Seaborn library itself having datasets

In [1]:

```
# listing all datasets from the seaborn library
import seaborn as sns
sns.get_dataset_names()
```

Out[1]:

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

Loading Data set from the seaborn library

In [2]:

```
df = sns.load_dataset('tips')
df.head()
```

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

In [3]:

```
import pandas as pd
tips=pd.read_csv('tips.csv')
tips.head()
```

Out[3]:

	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

Graphs

```
-> Uni-Variant

    -> Single column

-> Bi-Variant

    -> two columns

-> Multi-Variant

    -> more than 2 columns
```

Function interface

Relational plots

relplot

Figure-level interface for drawing relational plots onto a FacetGrid.

scatterplot

Draw a scatter plot with possibility of several semantic groupings.

lineplot

Draw a line plot with possibility of several semantic groupings.

Distribution plots

displot

Figure-level interface for drawing distribution plots onto a FacetGrid.

histplot

Plot univariate or bivariate histograms to show distributions of datasets.

kdeplot

Plot univariate or bivariate distributions using kernel density estimation.

ecdfplot

Plot empirical cumulative distribution functions.

rugplot

Plot marginal distributions by drawing ticks along the x and y axes.

distplot

DEPRECATED

Categorical plots

catplot

Figure-level interface for drawing categorical plots onto a FacetGrid.

stripplot

Draw a categorical scatterplot using jitter to reduce overplotting.

swarmplot

Draw a categorical scatterplot with points adjusted to be non-overlapping.

boxplot

Draw a box plot to show distributions with respect to categories.

violinplot

Draw a combination of boxplot and kernel density estimate.

boxenplot

Draw an enhanced box plot for larger datasets.

pointplot

Show point estimates and errors using dot marks.

barplot

Show point estimates and errors as rectangular bars.

countplot

Show the counts of observations in each categorical bin using bars.

Regression plots

lmplot

Plot data and regression model fits across a FacetGrid.

regplot

Plot data and a linear regression model fit.

residplot

Plot the residuals of a linear regression.

Matrix plots

heatmap

Plot rectangular data as a color-encoded matrix.

clustermap

Plot a matrix dataset as a hierarchically-clustered heatmap.

Multi-plot grids

Facet grids

FacetGrid

Multi-plot grid for plotting conditional relationships.

Pair grids

pairplot

Plot pairwise relationships in a dataset.

PairGrid

Subplot grid for plotting pairwise relationships in a dataset.

Joint grids

jointplot

Draw a plot of two variables with bivariate and univariate graphs.

JointGrid

Grid for drawing a bivariate plot with marginal univariate plots

rel plot

In [4]:

```
tips.columns
```

Out[4]:

```
Index(['total_bill', 'tip', 'sex', 'smoker', 'day', 'time', 'size'], dtype
='object')
```

In [5]:

```
tips.head()
```

Out[5]:

	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

In [6]:

```
tips.dtypes
```

Out[6]:

```
total_bill    float64
tip           float64
sex            object
smoker         object
day            object
time           object
size            int64
dtype: object
```

Data information of the each and every column

Column Name-1: total_bill

Stats Data information

```
    => Numerical with Continuous Values
```

Programming Data Type information

=> float

Example presented in data Frame

=> 16.99

tip

=> Numerical with Continuous Values => 3.31 => float

sex

=> Characters with Categorical Values => Male => object

smoker

=> Characters with Categorical Values => Yes => object

day

=> Characters with Categorical Values => Sun => object

time

=> Characters with Categorical Values => Dinner => object

size

=> numbers with categorical Values => 4 => int

In [7]:

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

In [8]:

```
df=sns.load_dataset('tips')
df.head()
```

Out[8]:

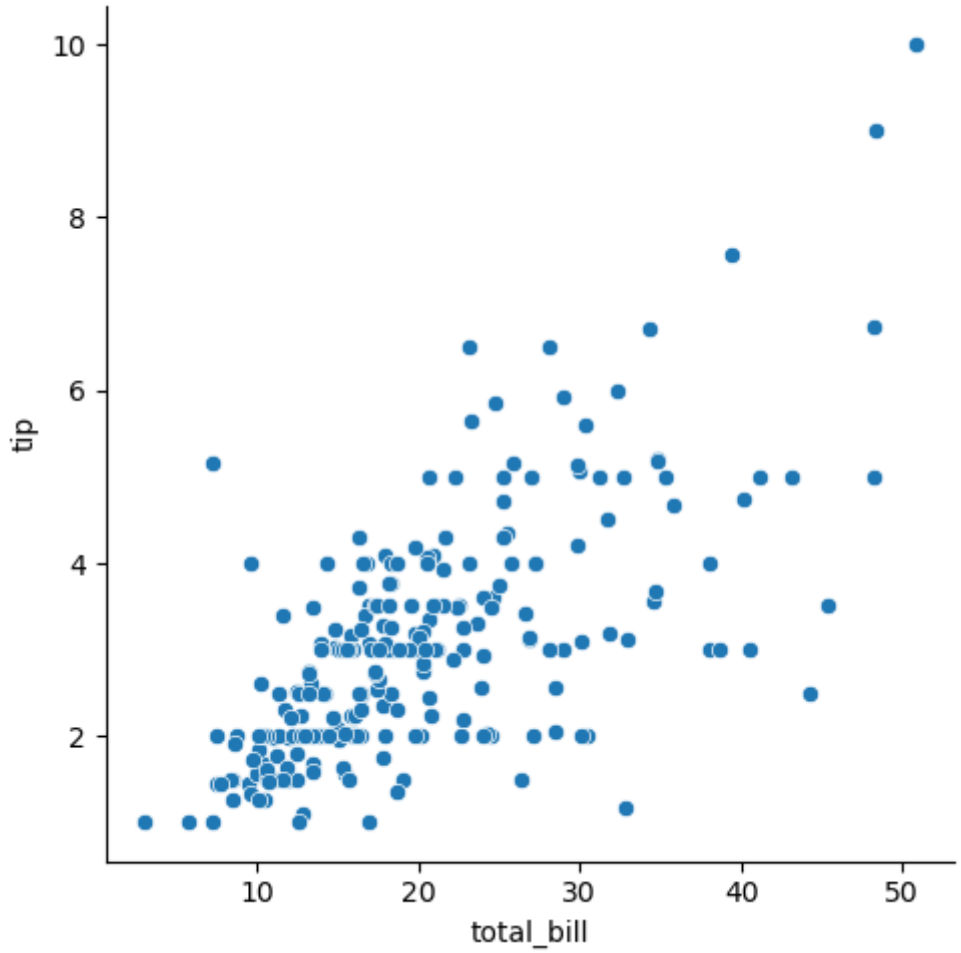
	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4

In [9]:

```
sns.relplot(data=df, x ='total_bill',y='tip')
```

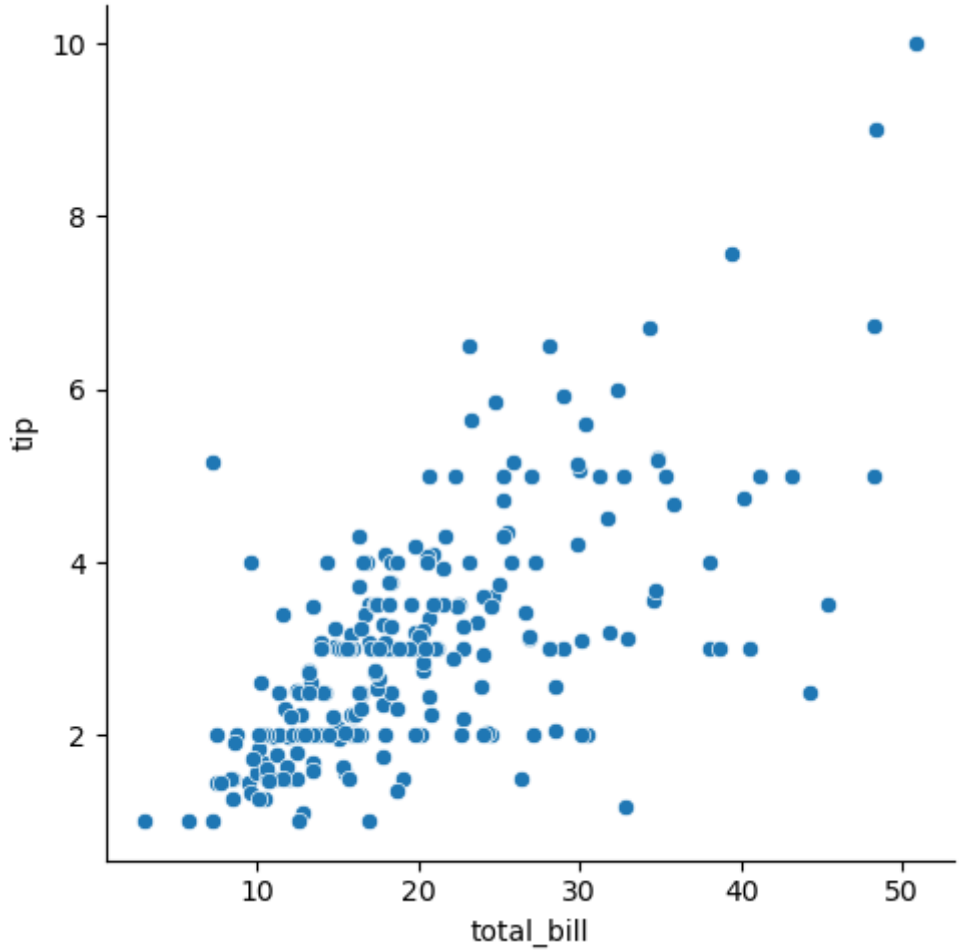
Out[9]:

```
<seaborn.axisgrid.FacetGrid at 0x1df7babbcd0>
```



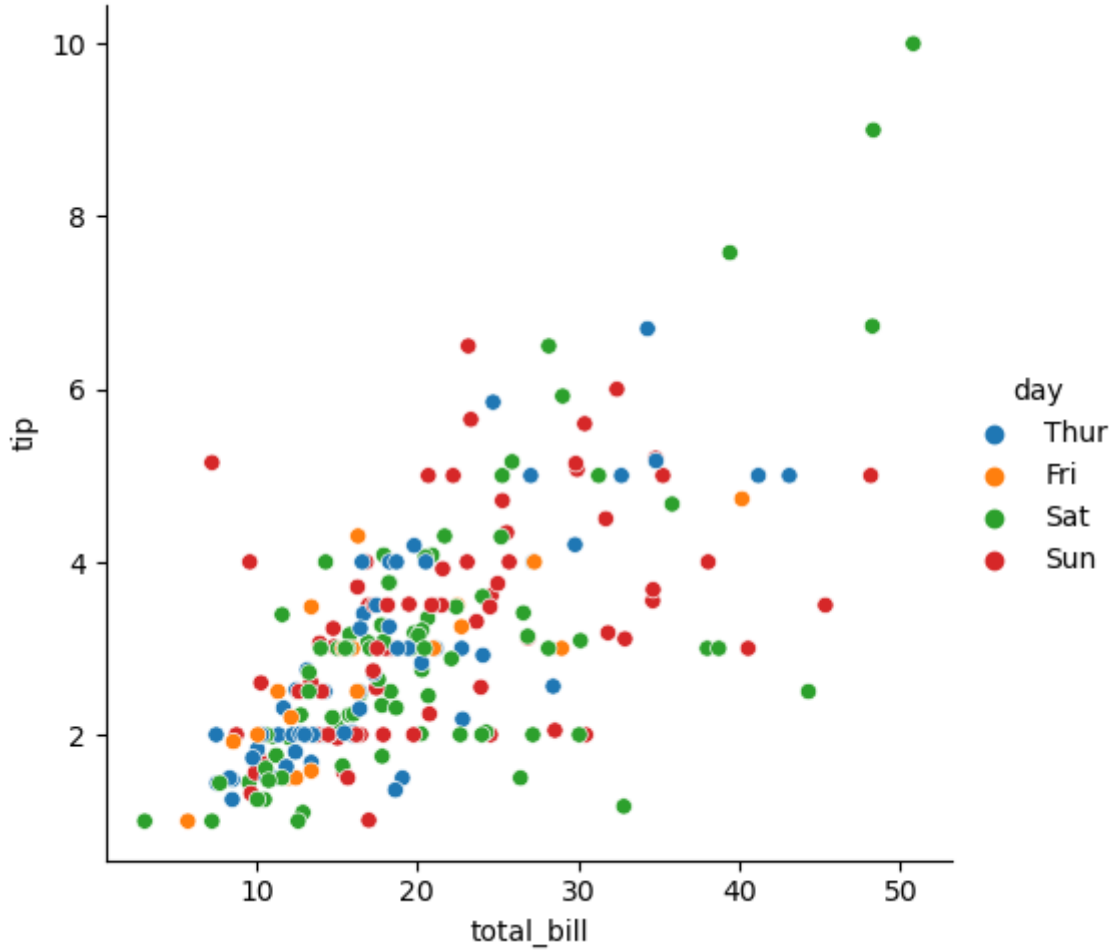
In [10]:

```
sns.relplot(data=df, x="total_bill", y="tip")
plt.show()
```



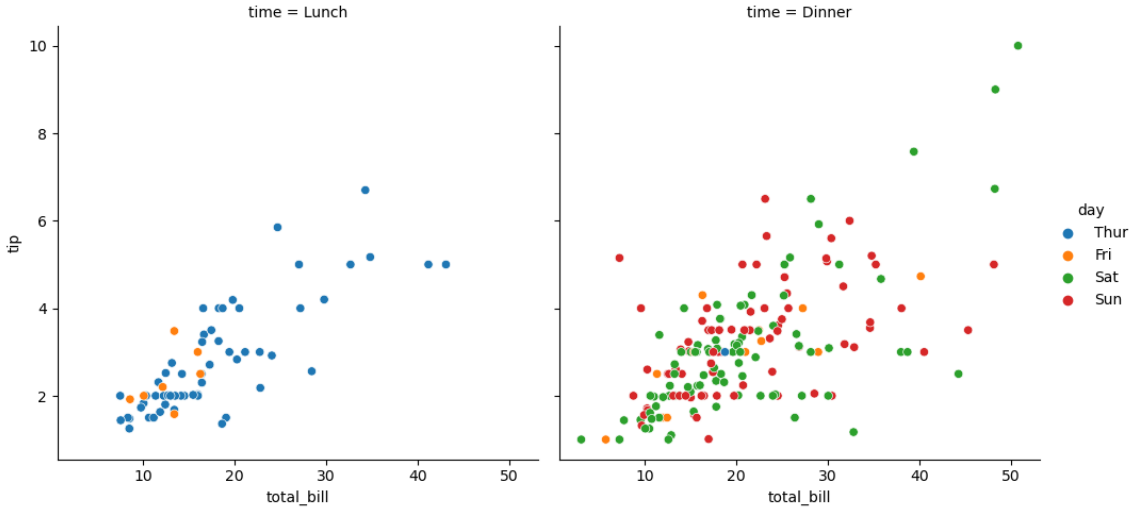
In [11]:

```
sns.relplot(data=df, x="total_bill", y="tip", hue="day")
plt.show()
```



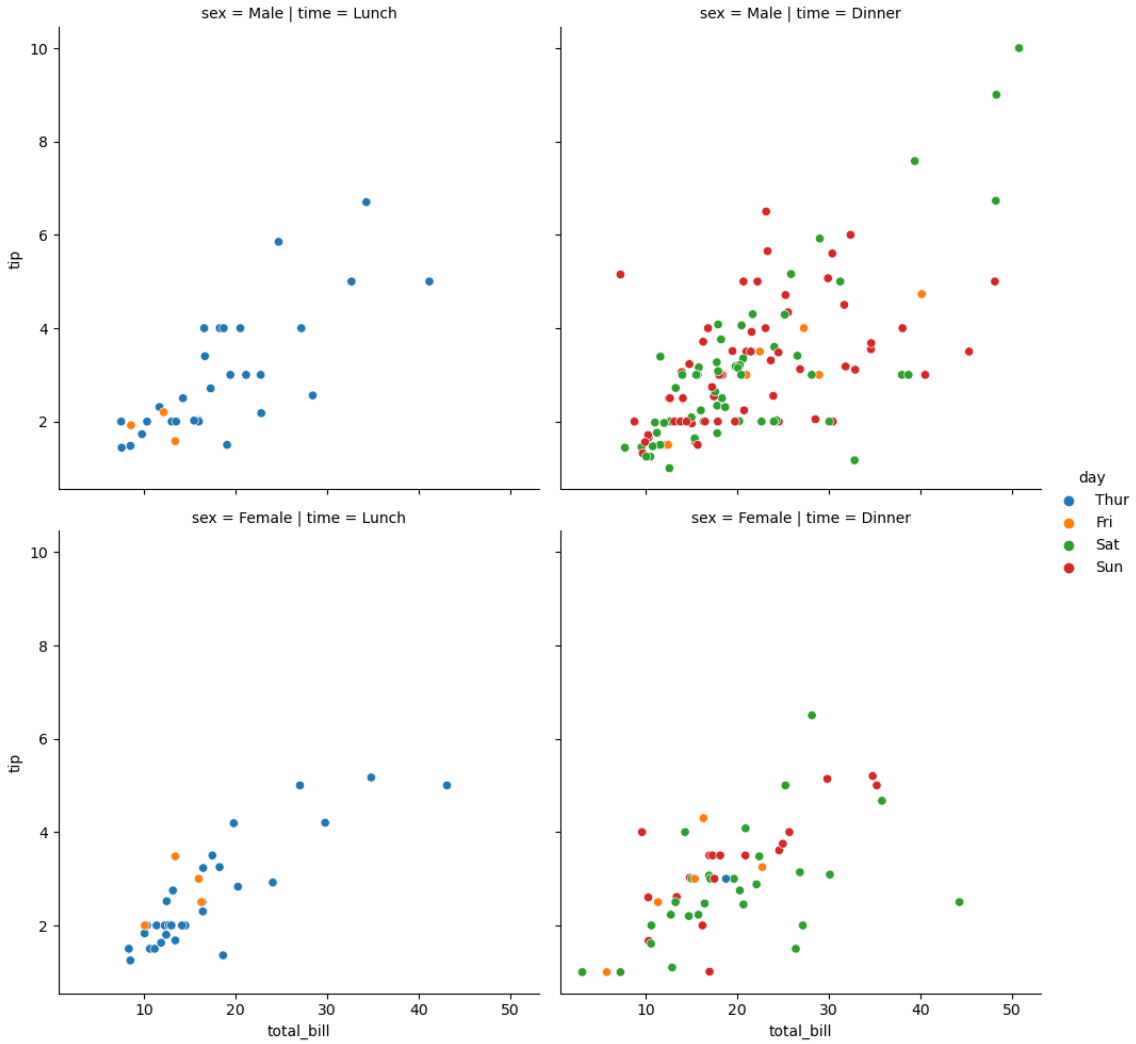
In [12]:

```
sns.relplot(data=df, x="total_bill", y="tip", hue="day", col="time")
plt.show()
```



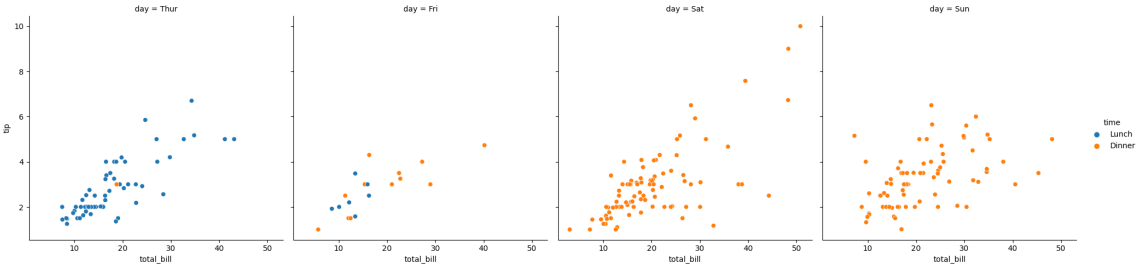
In [13]:

```
sns.relplot(data=df, x="total_bill", y="tip", hue="day", col="time", row="sex")
plt.show()
```



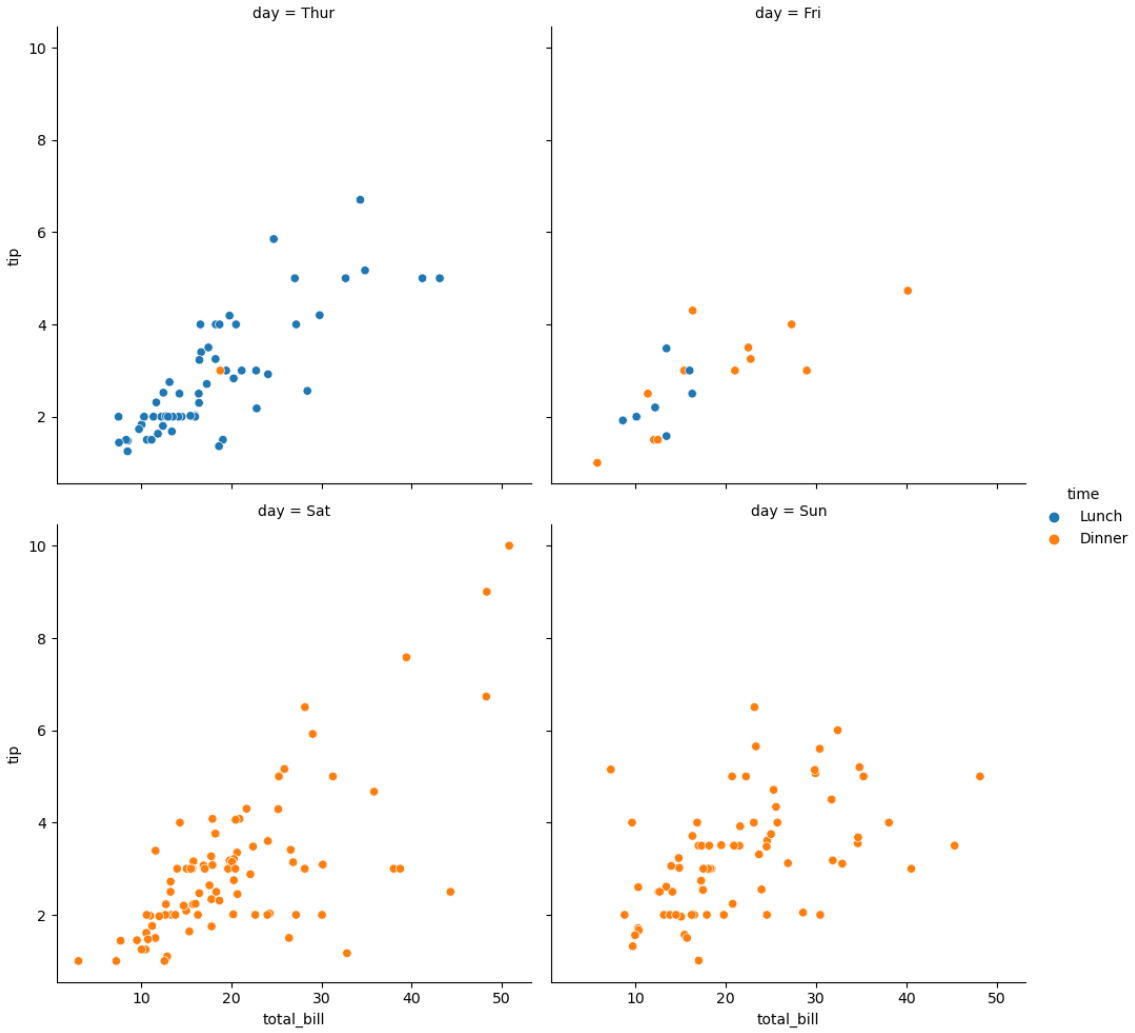
In [14]:

```
sns.relplot(data=df, x="total_bill", y="tip", hue="time", col="day")
plt.show()
```



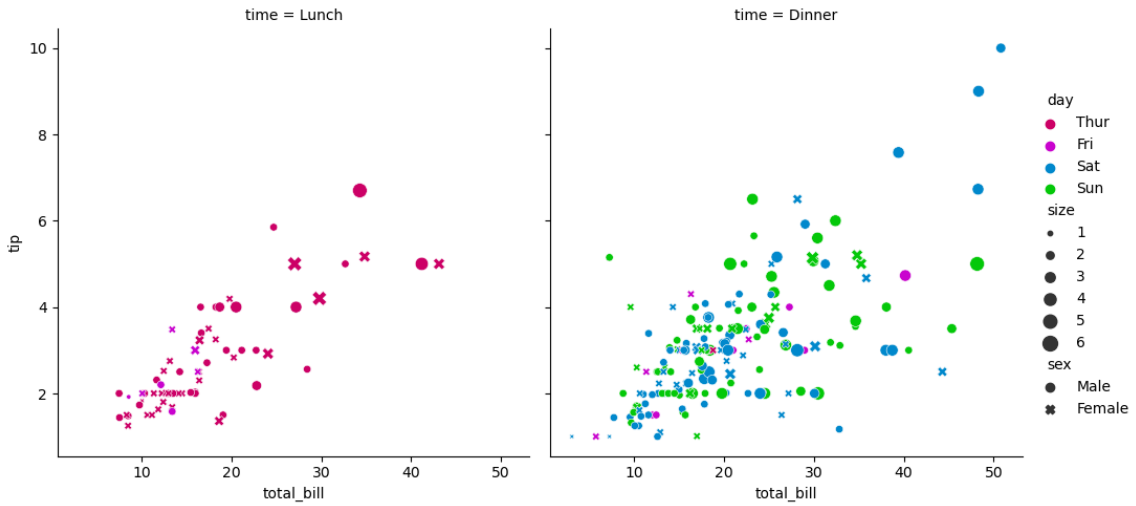
In [15]:

```
sns.relplot(data=df, x="total_bill", y="tip", hue="time", col="day", col_wrap=2)
plt.show()
```



In [16]:

```
sns.relplot(
    data=df, x="total_bill", y="tip", col="time",
    hue="day", size="size", style="sex",
    palette=['#CC0066', '#C903CE', '#0088CC', '#04C809'], sizes=(10, 100)
)
plt.show()
```



