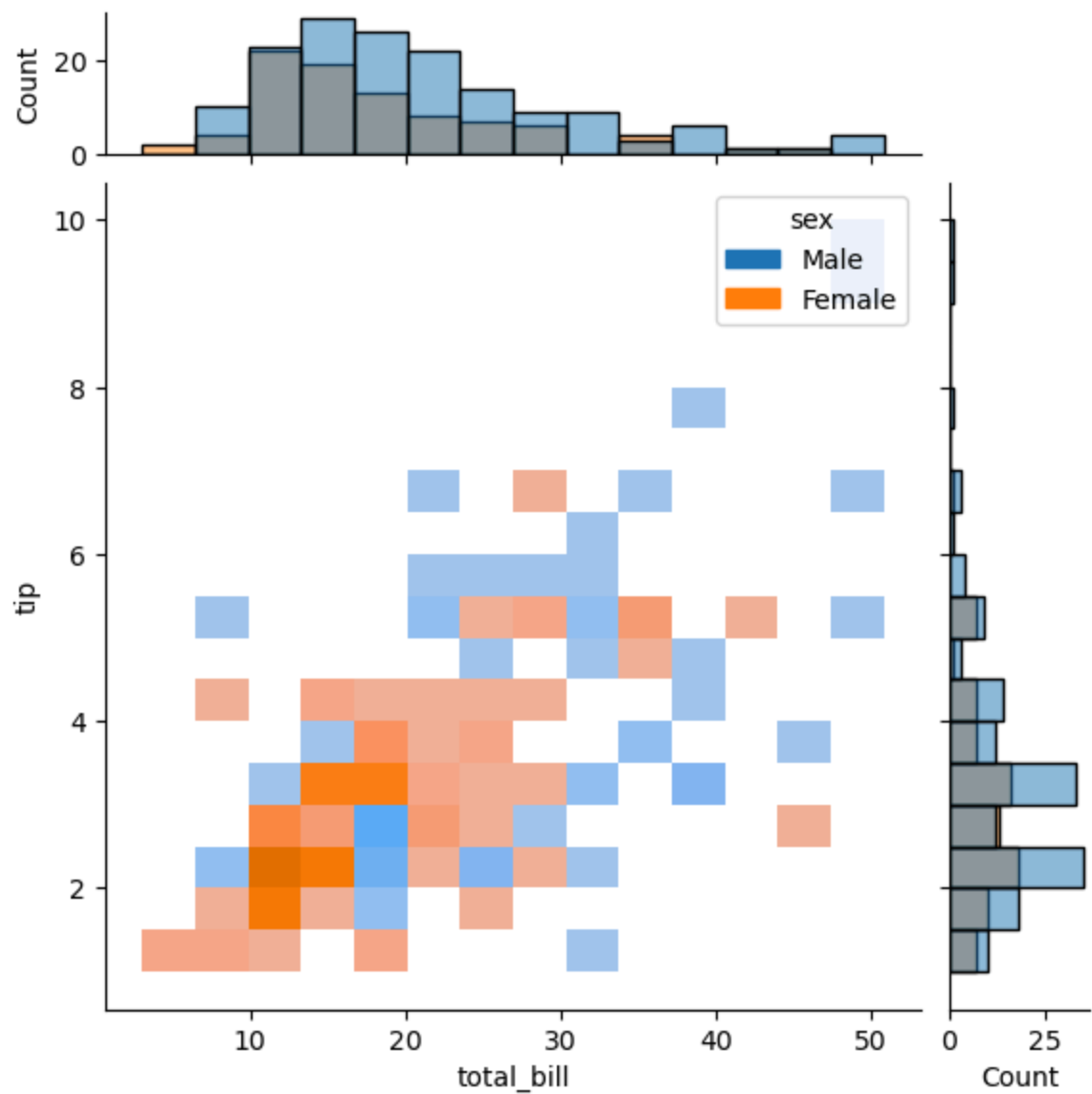


```
In [ ]: #Especificando que hue='sex' se muestre como kde
#kind='kde'
sns.jointplot(data=tips,x='total_bill',y='tip',hue='sex',kind='kde')
plt.show()
```



