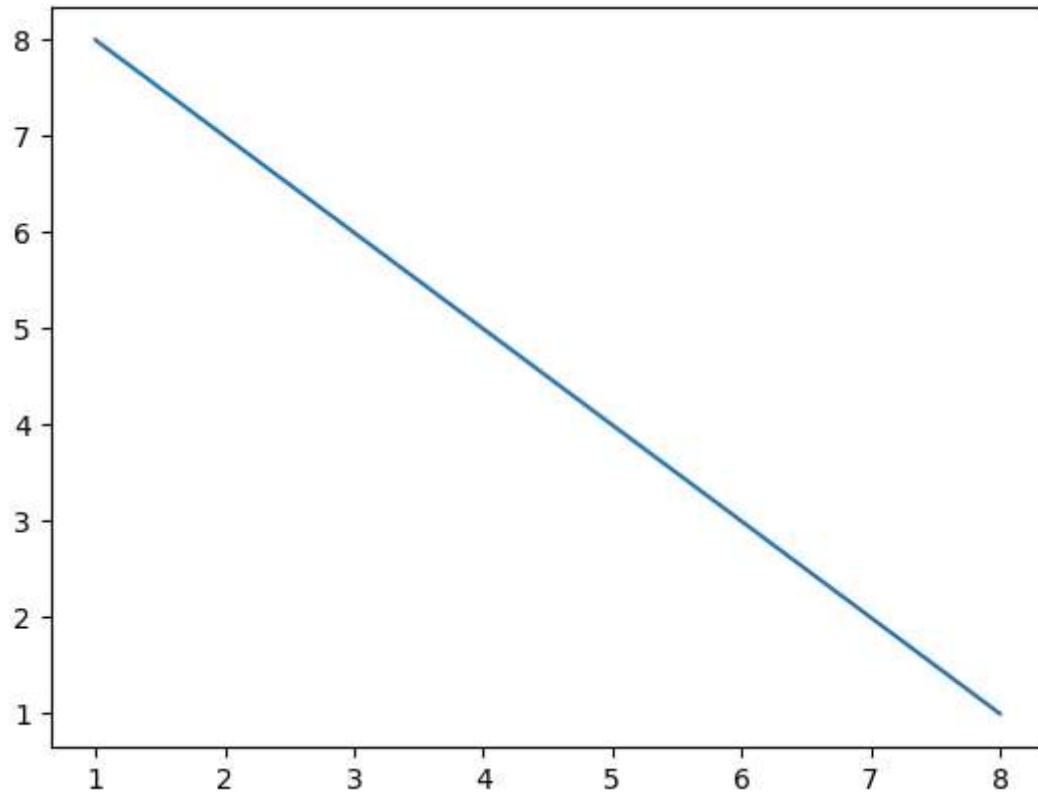


## Line Plot

```
In [2]: import seaborn as sns  
import matplotlib.pyplot as plt  
  
var = [1,2,3,4,5,6,7,8]  
var1 = [8,7,6,5,4,3,2,1]
```

matplotlib se bna hai niche wala graph

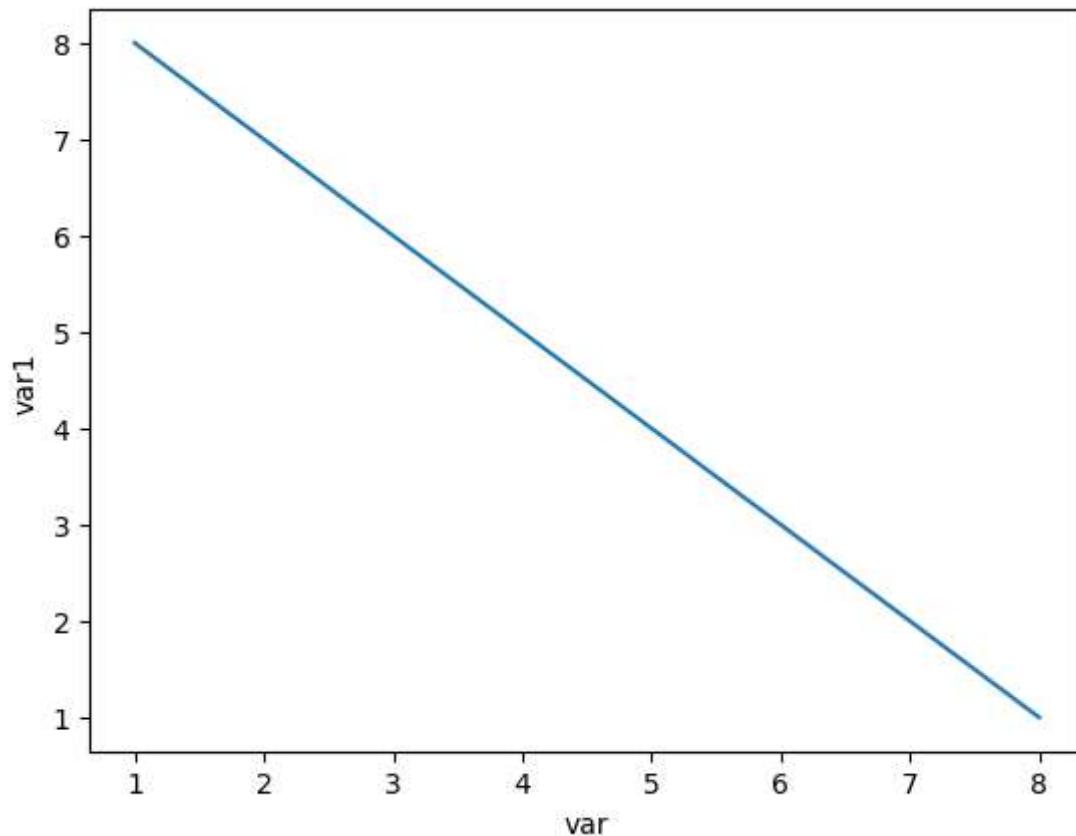
```
In [3]: plt.plot(var,var1)  
plt.show()
```



seaborn se bna hai niche wala graph

```
In [4]: var = [1,2,3,4,5,6,7,8]
var1 = [8,7,6,5,4,3,2,1]
import pandas as pd

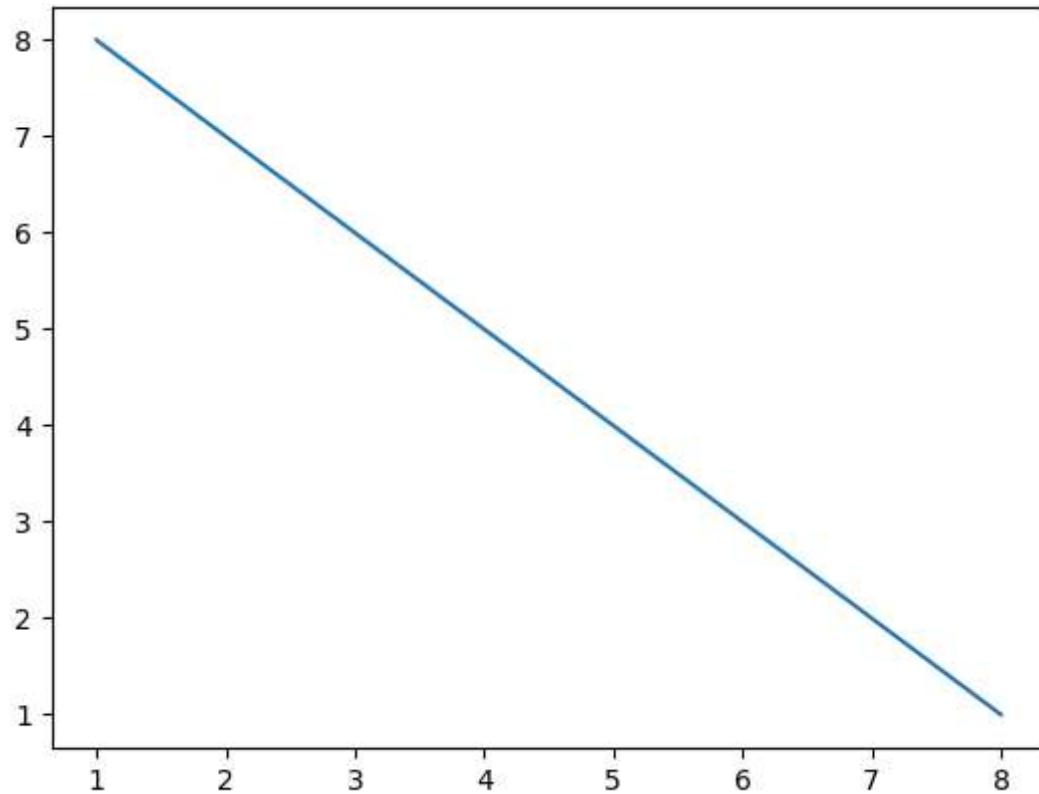
df = pd.DataFrame({"var":var, "var1":var1})
sns.lineplot(x='var', y='var1', data=df)
plt.show()
```



seaborn se bna hia niche wala graph bina data frame k

```
In [5]: var = [1,2,3,4,5,6,7,8]
var1 = [8,7,6,5,4,3,2,1]

sns.lineplot(x=var, y=var1)
plt.show()
```

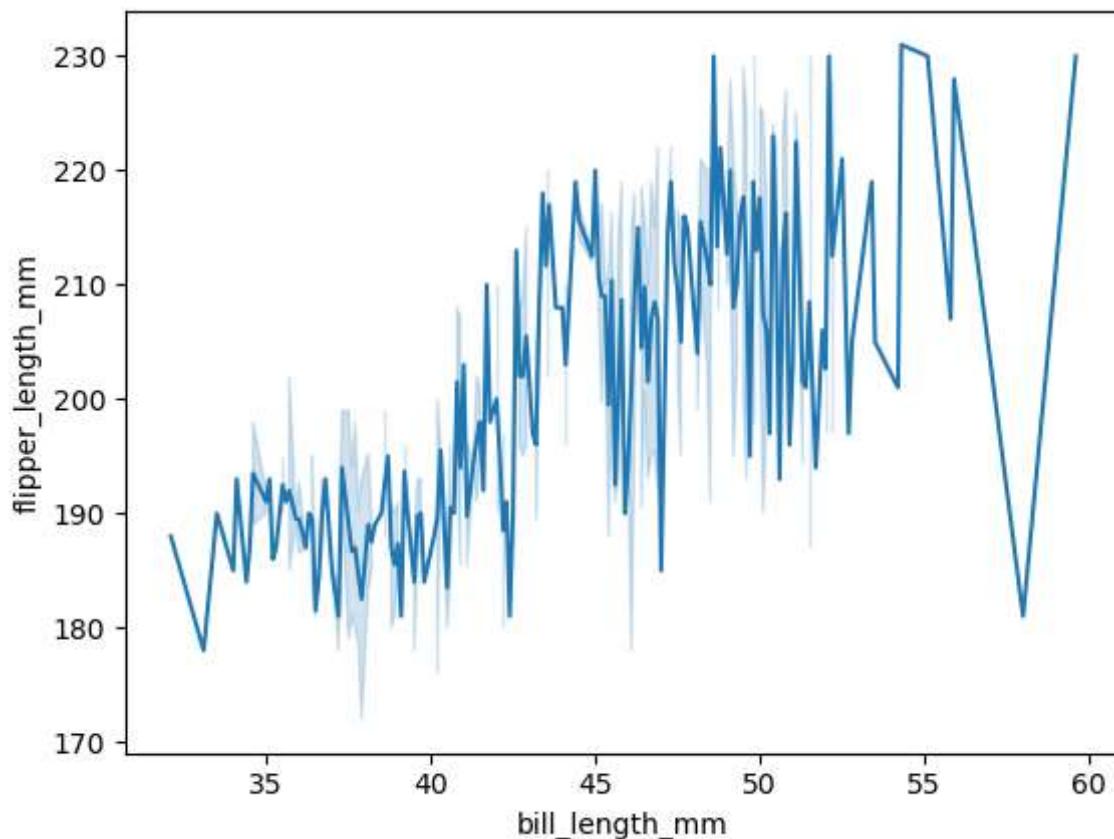


```
In [6]: df1 = sns.load_dataset("penguins")
```

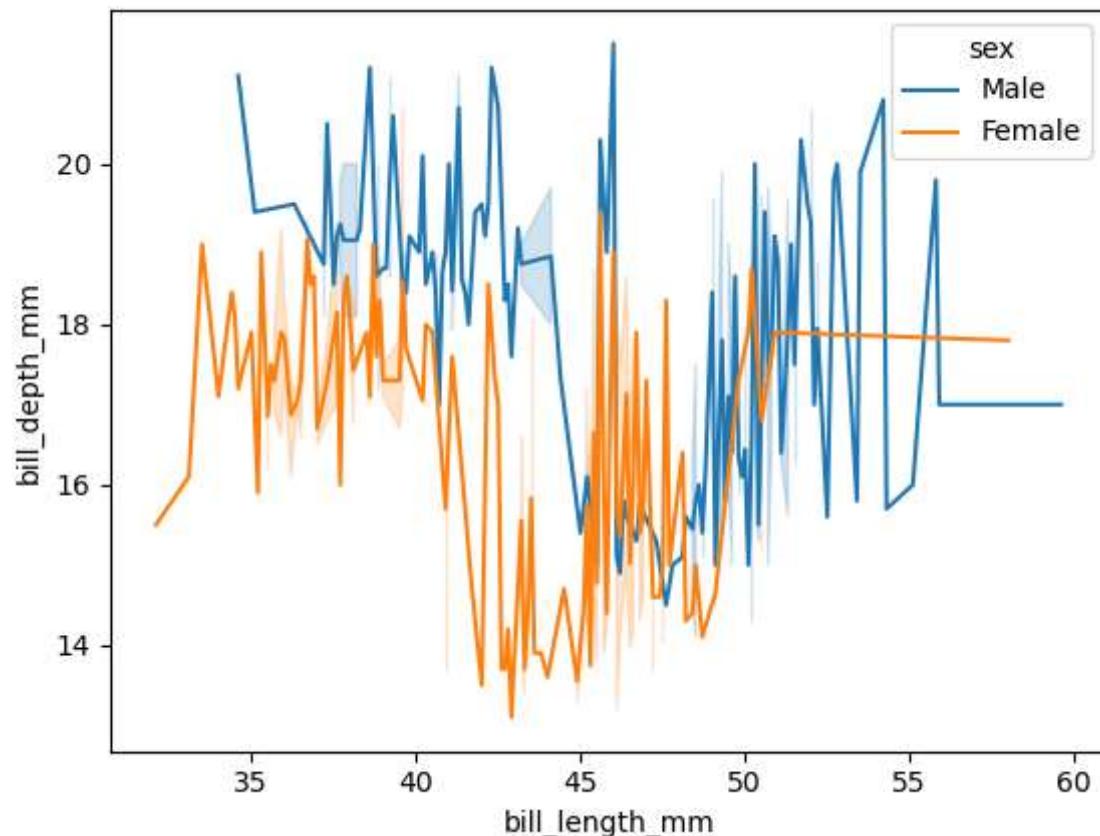
In [7]: df1

	species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	sex
0	Adelie	Torgersen	39.1	18.7	181.0	3750.0	Male
1	Adelie	Torgersen	39.5	17.4	186.0	3800.0	Female
2	Adelie	Torgersen	40.3	18.0	195.0	3250.0	Female
3	Adelie	Torgersen	NaN	NaN	NaN	NaN	NaN
4	Adelie	Torgersen	36.7	19.3	193.0	3450.0	Female
...	...	...	...	...	...	...	...
339	Gentoo	Biscoe	NaN	NaN	NaN	NaN	NaN
340	Gentoo	Biscoe	46.8	14.3	215.0	4850.0	Female
341	Gentoo	Biscoe	50.4	15.7	222.0	5750.0	Male
342	Gentoo	Biscoe	45.2	14.8	212.0	5200.0	Female
343	Gentoo	Biscoe	49.9	16.1	213.0	5400.0	Male

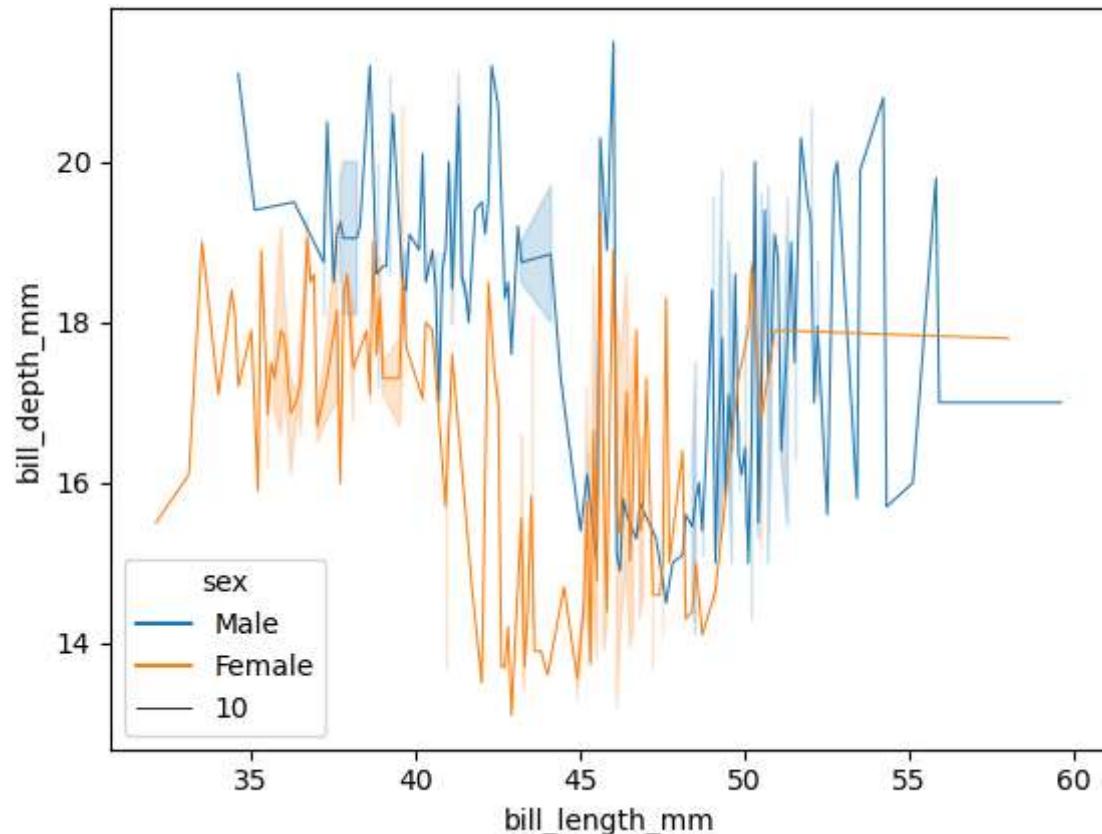
344 rows × 7 columns

In [8]: sns.lineplot(x="bill\_length\_mm", y="flipper\_length\_mm", data = df1)  
plt.show()

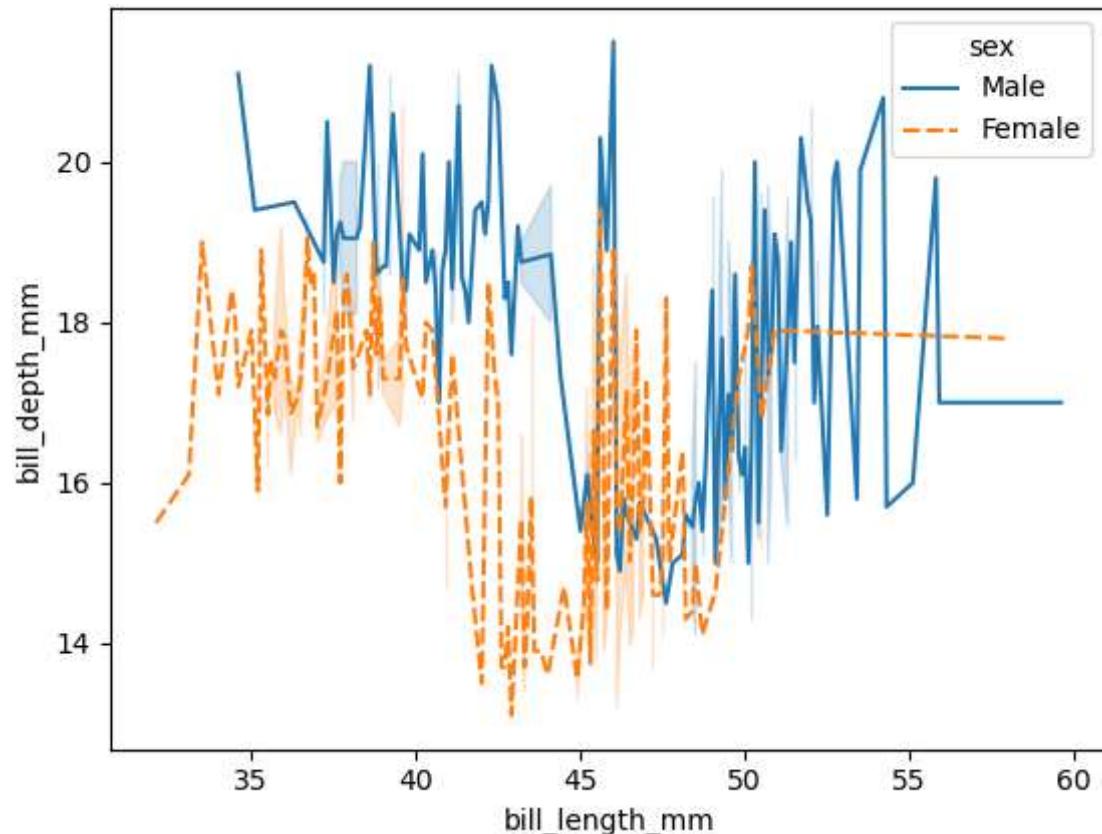
```
In [9]: sns.lineplot(x="bill_length_mm", y="bill_depth_mm", data = df1, hue="sex")
plt.show()
```



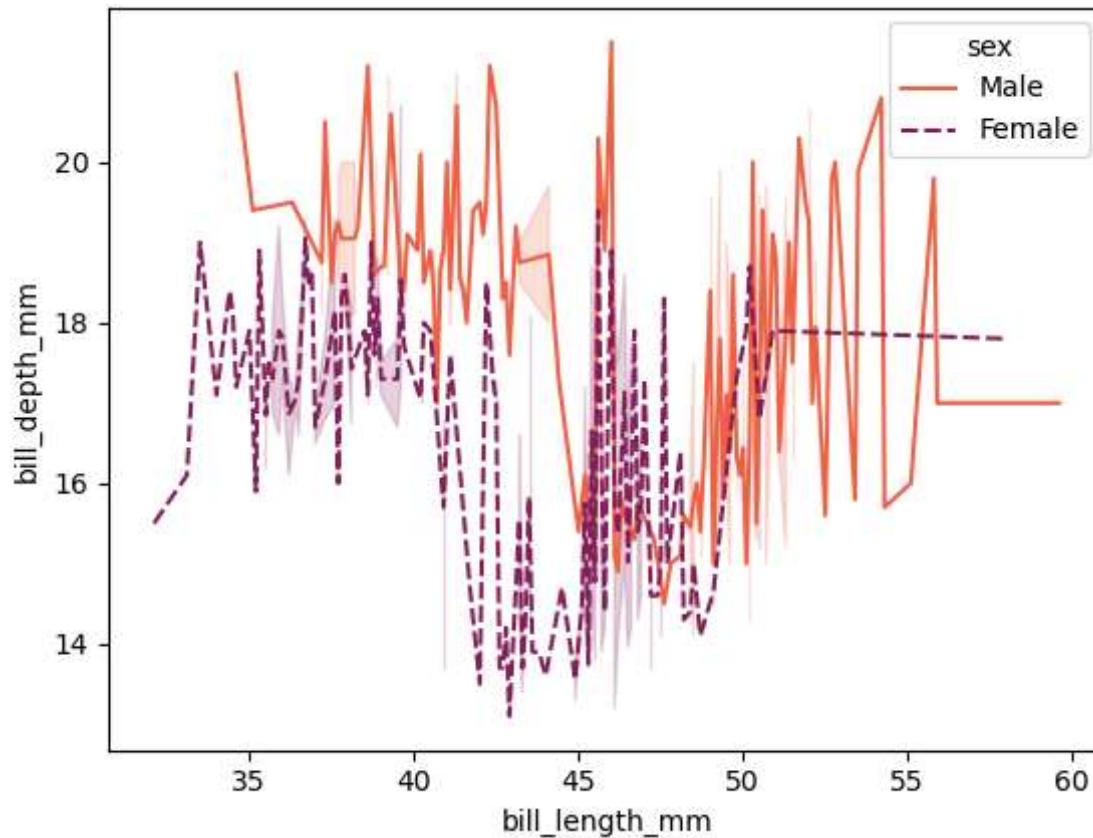
```
In [10]: sns.lineplot(x="bill_length_mm", y="bill_depth_mm", data = df1, hue="sex", size=10)
plt.show()
```



```
In [11]: sns.lineplot(x="bill_length_mm", y="bill_depth_mm", data = df1, hue="sex", style= "dashed")  
plt.show()
```

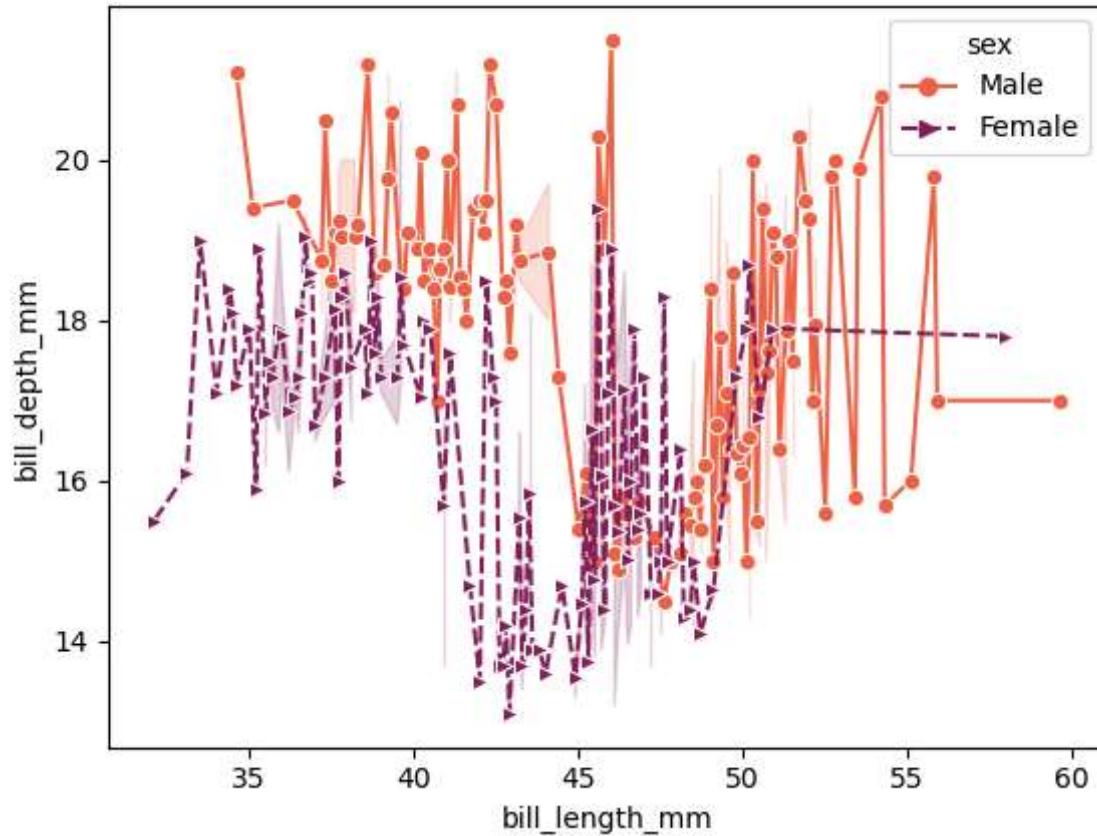


```
In [14]: sns.lineplot(x="bill_length_mm", y="bill_depth_mm", data = df1, hue="sex",
                     style= "sex", palette="rocket_r")
plt.show()
```



```
In [15]: sns.lineplot(x="bill_length_mm", y="bill_depth_mm", data = df1, hue="sex",
                     style= "sex", palette="rocket_r", markers = ["o", ">"])

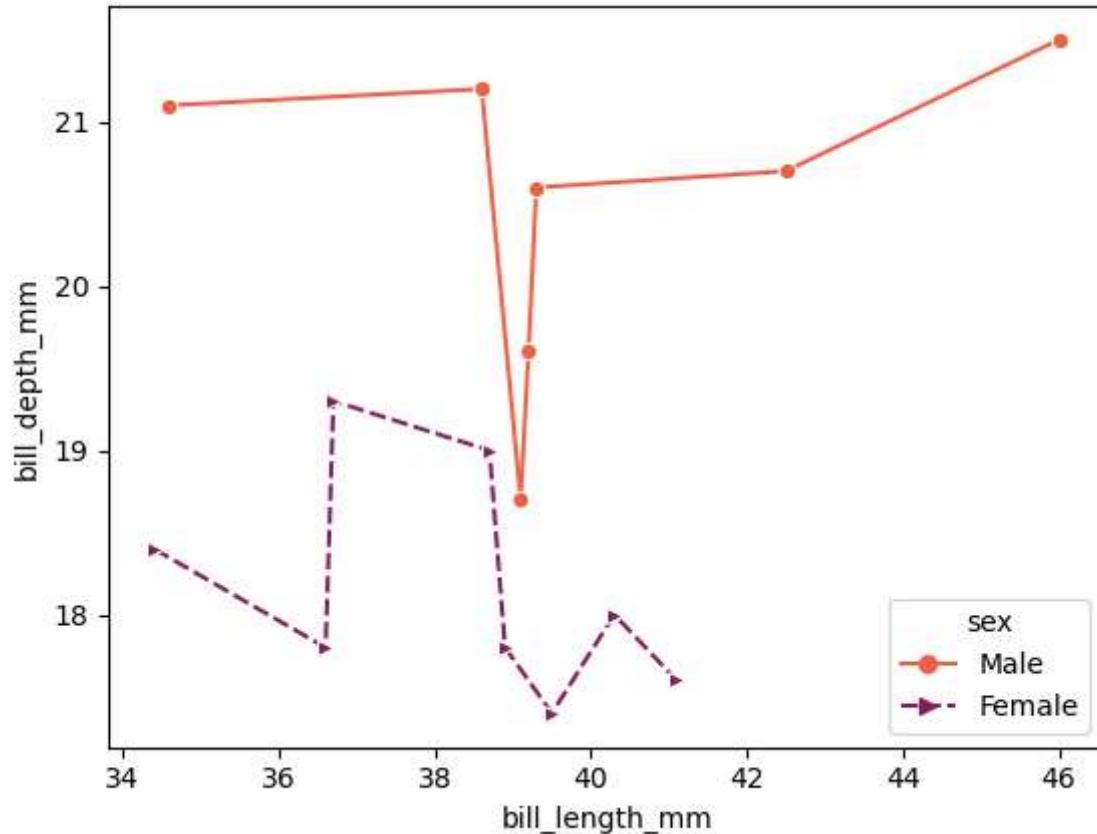
plt.show()
```



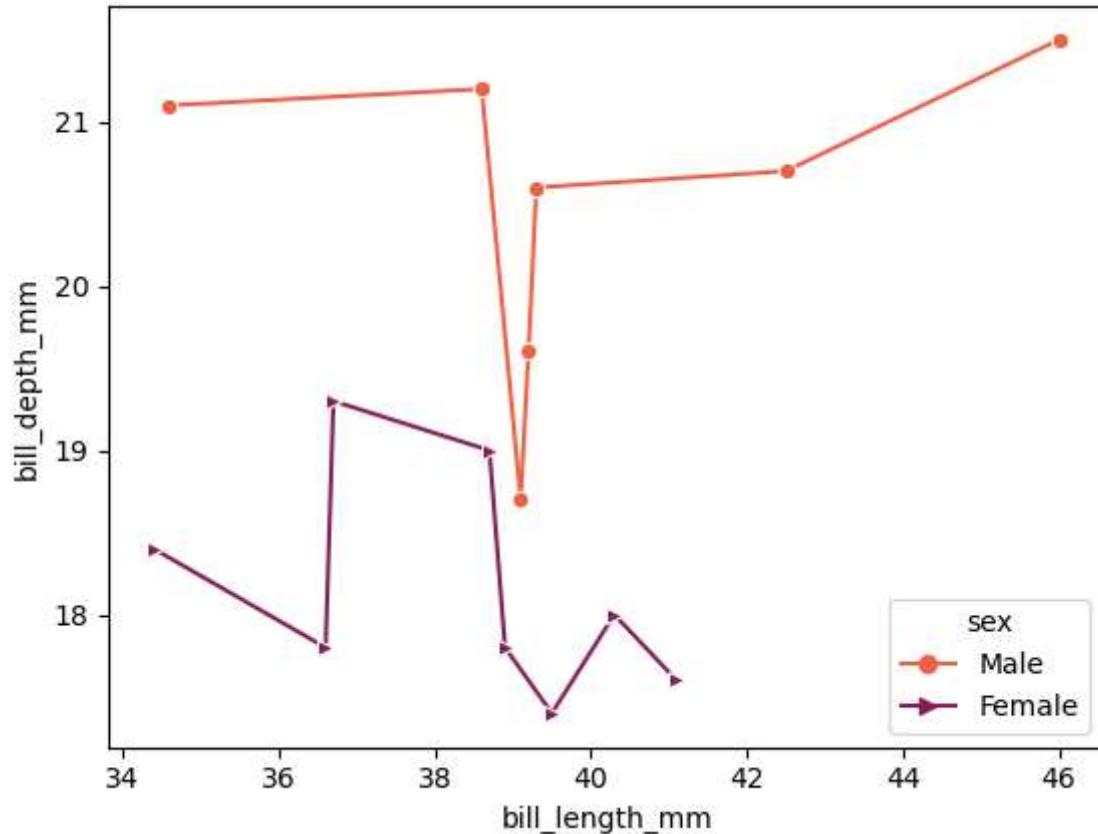
dataframe me dot head laga k dikhana hai 20 data k liye graphs ko

```
In [21]: df1 = sns.load_dataset("penguins").head(20)
```

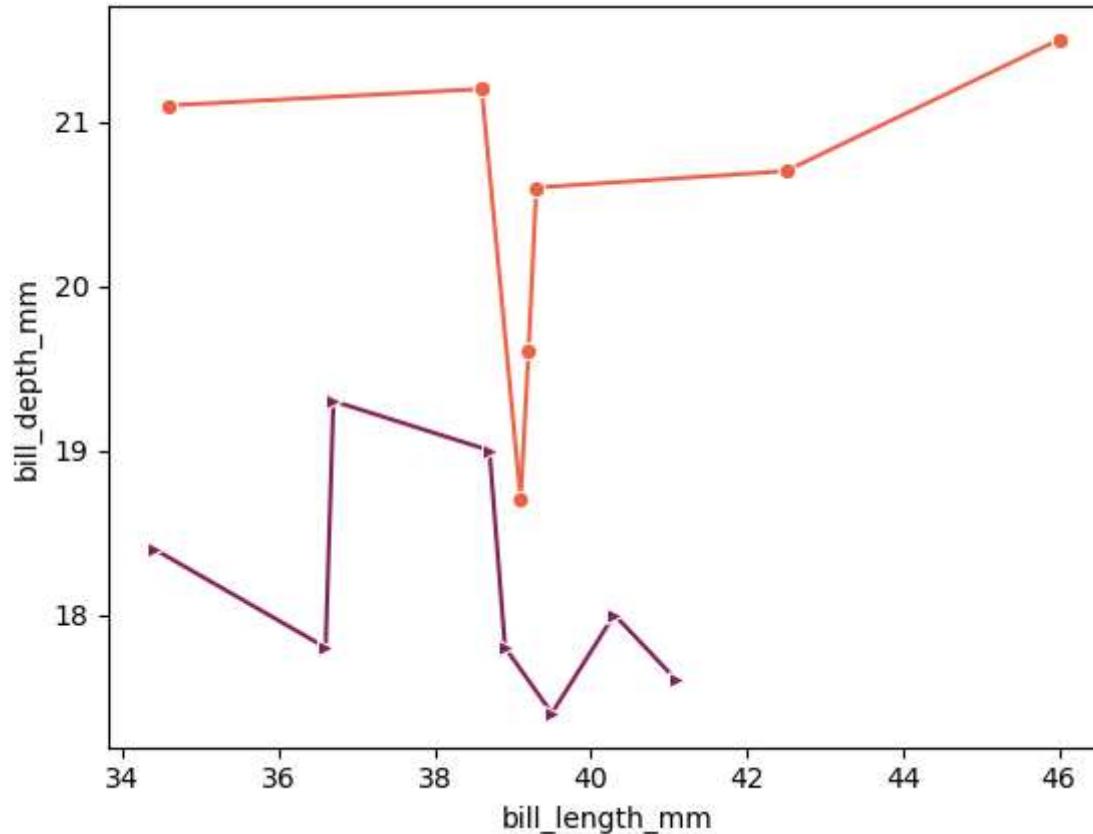
```
In [22]: sns.lineplot(x="bill_length_mm", y="bill_depth_mm", data = df1, hue="sex",
                     style= "sex", palette="rocket_r", markers = ["o", ">"], dashes = True)
plt.show()
```



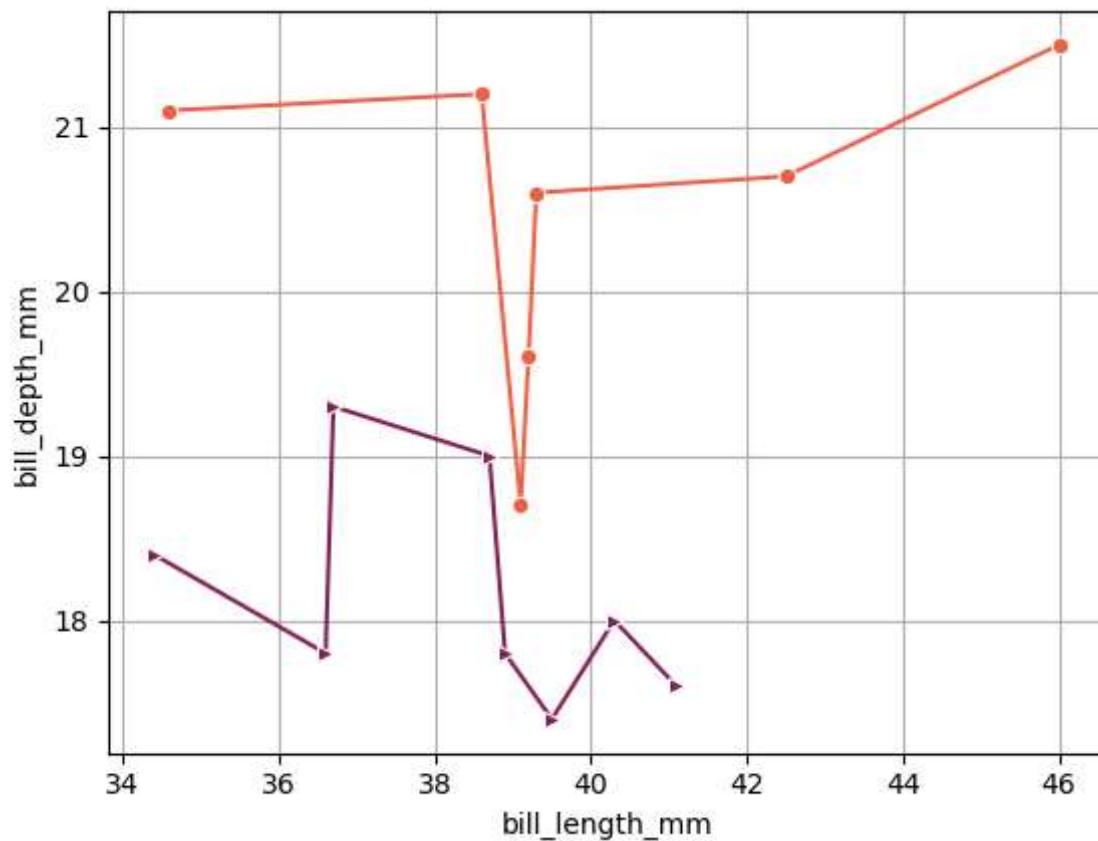
```
In [23]: sns.lineplot(x="bill_length_mm", y="bill_depth_mm", data = df1, hue="sex",
                     style= "sex", palette="rocket_r", markers = ["o", ">"], dashes = False)
plt.show()
```



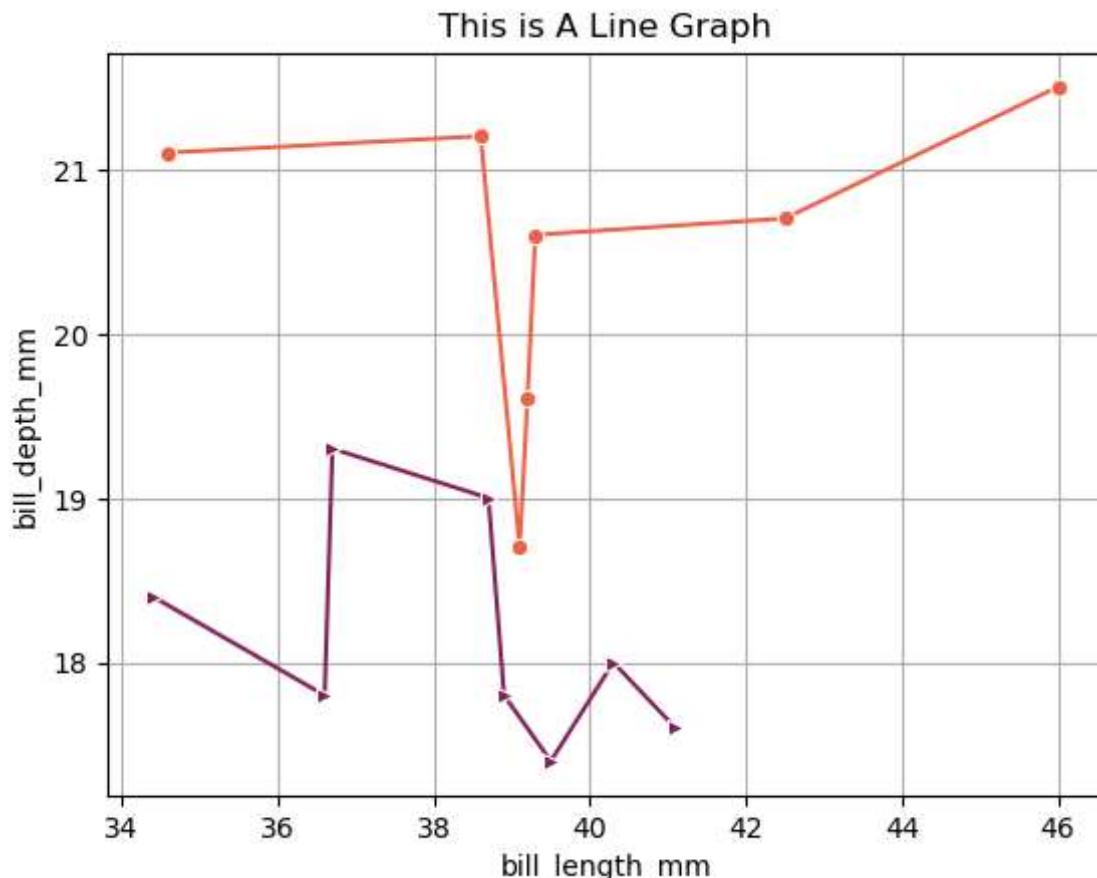
```
In [24]: sns.lineplot(x="bill_length_mm", y="bill_depth_mm", data = df1, hue="sex",
                     style= "sex", palette="rocket_r", markers = ["o", ">"], dashes = False)
plt.show()
```



```
In [25]: sns.lineplot(x="bill_length_mm", y="bill_depth_mm", data = df1, hue="sex",
                     style= "sex", palette="rocket_r", markers = ["o", ">"], dashes = False)
plt.grid()
plt.show()
```



```
In [26]: sns.lineplot(x="bill_length_mm", y="bill_depth_mm", data = df1, hue="sex",
                     style= "sex", palette="rocket_r", markers = ["o", ">"], dashes = False)
plt.grid()
plt.title("This is A Line Graph")
plt.show()
```



## Bar Plot in Seaborn

```
In [27]: import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd
```

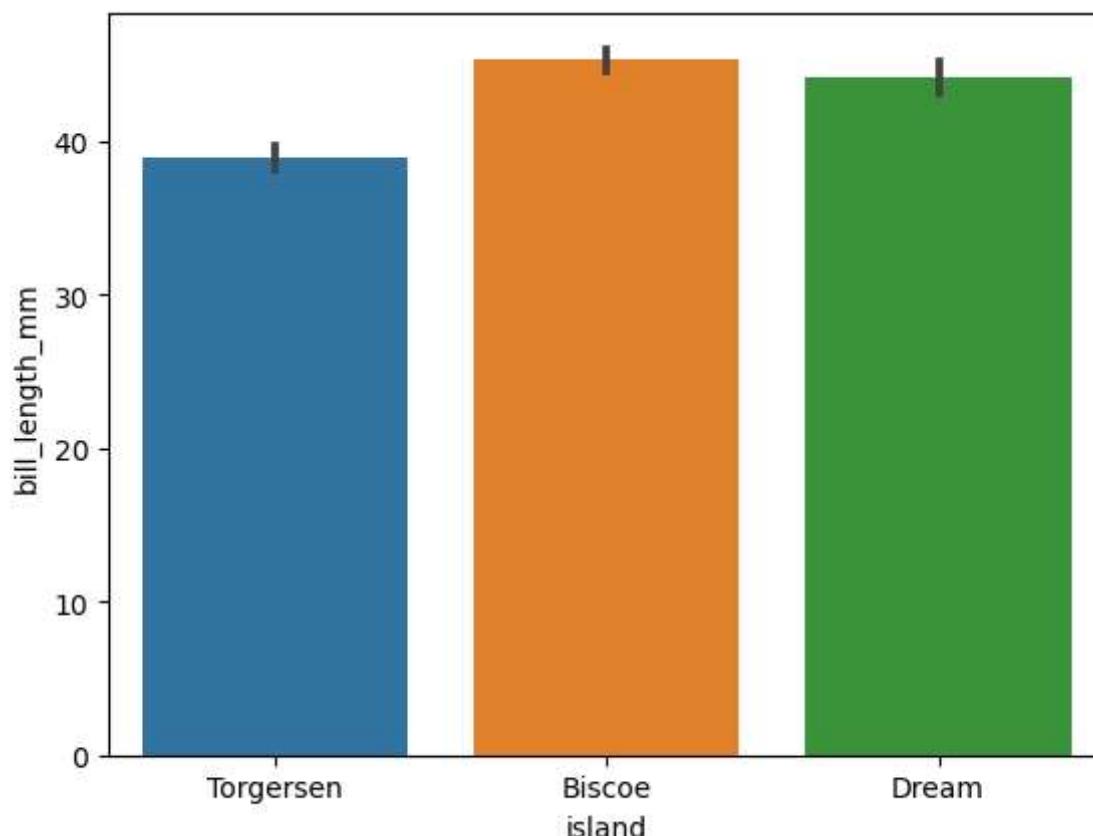
```
In [28]: df1 = sns.load_dataset("penguins")
```

In [37]: df1

Out[37]:

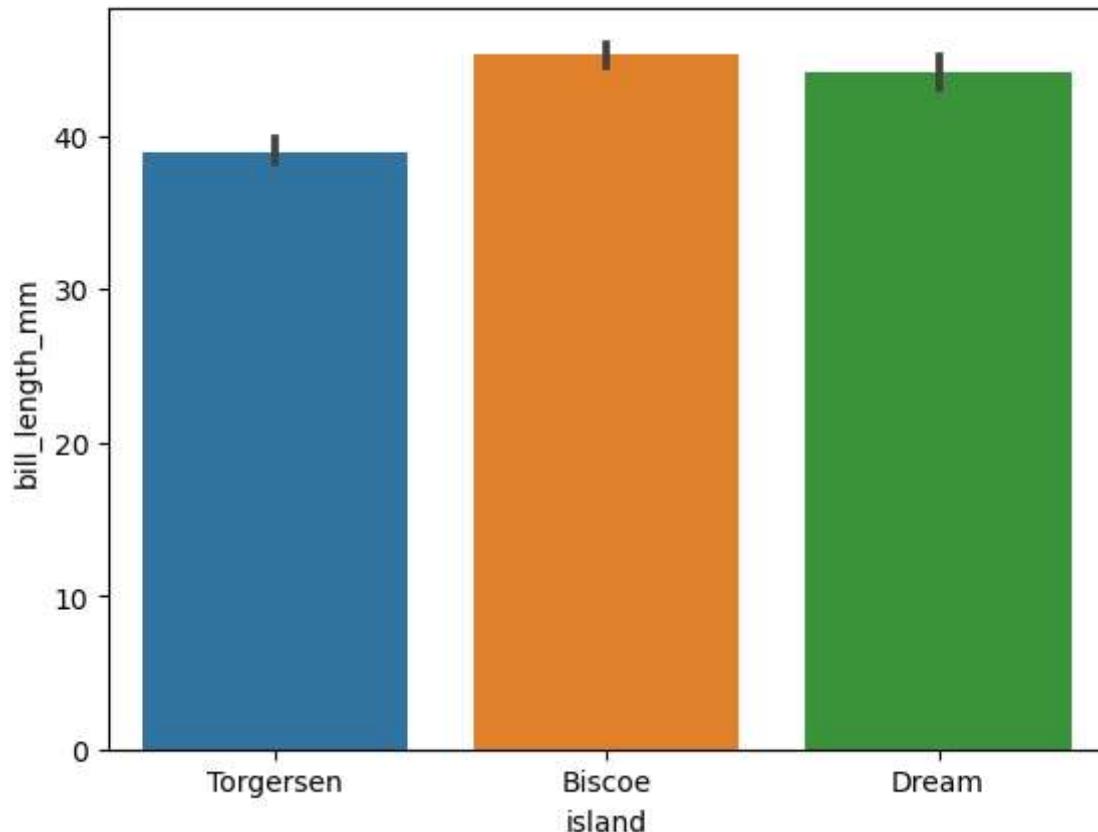
	species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	sex
0	Adelie	Torgersen	39.1	18.7	181.0	3750.0	Male
1	Adelie	Torgersen	39.5	17.4	186.0	3800.0	Female
2	Adelie	Torgersen	40.3	18.0	195.0	3250.0	Female
3	Adelie	Torgersen	NaN	NaN	NaN	NaN	NaN
4	Adelie	Torgersen	36.7	19.3	193.0	3450.0	Female
...	...	...	...	...	...	...	...
339	Gentoo	Biscoe	NaN	NaN	NaN	NaN	NaN
340	Gentoo	Biscoe	46.8	14.3	215.0	4850.0	Female
341	Gentoo	Biscoe	50.4	15.7	222.0	5750.0	Male
342	Gentoo	Biscoe	45.2	14.8	212.0	5200.0	Female
343	Gentoo	Biscoe	49.9	16.1	213.0	5400.0	Male

344 rows × 7 columns

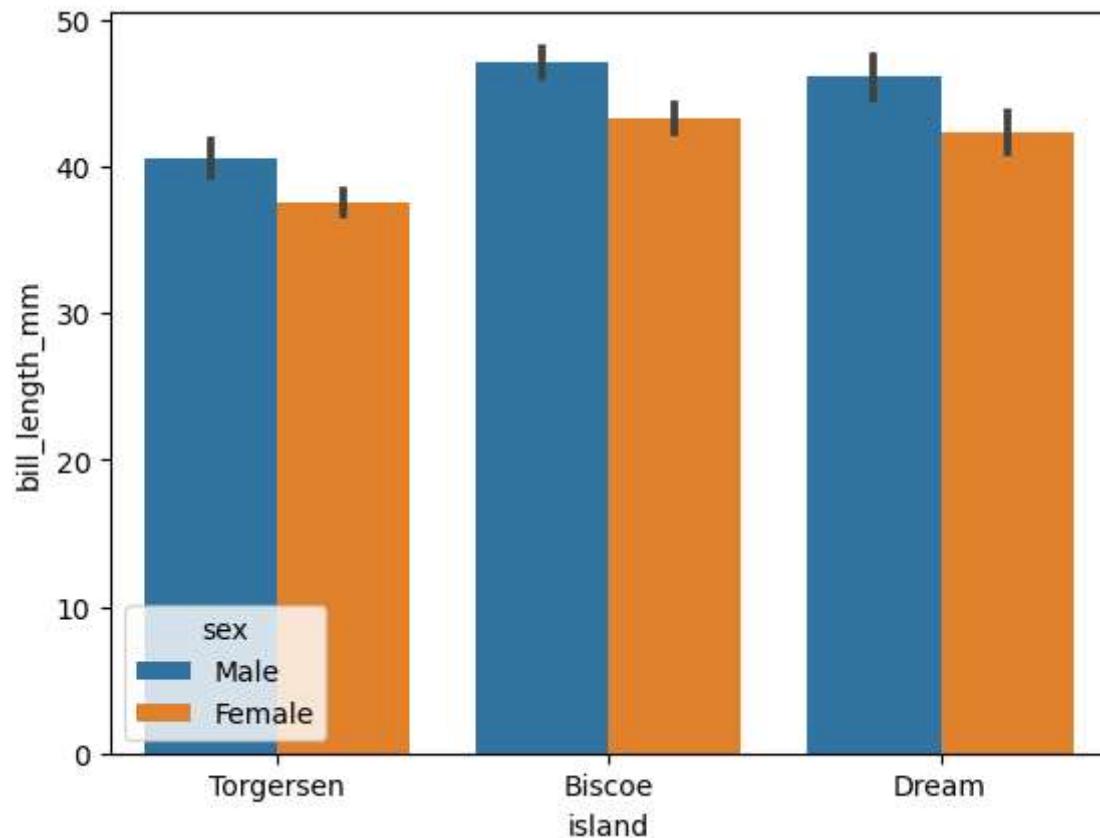
In [31]: sns.barplot(x= df1.island, y = df1.bill\_length\_mm)  
plt.show()

bar graph bnane ka dusra tarika

```
In [32]: sns.barplot(x="island", y ="bill_length_mm", data=df1)  
plt.show()
```

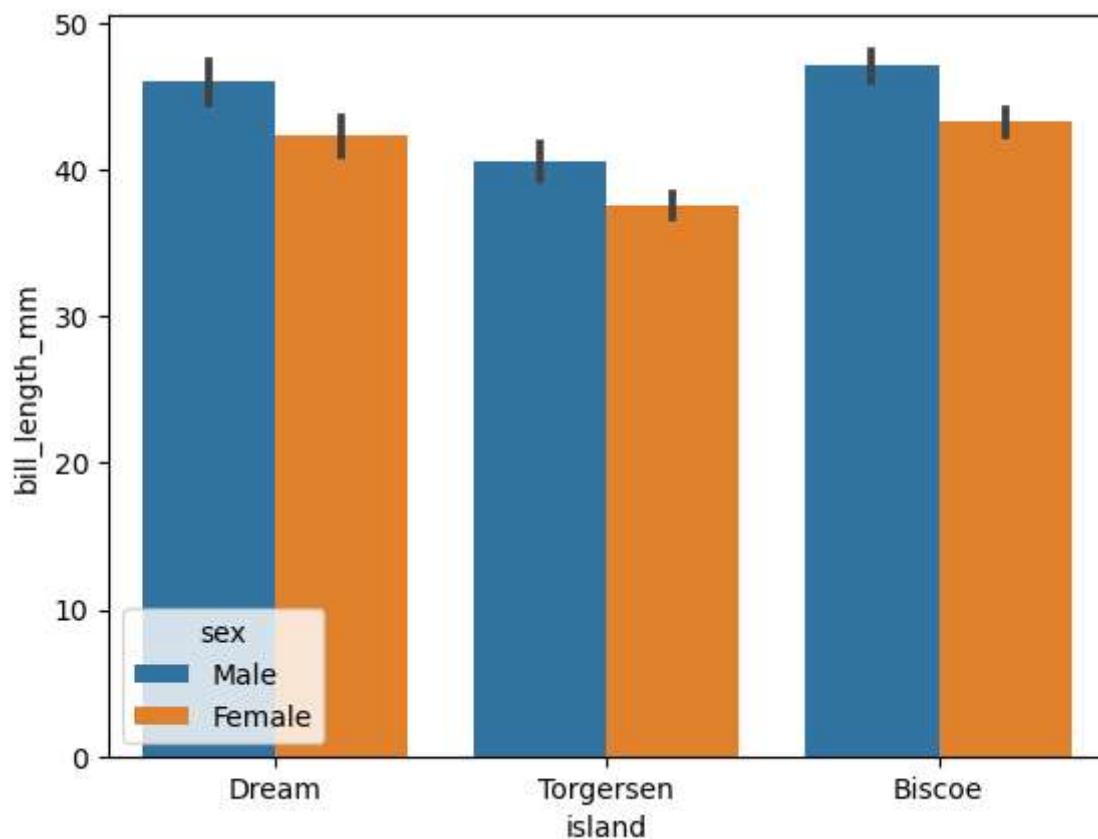


```
In [33]: sns.barplot(x="island", y ="bill_length_mm", data=df1, hue="sex")
plt.show()
```



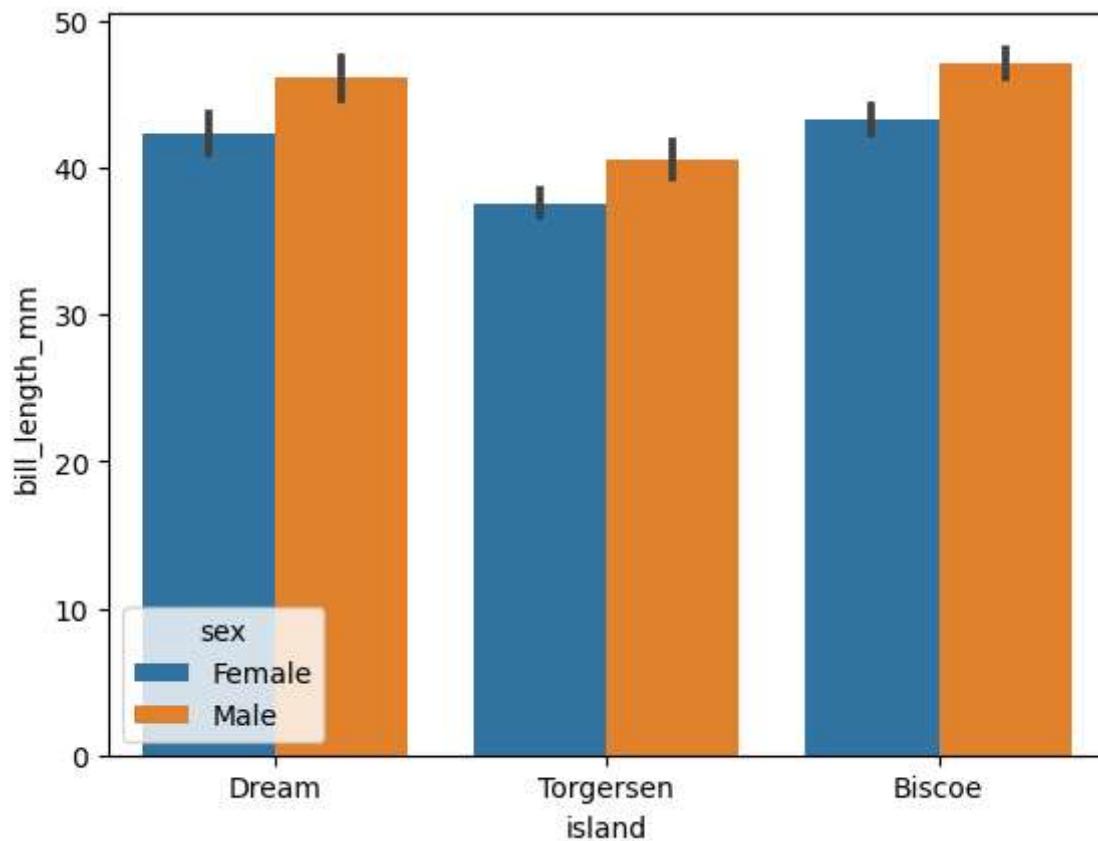
changing the order of the name of the island

```
In [38]: order1 = ["Dream", "Torgersen", "Biscoe"]
sns.barplot(x="island", y ="bill_length_mm", data=df1, hue="sex", order = order1)
plt.show()
```



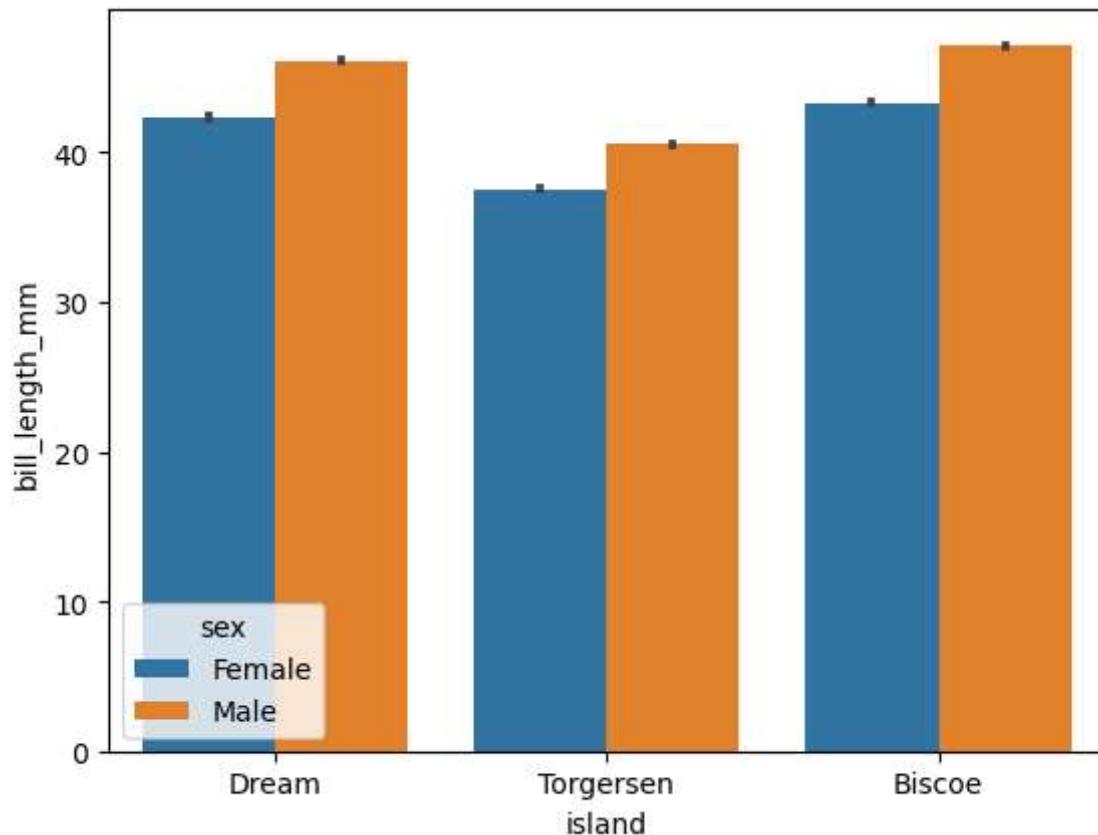
changing the order of the hue

```
In [39]: order1 = ["Dream", "Torgersen", "Biscoe"]
sns.barplot(x="island", y ="bill_length_mm", data=df1, hue="sex",
            order = order1, hue_order=['Female','Male'])
plt.show()
```

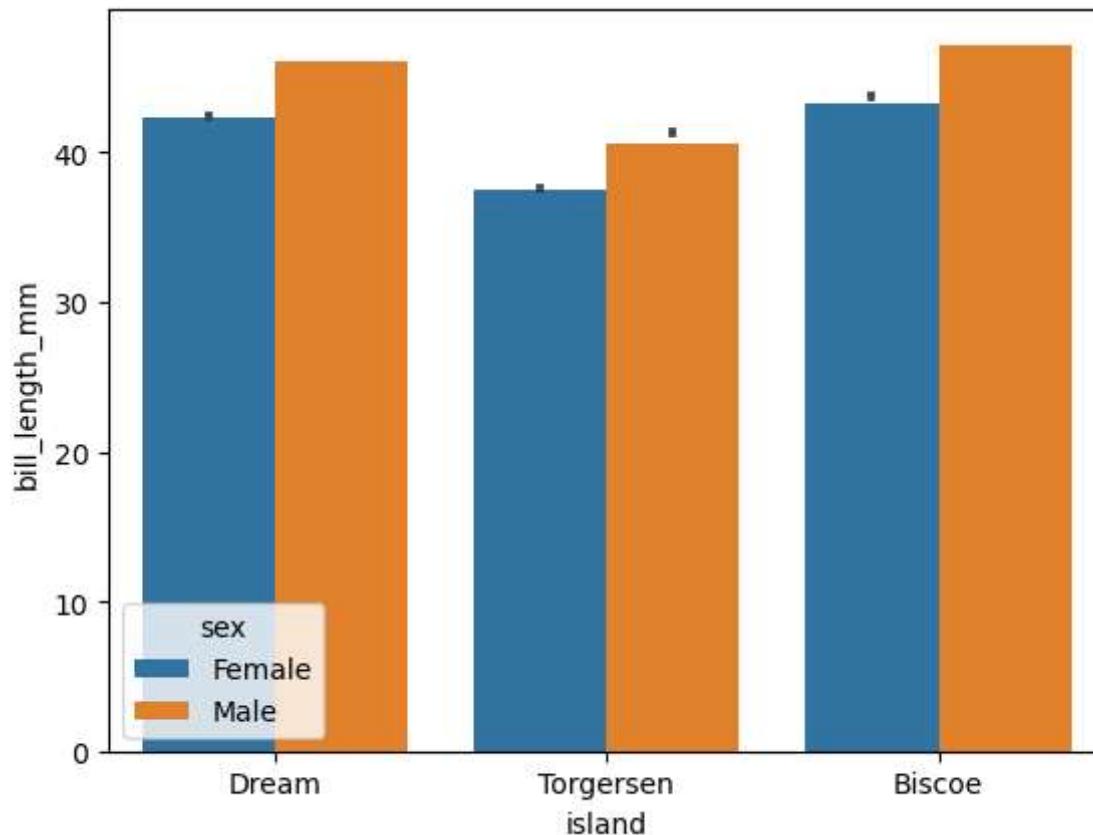


confidence interval

```
In [43]: order1 = ["Dream", "Torgersen", "Biscoe"]
sns.barplot(x="island", y ="bill_length_mm", data=df1, hue="sex",
             order = order1, hue_order=['Female','Male'],ci=10)
plt.show()
```



```
In [46]: order1 = ["Dream", "Torgersen", "Biscoe"]
sns.barplot(x="island", y ="bill_length_mm", data=df1, hue="sex",
             order = order1, hue_order=['Female','Male'],ci=10,n_boot=2)
plt.show()
```



orientation

```
In [47]: order1 = ["Dream", "Torgersen", "Biscoe"]
sns.barplot(x="island", y ="bill_length_mm", data=df1, hue="sex",
             order = order1, hue_order=['Female','Male'], orient="h")
plt.show()
```

```

-----  

TypeError                                     Traceback (most recent call last)  

~\AppData\Local\Temp\ipykernel_2520\3456891107.py in <module>  

    1 order1 = ["Dream", "Torgersen", "Biscoe"]  

----> 2 sns.barplot(x="island", y ="bill_length_mm", data=df1, hue="sex",  

    3                  order = order1, hue_order=['Female','Male'], orient="h")  

    4 plt.show()  

C:\ProgramData\Anaconda3\lib\site-packages\seaborn\_decorators.py in inner_f  

(*args, **kwargs)  

    44             )  

    45         kwargs.update({k: arg for k, arg in zip(sig.parameters, args)})  

---> 46     return f(**kwargs)  

    47     return inner_f  

    48  

C:\ProgramData\Anaconda3\lib\site-packages\seaborn\categorical.py in barplot  

(x, y, hue, data, order, hue_order, estimator, ci, n_boot, units, seed, orient,  

color, palette, saturation, errcolor, errwidth, capsize, dodge, ax, **kwargs)  

3180 ):  

3181  

-> 3182     plotter = _BarPlotter(x, y, hue, data, order, hue_order,  

3183                         estimator, ci, n_boot, units, seed,  

3184                         orient, color, palette, saturation,  

C:\ProgramData\Anaconda3\lib\site-packages\seaborn\categorical.py in __init__  

(self, x, y, hue, data, order, hue_order, estimator, ci, n_boot, units, seed,  

orient, color, palette, saturation, errcolor, errwidth, capsize, dodge)  

1582             errwidth, capsize, dodge):  

1583             """Initialize the plotter."""  

-> 1584         self.establish_variables(x, y, hue, data, orient,  

1585                             order, hue_order, units)  

1586         self.establish_colors(color, palette, saturation)  

C:\ProgramData\Anaconda3\lib\site-packages\seaborn\categorical.py in establish_variables  

(self, x, y, hue, data, orient, order, hue_order, units)  

154  

155             # Figure out the plotting orientation  

---> 156             orient = infer_orient(  

157                 x, y, orient, require_numeric=self.require_numeric  

158             )  

C:\ProgramData\Anaconda3\lib\site-packages\seaborn\_core.py in infer_orient  

(x, y, orient, require_numeric)  

1336     elif str(orient).startswith("h"):  

1337         if require_numeric and x_type != "numeric":  

-> 1338             raise TypeError(nonnumeric_dv_error.format("Horizontal",  

"x"))  

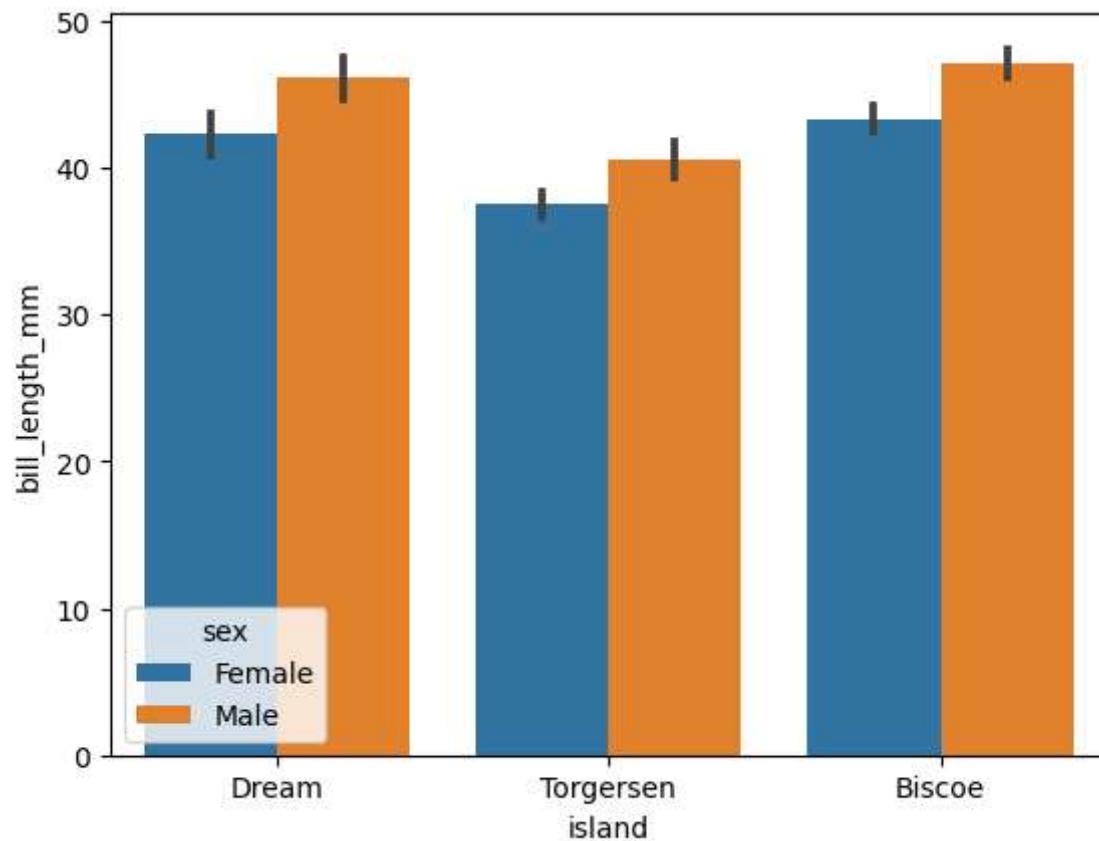
1339         return "h"  

1340

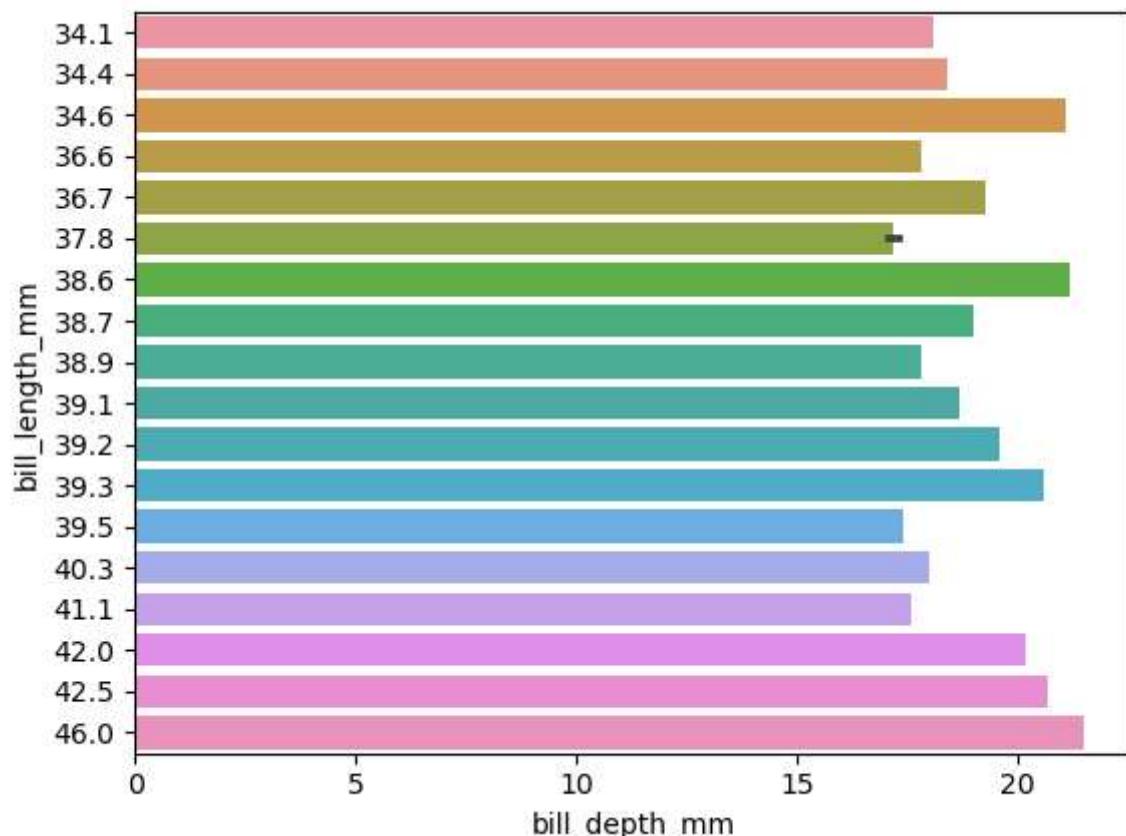
```

**TypeError:** Horizontal orientation requires numeric `x` variable.

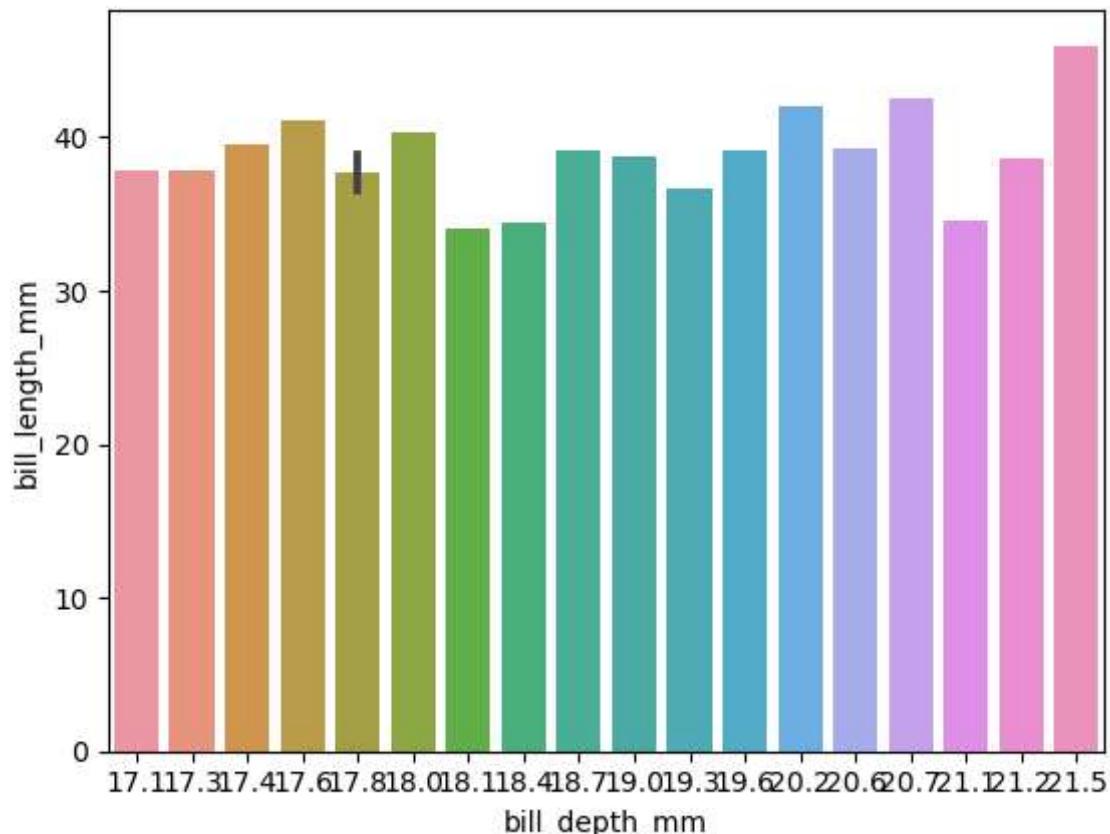
```
In [48]: order1 = ["Dream", "Torgersen", "Biscoe"]
sns.barplot(x="island", y ="bill_length_mm", data=df1, hue="sex",
             order = order1, hue_order=['Female','Male'], orient="v")
plt.show()
```



```
In [56]: sns.barplot(x="bill_depth_mm", y = "bill_length_mm", data=df1, orient="h")
plt.show()
```



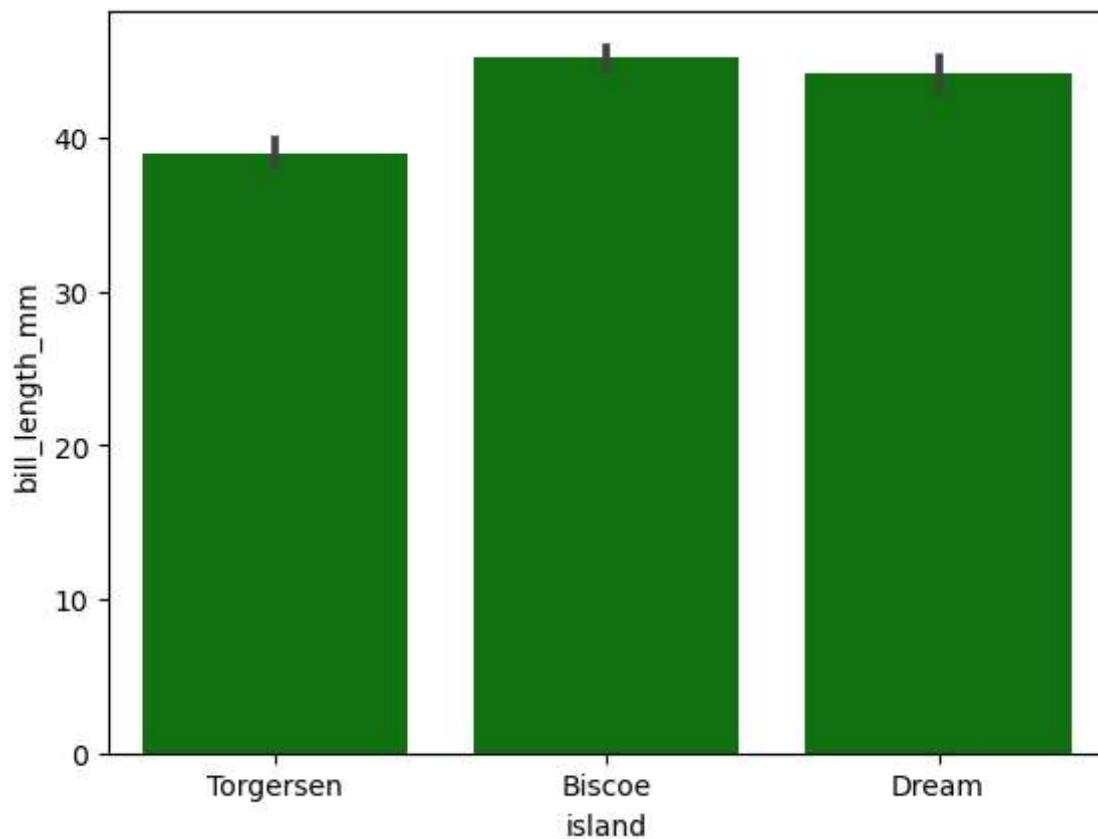
```
In [57]: sns.barplot(x="bill_depth_mm", y = "bill_length_mm", data=df1, orient="v")
plt.show()
```



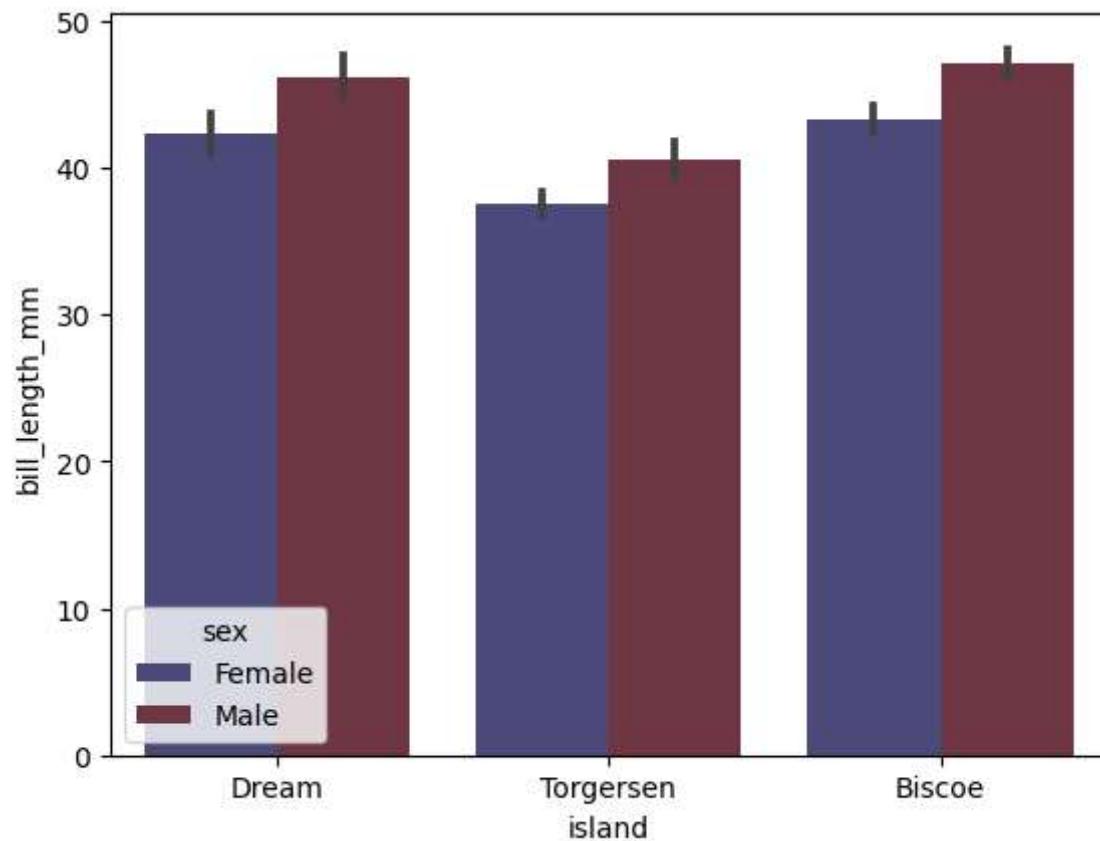
## changing the color

```
In [59]: df1 = sns.load_dataset("penguins")
```

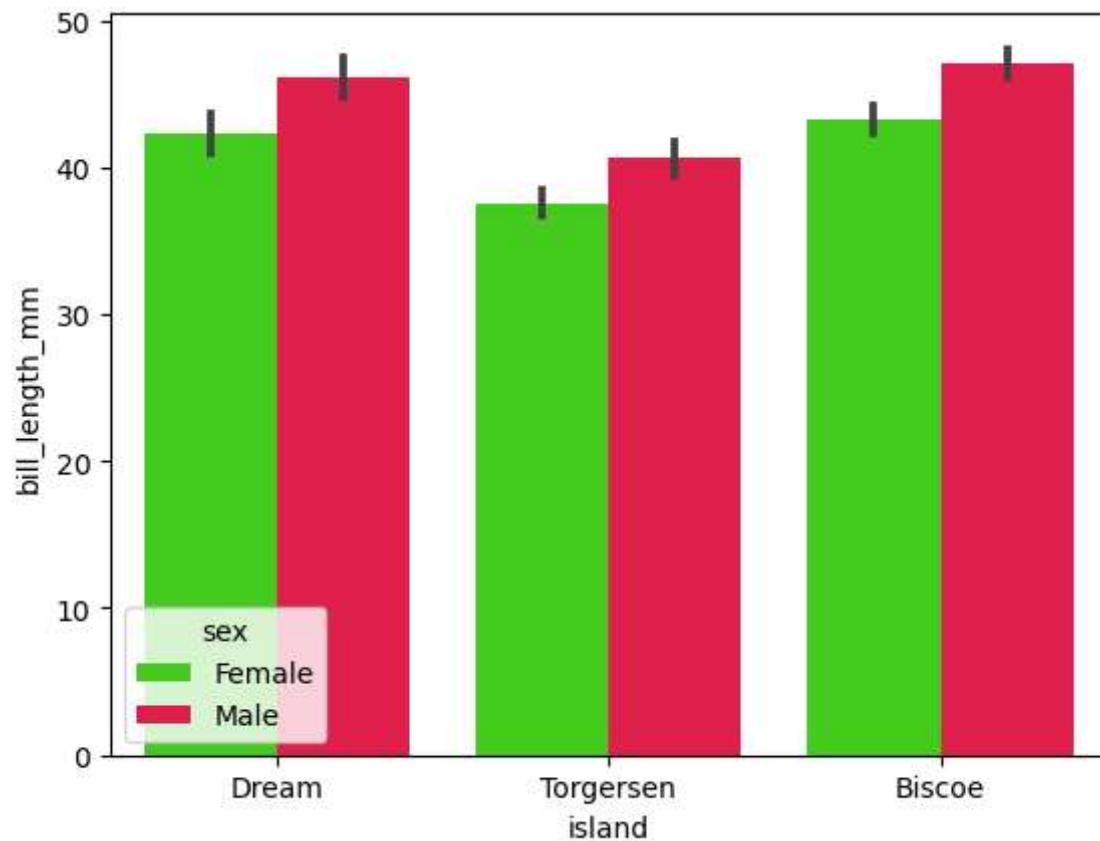
```
In [61]: order1 = ["Dream", "Torgersen", "Biscoe"]
sns.barplot(x="island", y ="bill_length_mm", data=df1, color="g")
plt.show()
```



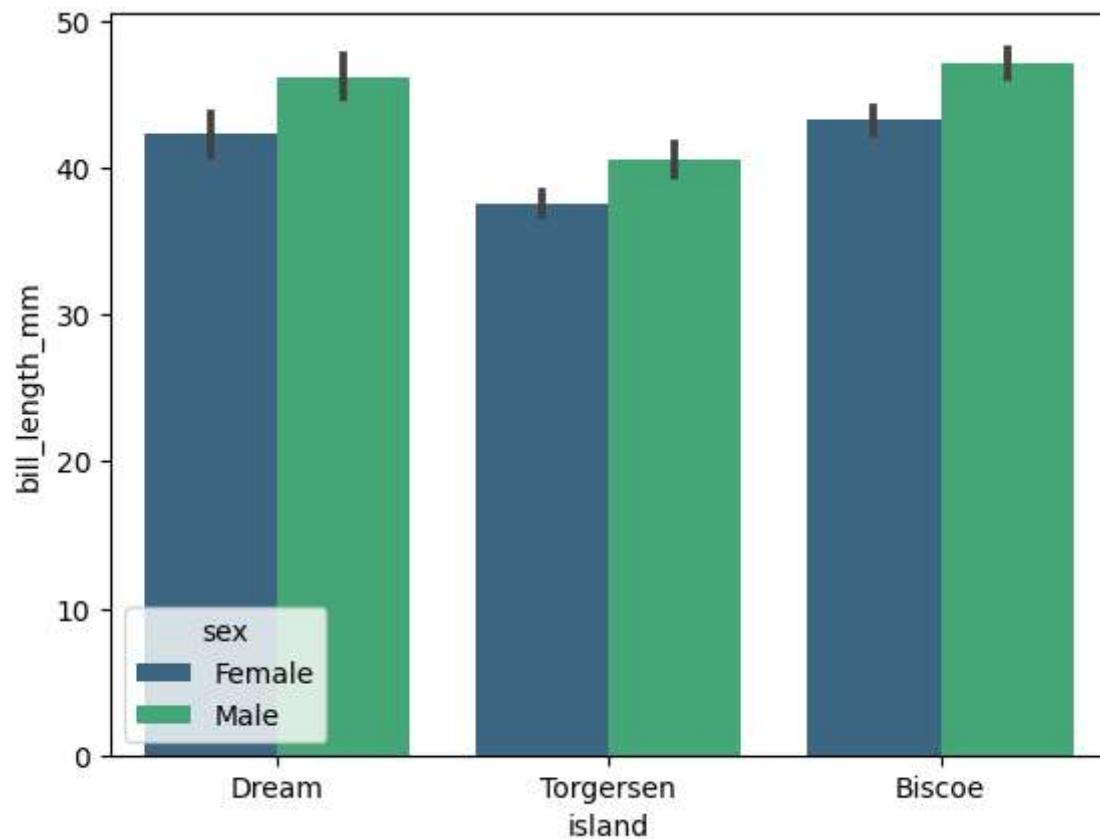
```
In [64]: order1 = ["Dream", "Torgersen", "Biscoe"]
sns.barplot(x="island", y ="bill_length_mm", data=df1, hue="sex",
             order = order1, hue_order=['Female','Male'], orient="v", palette='icefire')
plt.show()
```



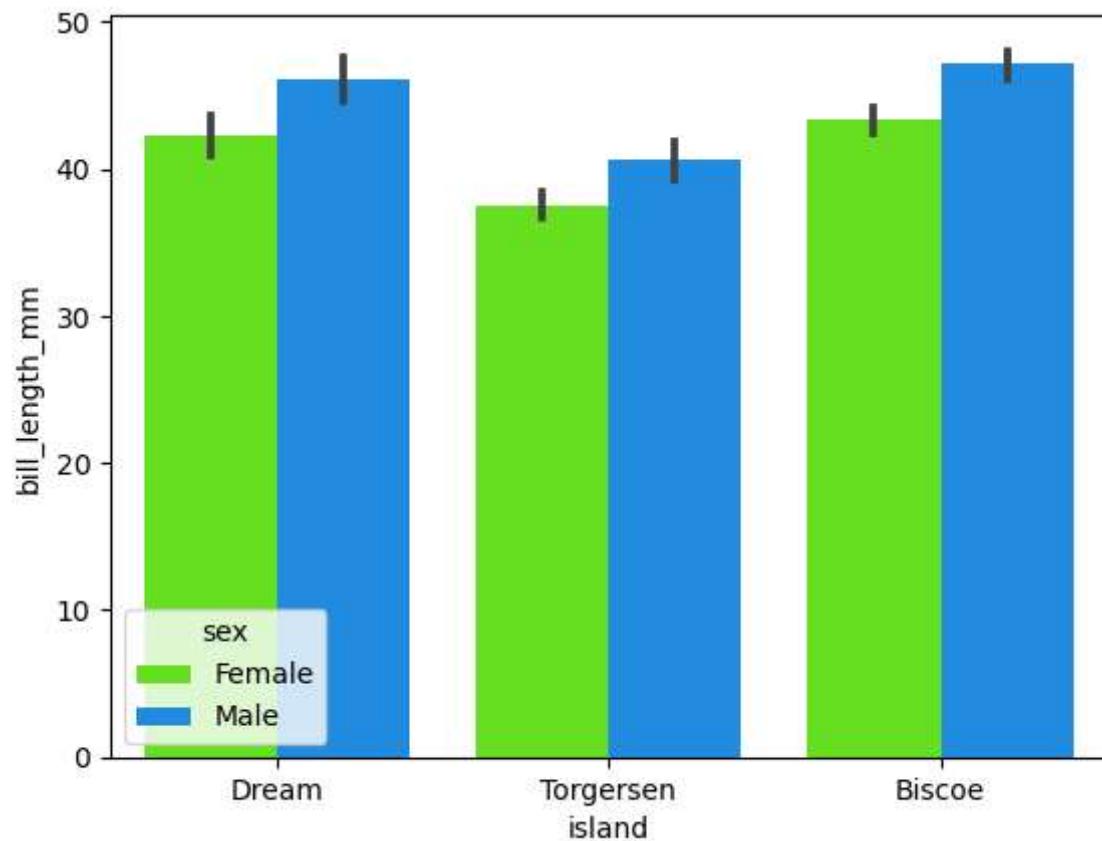
```
In [65]: order1 = ["Dream", "Torgersen", "Biscoe"]
sns.barplot(x="island", y ="bill_length_mm", data=df1, hue="sex",
             order = order1, hue_order=['Female','Male'], orient="v", palette='pr
plt.show()
```



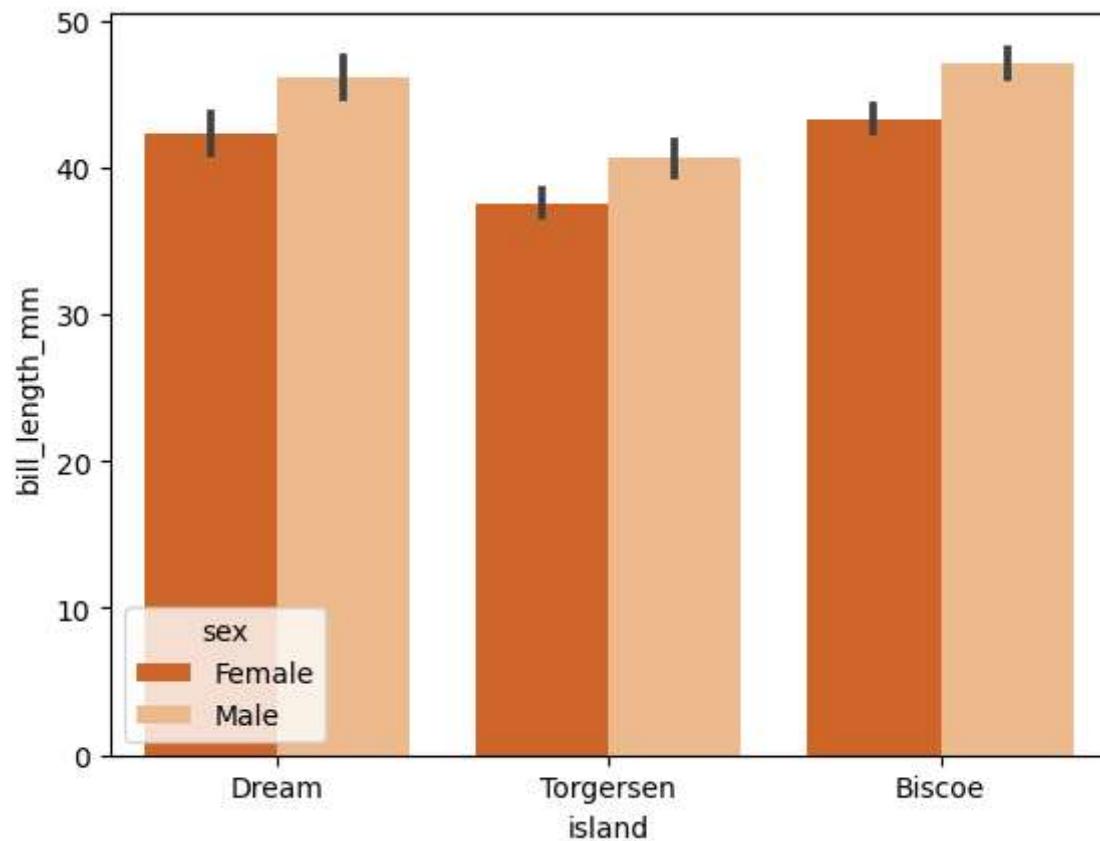
```
In [66]: order1 = ["Dream", "Torgersen", "Biscoe"]
sns.barplot(x="island", y ="bill_length_mm", data=df1, hue="sex",
             order = order1, hue_order=['Female','Male'], orient="v", palette='viridis')
plt.show()
```



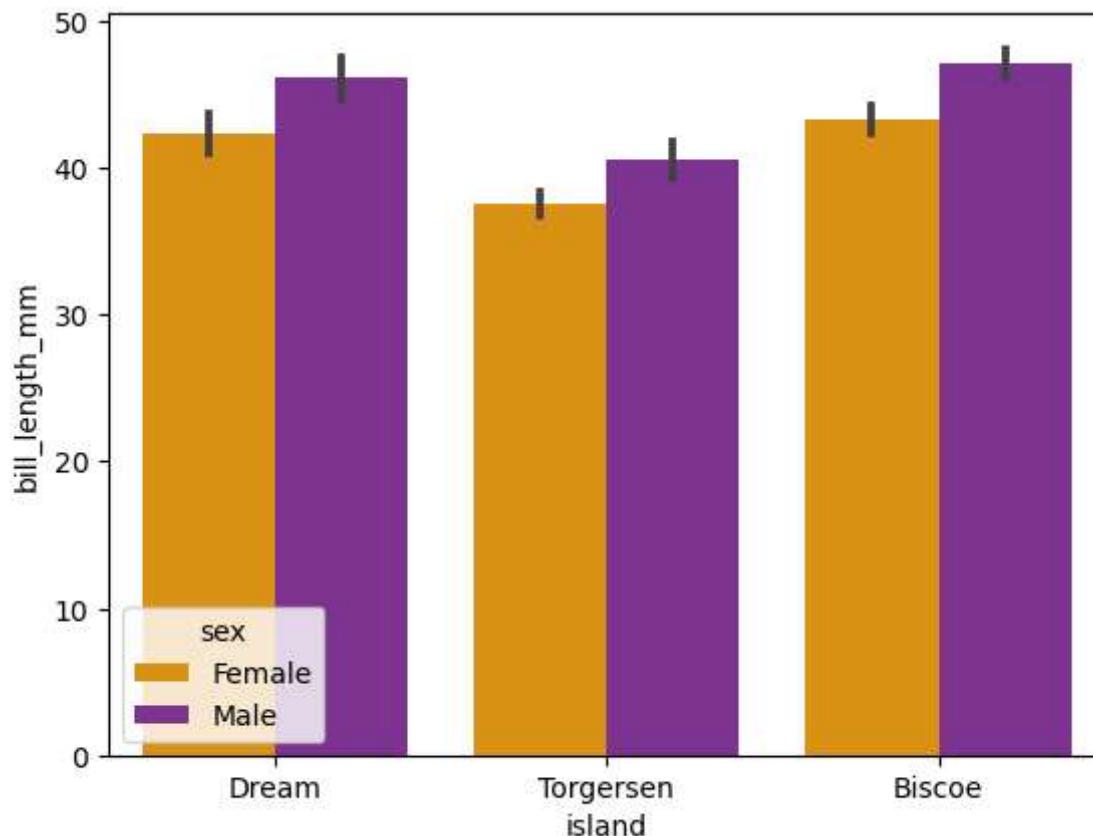
```
In [67]: order1 = ["Dream", "Torgersen", "Biscoe"]
sns.barplot(x="island", y ="bill_length_mm", data=df1, hue="sex",
             order = order1, hue_order=['Female','Male'], orient="v", palette='g'
plt.show()
```



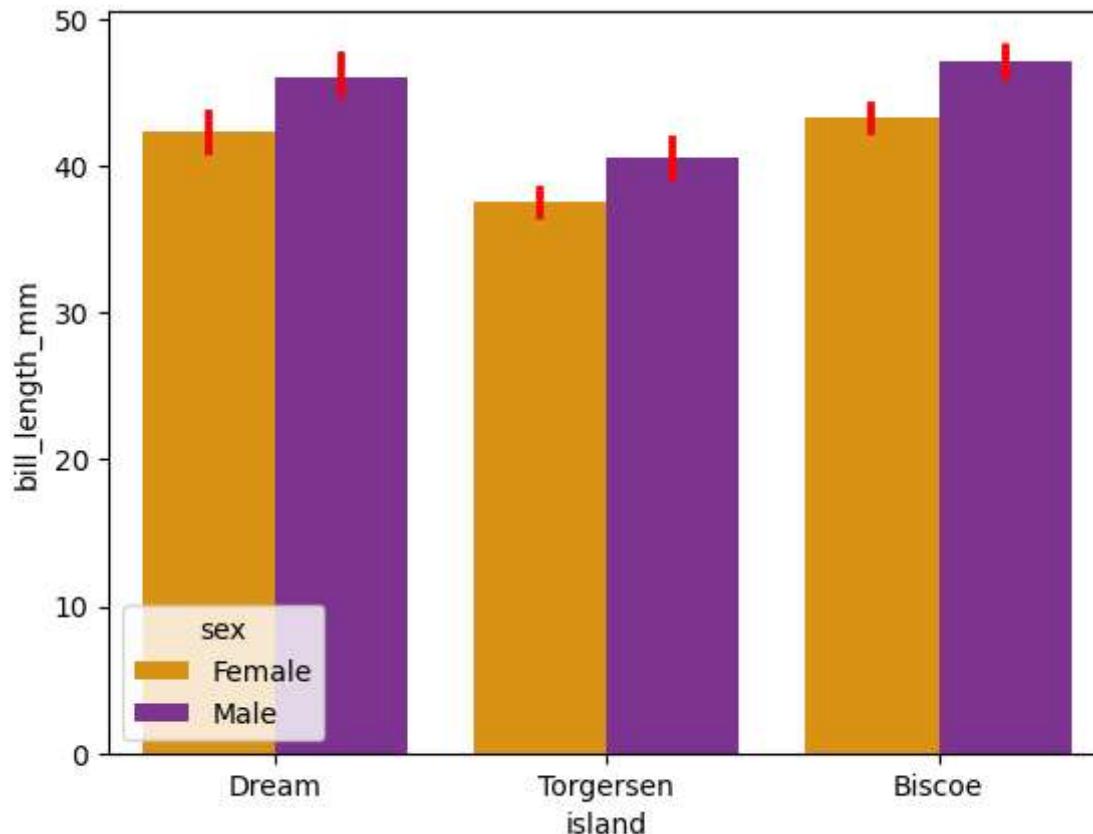
```
In [69]: order1 = ["Dream", "Torgersen", "Biscoe"]
sns.barplot(x="island", y ="bill_length_mm", data=df1, hue="sex",
             order = order1, hue_order=['Female','Male'], orient="v", palette='Oranges')
plt.show()
```



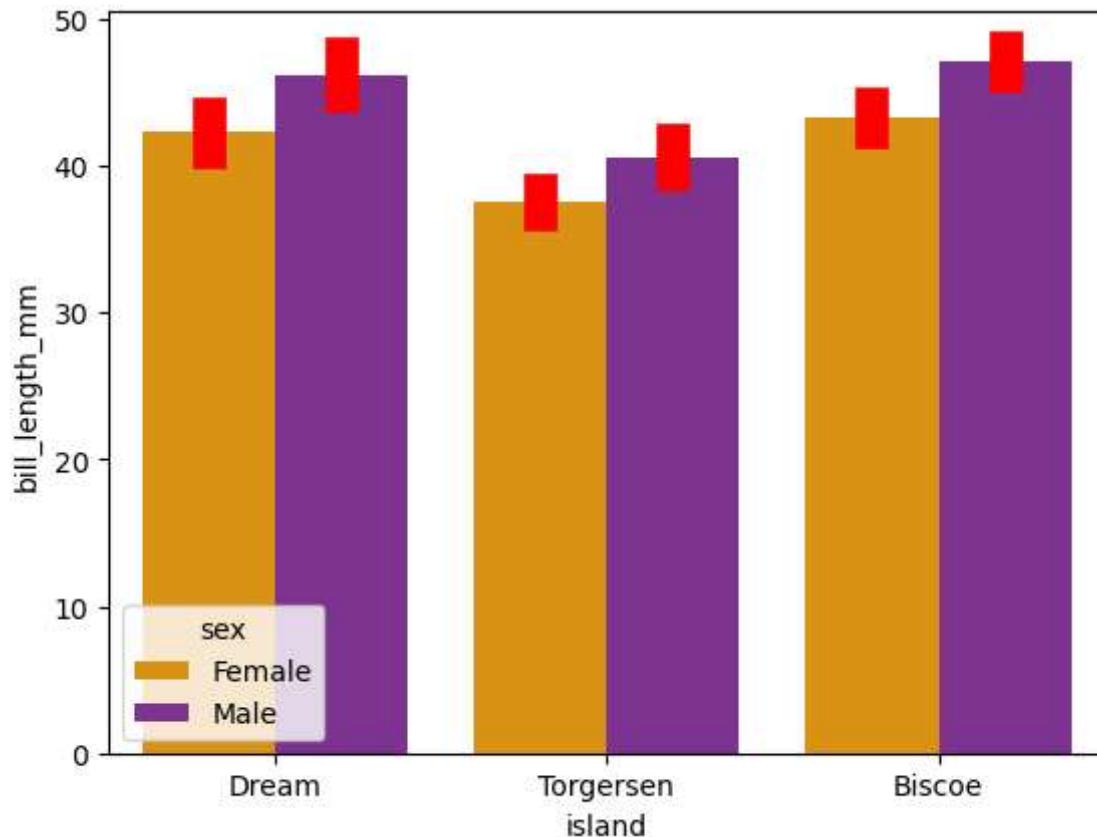
```
In [76]: order1 = ["Dream", "Torgersen", "Biscoe"]
sns.barplot(x="island", y ="bill_length_mm", data=df1, hue="sex",
            order = order1, hue_order=['Female','Male'], orient="v", palette='CMRmap',
            saturation = 0.9)
plt.show()
```



```
In [78]: order1 = ["Dream", "Torgersen", "Biscoe"]
sns.barplot(x="island", y ="bill_length_mm", data=df1, hue="sex",
            order = order1, hue_order=['Female','Male'], orient="v", palette='CMRmap',
            saturation = 0.9, errcolor="r")
plt.show()
```

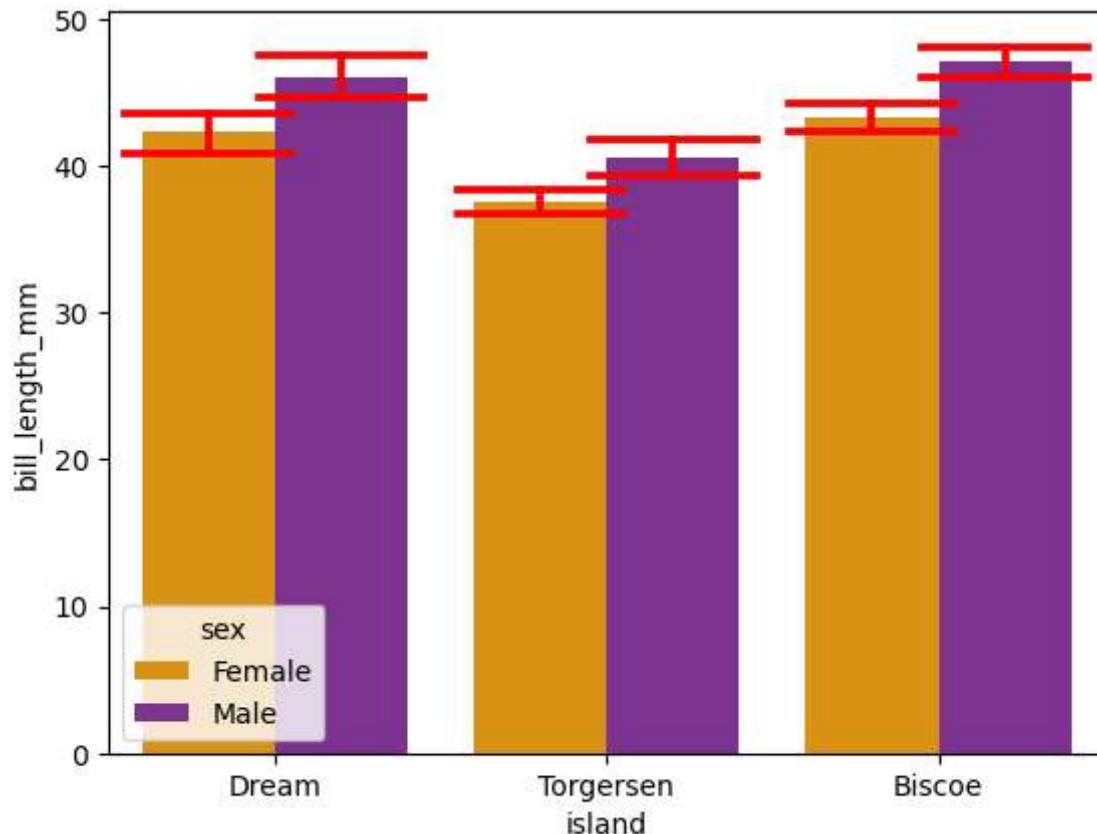


```
In [79]: order1 = ["Dream", "Torgersen", "Biscoe"]
sns.barplot(x="island", y ="bill_length_mm", data=df1, hue="sex",
            order = order1, hue_order=['Female', 'Male'], orient="v", palette='CMRmap',
            saturation = 0.9, errcolor="r", errwidth=12)
plt.show()
```



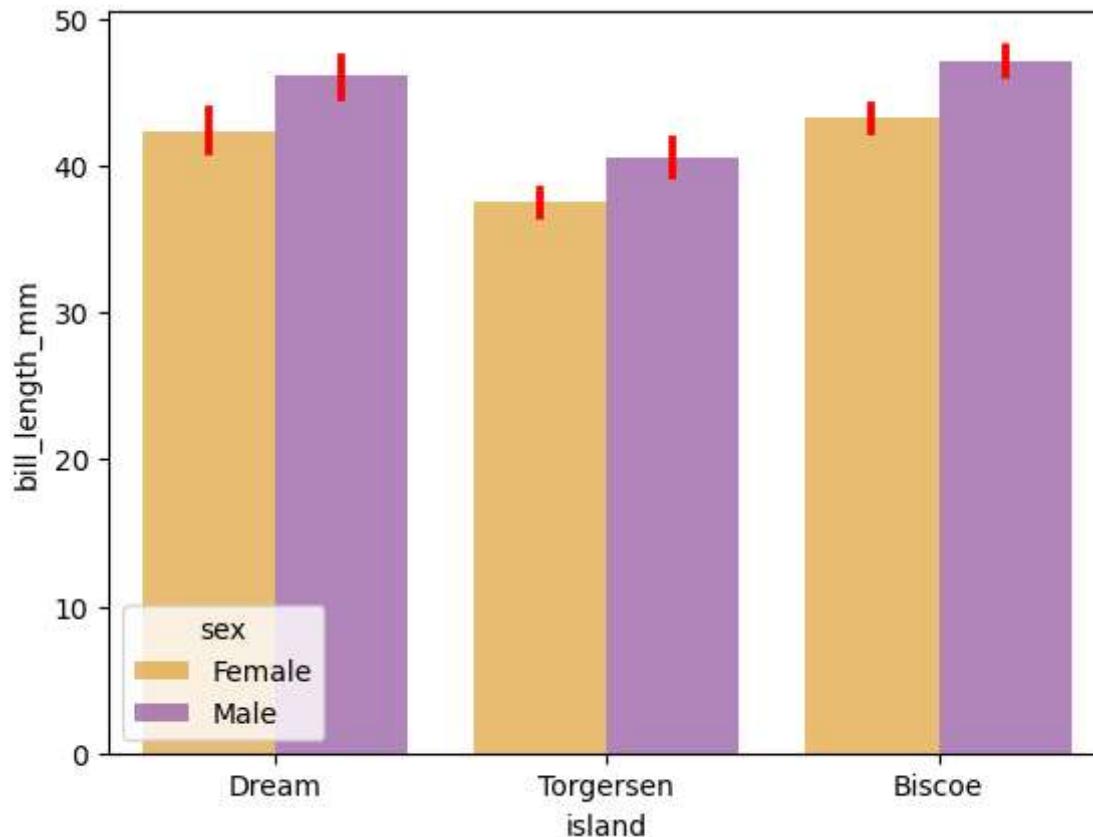
## capsize

```
In [82]: order1 = ["Dream", "Torgersen", "Biscoe"]
sns.barplot(x="island", y ="bill_length_mm", data=df1, hue="sex",
            order = order1, hue_order=['Female','Male'], orient="v", palette='CMRmap',
            saturation = 0.9, errcolor="r", capsize = 0.5)
plt.show()
```



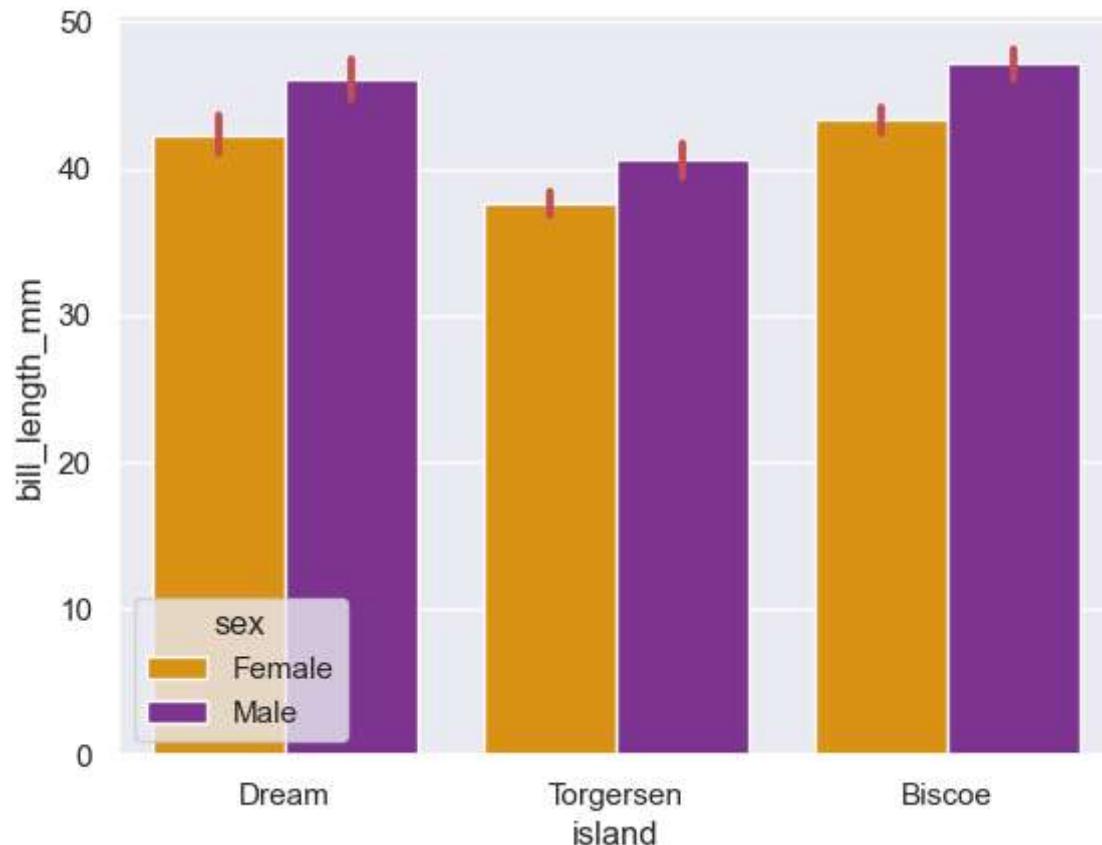
## alpha

```
In [85]: order1 = ["Dream", "Torgersen", "Biscoe"]
sns.barplot(x="island", y ="bill_length_mm", data=df1, hue="sex",
            order = order1, hue_order=['Female','Male'], orient="v", palette='CMRmap',
            saturation = 0.9, errcolor="r", alpha = 0.6)
plt.show()
```

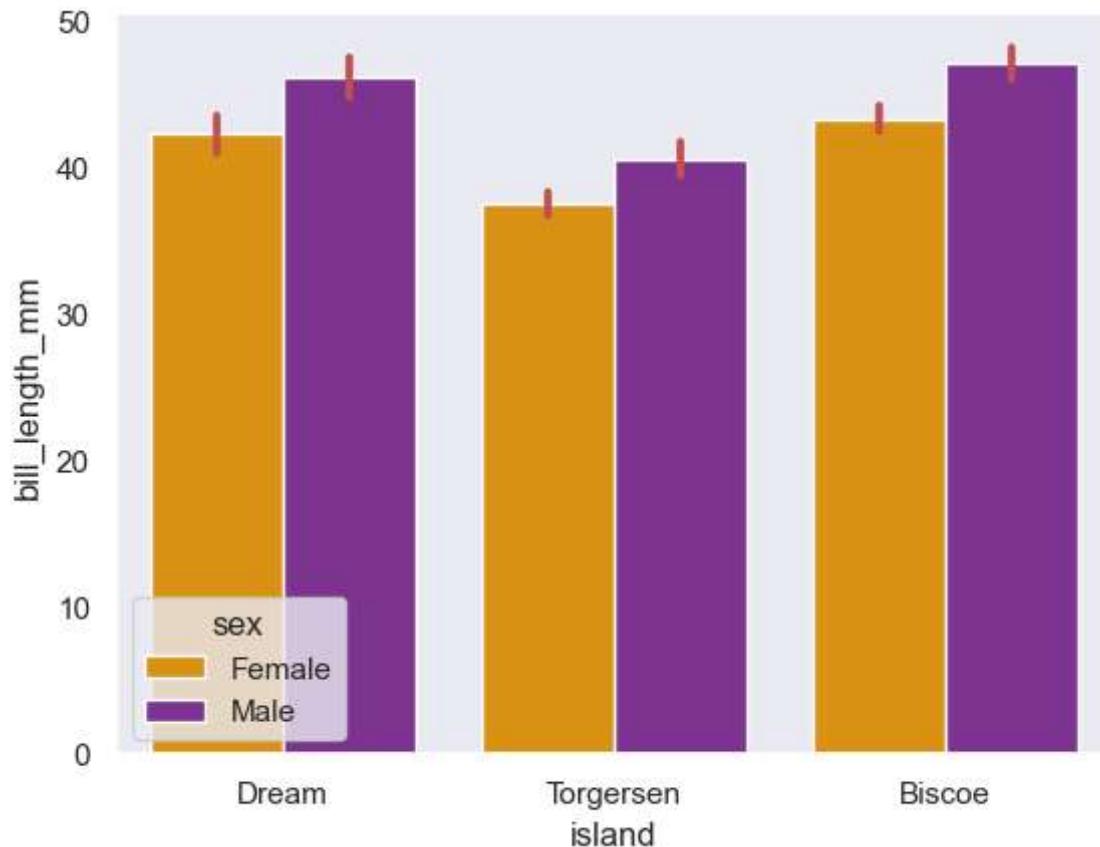


## Setting Grid

```
In [87]: sns.set(style="darkgrid")
order1 = ["Dream", "Torgersen", "Biscoe"]
sns.barplot(x="island", y ="bill_length_mm", data=df1, hue="sex",
            order = order1, hue_order=['Female', 'Male'], orient="v", palette='CMRmap',
            saturation = 0.9, errcolor="r")
plt.show()
```



```
In [144]: sns.set(style="dark")
order1 = ["Dream", "Torgersen", "Biscoe"]
sns.barplot(x="island", y ="bill_length_mm", data=df1, hue="sex",
            order = order1, hue_order=['Female', 'Male'], orient="v", palette='CMRmap',
            saturation = 0.9, errcolor="r")
plt.show()
```



```
In [ ]:
```

## Histogram in Seaborn

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

```
In [2]: df1 = sns.load_dataset("penguins")
```

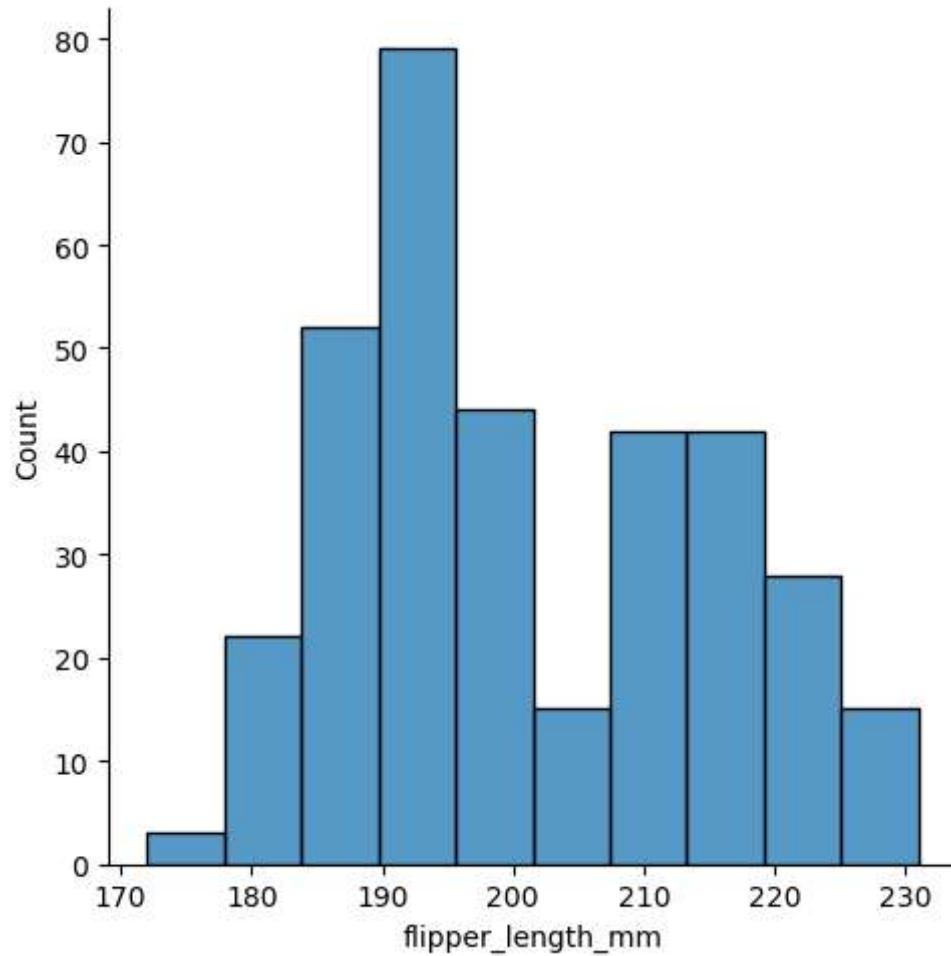
In [3]: df1

	species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	sex
0	Adelie	Torgersen	39.1	18.7	181.0	3750.0	Male
1	Adelie	Torgersen	39.5	17.4	186.0	3800.0	Female
2	Adelie	Torgersen	40.3	18.0	195.0	3250.0	Female
3	Adelie	Torgersen	NaN	NaN	NaN	NaN	NaN
4	Adelie	Torgersen	36.7	19.3	193.0	3450.0	Female
...	...	...	...	...	...	...	...
339	Gentoo	Biscoe	NaN	NaN	NaN	NaN	NaN
340	Gentoo	Biscoe	46.8	14.3	215.0	4850.0	Female
341	Gentoo	Biscoe	50.4	15.7	222.0	5750.0	Male
342	Gentoo	Biscoe	45.2	14.8	212.0	5200.0	Female
343	Gentoo	Biscoe	49.9	16.1	213.0	5400.0	Male

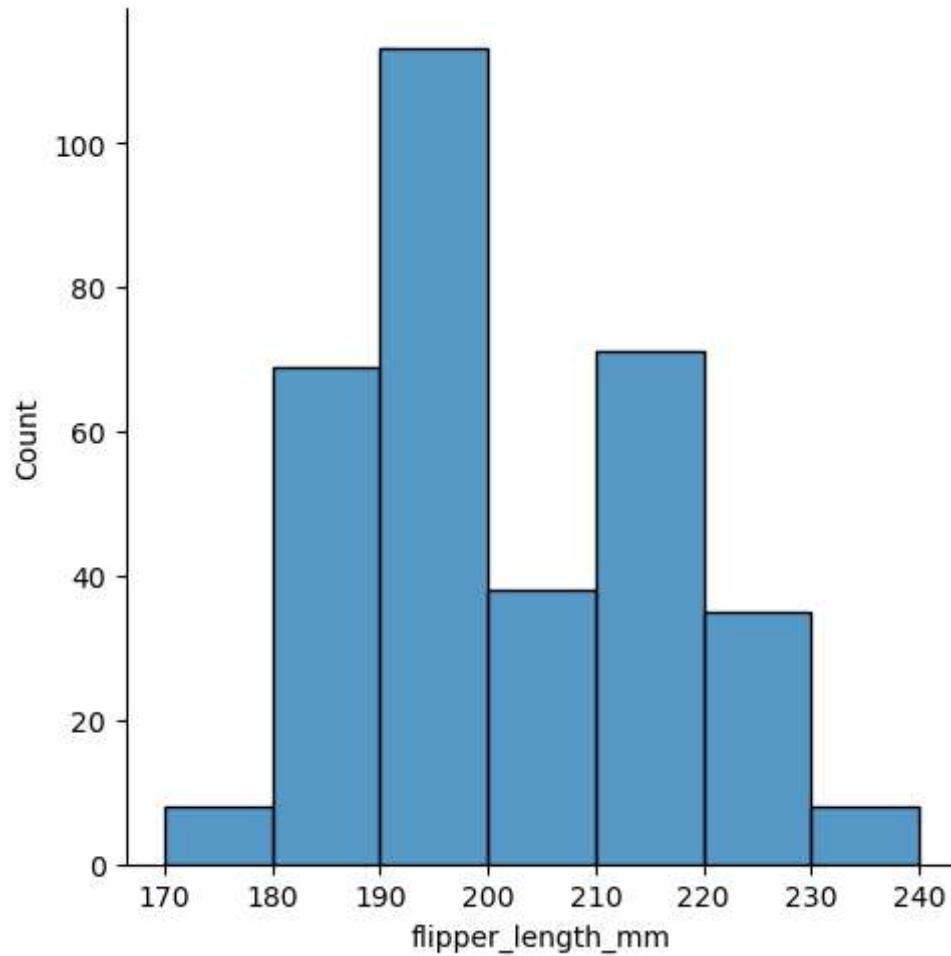
344 rows × 7 columns



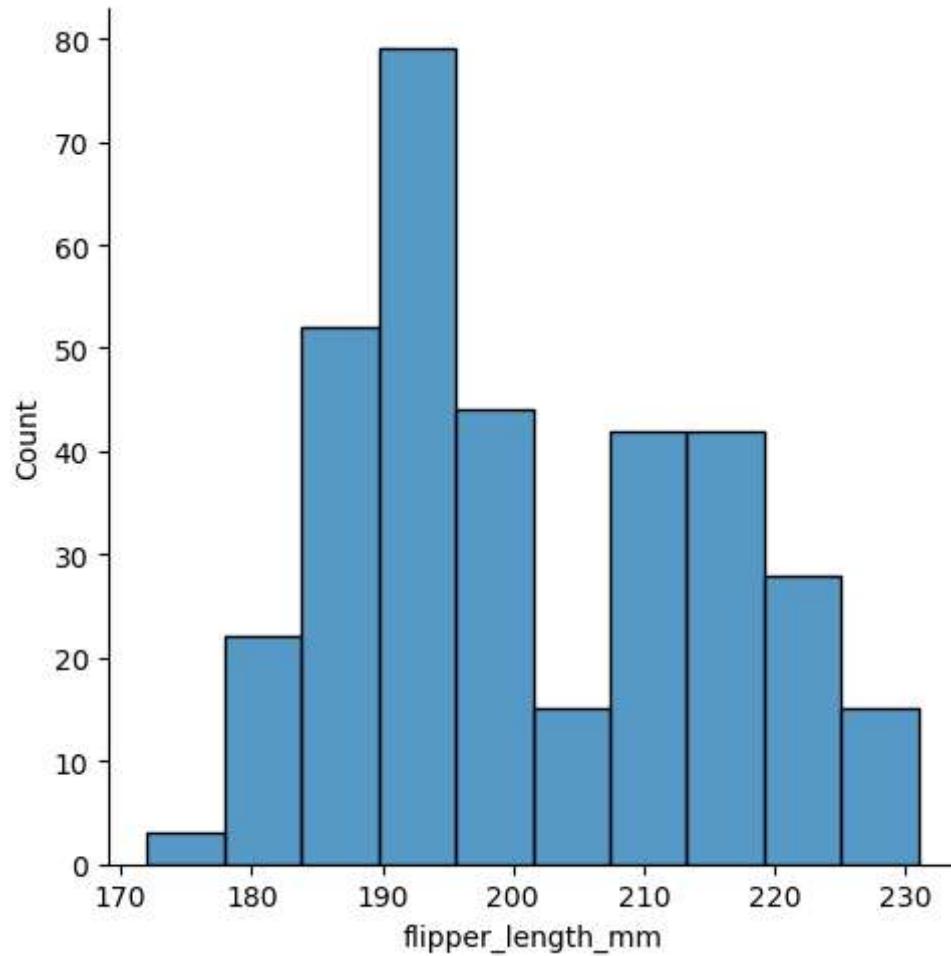
```
In [4]: sns.displot(df1["flipper_length_mm"])
plt.show()
```



```
In [5]: sns.displot(df1["flipper_length_mm"], bins=[170,180,190,200,210,220,230,240])  
plt.show()
```

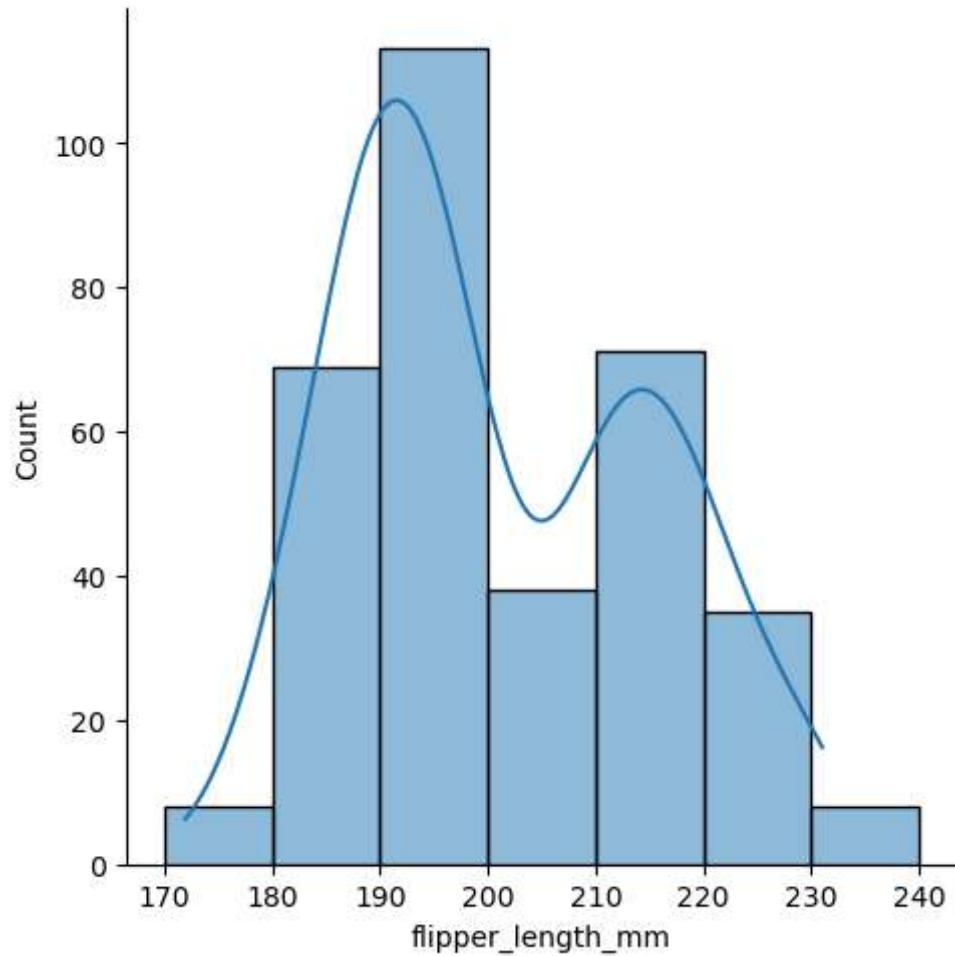


```
In [6]: sns.displot(df1["flipper_length_mm"])
plt.show()
```

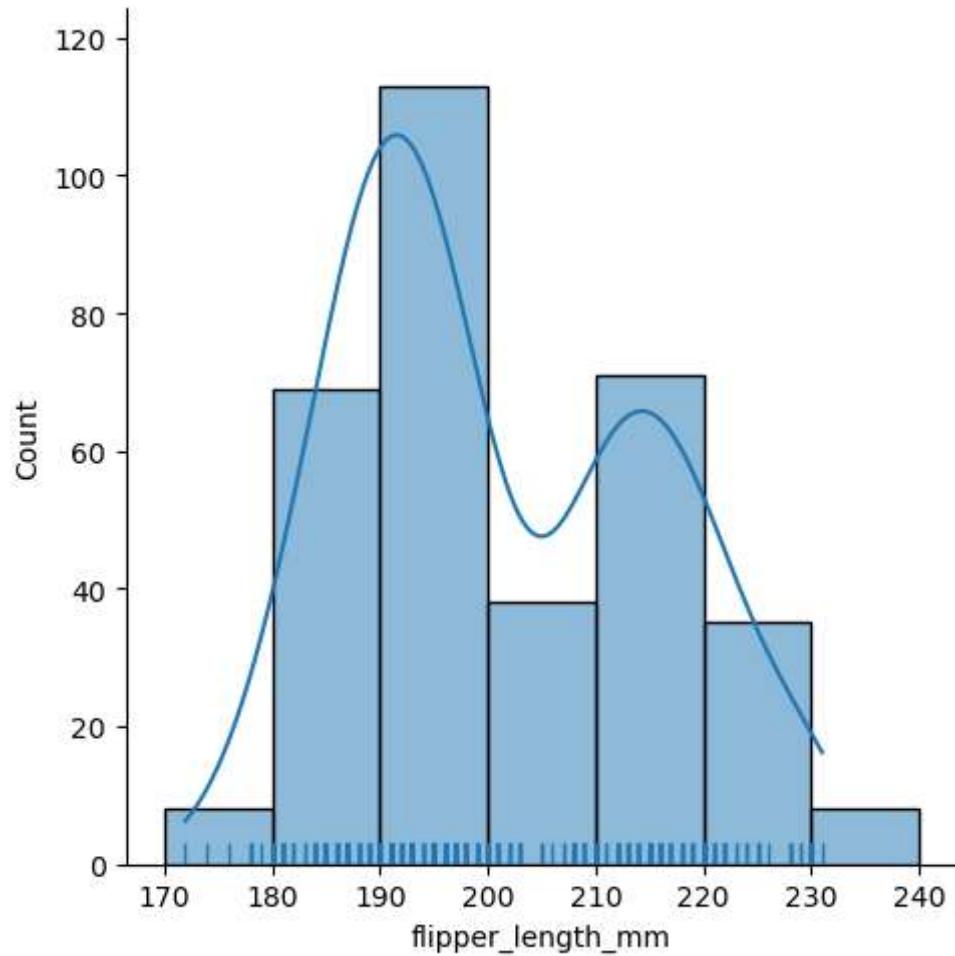


## kde - A kernel density estimate (KDE) plot

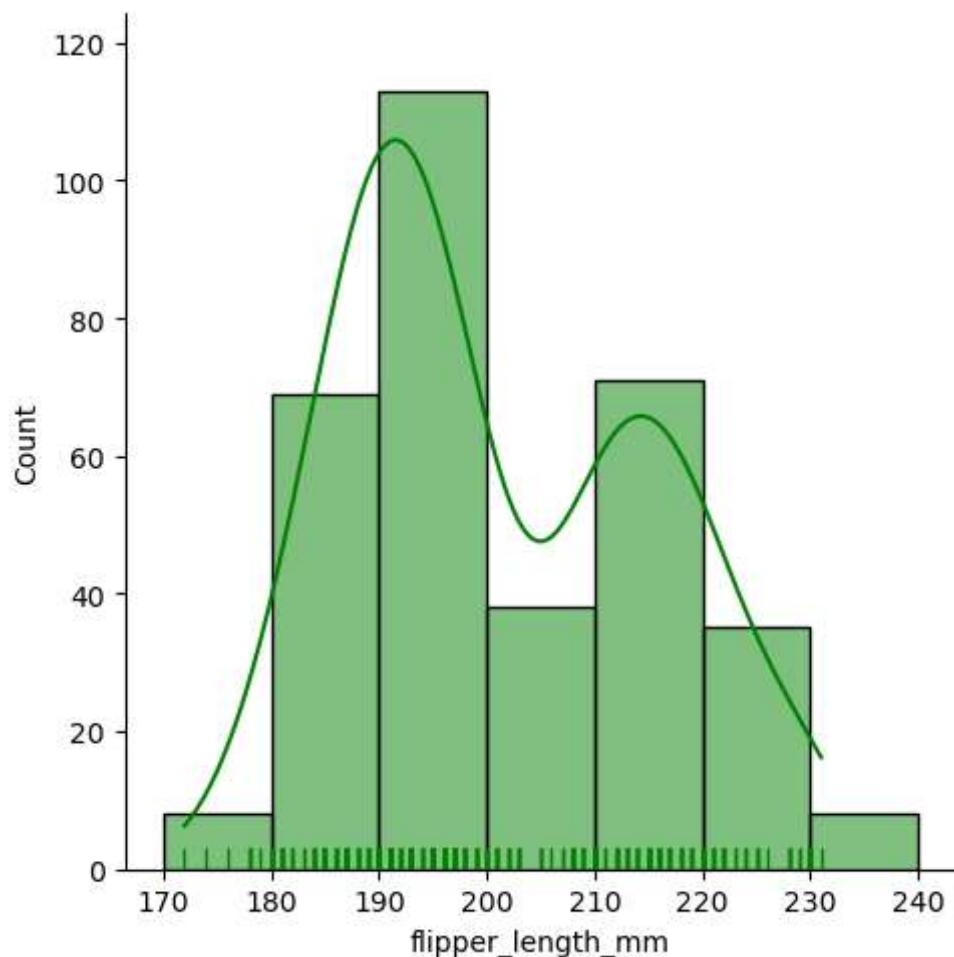
```
In [7]: sns.displot(df1["flipper_length_mm"], bins=[170,180,190,200,210,220,230,240],  
plt.show()
```



```
In [8]: sns.displot(df1["flipper_length_mm"], bins=[170,180,190,200,210,220,230,240],  
plt.show()
```

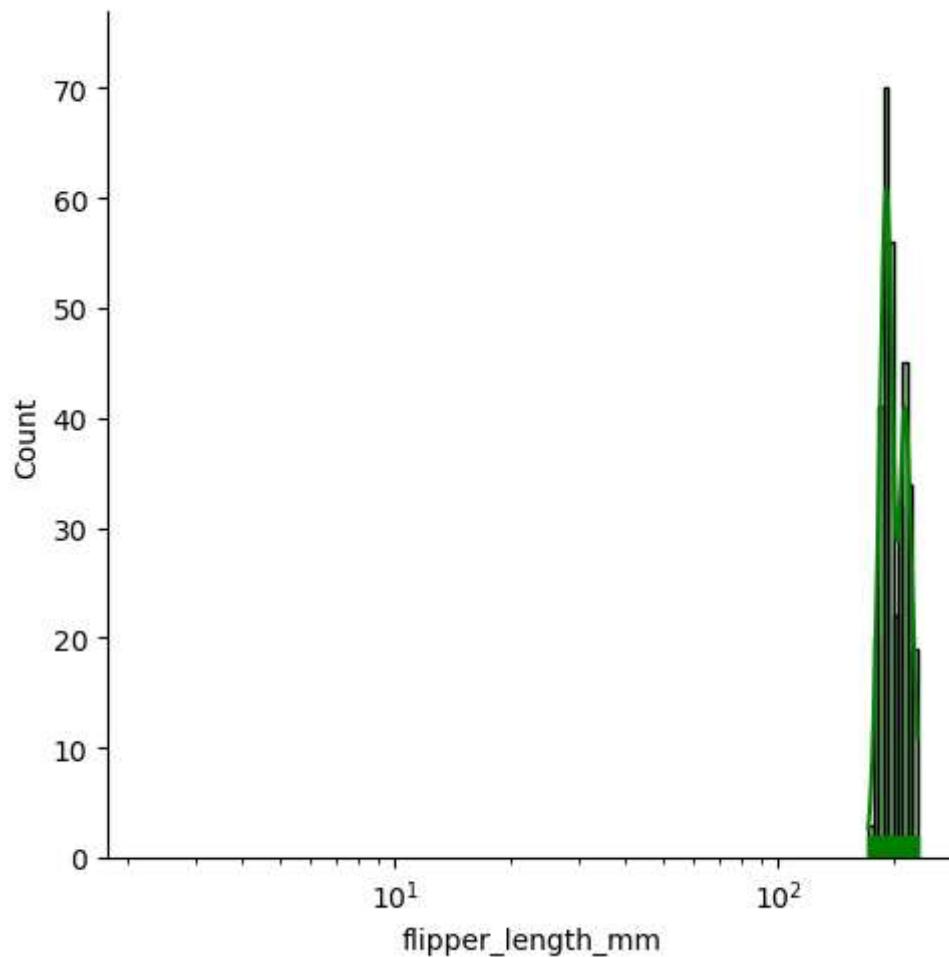


```
In [9]: sns.displot(df1["flipper_length_mm"], bins=[170,180,190,200,210,220,230,240],  
                  kde=True, rug = True, color="g")  
plt.show()
```



## logscale

```
In [10]: sns.displot(df1["flipper_length_mm"],  
                    kde=True, rug = True, color="g", log_scale = True)  
plt.show()
```



## scatter plot in seaborn

```
In [17]: import seaborn as sns  
import matplotlib.pyplot as plt  
import pandas as pd
```

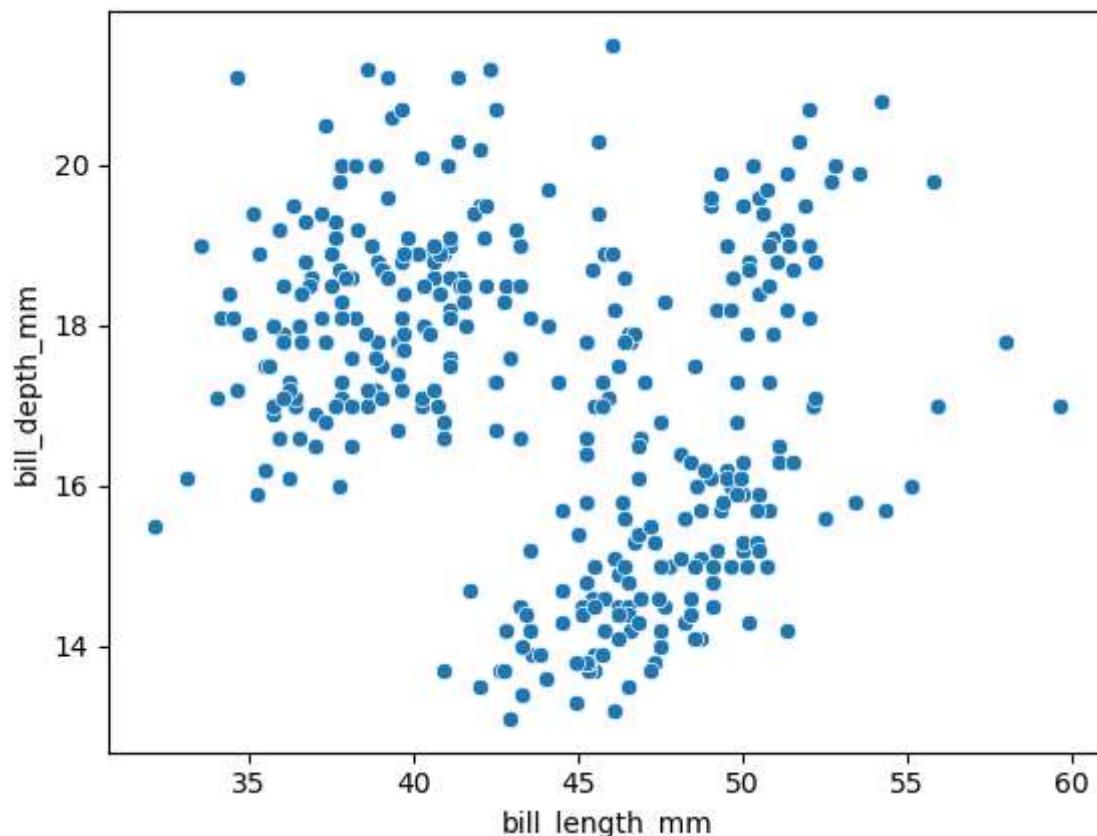
```
In [18]: df1 = sns.load_dataset("penguins")
```

In [19]: df1

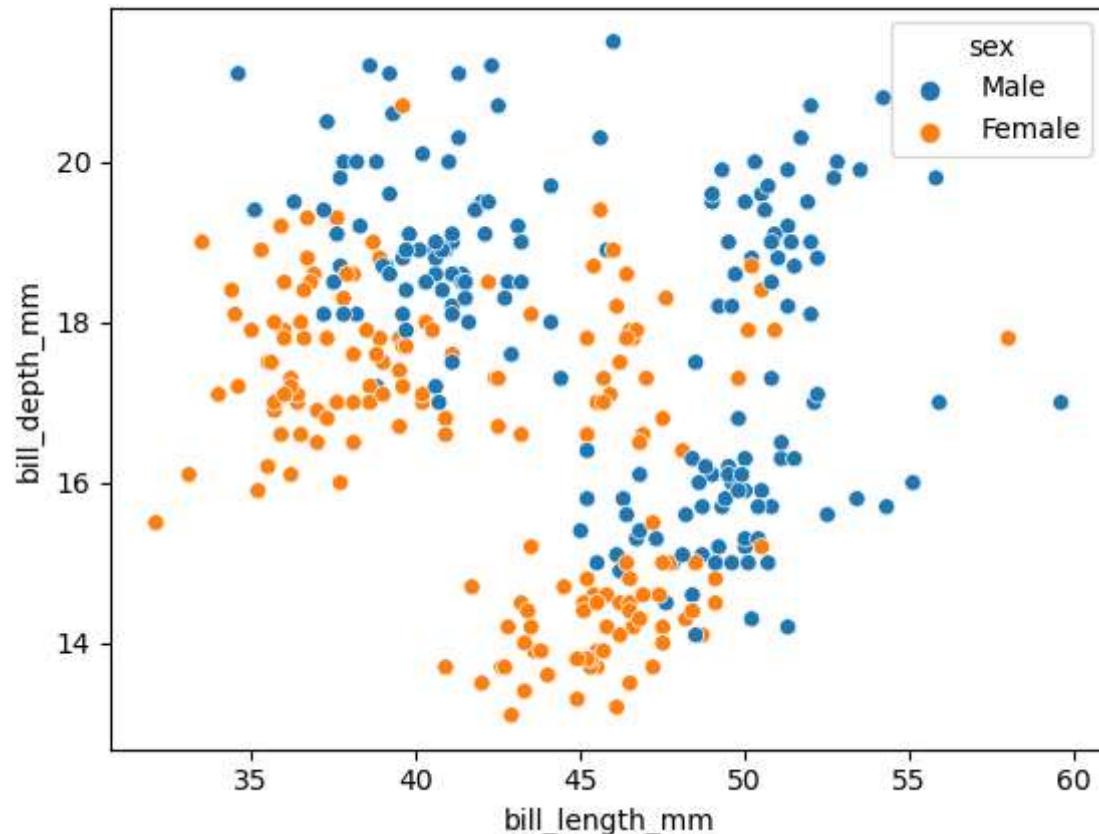
Out[19]:

	species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	sex
0	Adelie	Torgersen	39.1	18.7	181.0	3750.0	Male
1	Adelie	Torgersen	39.5	17.4	186.0	3800.0	Female
2	Adelie	Torgersen	40.3	18.0	195.0	3250.0	Female
3	Adelie	Torgersen	NaN	NaN	NaN	NaN	NaN
4	Adelie	Torgersen	36.7	19.3	193.0	3450.0	Female
...	...	...	...	...	...	...	...
339	Gentoo	Biscoe	NaN	NaN	NaN	NaN	NaN
340	Gentoo	Biscoe	46.8	14.3	215.0	4850.0	Female
341	Gentoo	Biscoe	50.4	15.7	222.0	5750.0	Male
342	Gentoo	Biscoe	45.2	14.8	212.0	5200.0	Female
343	Gentoo	Biscoe	49.9	16.1	213.0	5400.0	Male

344 rows × 7 columns

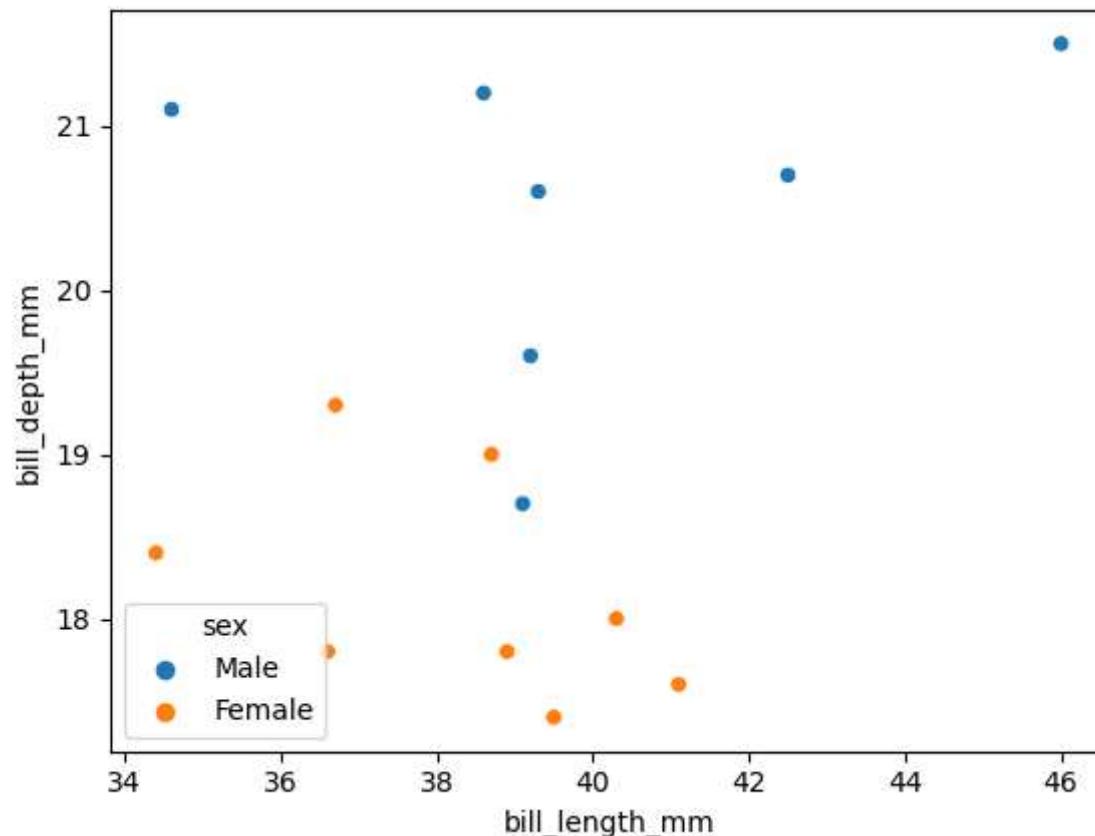
In [20]: sns.scatterplot(x="bill\_length\_mm", y = "bill\_depth\_mm", data = df1)  
plt.show()

```
In [21]: sns.scatterplot(x="bill_length_mm", y ="bill_depth_mm", data = df1, hue = "sex")
plt.show()
```

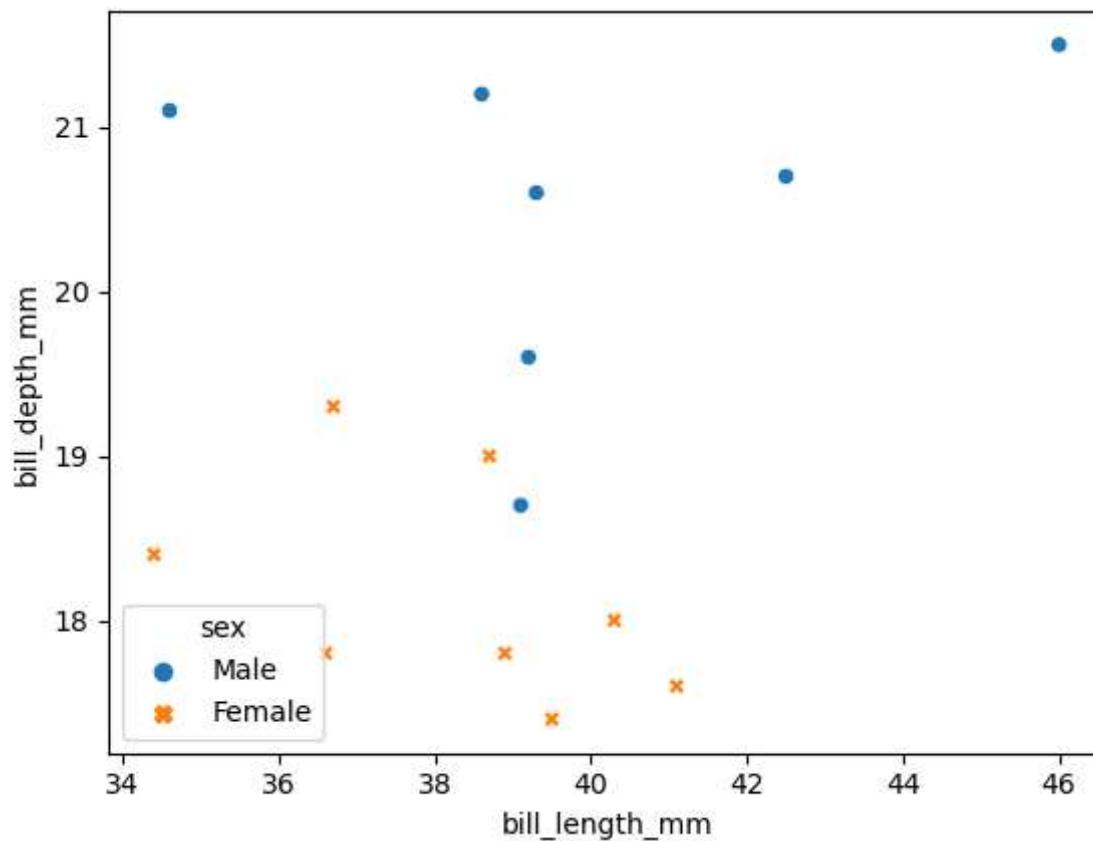


```
In [23]: df1 = sns.load_dataset("penguins").head(20)
```

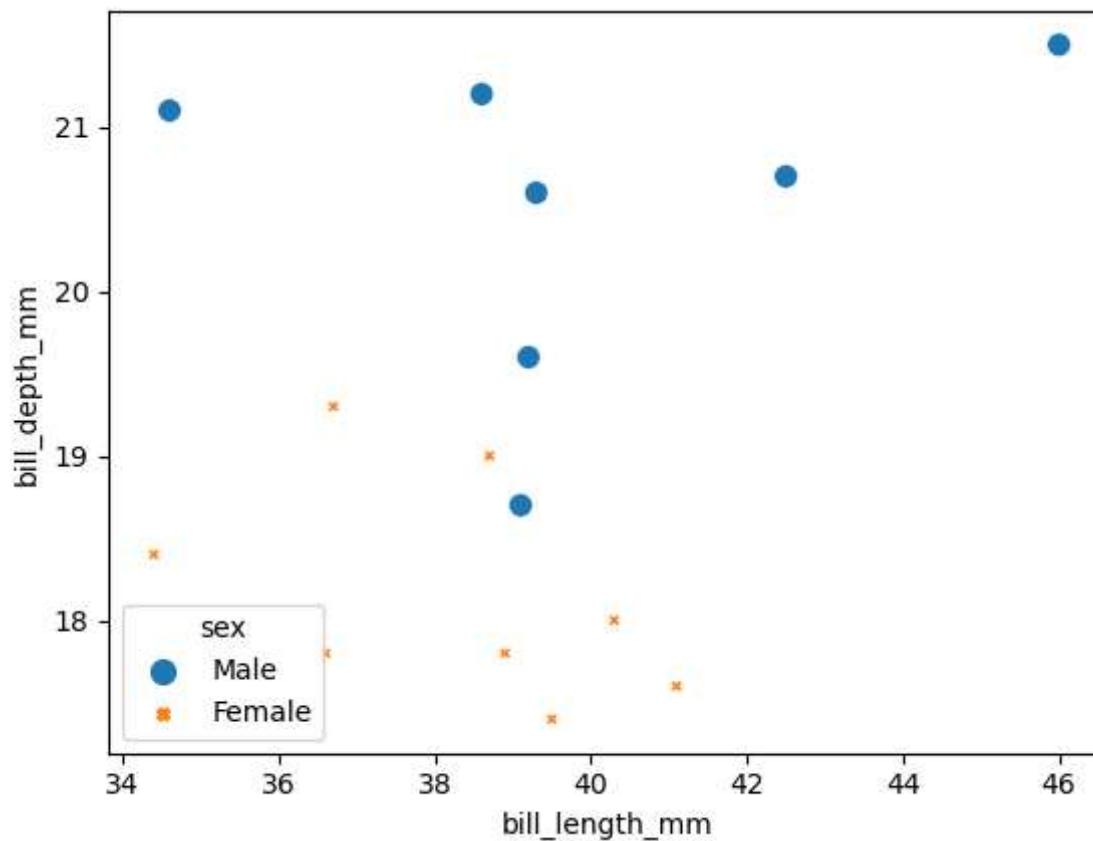
```
In [24]: sns.scatterplot(x="bill_length_mm", y ="bill_depth_mm", data = df1, hue = "sex  
plt.show()
```



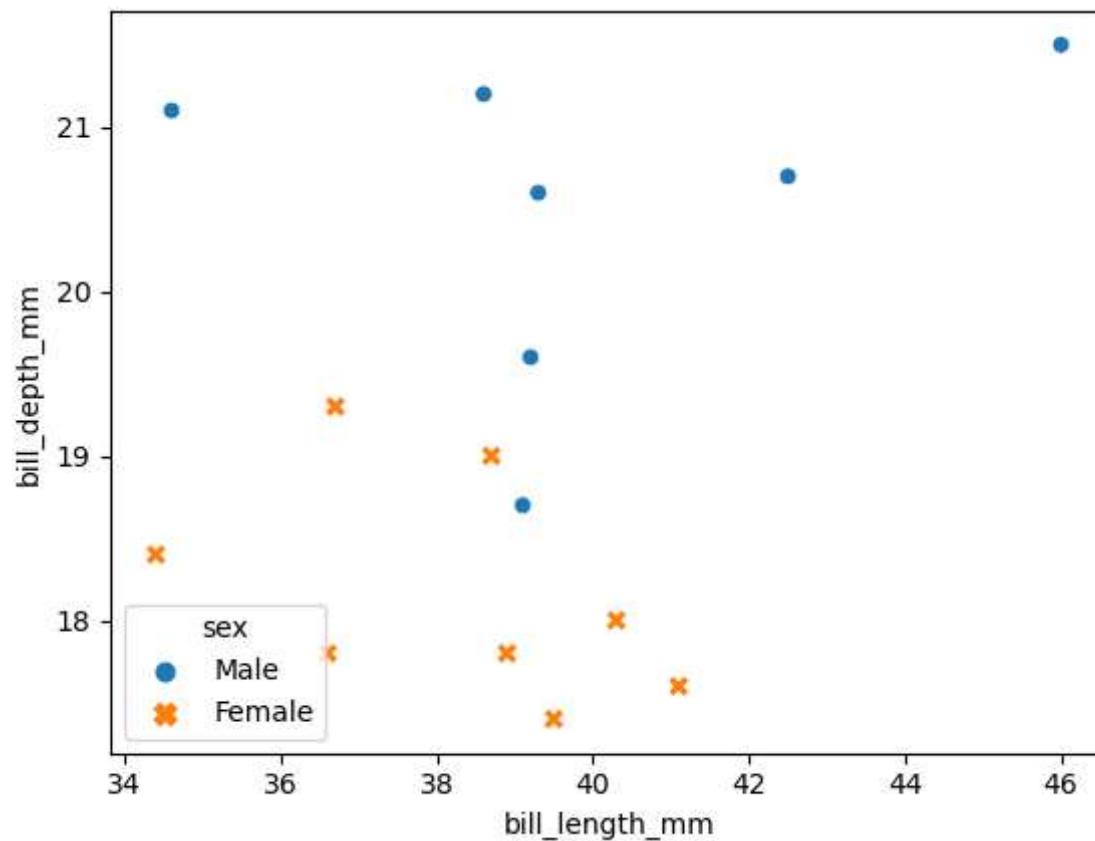
```
In [25]: sns.scatterplot(x="bill_length_mm", y ="bill_depth_mm",
                      data = df1, hue = "sex", style = "sex")
plt.show()
```



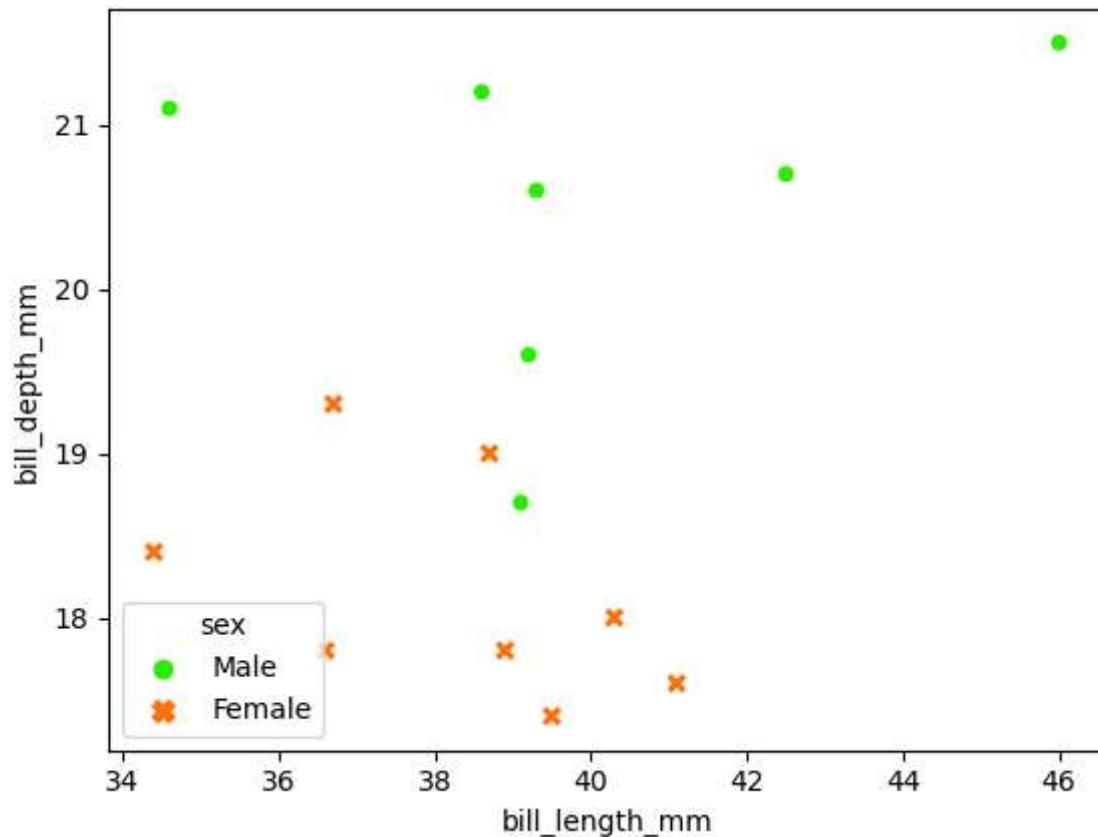
```
In [26]: sns.scatterplot(x="bill_length_mm", y ="bill_depth_mm",
                      data = df1, hue = "sex", style = "sex", size = "sex")
plt.show()
```



```
In [28]: sns.scatterplot(x="bill_length_mm", y ="bill_depth_mm",
                      data = df1, hue = "sex", style = "sex", size = "sex", sizes =(40, 100))
plt.show()
```

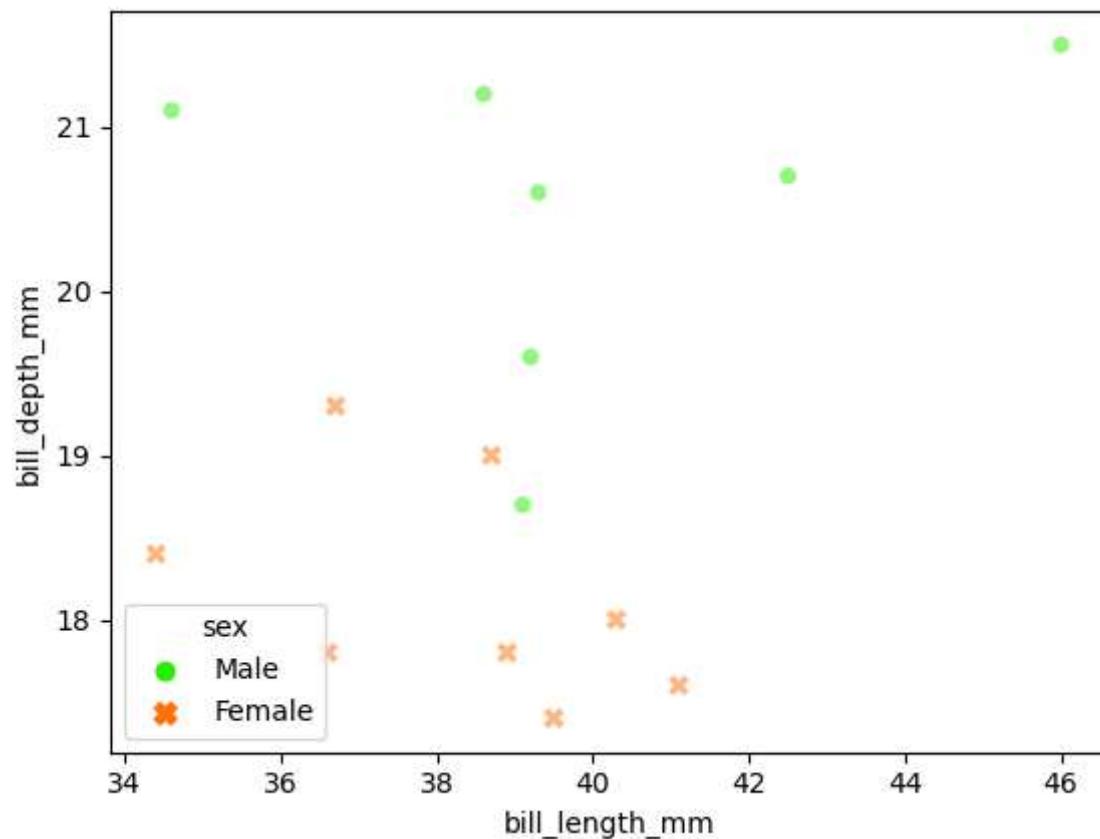


```
In [29]: sns.scatterplot(x="bill_length_mm", y ="bill_depth_mm",
                      data = df1, hue = "sex", style = "sex", size = "sex", sizes =(40, 100),
                      palette = "gist_ncar")
plt.show()
```

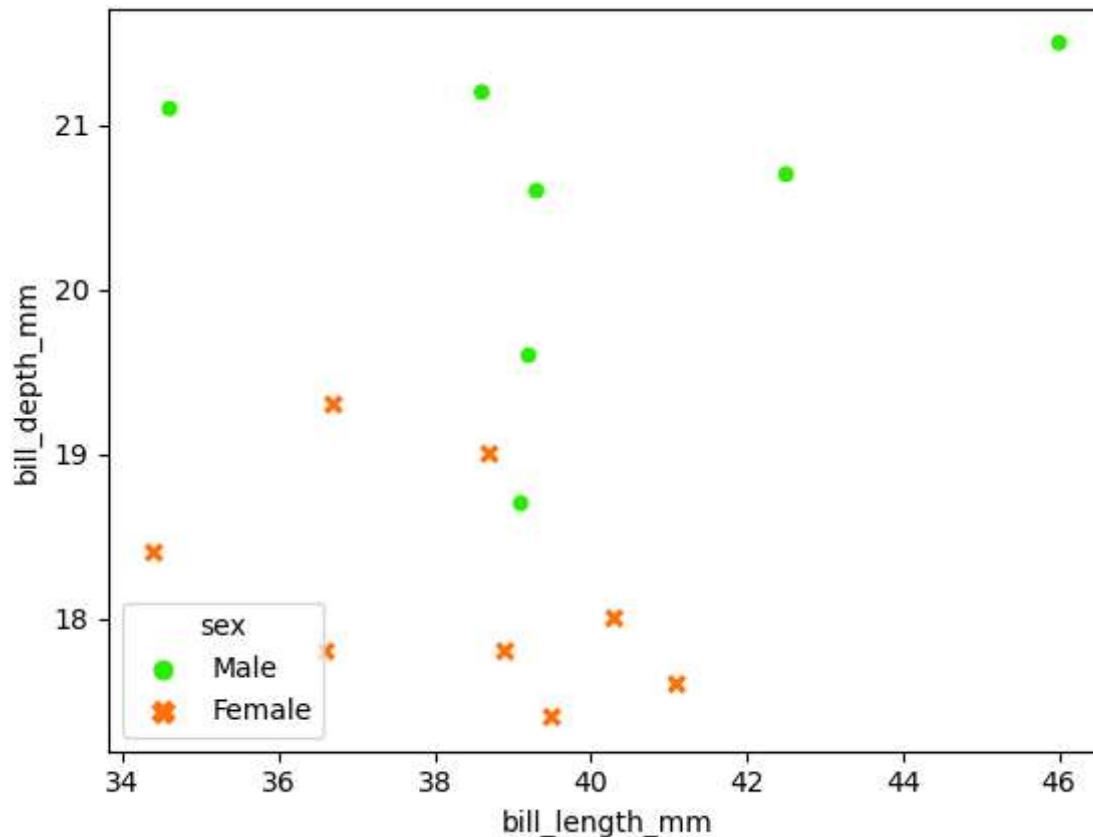


## alpha

```
In [30]: sns.scatterplot(x="bill_length_mm", y ="bill_depth_mm",
                      data = df1, hue = "sex", style = "sex", size = "sex", sizes =(40, 100),
                      palette = "gist_ncar", alpha = 0.5)
plt.show()
```

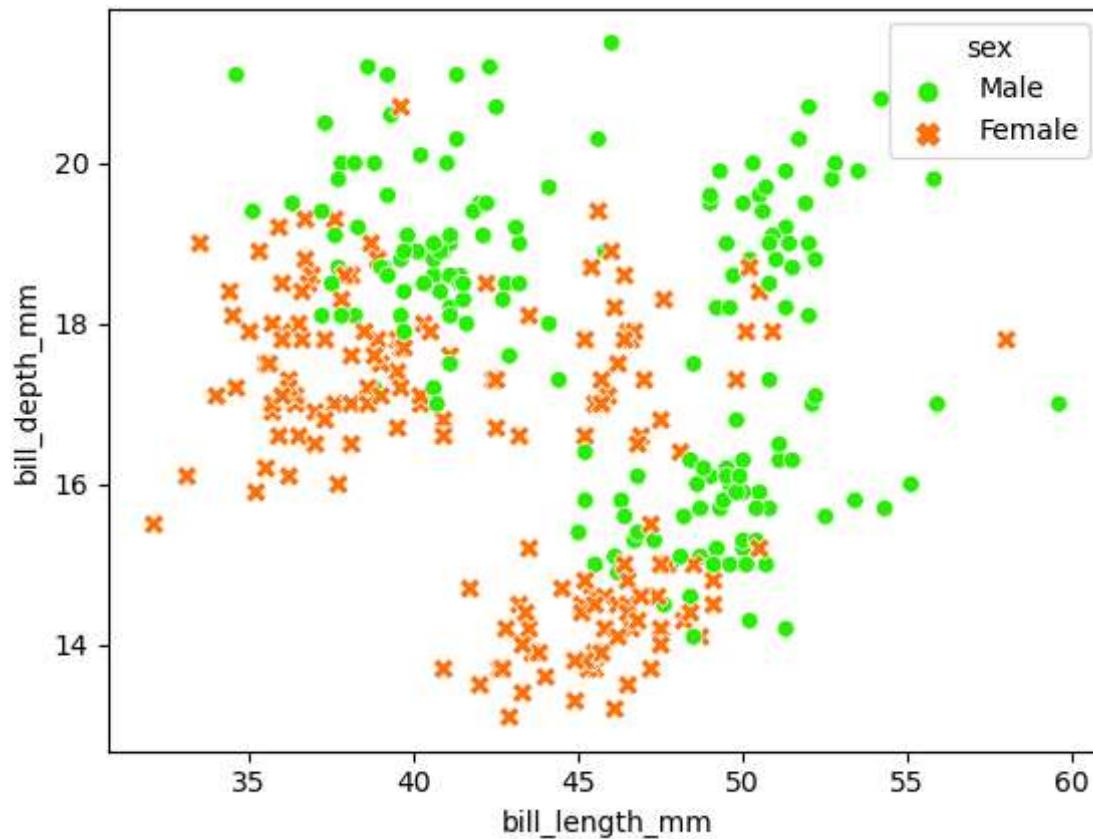


```
In [32]: m = {"Male":"+","Female":"o"}  
sns.scatterplot(x="bill_length_mm", y ="bill_depth_mm",  
                 data = df1, hue = "sex", style = "sex", size = "sex", sizes =(40, 100),  
                 palette = "gist_ncar")  
plt.show()
```



```
In [33]: df1 = sns.load_dataset("penguins")
```

```
In [34]: m = {"Male":"+","Female":"o"}  
sns.scatterplot(x="bill_length_mm", y ="bill_depth_mm",  
                 data = df1, hue = "sex", style = "sex", size = "sex", sizes =(40, 100),  
                 palette = "gist_ncar")  
plt.show()
```



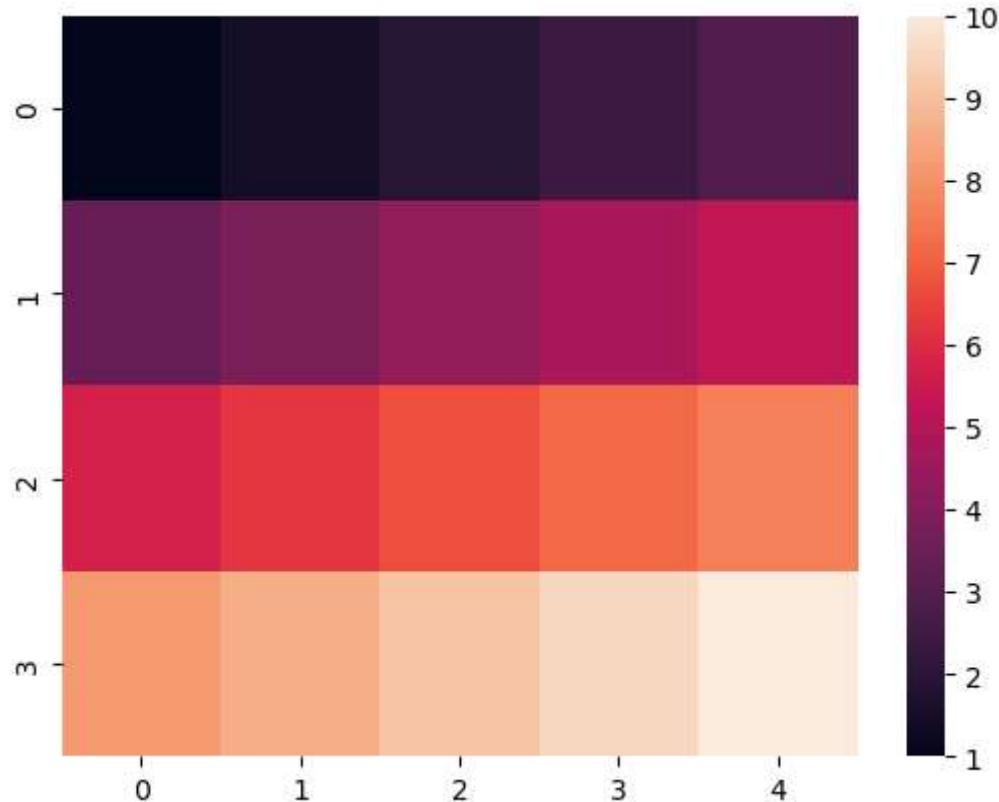
## Heatmap

```
In [2]: import seaborn as sns  
import matplotlib.pyplot as plt  
import pandas as pd  
import numpy as np
```

```
In [3]: var = np.linspace(1,10,20).reshape(4,5)  
var
```

```
Out[3]: array([[ 1.          ,  1.47368421,  1.94736842,  2.42105263,  2.89473684],  
                [ 3.36842105,  3.84210526,  4.31578947,  4.78947368,  5.26315789],  
                [ 5.73684211,  6.21052632,  6.68421053,  7.15789474,  7.63157895],  
                [ 8.10526316,  8.57894737,  9.05263158,  9.52631579,  10.        ]])
```

```
In [4]: sns.heatmap(var)
plt.show()
```



```
In [5]: data = sns.load_dataset("anagrams")
```

In [6]: data

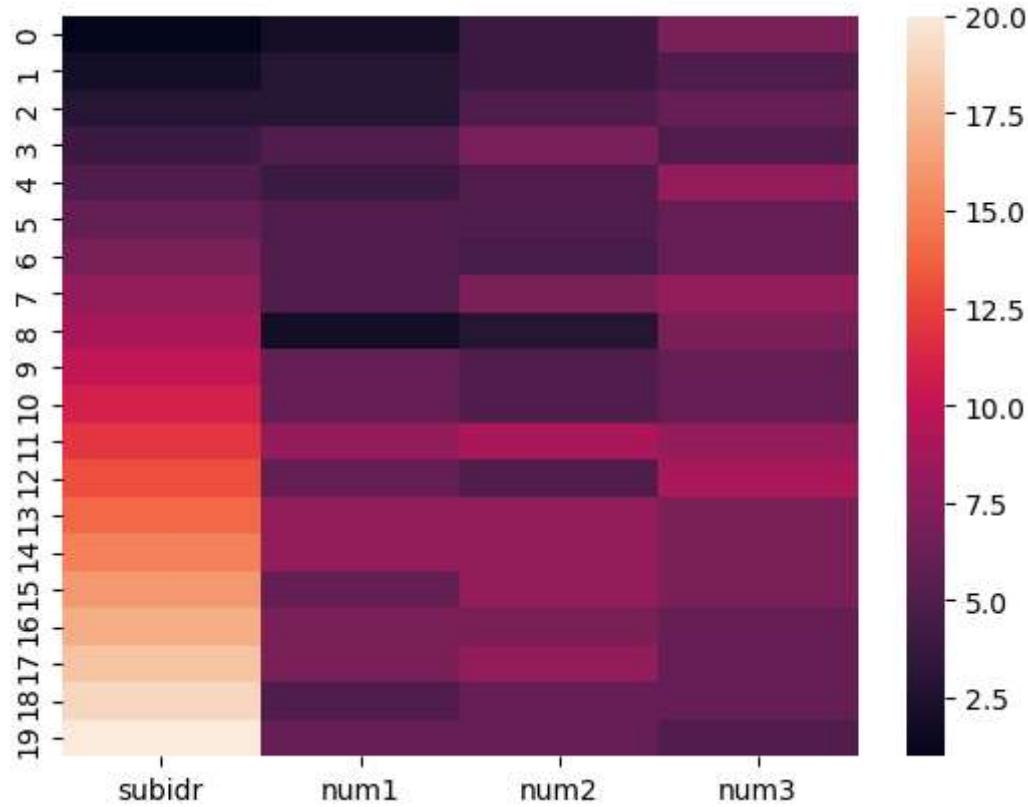
	subidr	attnr	num1	num2	num3
0	1	divided	2	4.0	7
1	2	divided	3	4.0	5
2	3	divided	3	5.0	6
3	4	divided	5	7.0	5
4	5	divided	4	5.0	8
5	6	divided	5	5.0	6
6	7	divided	5	4.5	6
7	8	divided	5	7.0	8
8	9	divided	2	3.0	7
9	10	divided	6	5.0	6
10	11	focused	6	5.0	6
11	12	focused	8	9.0	8
12	13	focused	6	5.0	9
13	14	focused	8	8.0	7
14	15	focused	8	8.0	7
15	16	focused	6	8.0	7
16	17	focused	7	7.0	6
17	18	focused	7	8.0	6
18	19	focused	5	6.0	6
19	20	focused	6	6.0	5

```
In [7]: data = sns.load_dataset("anagrams")
x = data.drop(columns=["attnr"],axis = 1)
x
```

Out[7]:

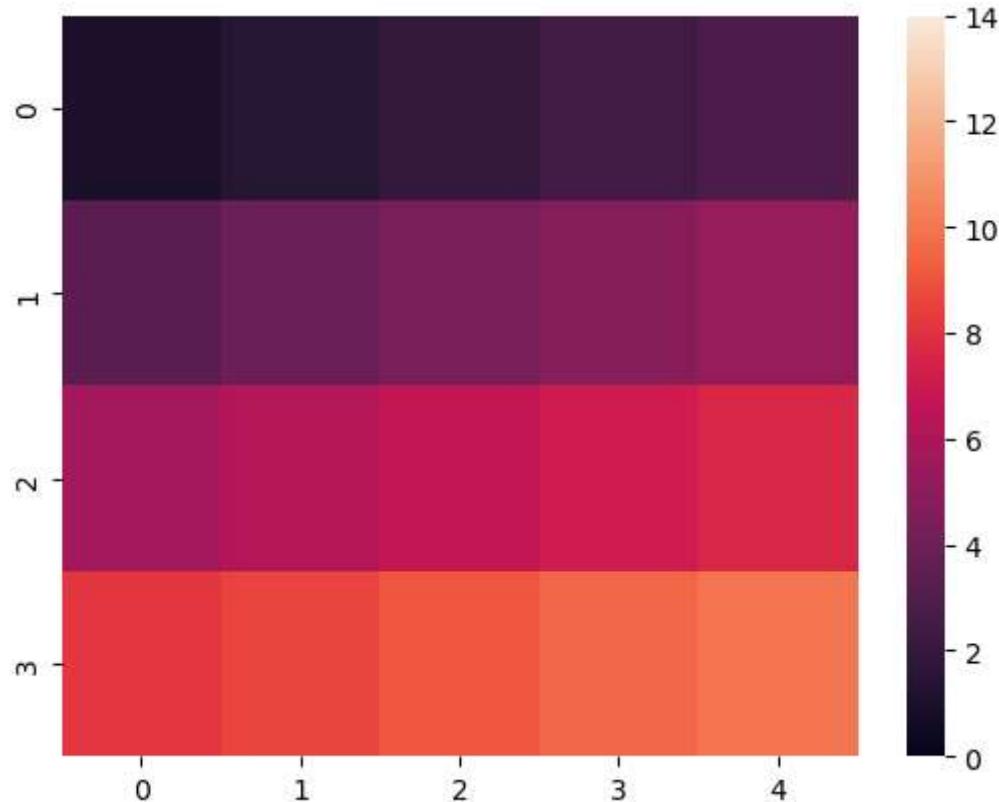
	subidr	num1	num2	num3
0	1	2	4.0	7
1	2	3	4.0	5
2	3	3	5.0	6
3	4	5	7.0	5
4	5	4	5.0	8
5	6	5	5.0	6
6	7	5	4.5	6
7	8	5	7.0	8
8	9	2	3.0	7
9	10	6	5.0	6
10	11	6	5.0	6
11	12	8	9.0	8
12	13	6	5.0	9
13	14	8	8.0	7
14	15	8	8.0	7
15	16	6	8.0	7
16	17	7	7.0	6
17	18	7	8.0	6
18	19	5	6.0	6
19	20	6	6.0	5

```
In [8]: sns.heatmap(x)
plt.show()
```

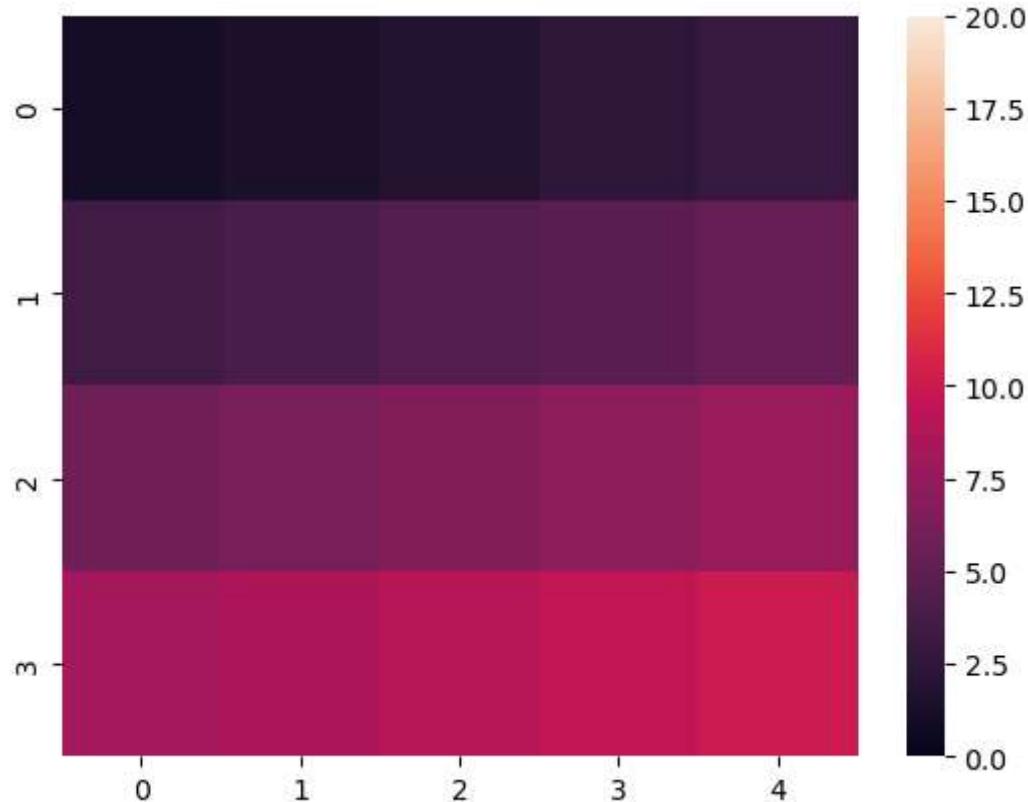


## expanding the range of a heat map

```
In [9]: sns.heatmap(var, vmin=0, vmax=14)  
plt.show()
```

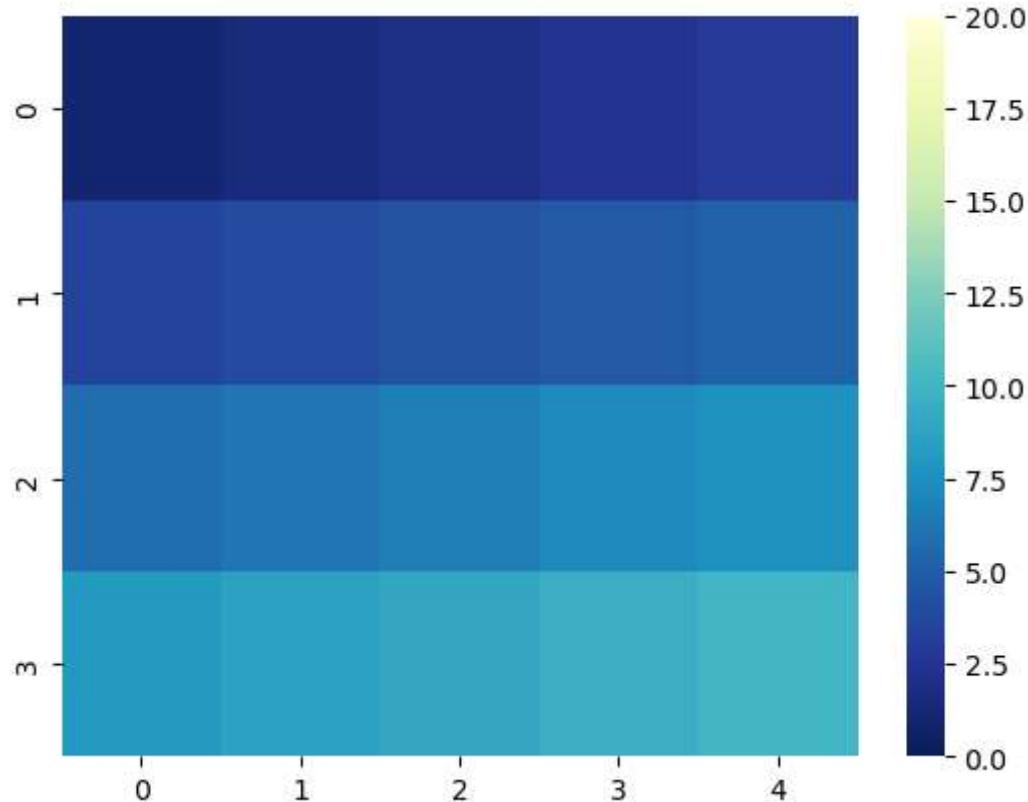


```
In [10]: sns.heatmap(var, vmin=0, vmax=20)  
plt.show()
```

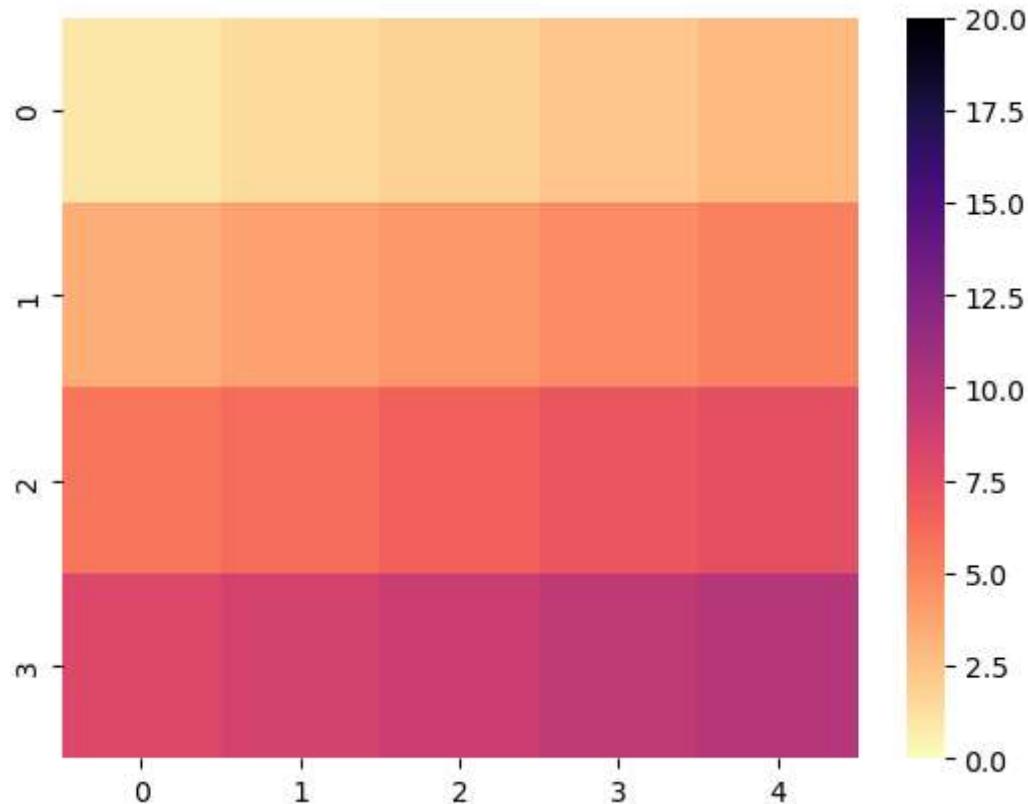


## changing the color of a heatmap

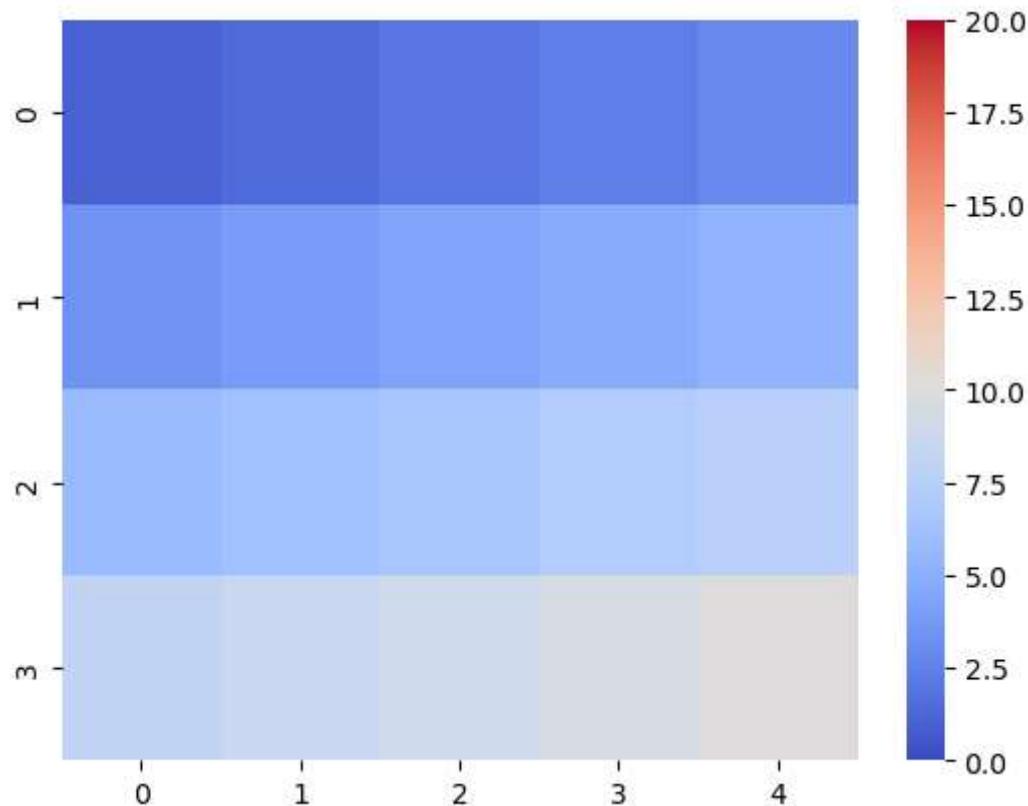
```
In [11]: sns.heatmap(var, vmin=0, vmax=20, cmap = "YlGnBu_r")
plt.show()
```



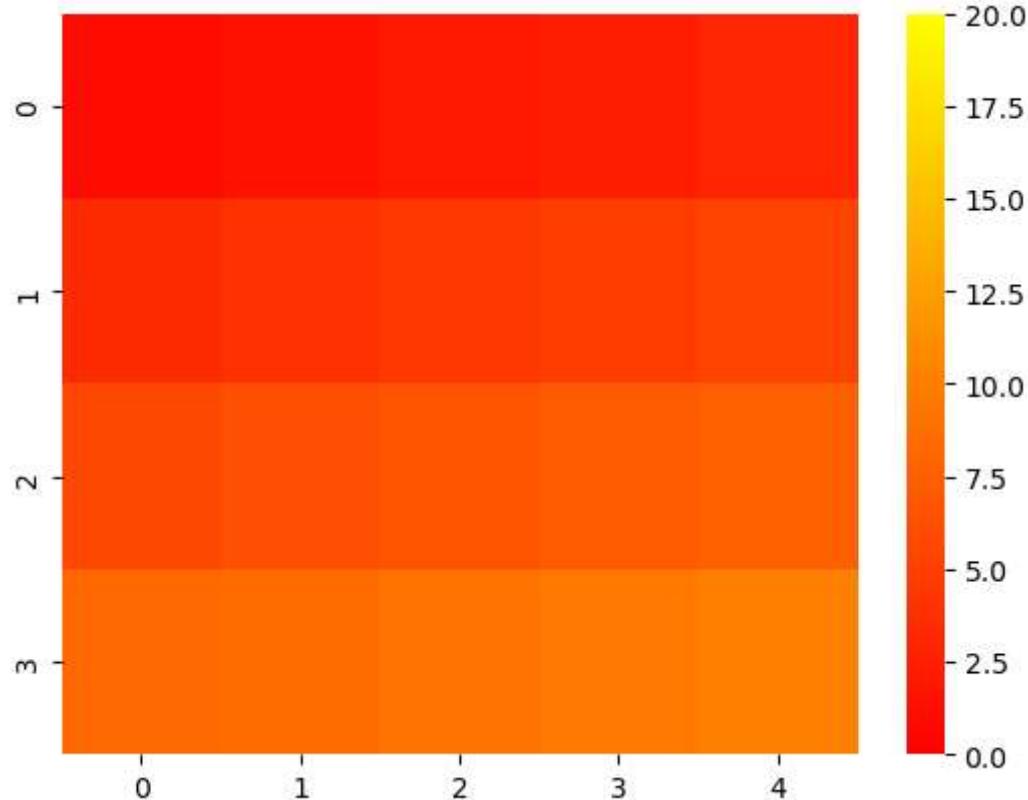
```
In [12]: sns.heatmap(var, vmin=0, vmax=20, cmap = "magma_r")
plt.show()
```



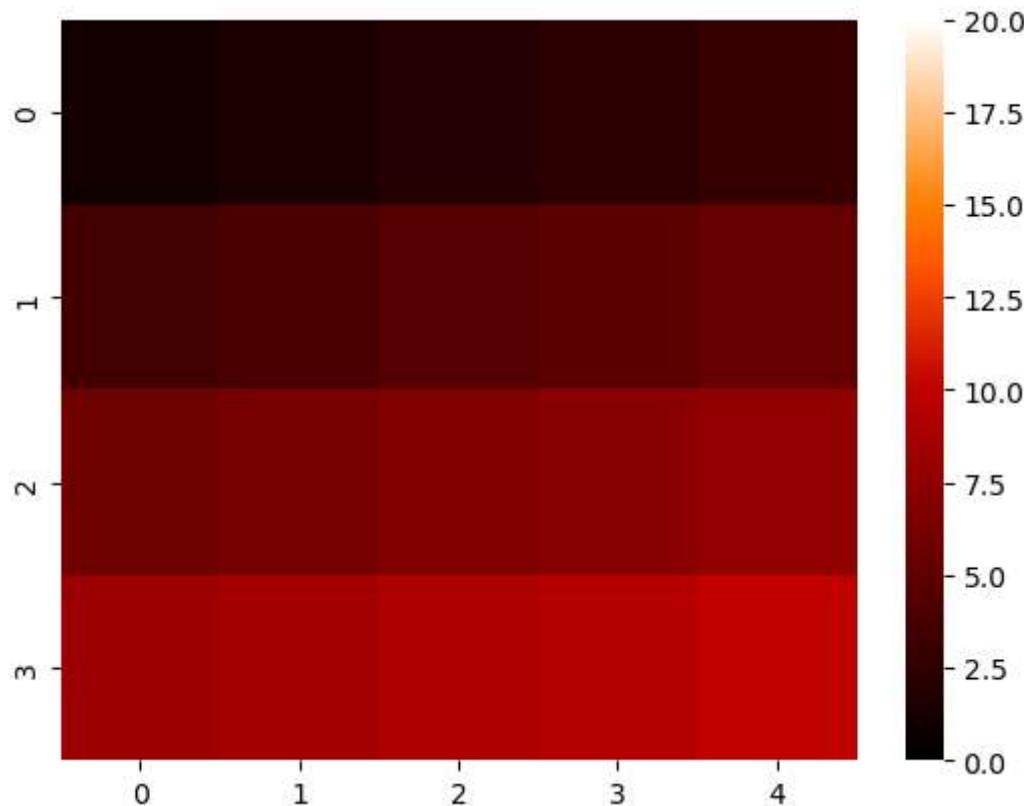
```
In [13]: sns.heatmap(var, vmin=0, vmax=20, cmap = "coolwarm")
plt.show()
```



```
In [14]: sns.heatmap(var, vmin=0, vmax=20, cmap = "autumn")
plt.show()
```



```
In [15]: sns.heatmap(var, vmin=0, vmax=20, cmap = "gist_heat")
plt.show()
```

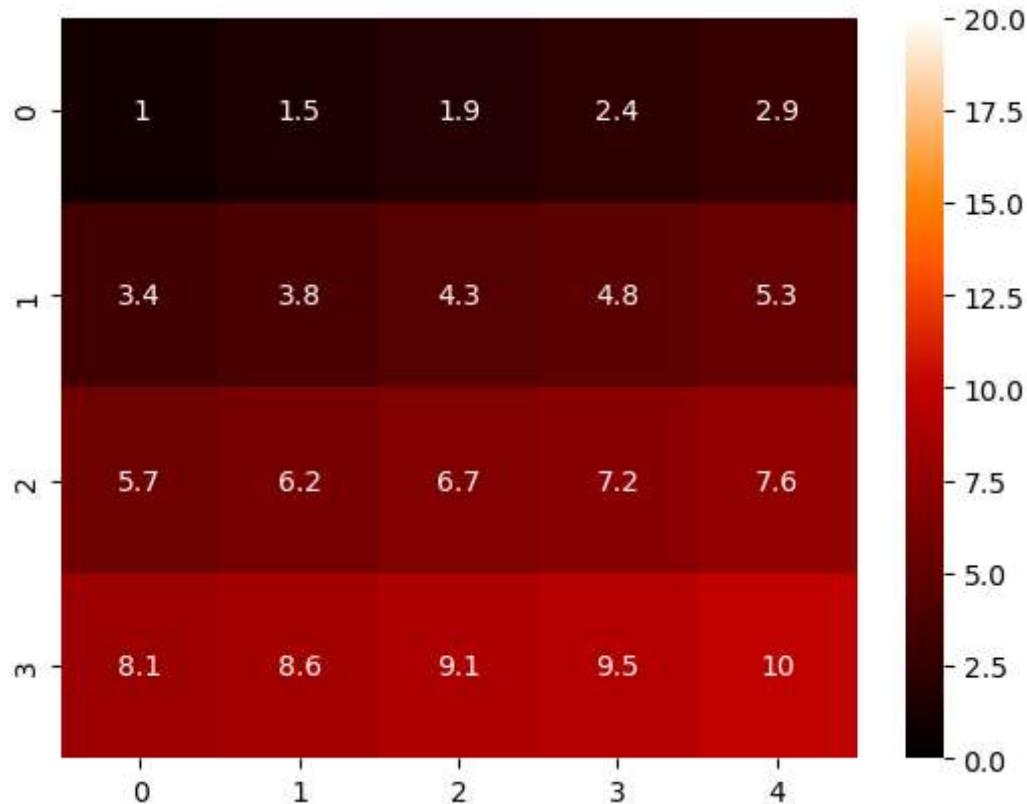


## showing values in the heat map

```
In [16]: var = np.linspace(1,10,20).reshape(4,5)
var
```

```
Out[16]: array([[ 1.          ,  1.47368421,  1.94736842,  2.42105263,  2.89473684],
   [ 3.36842105,  3.84210526,  4.31578947,  4.78947368,  5.26315789],
   [ 5.73684211,  6.21052632,  6.68421053,  7.15789474,  7.63157895],
   [ 8.10526316,  8.57894737,  9.05263158,  9.52631579, 10.        ]])
```

```
In [17]: sns.heatmap(var, vmin=0, vmax=20, cmap = "gist_heat", annot = True)
plt.show()
```



## naming the heatmap as per your wish

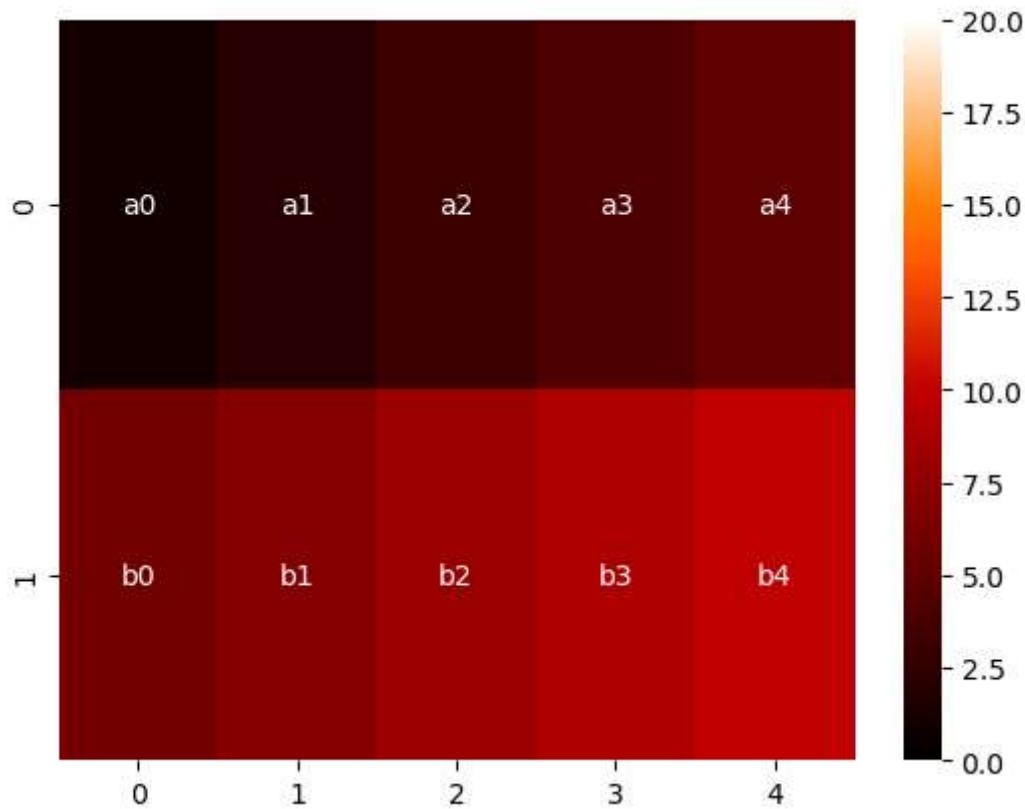
```
In [18]: var = np.linspace(1,10,10).reshape(2,5)
var
```

```
Out[18]: array([[ 1.,  2.,  3.,  4.,  5.],
   [ 6.,  7.,  8.,  9., 10.]])
```

```
In [19]: ar = np.array([["a0", "a1","a2", "a3","a4"],["b0","b1","b2","b3","b4"]])
ar
```

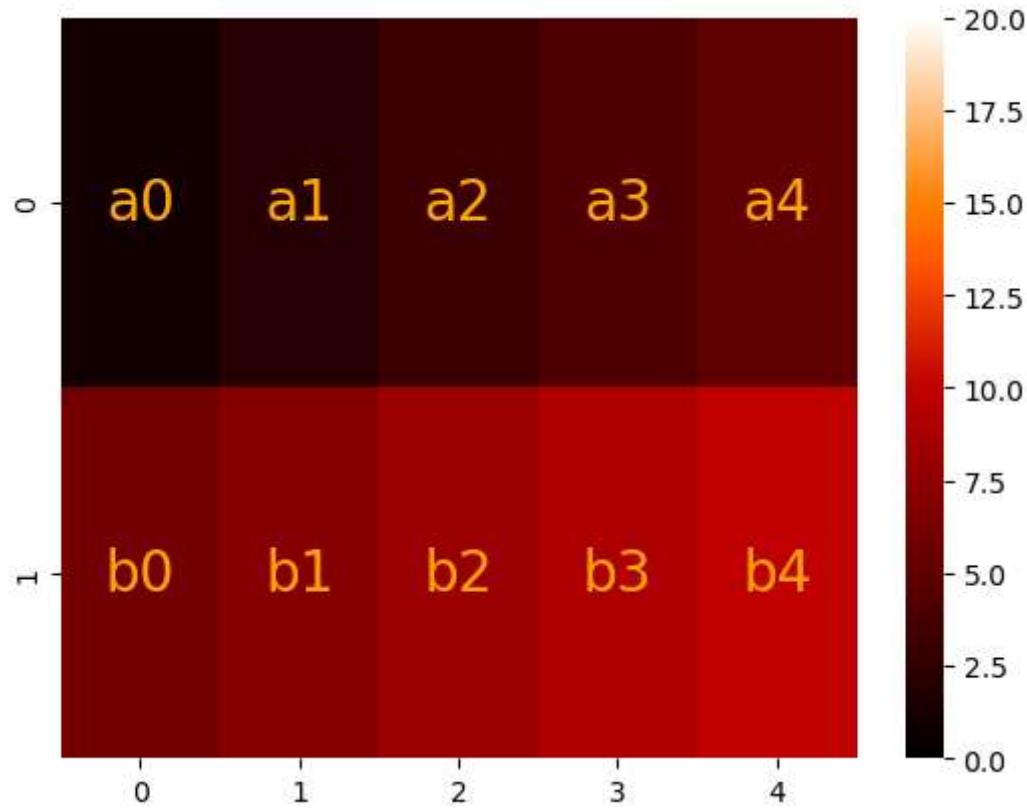
```
Out[19]: array([['a0', 'a1', 'a2', 'a3', 'a4'],
   ['b0', 'b1', 'b2', 'b3', 'b4']], dtype='|U2')
```

```
In [20]: sns.heatmap(var, vmin=0, vmax=20, cmap = "gist_heat", annot = ar, fmt ="s")
plt.show()
```



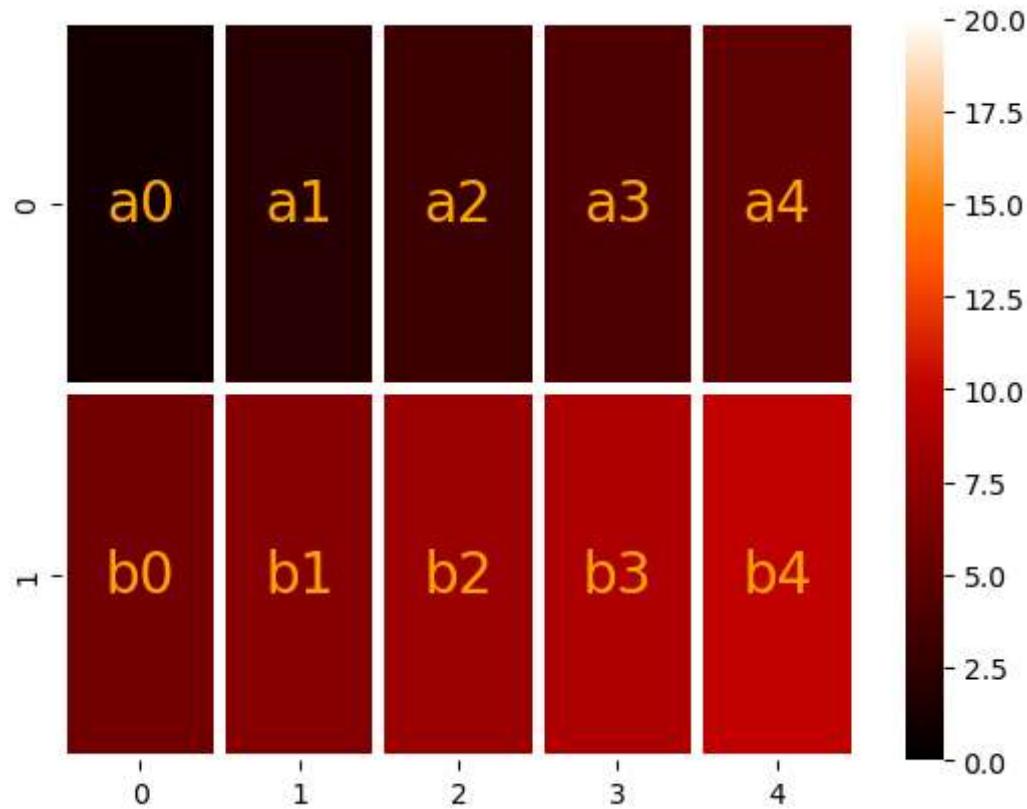
changing he color and size of the text inside the heatmap

```
In [21]: y = {"fontsize":20,"color":"orange"}  
sns.heatmap(var, vmin=0, vmax=20, cmap = "gist_heat", annot = ar, fmt ="s", an  
plt.show()
```

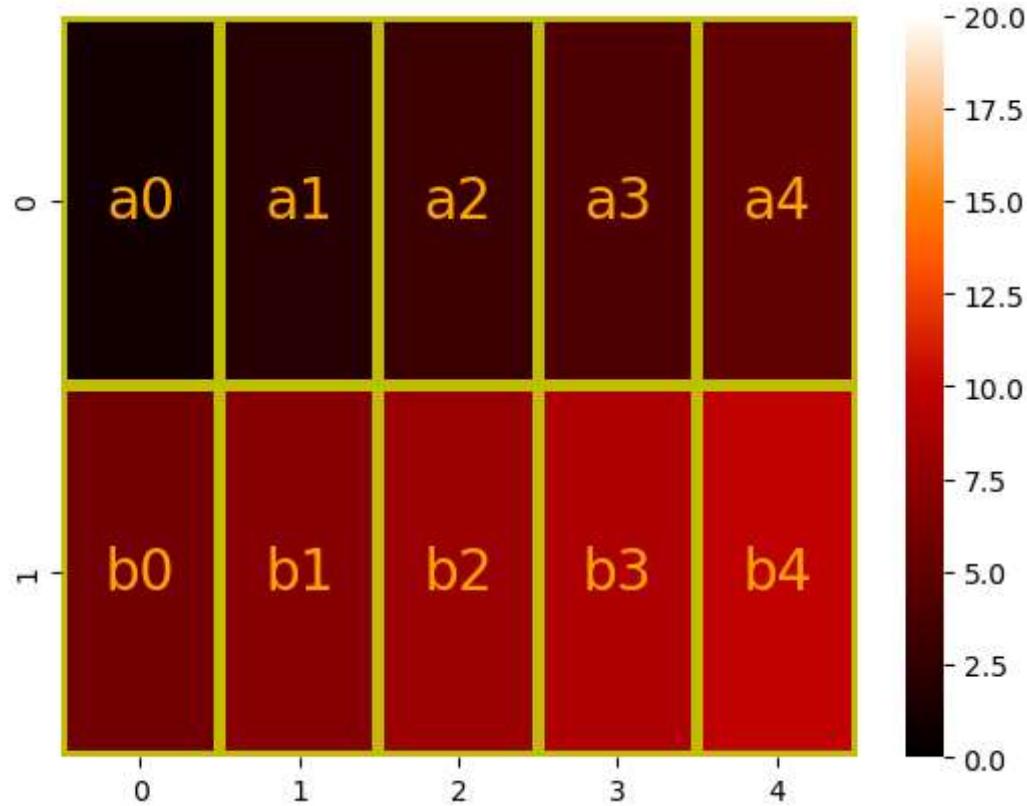


## linewidth

```
In [22]: y = {"fontsize":20,"color":"orange"}  
sns.heatmap(var, vmin=0, vmax=20, cmap = "gist_heat", annot = ar, fmt ="s",  
            annot_kws=y, linewidth= 4)  
plt.show()
```

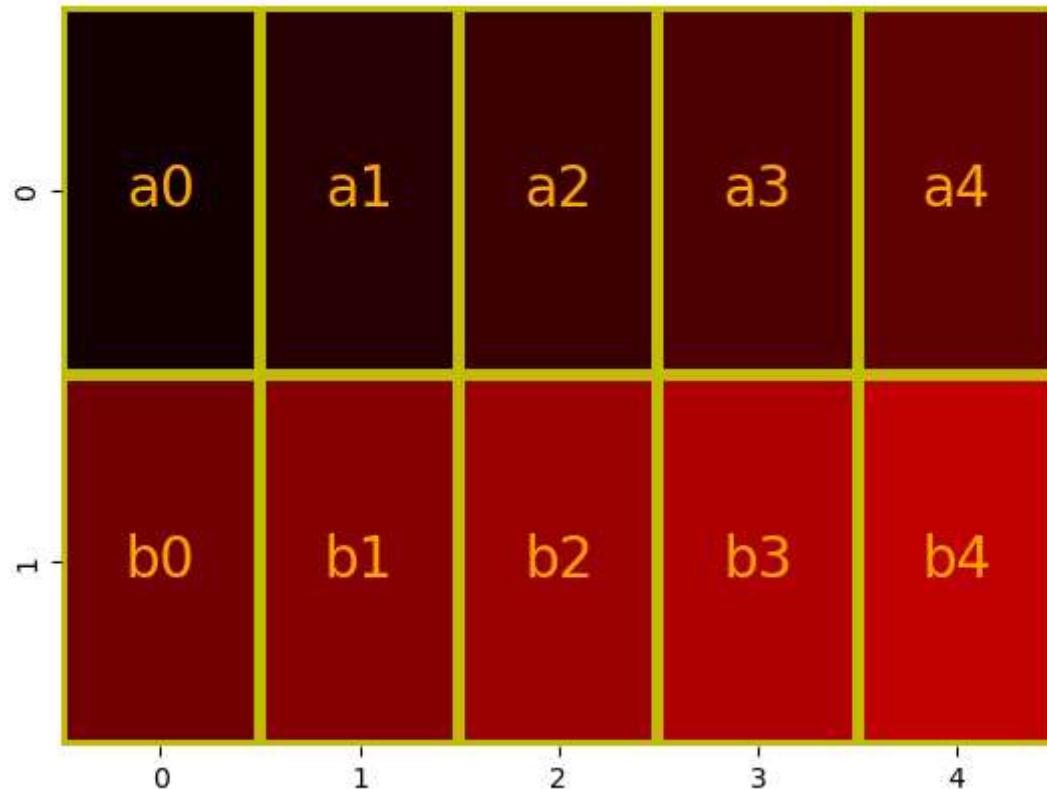


```
In [23]: y = {"fontsize":20,"color":"orange"}  
sns.heatmap(var, vmin=0, vmax=20, cmap = "gist_heat", annot = ar, fmt ="s",  
            annot_kws=y, linewidth= 4, linecolor="y")  
plt.show()
```



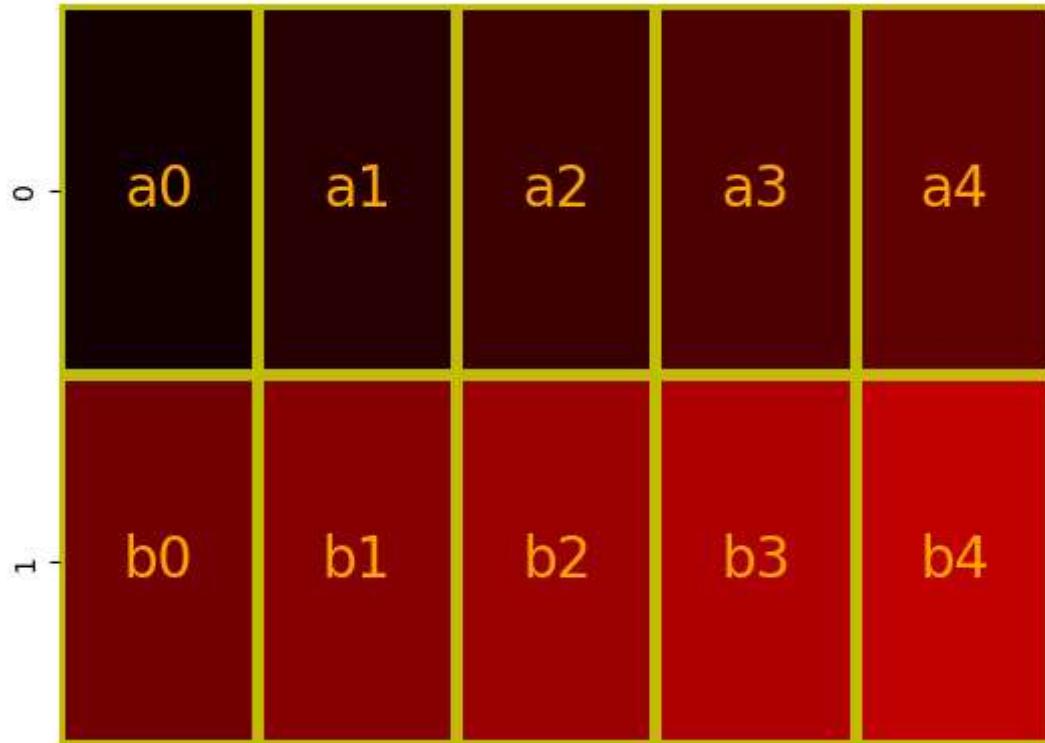
removing color bar

```
In [24]: y = {"fontsize":20,"color":"orange"}  
sns.heatmap(var, vmin=0, vmax=20, cmap = "gist_heat", annot = ar, fmt ="s",  
            annot_kws=y, linewidth= 4, linecolor="y", cbar= False)  
plt.show()
```

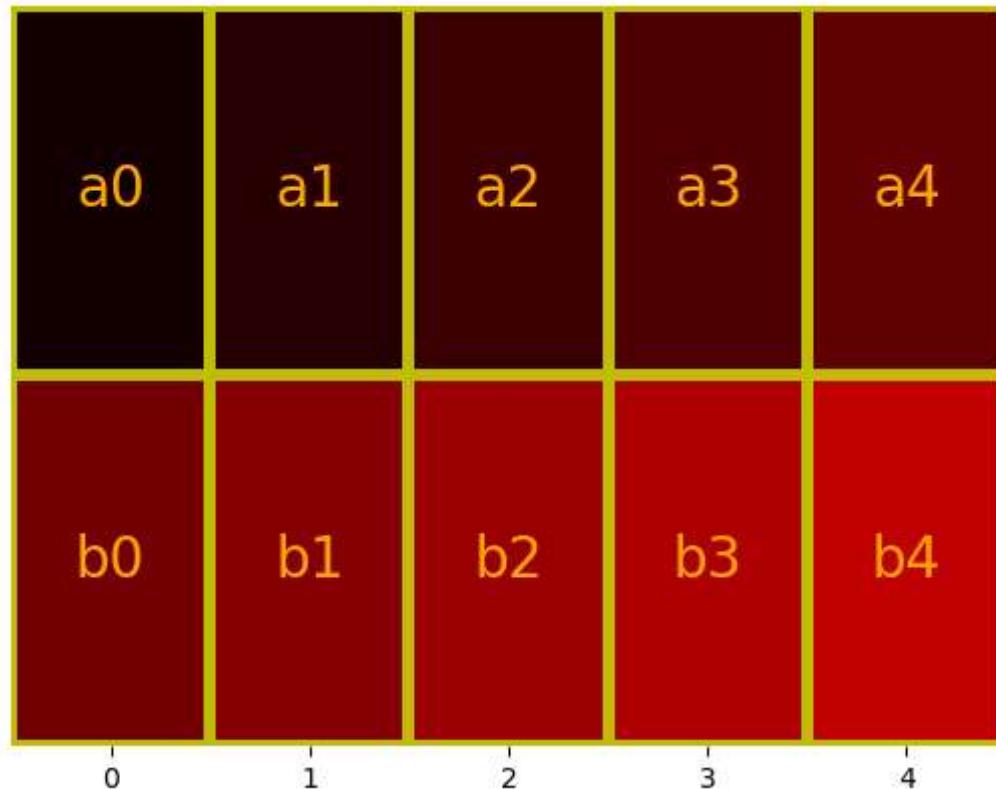


removing the labels

```
In [25]: y = {"fontsize":20,"color":"orange"}  
sns.heatmap(var, vmin=0, vmax=20, cmap = "gist_heat", annot = ar, fmt ="s",  
            annot_kws=y, linewidth= 4, linecolor="y", cbar= False, xticklabels=  
            plt.show()
```

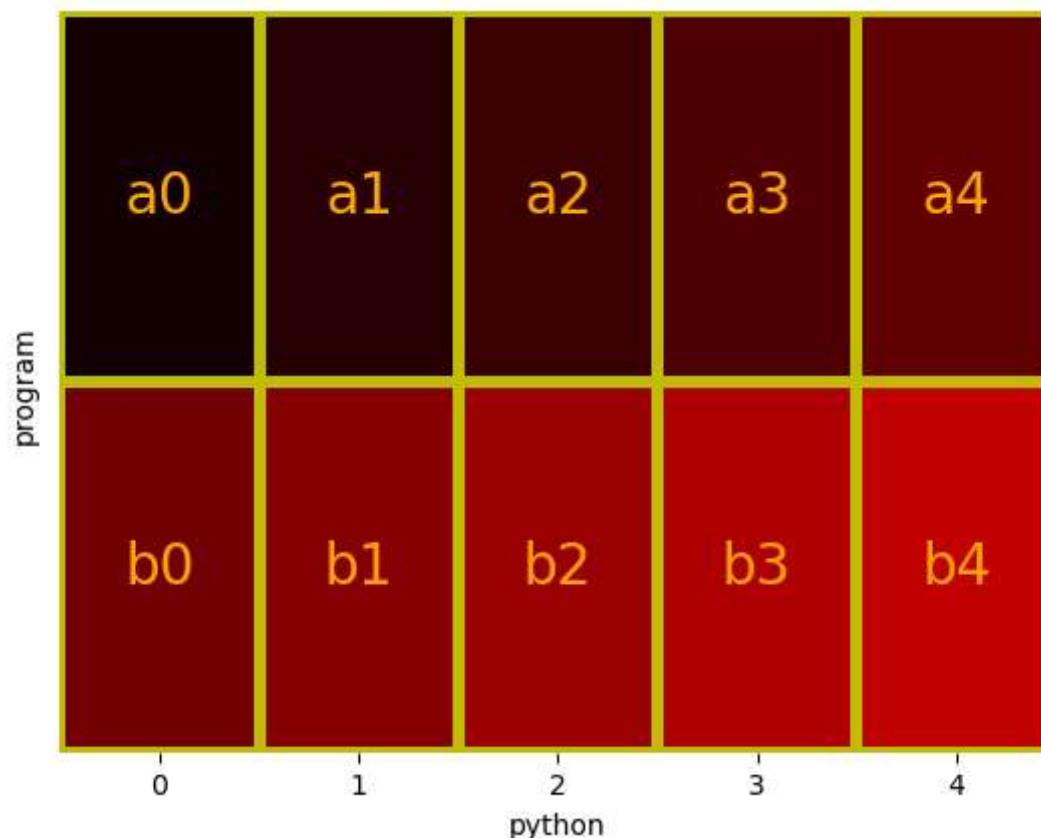


```
In [26]: y = {"fontsize":20,"color":"orange"}  
sns.heatmap(var, vmin=0, vmax=20, cmap = "gist_heat", annot = ar, fmt ="s",  
            annot_kws=y, linewidth= 4, linecolor="y", cbar= False, yticklabels=  
            plt.show()
```



setting the label

```
In [28]: y = {"fontsize":20,"color":"orange"}  
v = sns.heatmap(var, vmin=0, vmax=20, cmap = "gist_heat", annot = ar, fmt ="s"  
                 annot_kws=y, linewidth= 4, linecolor="y", cbar= False, yticklabels=  
v.set(xlabel = "python", ylabel= "program")  
plt.show()
```



## Count Plot

```
In [29]: import matplotlib.pyplot as plt  
import pandas as pd  
import seaborn as sns
```

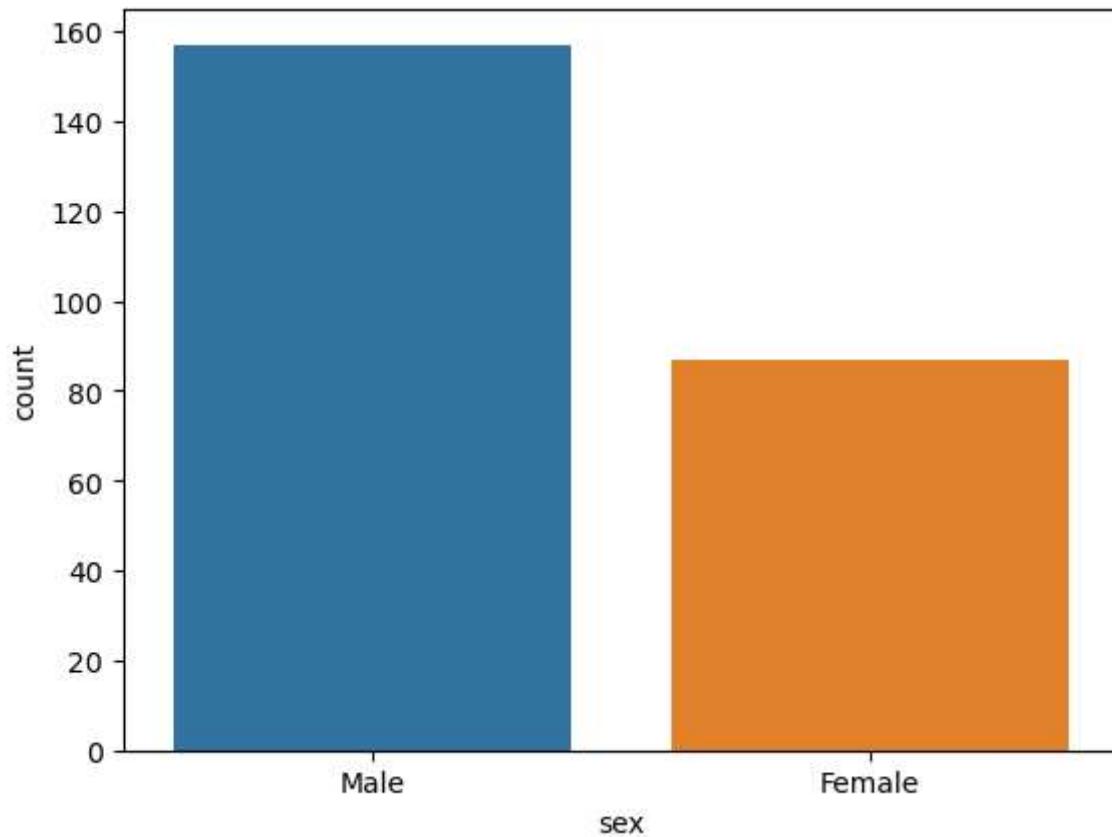
```
In [30]: df = sns.load_dataset("tips")
```

In [31]: df

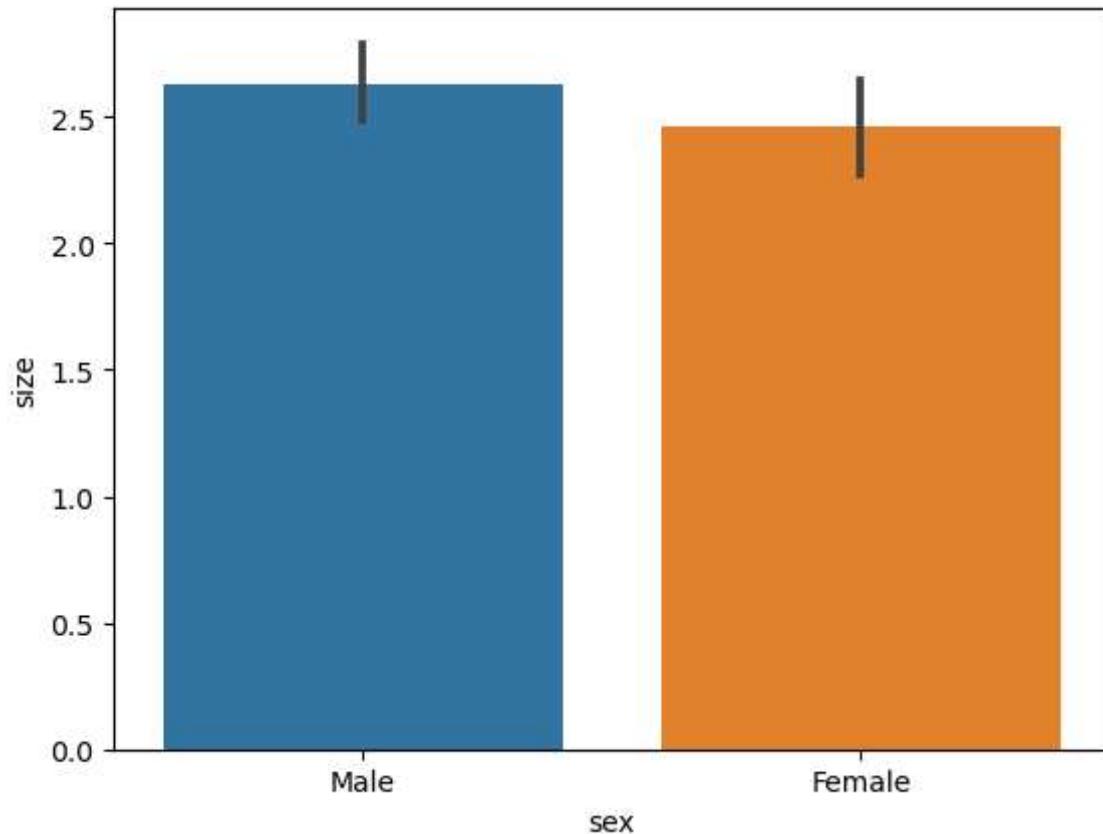
Out[31]:

	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

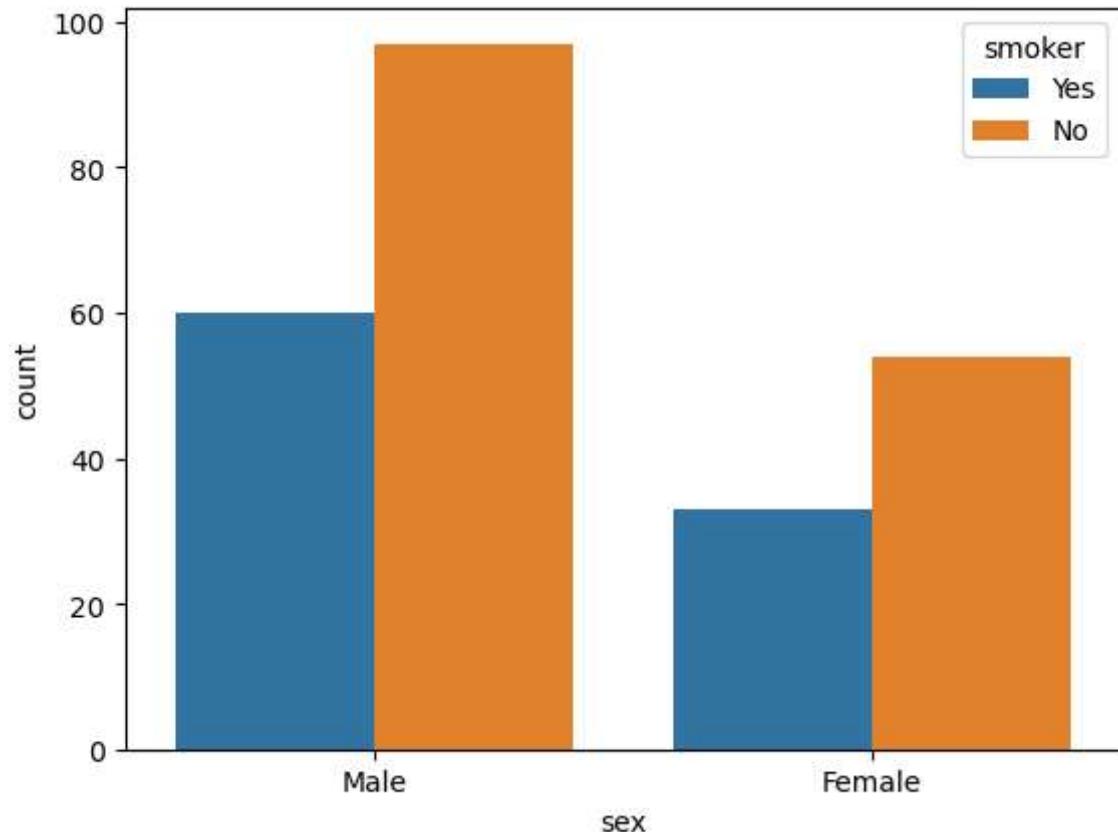
244 rows × 7 columns

In [33]: sns.countplot(x ="sex", data = df)  
plt.show()

```
In [37]: sns.barplot(x="sex", y="size", data = df)  
plt.show()
```

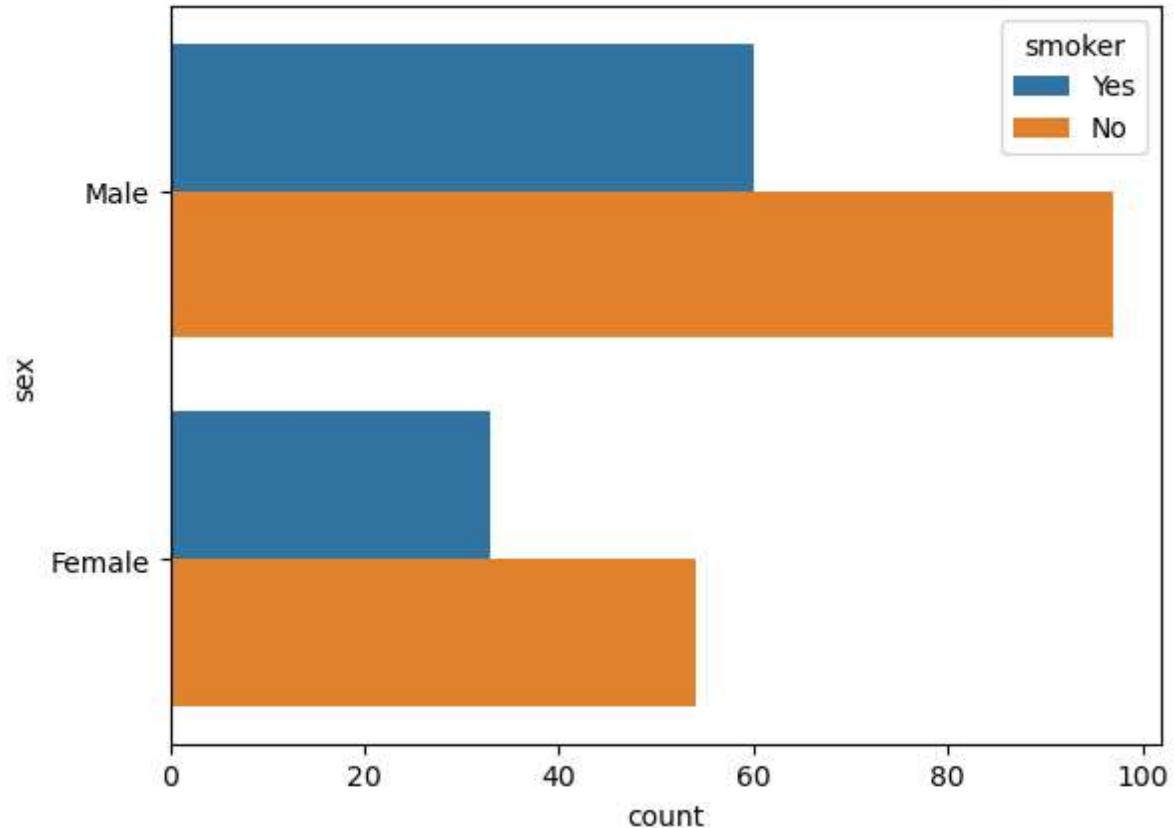


```
In [38]: sns.countplot(x ="sex", data = df, hue = "smoker")
plt.show()
```



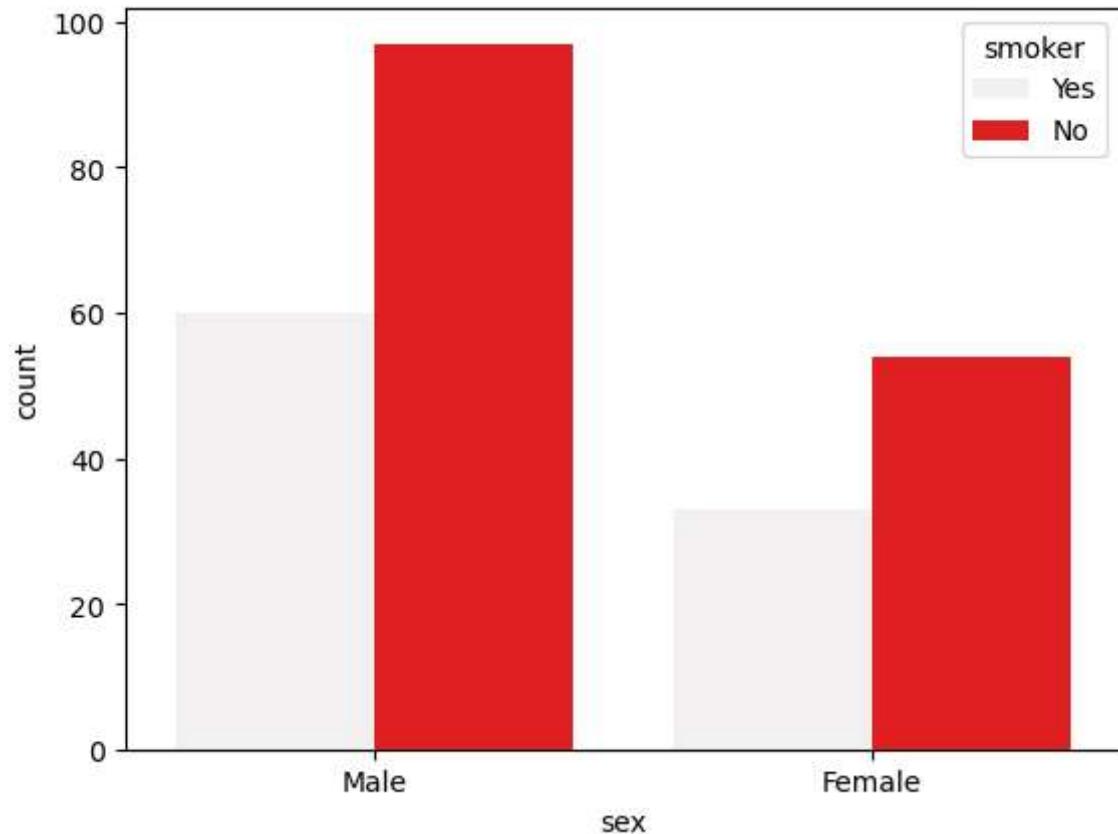
changing the orientation

```
In [39]: sns.countplot(y ="sex", data = df, hue = "smoker")
plt.show()
```

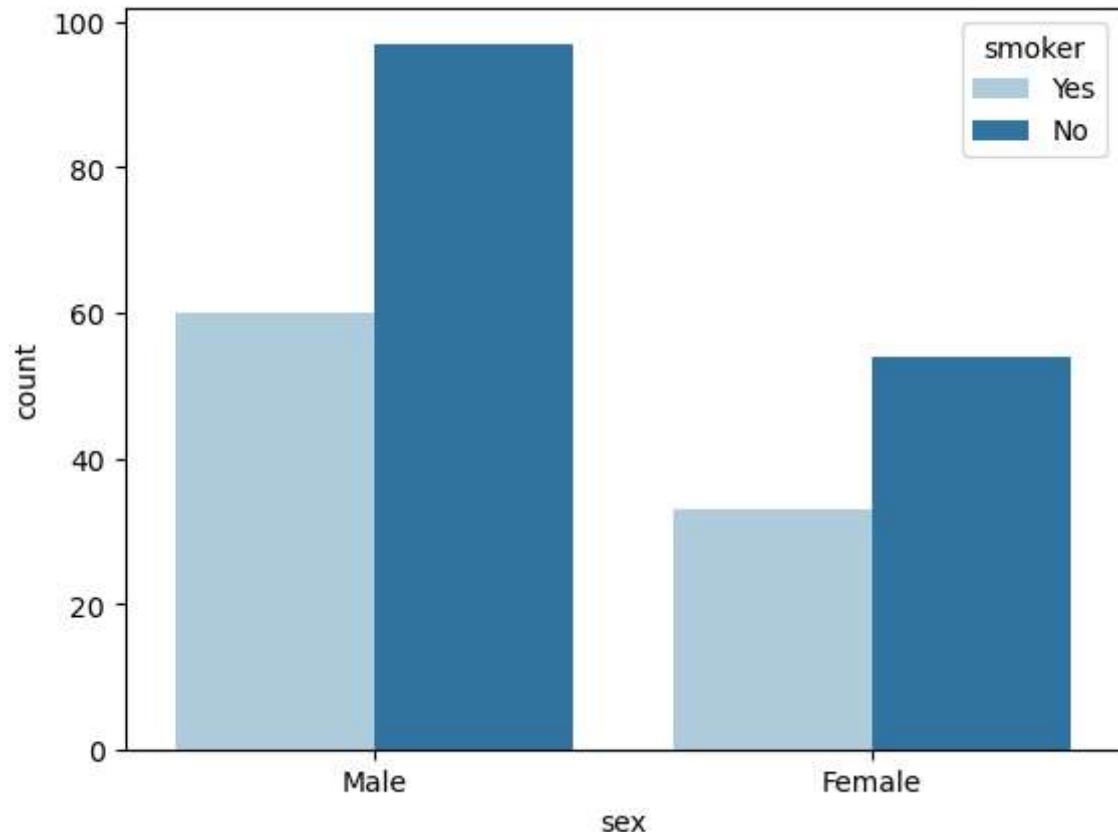


## changing the color

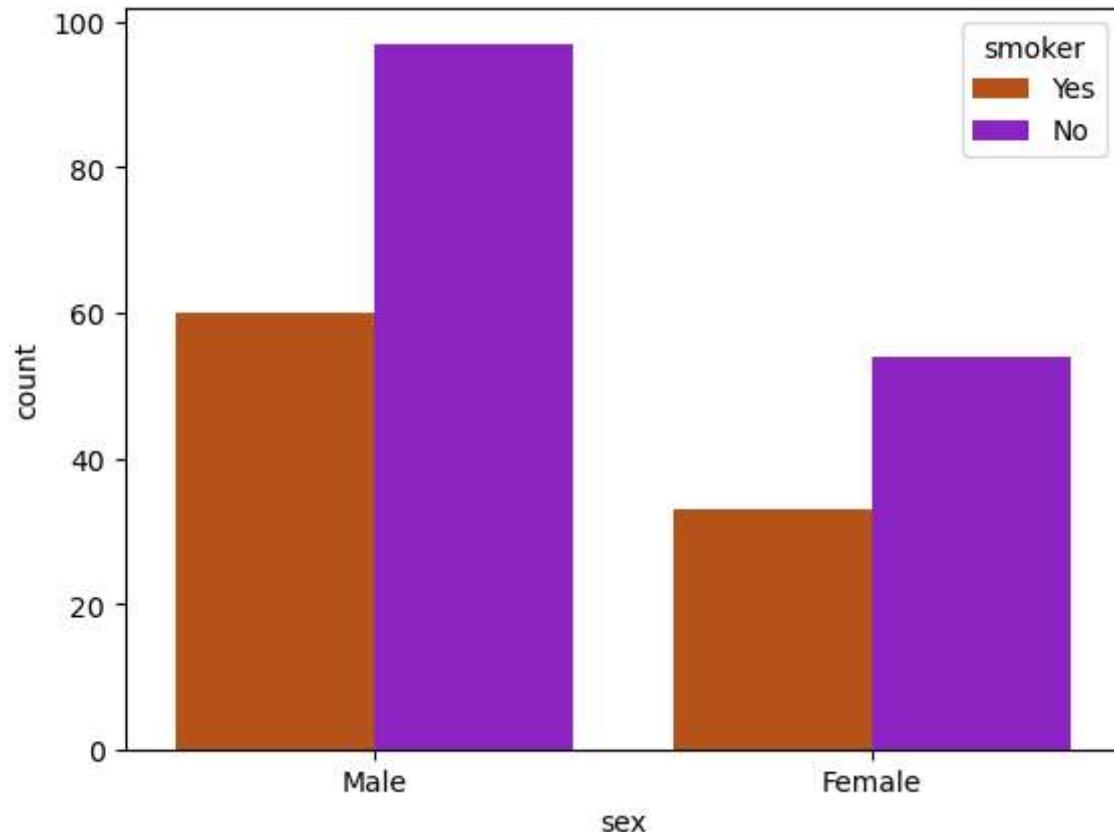
```
In [40]: sns.countplot(x ="sex", data = df, hue = "smoker", color = "r")
plt.show()
```



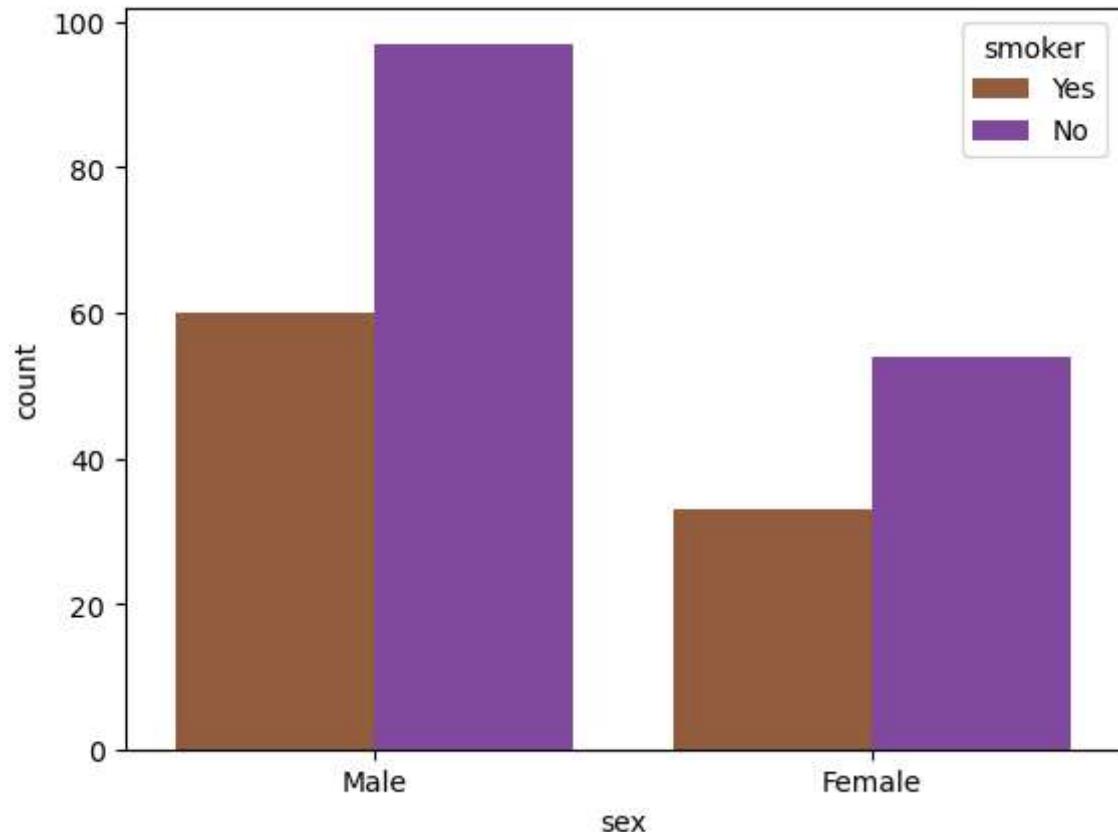
```
In [42]: sns.countplot(x ="sex", data = df, hue = "smoker", palette = "Paired")
plt.show()
```



```
In [44]: sns.countplot(x ="sex", data = df, hue = "smoker", palette = "gnuplot_r")
plt.show()
```



```
In [45]: sns.countplot(x ="sex", data = df, hue = "smoker", palette = "gnuplot_r", saturation = 0.8)
```



## Violin in Seaborn

```
In [47]: import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
```

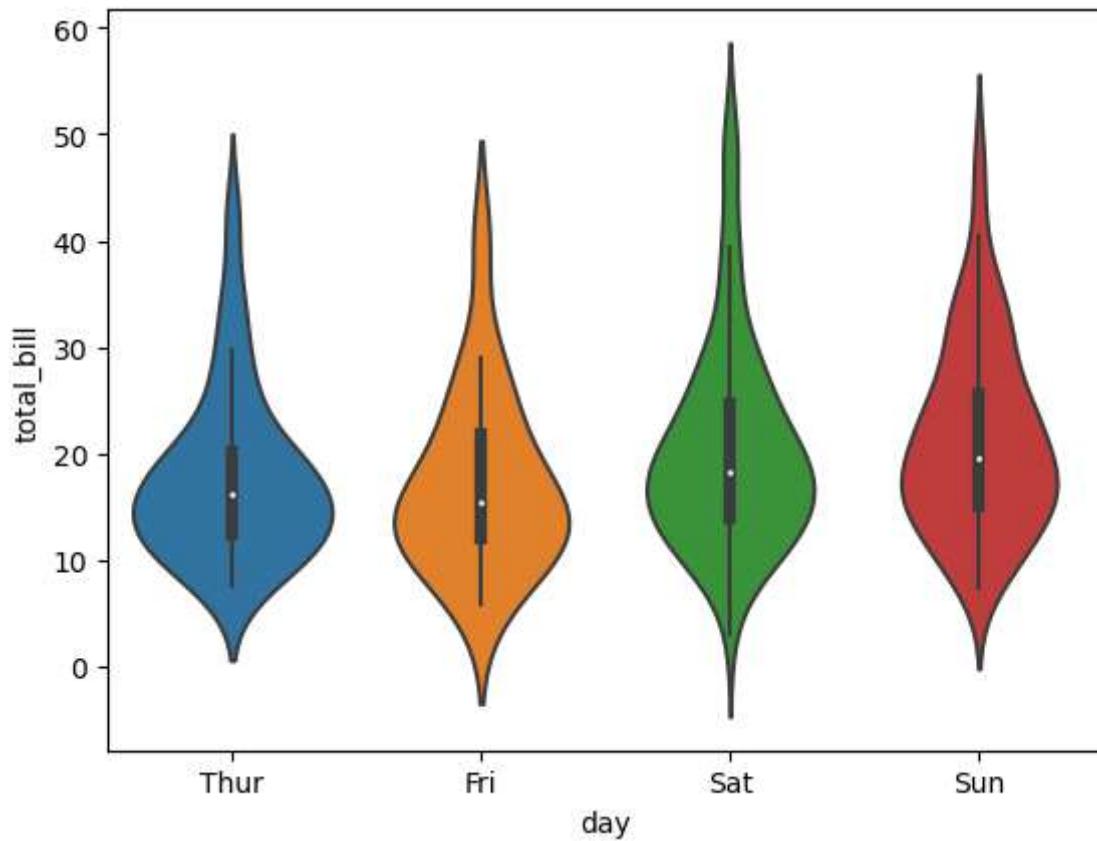
```
In [49]: df = sns.load_dataset("tips")
```

In [50]: df

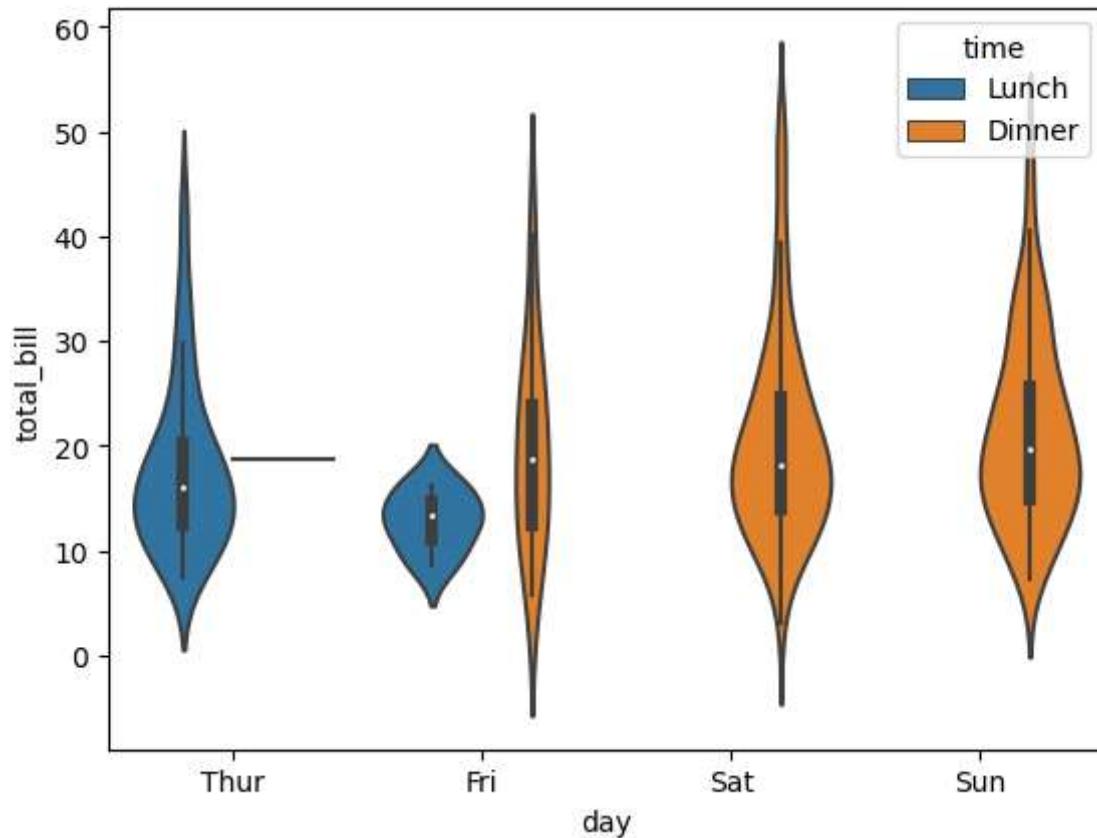
Out[50]:

	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

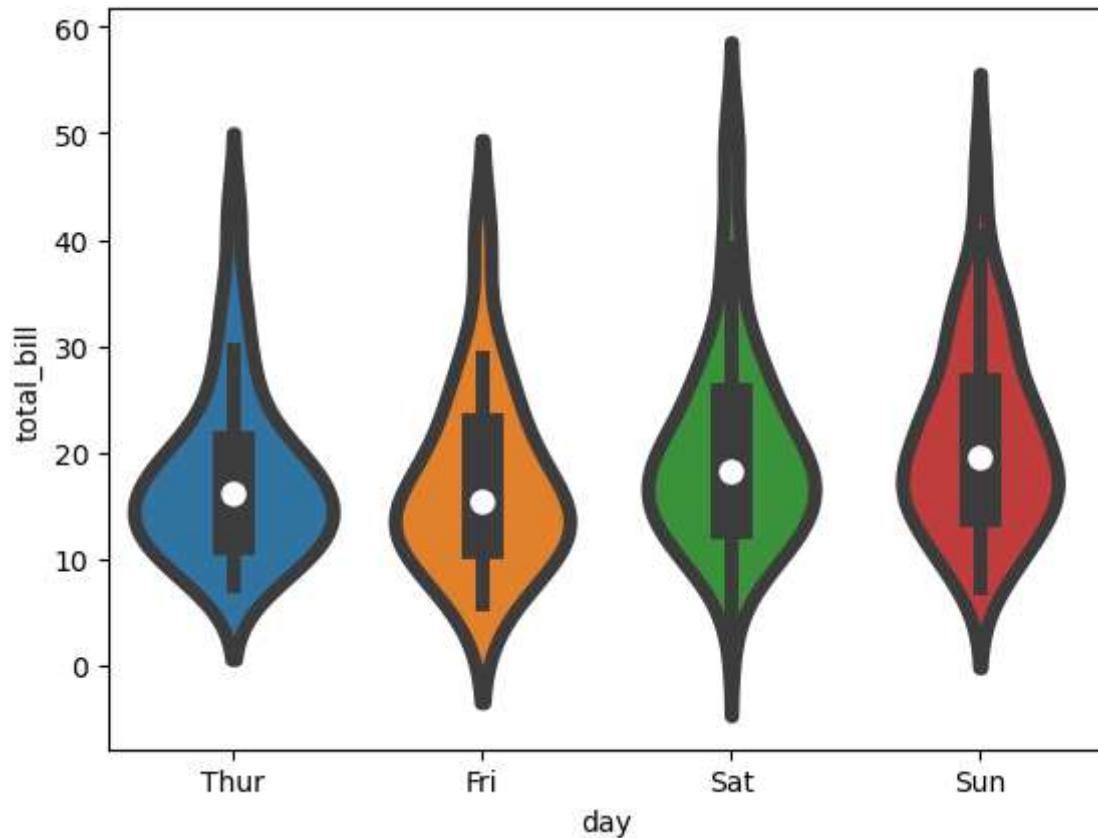
In [53]: sns.violinplot(x="day", y="total\_bill", data = df)  
plt.show()

```
In [54]: sns.violinplot(x="day", y = "total_bill", data = df, hue ="time")
plt.show()
```



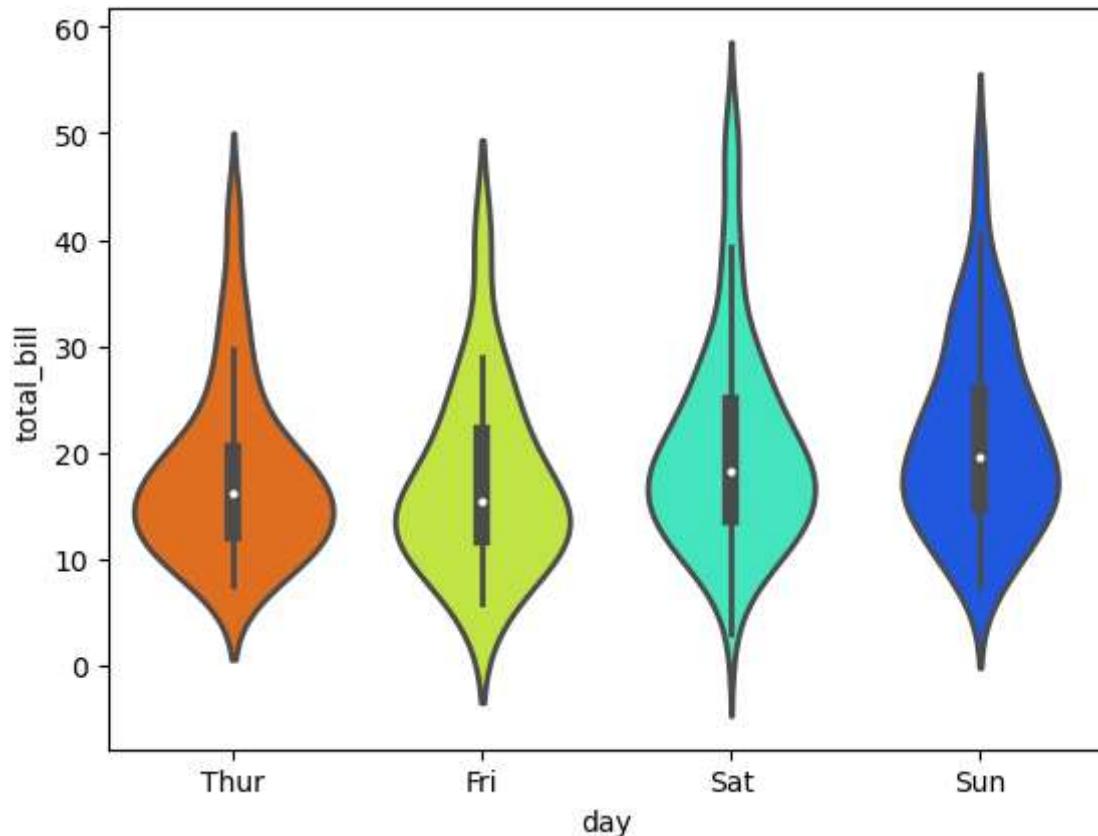
## line thickness

```
In [55]: sns.violinplot(x="day", y="total_bill", data = df, linewidth= 5)  
plt.show()
```

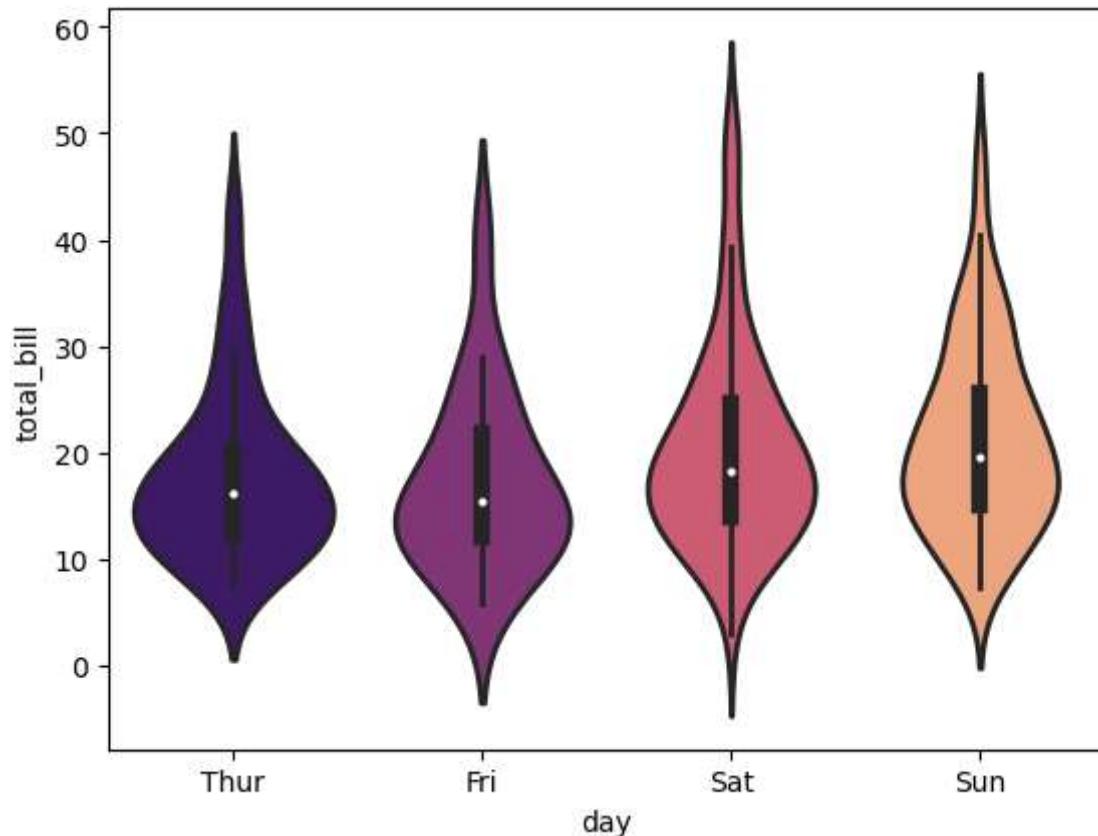


## changing the color

```
In [62]: sns.violinplot(x="day", y="total_bill", data = df, linewidth= 2, palette = "jet")
plt.show()
```

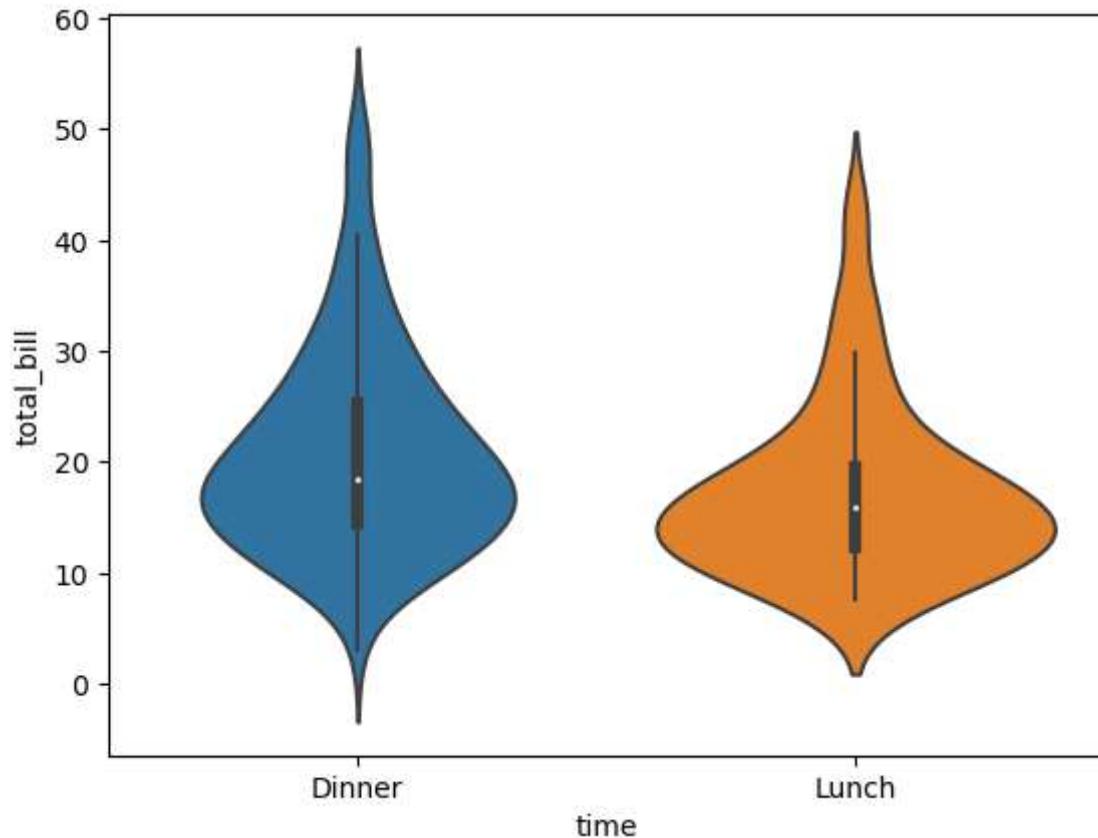


```
In [63]: sns.violinplot(x="day", y="total_bill", data = df, linewidth= 2, palette = "magma")
plt.show()
```



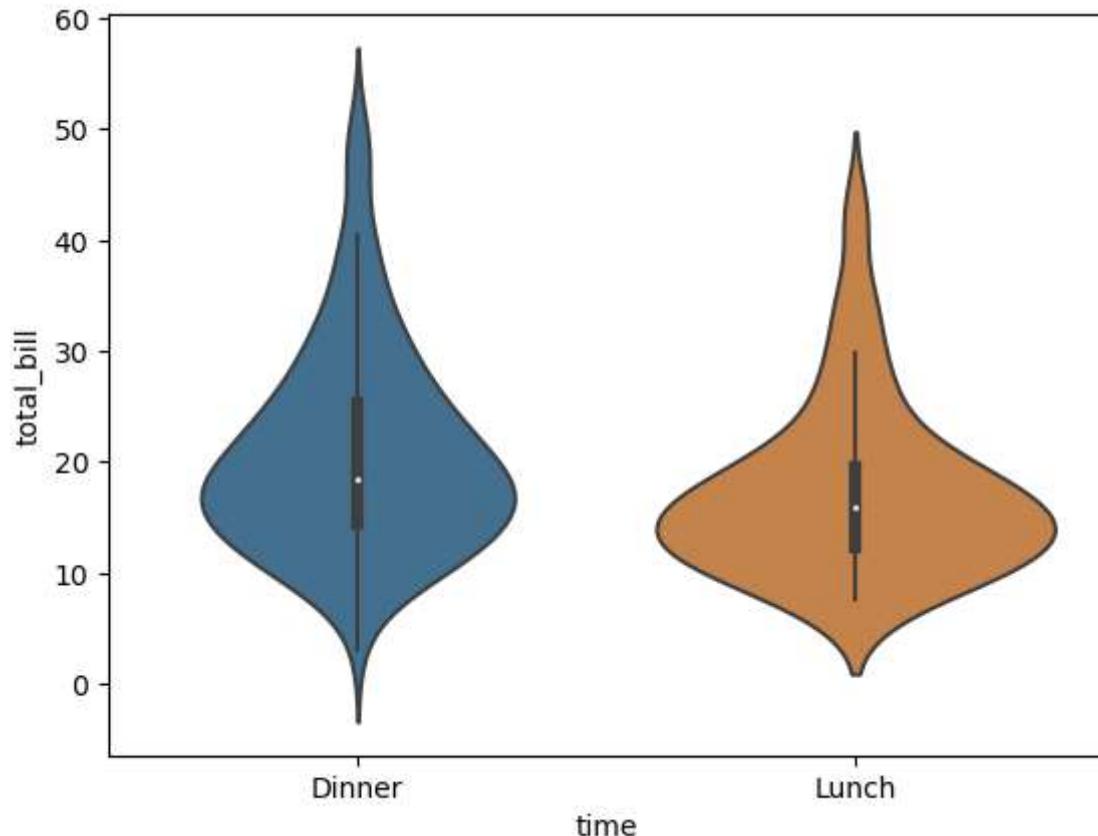
## setting the order

```
In [64]: sns.violinplot(x="time", y="total_bill", data = df, order = ["Dinner", "Lunch"]
plt.show()
```

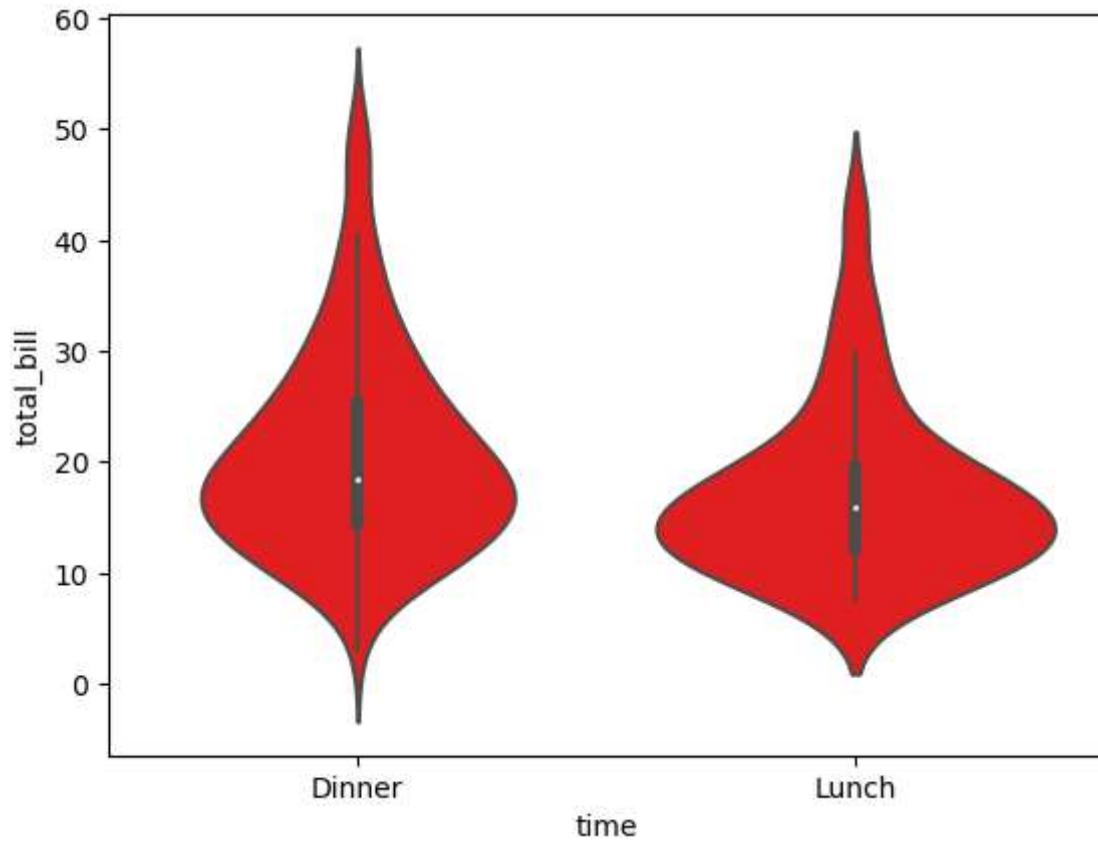


saturation

```
In [65]: sns.violinplot(x="time", y="total_bill", data = df, order = ["Dinner", "Lunch"]
plt.show()
```

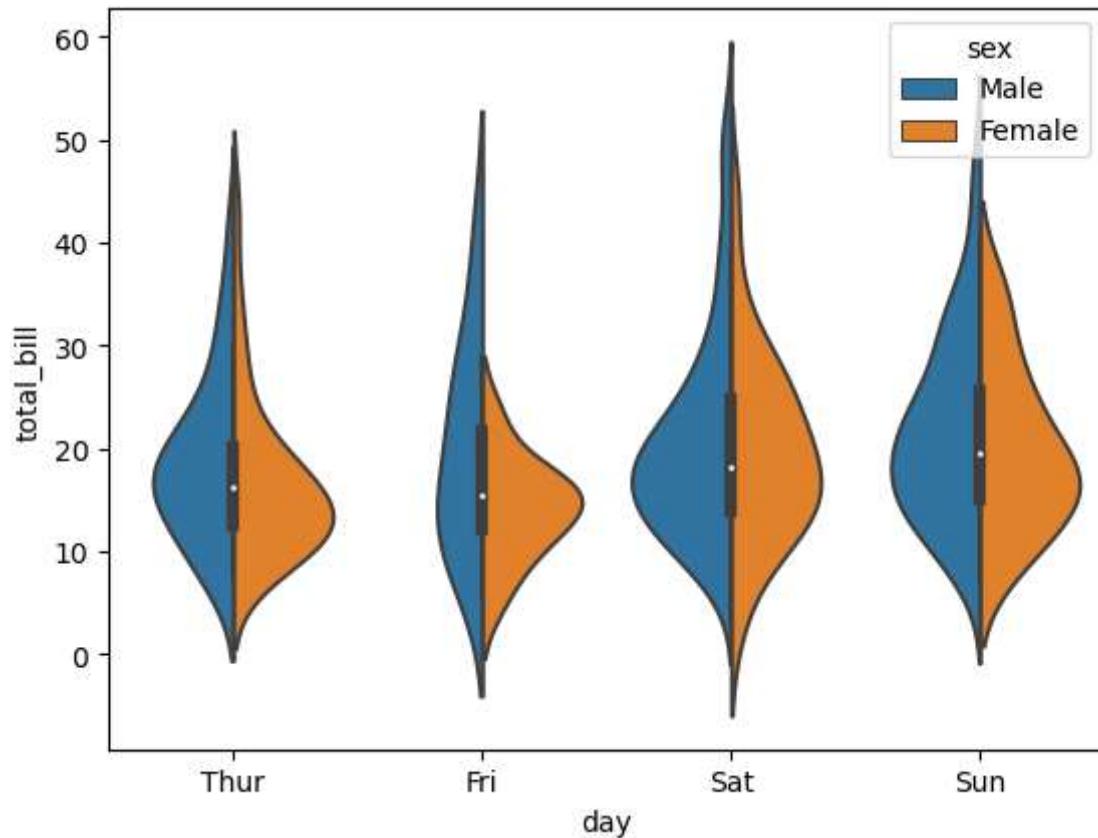


```
In [66]: sns.violinplot(x="time", y="total_bill", data = df, order = ["Dinner", "Lunch"]
plt.show()
```

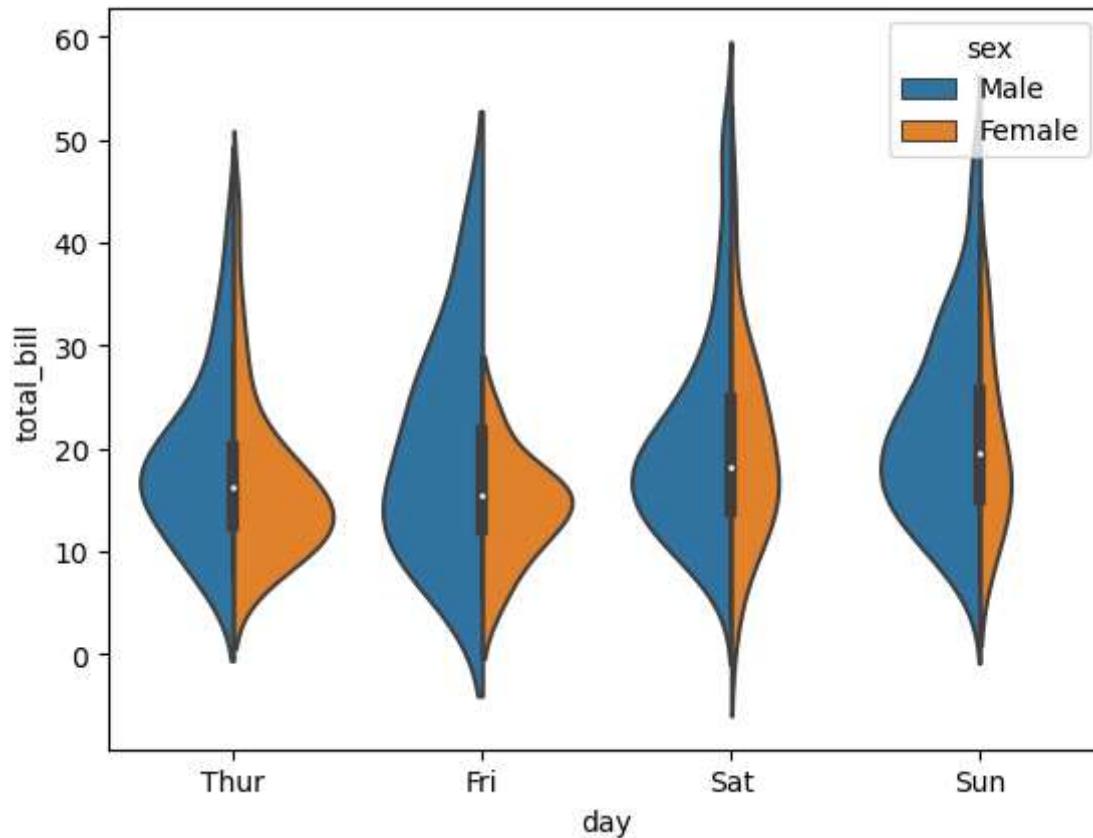


categorizing the data into male and female

```
In [67]: sns.violinplot(x="day", y="total_bill", data = df, hue = "sex", split = True)  
plt.show()
```

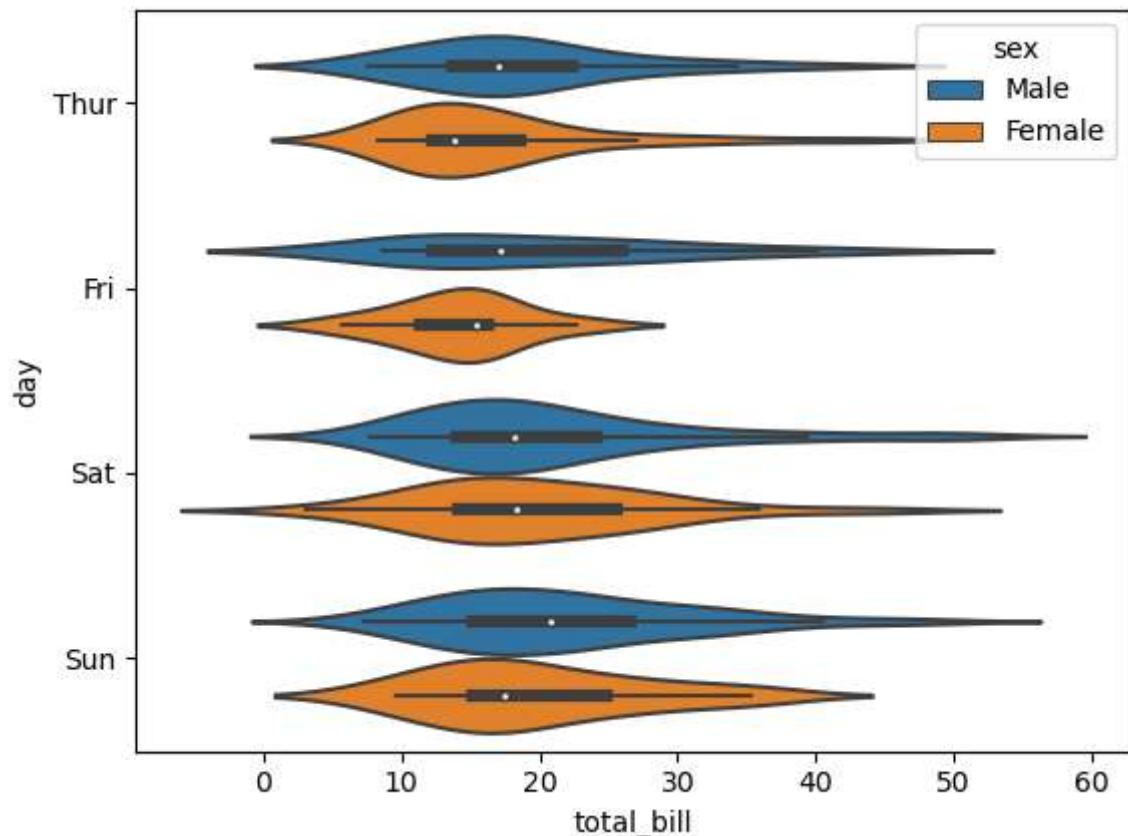


```
In [68]: sns.violinplot(x="day", y="total_bill", data = df, hue = "sex", split = True,  
plt.show()
```



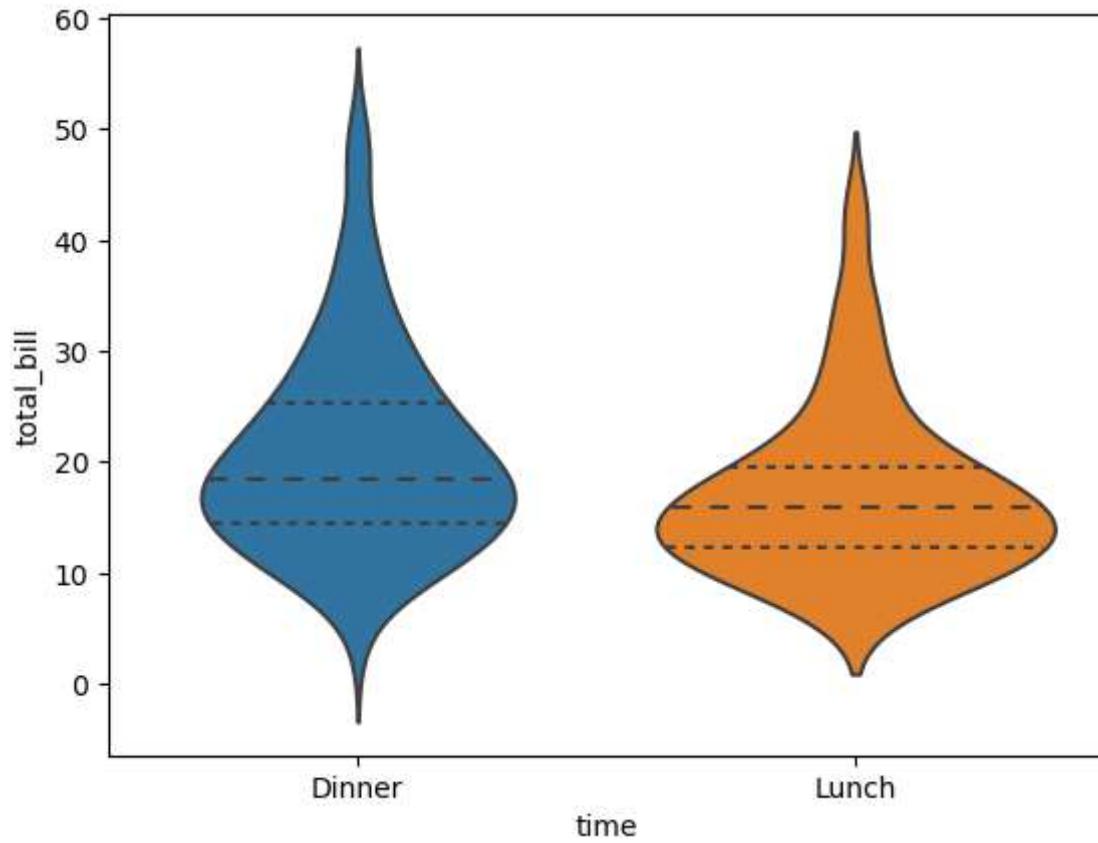
changing the orientation of the violin plot

```
In [69]: sns.violinplot(y="day", x="total_bill", data = df, hue = "sex")
plt.show()
```

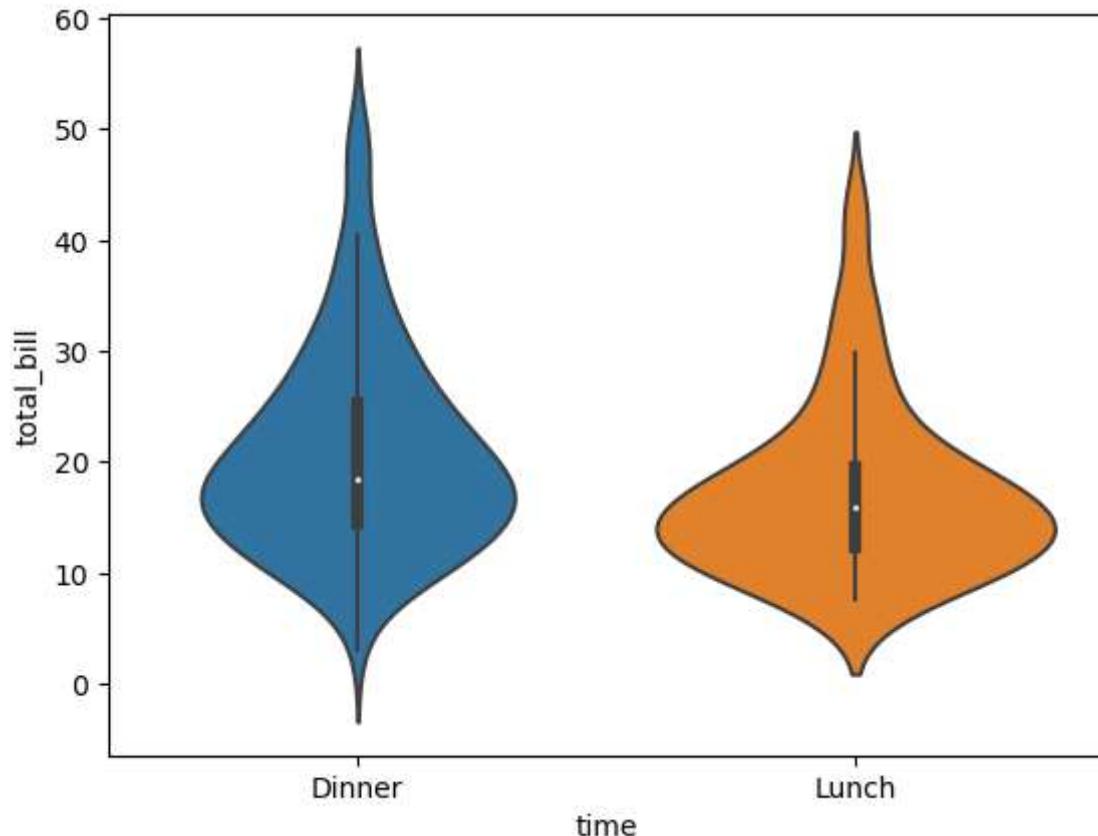


inner parameter

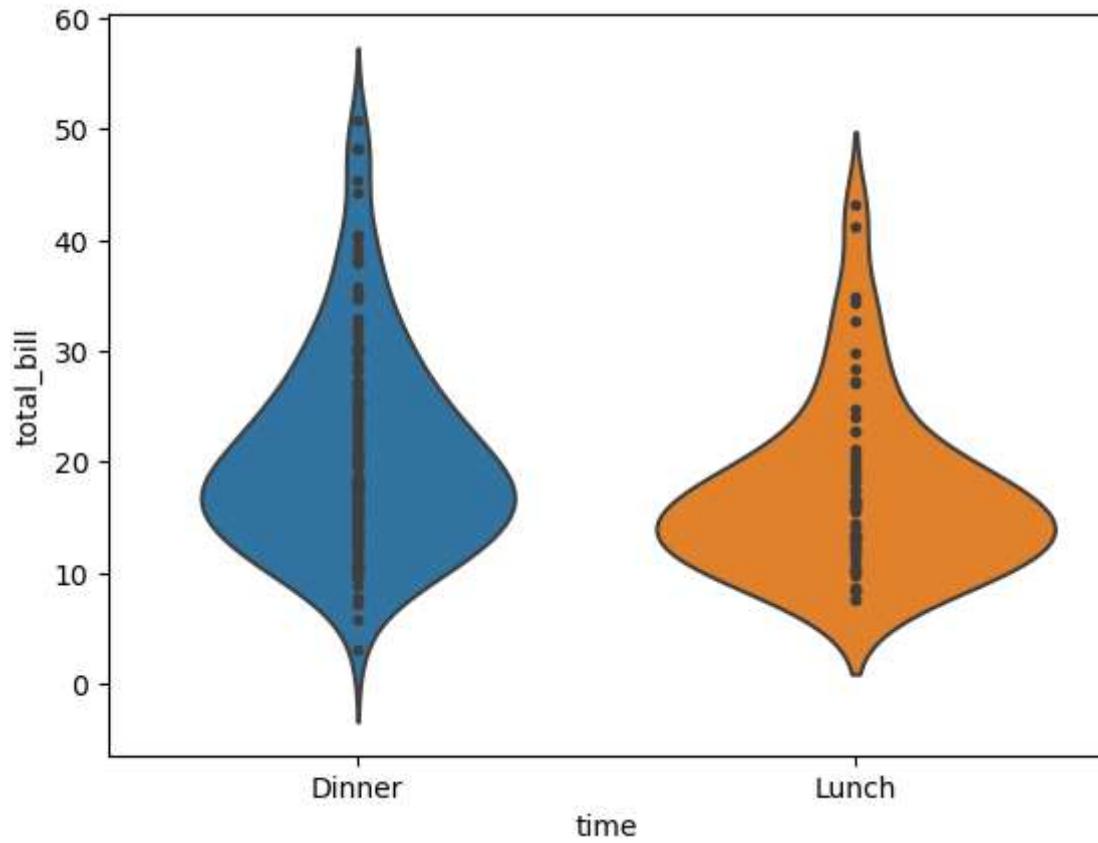
```
In [70]: sns.violinplot(x="time", y="total_bill", data = df, order = ["Dinner", "Lunch"]
plt.show()
```



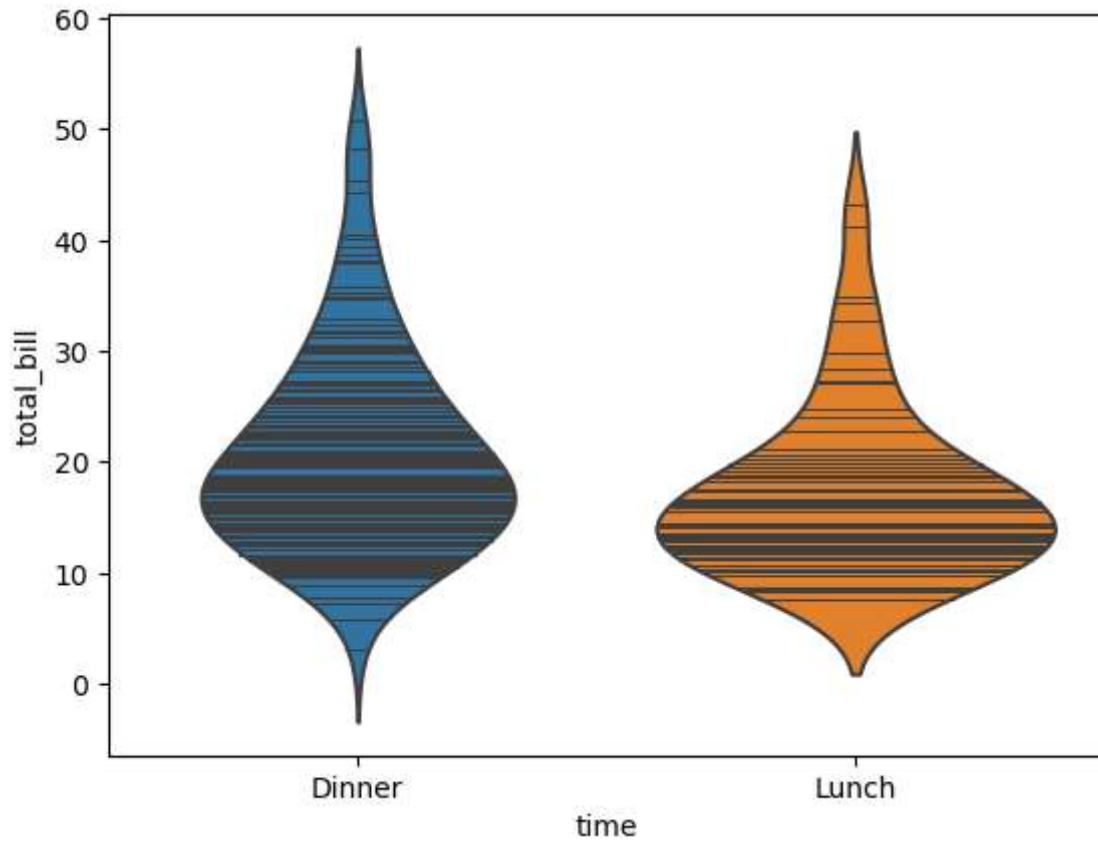
```
In [71]: sns.violinplot(x="time", y="total_bill", data = df, order = ["Dinner", "Lunch"]
plt.show()
```



```
In [73]: sns.violinplot(x="time", y="total_bill", data = df, order = ["Dinner", "Lunch"]
plt.show()
```

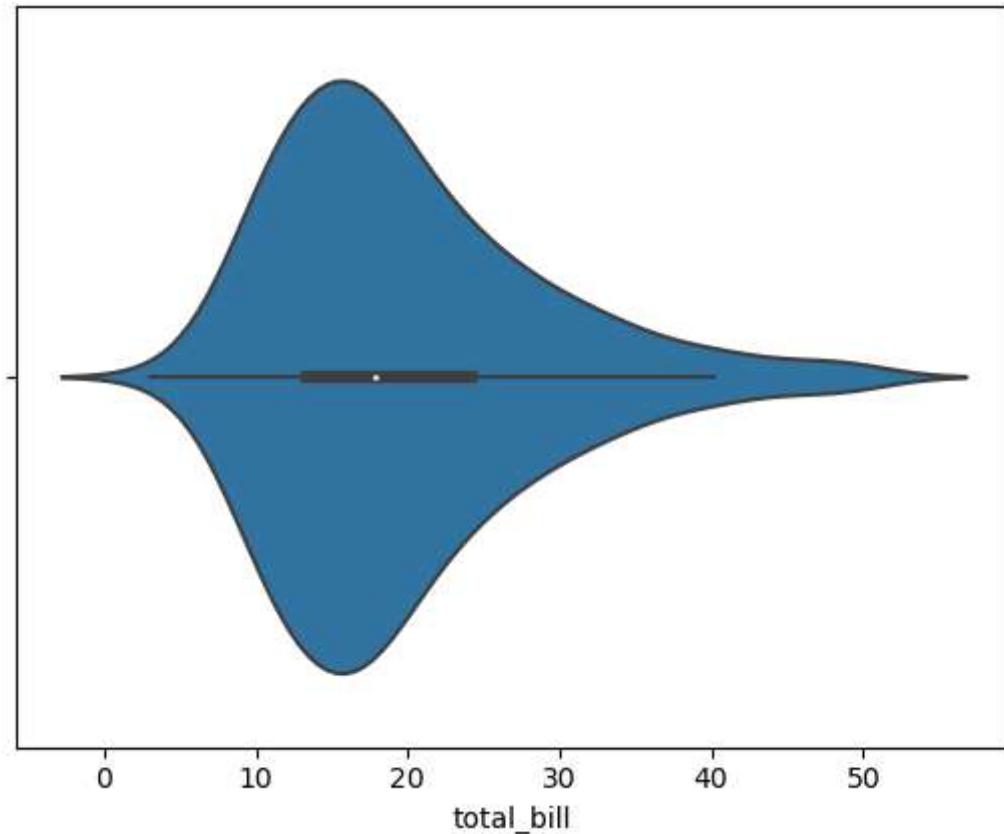


```
In [74]: sns.violinplot(x="time", y="total_bill", data = df, order = ["Dinner", "Lunch"]
plt.show()
```

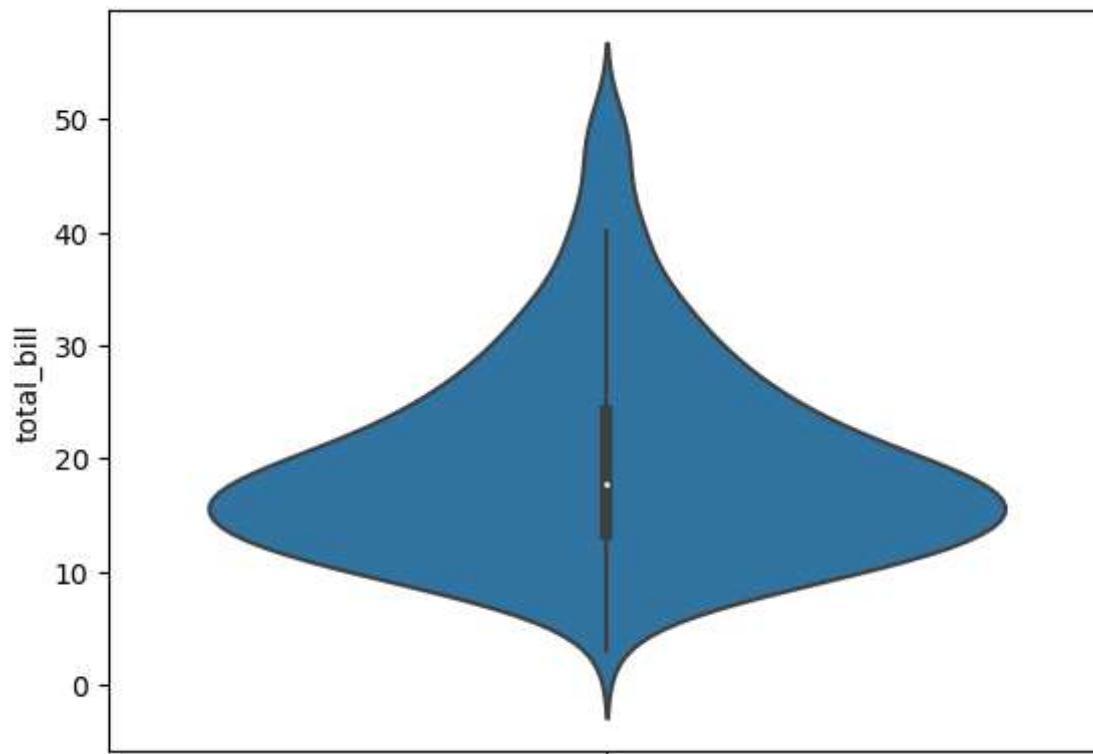


drawing single violin plot

```
In [75]: sns.violinplot(x=df["total_bill"])
plt.show()
```



```
In [76]: sns.violinplot(y=df["total_bill"])
plt.show()
```



# Pair Plot in Seaborn

```
In [77]: import matplotlib.pyplot as plt  
import seaborn as sns  
import pandas as pd
```

```
In [79]: df =sns.load_dataset("tips")
```

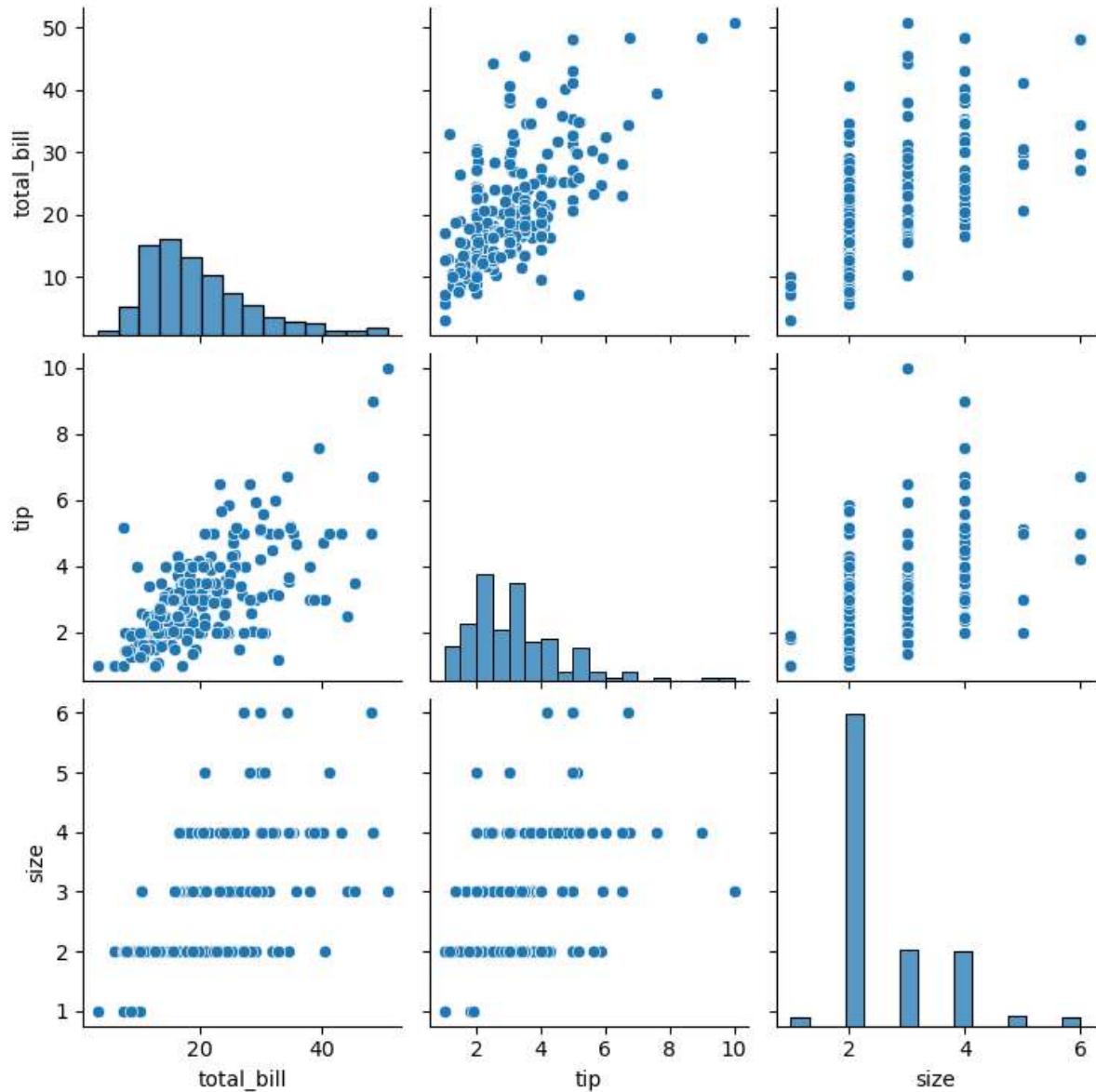
```
In [80]: df
```

Out[80]:

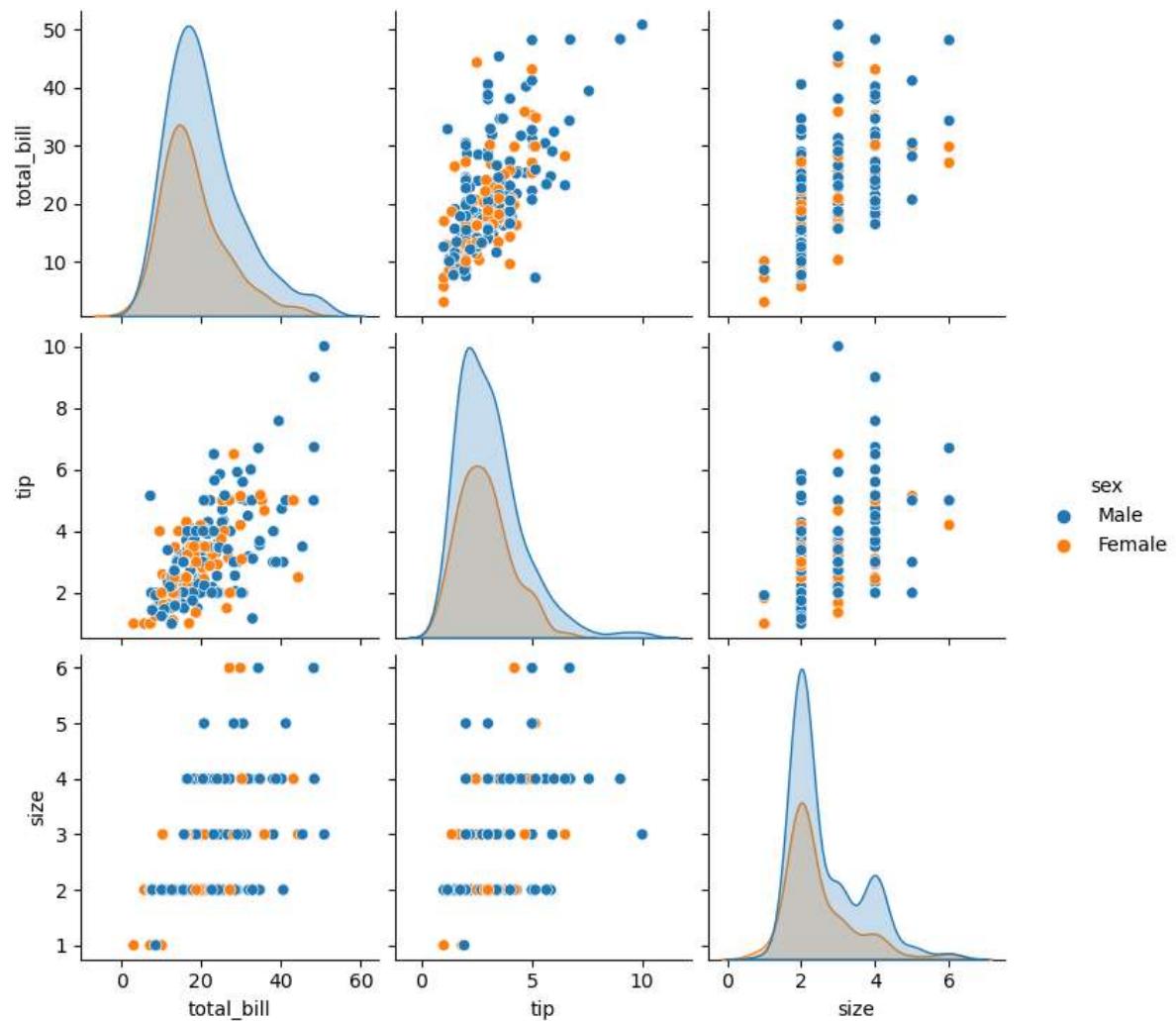
	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4
...	...	...	...	...	...	...	...
239	29.03	5.92	Male	No	Sat	Dinner	3
240	27.18	2.00	Female	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

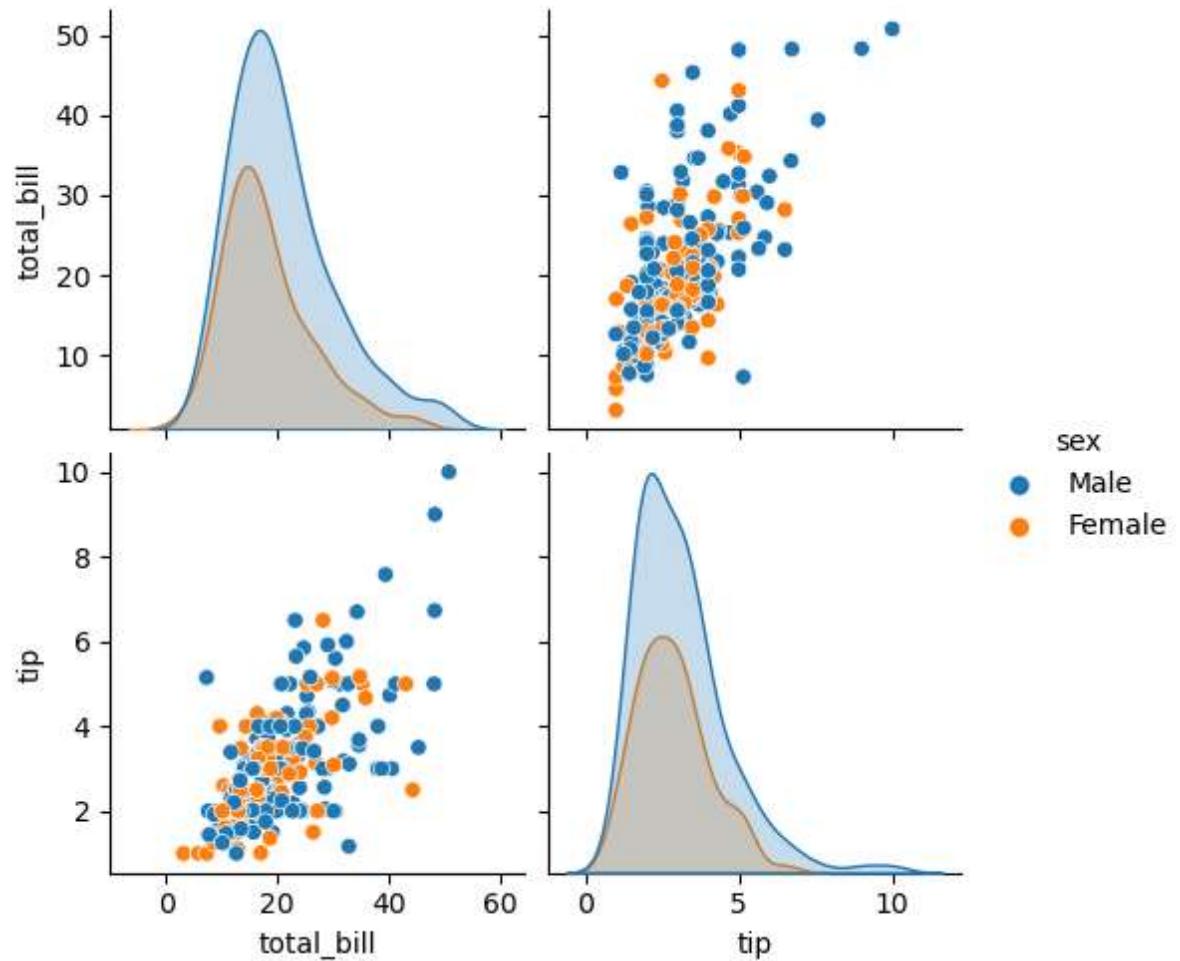
```
In [81]: sns.pairplot(df)
plt.show()
```



```
In [82]: sns.pairplot(df, hue="sex")
plt.show()
```

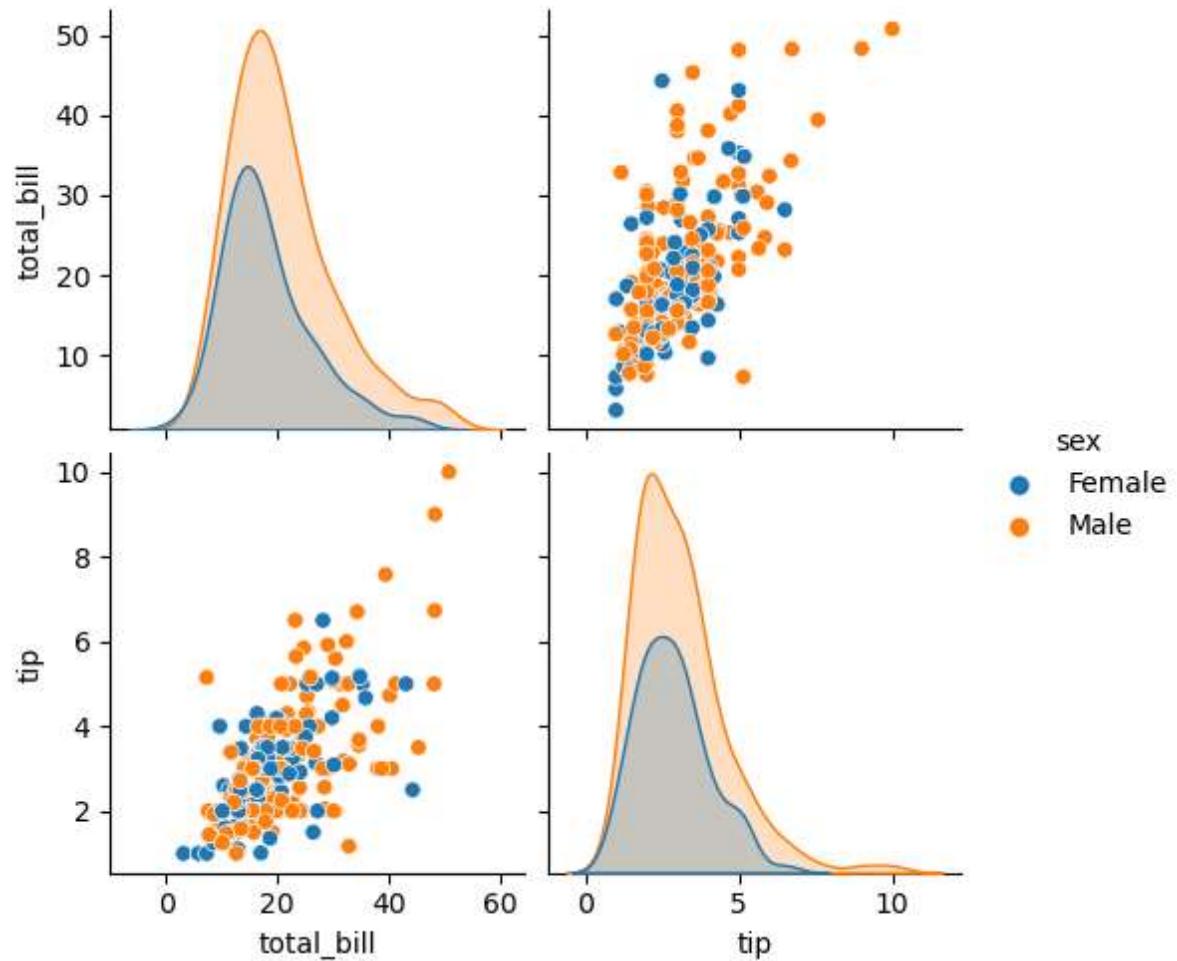


```
In [89]: sns.pairplot(df, vars=["total_bill", "tip"], hue="sex")
plt.show()
```

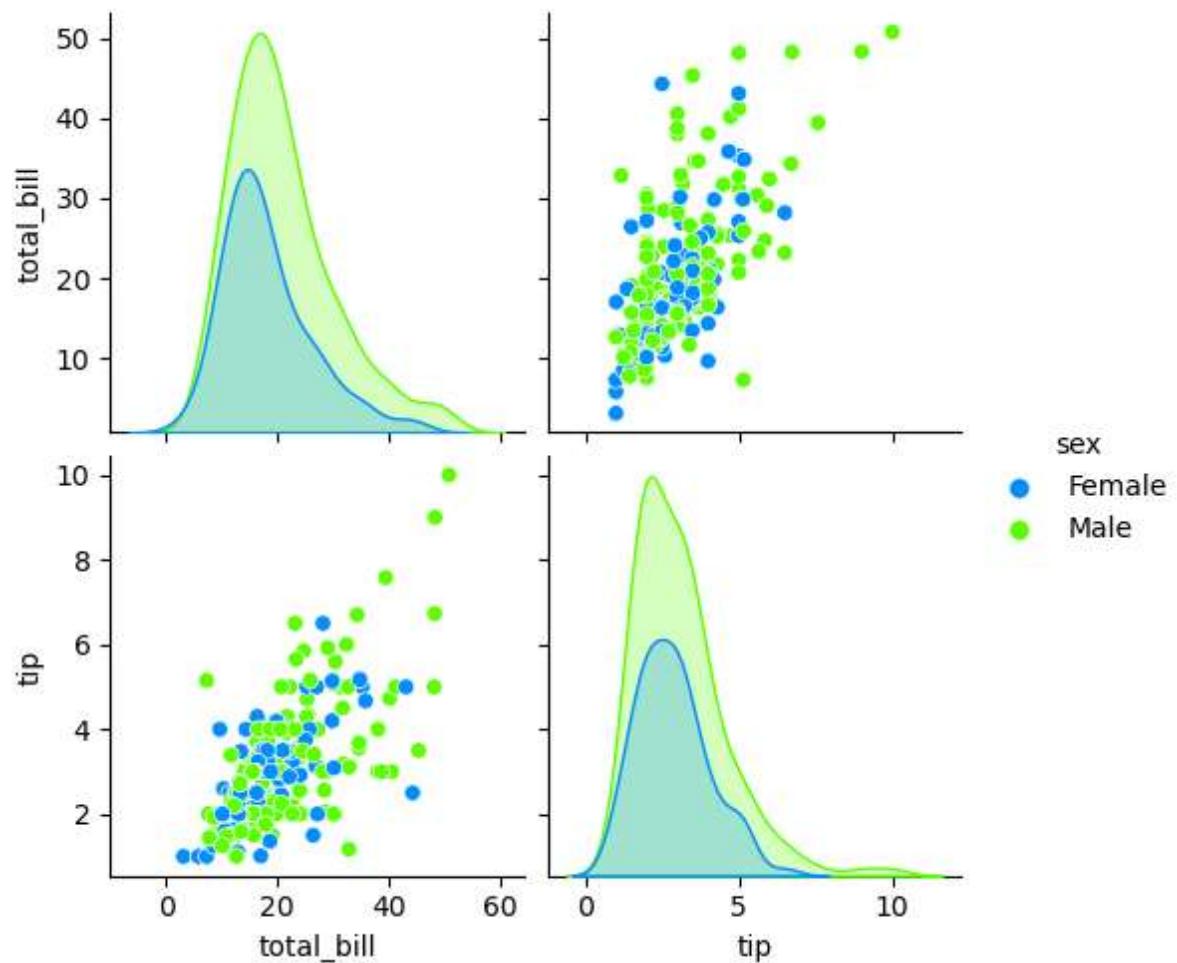


## Changing the Hue Order

```
In [90]: sns.pairplot(df, vars=["total_bill", "tip"], hue="sex", hue_order = ["Female", "Male"])
plt.show()
```

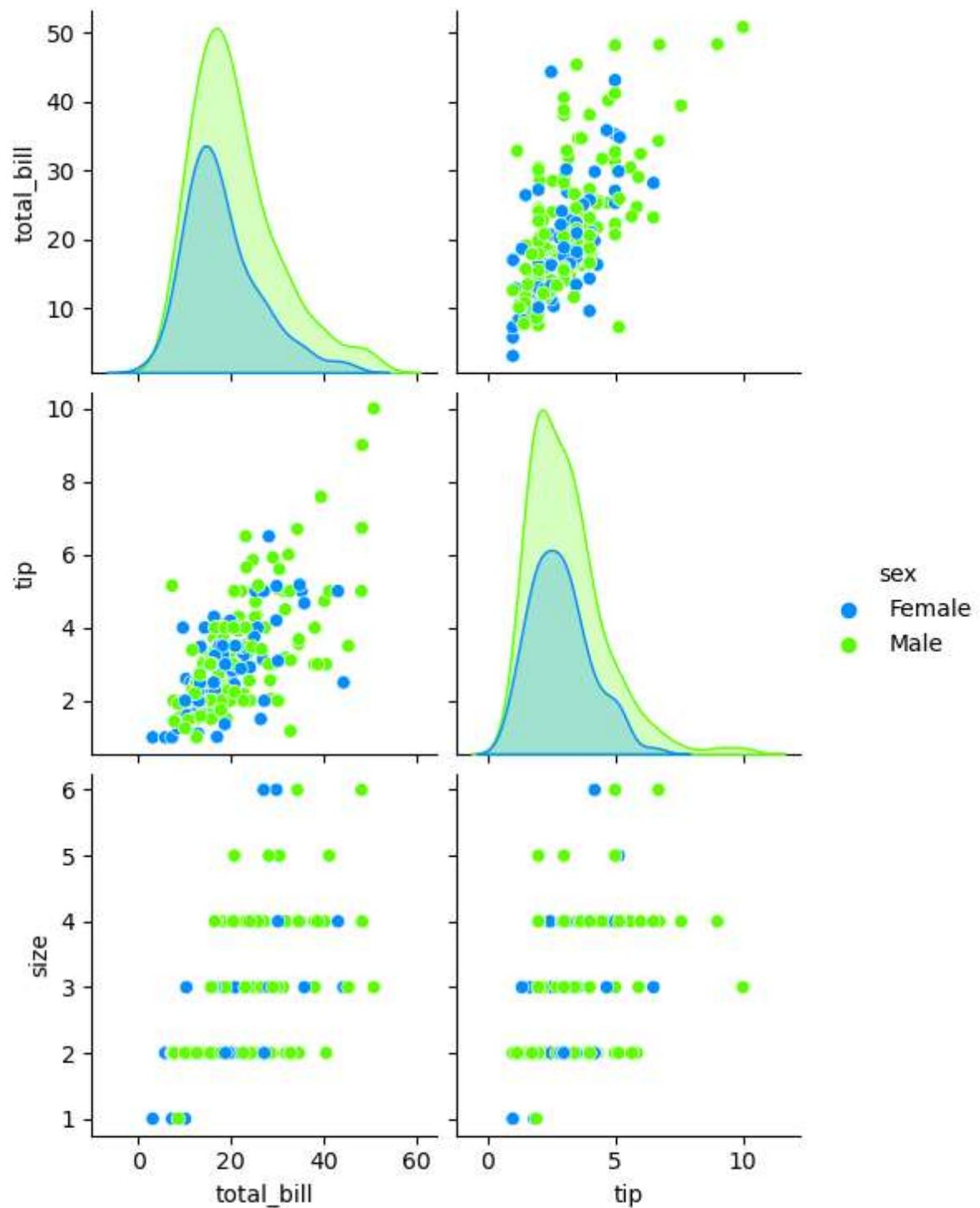


```
In [102]: sns.pairplot(df, vars=["total_bill", "tip"], hue="sex", hue_order = ["Female", "Male"], palette = "gist_rainbow_r")
plt.show()
```

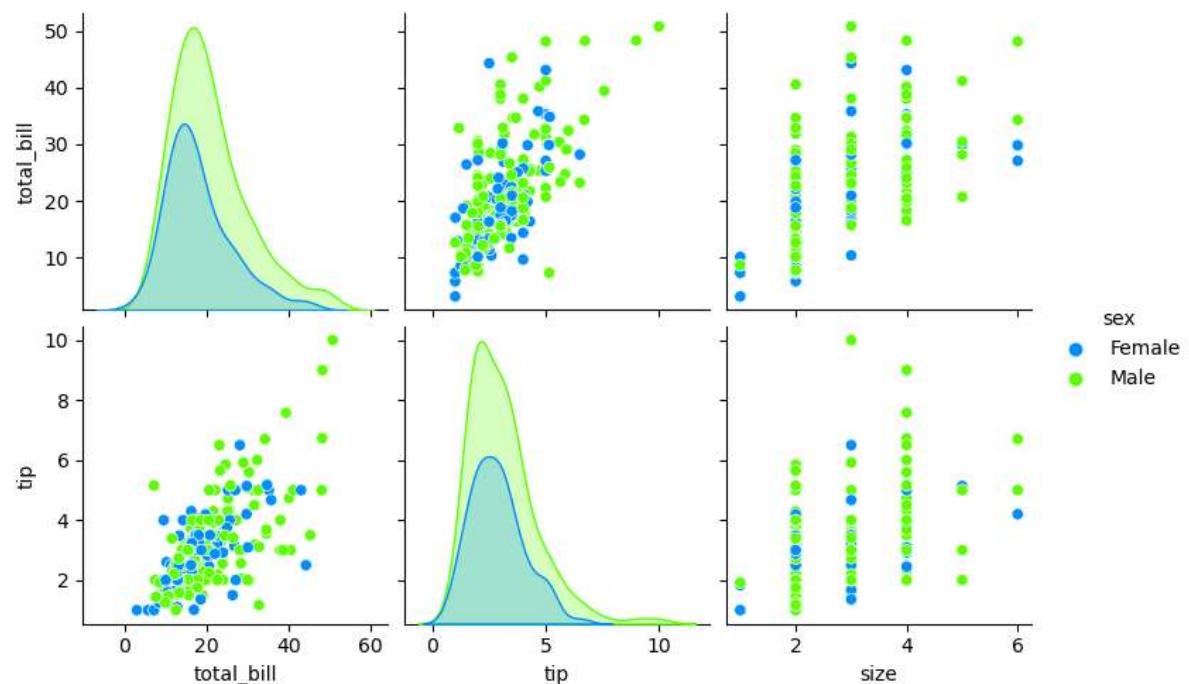


## changing the name of x and y axis

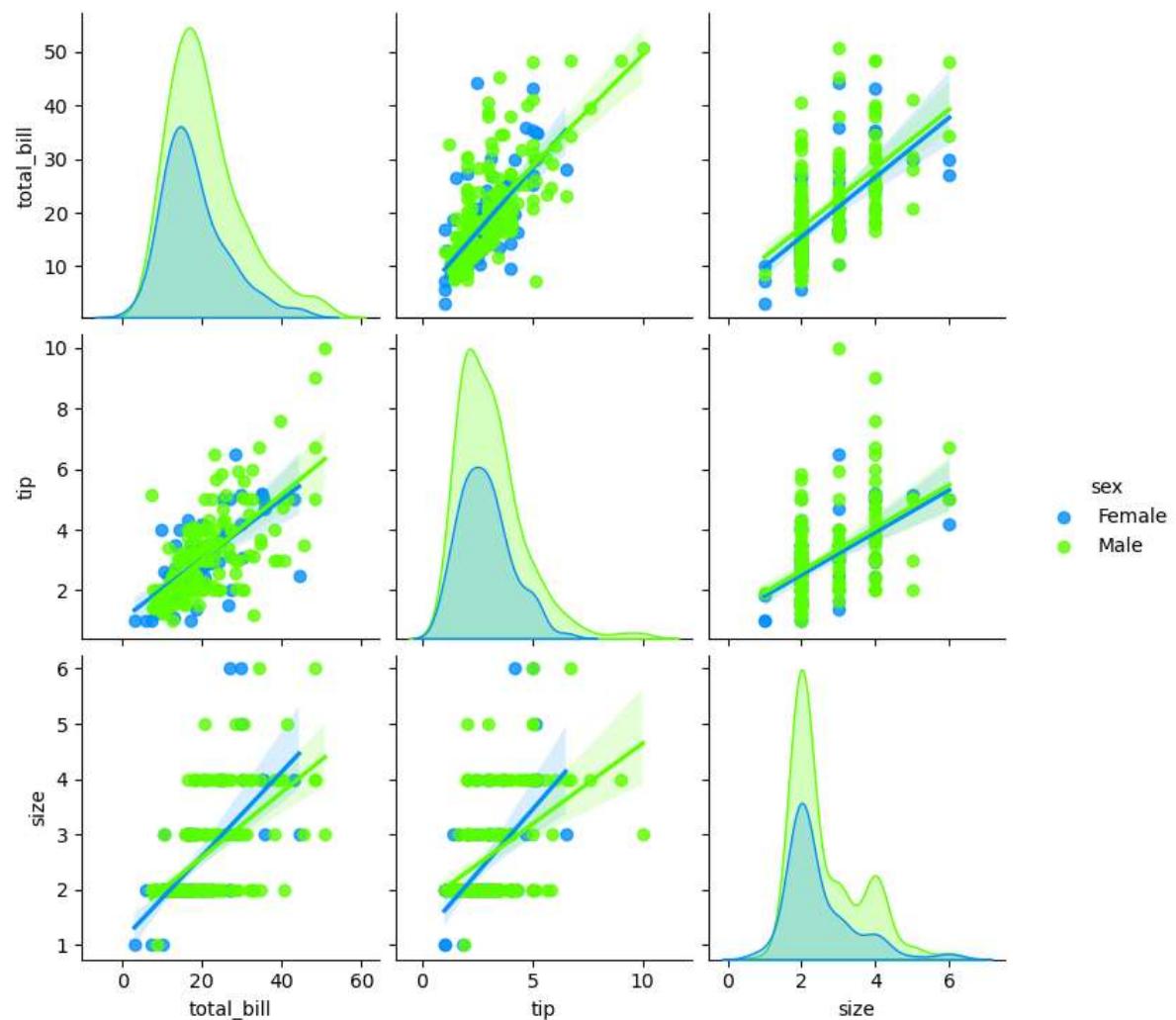
```
In [104]: sns.pairplot(df,hue="sex",hue_order = ["Female","Male"], x_vars= ["total_bill"]
                      palette = "gist_rainbow_r")
plt.show()
```



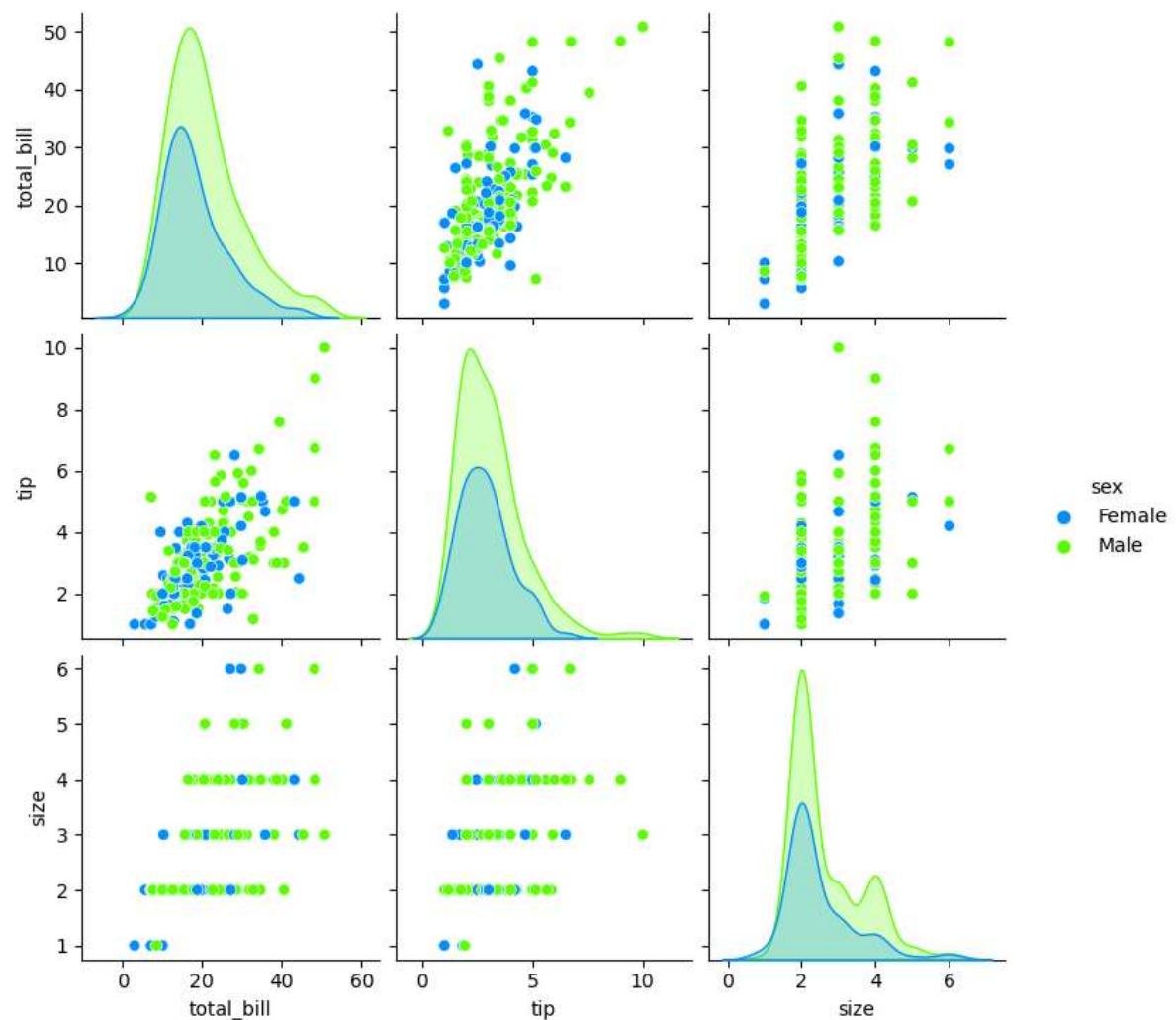
```
In [105]: sns.pairplot(df,hue="sex",hue_order = ["Female","Male"], y_vars= ["total_bill"]
                      palette = "gist_rainbow_r")
plt.show()
```



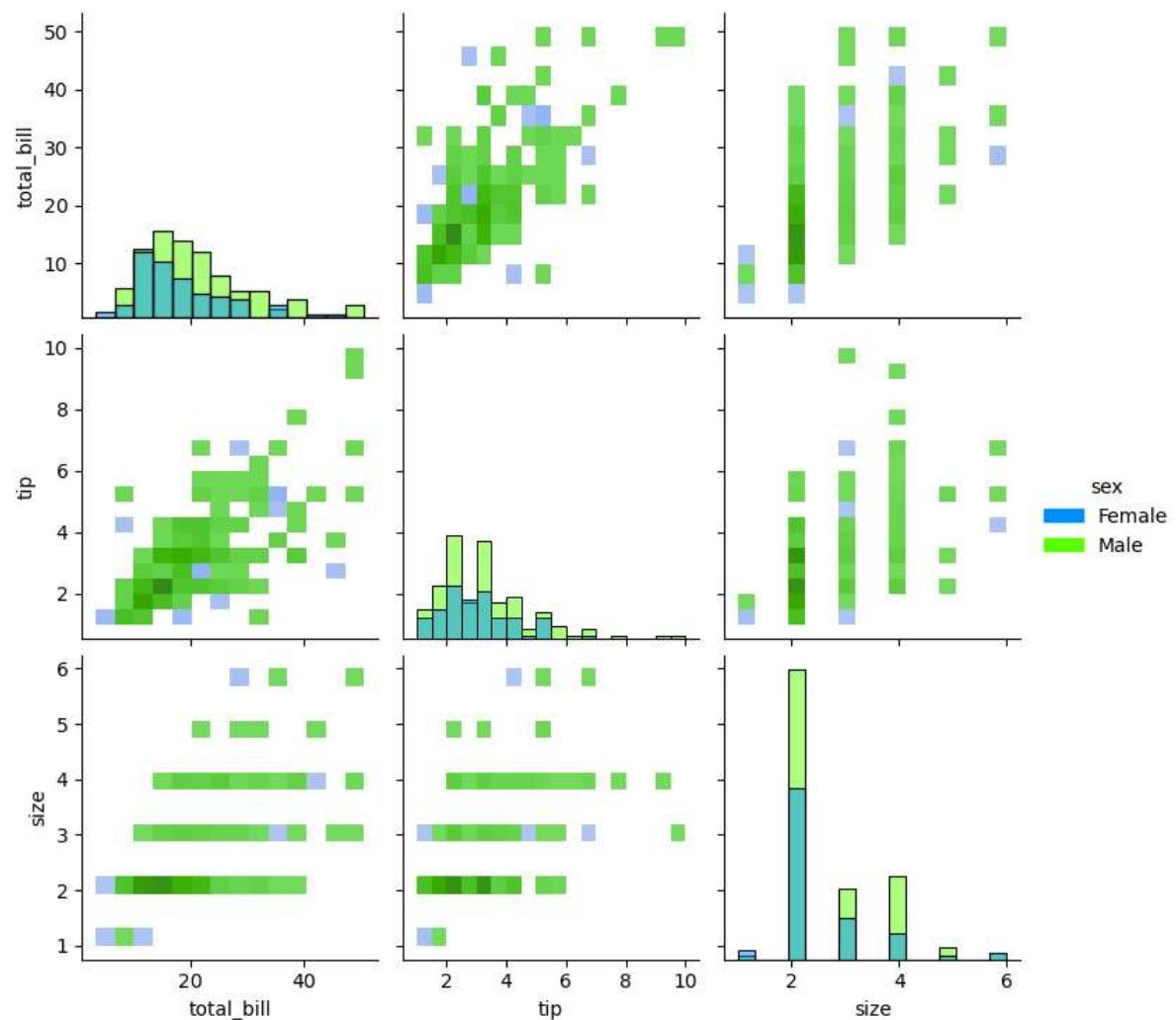
```
In [107]: sns.pairplot(df,hue="sex",hue_order = ["Female","Male"],kind="reg",
                     palette = "gist_rainbow_r")
plt.show()
```



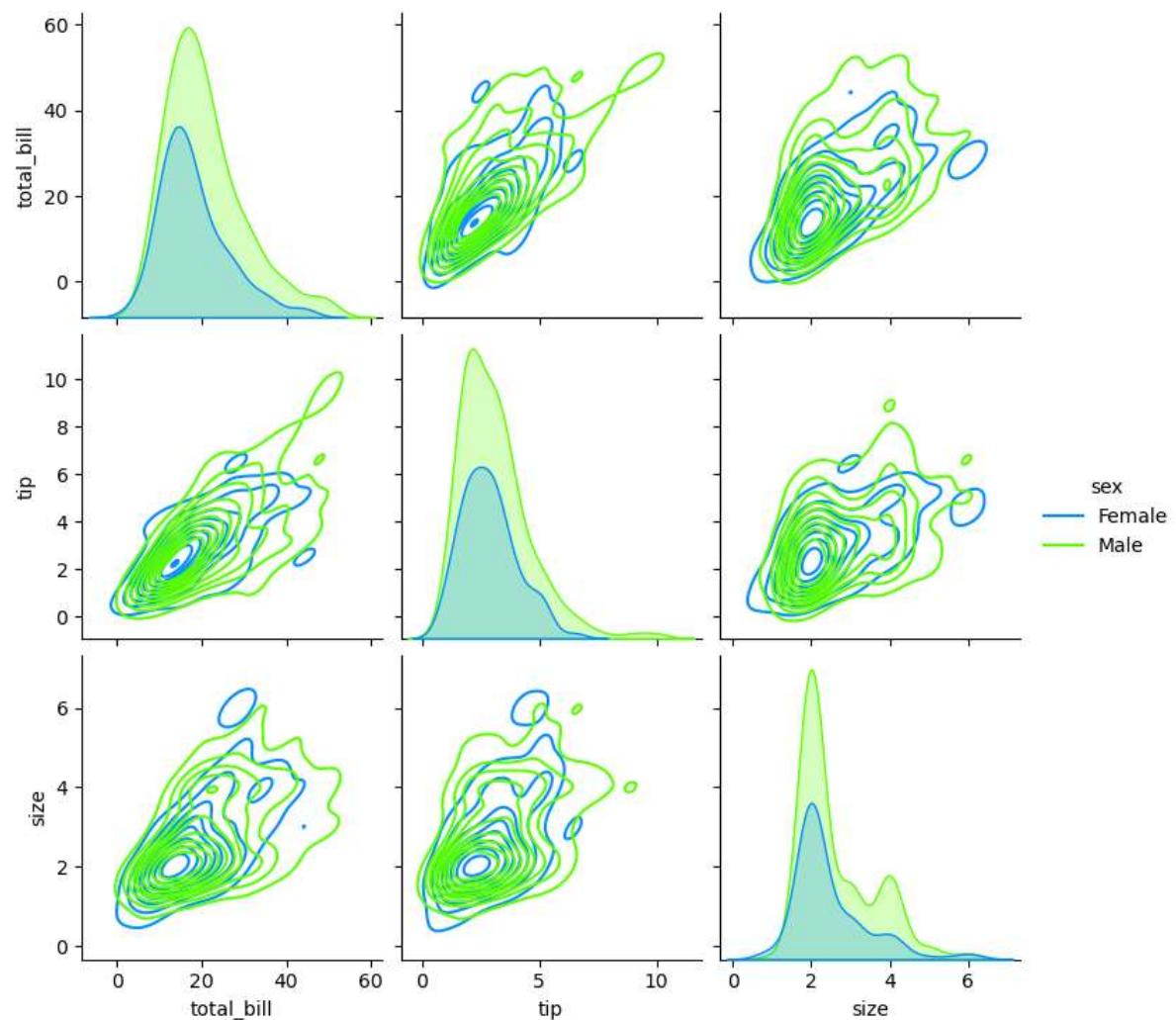
```
In [108]: sns.pairplot(df,hue="sex",hue_order = ["Female","Male"],kind="scatter",
                     palette = "gist_rainbow_r")
plt.show()
```



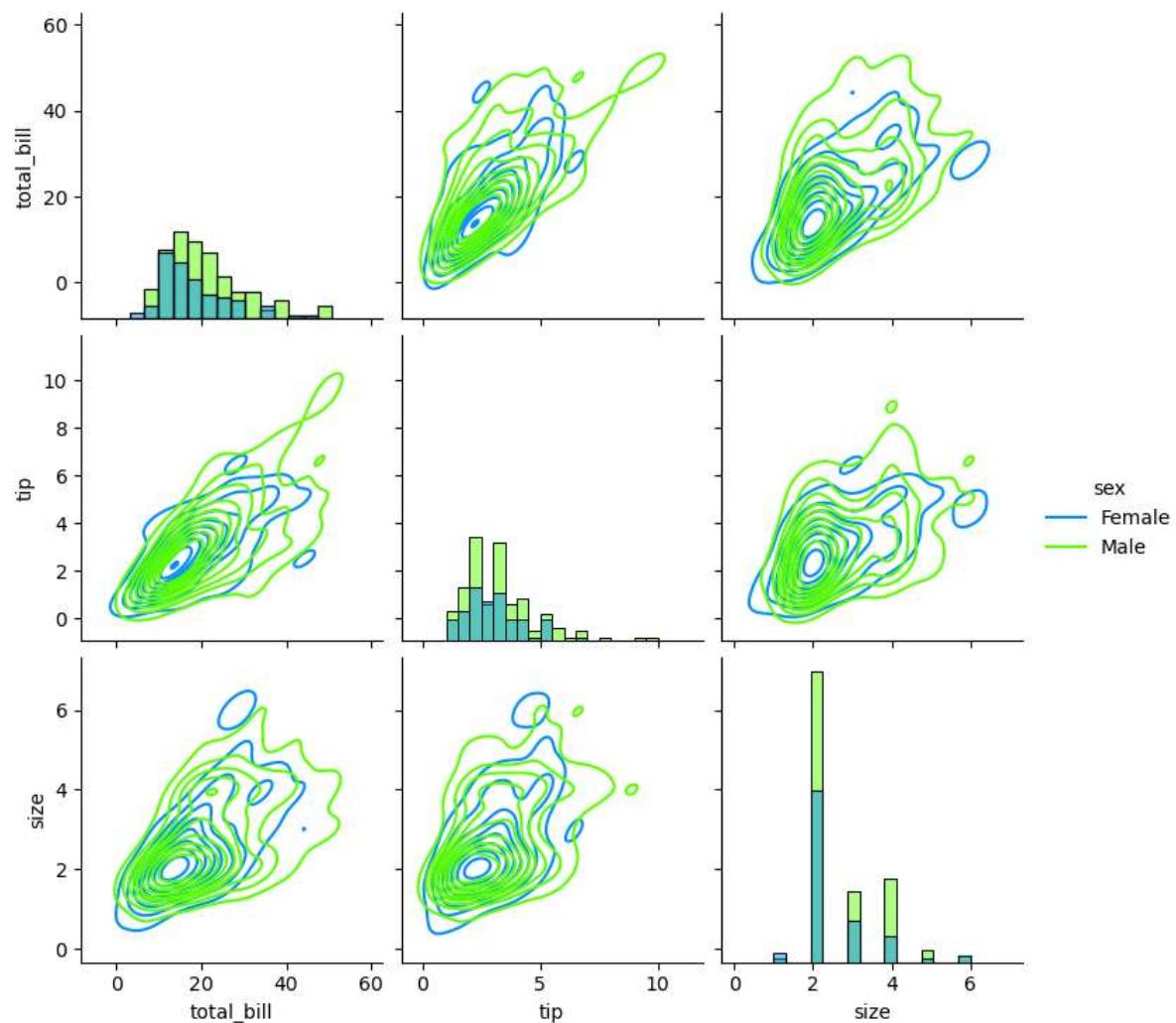
```
In [109]: sns.pairplot(df,hue="sex",hue_order = ["Female","Male"],kind="hist",
                     palette = "gist_rainbow_r")
plt.show()
```



```
In [110]: sns.pairplot(df,hue="sex",hue_order = ["Female","Male"],kind="kde",
                     palette = "gist_rainbow_r")
plt.show()
```



```
In [113]: sns.pairplot(df,hue="sex",hue_order = ["Female","Male"],kind="kde",diag_kind =
                           palette = "gist_rainbow_r")
plt.show()
```



## Strip Plot in Seaborn

```
In [121]: import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
```

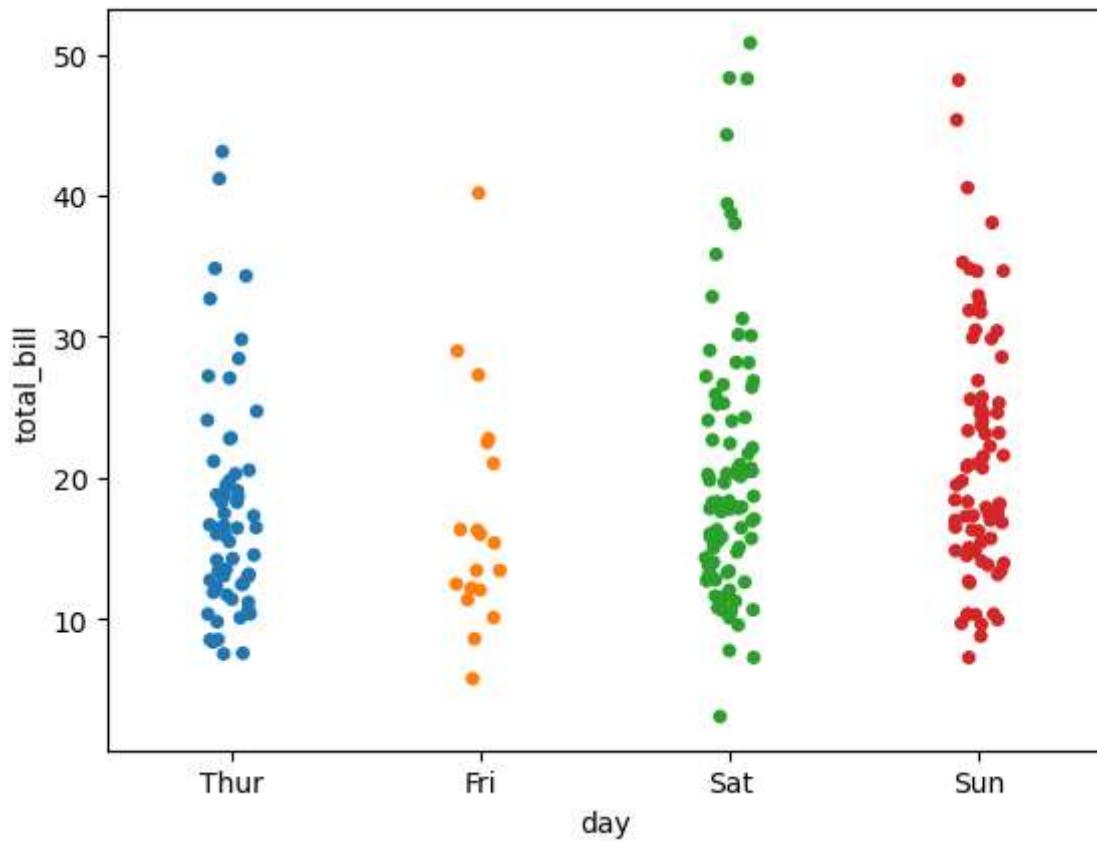
```
In [124]: df = sns.load_dataset("tips")
```

In [125]: df

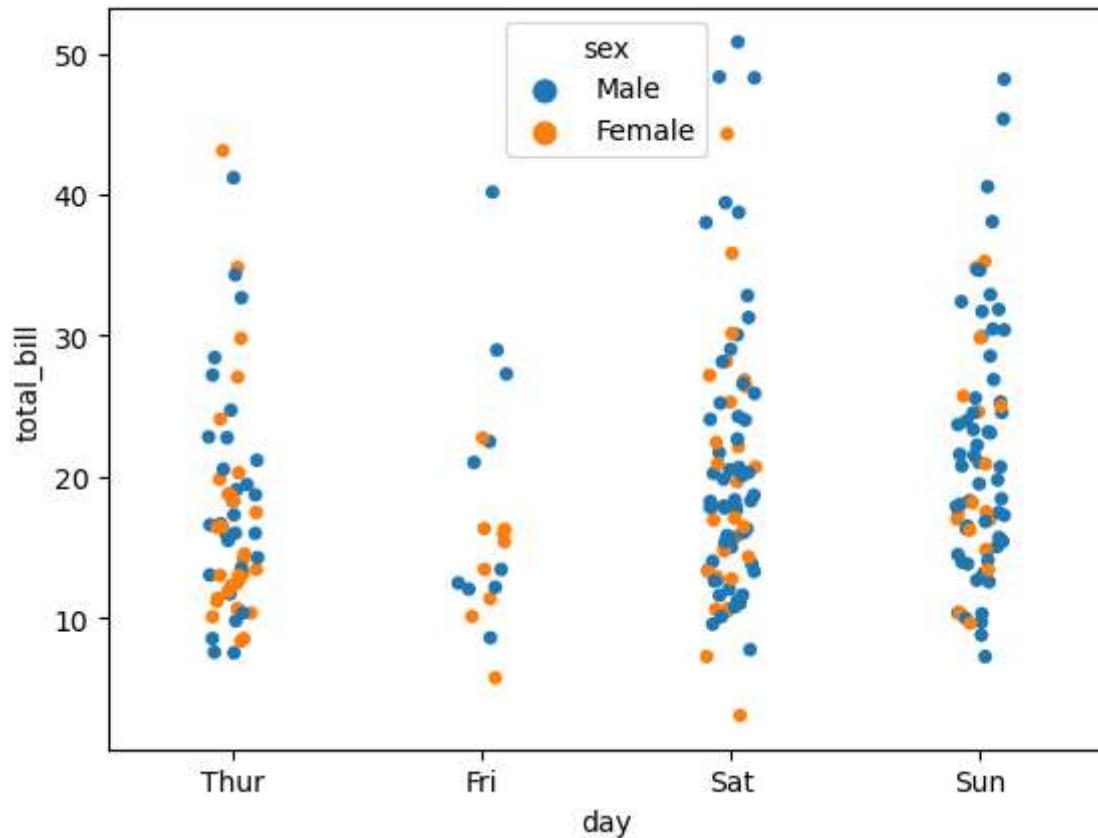
Out[125]:

	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

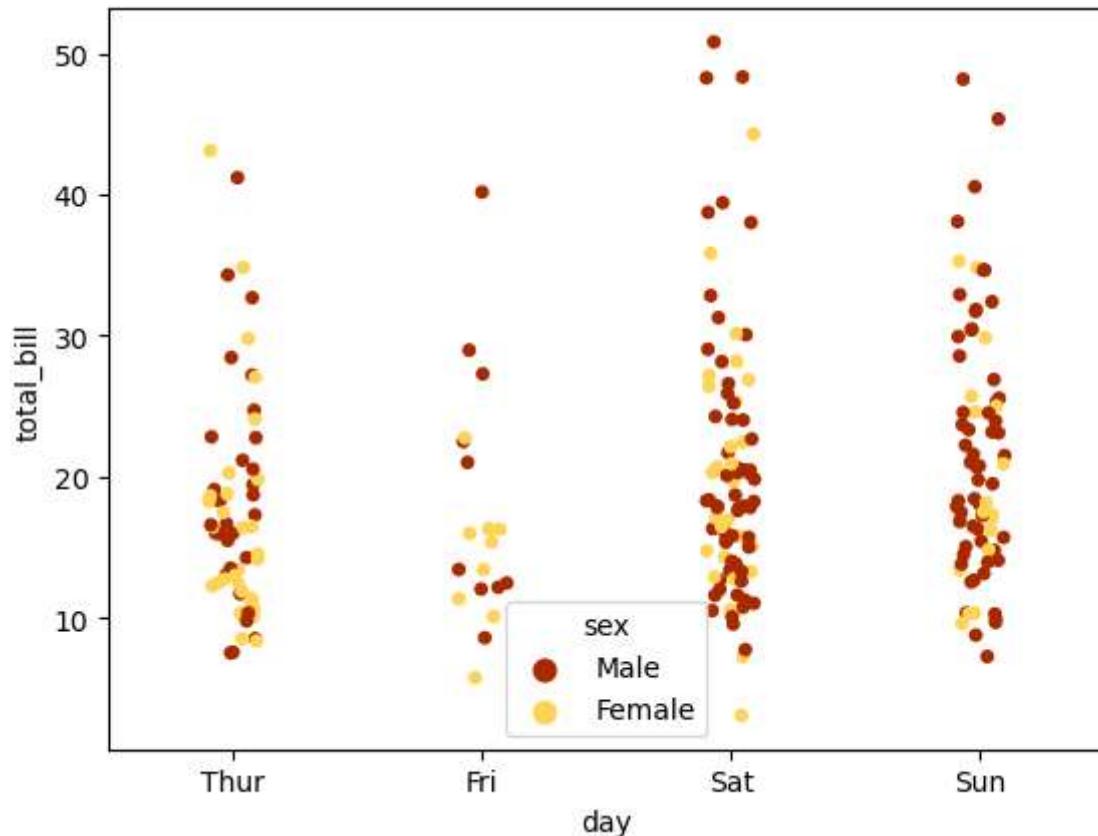
In [126]: sns.stripplot(x="day", y="total\_bill", data = df)  
plt.show()

```
In [127]: sns.stripplot(x="day", y="total_bill", data = df, hue = "sex")
plt.show()
```



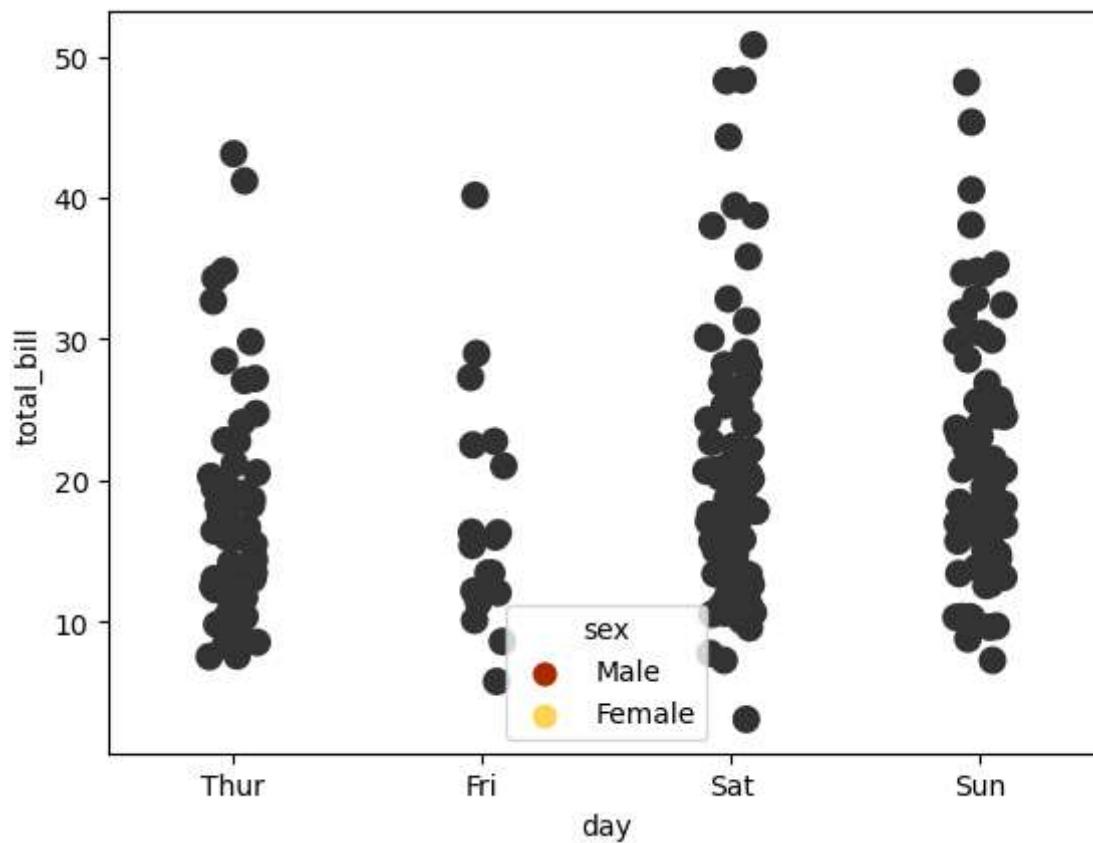
## changing the color of the graph

```
In [135]: sns.stripplot(x="day", y="total_bill", data = df, hue = "sex", palette = "afmhot")
plt.show()
```



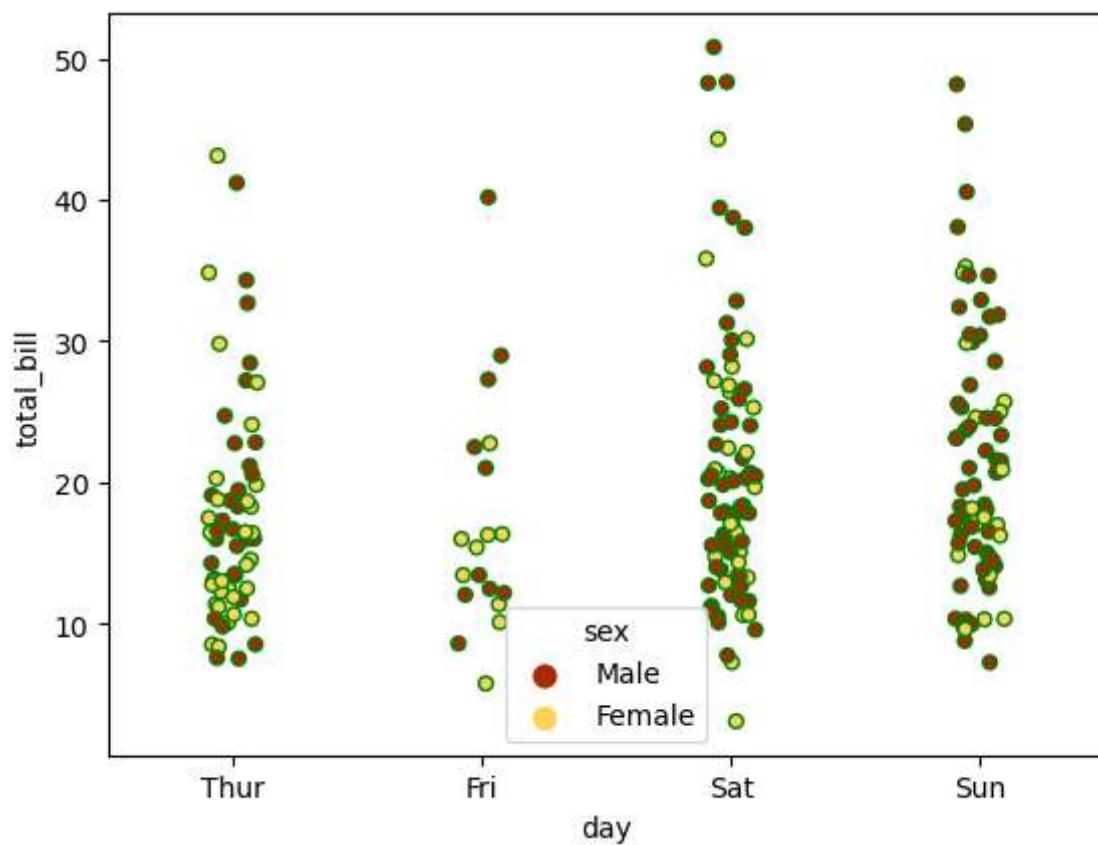
linewidth - changes

```
In [136]: sns.stripplot(x="day", y="total_bill", data = df, hue = "sex", palette = "afmhot"
                      linewidth= 5)
plt.show()
```



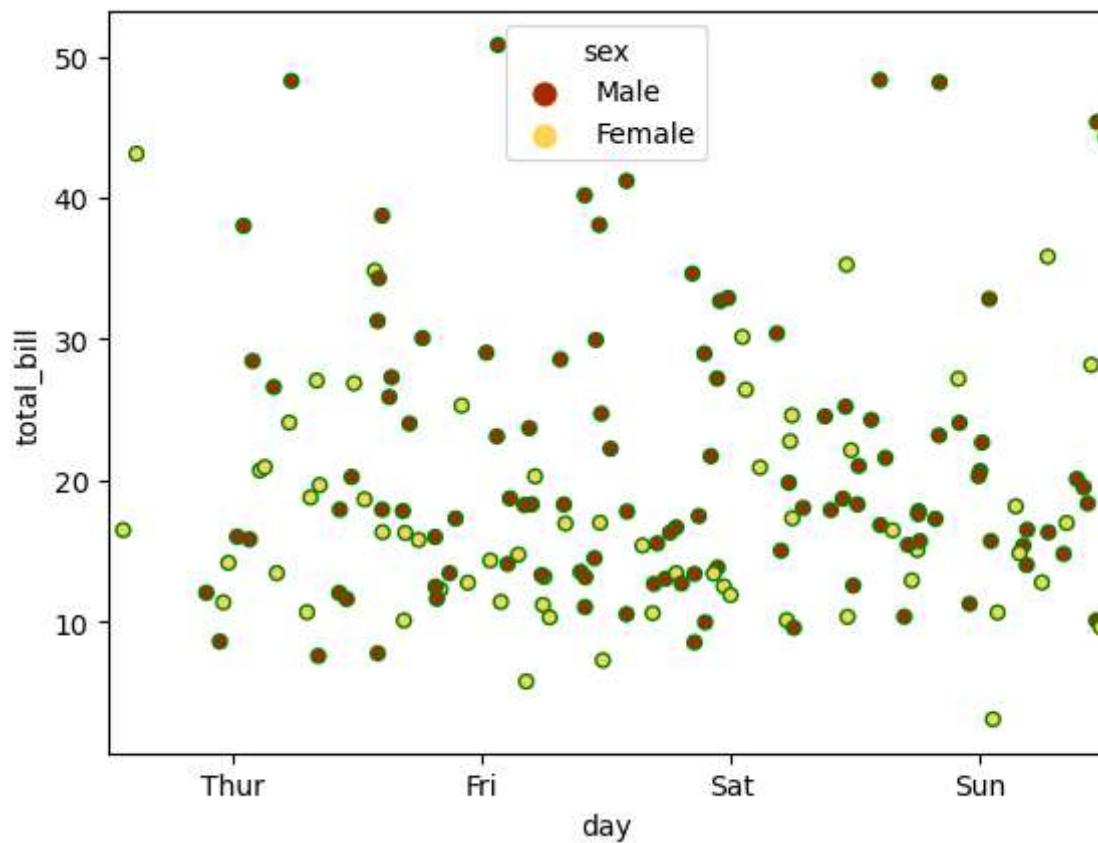
## edgecolor

```
In [137]: sns.stripplot(x="day", y="total_bill", data = df, hue = "sex", palette = "afmhot"
                      linewidth= 1, edgecolor="green")
plt.show()
```

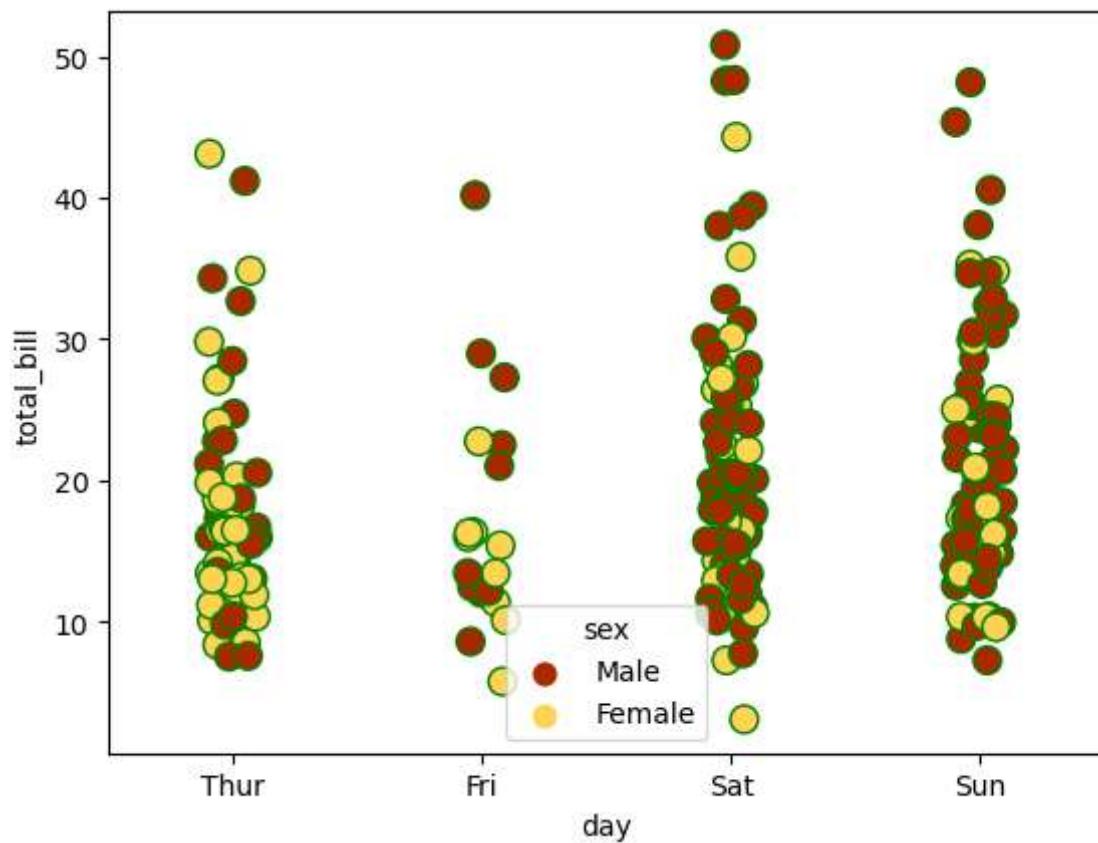


changing the Displacement

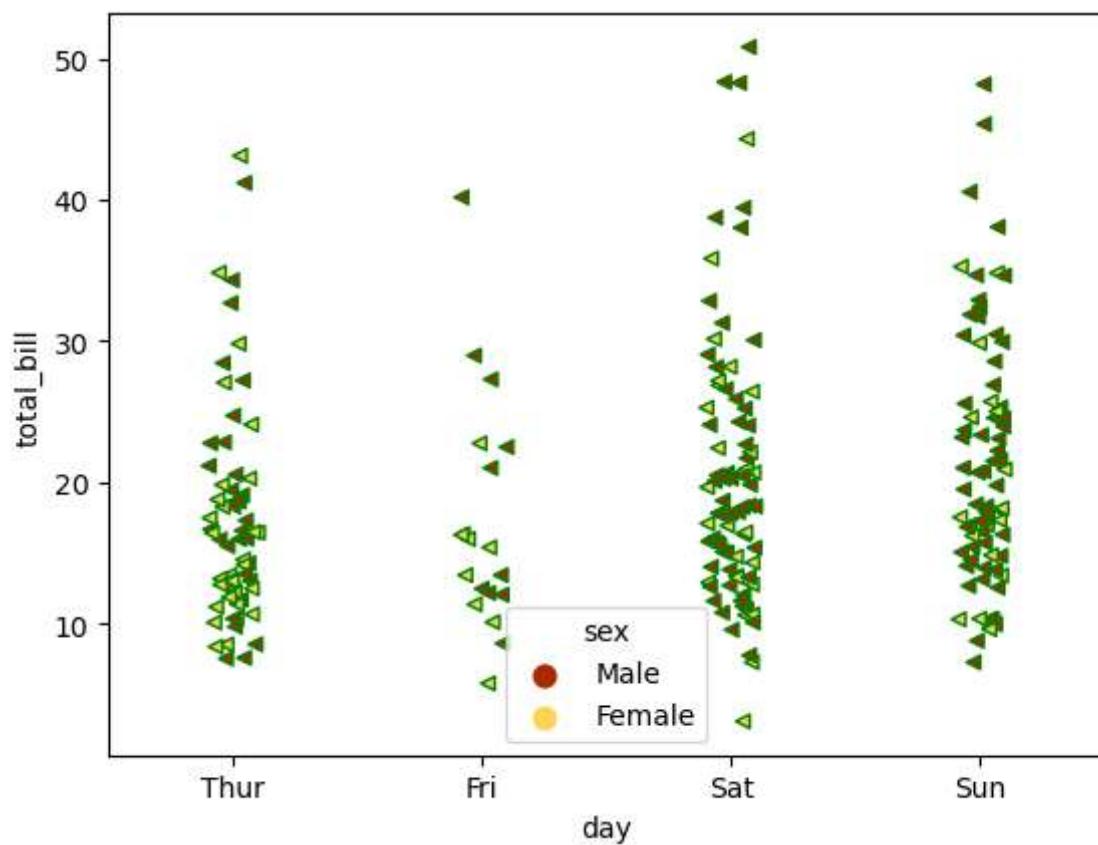
```
In [138]: sns.stripplot(x="day", y="total_bill", data = df, hue = "sex", palette = "afmhot"
                      linewidth= 1, edgecolor="green", jitter =2)
plt.show()
```



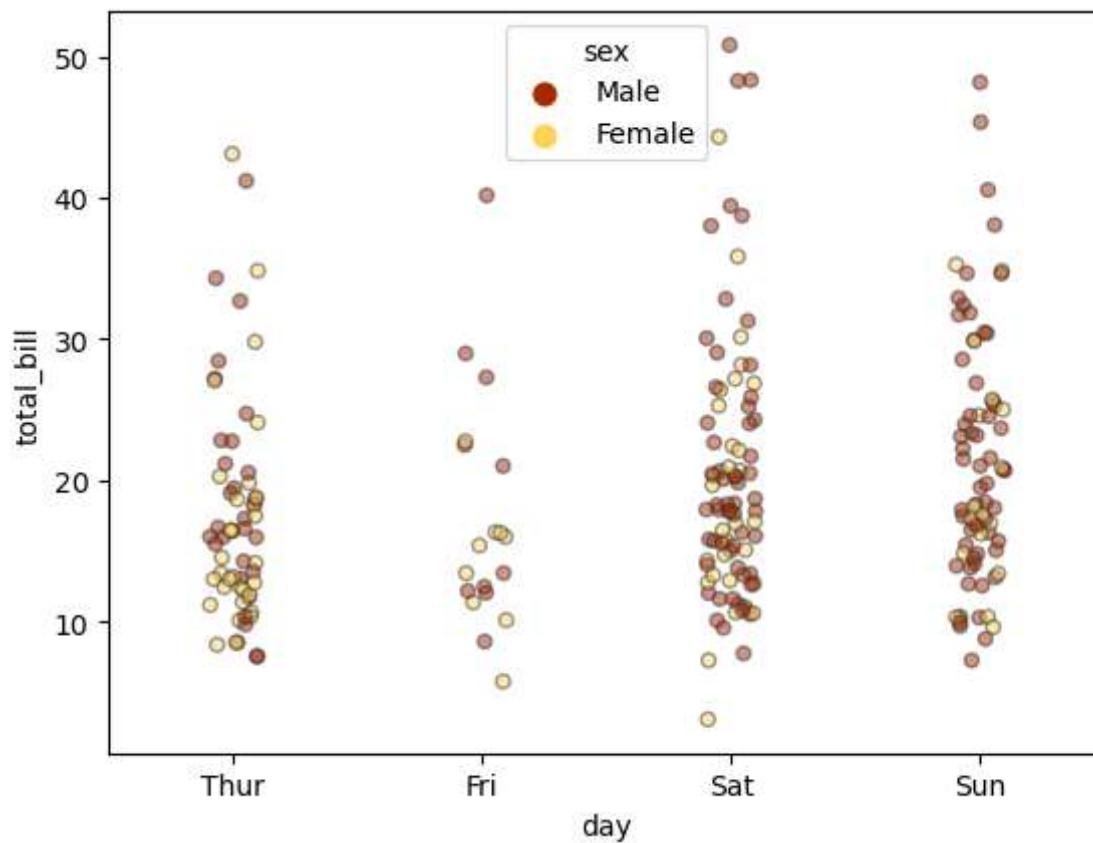
```
In [139]: sns.stripplot(x="day", y="total_bill", data = df, hue = "sex", palette = "afmhot"
                      linewidth= 1, edgecolor="green", size=10)
plt.show()
```



```
In [143]: sns.stripplot(x="day", y="total_bill", data = df, hue = "sex", palette = "afmhot"
                      linewidth= 1, edgecolor="green",marker="<")
plt.show()
```

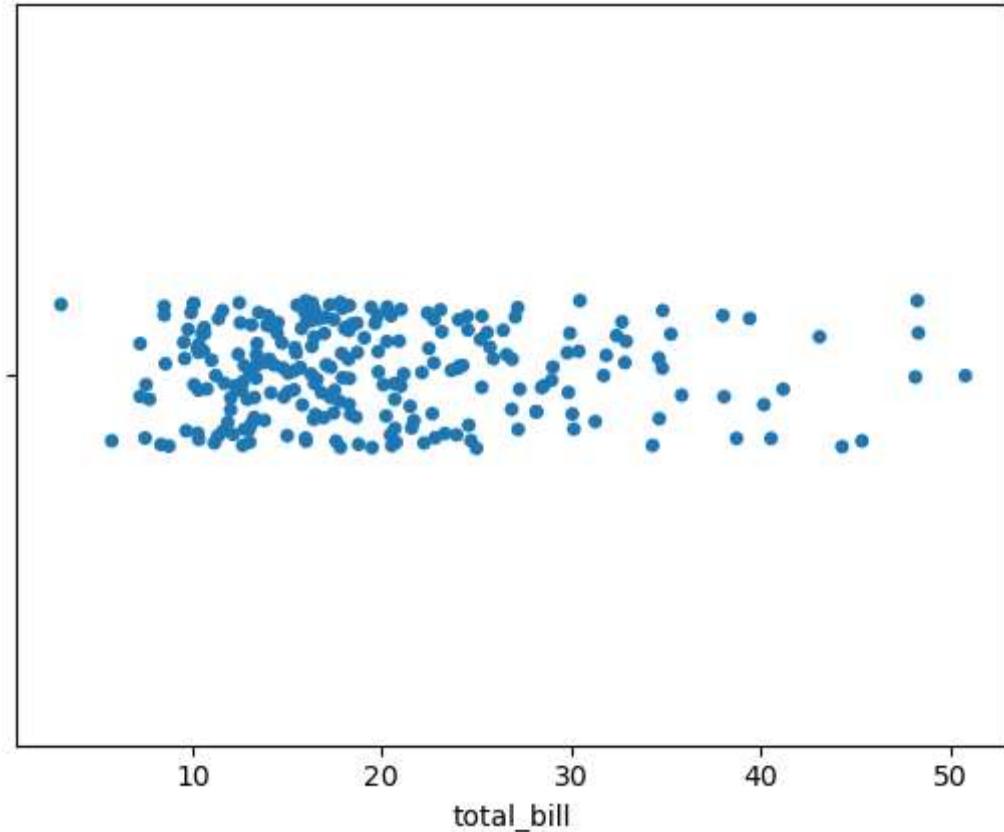


```
In [147]: sns.stripplot(x="day", y="total_bill", data = df, hue = "sex", palette = "afmhot"
                      linewidth= 1, alpha=0.5)
plt.show()
```

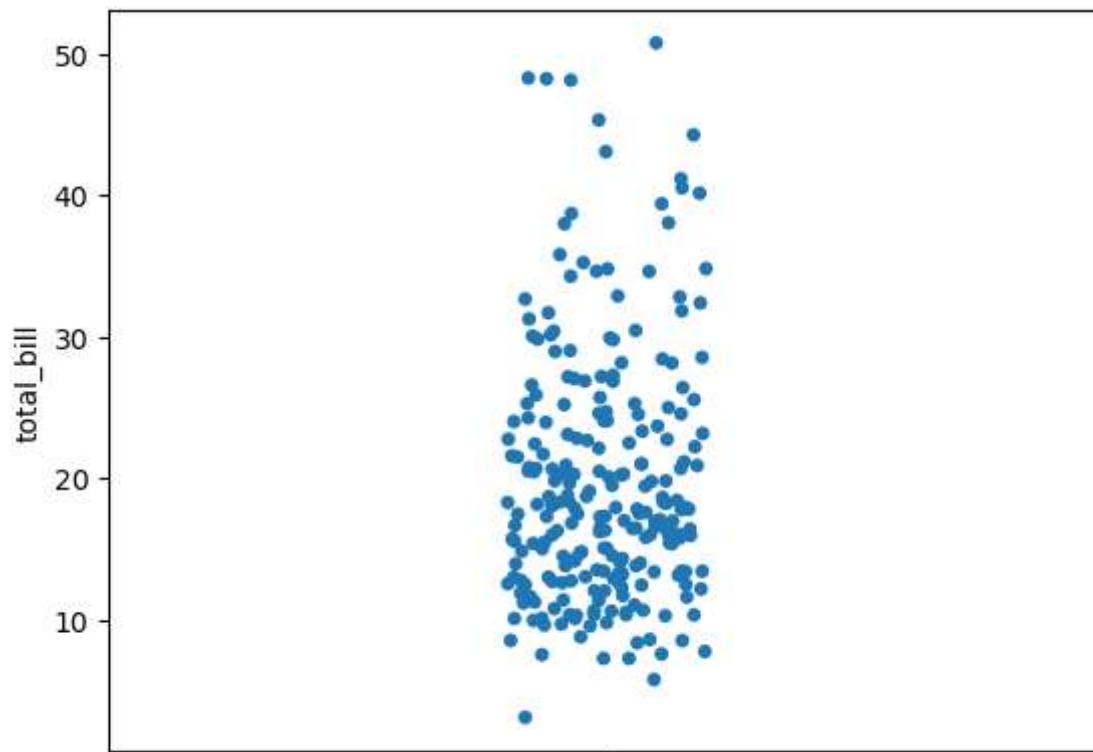


## Drawing a single strip plot

```
In [149]: sns.stripplot(x=df['total_bill'])
plt.show()
```



```
In [150]: sns.stripplot(y=df['total_bill'])
plt.show()
```



# Box Plot in Seaborn

```
In [151]: import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
```

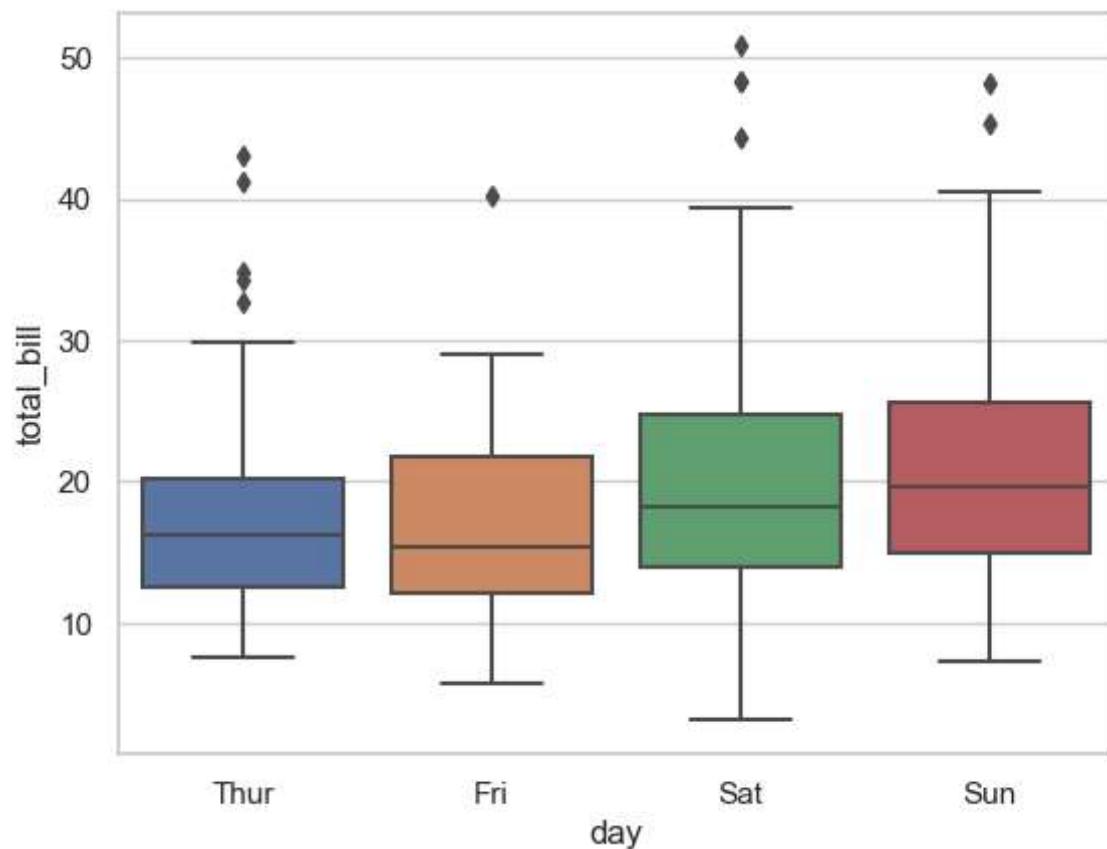
```
In [152]: df = sns.load_dataset("tips")
df
```

Out[152]:

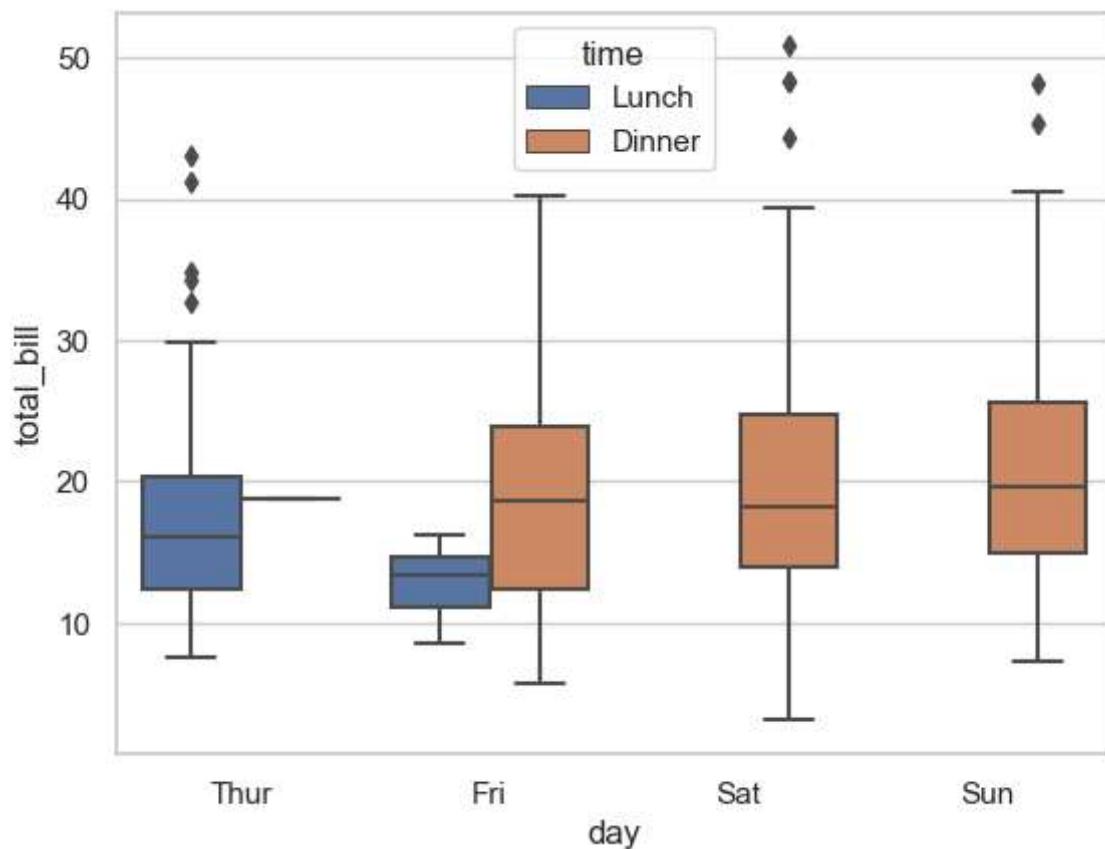
	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4
...	...	...	...	...	...	...	...
239	29.03	5.92	Male	No	Sat	Dinner	3
240	27.18	2.00	Female	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

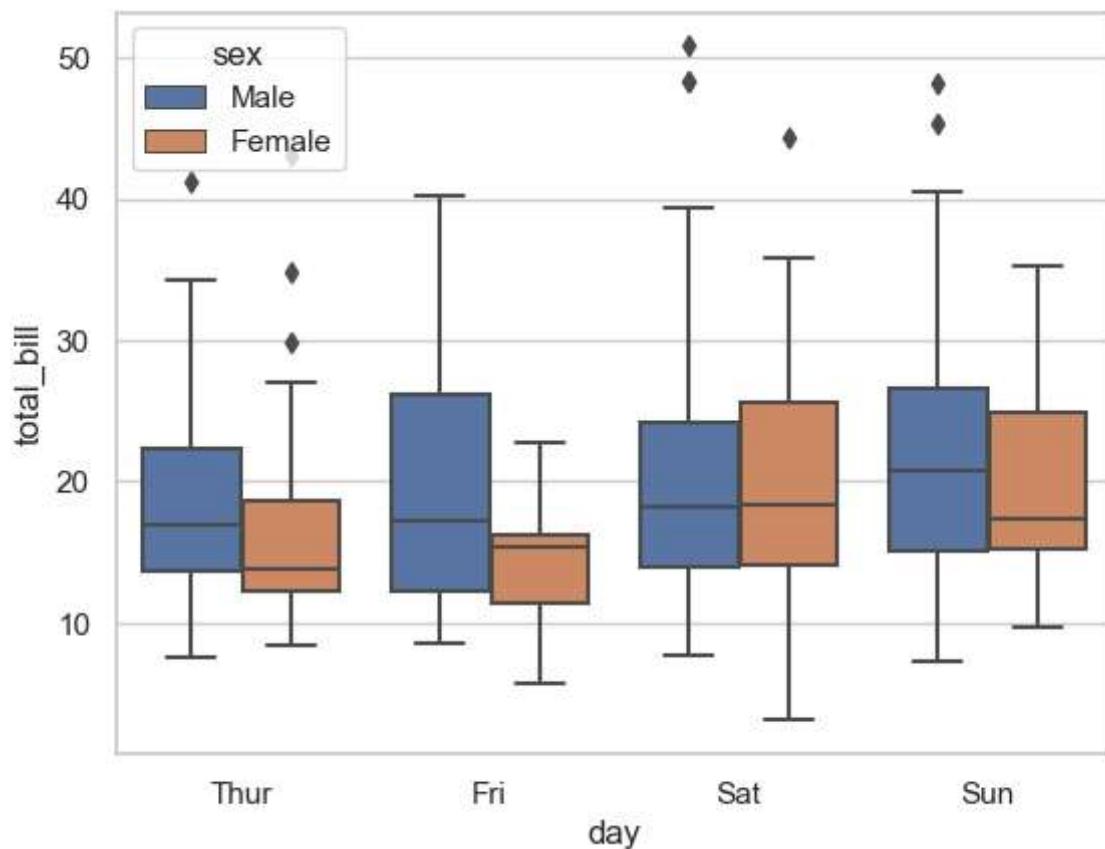
```
In [153]: sns.set(style="whitegrid")
sns.boxplot(x="day",y="total_bill",data = df)
plt.show()
```



```
In [154]: sns.set(style="whitegrid")
sns.boxplot(x="day",y="total_bill",data = df,hue="time")
plt.show()
```

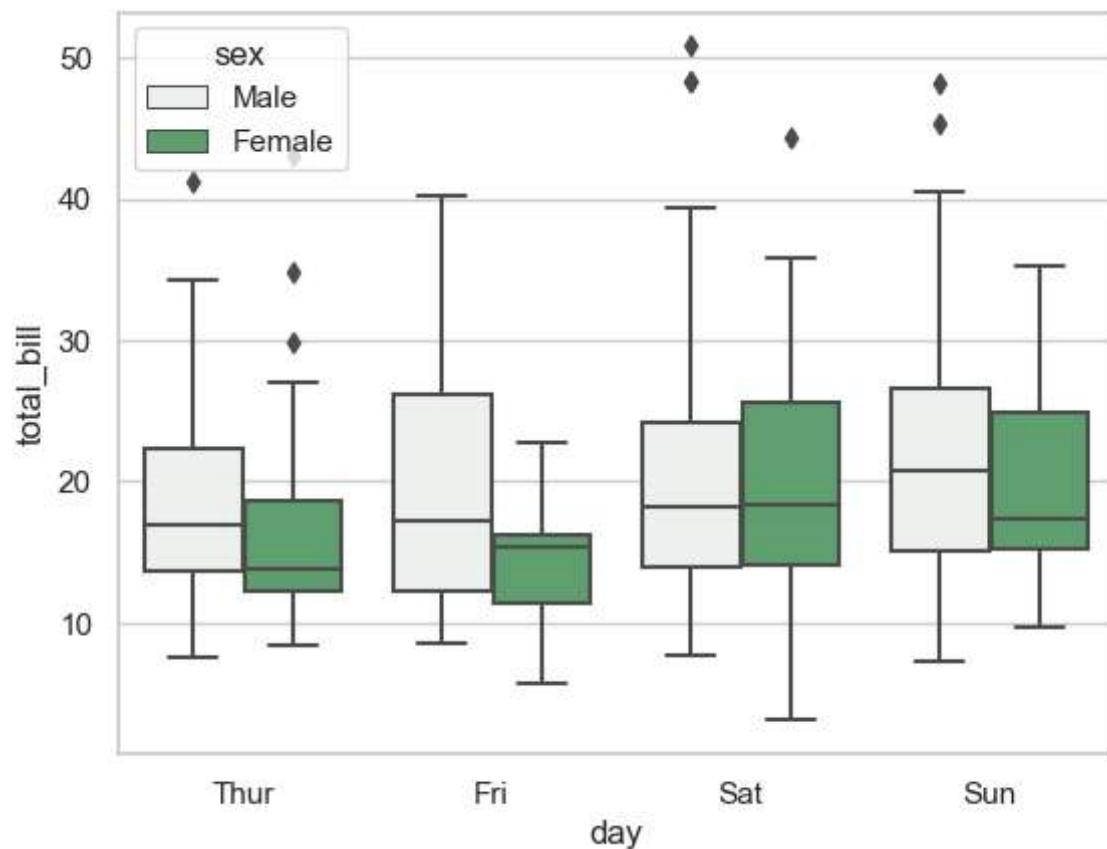


```
In [155]: sns.set(style="whitegrid")
sns.boxplot(x="day",y="total_bill",data = df,hue="sex")
plt.show()
```

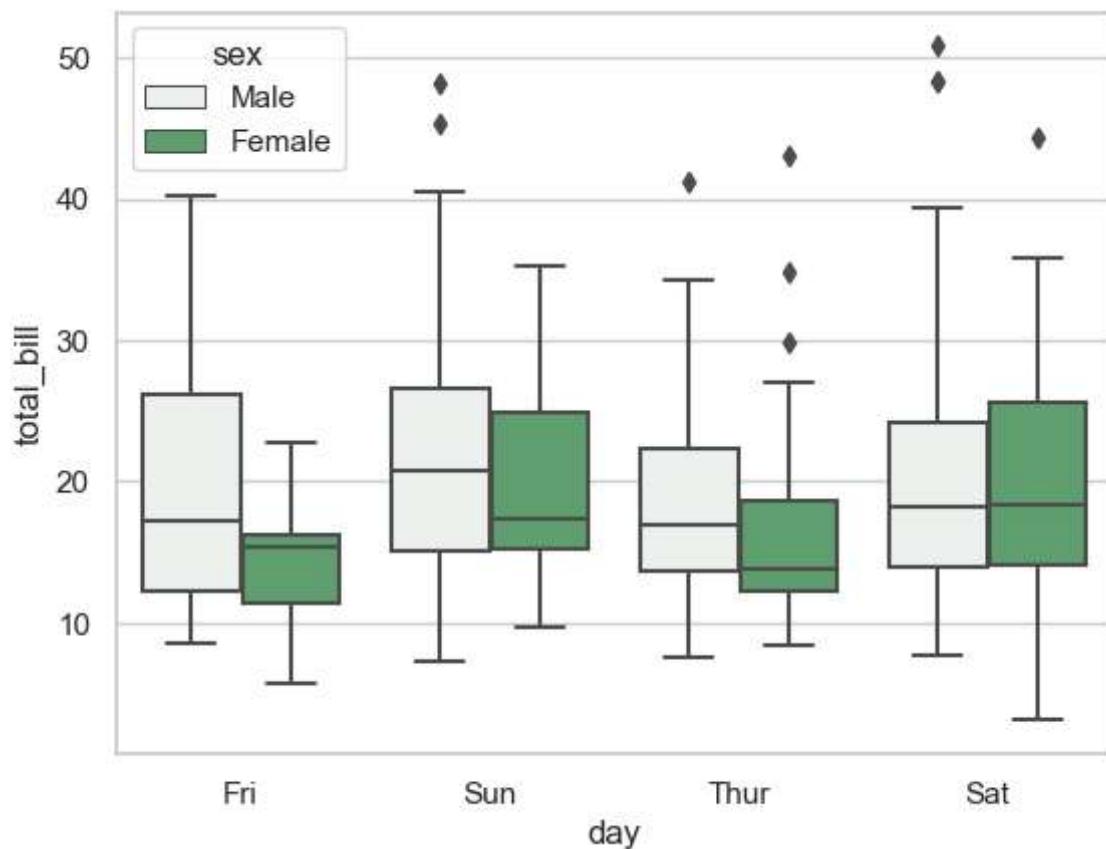


changing the color

```
In [156]: sns.set(style="whitegrid")
sns.boxplot(x="day",y="total_bill",data = df,hue="sex",color="g")
plt.show()
```

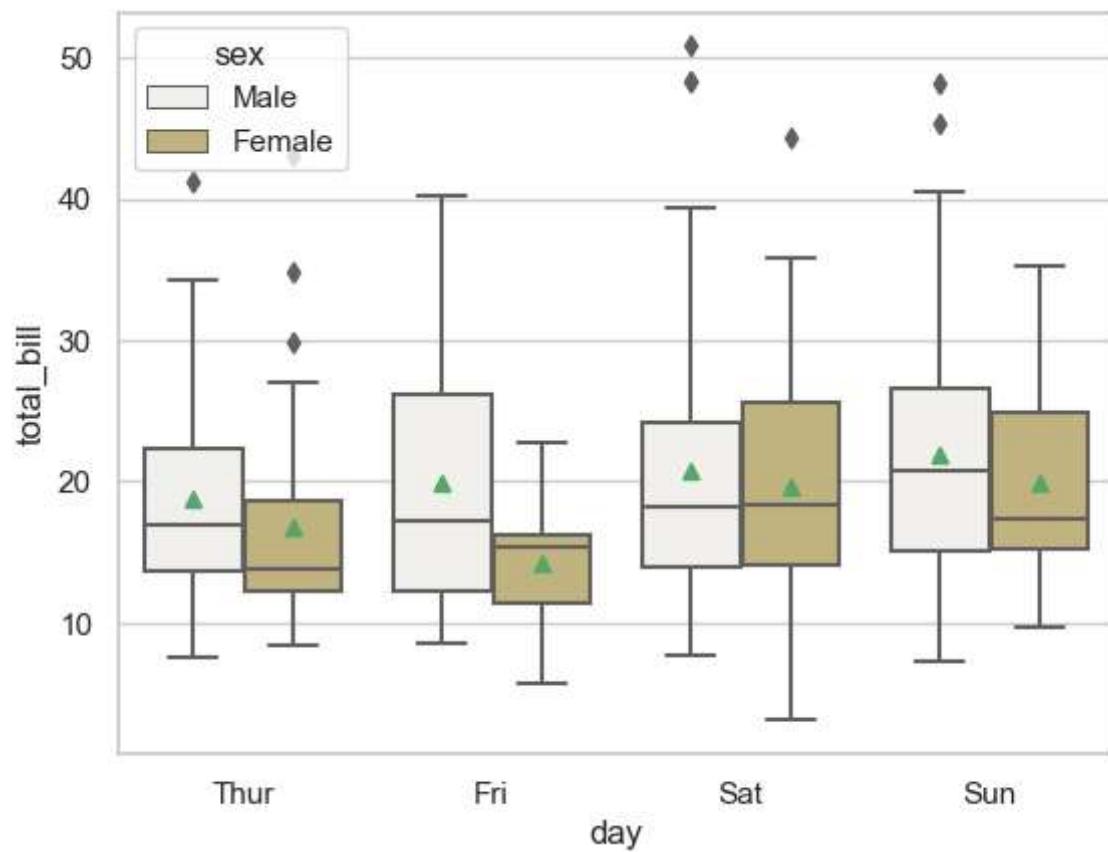


```
In [157]: sns.set(style="whitegrid")
sns.boxplot(x="day",y="total_bill",data = df,hue="sex",color="g",order=["Fri",
plt.show()
```

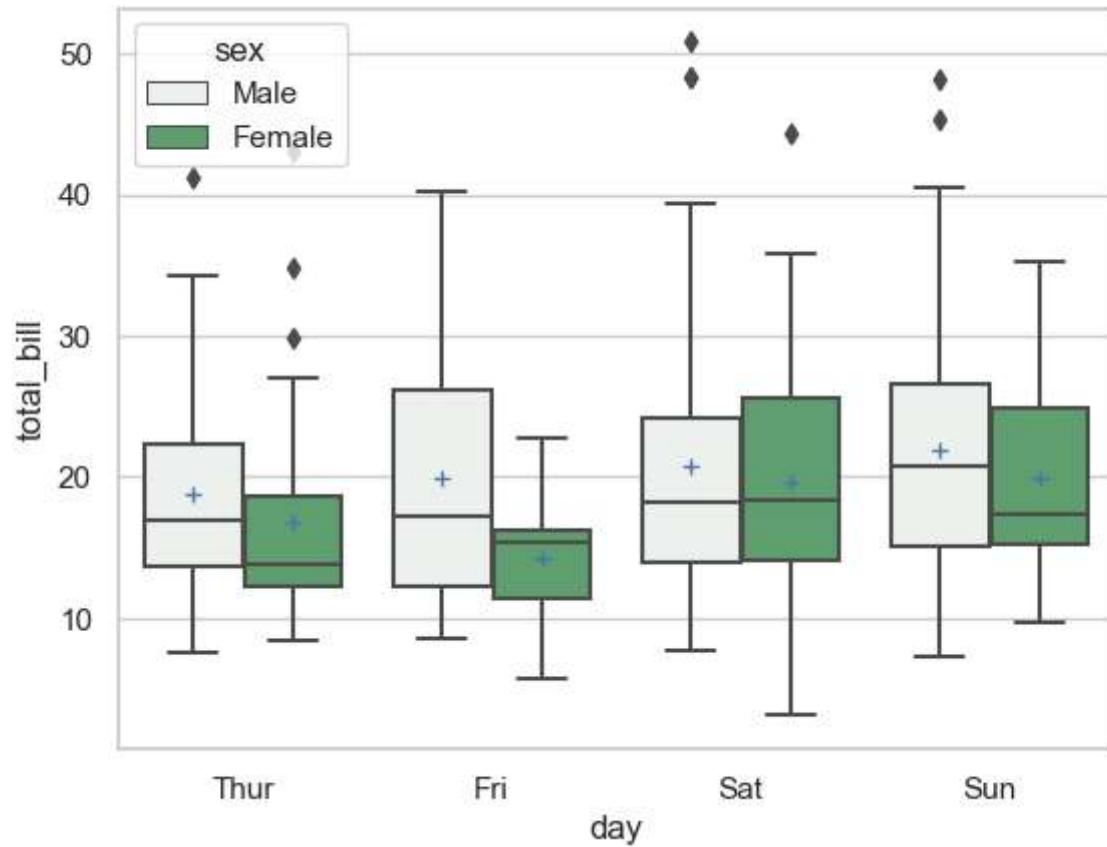


for showing the mean

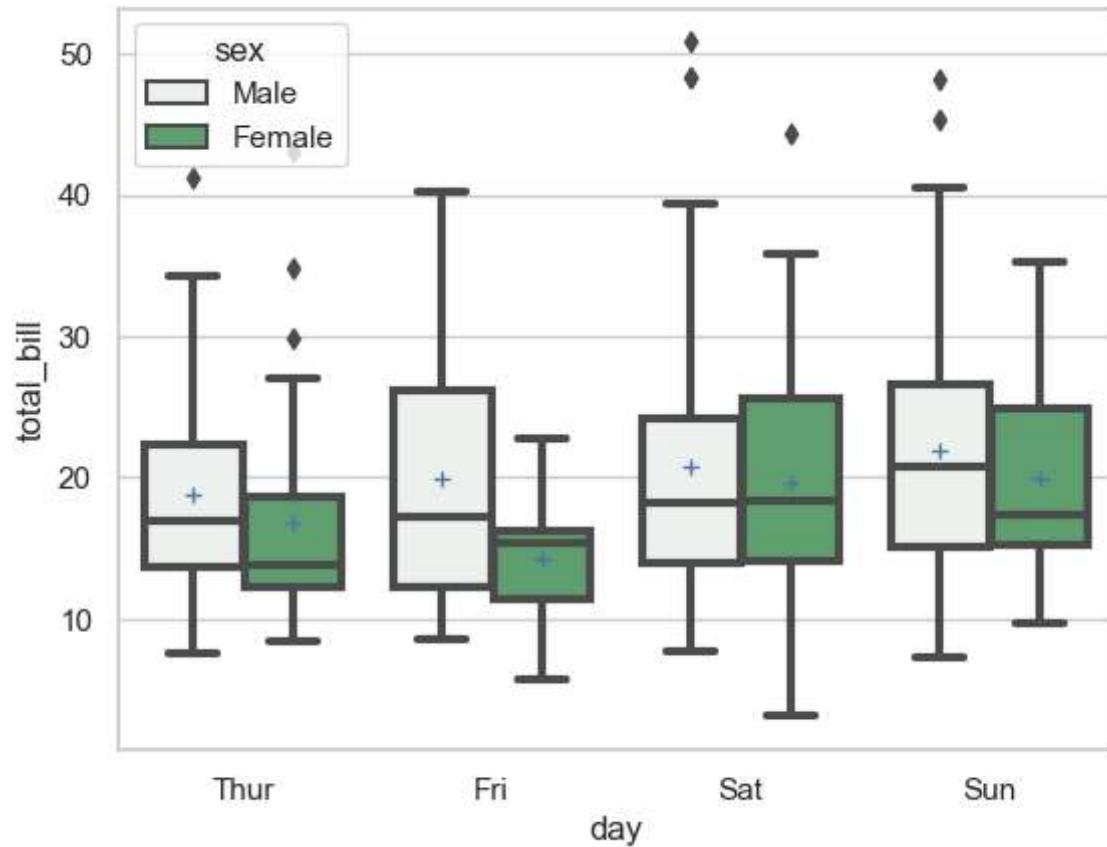
```
In [159]: sns.set(style="whitegrid")
sns.boxplot(x="day",y="total_bill",data = df,hue="sex",color="y",showmeans=True)
plt.show()
```



```
In [161]: sns.set(style="whitegrid")
sns.boxplot(x="day",y="total_bill",data = df,hue="sex",color="g",
            showmeans=True,meanprops={"marker": "+", "markeredgecolor": "b"})
plt.show()
```

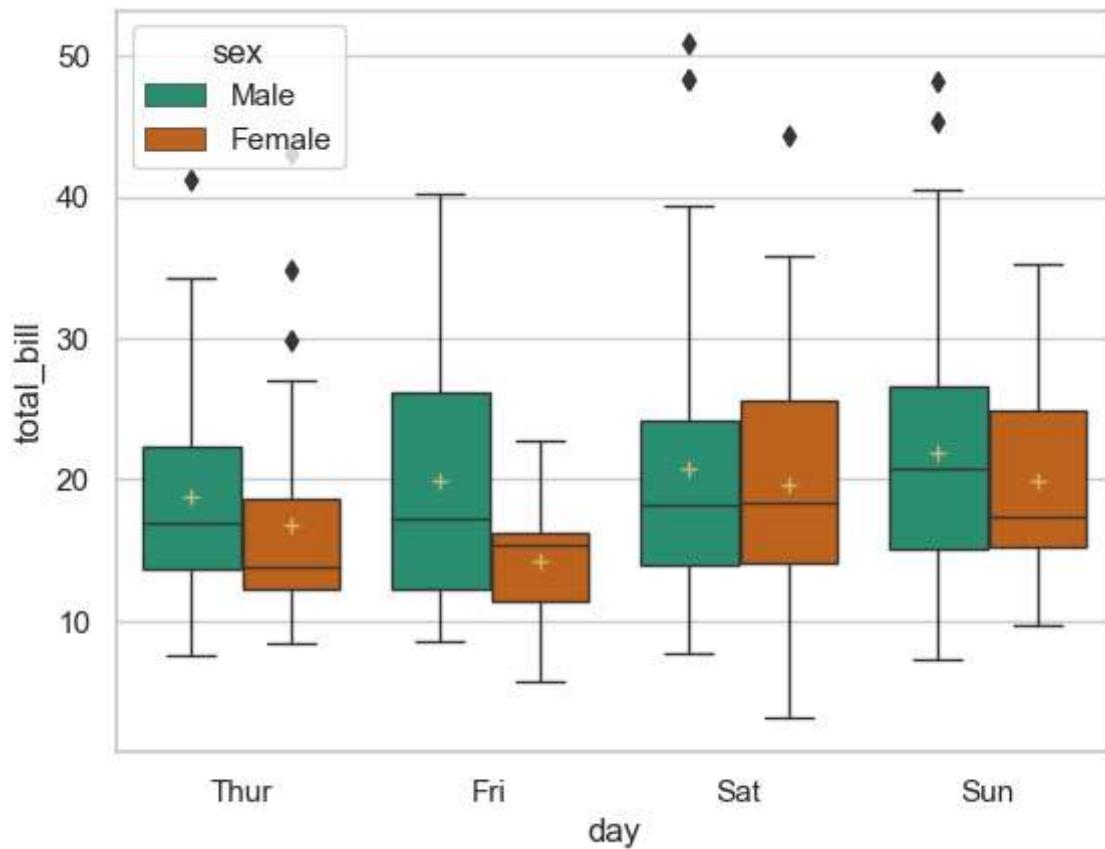


```
In [162]: sns.set(style="whitegrid")
sns.boxplot(x="day",y="total_bill",data = df,hue="sex",color="g",
            showmeans=True,meanprops={"marker": "+", "markeredgecolor": "b"}, 
            linewidth=3)
plt.show()
```



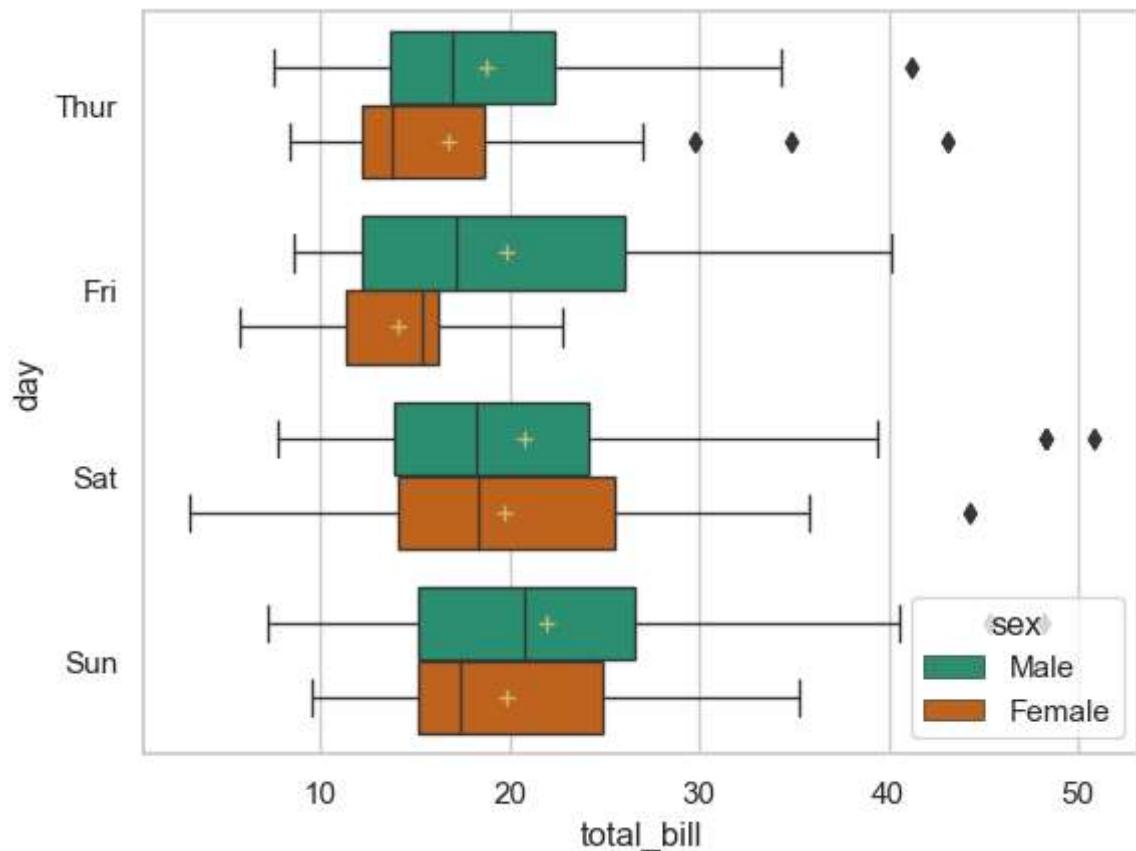
changing the color

```
In [169]: sns.set(style="whitegrid")
sns.boxplot(x="day",y="total_bill",data = df,hue="sex",
            showmeans=True,meanprops={"marker": "+", "markeredgecolor": "y"}, 
            linewidth=1,palette="Dark2")
plt.show()
```

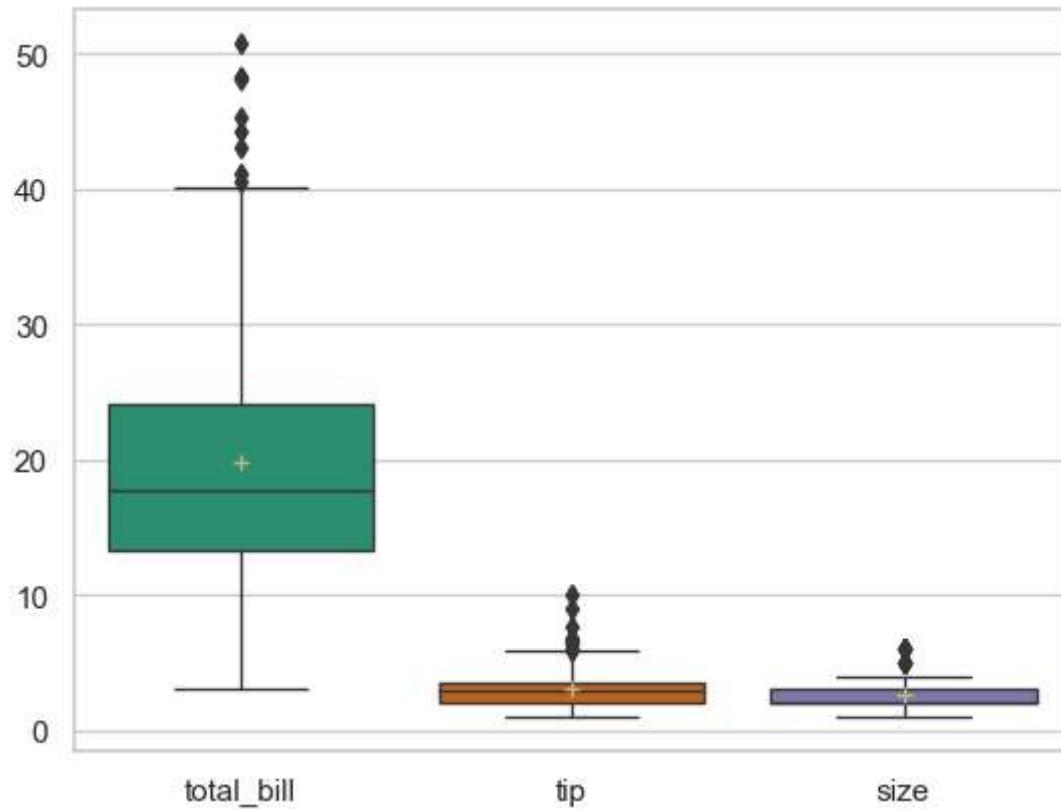


changing the orientation

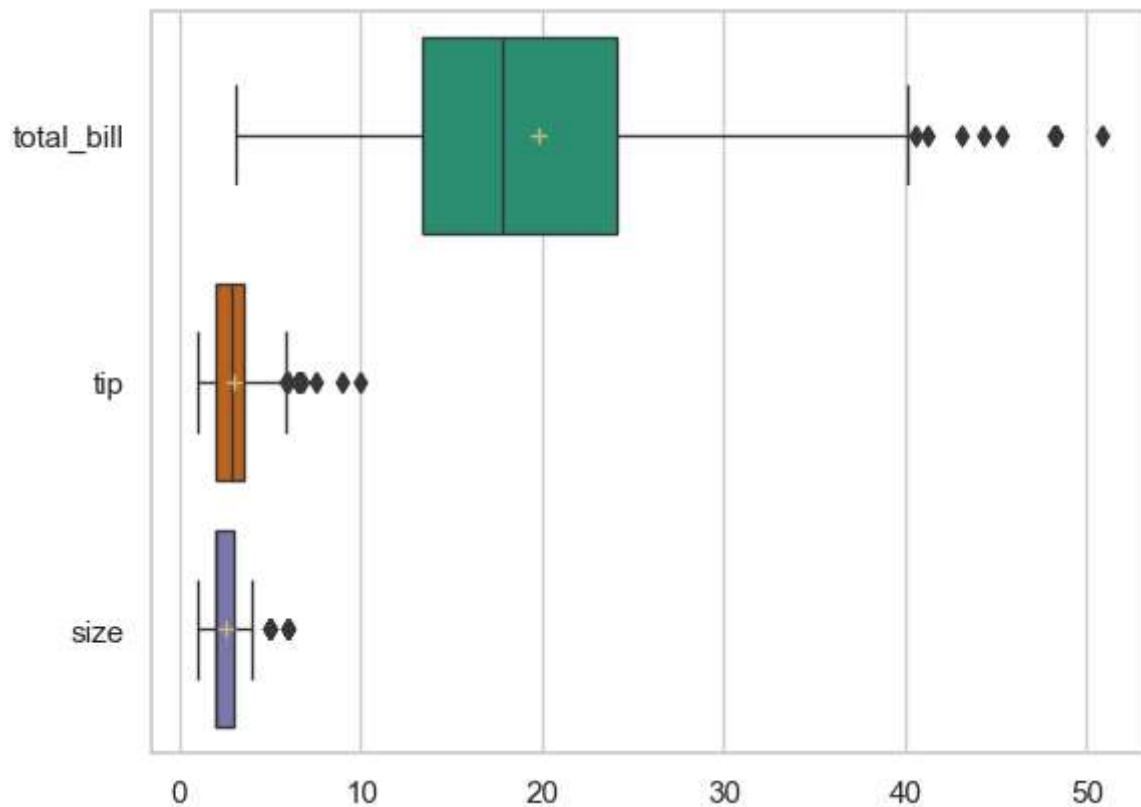
```
In [170]: sns.set(style="whitegrid")
sns.boxplot(y="day",x="total_bill",data = df,hue="sex",
            showmeans=True,meanprops={"marker": "+", "markeredgecolor": "y"},
            linewidth=1,palette="Dark2")
plt.show()
```



```
In [171]: sns.set(style="whitegrid")
sns.boxplot(data=df,
            showmeans=True,meanprops={"marker": "+", "markeredgecolor": "y"},
            linewidth=1,palette="Dark2")
plt.show()
```

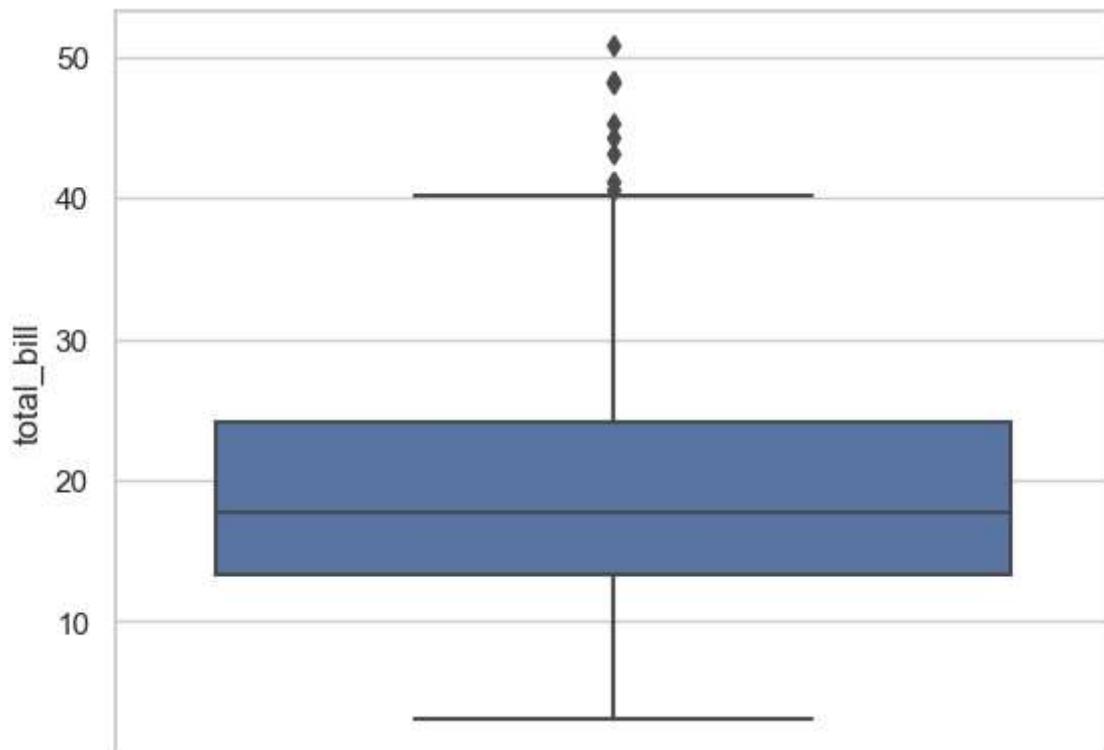


```
In [172]: sns.set(style="whitegrid")
sns.boxplot(data=df,
            showmeans=True,meanprops={"marker": "+", "markeredgecolor": "y",
            linewidth=1,palette="Dark2",orient="h")
plt.show()
```



drawing a single graph

```
In [173]: sns.set(style="whitegrid")
sns.boxplot(y= df['total_bill'])
plt.show()
```



## Factor plot in Seaborn

```
In [174]: import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
```

```
In [175]: df = sns.load_dataset("tips")
```

In [176]: df

Out[176]:

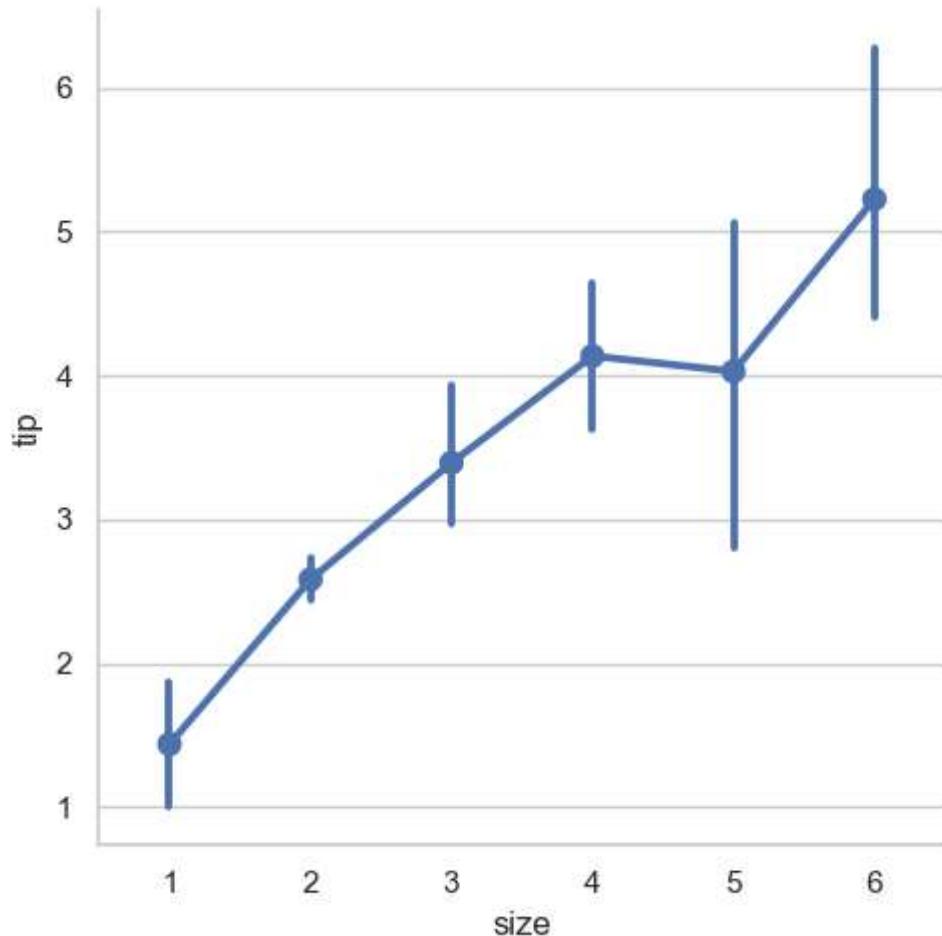
	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4
...	...	...	...	...	...	...	...
239	29.03	5.92	Male	No	Sat	Dinner	3
240	27.18	2.00	Female	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 [179]: sns.factorplot(x="size",y="tip",data=df)
plt.show()
```

C:\ProgramData\Anaconda3\lib\site-packages\seaborn\categorical.py:3717: UserWarning: The `factorplot` function has been renamed to `catplot`. The original name will be removed in a future release. Please update your code. Note that the default `kind` in `factorplot` (`'point'`) has changed `'strip'` in `catplot`.

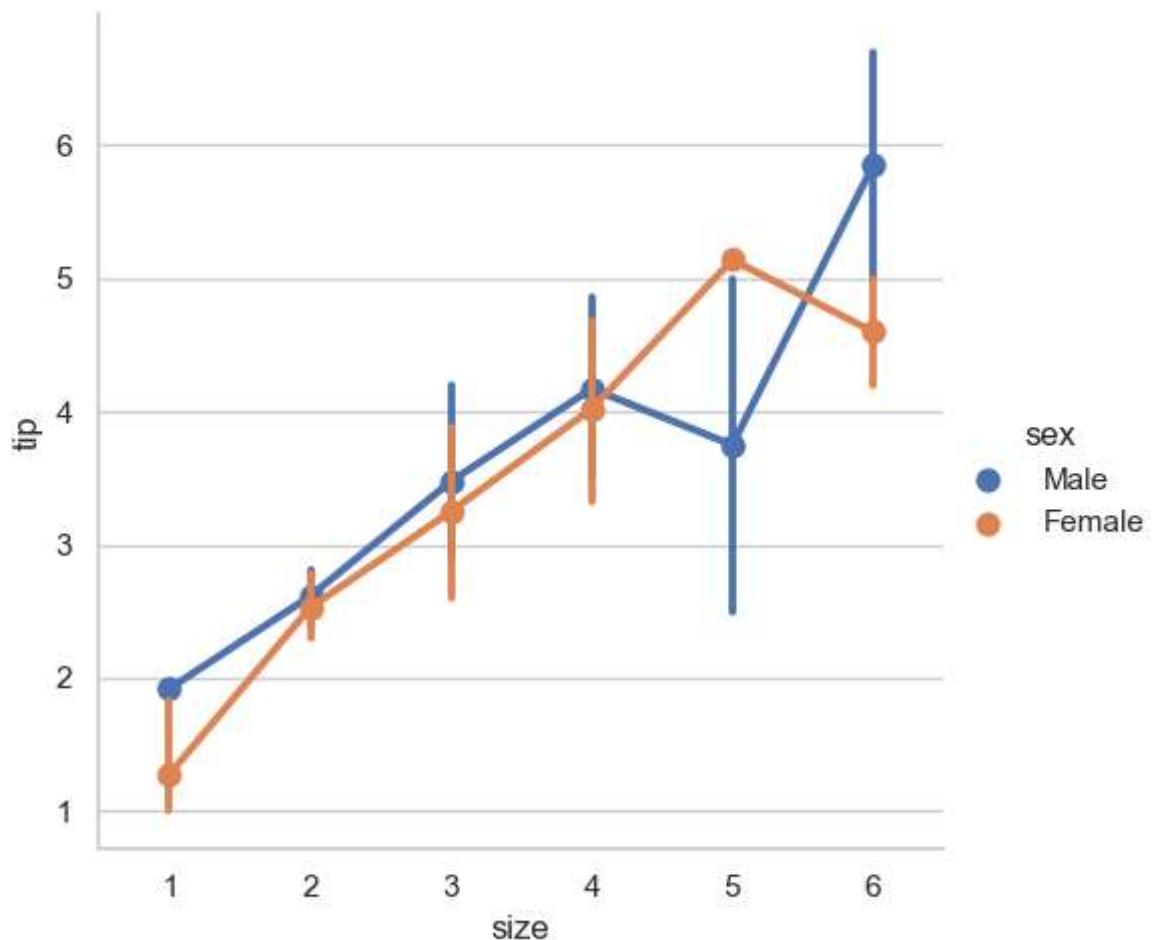
```
warnings.warn(msg)
```



```
In [180]: sns.factorplot(x="size",y="tip",data=df,hue="sex")
plt.show()
```

C:\ProgramData\Anaconda3\lib\site-packages\seaborn\categorical.py:3717: UserWarning: The `factorplot` function has been renamed to `catplot`. The original name will be removed in a future release. Please update your code. Note that the default `kind` in `factorplot` (`'point'`) has changed `'strip'` in `catplot`.

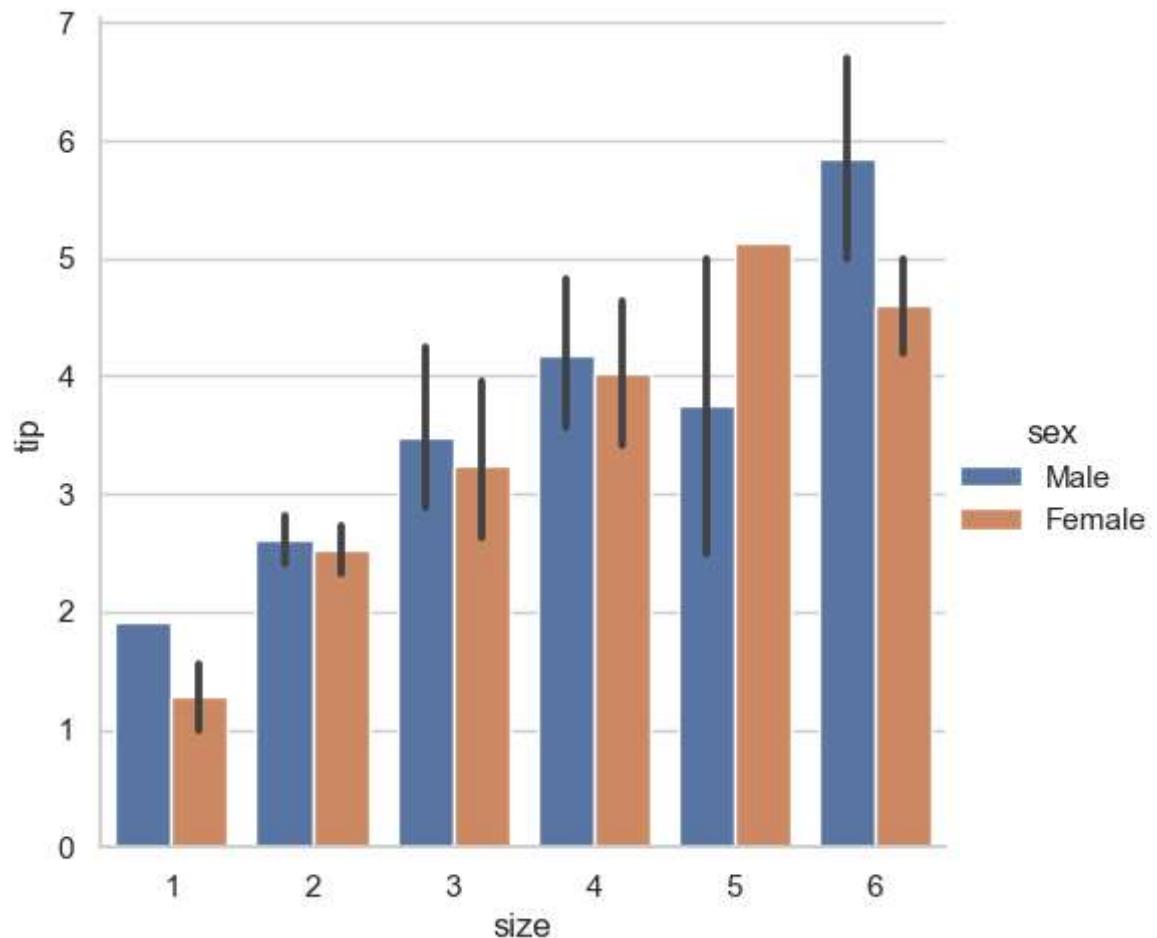
```
warnings.warn(msg)
```



```
In [181]: sns.factorplot(x="size",y="tip",data=df,hue="sex",kind="bar")
plt.show()
```

C:\ProgramData\Anaconda3\lib\site-packages\seaborn\categorical.py:3717: UserWarning: The `factorplot` function has been renamed to `catplot`. The original name will be removed in a future release. Please update your code. Note that the default `kind` in `factorplot` (`'point'`) has changed `'strip'` in `catplot`.

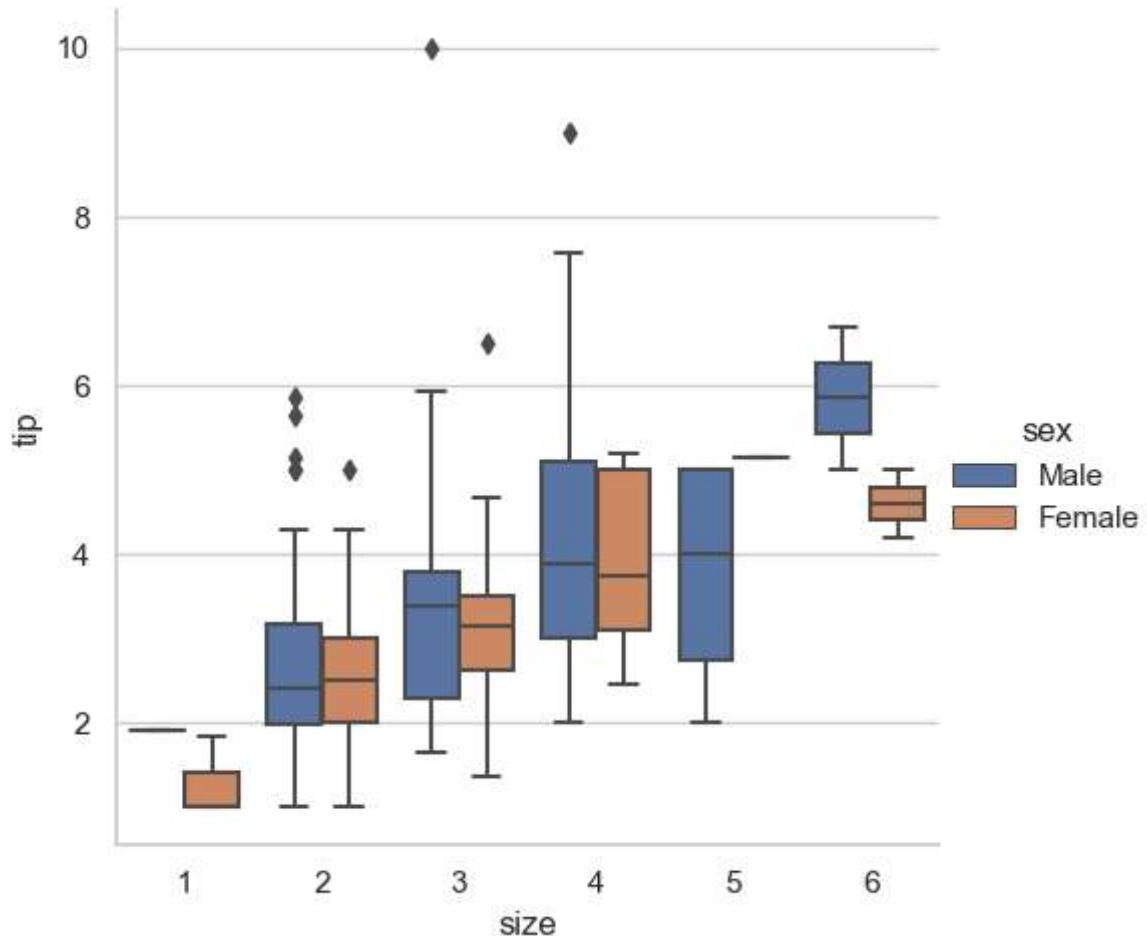
```
warnings.warn(msg)
```



```
In [182]: sns.factorplot(x="size",y="tip",data=df,hue="sex",kind="box")
plt.show()
```

C:\ProgramData\Anaconda3\lib\site-packages\seaborn\categorical.py:3717: UserWarning: The `factorplot` function has been renamed to `catplot`. The original name will be removed in a future release. Please update your code. Note that the default `kind` in `factorplot` (`'point'`) has changed `'strip'` in `catplot`.

```
warnings.warn(msg)
```



## Cat plot

```
In [189]: import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
```

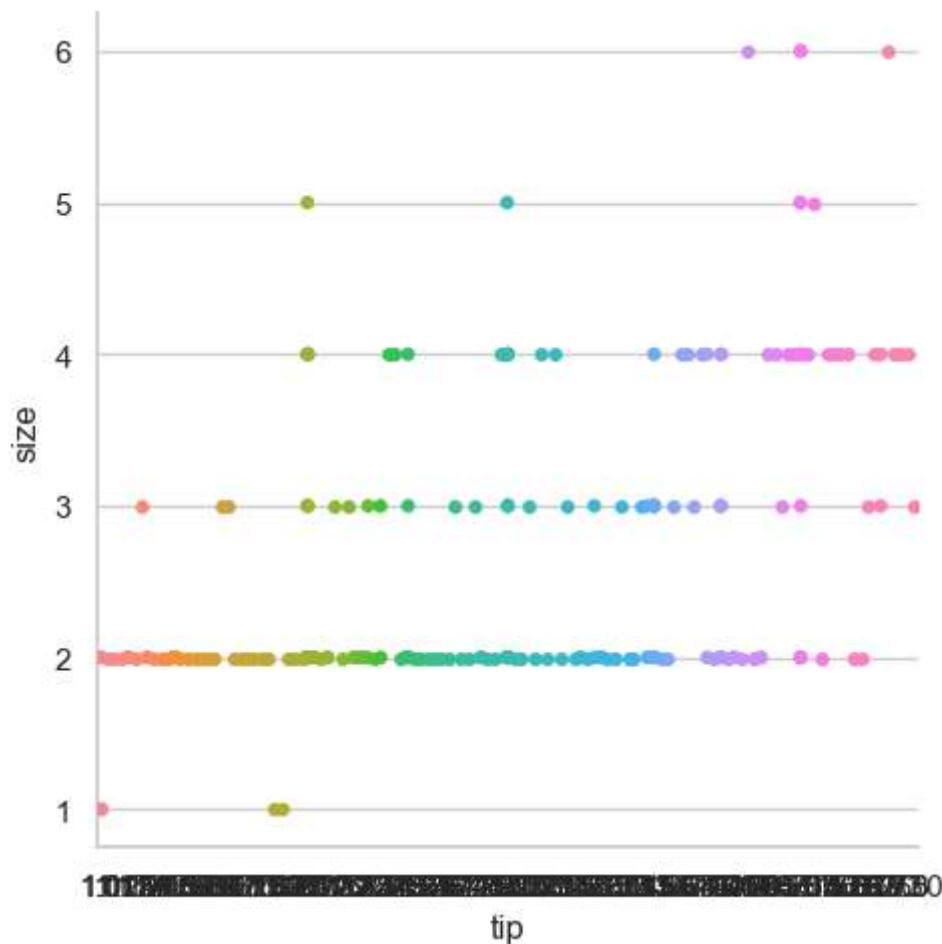
```
In [190]: df = sns.load_dataset("tips")
```

In [191]: df

Out[191]:

	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 [193]: sns.catplot(x='tip',y='size',data= df)  
plt.show()

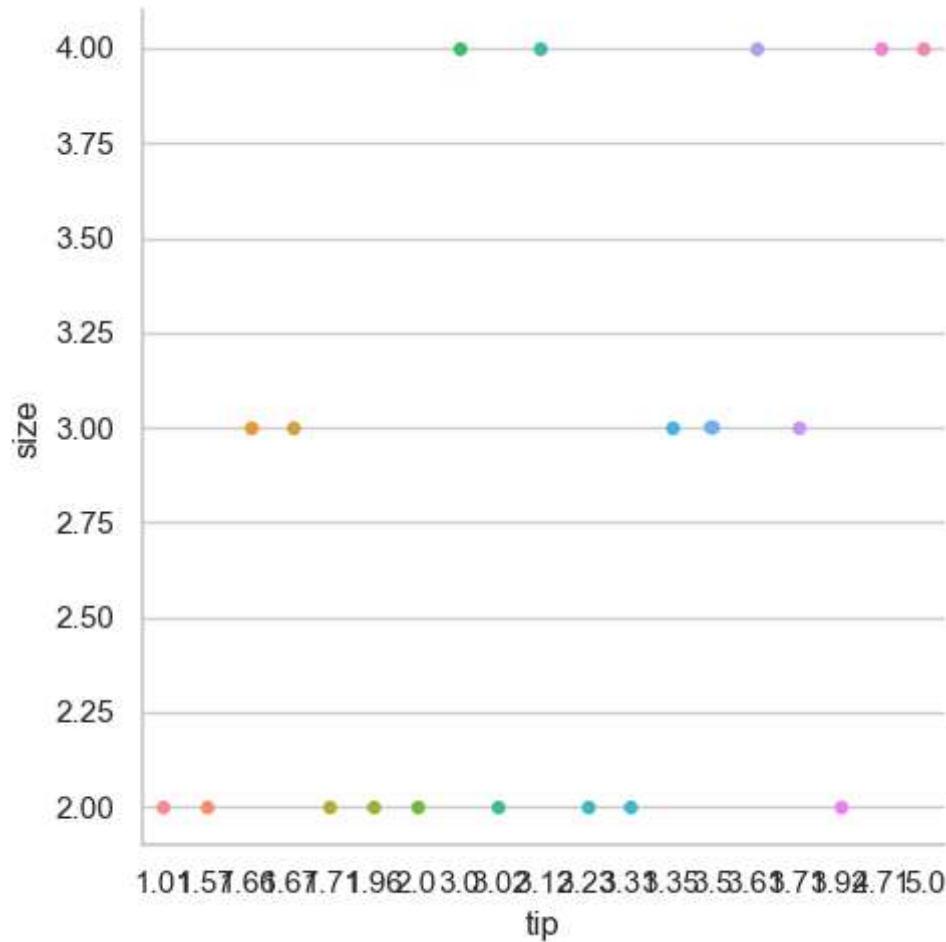
```
In [194]: df = sns.load_dataset("tips").head(20)
```

```
In [195]: df
```

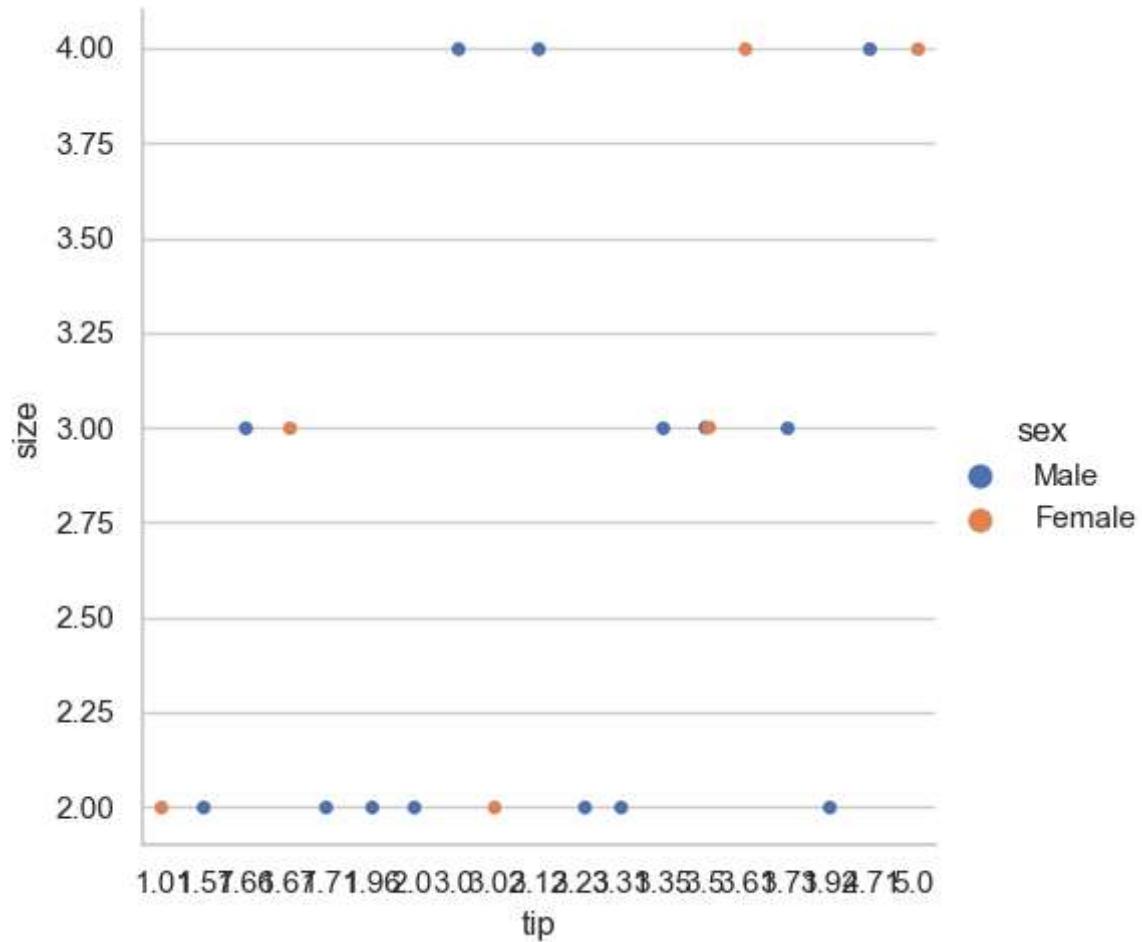
```
Out[195]:
```

	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
5	25.29	4.71	Male	No	Sun	Dinner	4
6	8.77	2.00	Male	No	Sun	Dinner	2
7	26.88	3.12	Male	No	Sun	Dinner	4
8	15.04	1.96	Male	No	Sun	Dinner	2
9	14.78	3.23	Male	No	Sun	Dinner	2
10	10.27	1.71	Male	No	Sun	Dinner	2
11	35.26	5.00	Female	No	Sun	Dinner	4
12	15.42	1.57	Male	No	Sun	Dinner	2
13	18.43	3.00	Male	No	Sun	Dinner	4
14	14.83	3.02	Female	No	Sun	Dinner	2
15	21.58	3.92	Male	No	Sun	Dinner	2
16	10.33	1.67	Female	No	Sun	Dinner	3
17	16.29	3.71	Male	No	Sun	Dinner	3
18	16.97	3.50	Female	No	Sun	Dinner	3
19	20.65	3.35	Male	No	Sat	Dinner	3

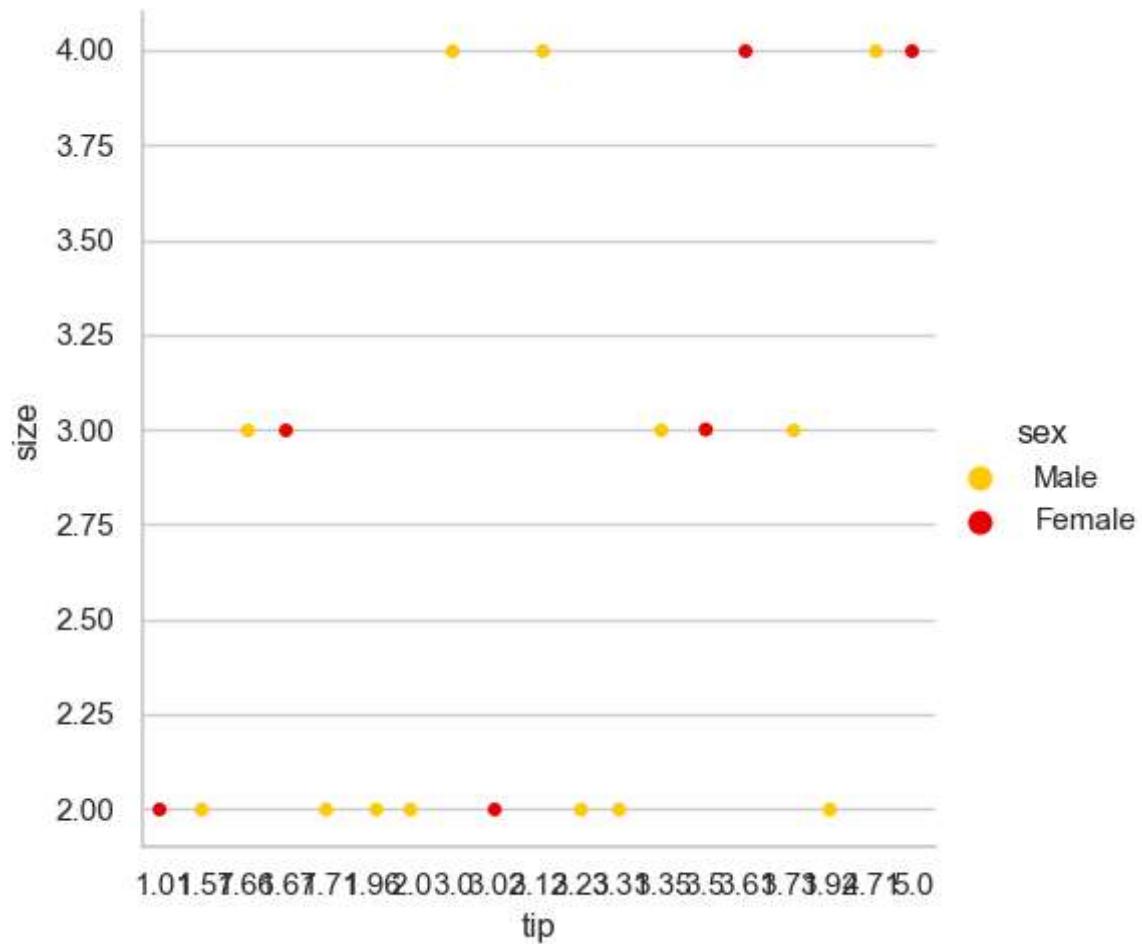
```
In [196]: sns.catplot(x='tip',y='size',data= df)
plt.show()
```



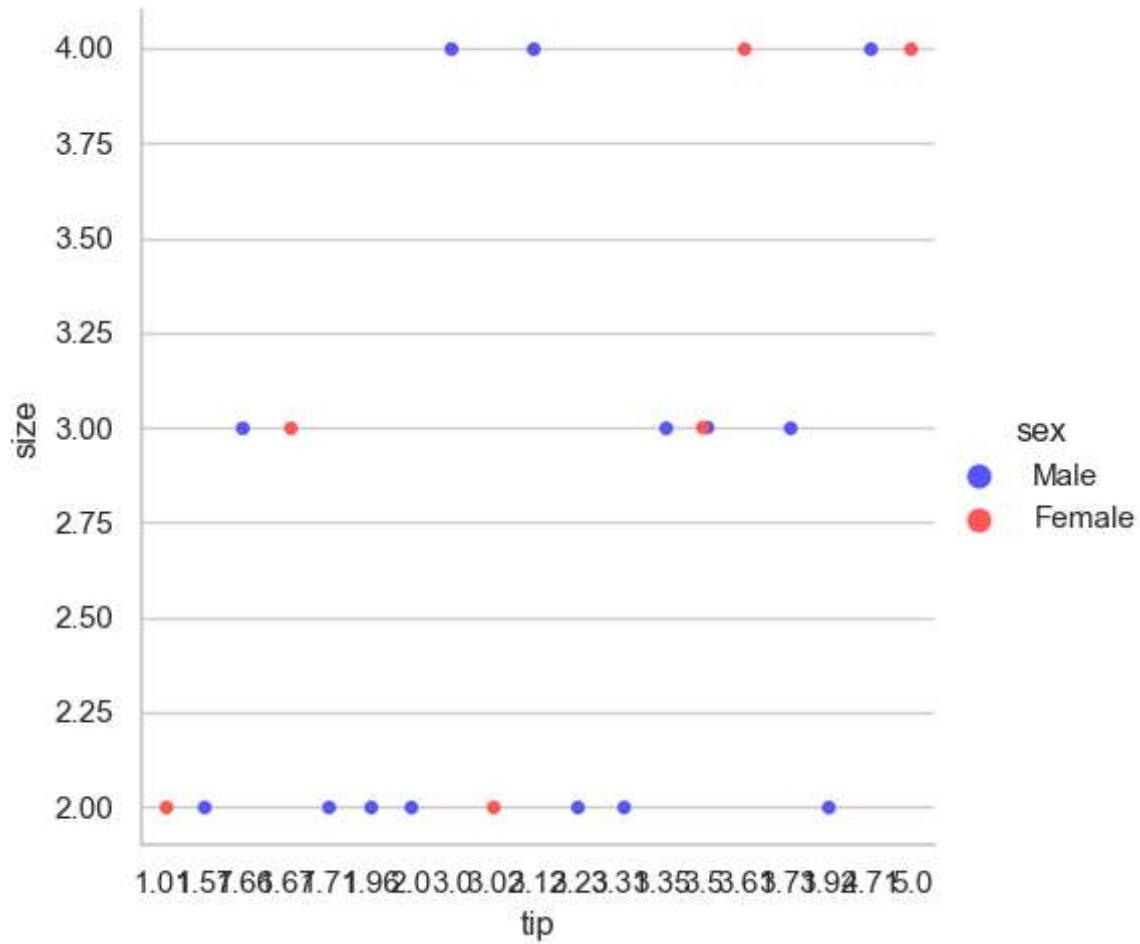
```
In [197]: sns.catplot(x='tip',y='size',data= df,hue="sex")
plt.show()
```



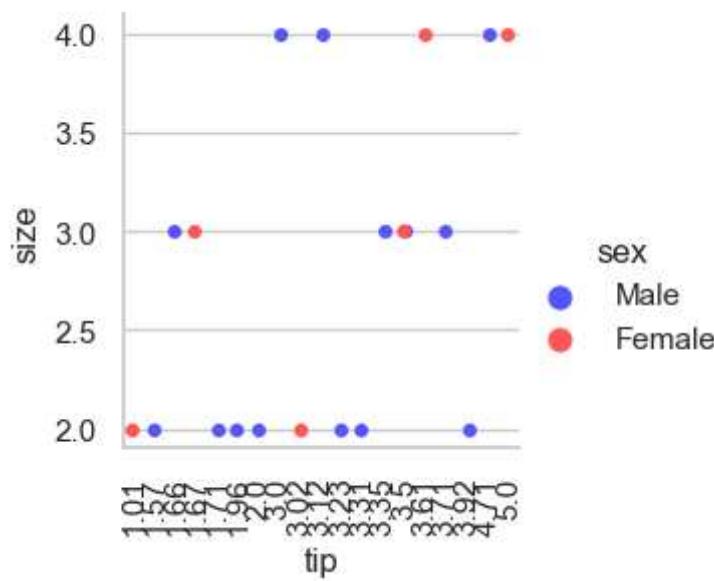
```
In [199]: sns.catplot(x='tip',y='size',data= df,hue="sex",palette="hot_r")
plt.show()
```



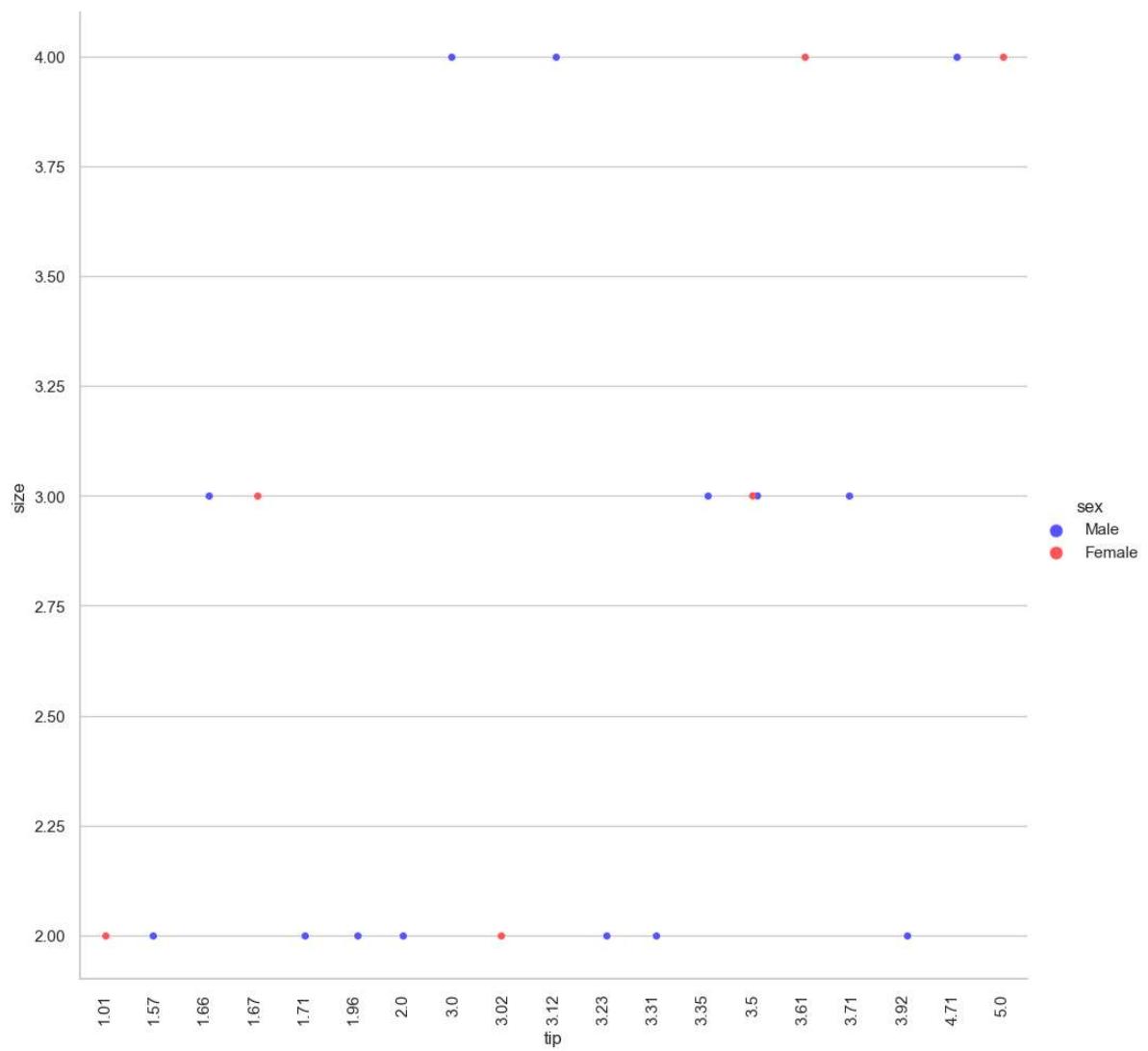
```
In [203]: sns.catplot(x='tip',y='size',data= df,hue="sex",palette="seismic")
plt.show()
```



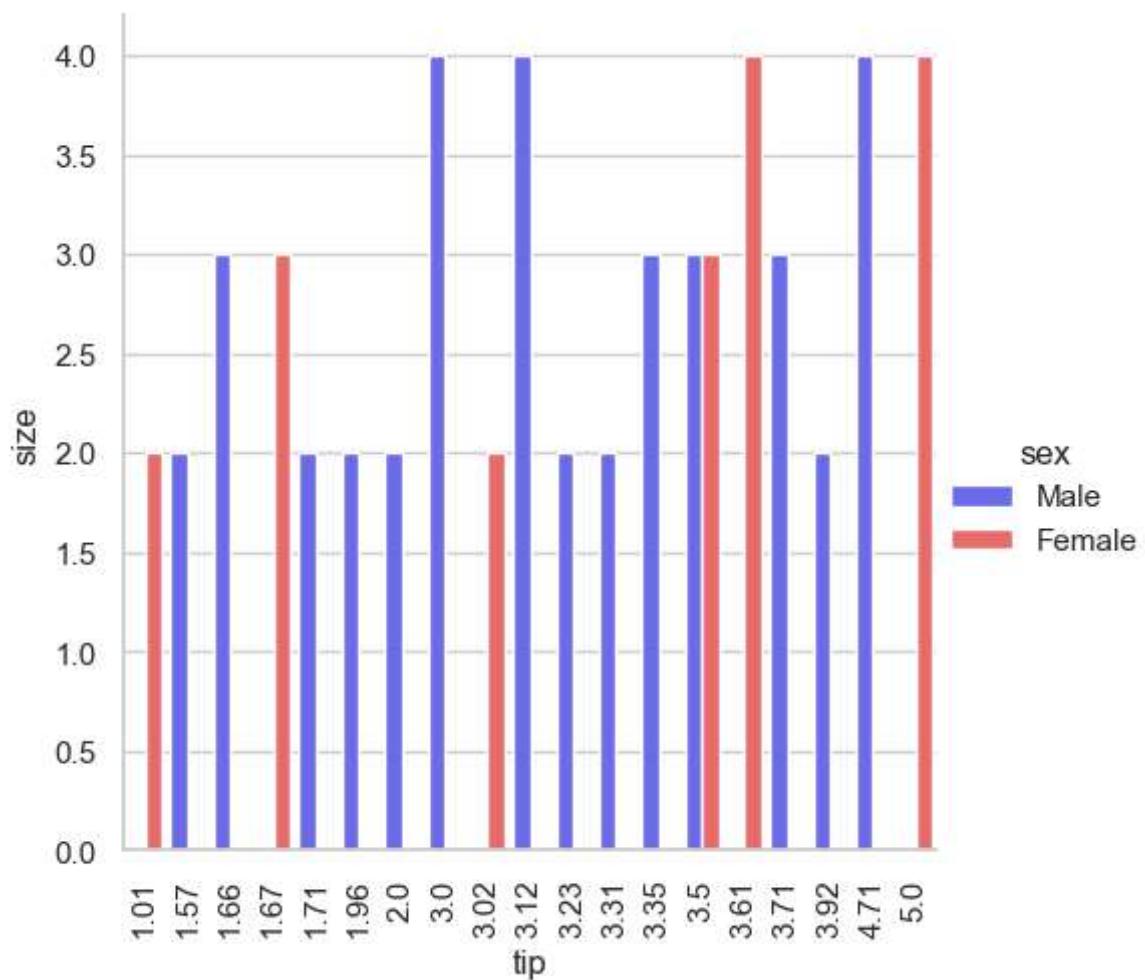
```
In [206]: sns.catplot(x='tip',y='size',data= df,hue="sex",palette="seismic", height=3)
plt.xticks(rotation="vertical")
plt.show()
```



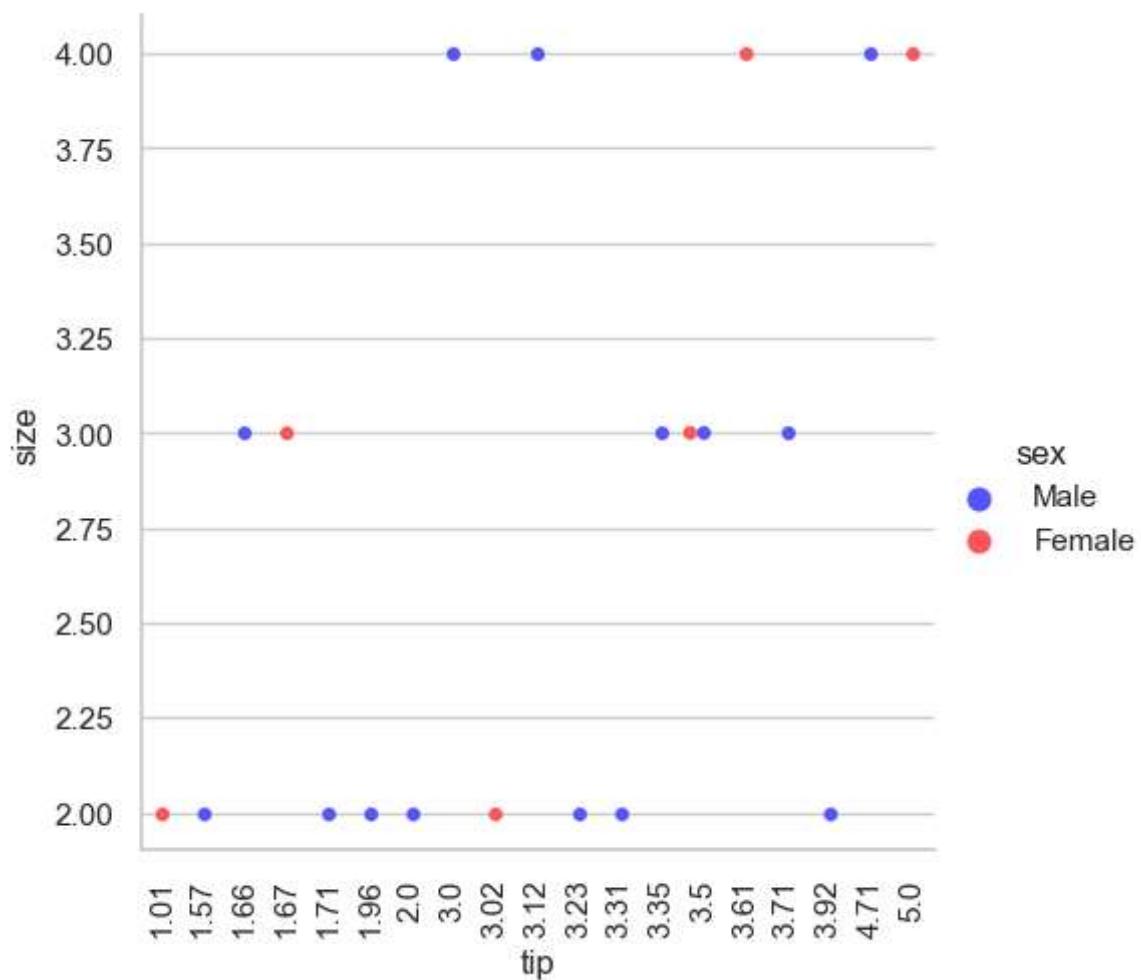
```
In [207]: sns.catplot(x='tip',y='size',data= df,hue="sex",palette="seismic", height=10)
plt.xticks(rotation="vertical")
plt.show()
```



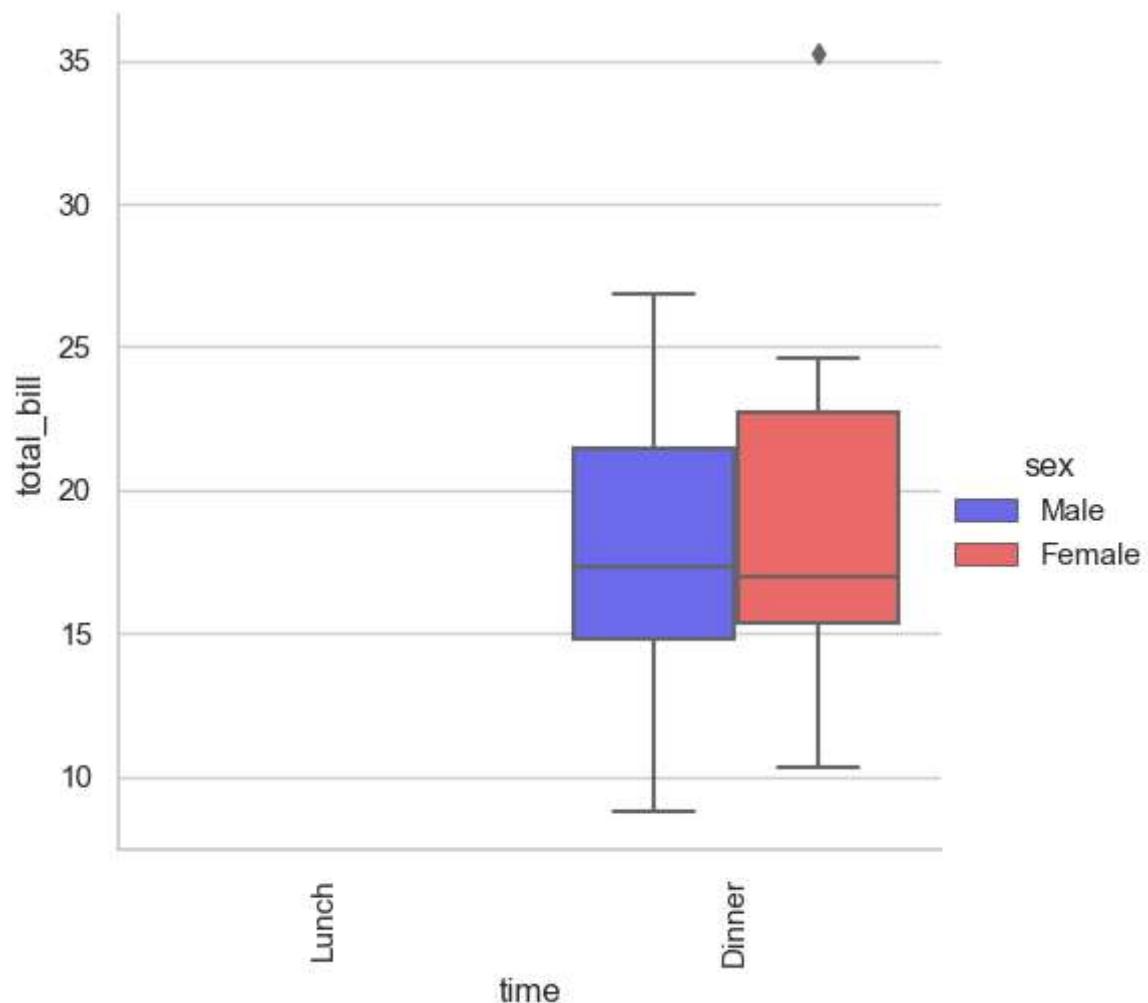
```
In [208]: sns.catplot(x='tip',y='size',data= df,hue="sex",palette="seismic", kind="bar")
plt.xticks(rotation="vertical")
plt.show()
```



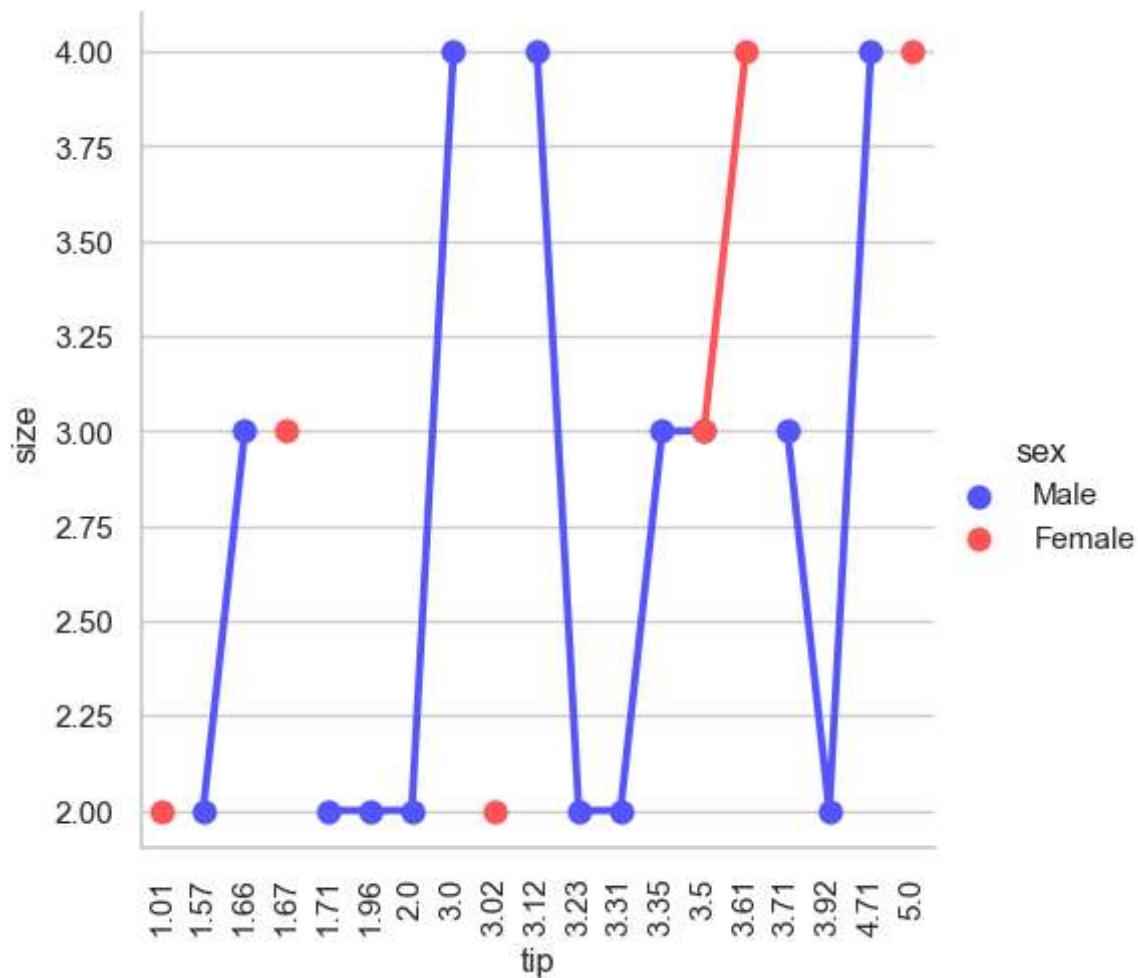
```
In [209]: sns.catplot(x='tip',y='size',data= df,hue="sex",palette="seismic", kind="swarm")
plt.xticks(rotation="vertical")
plt.show()
```



```
In [213]: sns.catplot(x='time',y='total_bill',data= df,hue="sex",palette="seismic", kind="box")
plt.xticks(rotation="vertical")
plt.show()
```



```
In [214]: sns.catplot(x='tip',y='size',data= df,hue="sex",palette="seismic", kind="point"
plt.xticks(rotation="vertical")
plt.show()
```



## Styling Plot in Seaborn

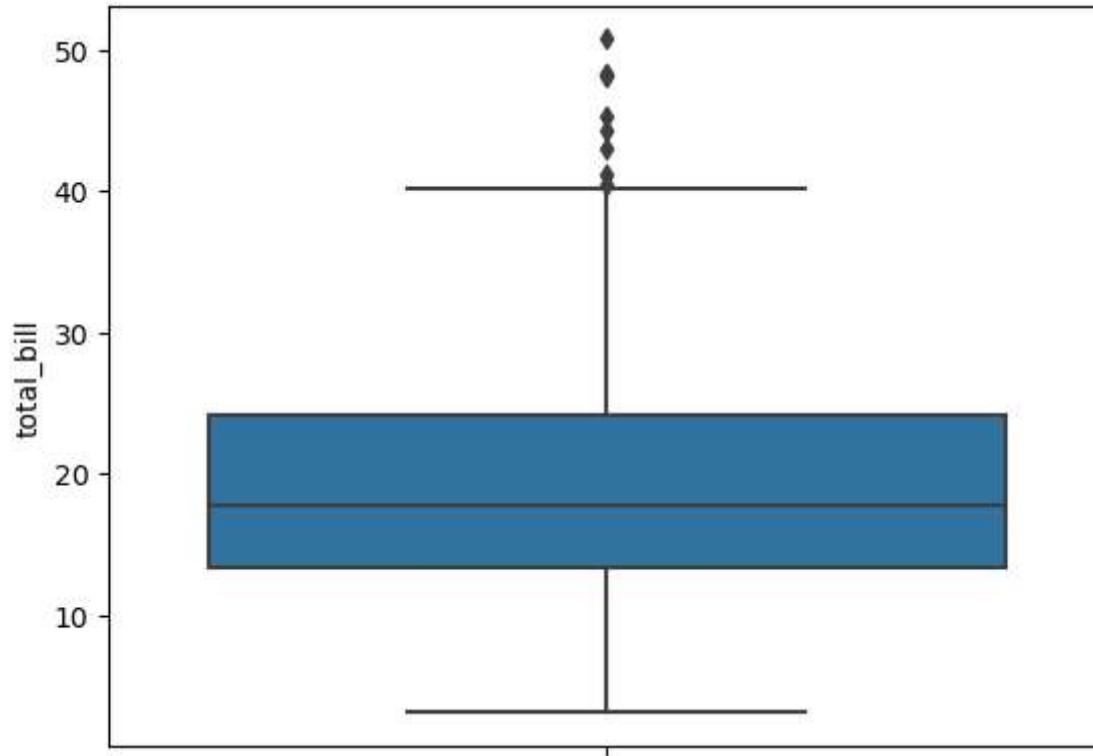
```
In [1]: import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
```

```
In [2]: df = sns.load_dataset("tips")  
df
```

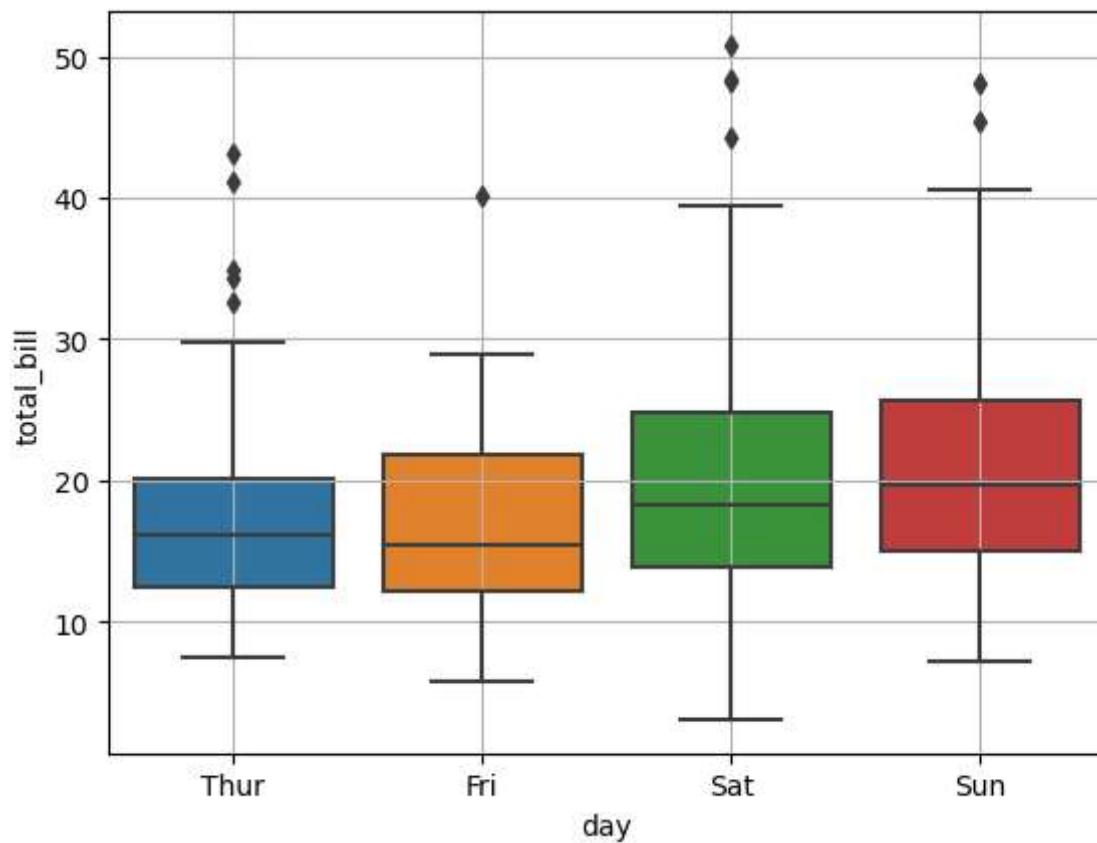
```
Out[2]:   total_bill  tip    sex  smoker  day    time  size  
0      16.99  1.01  Female    No  Sun  Dinner     2  
1      10.34  1.66   Male    No  Sun  Dinner     3  
2      21.01  3.50   Male    No  Sun  Dinner     3  
3      23.68  3.31   Male    No  Sun  Dinner     2  
4      24.59  3.61  Female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

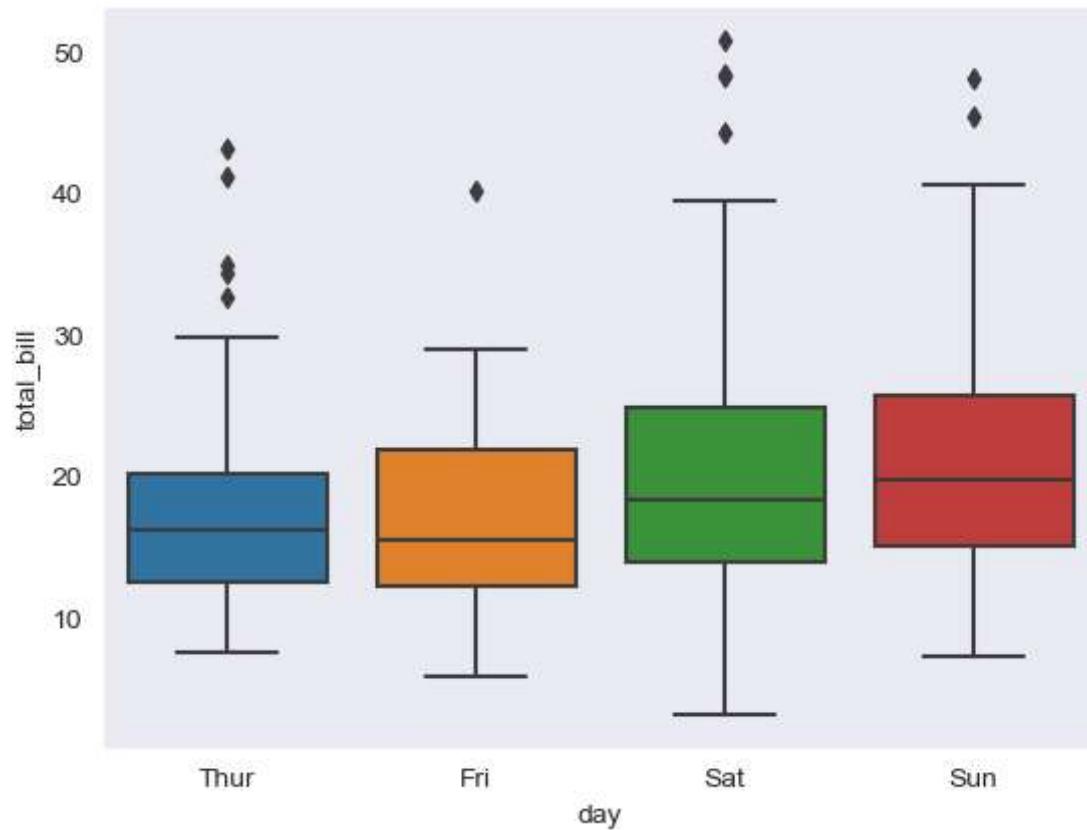
```
In [3]: sns.boxplot(y=df["total_bill"])  
plt.show()
```



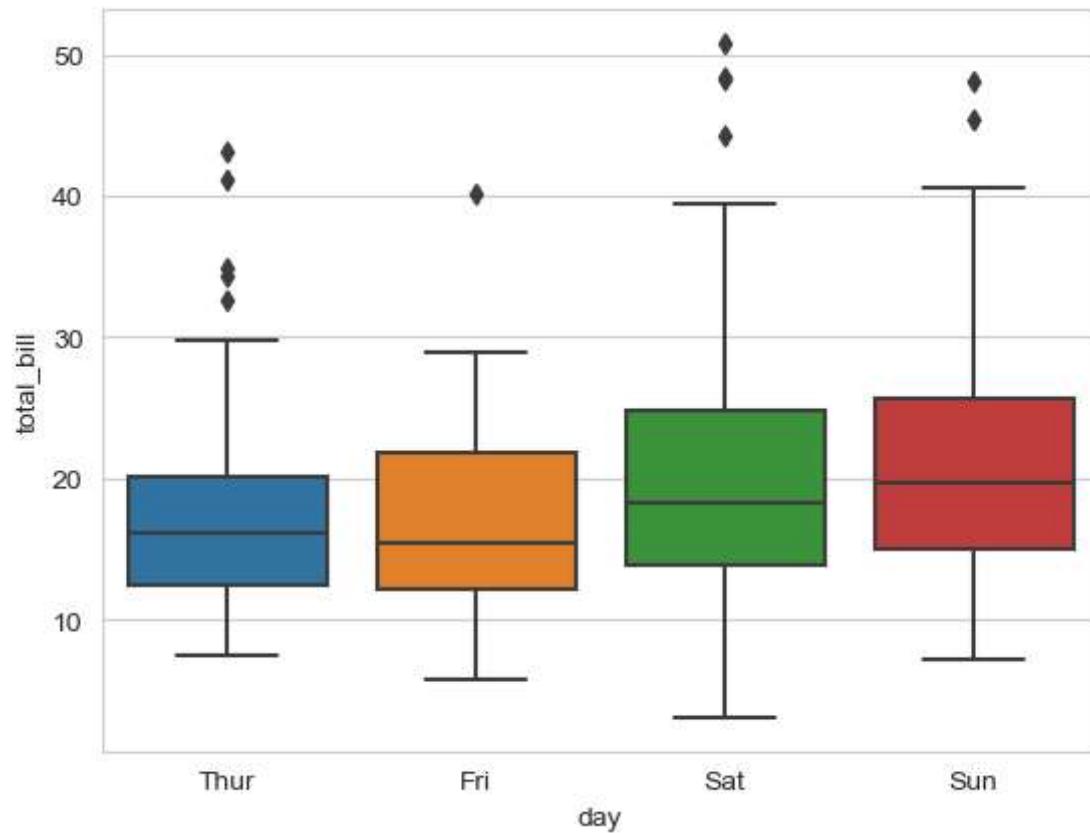
```
In [4]: sns.boxplot(x="day", y="total_bill", data=df)
plt.grid()
plt.show()
```



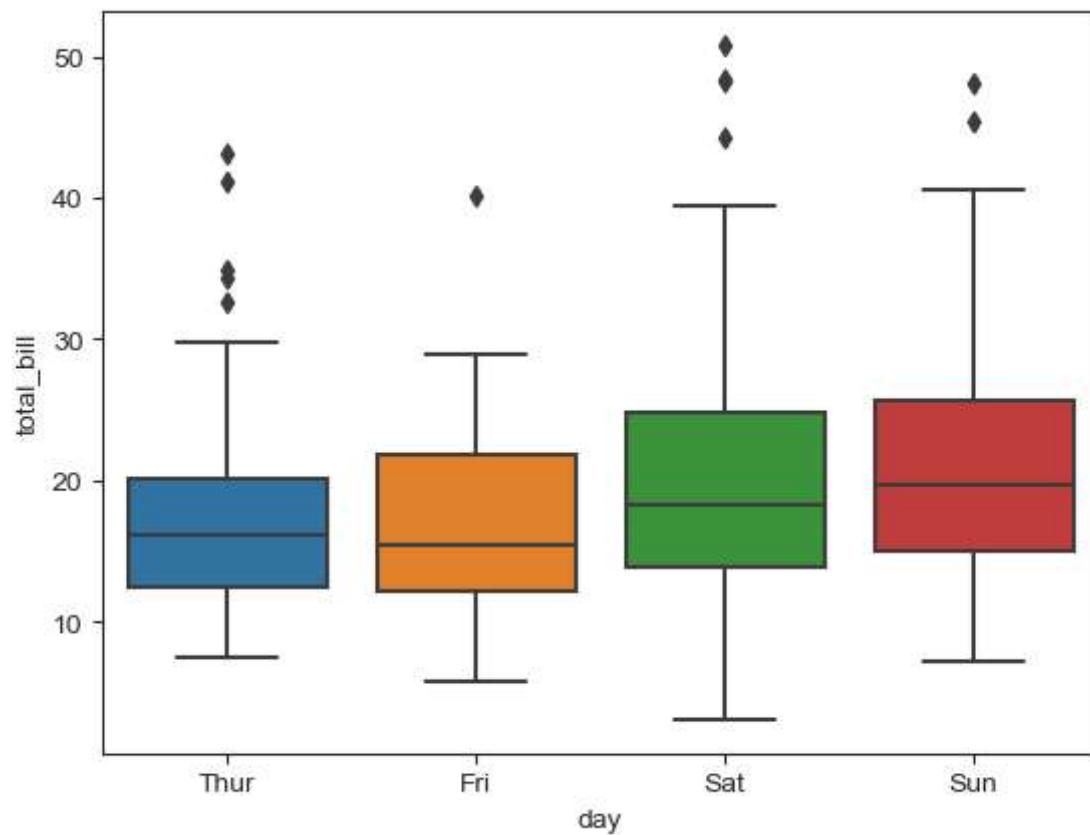
```
In [5]: sns.set_style("dark")
sns.boxplot(x="day", y="total_bill", data=df)
plt.show()
```



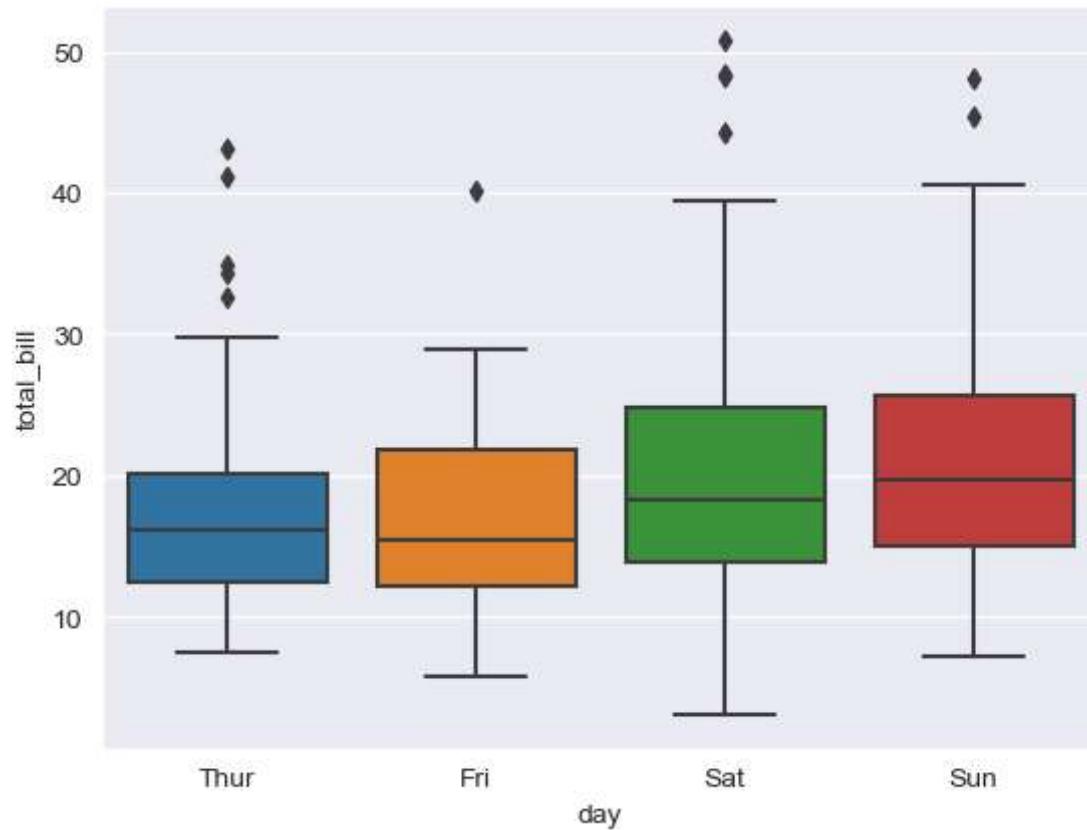
```
In [6]: sns.set_style("whitegrid")
sns.boxplot(x="day", y="total_bill", data=df)
plt.show()
```



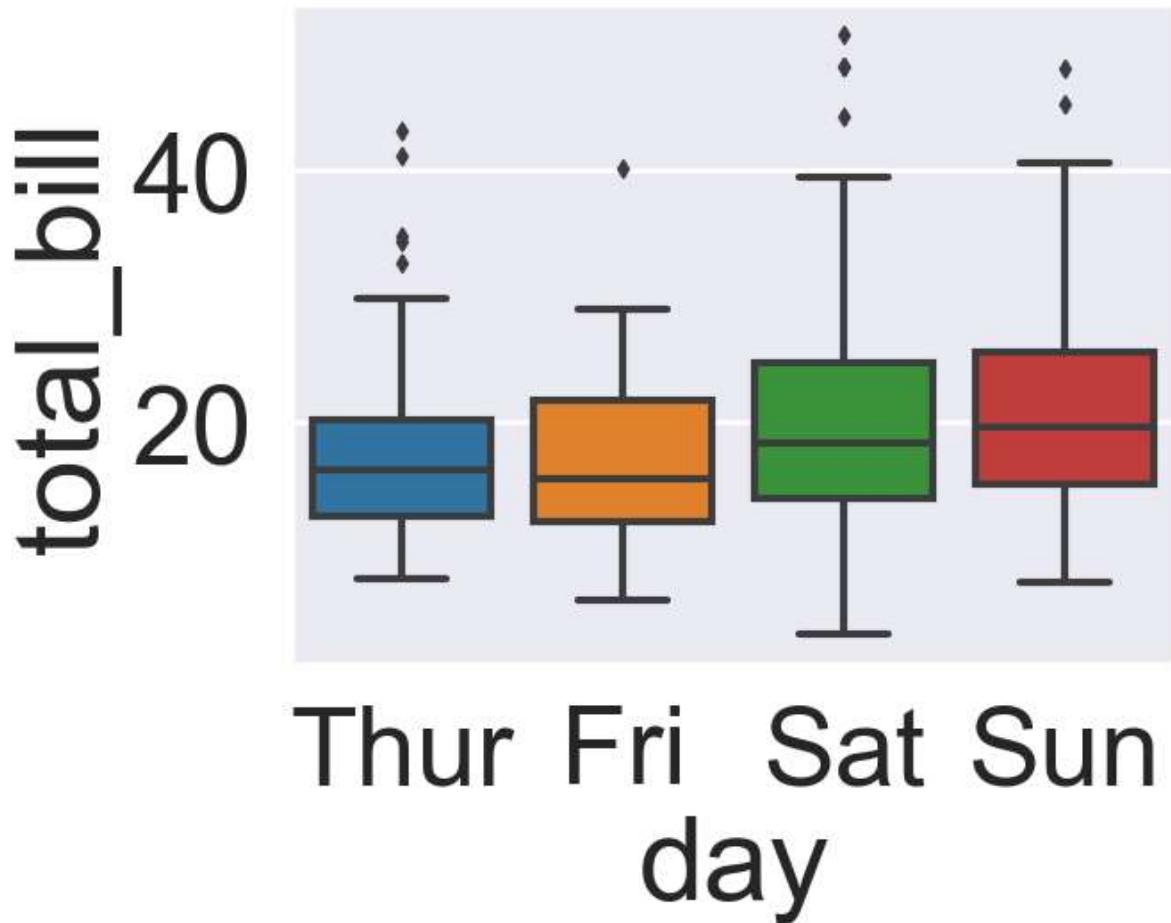
```
In [7]: sns.set_style("ticks")
sns.boxplot(x="day", y="total_bill", data=df)
plt.show()
```



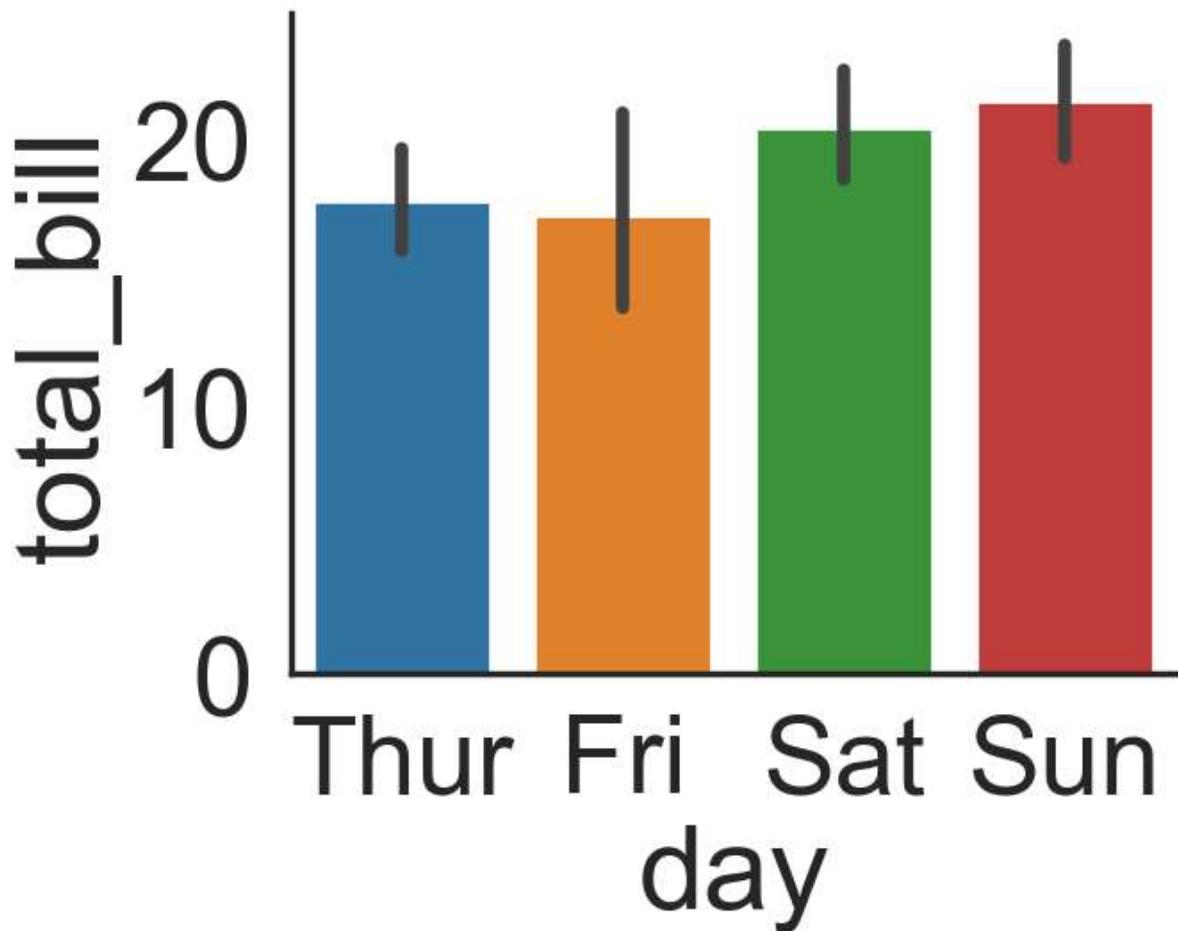
```
In [8]: sns.set_style("darkgrid")
sns.boxplot(x="day", y="total_bill", data=df)
plt.show()
```



```
In [9]: sns.set_context("poster",font_scale=2)
sns.boxplot(x="day", y="total_bill", data=df)
plt.show()
```



```
In [11]: sns.set_style("white")
sns.barplot(x="day", y="total_bill", data=df)
sns.despine()
plt.show()
```



## Facet Grid Plot

```
In [12]: import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
```

In [13]: df

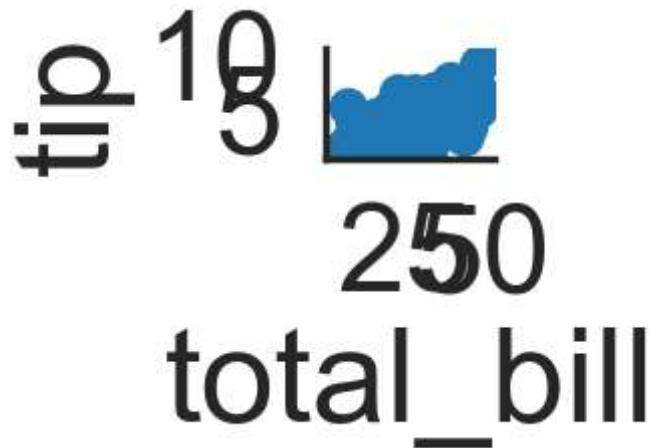
	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4
...	...	...	...	...	...	...	...
239	29.03	5.92	Male	No	Sat	Dinner	3
240	27.18	2.00	Female	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 × 7 columns

```
In [15]: ab = sns.FacetGrid(df)
ab.map(plt.scatter,"total_bill","tip")
plt.show()
```

C:\ProgramData\Anaconda3\lib\site-packages\seaborn\axisgrid.py:88: UserWarning: Tight layout not applied. The left and right margins cannot be made large enough to accommodate all axes decorations.

```
    self._figure.tight_layout(*args, **kwargs)
```



In [ ]: