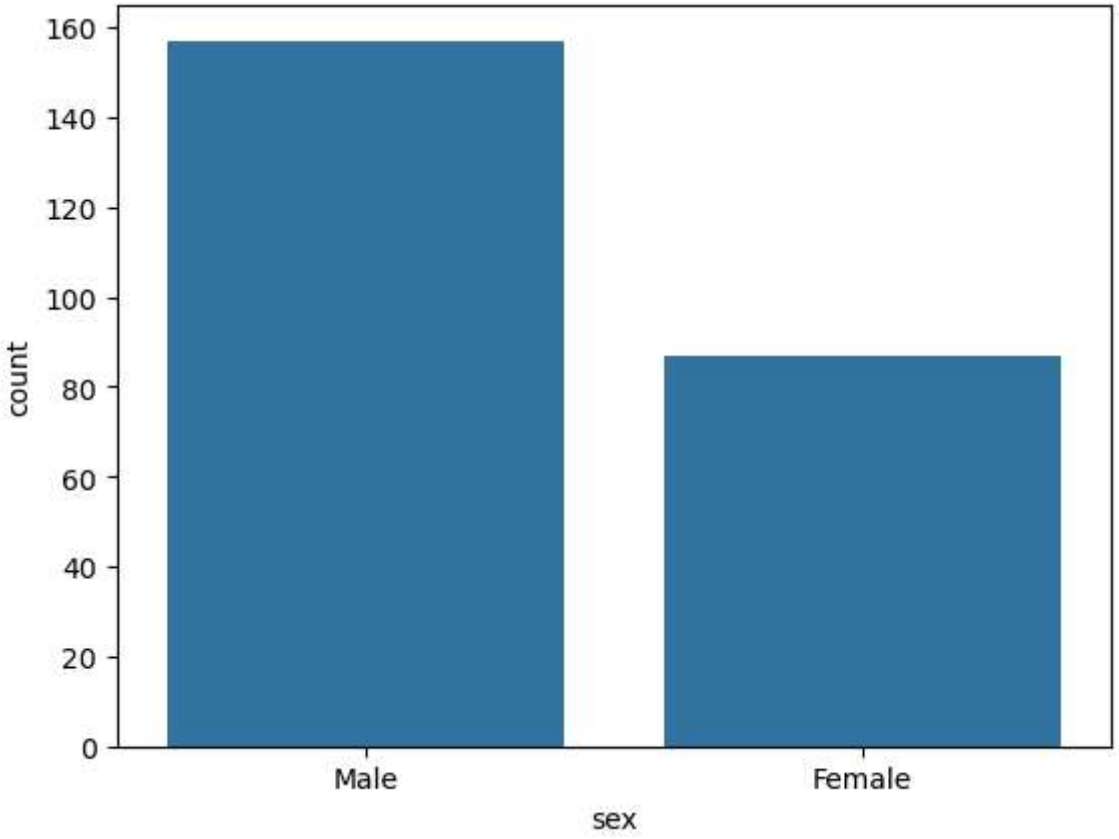
In [ ]:
```
# VIVEK-CHAUHAN-ADVANCED-DATA-ANALYTICS-SEABORN-COUNTPLOT-HISTOGRAM-PAIRPLOT
```

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

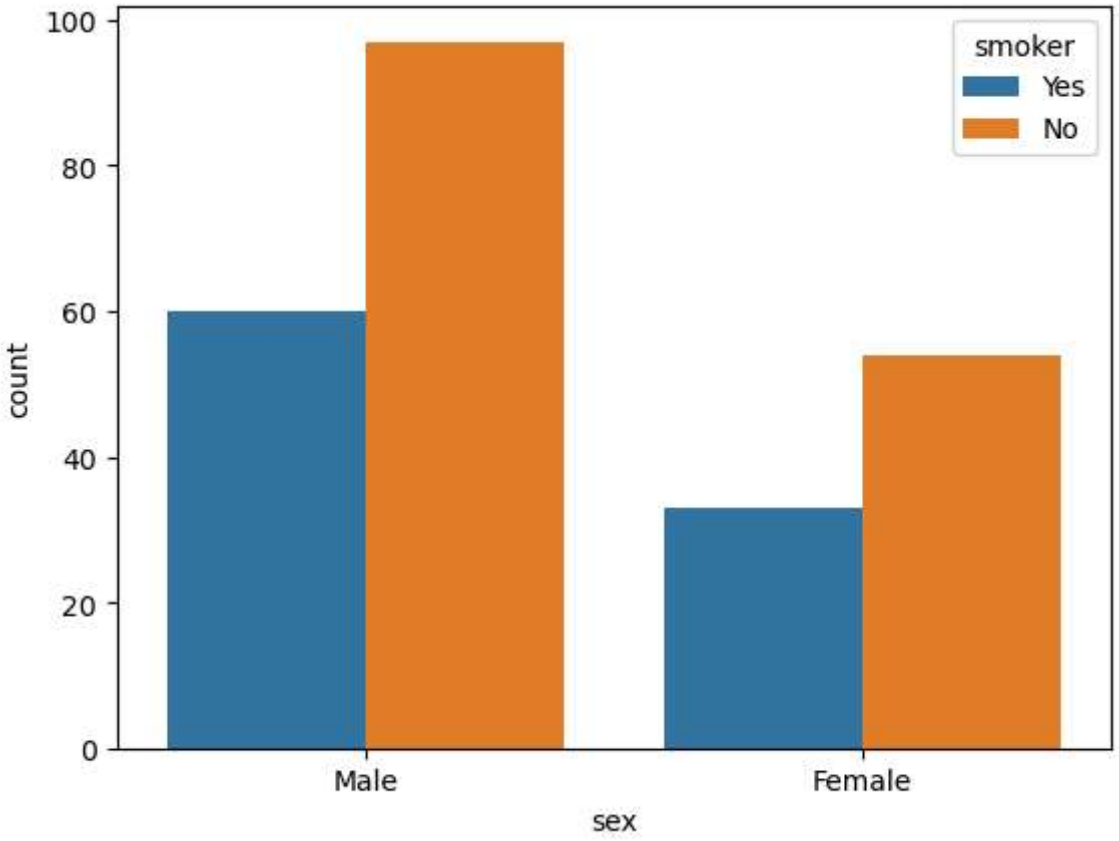
In [7]:
```
df = sns.load_dataset("tips")
print(df)
```

```
     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 x 7 columns]
```