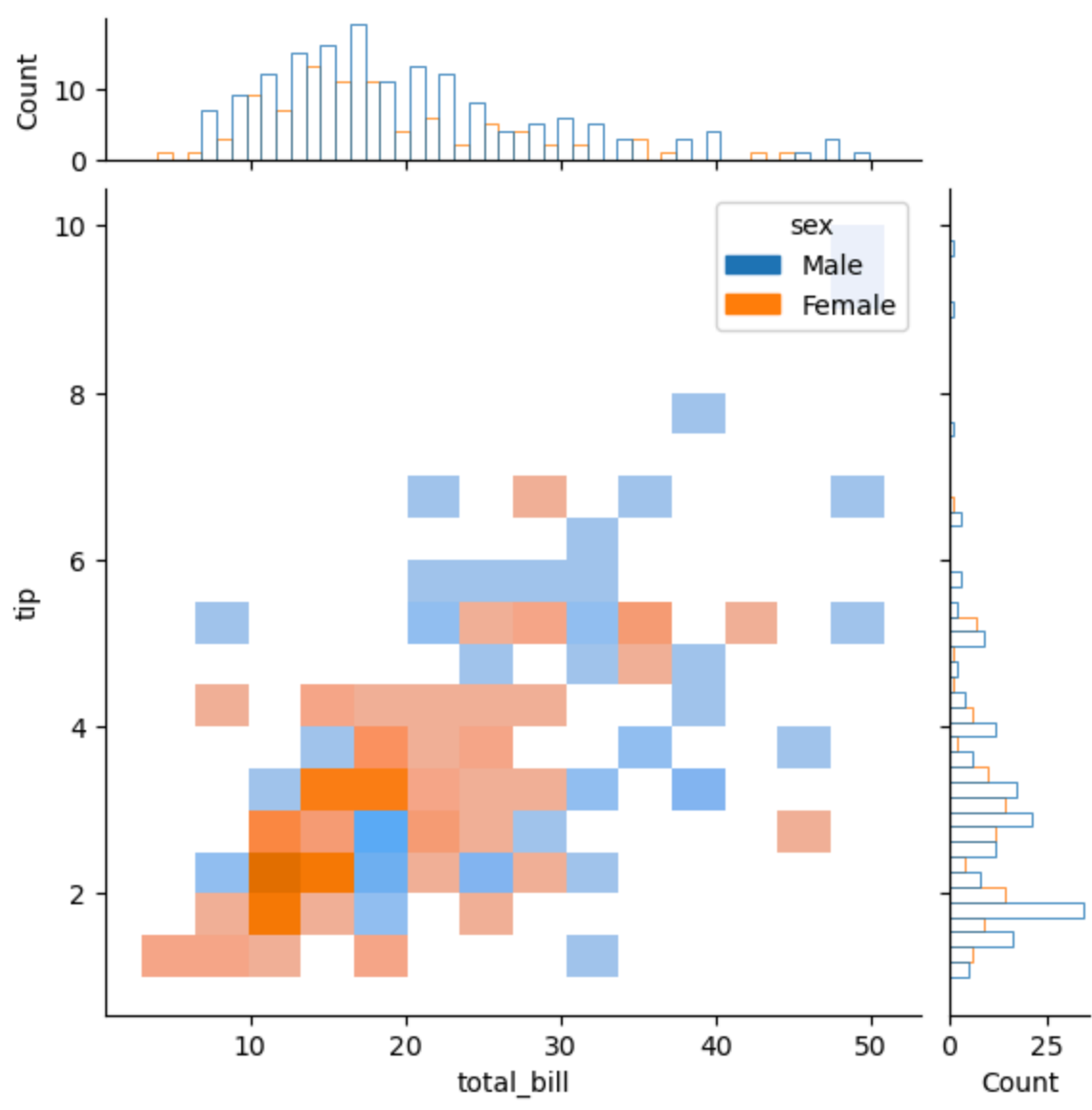
Marginal ticks=True

```
In [ ]: #Especificando que hue='sex' se muestre como hist
#kind='hist'
#Mas valor en los ejes
sns.jointplot(data=tips,x='total_bill',y='tip',hue='sex',kind='hist',marginal_ticks=True)
plt.show()
```