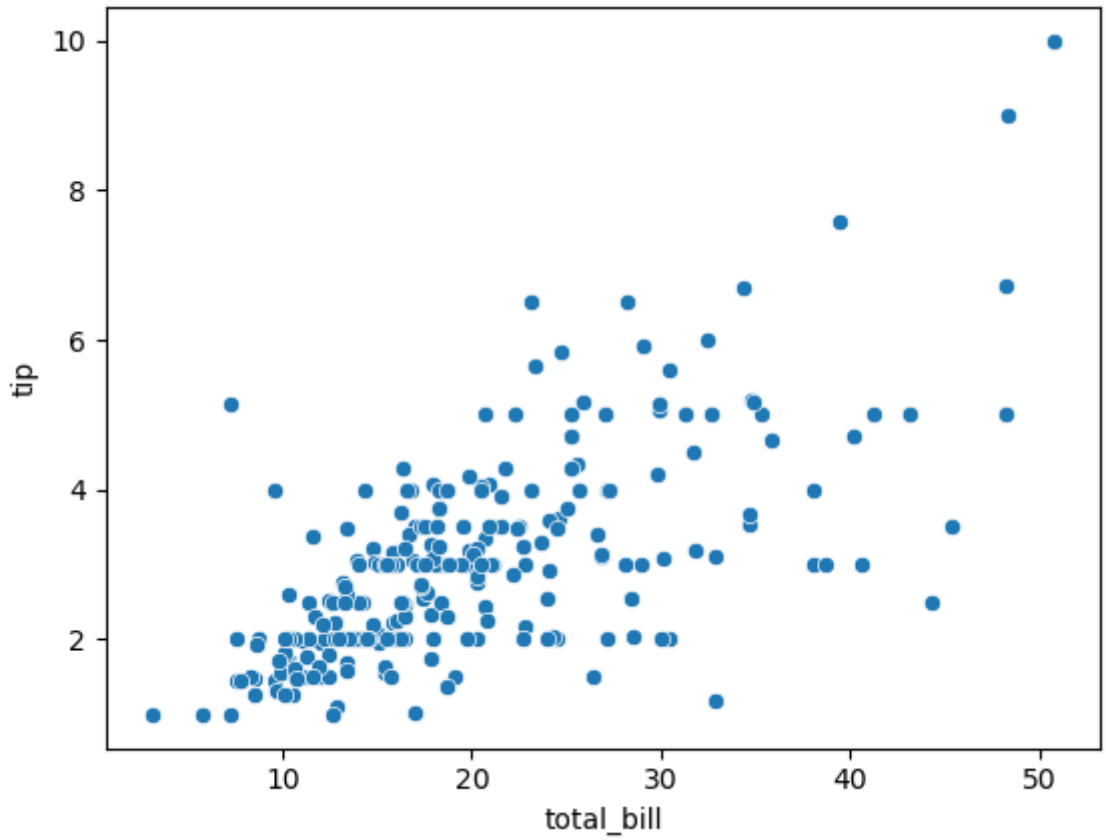
In [17]:

```
tips = sns.load_dataset("tips")
```

Scatter plot

In [18]:

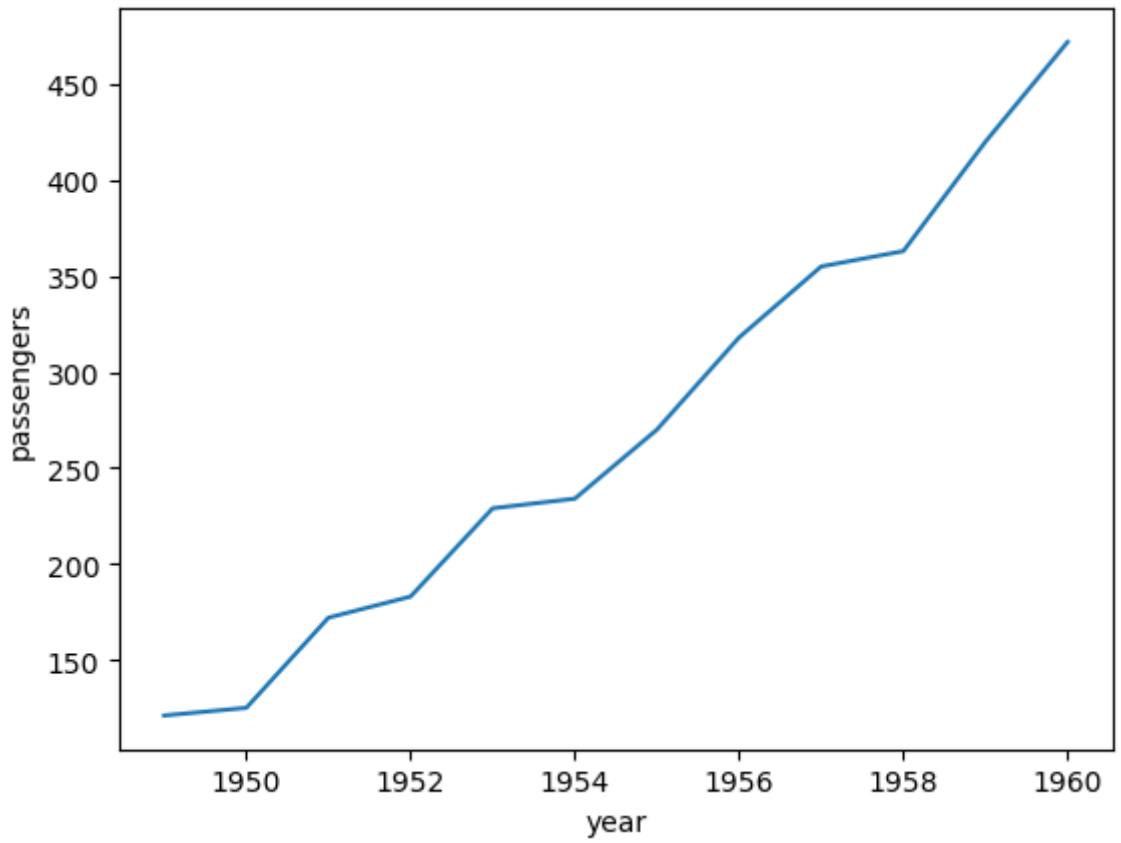
```
sns.scatterplot(data=tips, x="total_bill", y="tip")
plt.show()
```



lineplot

In [19]:

```
flights = sns.load_dataset("flights")
may_flights = flights.query("month == 'May'")
sns.lineplot(data=may_flights, x="year", y="passengers")
plt.show()
```



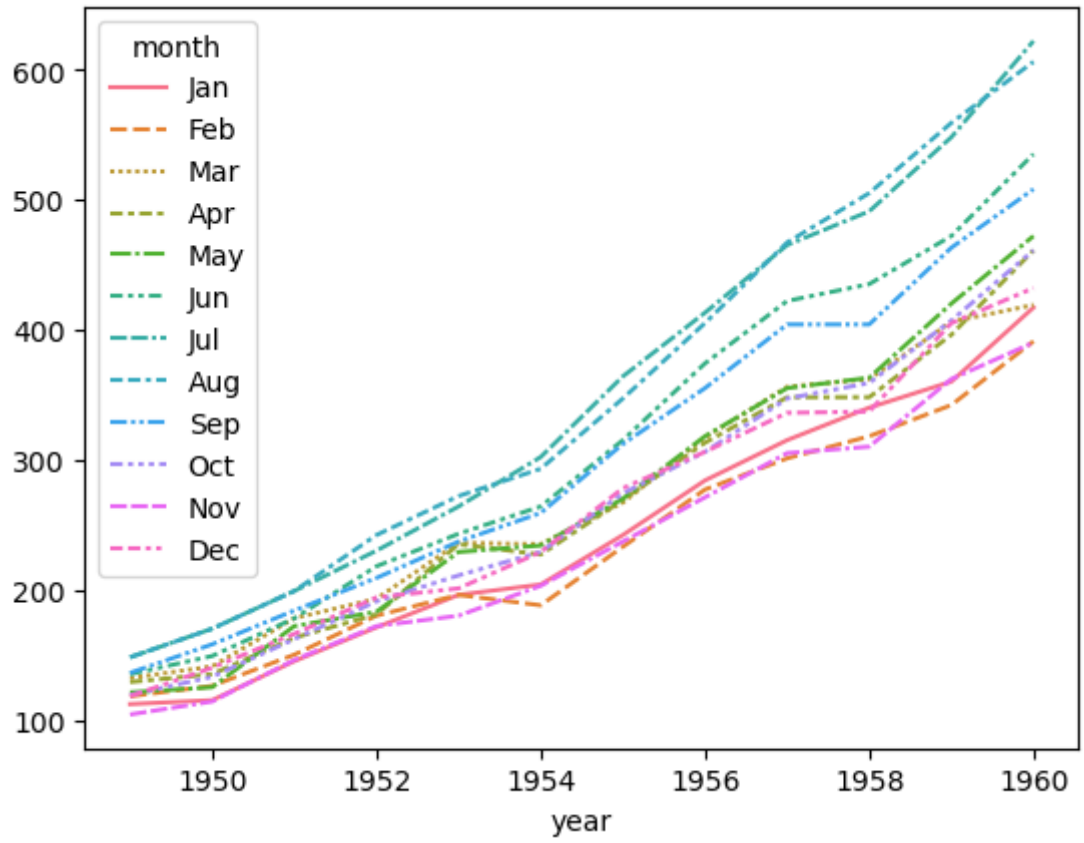
In [20]:

```
flights_wide = flights.pivot("year", "month", "passengers")
sns.lineplot(data=flights_wide)
```

```
C:\Users\divesh\AppData\Local\Temp\ipykernel_25108\4151339522.py:1: Future
Warning: In a future version of pandas all arguments of DataFrame.pivot wi
ll be keyword-only.
  flights_wide = flights.pivot("year", "month", "passengers")
```

Out[20]:

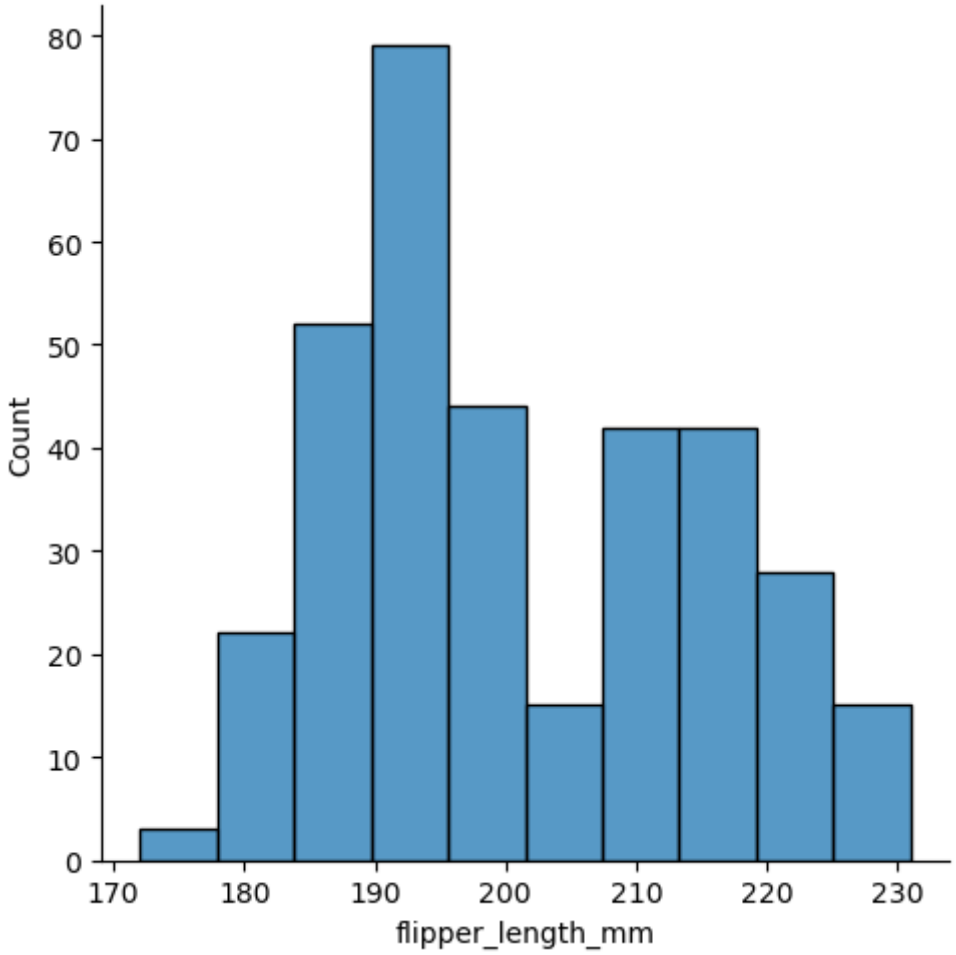
```
<Axes: xlabel='year'>
```



displot

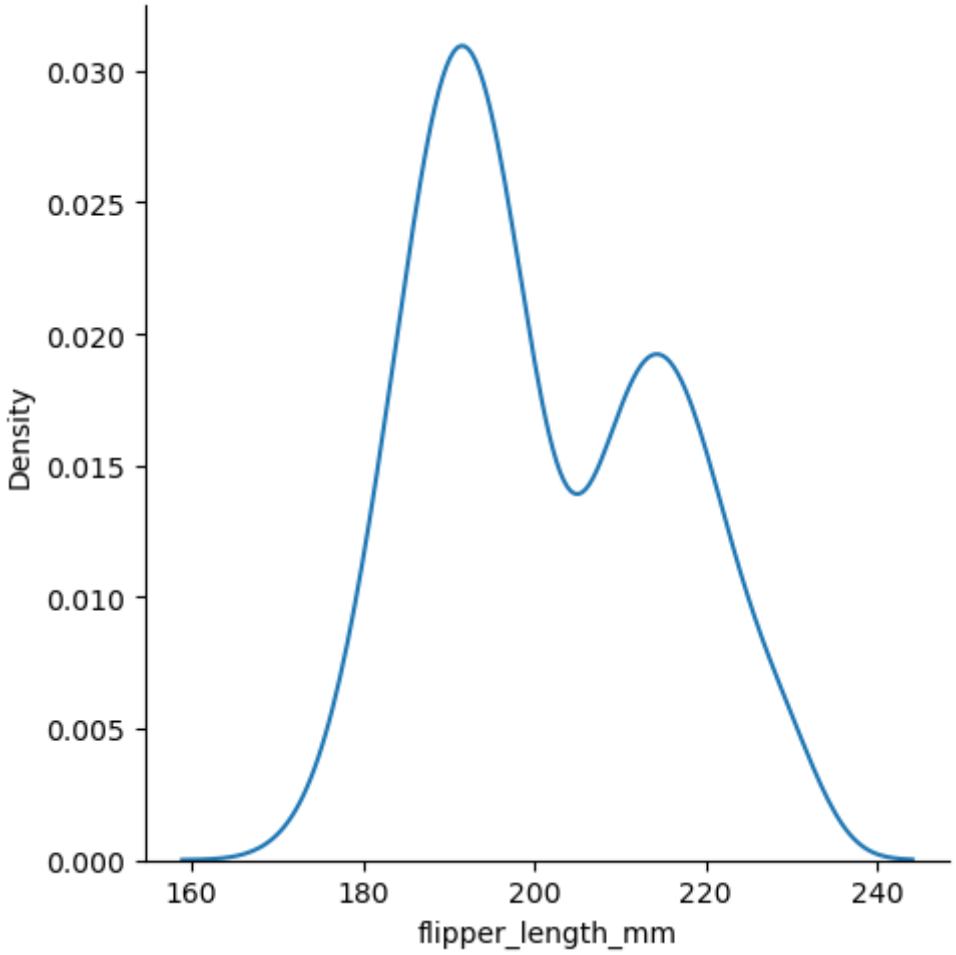
In [21]:

```
penguins = sns.load_dataset("penguins")
sns.displot(data=penguins, x="flipper_length_mm")
plt.show()
```



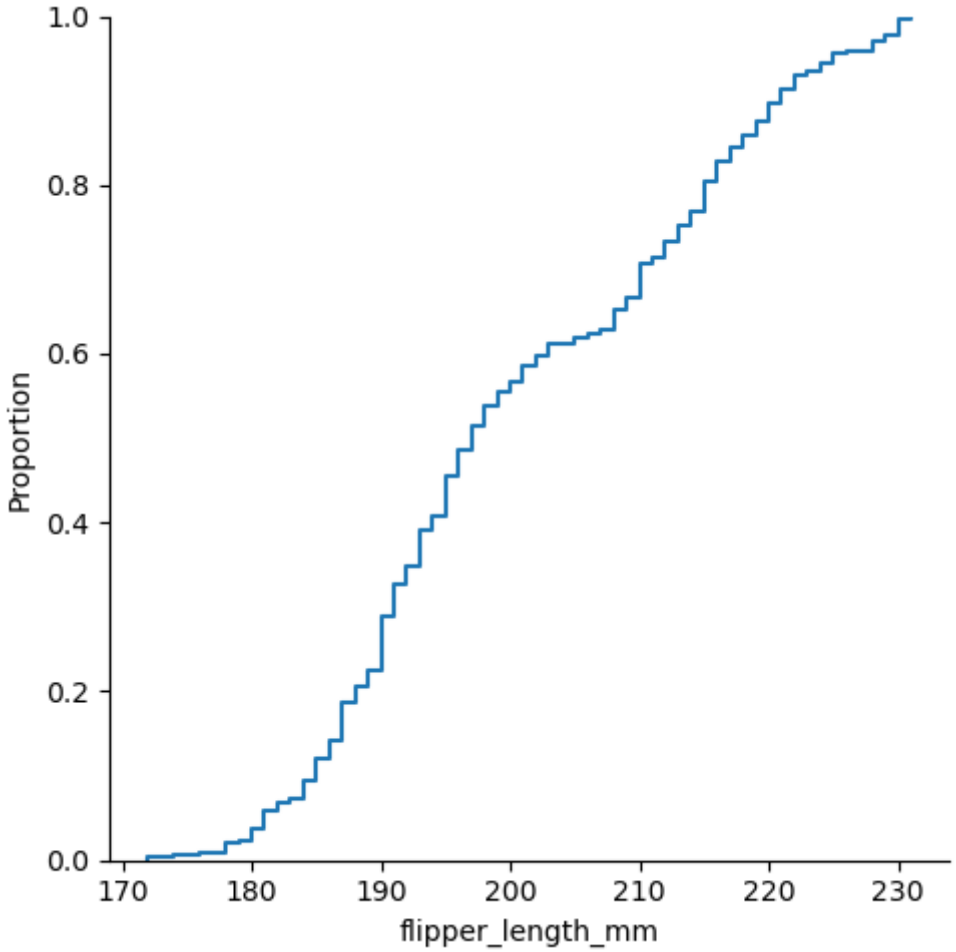
In [22]:

```
sns.displot(data=penguins, x="flipper_length_mm", kind="kde")
plt.show()
```



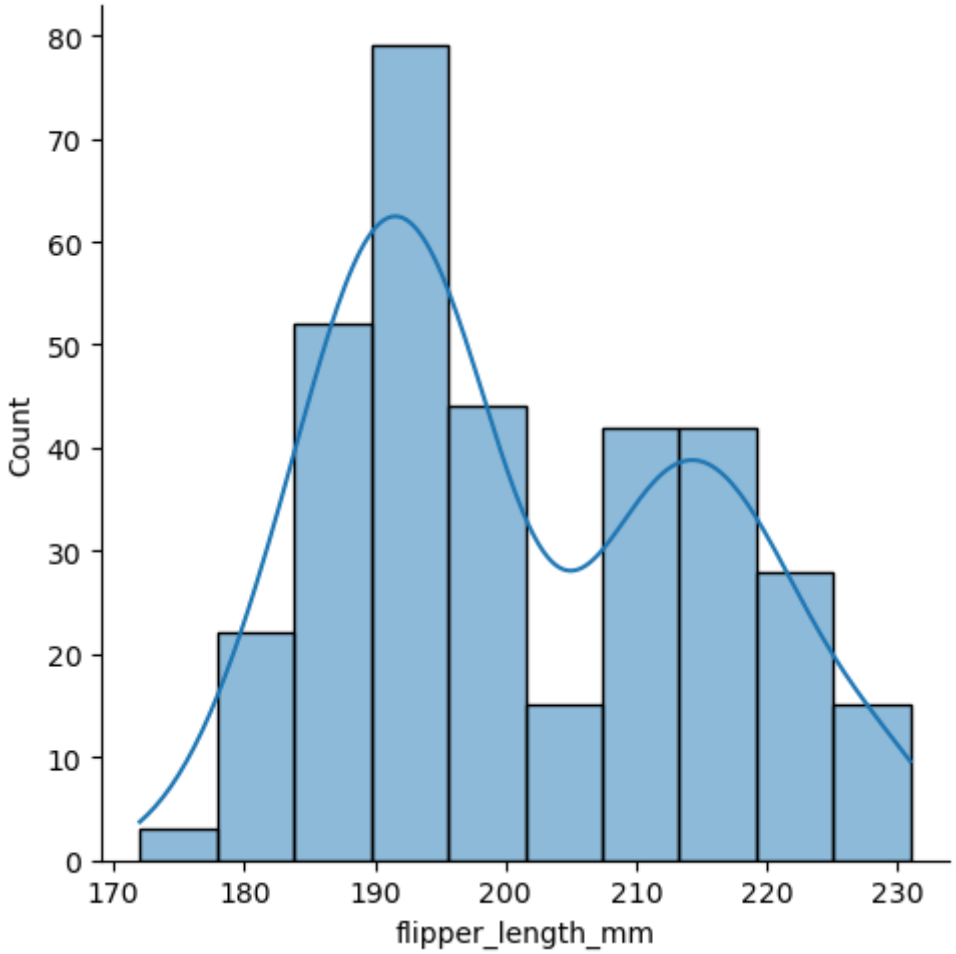
In [23]:

```
sns.displot(data=penguins, x="flipper_length_mm", kind="ecdf")
plt.show()
```



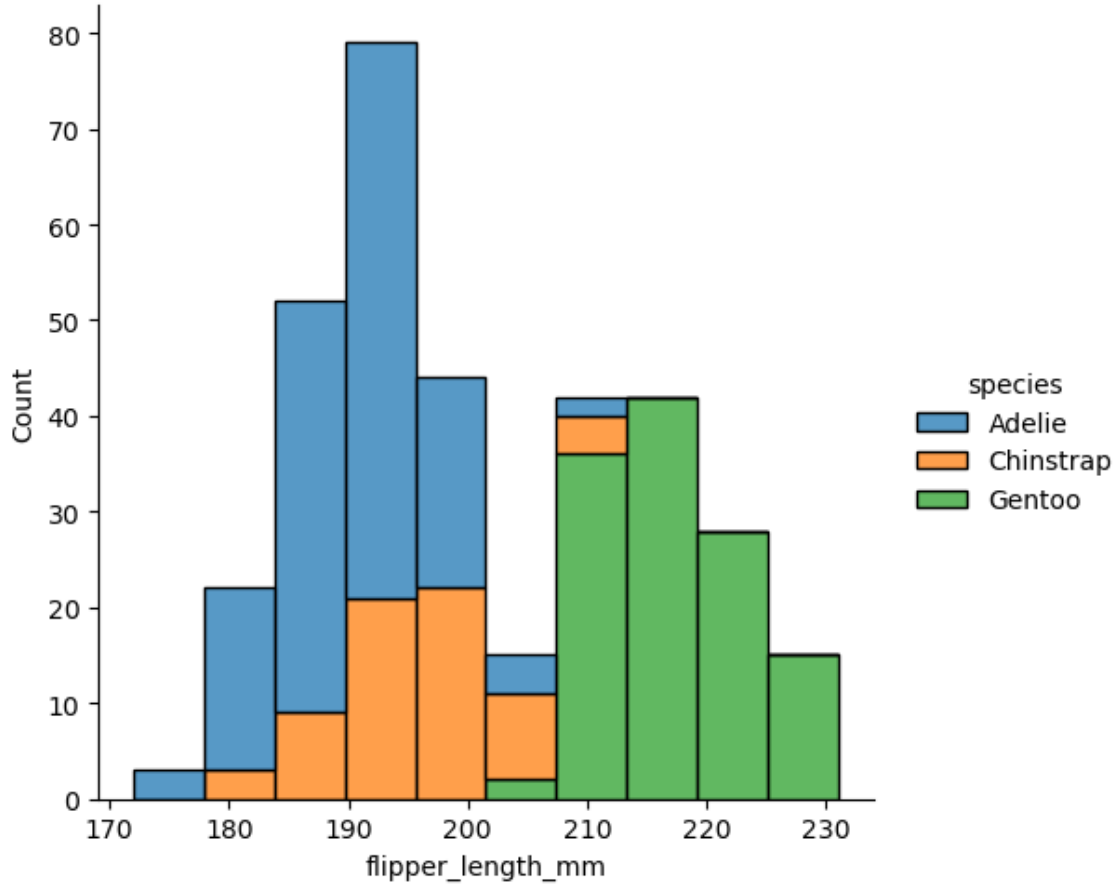
In [24]:

```
sns.displot(data=penguins, x="flipper_length_mm", kde=True)
plt.show()
```



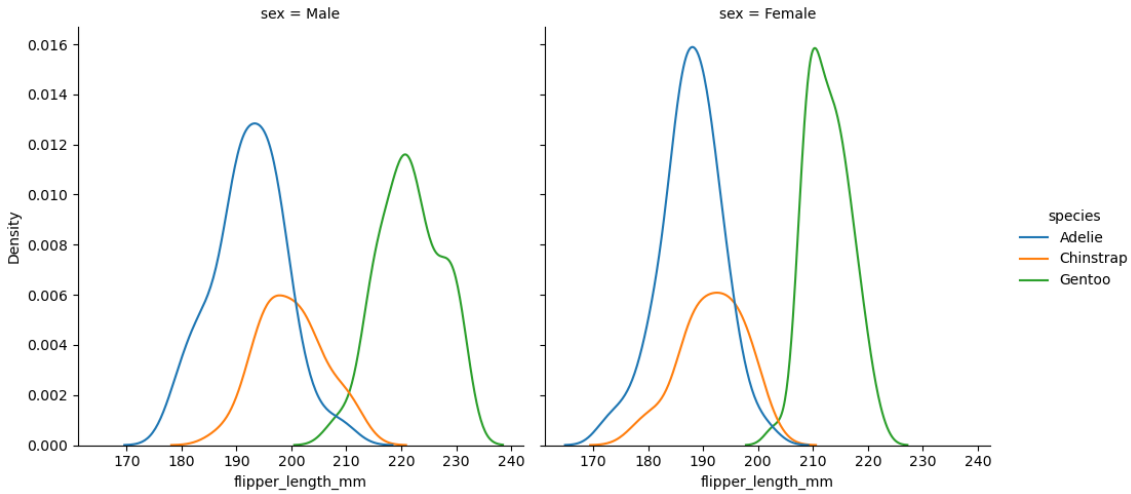
In [25]:

```
sns.displot(data=penguins, x="flipper_length_mm", hue="species", multiple="stack")
plt.show()
```



In [26]:

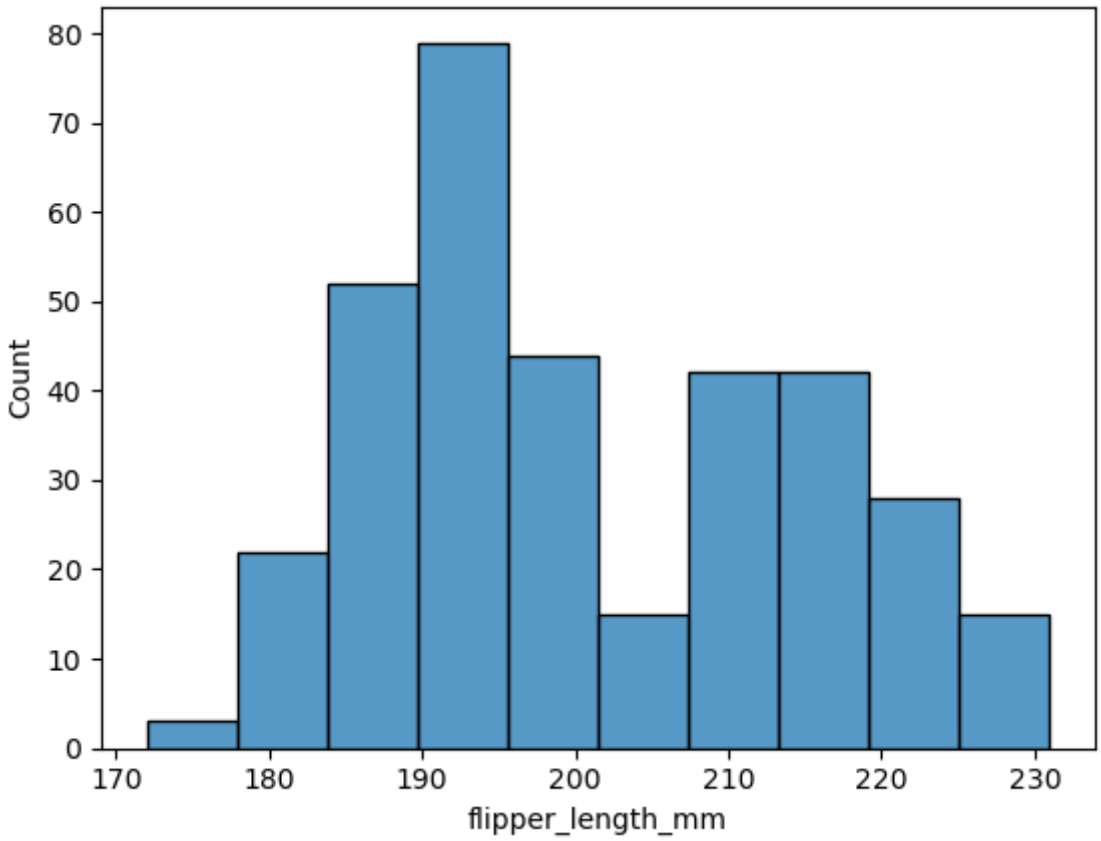
```
sns.displot(data=penguins, x="flipper_length_mm", hue="species", col="sex", kind="kde")
plt.show()
```



histplot

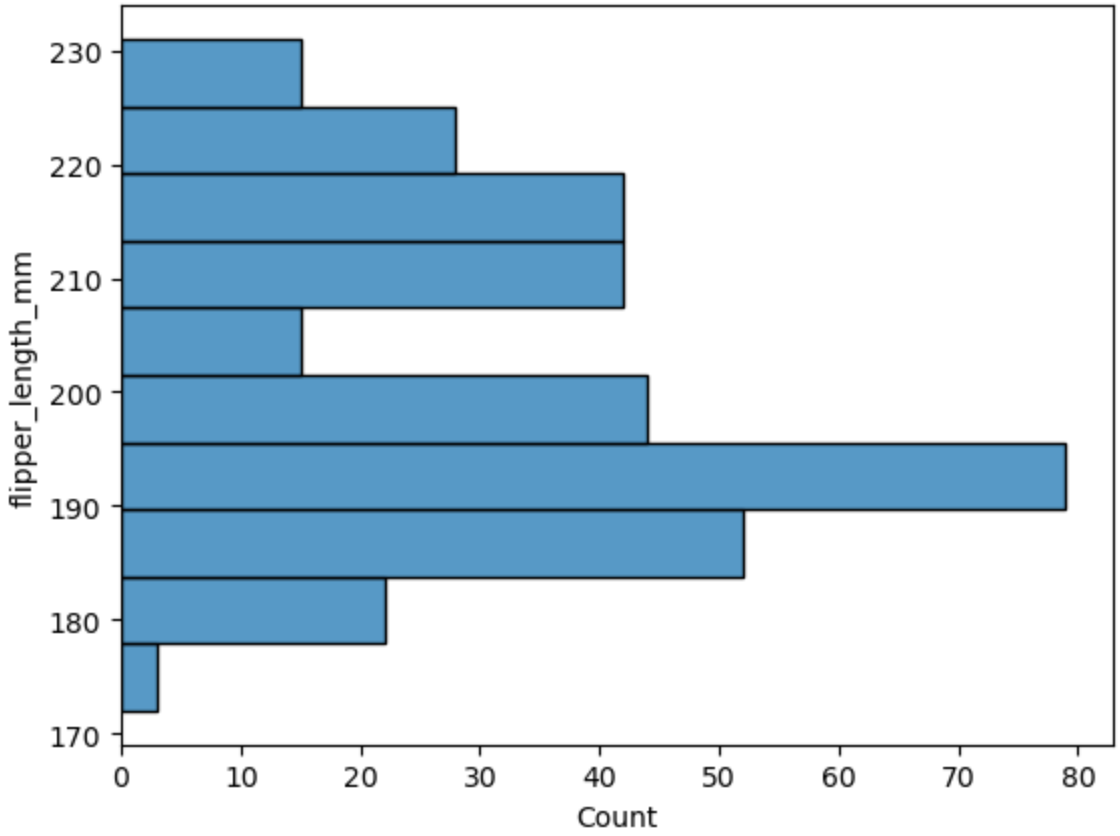
In [27]:

```
sns.histplot(data=penguins, x="flipper_length_mm")
plt.show()
```



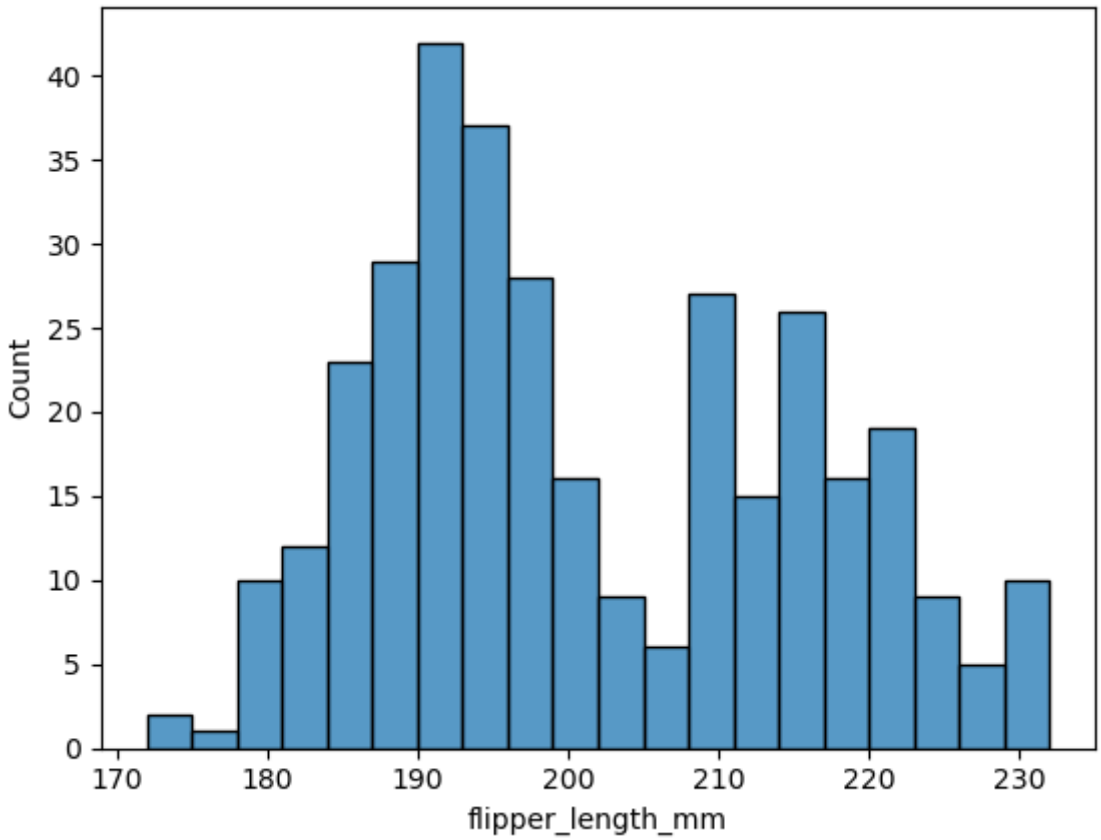
In [28]:

```
sns.histplot(data=penguins, y="flipper_length_mm")
plt.show()
```



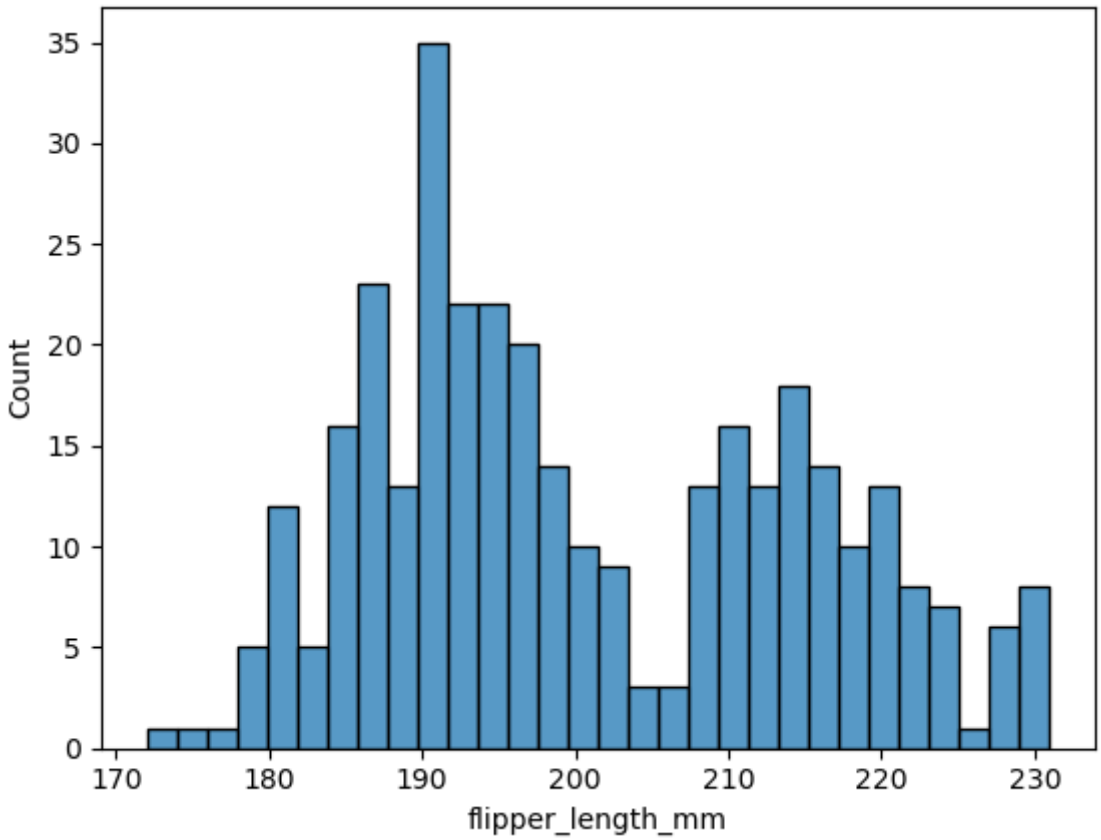
In [29]:

```
sns.histplot(data=penguins, x="flipper_length_mm", binwidth=3)
plt.show()
```



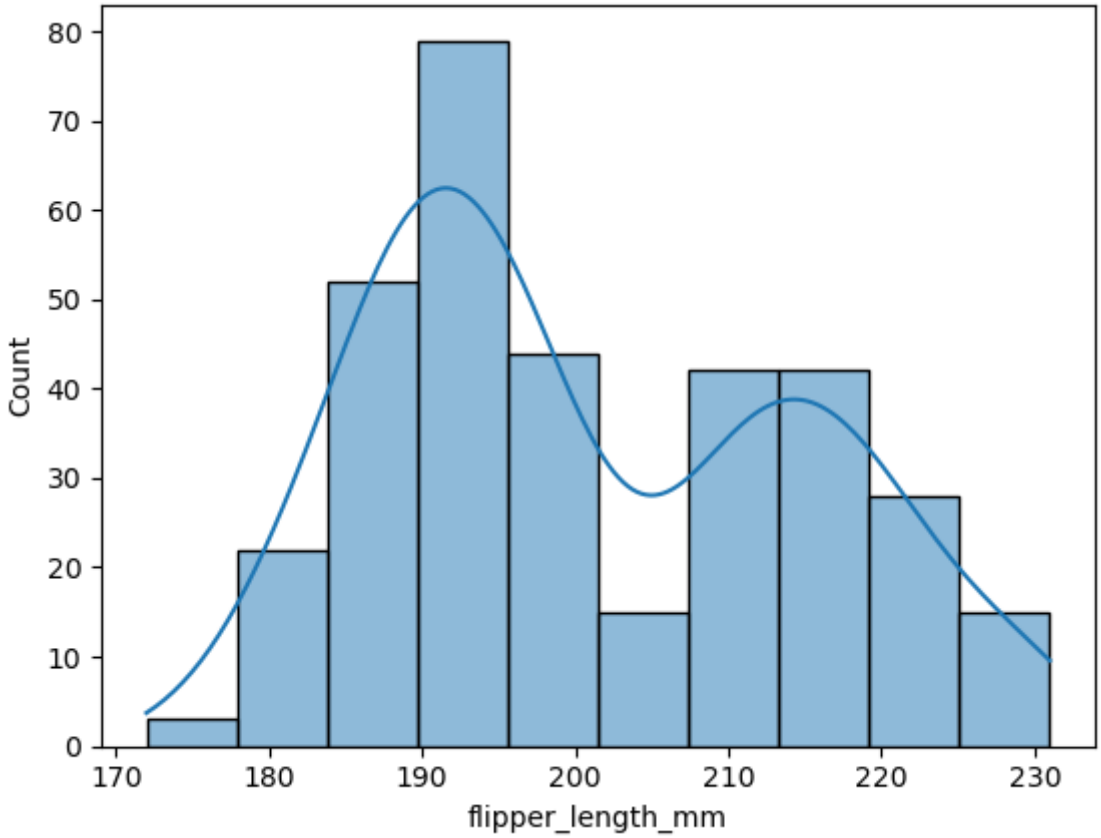
In [30]:

```
sns.histplot(data=penguins, x="flipper_length_mm", bins=30)
plt.show()
```



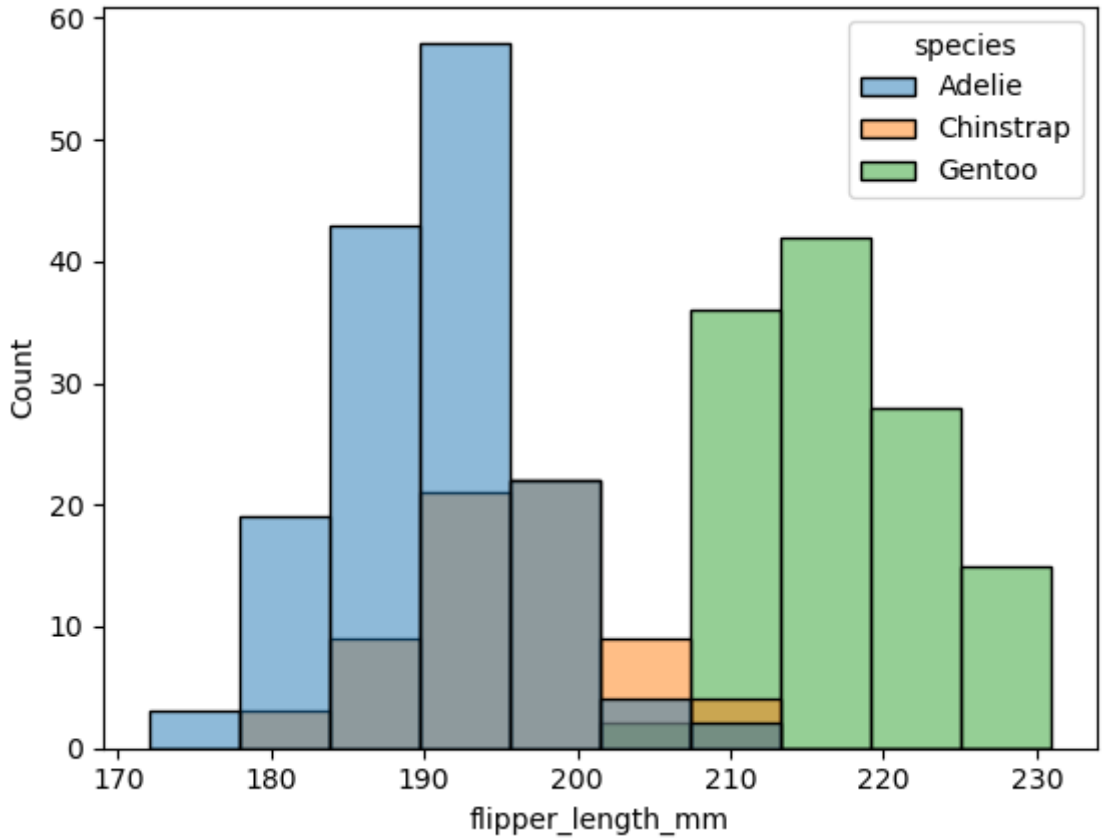
In [31]:

```
sns.histplot(data=penguins, x="flipper_length_mm", kde=True)
plt.show()
```



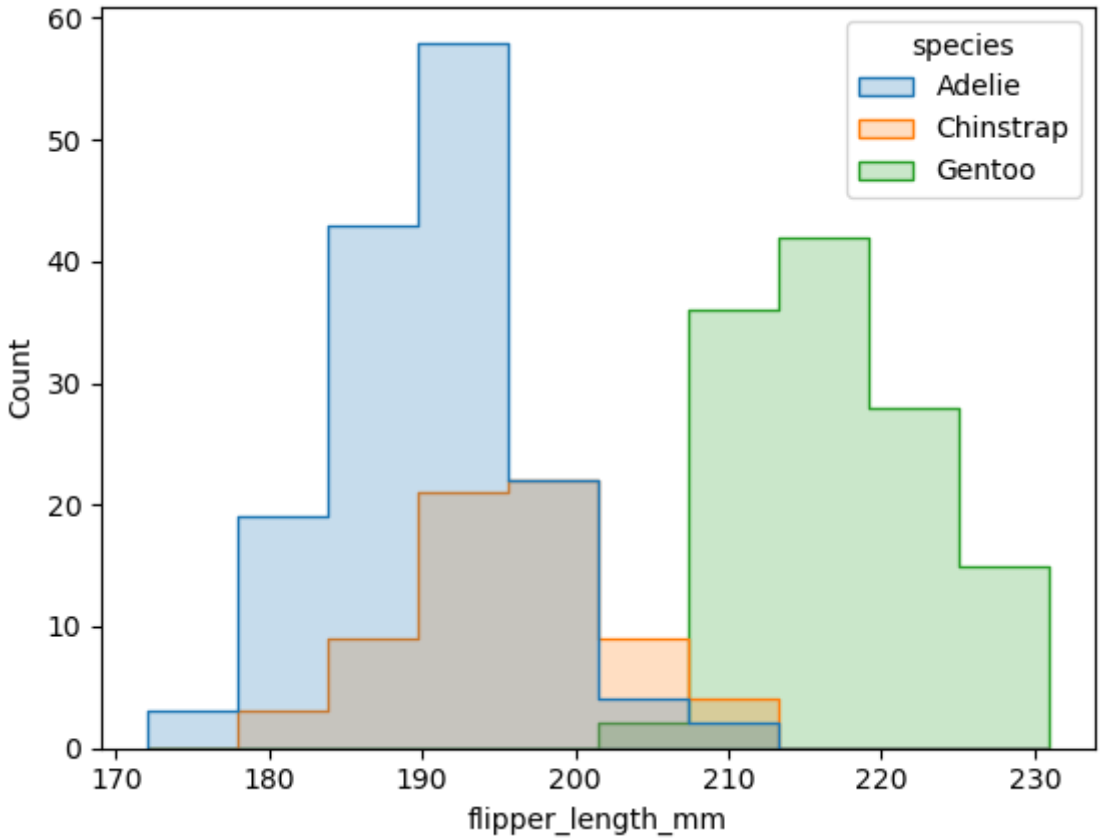
In [32]:

```
sns.histplot(data=penguins, x="flipper_length_mm", hue="species")
plt.show()
```



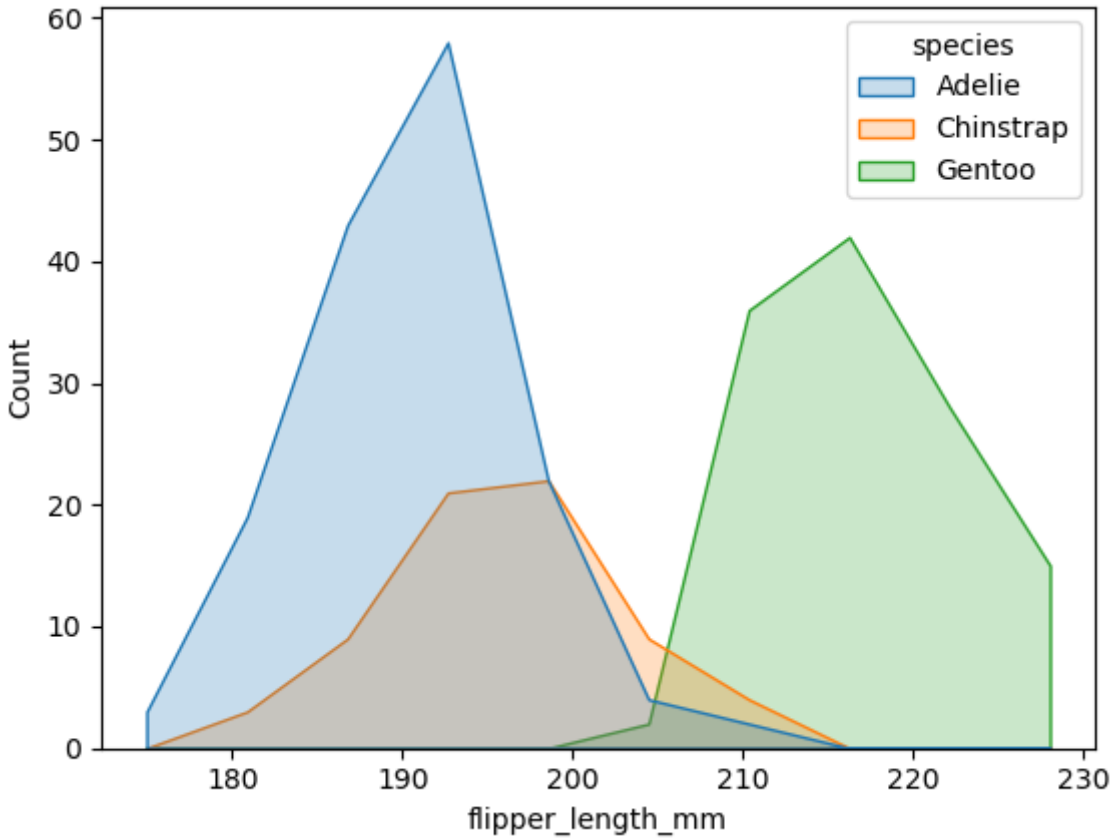
In [33]:

```
sns.histplot(penguins, x="flipper_length_mm", hue="species", element="step")
plt.show()
```



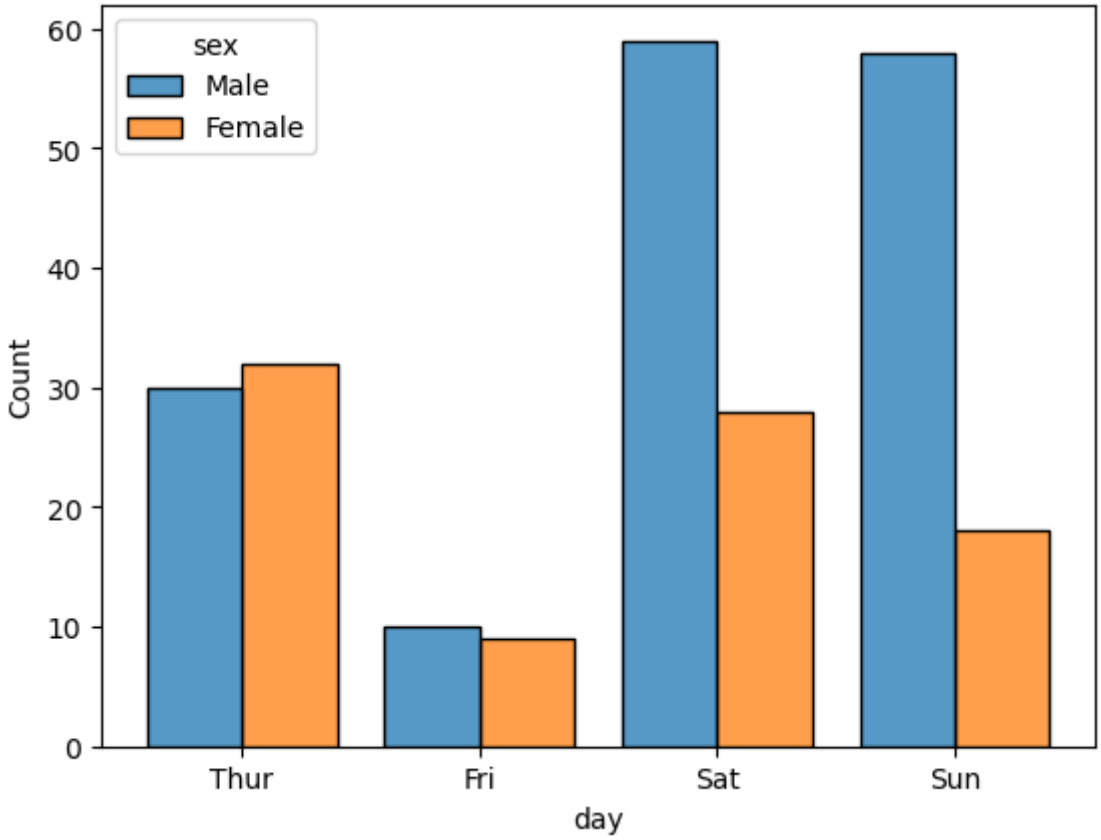
In [34]:

```
sns.histplot(penguins, x="flipper_length_mm", hue="species", element="poly")
plt.show()
```



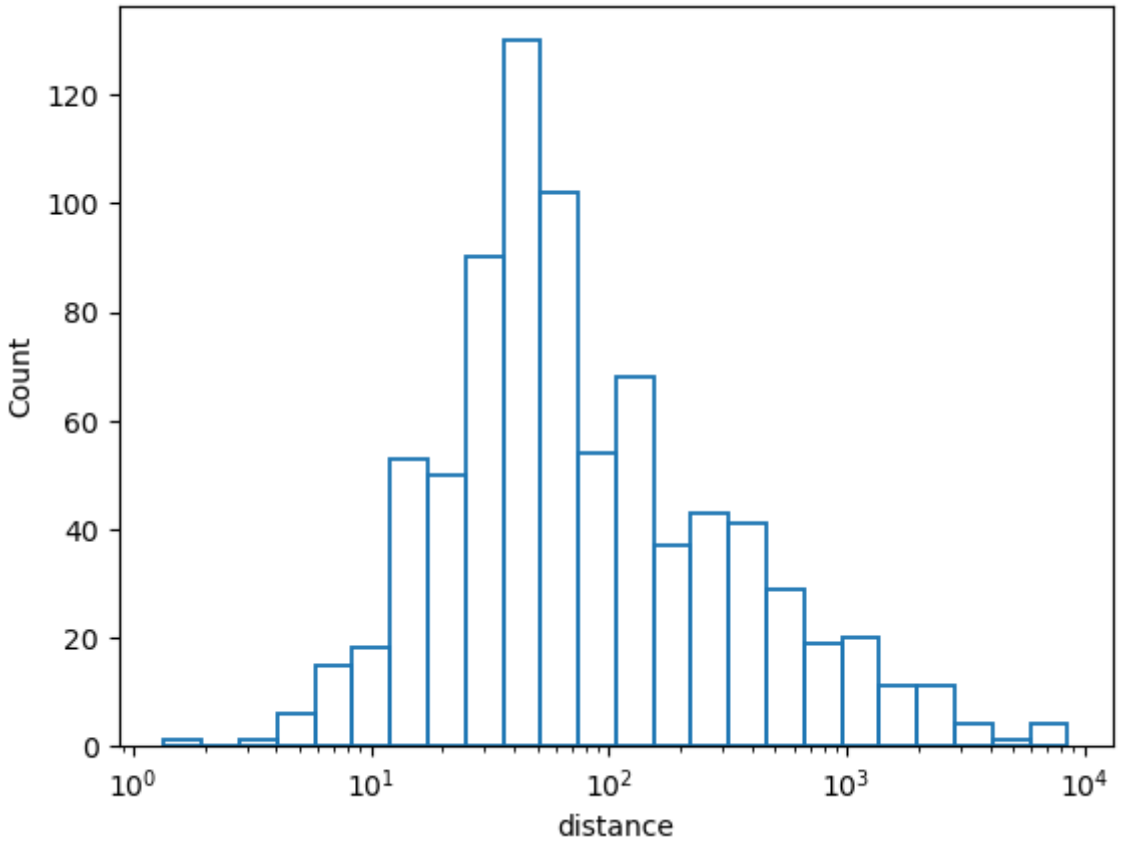
In [35]:

```
sns.histplot(data=tips, x="day", hue="sex", multiple="dodge", shrink=.8)
plt.show()
```



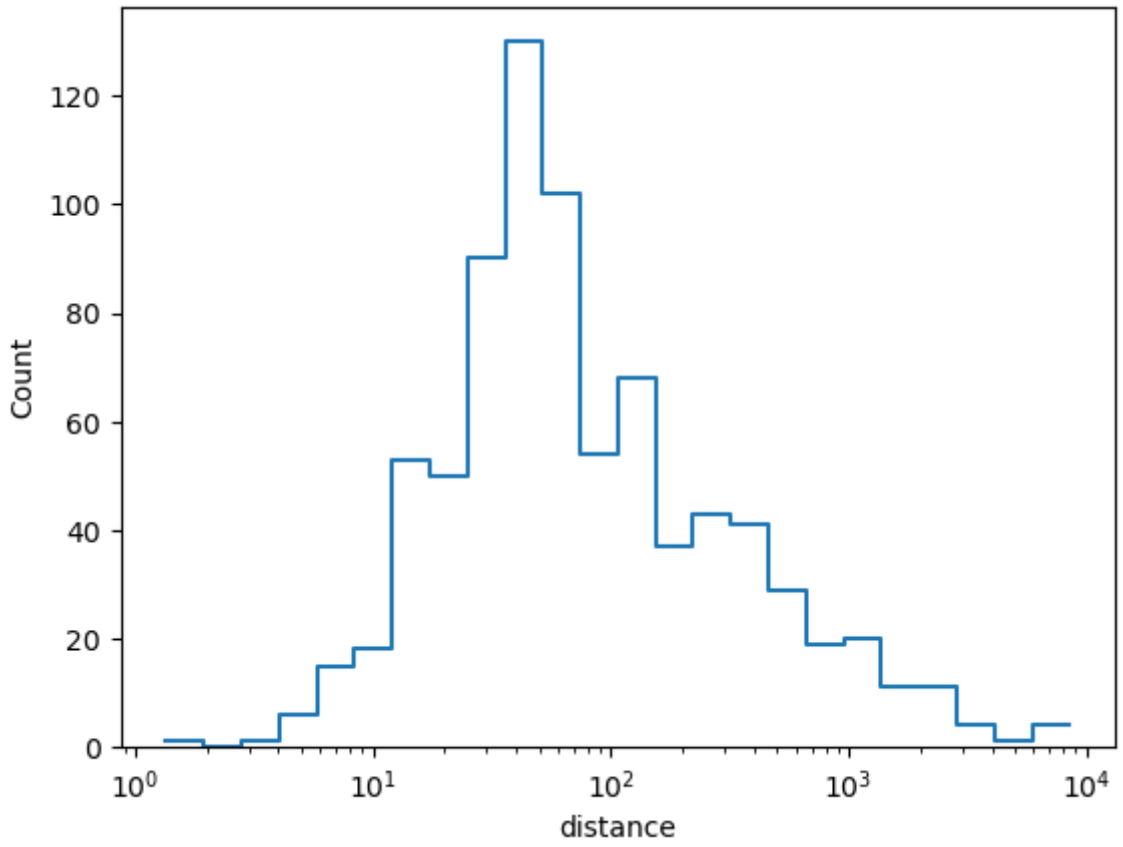
In [36]:

```
planets = sns.load_dataset("planets")
sns.histplot(data=planets, x="distance", log_scale=True, fill=False)
plt.show()
```



In [37]:

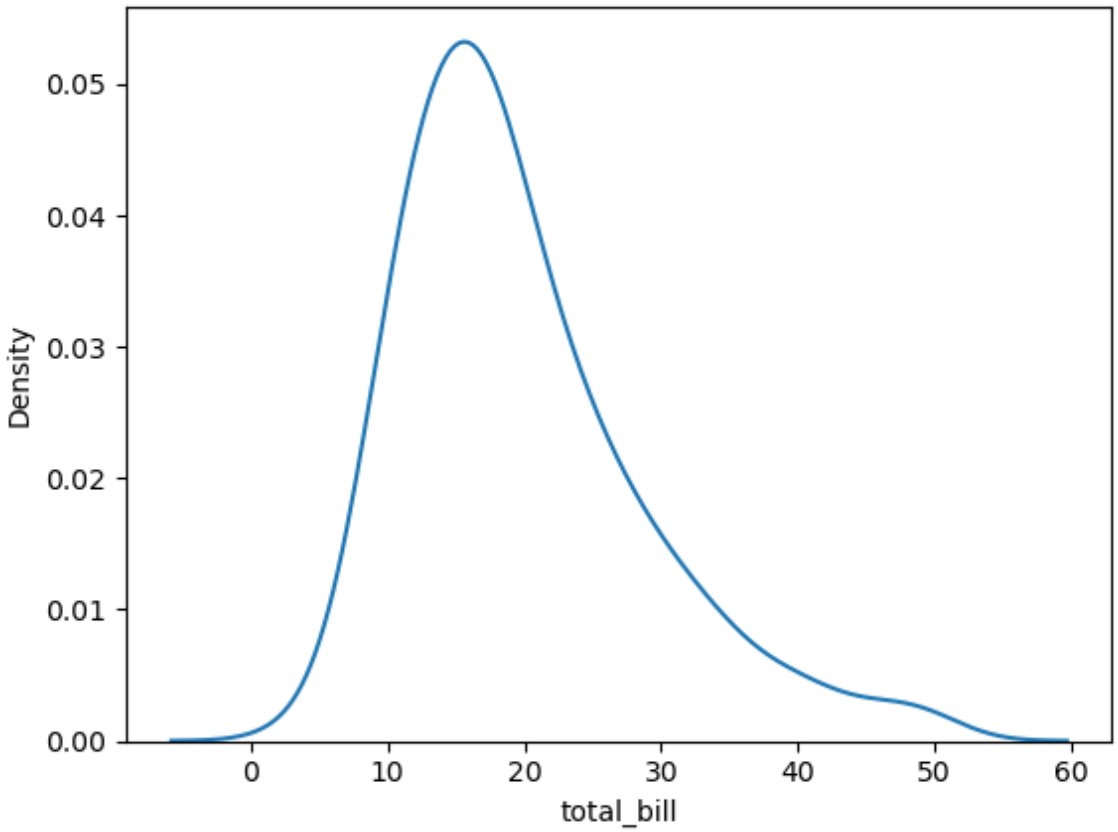
```
sns.histplot(data=planets, x="distance", log_scale=True, element="step", fill=False)
plt.show()
```



kdeplot

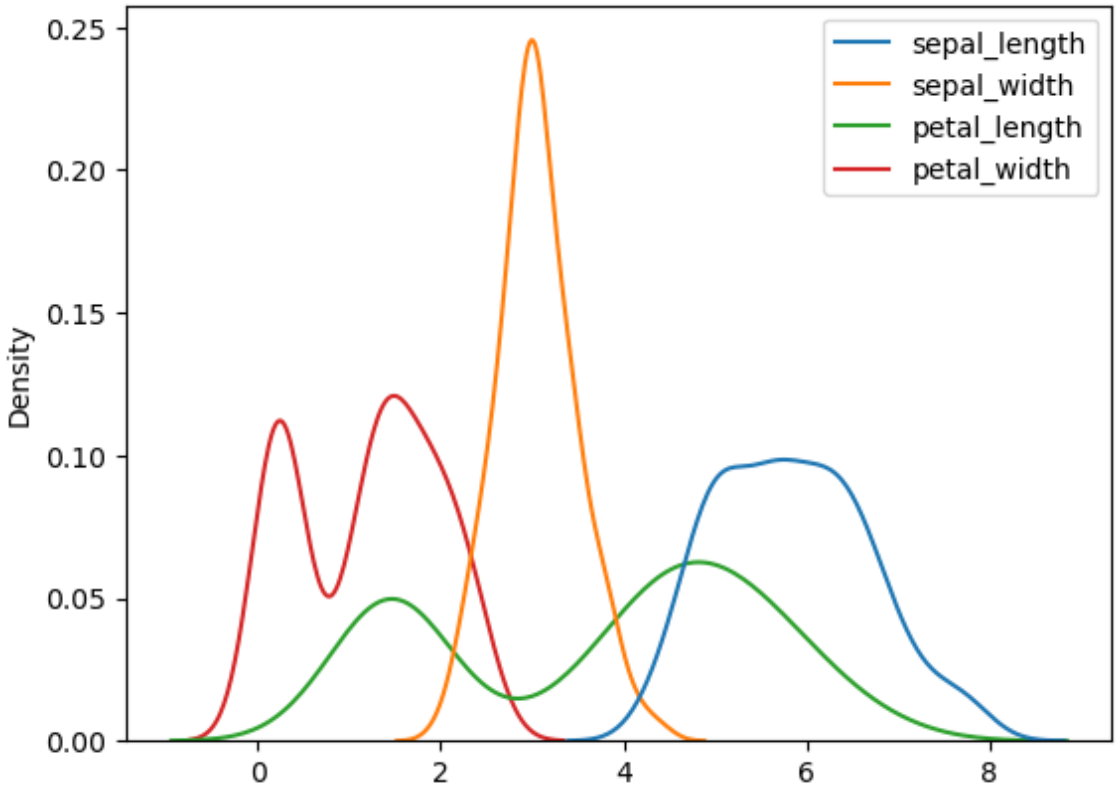
In [38]:

```
tips = sns.load_dataset("tips")
sns.kdeplot(data=tips, x="total_bill")
plt.show()
```



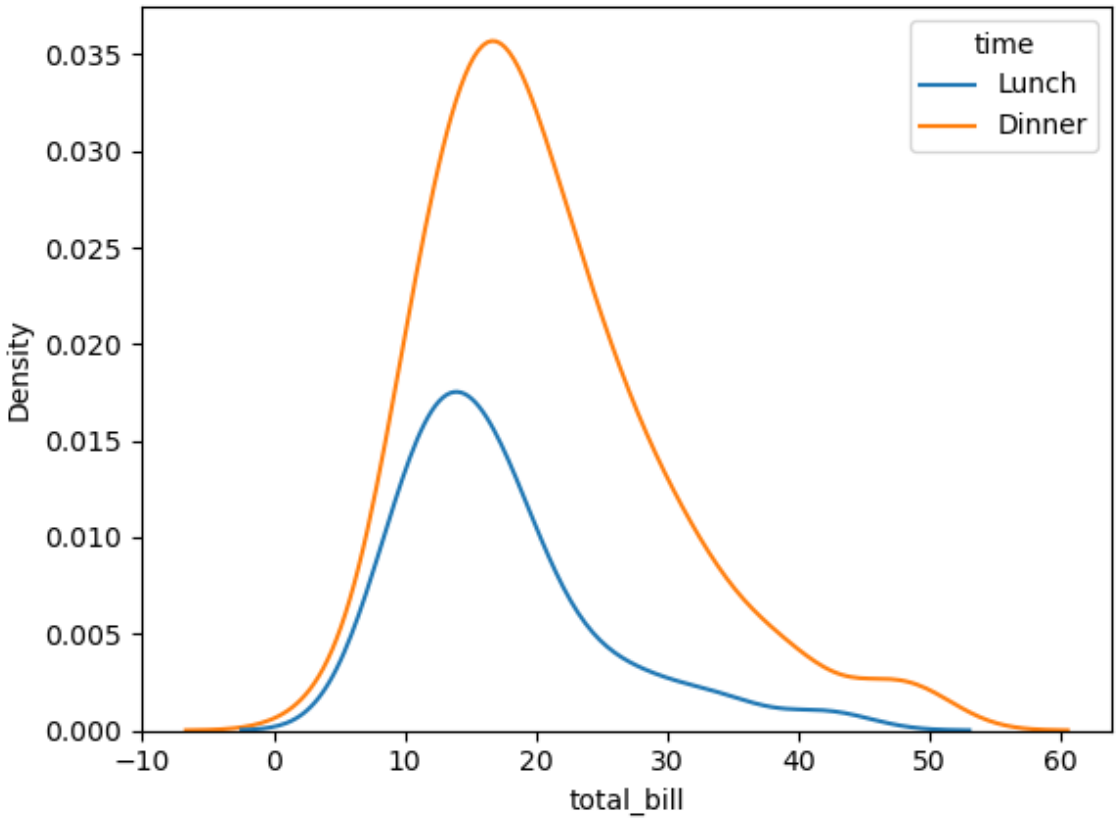
In [39]:

```
iris = sns.load_dataset("iris")
sns.kdeplot(data=iris)
plt.show()
```



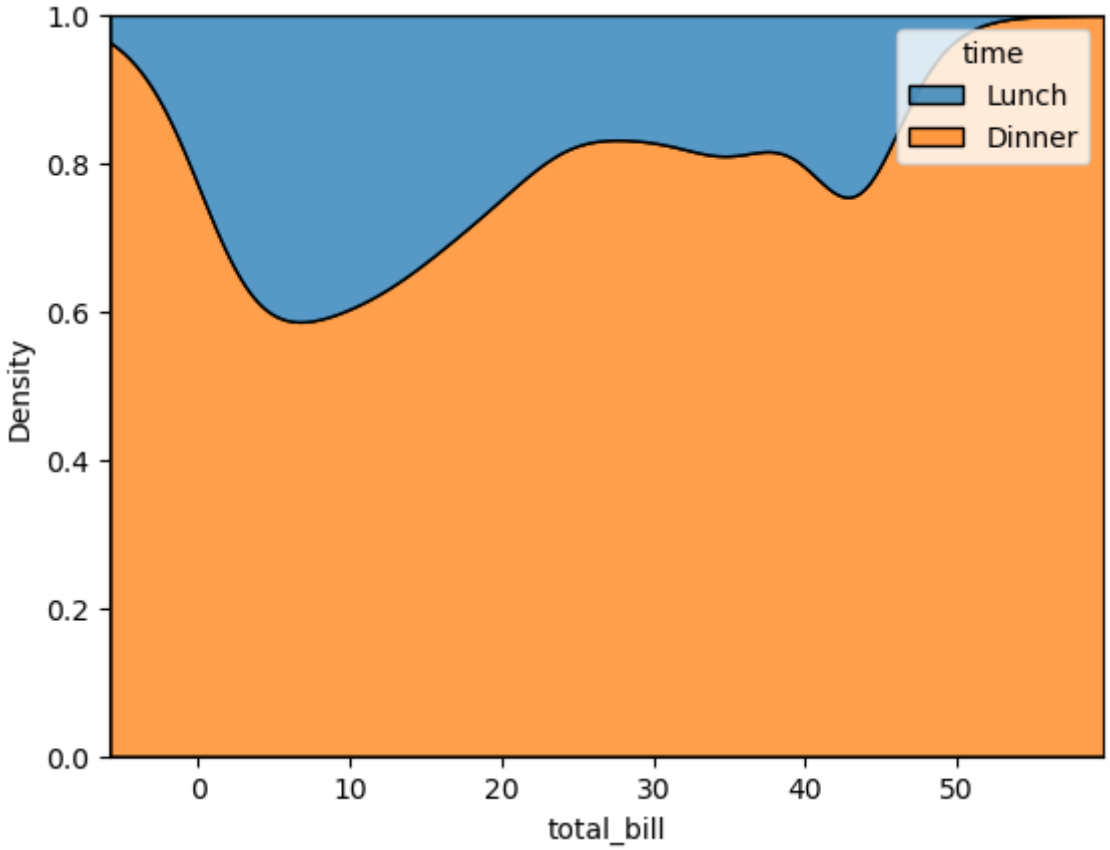
In [40]:

```
sns.kdeplot(data=tips, x="total_bill", hue="time")
plt.show()
```



In [41]:

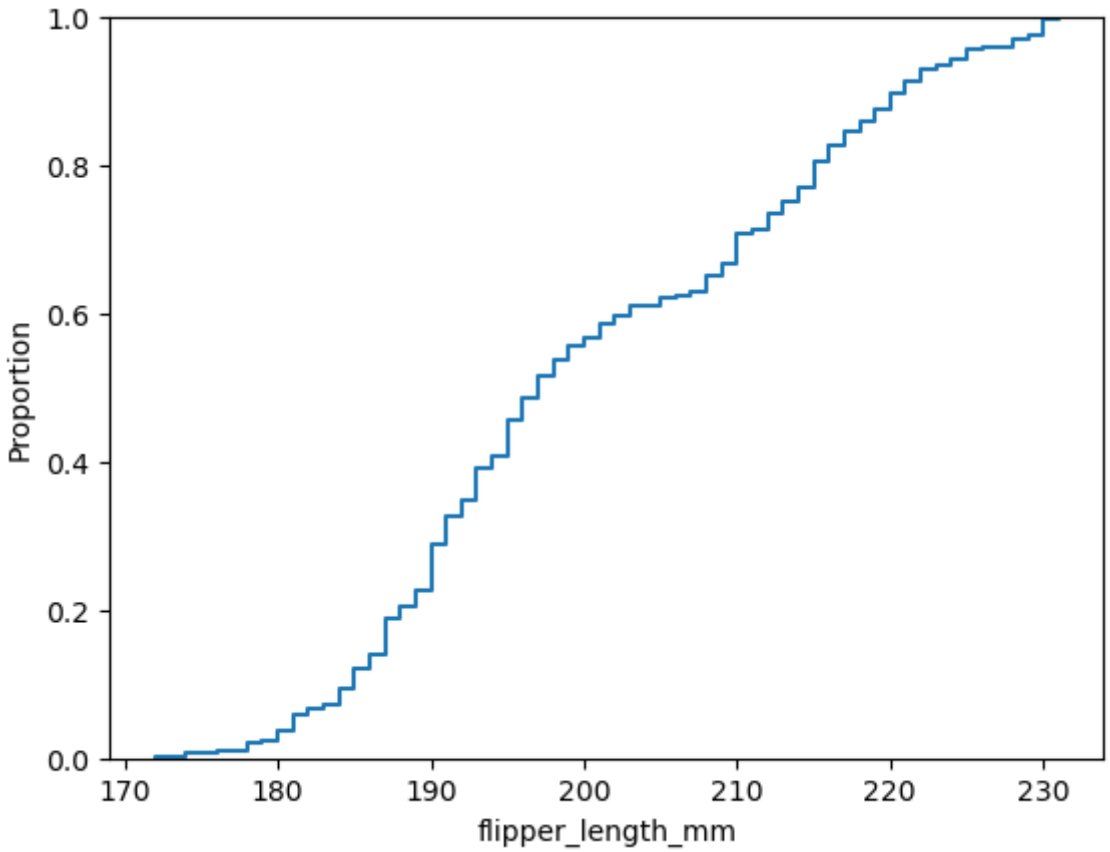
```
sns.kdeplot(data=tips, x="total_bill", hue="time", multiple="fill")
plt.show()
```



ecdfplot

In [42]:

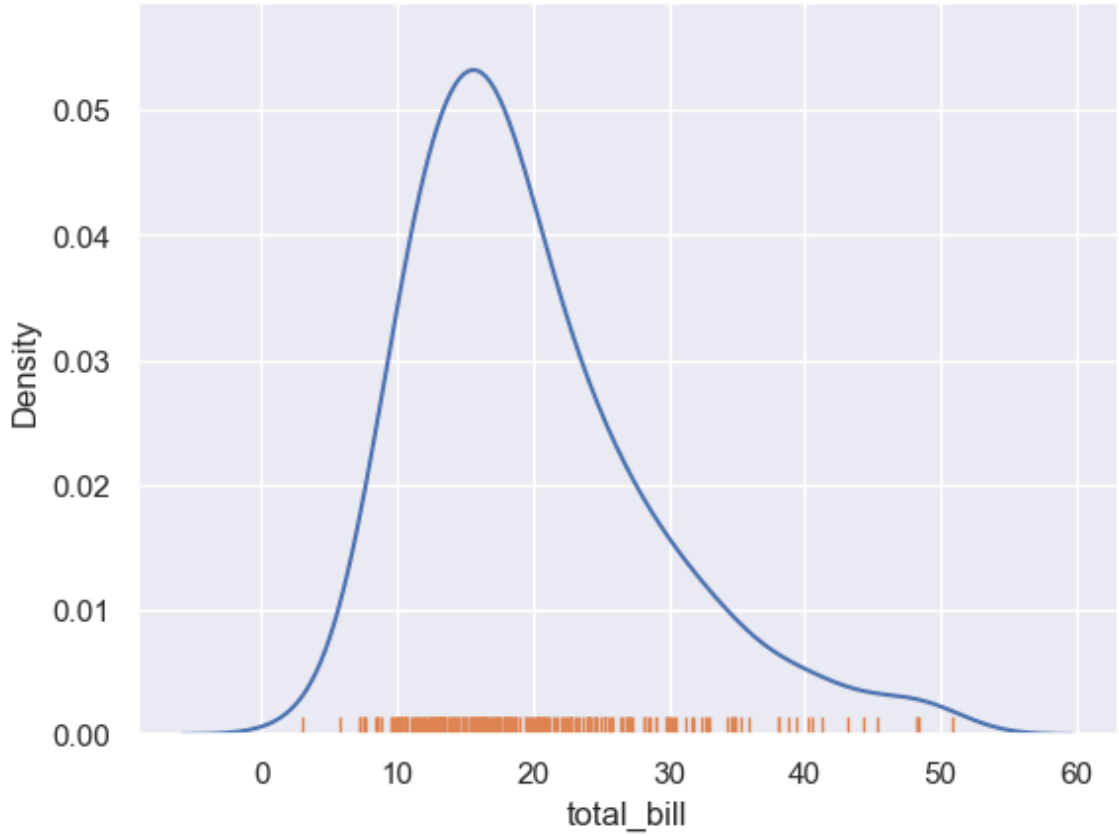
```
penguins = sns.load_dataset("penguins")
sns.ecdfplot(data=penguins, x="flipper_length_mm")
plt.show()
```



rugplot

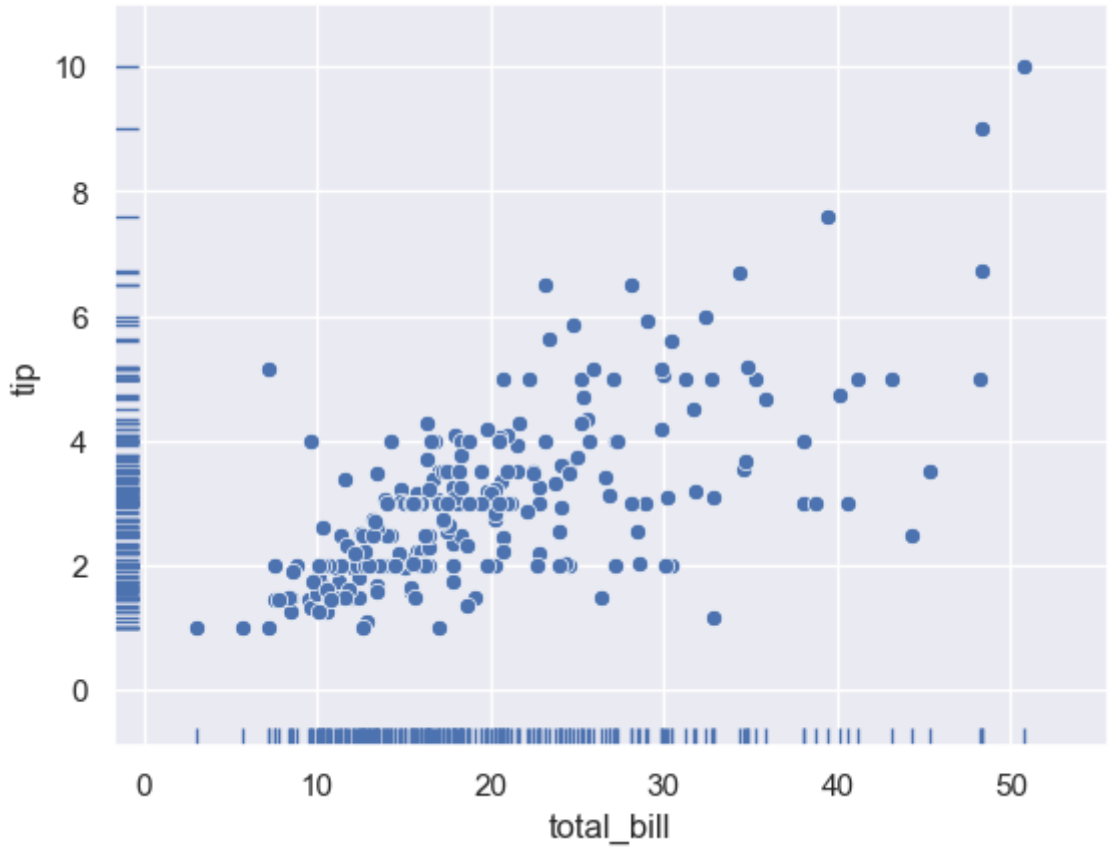
In [43]:

```
import seaborn as sns; sns.set_theme()
tips = sns.load_dataset("tips")
sns.kdeplot(data=tips, x="total_bill")
sns.rugplot(data=tips, x="total_bill")
plt.show()
```



In [44]:

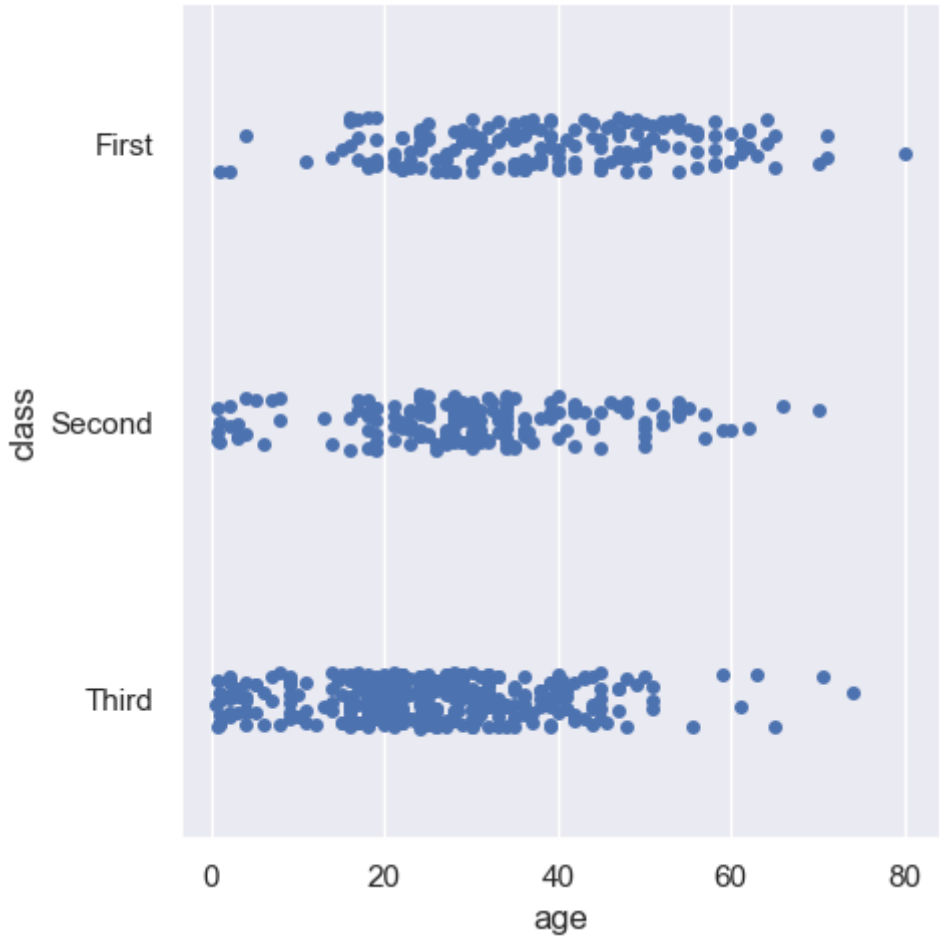
```
sns.scatterplot(data=tips, x="total_bill", y="tip")
sns.rugplot(data=tips, x="total_bill", y="tip")
plt.show()
```



catplot

In [45]:

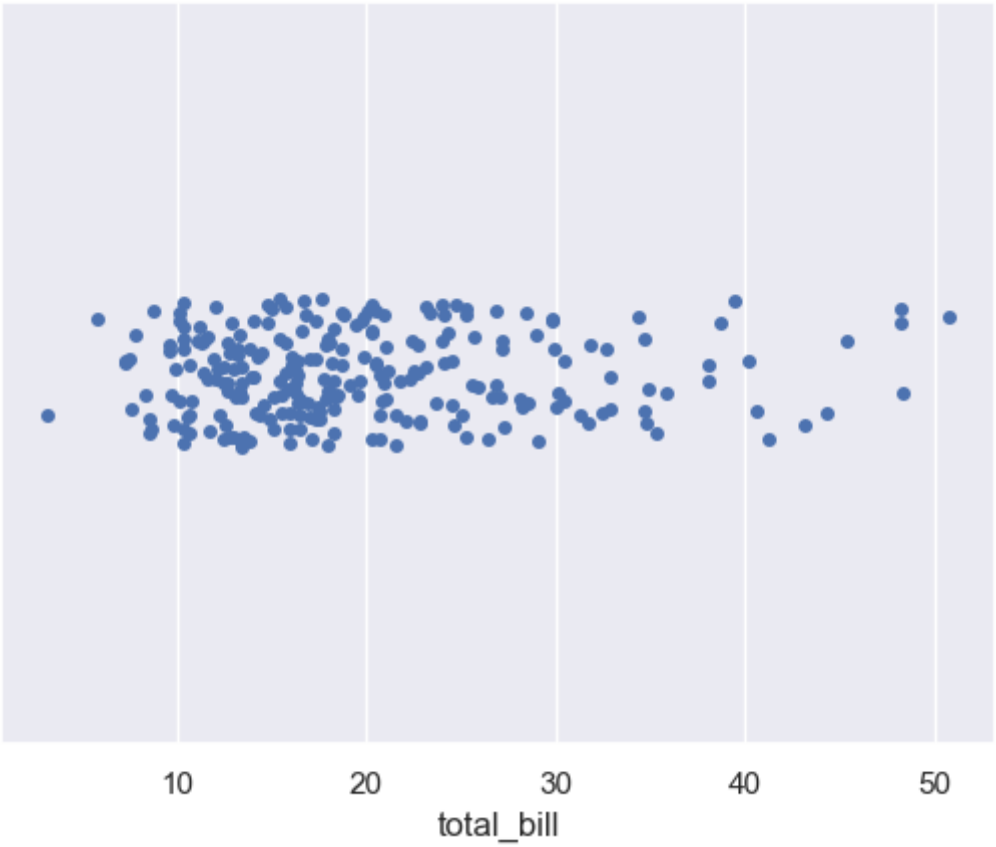
```
df = sns.load_dataset("titanic")
sns.catplot(data=df, x="age", y="class")
plt.show()
```



stripplot

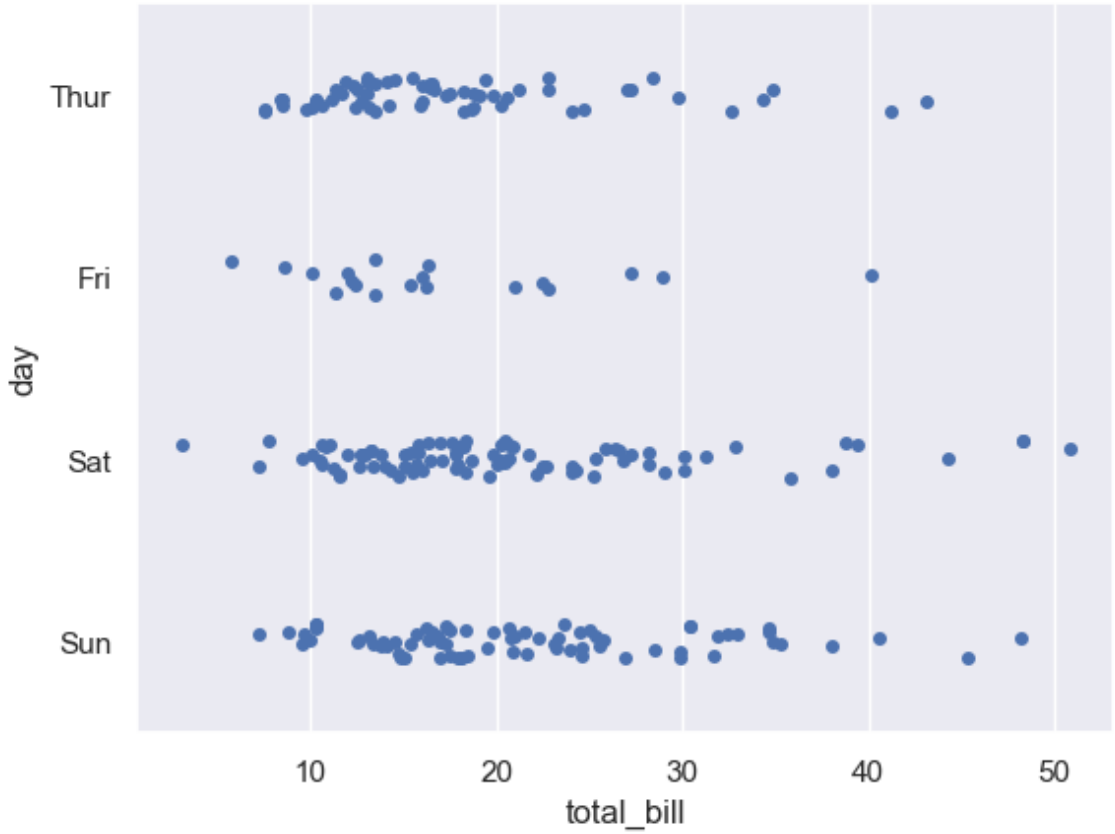
In [46]:

```
sns.stripplot(data=tips, x="total_bill")
plt.show()
```



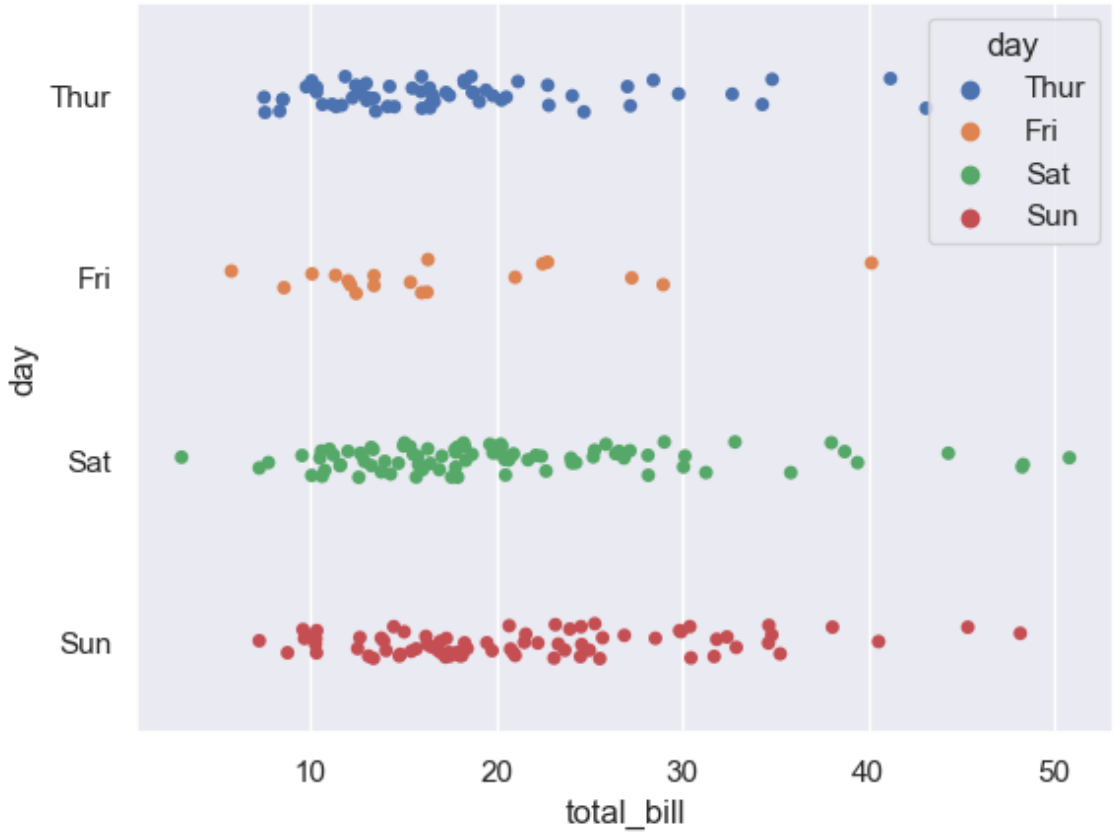
In [47]:

```
sns.stripplot(data=tips, x="total_bill", y="day")
plt.show()
```



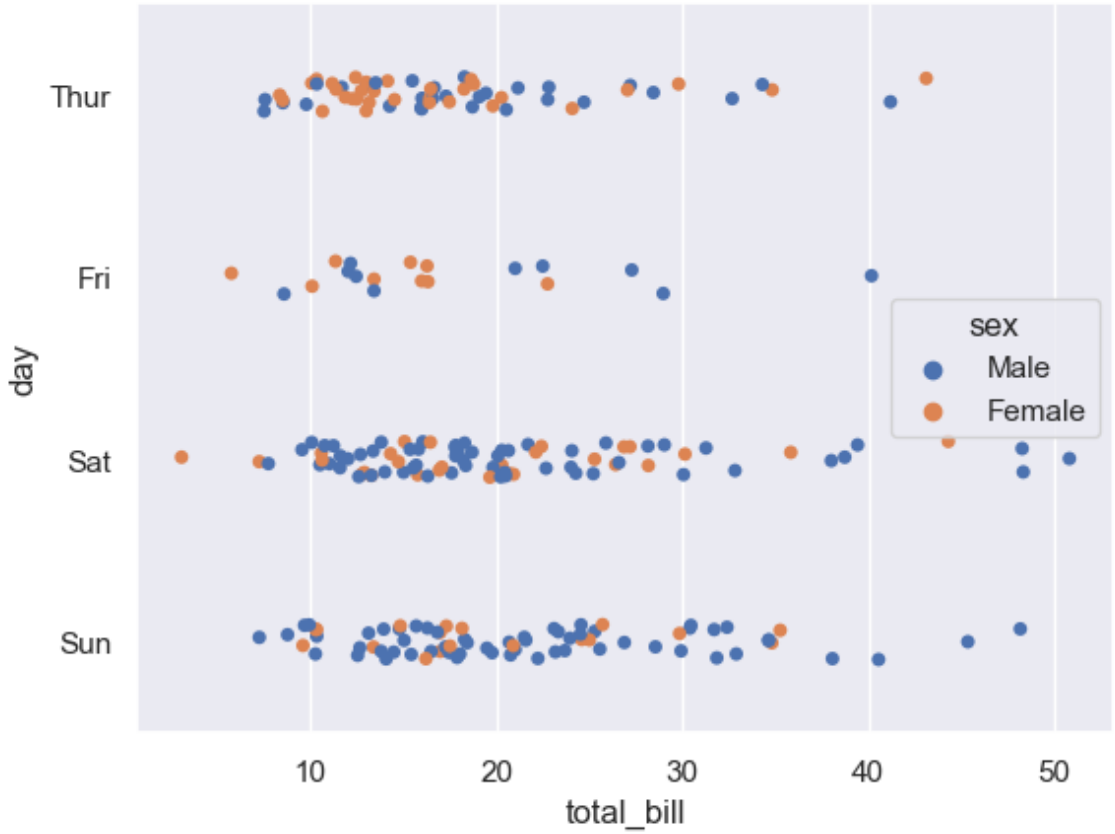
In [48]:

```
sns.stripplot(data=tips, x="total_bill", y="day", hue="day")
plt.show()
```



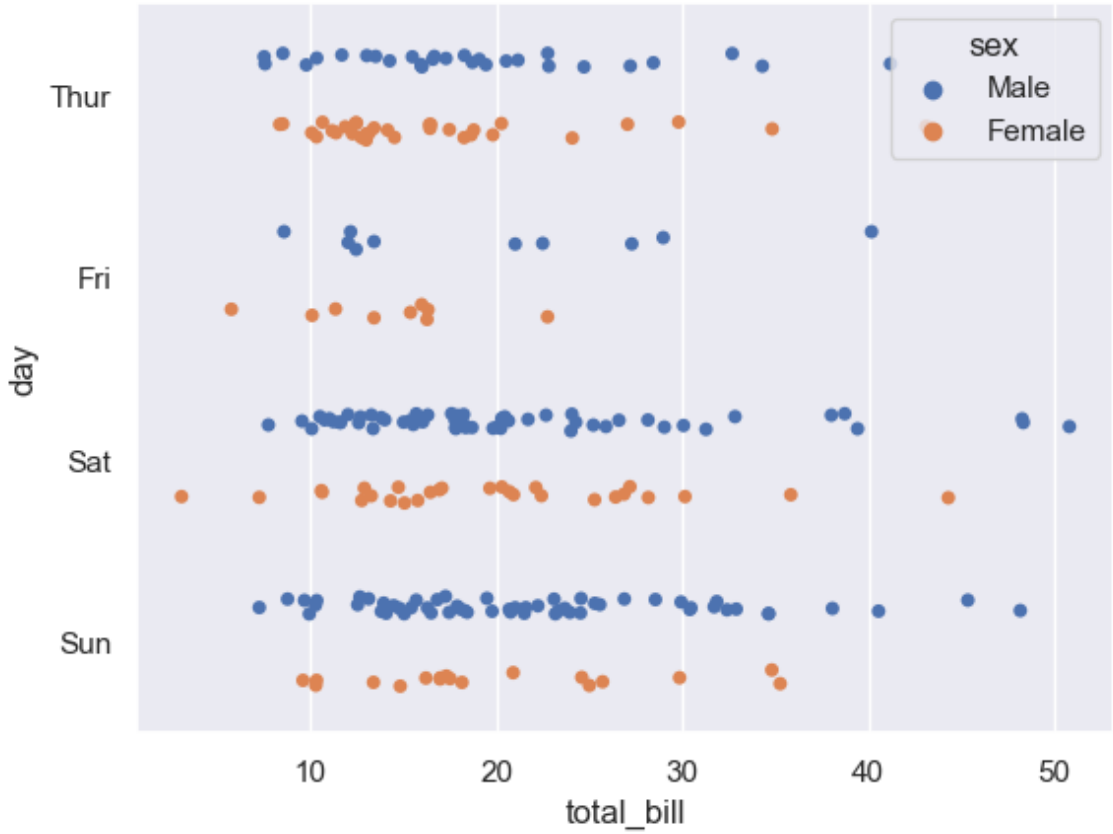
In [49]:

```
sns.stripplot(data=tips, x="total_bill", y="day", hue="sex")
plt.show()
```



In [50]:

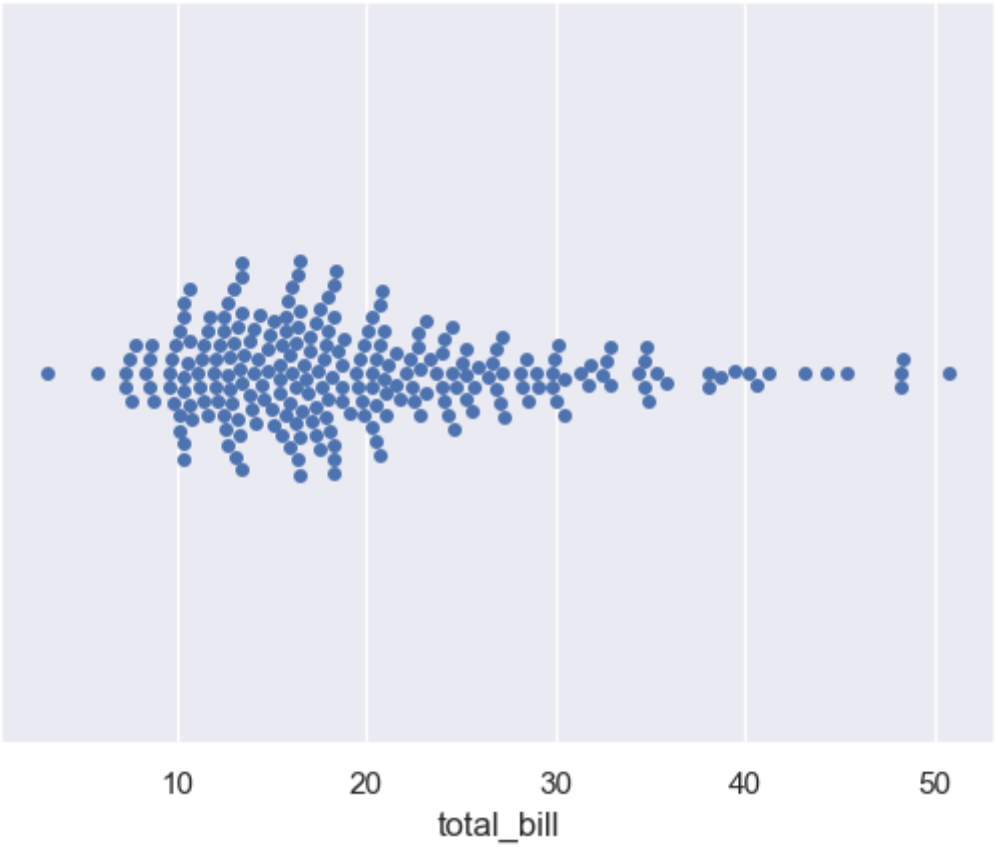
```
sns.stripplot(data=tips, x="total_bill", y="day", hue="sex", dodge=True)
plt.show()
```



swarmplot

In [51]:

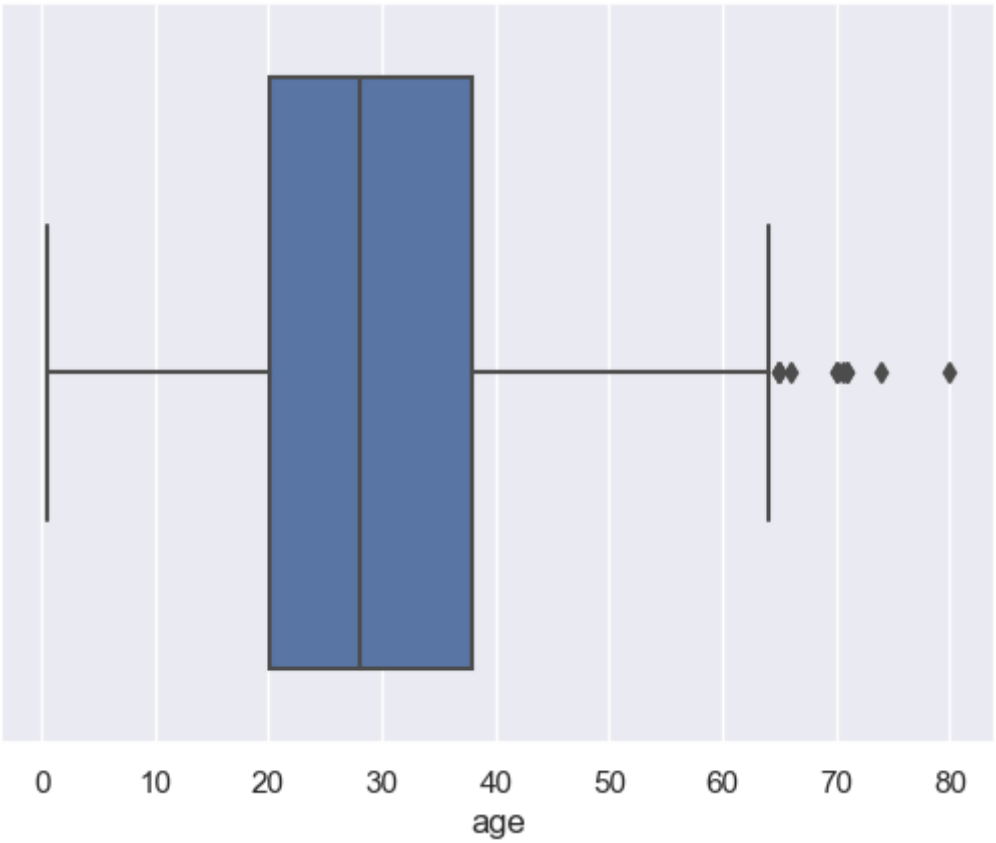
```
sns.swarmplot(data=tips, x="total_bill")
plt.show()
```



boxplot

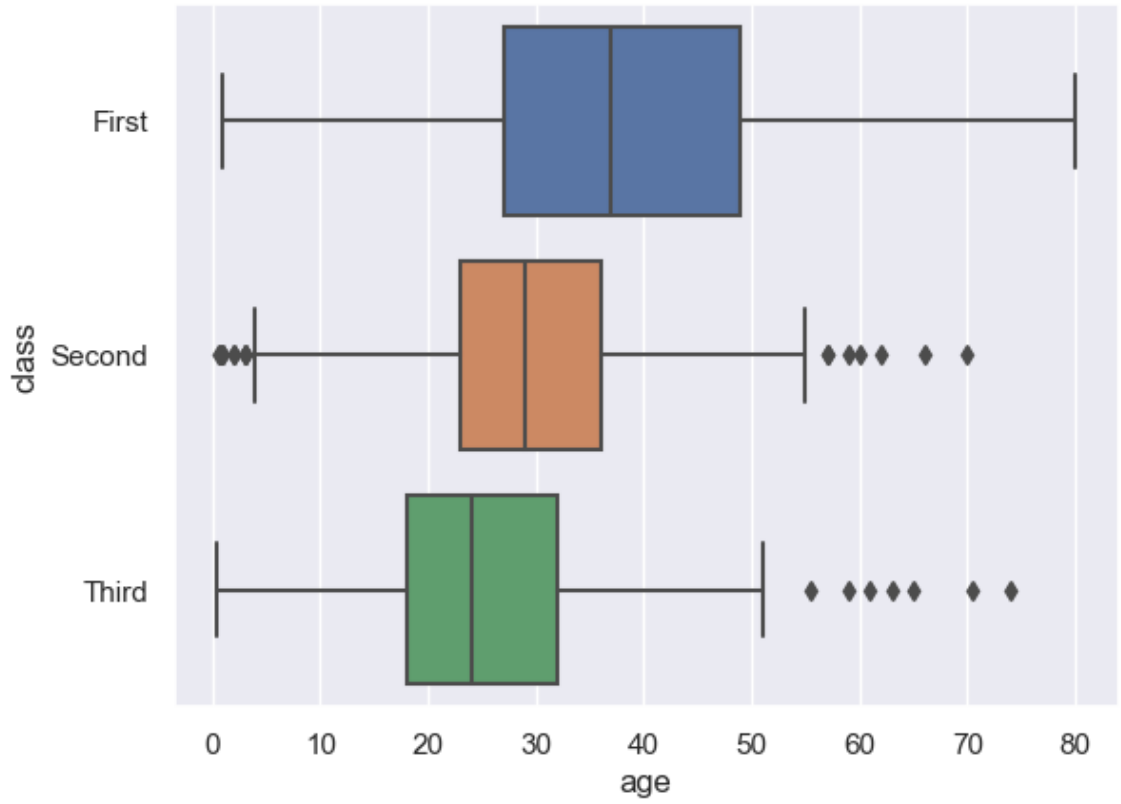
In [52]:

```
df = sns.load_dataset("titanic")
sns.boxplot(data = df,x="age")
plt.show()
```



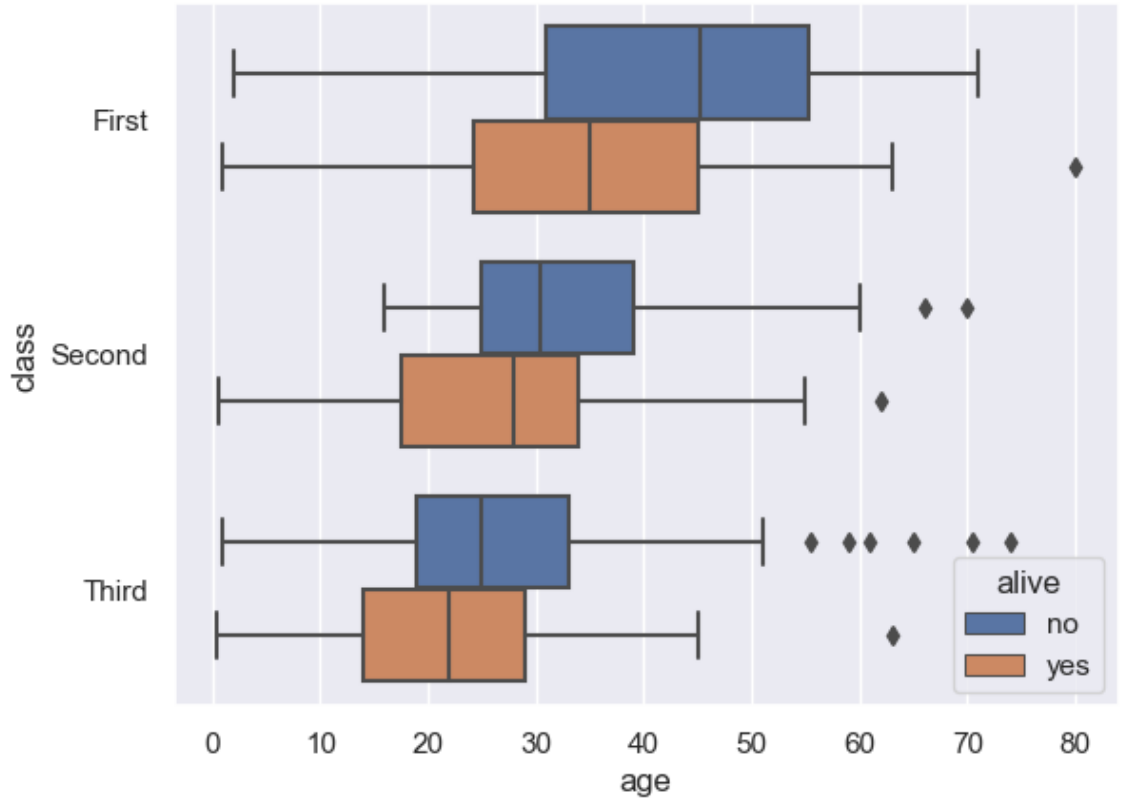
In [53]:

```
sns.boxplot(data=df, x="age", y="class")
plt.show()
```



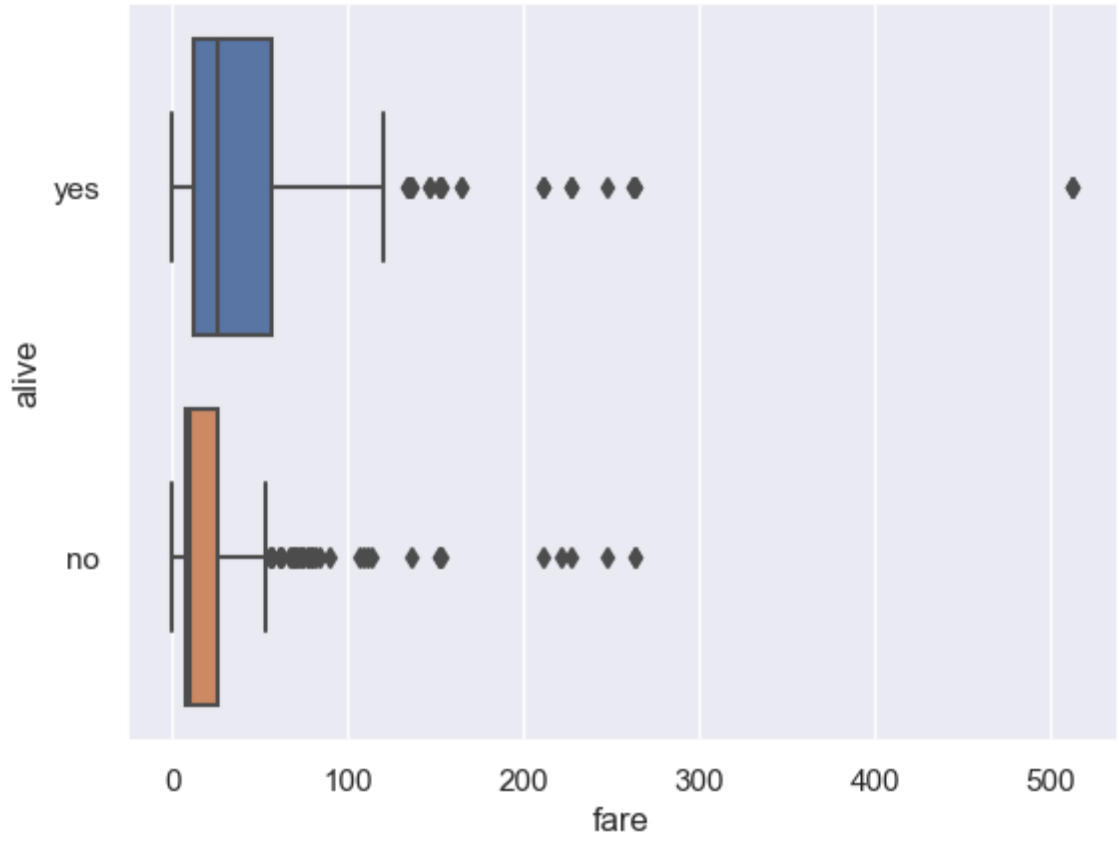
In [54]:

```
sns.boxplot(data=df, x="age", y="class", hue="alive")
plt.show()
```



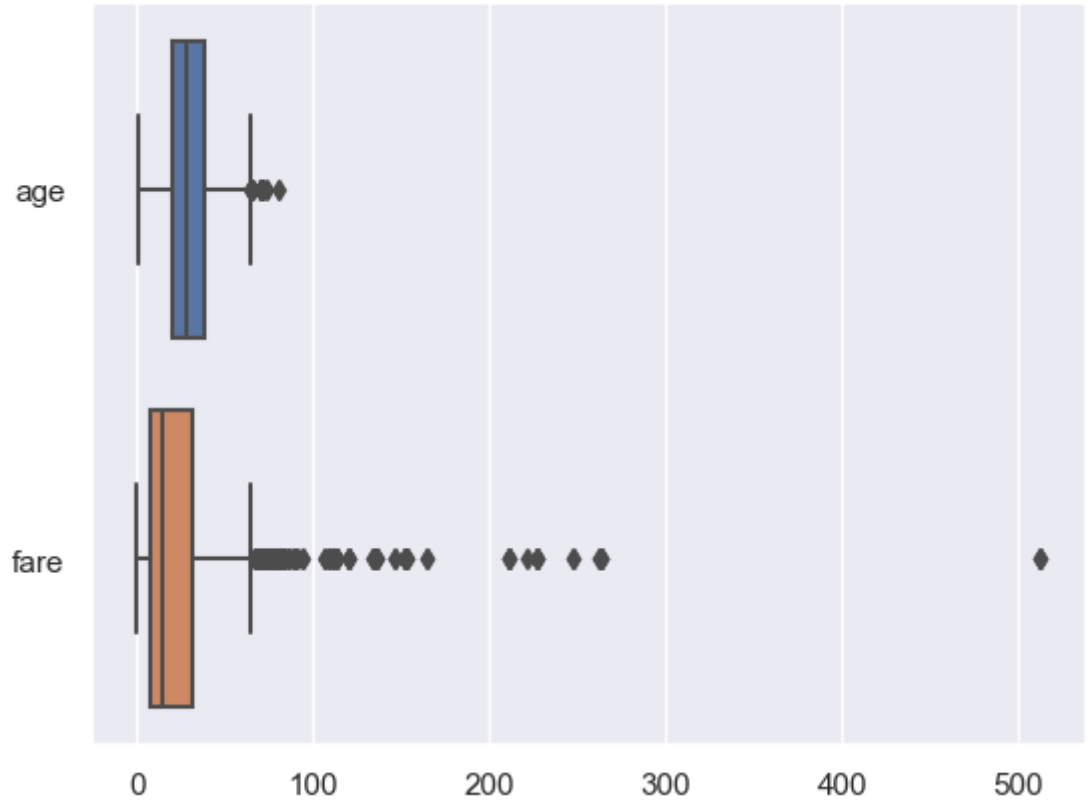
In [55]:

```
sns.boxplot(data=df, x="fare", y="alive", order=["yes", "no"])
plt.show()
```



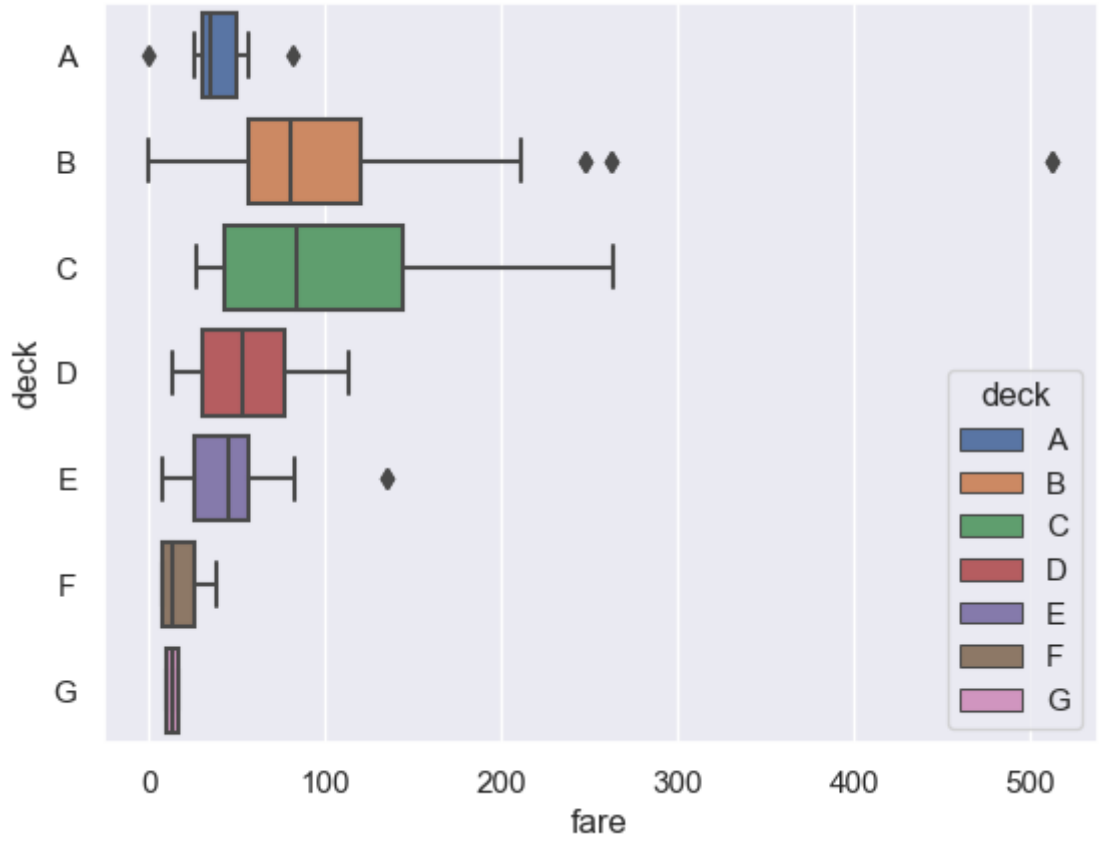
In [56]:

```
sns.boxplot(data=df[["age", "fare"]], orient="h")
plt.show()
```



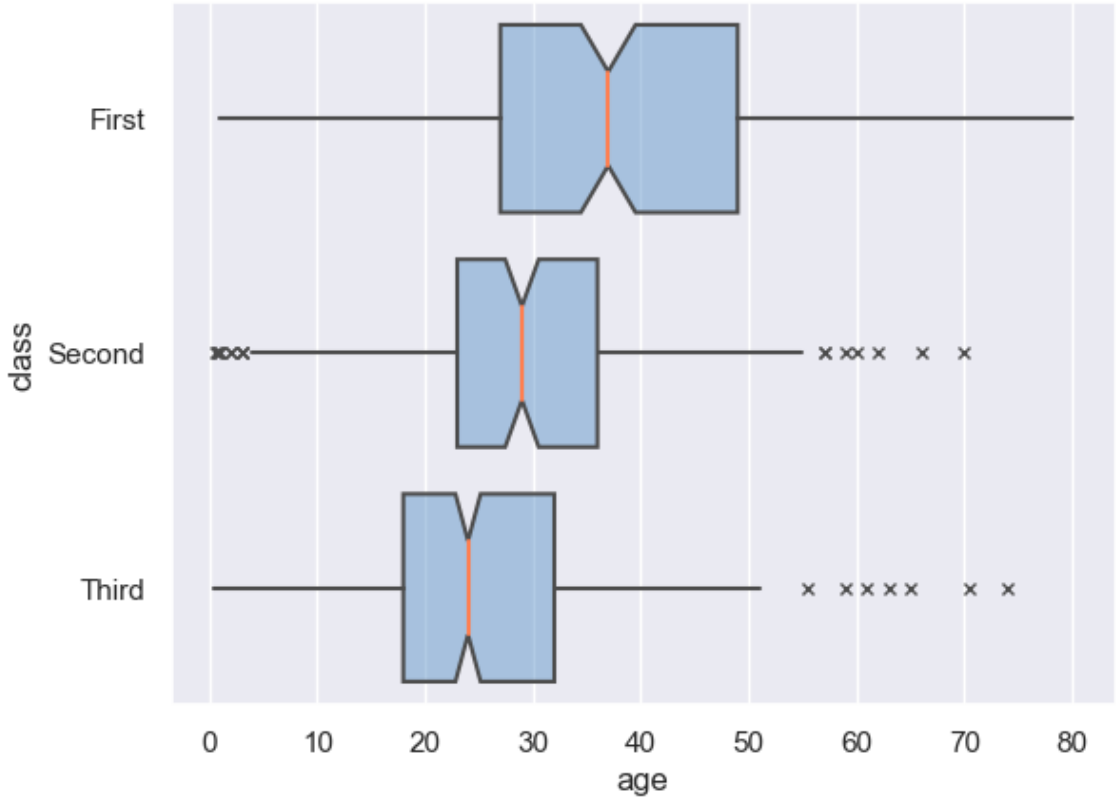
In [57]:

```
sns.boxplot(data=df, x="fare", y="deck", hue="deck", dodge=False)
plt.show()
```



In [58]:

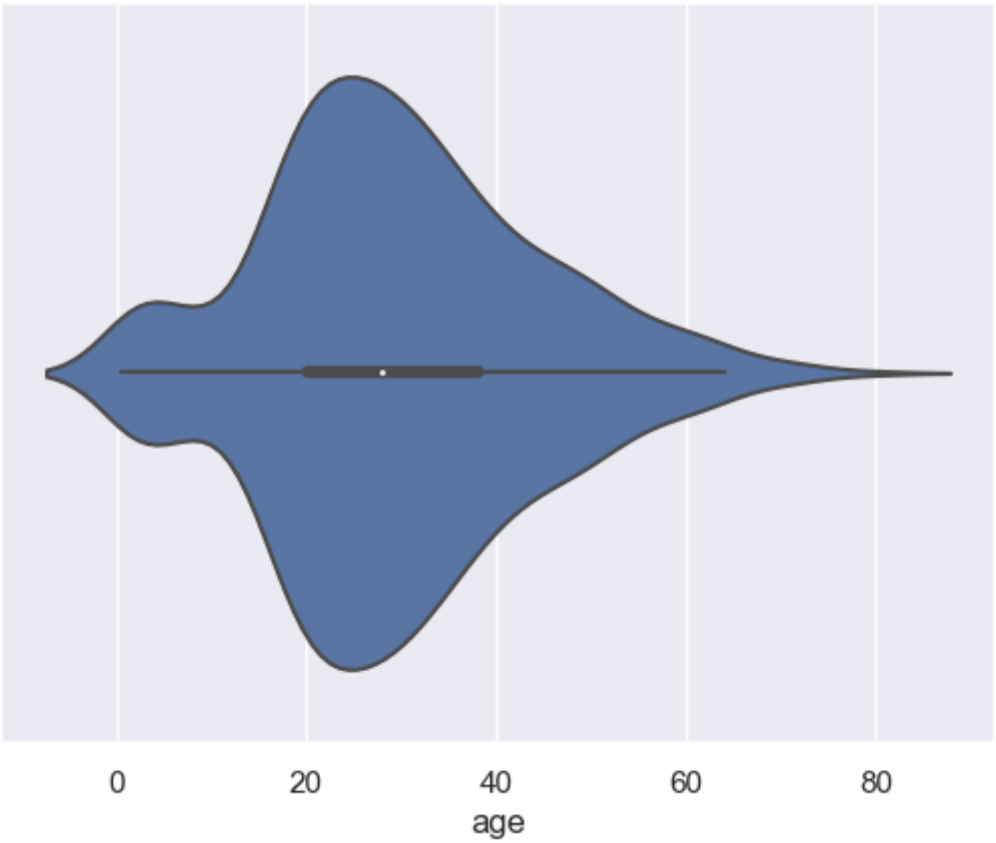
```
sns.boxplot(
    data=df, x="age", y="class",
    notch=True, showcaps=False,
    flierprops={"marker": "x"},
    boxprops={"facecolor": (.4, .6, .8, .5)},
    medianprops={"color": "coral"},
)
plt.show()
```



violinplot

In [59]:

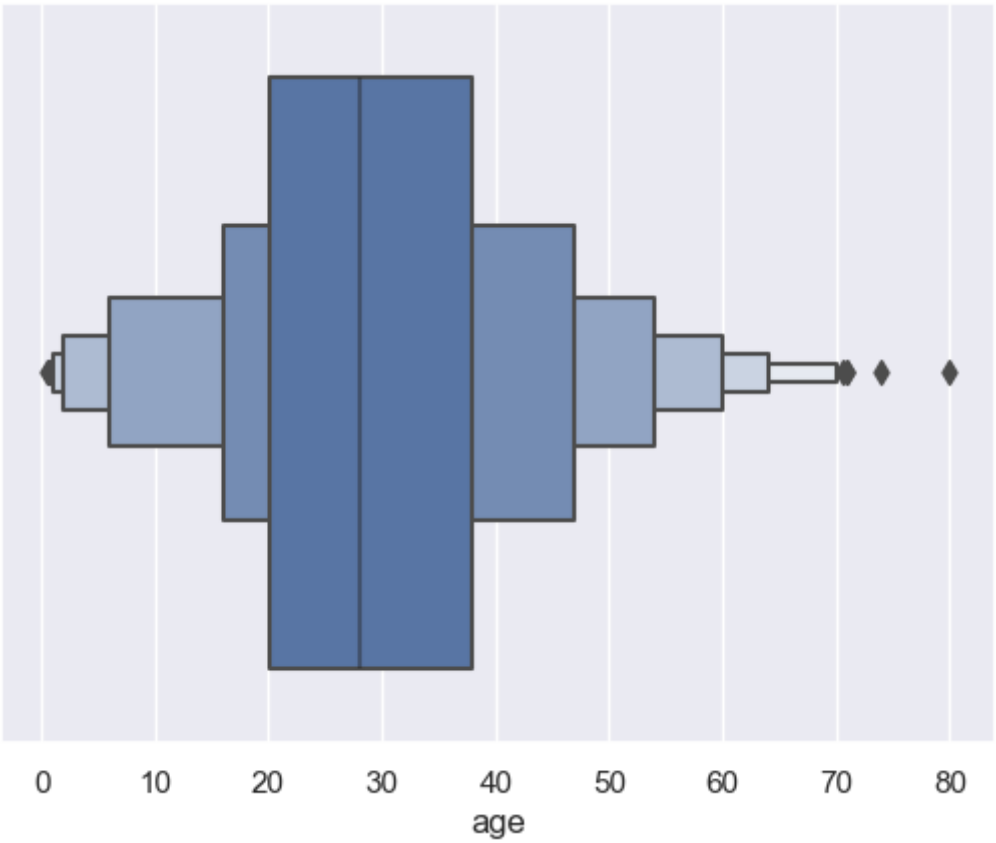
```
df = sns.load_dataset("titanic")
sns.violinplot(x=df["age"])
plt.show()
```



boxenplot

In [60]:

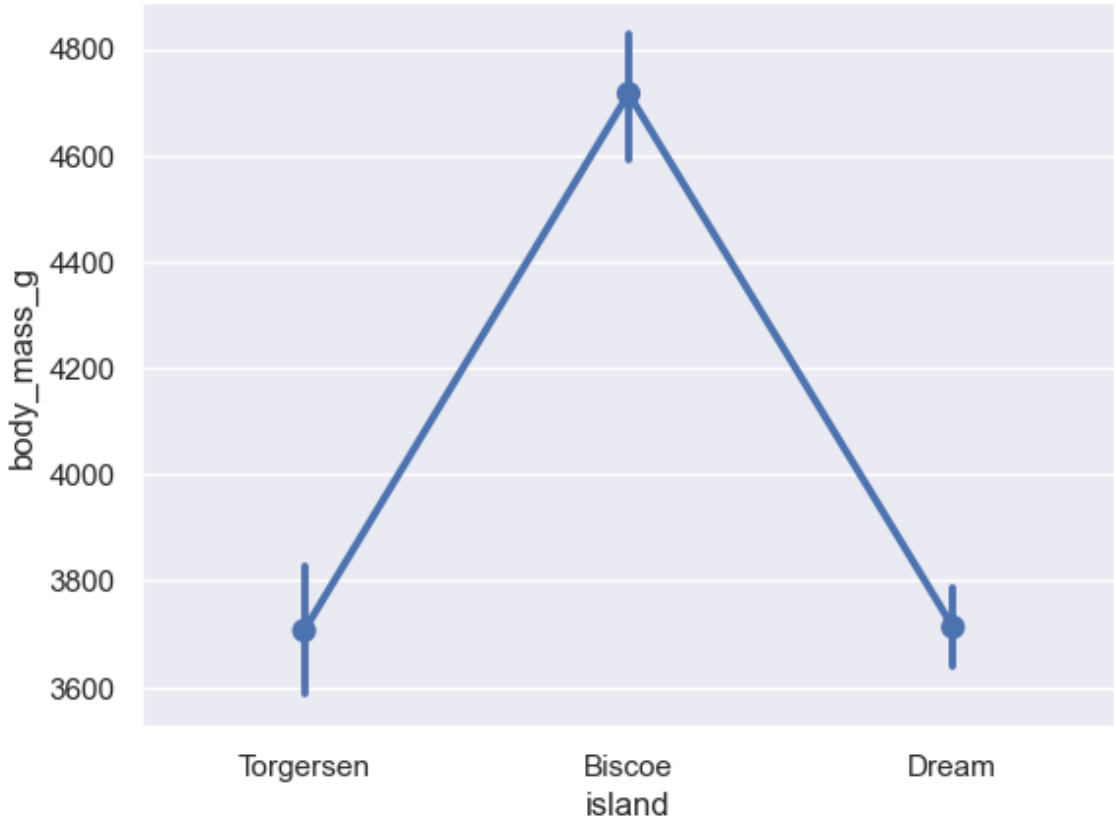
```
df = sns.load_dataset("titanic")
sns.boxenplot(x=df["age"])
plt.show()
```



pointplot

In [61]:

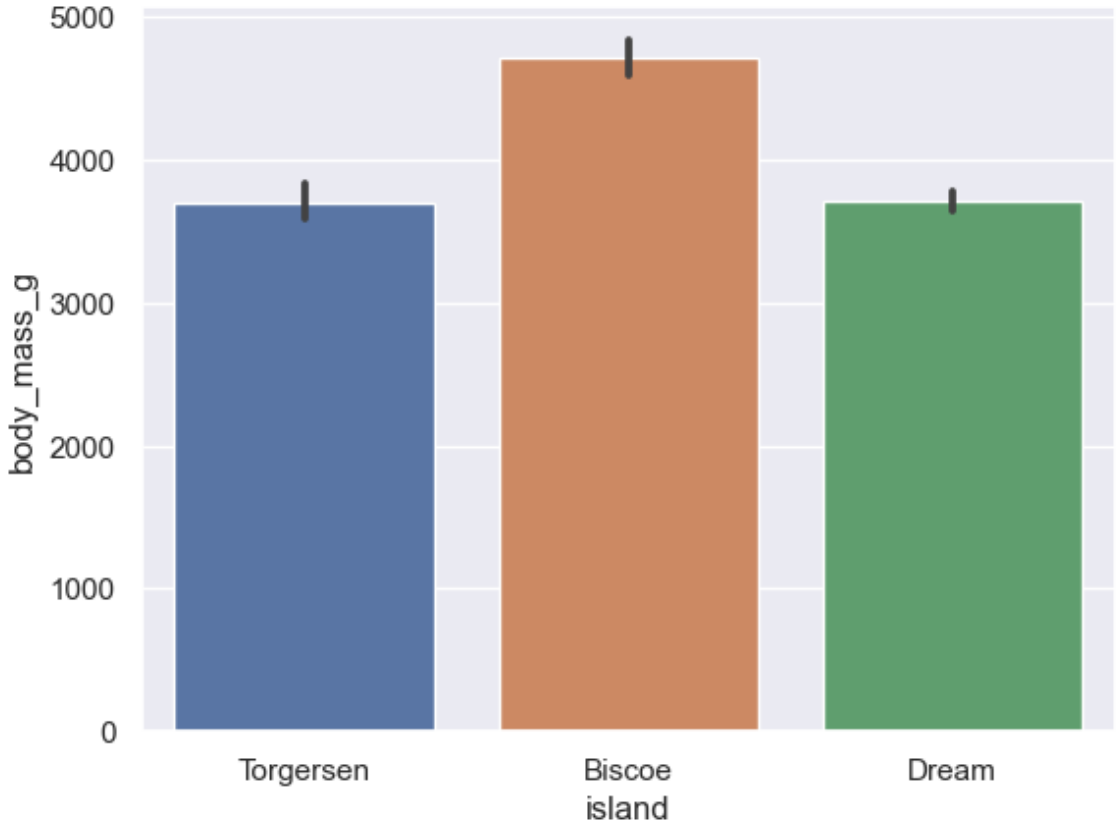
```
df = sns.load_dataset("penguins")
sns.pointplot(data=df, x="island", y="body_mass_g")
plt.show()
```



barplot

In [62]:

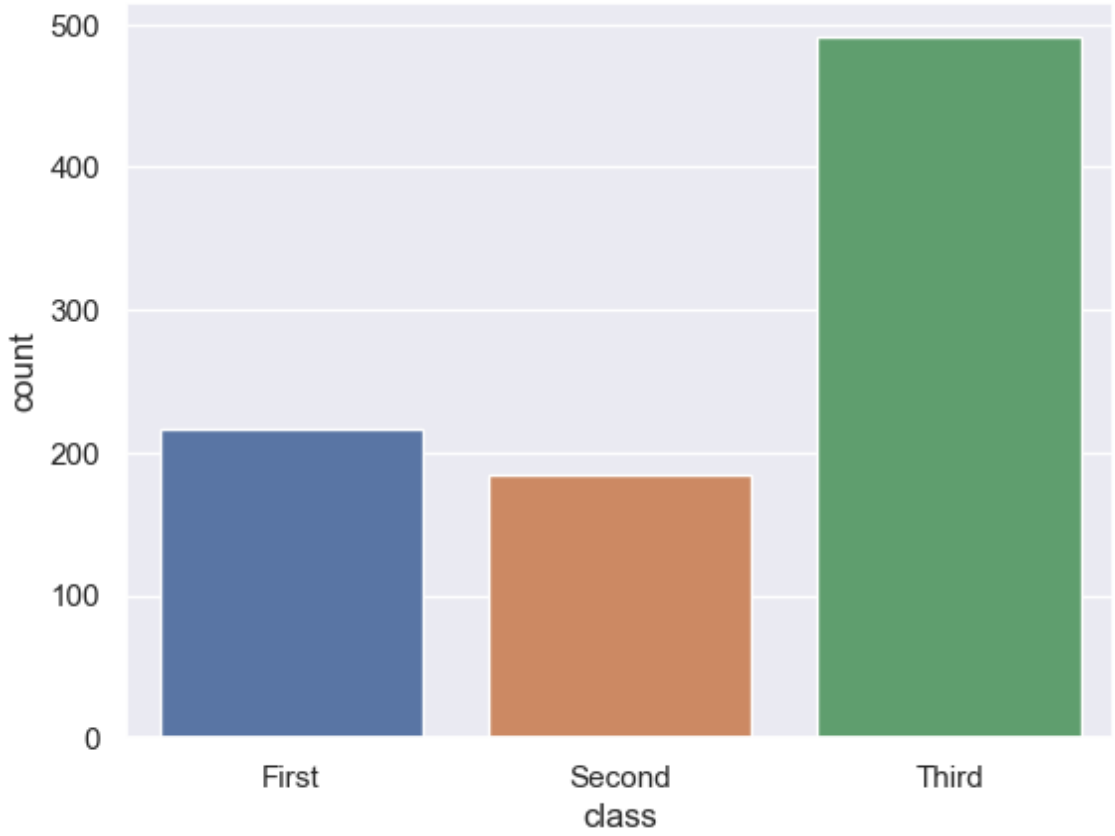
```
df = sns.load_dataset("penguins")
sns.barplot(data=df, x="island", y="body_mass_g")
plt.show()
```



countplot

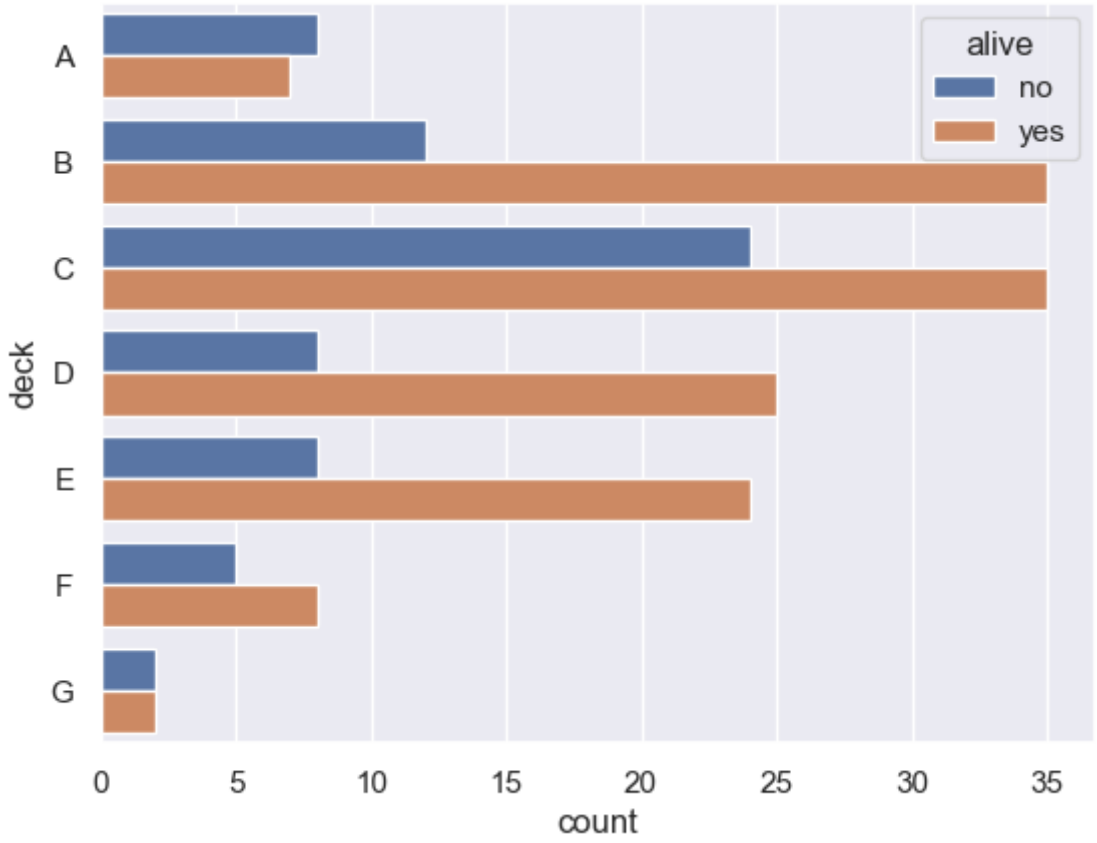
In [63]:

```
df = sns.load_dataset("titanic")
sns.countplot(x=df["class"])
plt.show()
```



In [64]:

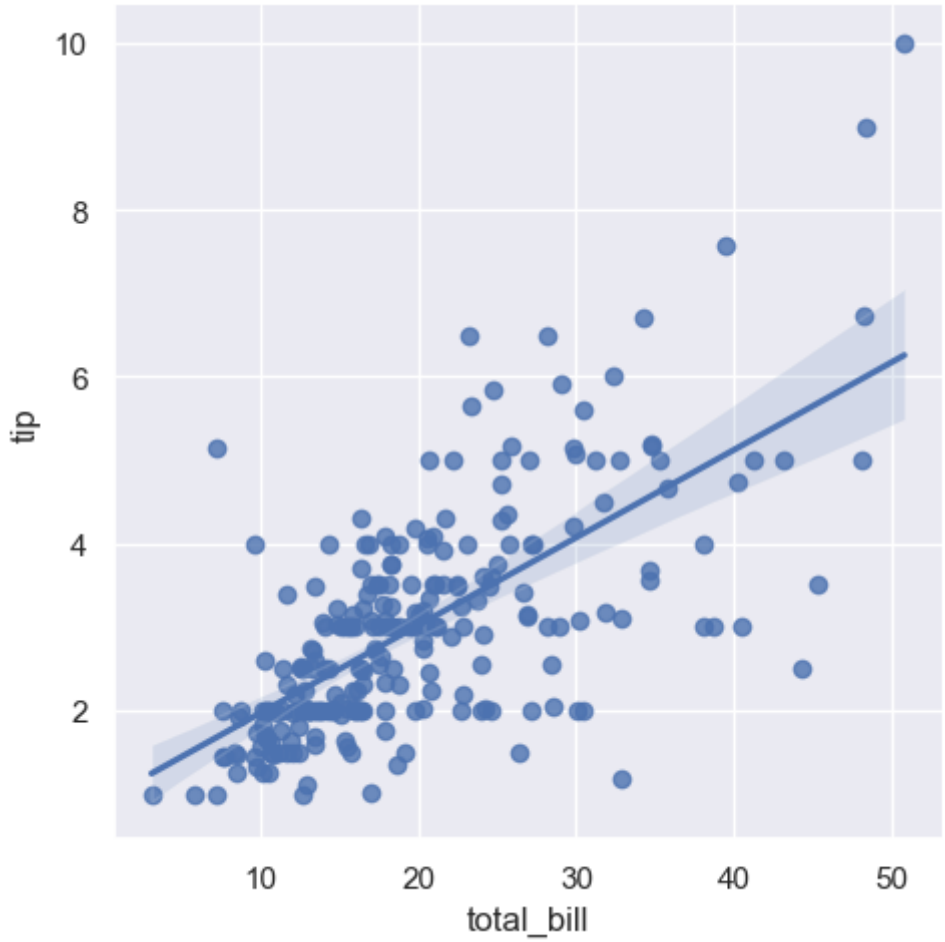
```
sns.countplot(data=df, y="deck", hue="alive")
plt.show()
```



lmplot

In [65]:

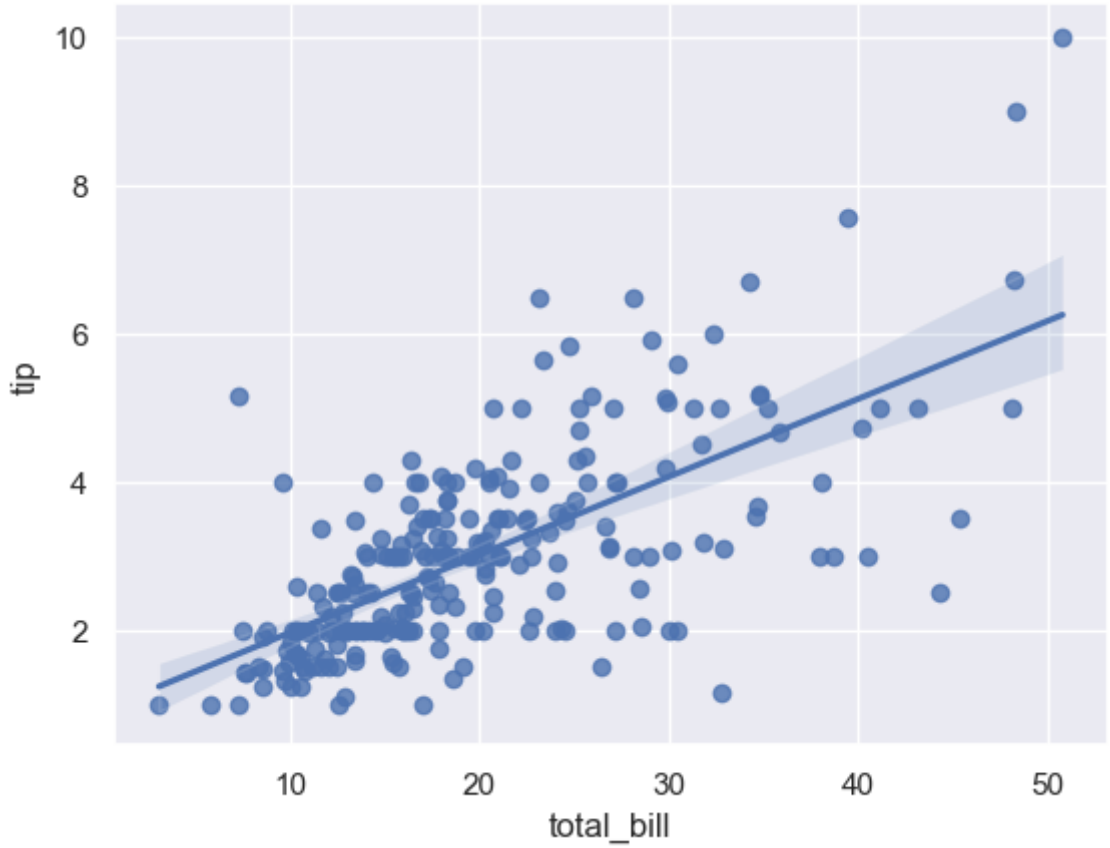
```
tips = sns.load_dataset("tips")
g = sns.lmplot(x="total_bill", y="tip", data=tips)
plt.show()
```



regplot

In [66]:

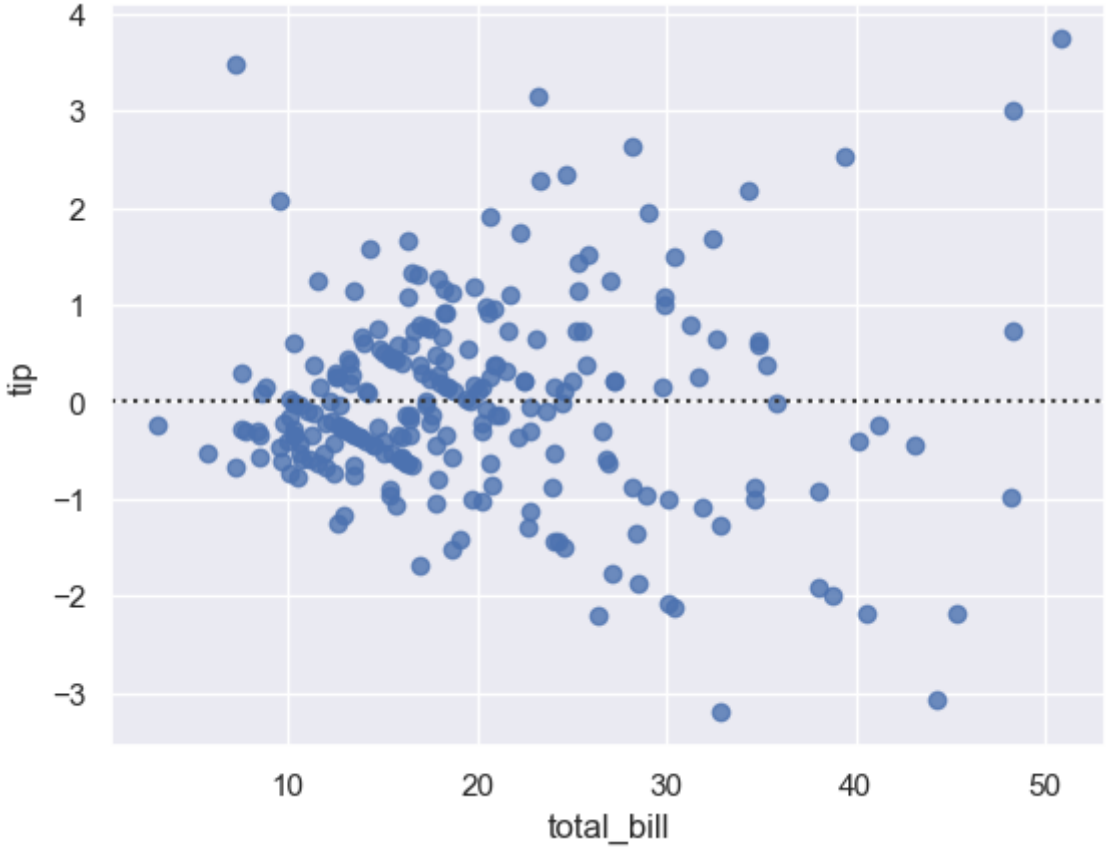
```
import seaborn as sns; sns.set_theme(color_codes=True)
tips = sns.load_dataset("tips")
ax = sns.regplot(x="total_bill", y="tip", data=tips)
plt.show()
```



residplot

In [67]:

```
ax = sns.residplot(x="total_bill", y="tip", data=tips)
plt.show()
```

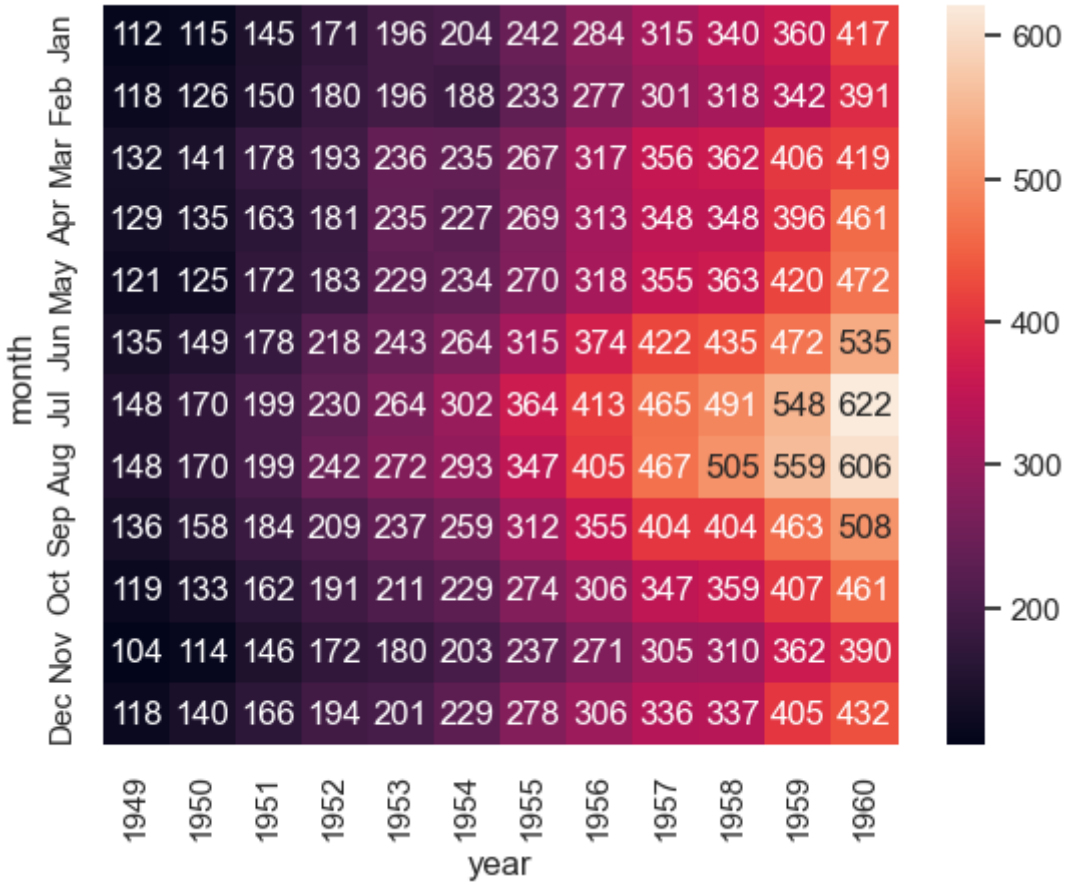


heatmap

In [68]:

```
flights = sns.load_dataset("flights")
flights = flights.pivot("month", "year", "passengers")
sns.heatmap(flights, annot=True, fmt="d")
plt.show()
```

```
C:\Users\divesh\AppData\Local\Temp\ipykernel_25108\2071793689.py:2: Future
Warning: In a future version of pandas all arguments of DataFrame.pivot wi
ll be keyword-only.
  flights = flights.pivot("month", "year", "passengers")
```



In [69]:

```
tips = sns.load_dataset('tips')
tips.head()
```

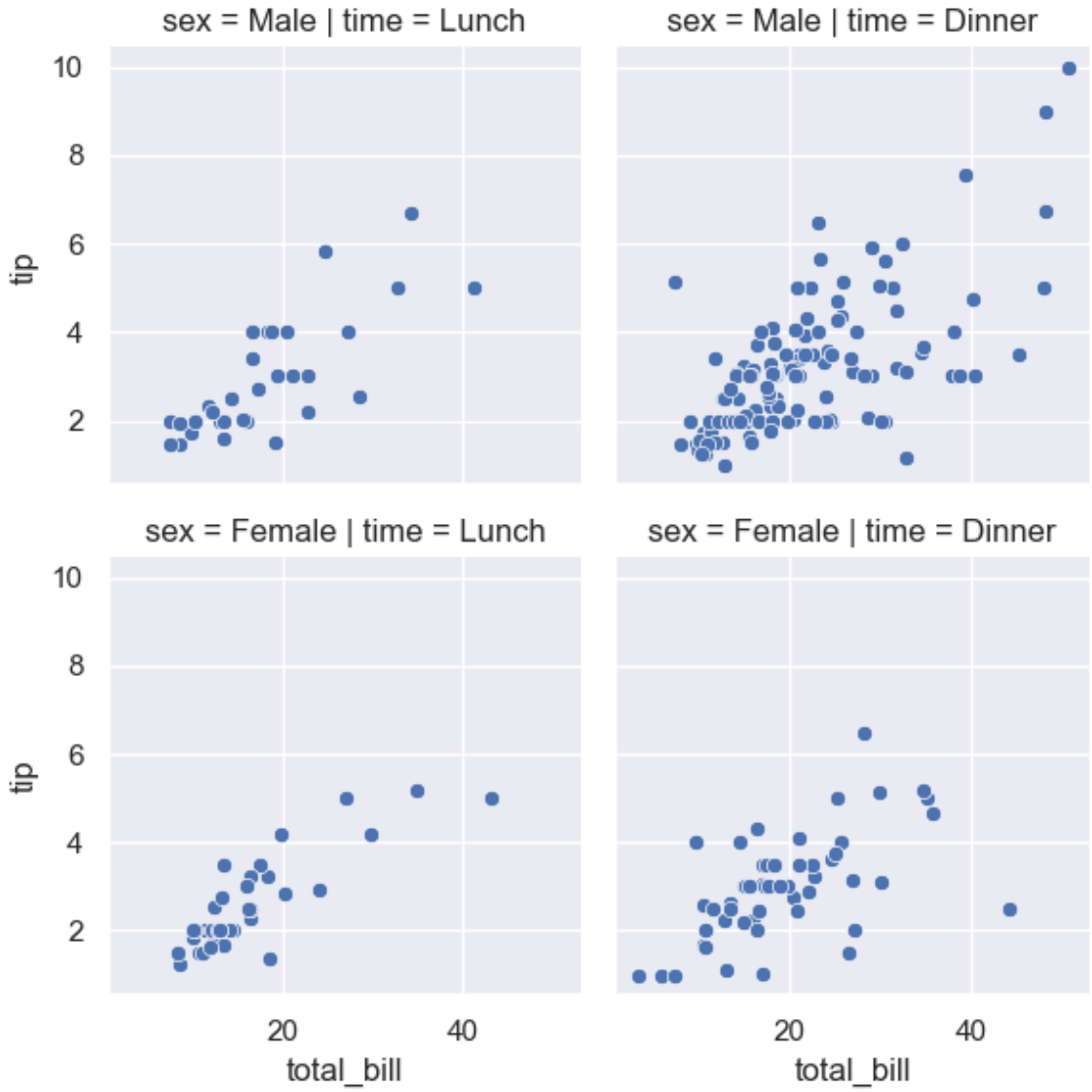
Out[69]:

	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

FacetGrid

In [70]:

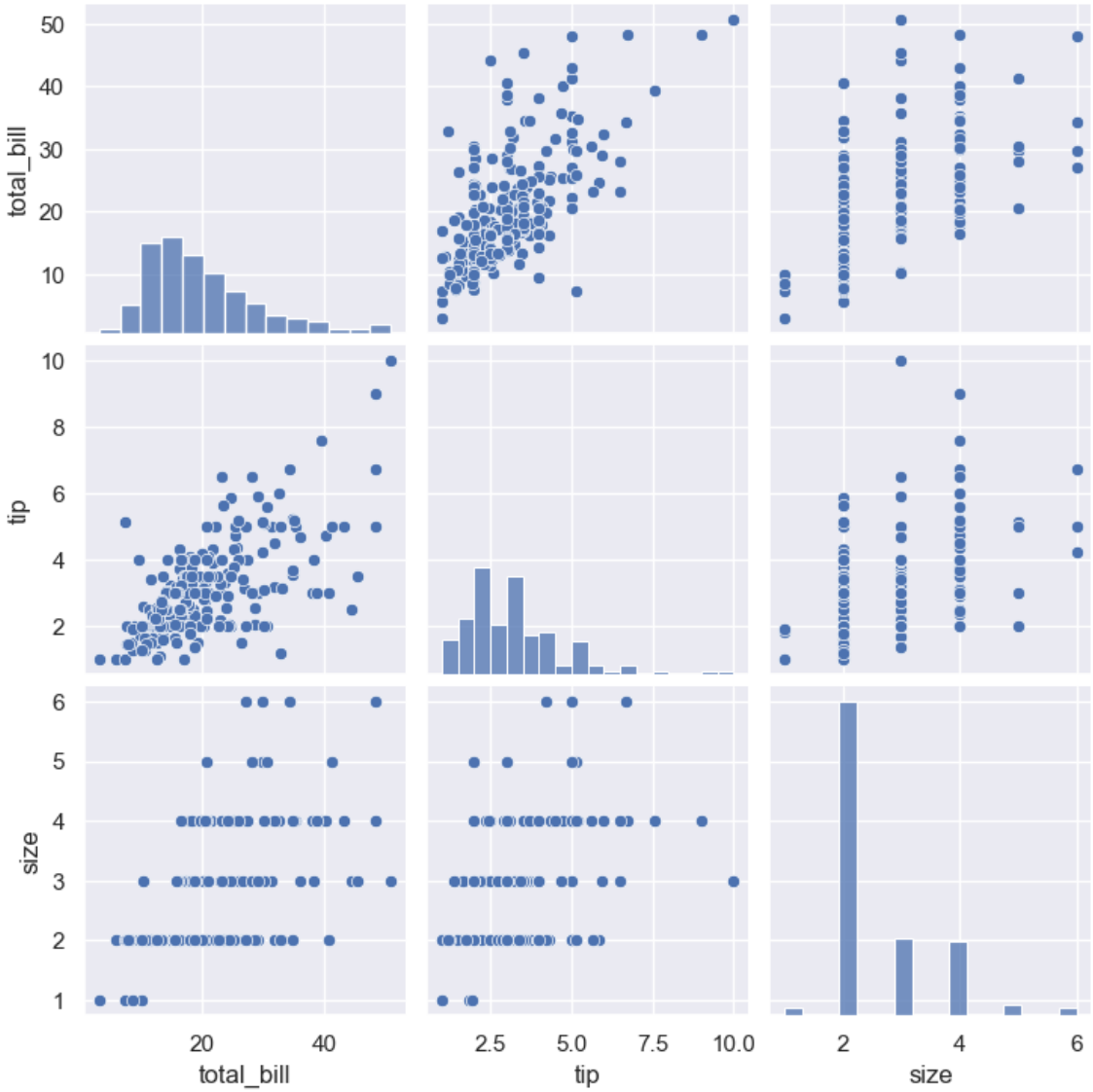
```
g = sns.FacetGrid(tips, col="time",  row="sex")
g.map(sns.scatterplot, "total_bill", "tip")
plt.show()
```



pairplot

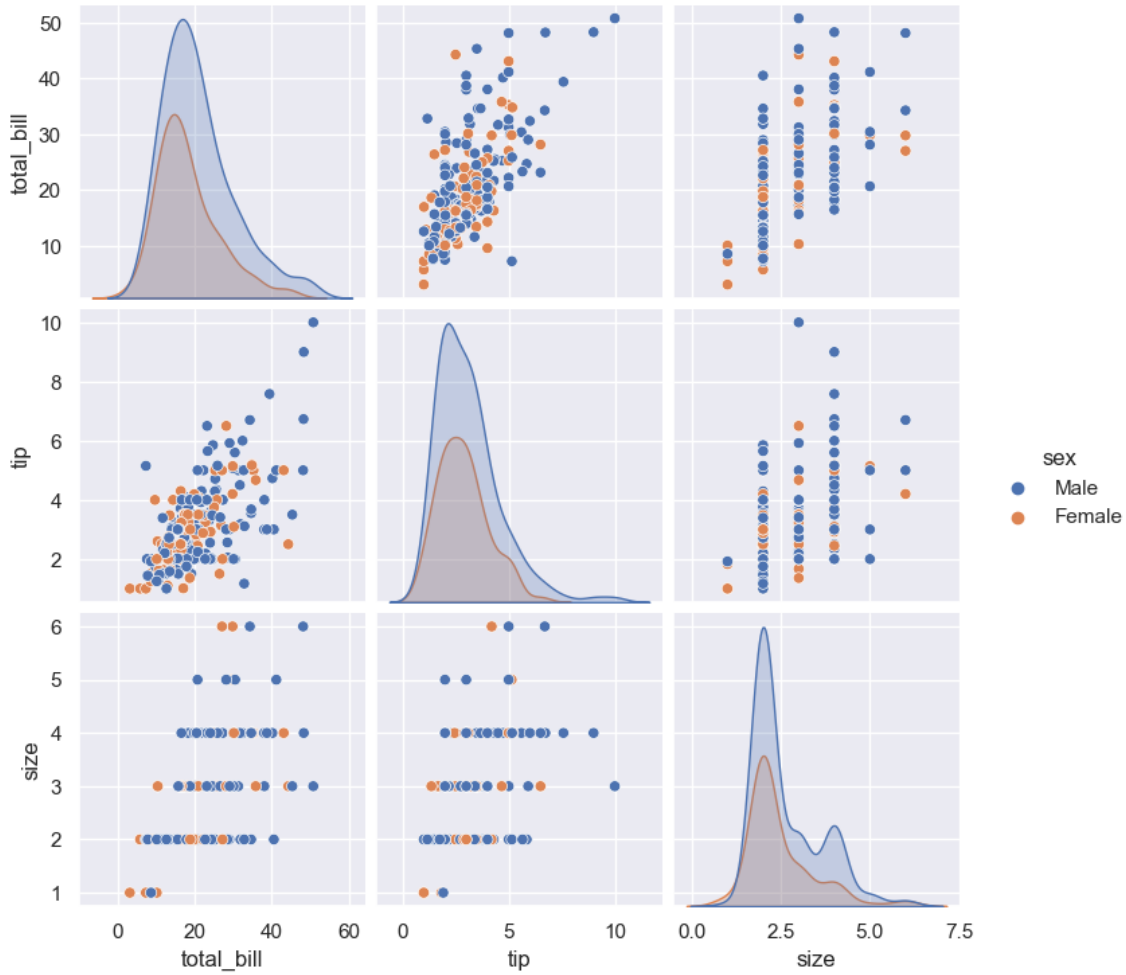
In [71]:

```
sns.pairplot(tips)
plt.show()
```



In [72]:

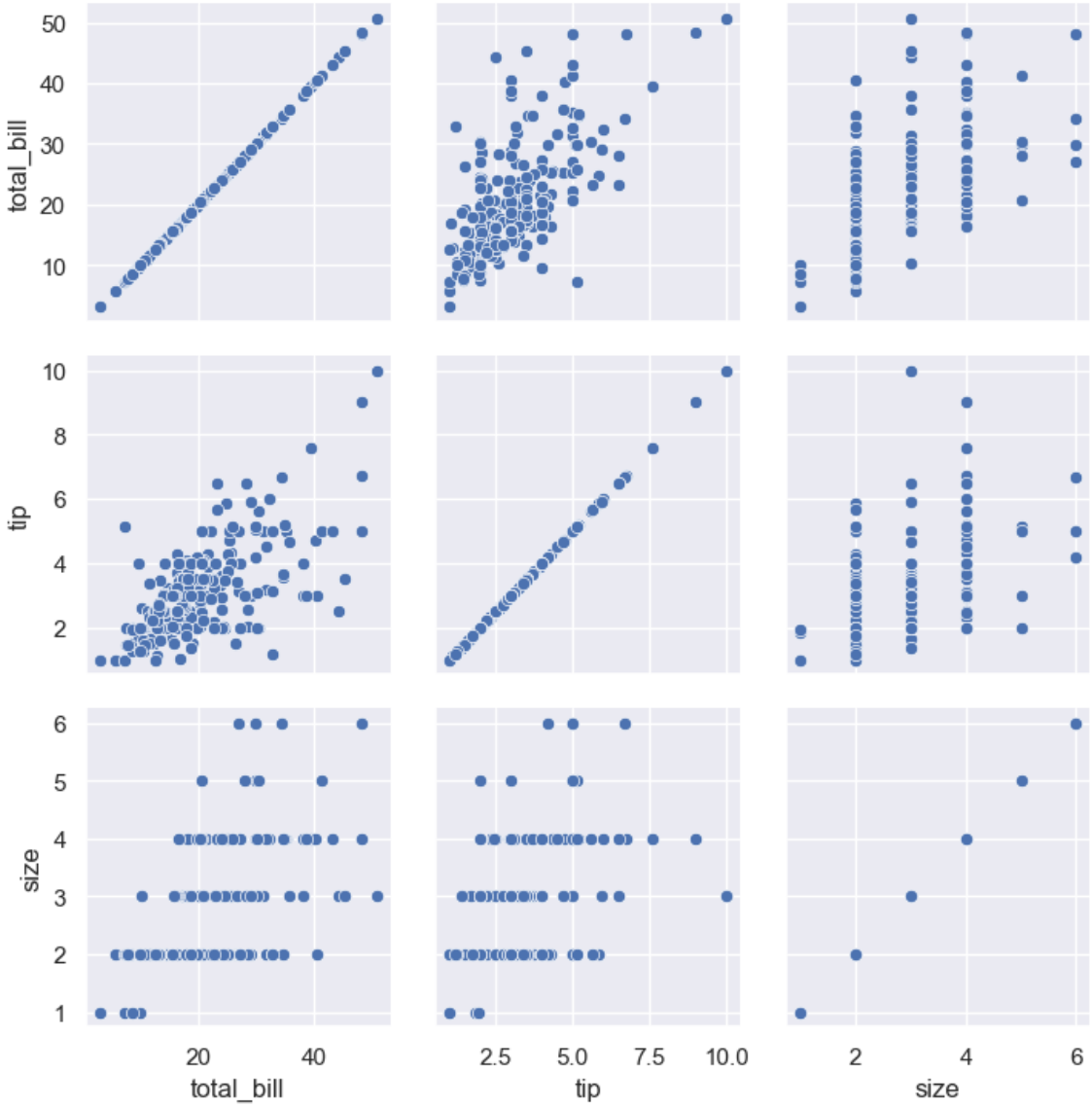
```
sns.pairplot(tips, hue="sex")
plt.show()
```



PairGrid

In [73]:

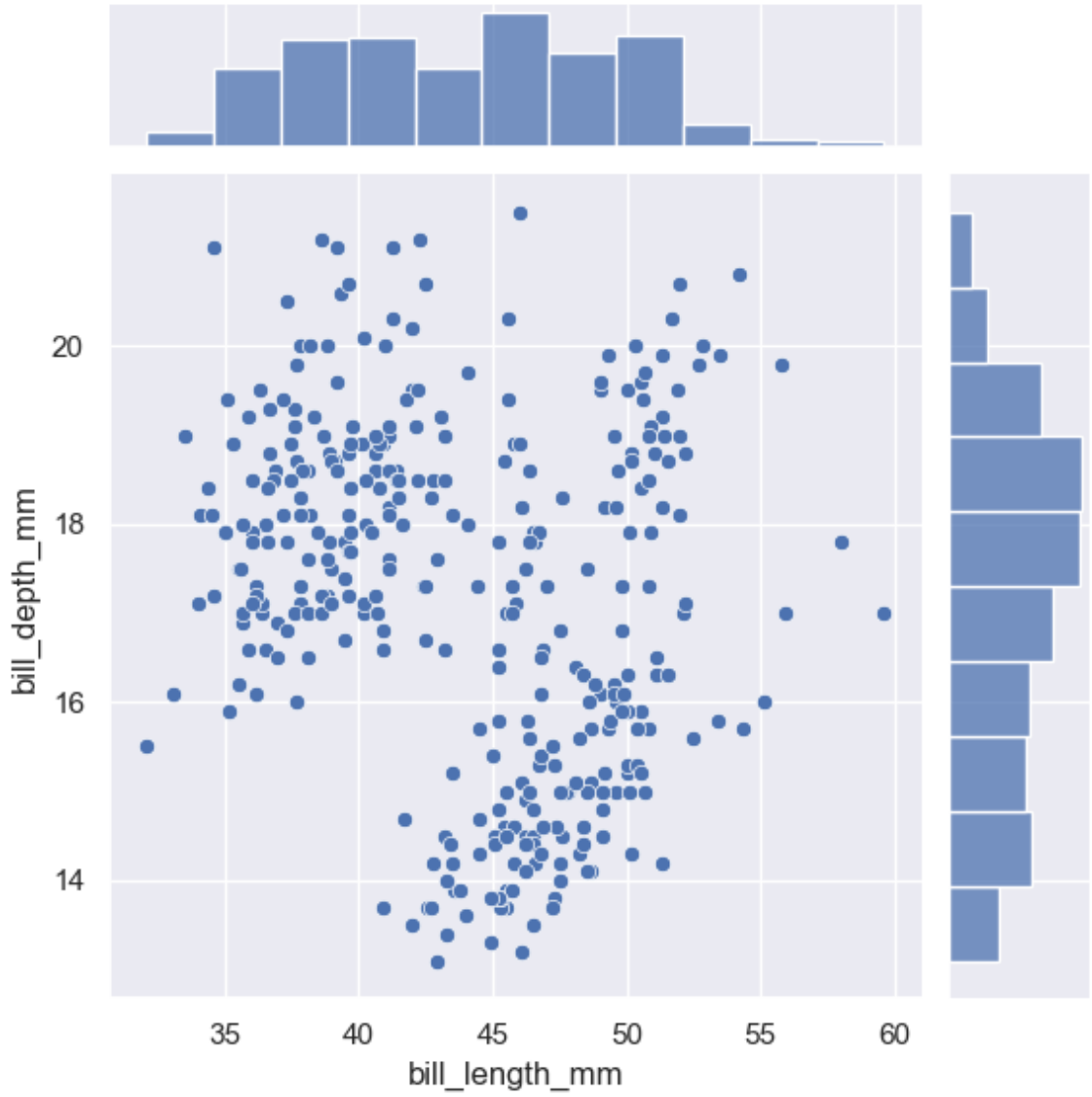
```
g = sns.PairGrid(tips)
g.map(sns.scatterplot)
plt.show()
```



jointplot

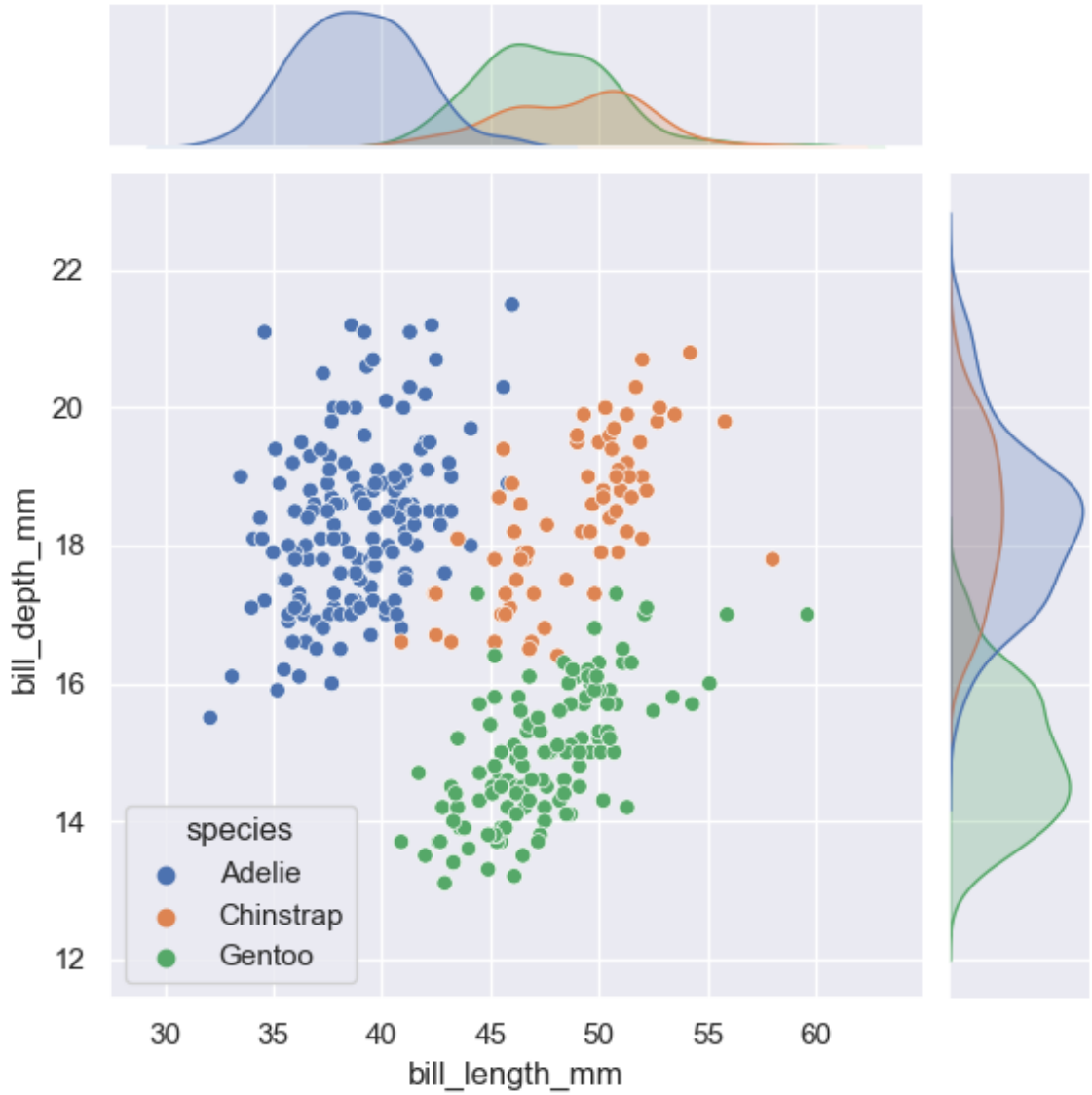
In [74]:

```
penguins = sns.load_dataset("penguins")
sns.jointplot(data=penguins, x="bill_length_mm", y="bill_depth_mm")
plt.show()
```



In [75]:

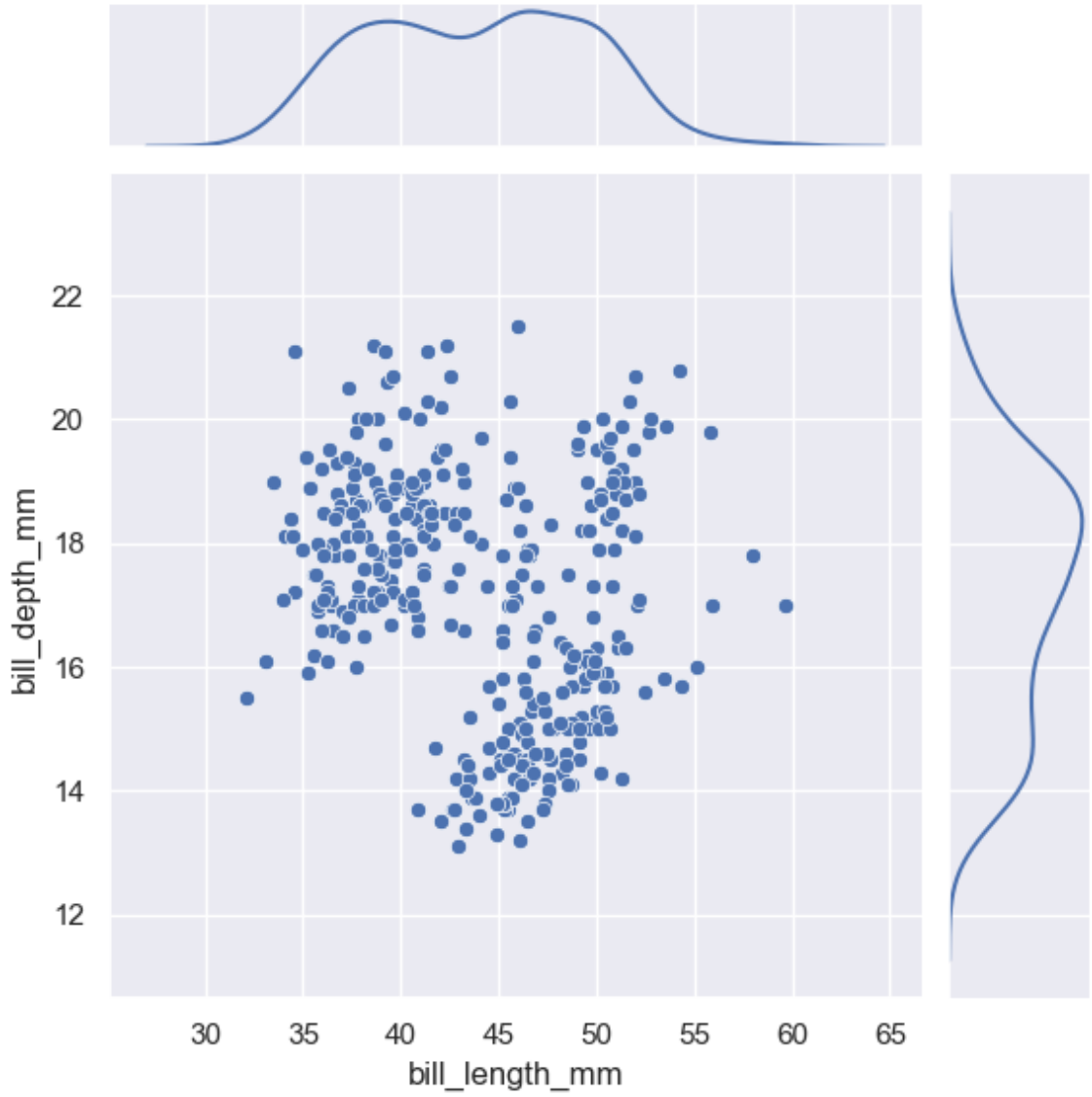
```
sns.jointplot(data=penguins, x="bill_length_mm", y="bill_depth_mm", hue="species")
plt.show()
```



JointGrid

In [76]:

```
g = sns.JointGrid(data=penguins, x="bill_length_mm", y="bill_depth_mm")
g.plot(sns.scatterplot, sns.kdeplot)
plt.show()
```



In [ ]:

```

```