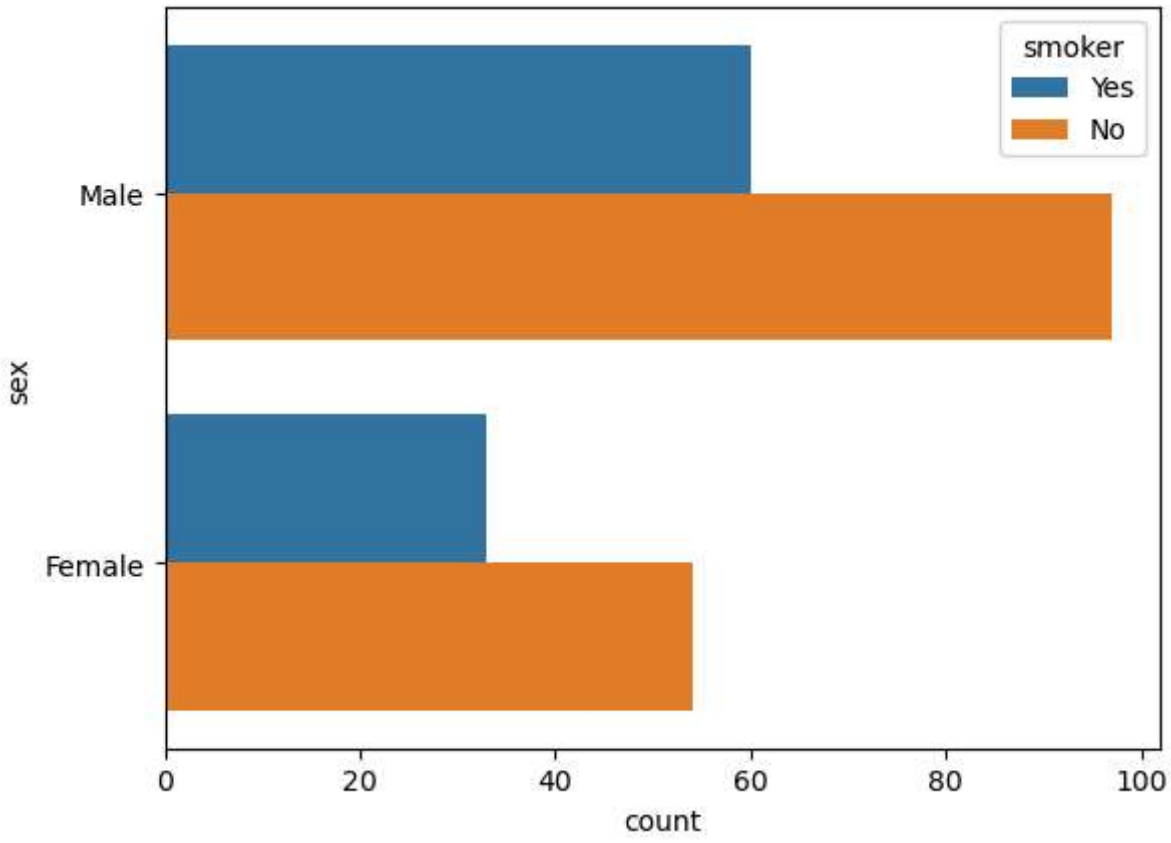
In [15]:
```
# count plot working on a single axis cause another axis is used for counting
# count plot on single categorical variable
a = sns.countplot(x="sex",data=df)
plt.show()
```



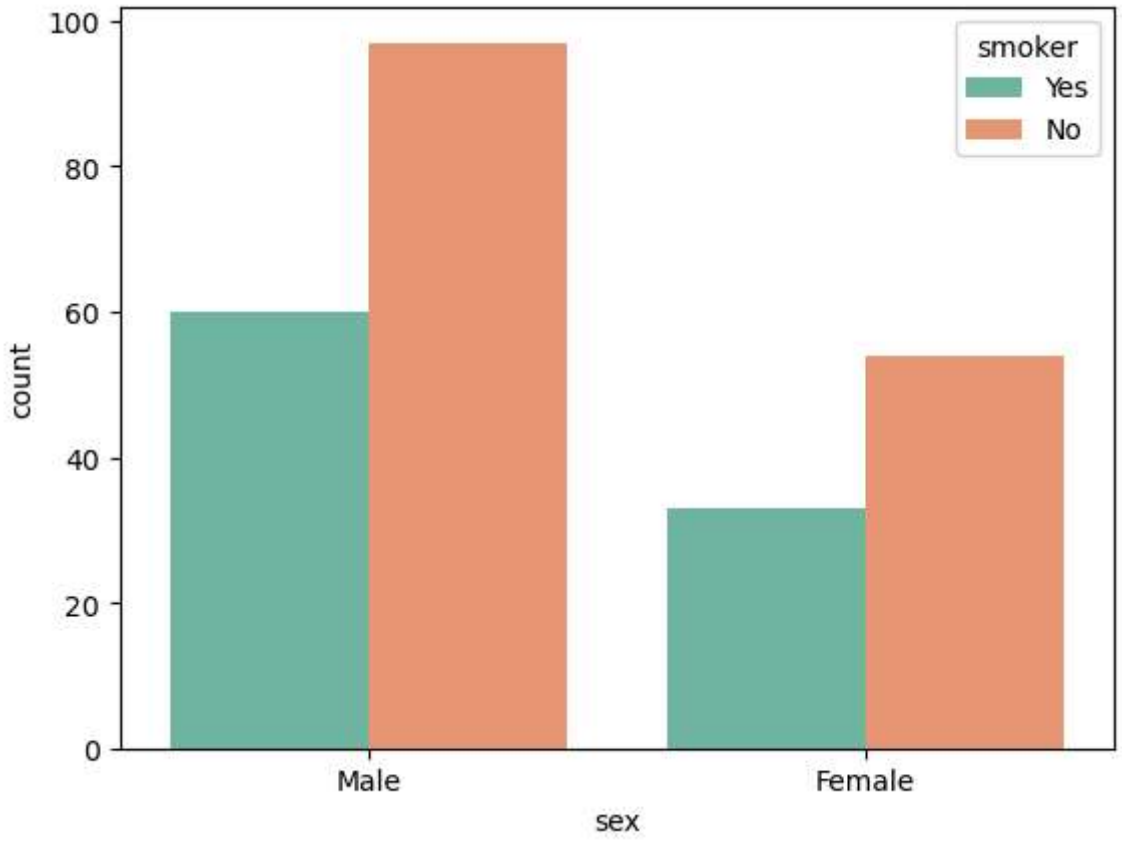
In [11]:
```
# count plot on two categorical variable
a = sns.countplot(x="sex",data=df,hue="smoker")
plt.show()
```