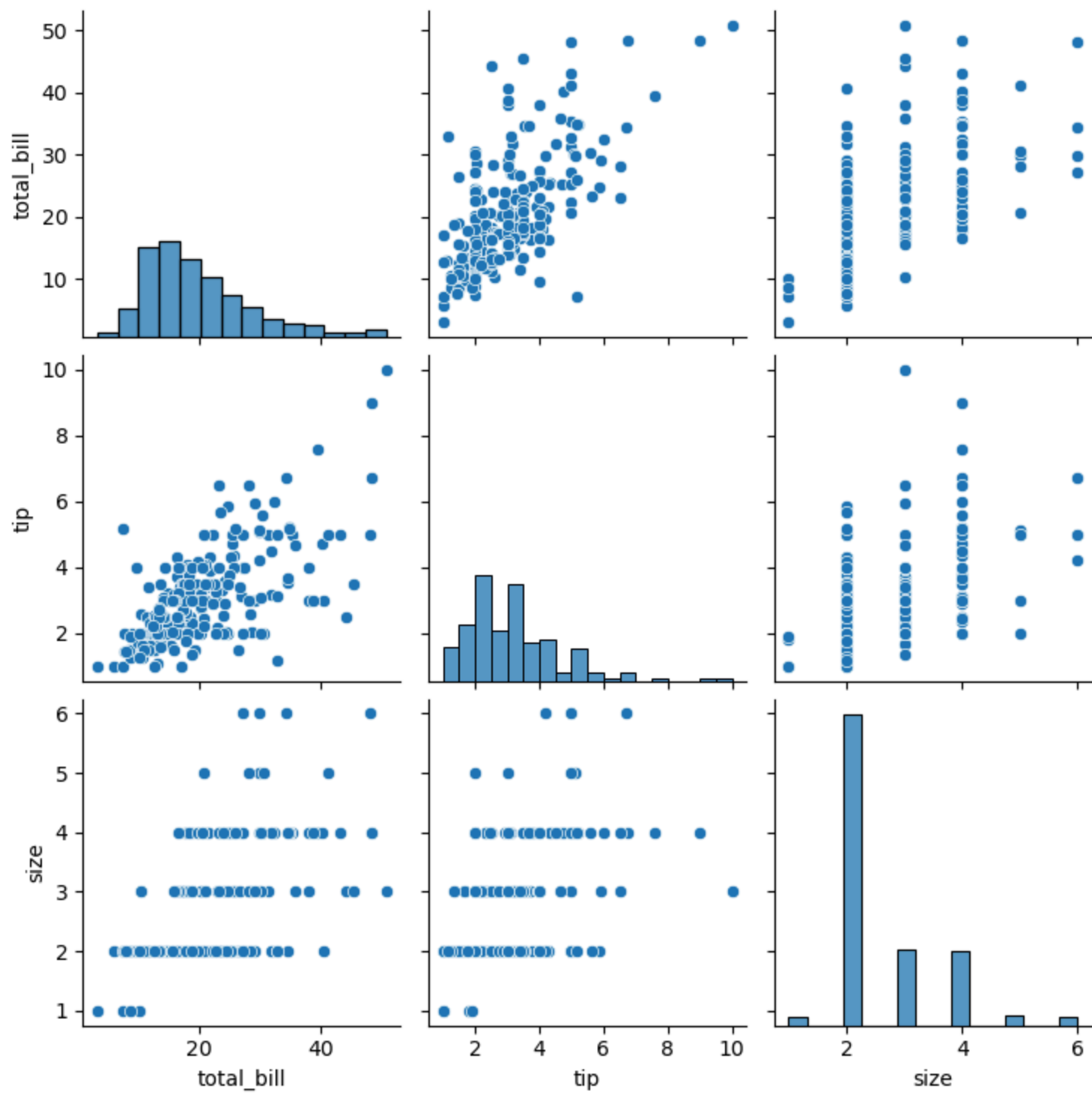
Marginal_kws

```
In [ ]: #Especificando que hue='sex' se muestre como hist
#kind='hist'
#Mas valor en los ejes
sns.jointplot(data=tips,x='total_bill',y='tip',hue='sex',kind='hist',marginal_ticks=True,
              marginal_kws=dict(bins=25,fill=False,multiple='dodge'))
plt.show()
```



Pairplot

```
In [ ]: #Pairplot
sns.pairplot(data=tips)
plt.show()
```



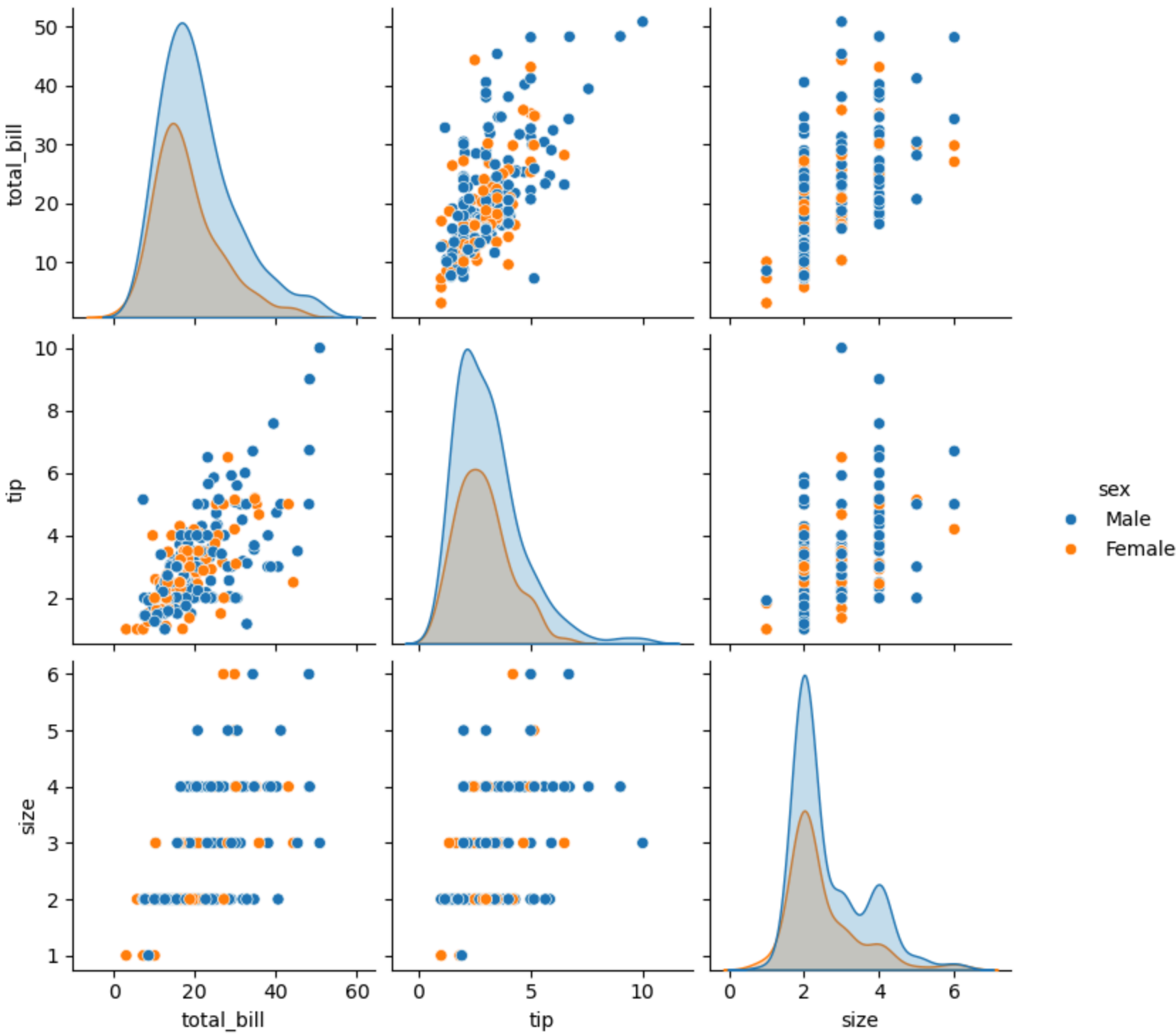
Explicación:

Este método toma `total_bill` , `tip` y `size` . Las 3 son variables numericas y sirven muy bien para la relación entre si, entonces `pairplot()` de manera automatica las relaciona entre si. Notas:

- Siempre la relación de la variable con si misma, nos dará un histograma.
- En otra forma puede hacer scatter plot

Personalizando

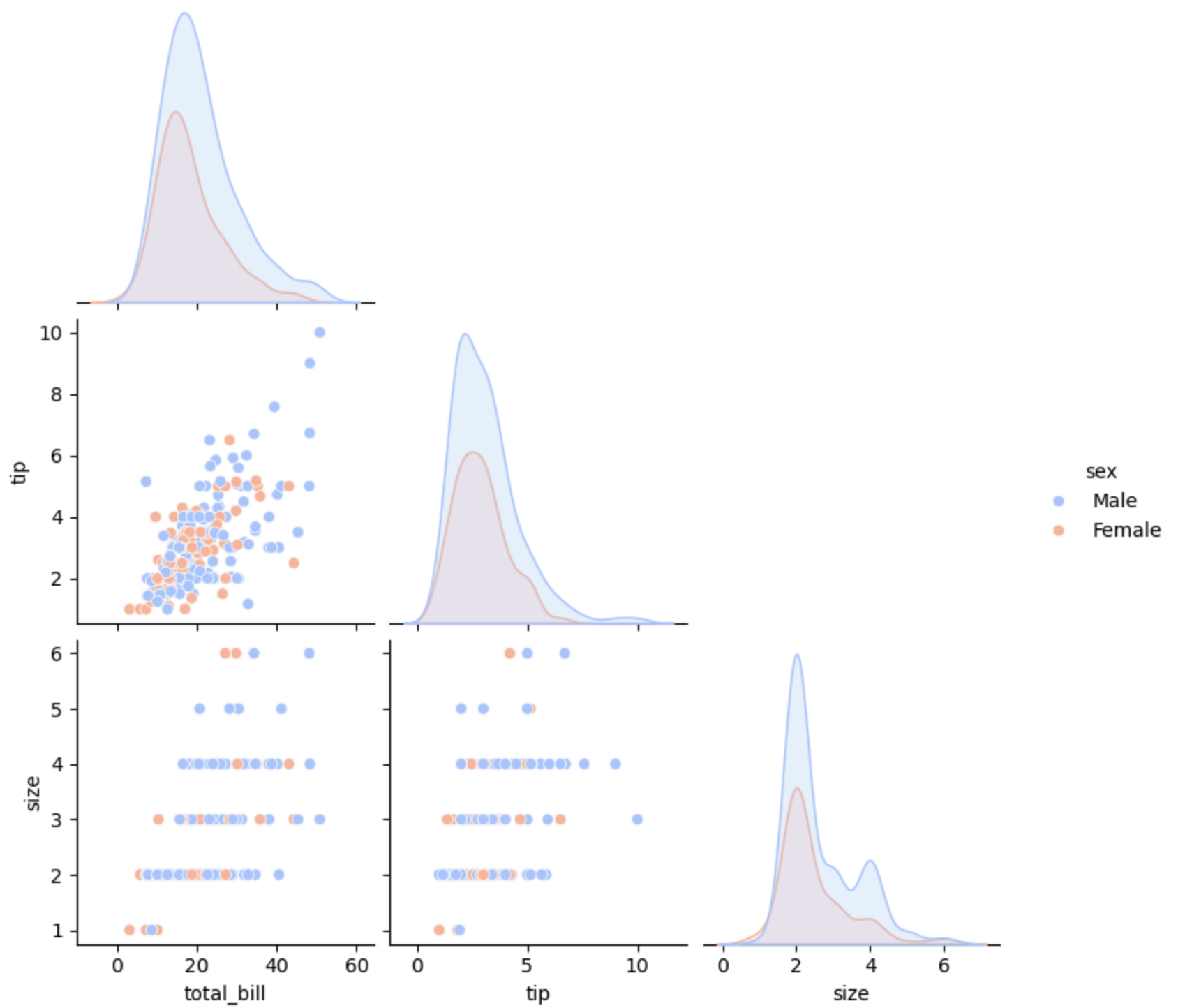
```
In [ ]: #Pairplot
sns.pairplot(data=tips,hue='sex')
plt.show()
```



Ahora considera que es mejor representar con un diagrama de KDE la relación de variable consigo mismo.

palette= Y corner=True

```
In [ ]: #Pairplot
sns.pairplot(data=tips,hue='sex',palette='coolwarm',corner=True)
plt.show()
```

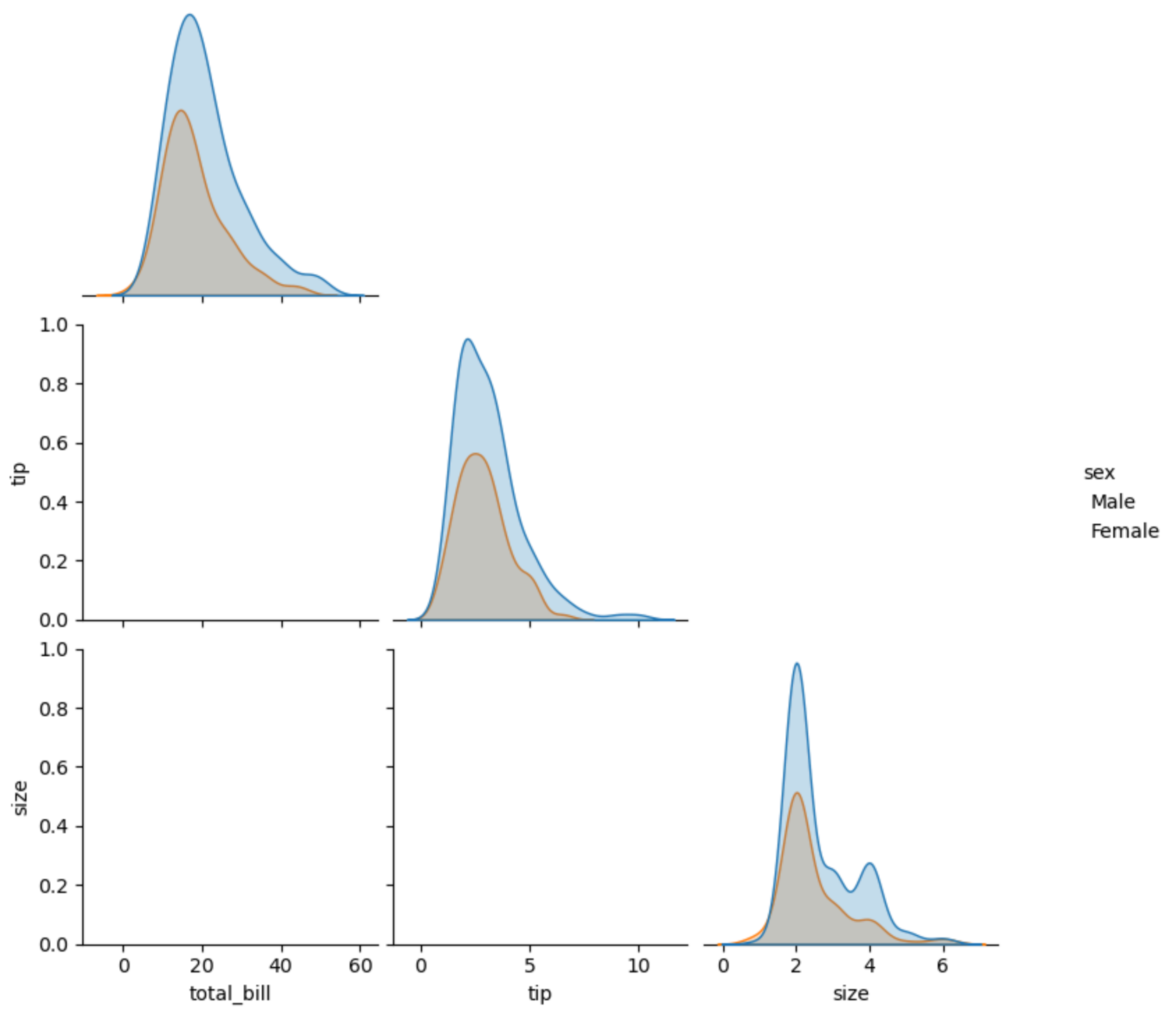


Explicación de Corner:

Corner me sirve para no tener información duplicada, es decir como se relacionan por pares, puede que existan relaciones duplicadas y pueden parecer gráficas diferentes, pero en realidad no lo son.

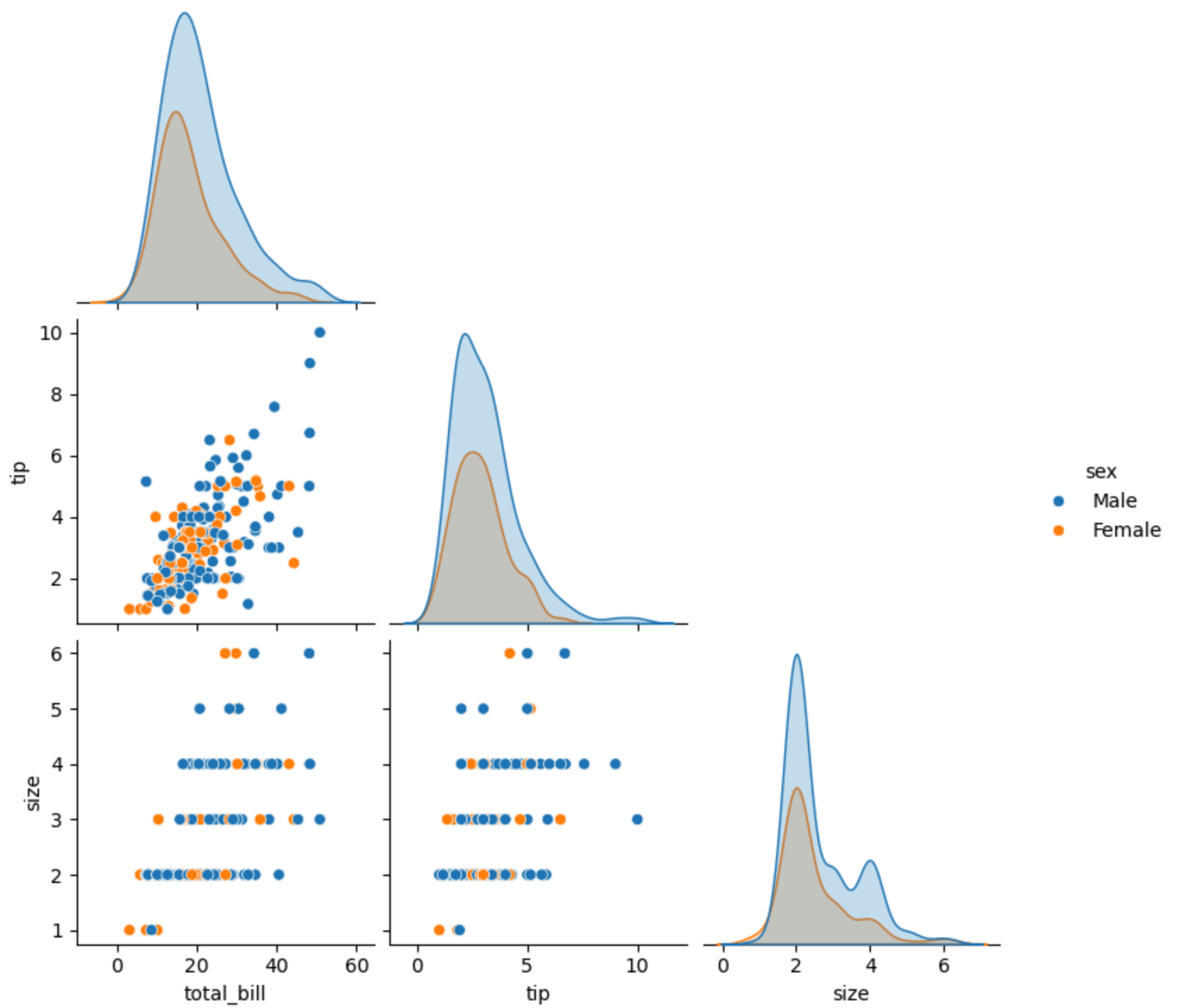
Kind='line'

```
In [ ]: #Pairplot
sns.pairplot(data=tips,hue='sex',corner=True,kind='line')
plt.show()
```



Kind='scatter'

```
In [ ]: #Pairplot
sns.pairplot(data=tips,hue='sex',corner=True,kind='scatter')
plt.show()
```



Esta es una herramienta muy útil a nivel de analizar tus datos a primera vista

Referencias:

- [Implot](#)
- [Pairplot](#)
- [Joinplot](#)