In [13]:
```
# now we can plot the bar horizontally just write y-axis instead of x-axis
# count plot on two categorical variable
a = sns.countplot(y="sex",data=df,hue="smoker")
plt.show()
```

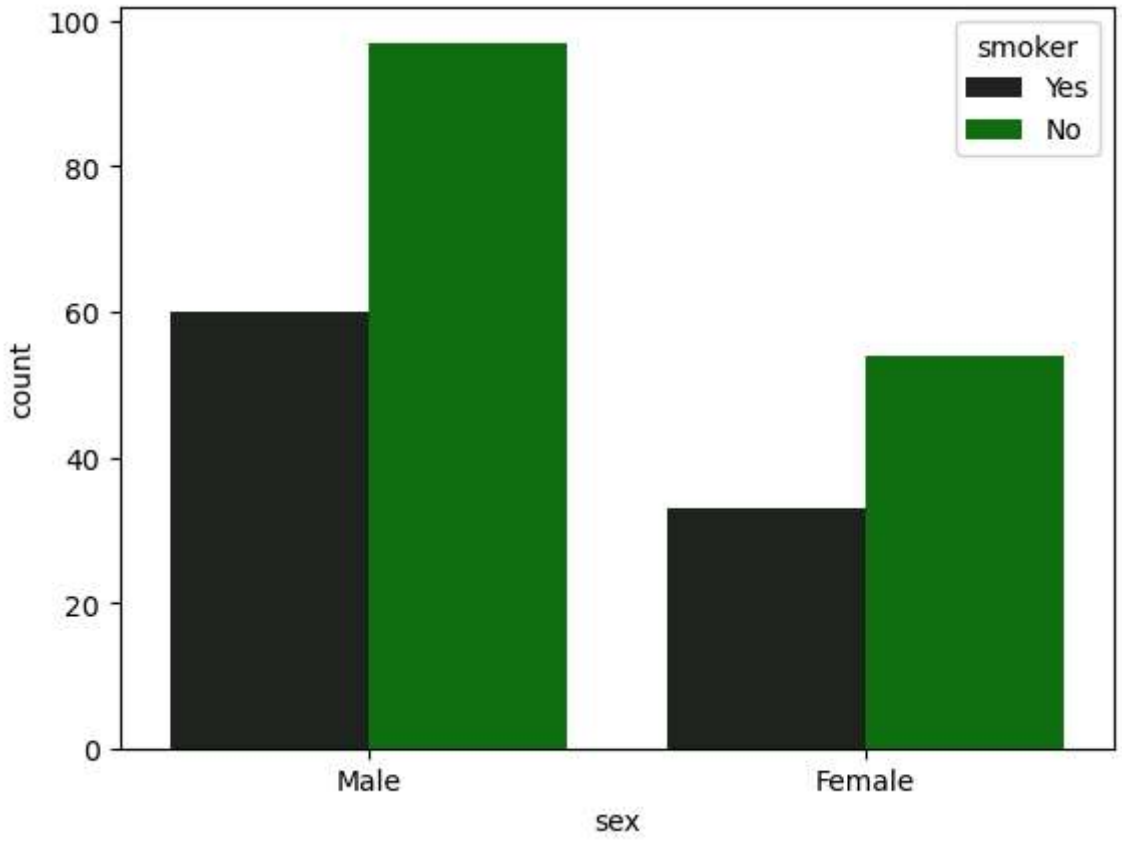


In [19]:
```
# we can use different color palette for designing
# count plot on two categorical variable
a = sns.countplot(x="sex",data=df,hue="smoker",palette="Set2")
plt.show()
```

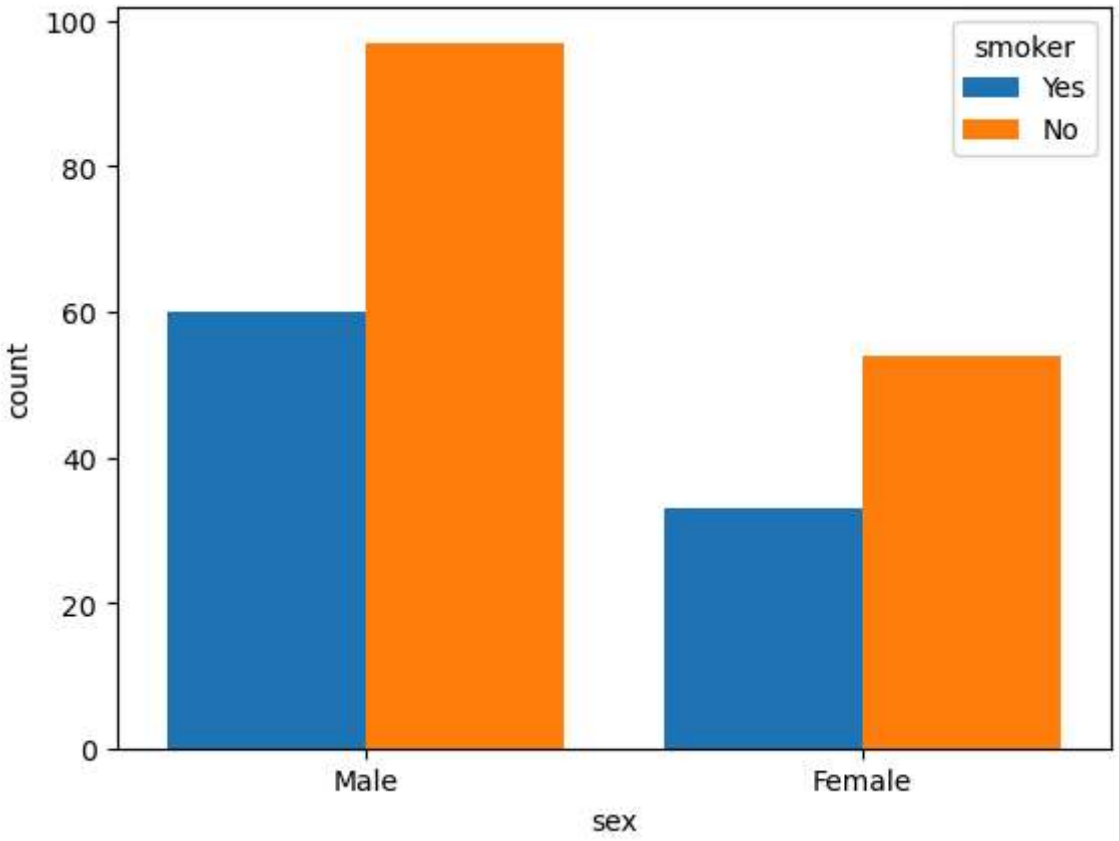


In [23]:
```
# using a color parameter in the plot
# count plot on two categorical variable
a = sns.countplot(x="sex",data=df,hue="smoker",color="green")
plt.show()
```

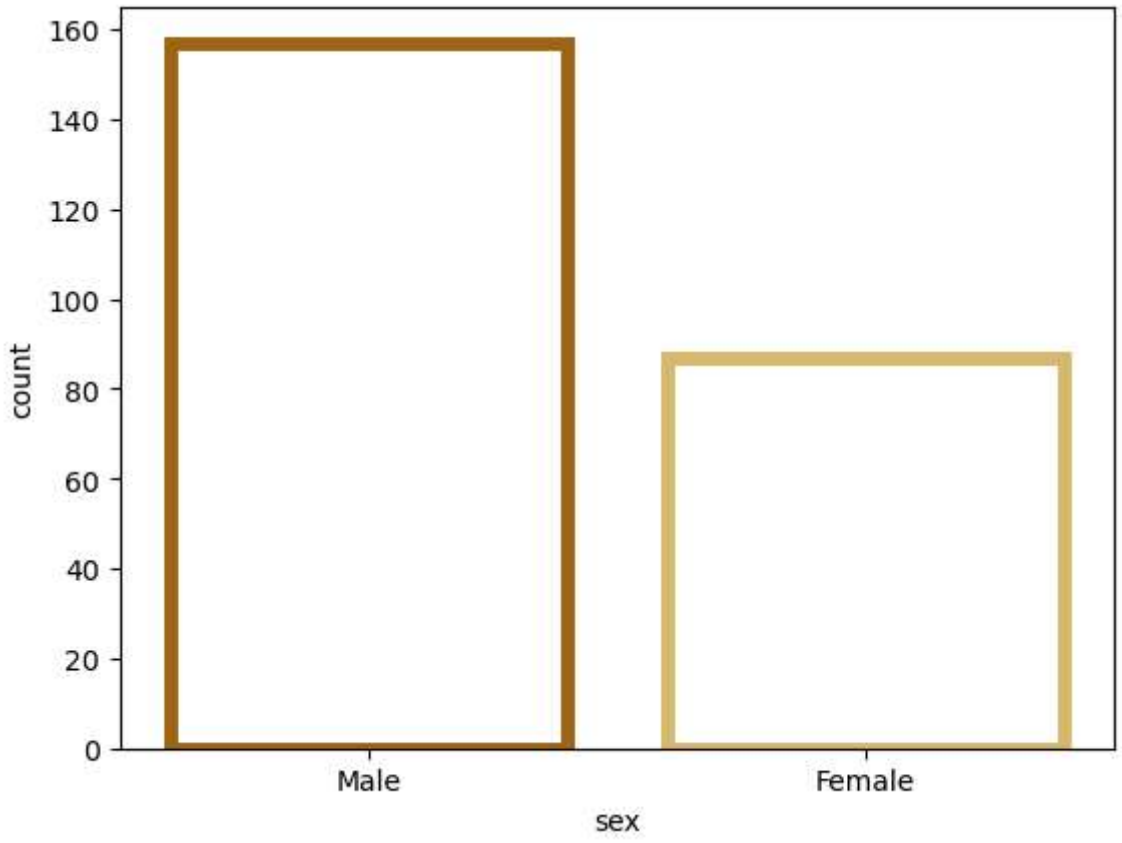
```
C:\Users\fv8.DESKTOP-N5HA3AQ\AppData\Local\Temp\ipykernel_10068\1220070453.py:3: Fut
ureWarning:

Setting a gradient palette using color= is deprecated and will be removed in v0.14.
0. Set `palette='dark:green'` for the same effect.

  a = sns.countplot(x="sex",data=df,hue="smoker",color="green")
```



In [51]:
```
# we can use the saturation parameter means reducing the color opacity in the plot
# count plot on two categorical variable
a = sns.countplot(x="sex",data=df,hue="smoker",saturation=1) # set 1 for perfectly
plt.show()
```



In [59]:
```
# we use matplotlib.axes.axes.bar() to control the style
# count plot on two categorical variable
a = sns.countplot(x="sex",data=df,facecolor="white",linewidth=5,edgecolor=sns.color
plt.show()
```

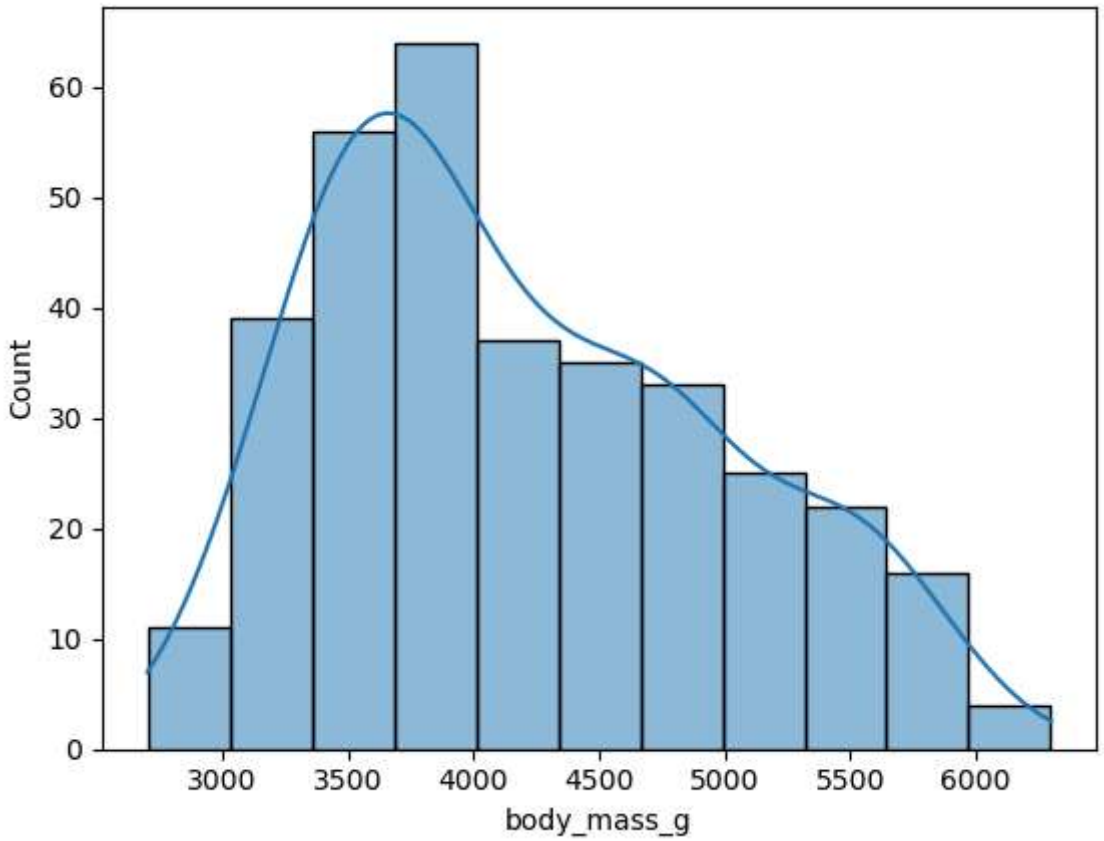


In [1]:
```
# now,we going to create the histogram plot

# generating the random numbers
np.random.seed(1) # basically initialized the staring number with 1
a = np.random.randn(1000) # up to range of 1000
a = pd.Series(a,name="Numerical Variable")

# plot the histogram
sns.histplot(data=a,kde=True) # kde stands for kernel density estimate for smooth d
plt.show()
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[1], line 4
      1 # now,we going to create the histogram plot
      2
      3 # generating the random numbers
----> 4 np.random.seed(1) # basically initialized the staring number with 1
      5 a = np.random.randn(1000) # up to range of 1000
      6 a = pd.Series(a,name="Numerical Variable")

NameError: name 'np' is not defined
```

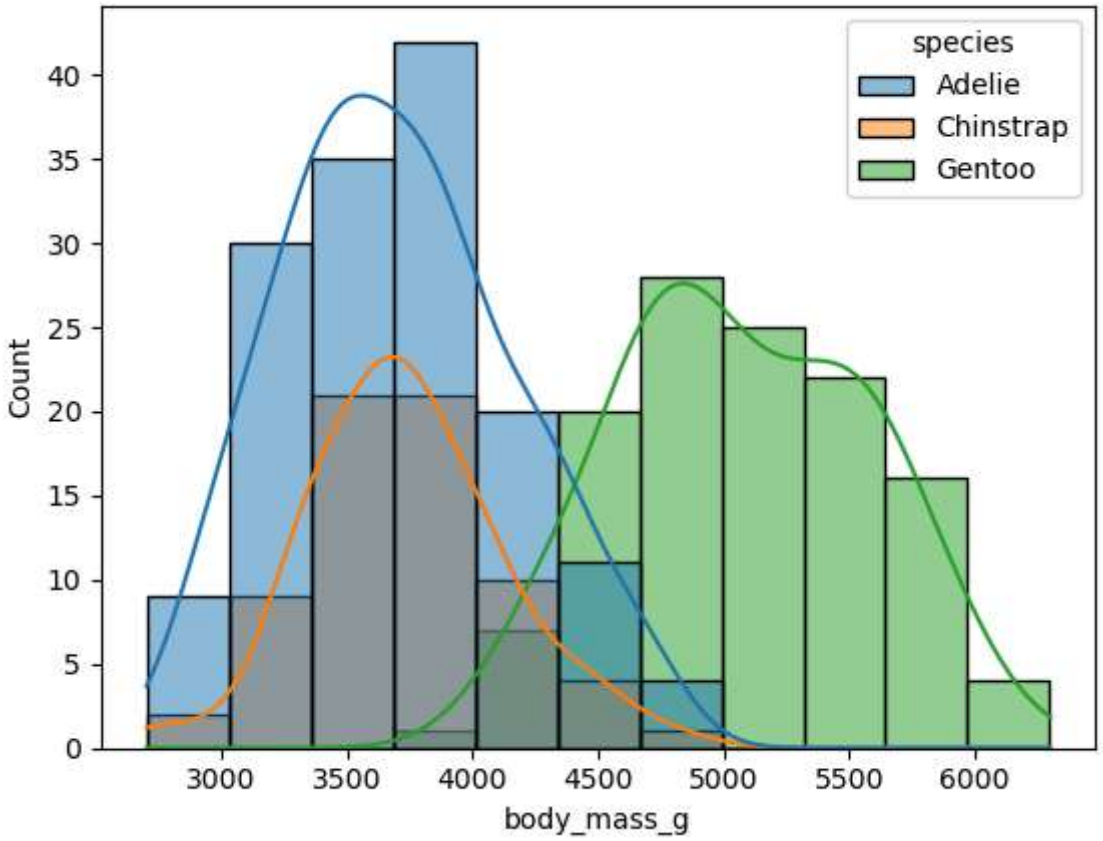
In [73]:
```
# let's work on the default dataset name is penguins
df = sns.load_dataset("penguins")
df
# plot the histogram
sns.histplot(x="body_mass_g",data=df,kde=True)
```

Out[73]: <Axes: xlabel='body_mass_g', ylabel='Count'>



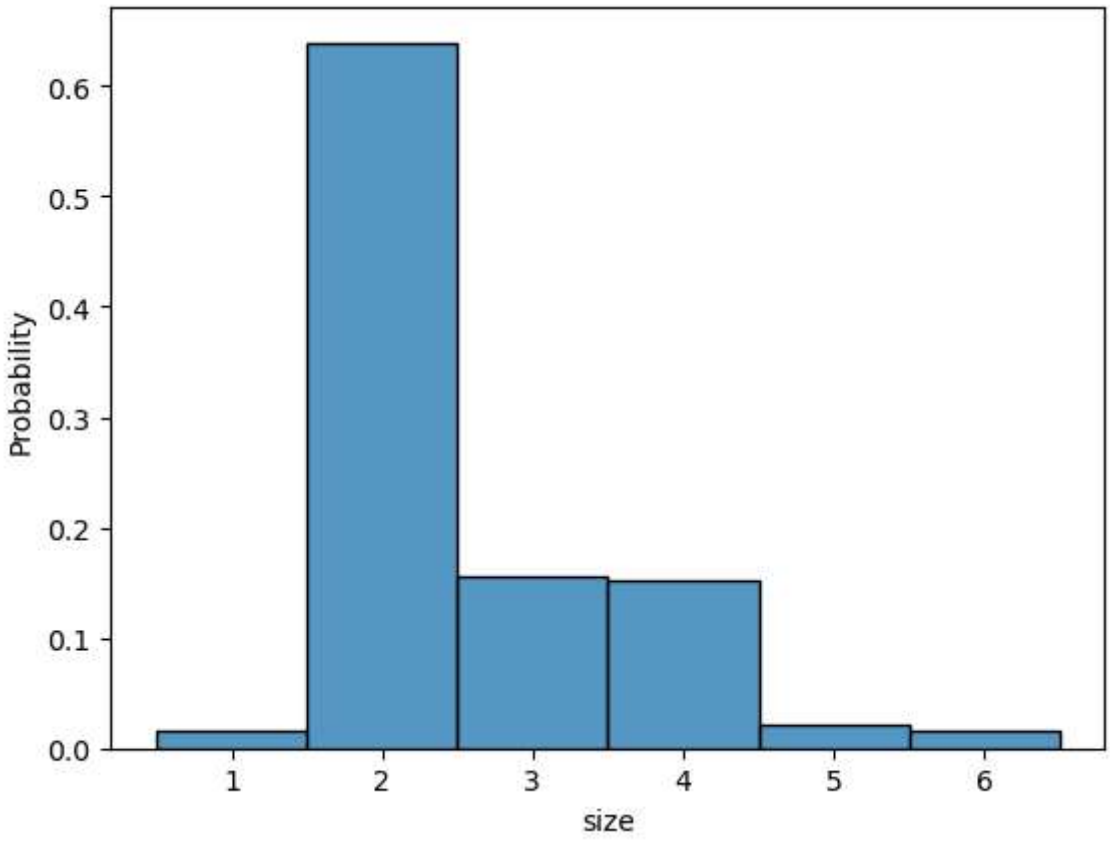
In [75]:
```
# with the help of hue we can visualize more
# let's work on the default dataset name is penguins
df = sns.load_dataset("penguins")
df
# plot the histogram
sns.histplot(x="body_mass_g",data=df,kde=True,hue="species")
```

Out[75]: <Axes: xlabel='body_mass_g', ylabel='Count'>



In [85]:
```
# let's calculate the probability by using stat
# let's work on the default dataset name is penguins
df = sns.load_dataset("tips")
df
# plot the histogram
sns.histplot(x="size",data=df,discrete=True,stat="probability")
```

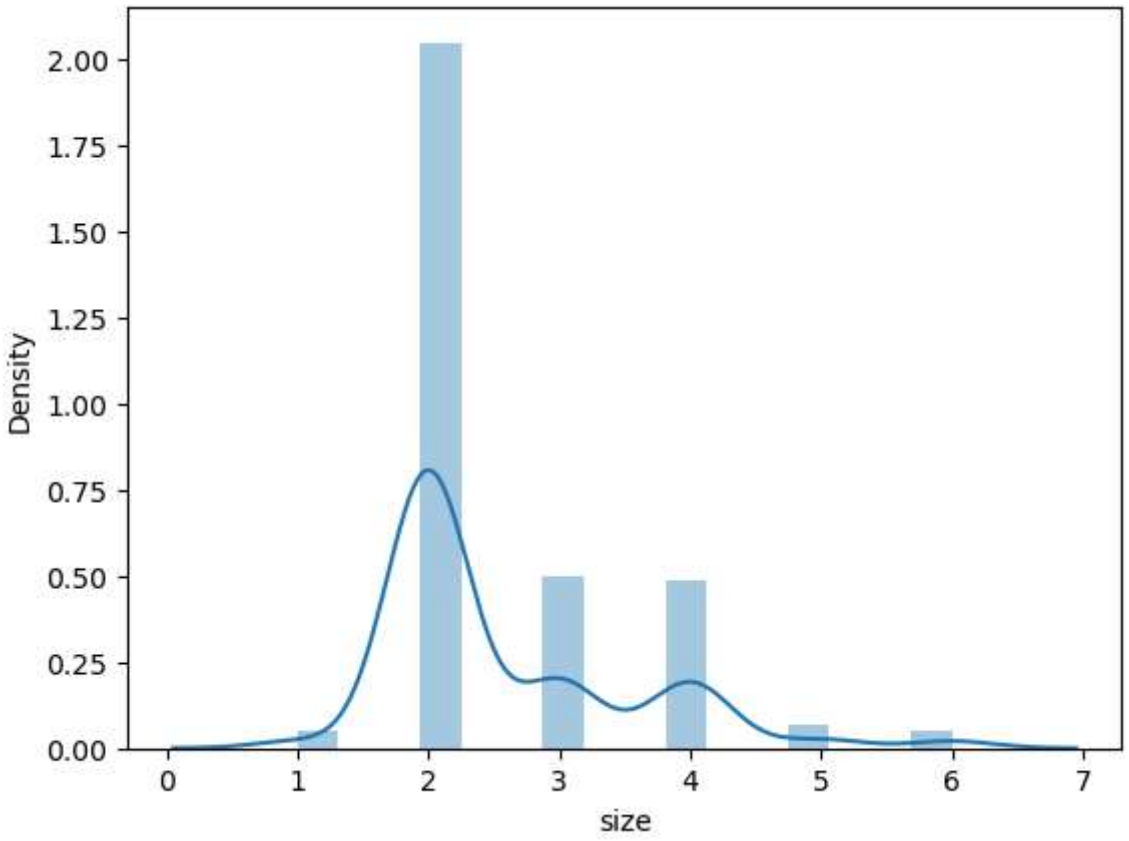
Out[85]: <Axes: xlabel='size', ylabel='Probability'>



In [105…
```
# how to add outline or edge color to histogram in seaborn
df = sns.load_dataset("tips")
df

# depict illustration
sns.distplot(df["size"])
```

```
C:\Users\fv8.DESKTOP-N5HA3AQ\AppData\Local\Temp\ipykernel_10068\3574982050.py:6: 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["size"])
```

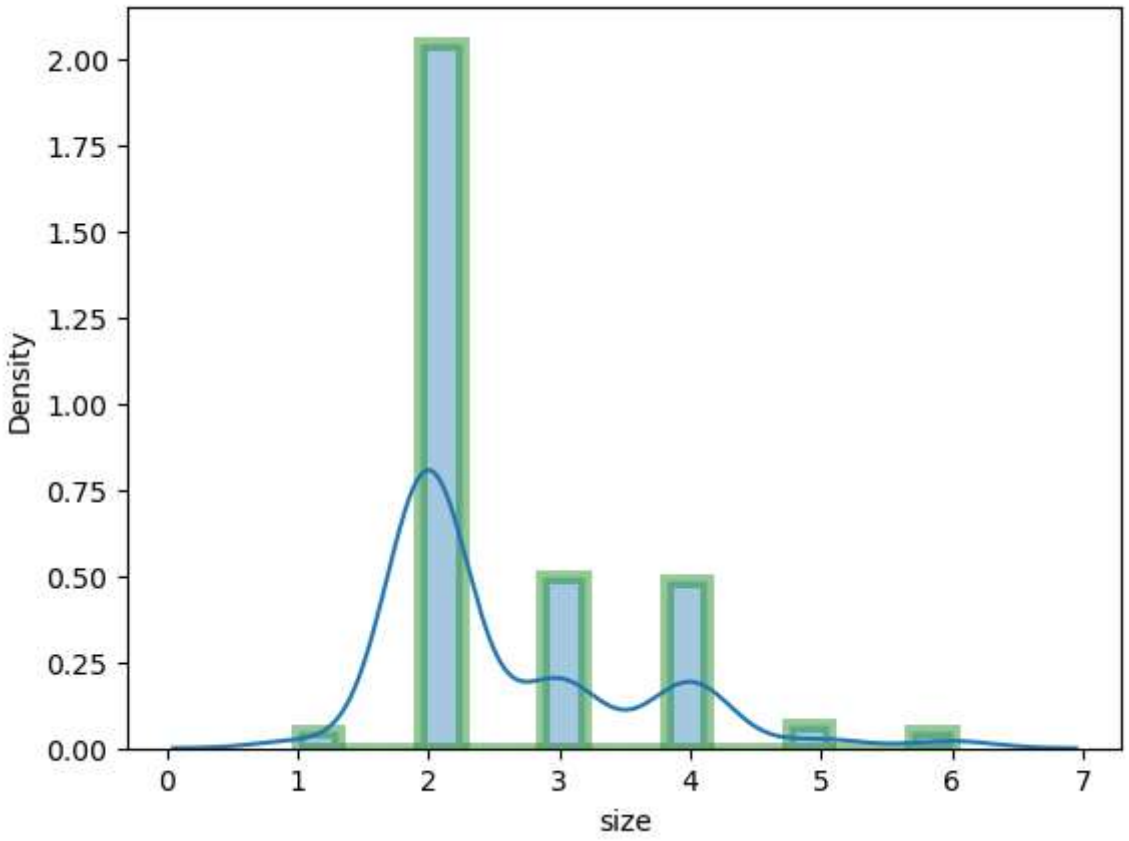
Out[105… <Axes: xlabel='size', ylabel='Density'>



In [107…
```
# we can give histogram keywords for adding an edge color to it
# depict illustration
sns.distplot(df["size"],hist_kws=dict(edgecolor="green",linewidth=5))
```

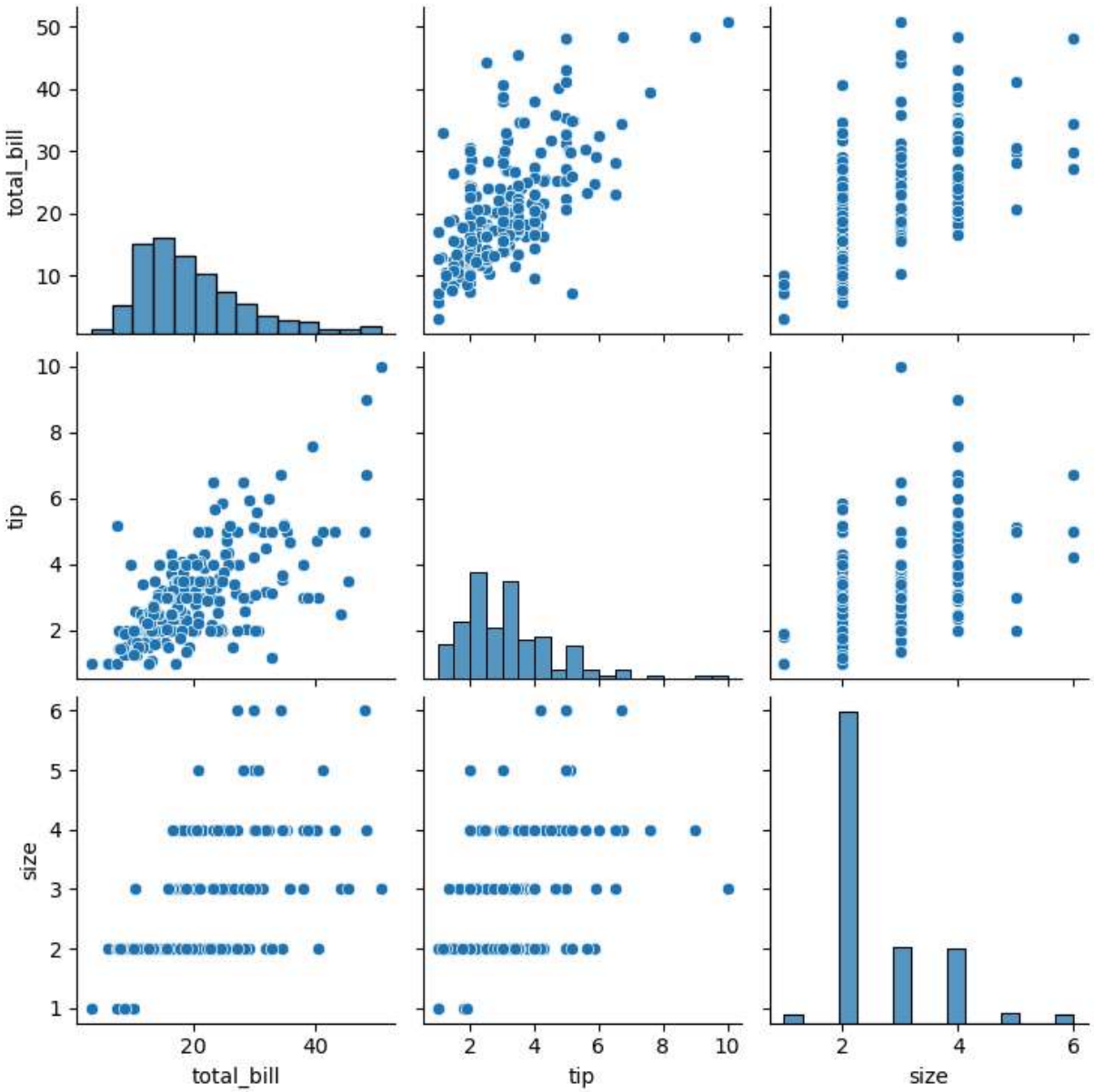
```
C:\Users\fv8.DESKTOP-N5HA3AQ\AppData\Local\Temp\ipykernel_10068\3237224193.py:3: UserWarning:

`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["size"],hist_kws=dict(edgecolor="green",linewidth=5))
```

Out[107…
```
<Axes: xlabel='size', ylabel='Density'>
```



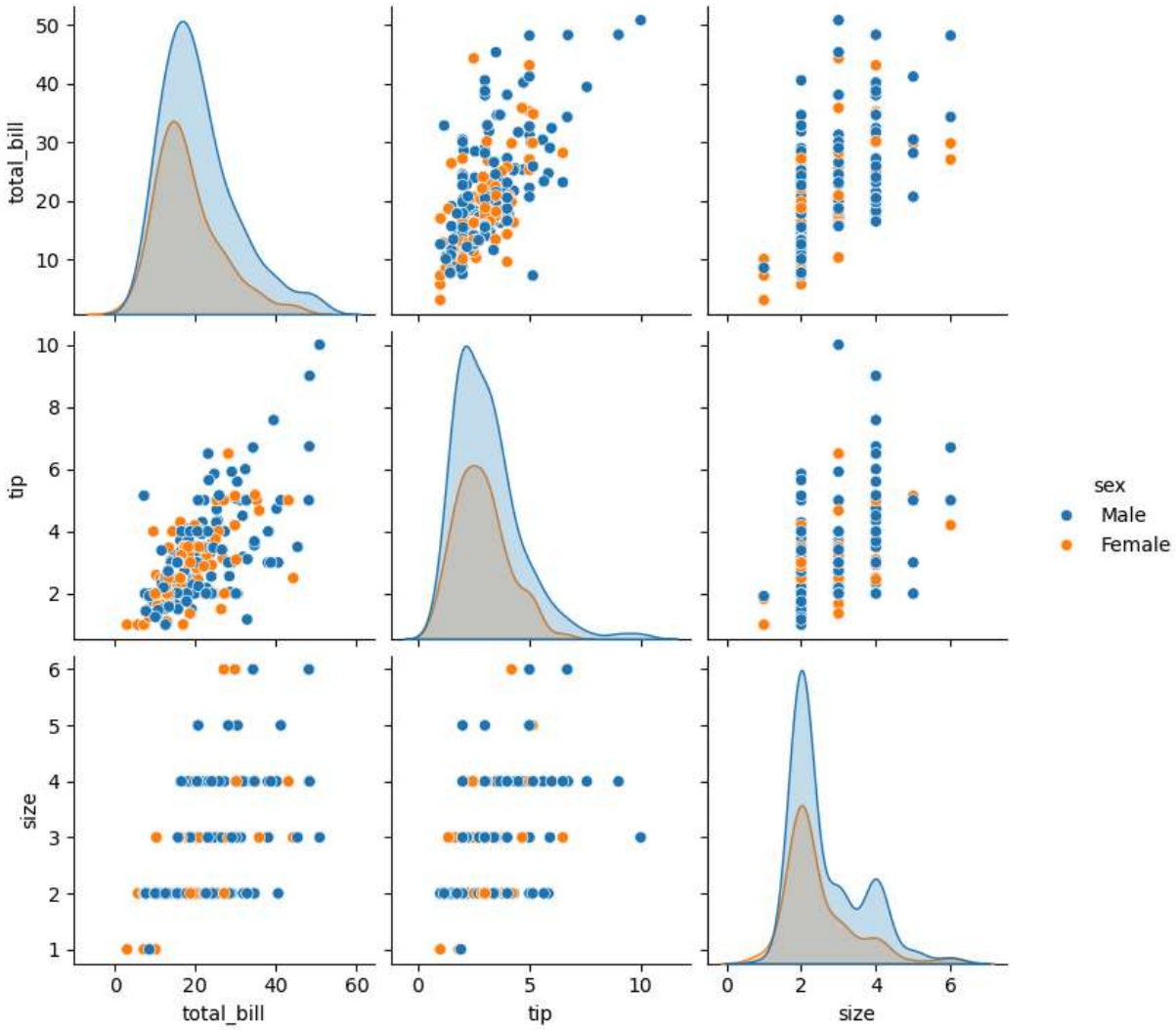
In [123…

```
# let's create the pair plot
df= sns.load_dataset("tips")
sns.pairplot(df)
plt.show()
```



In [121…

```
# let's create the pair plot
df= sns.load_dataset("tips")
sns.pairplot(df,hue="sex")
plt.show()
```


total_bill
tip
size
sex
Male
Female
0
20
40
60
0
5
10
0
2
4
6
1
2
3
4
5
6
2
4
6
8
10
10
20
30
40
50

In [ ]: