

Lecture 6-3

Visual exploration with Seaborn

Week 6 Friday

Miles Chen, PhD

References:

- https://seaborn.pydata.org/tutorial/function_overview.html
- <https://seaborn.pydata.org/generated/seaborn.displot.html>
- <https://seaborn.pydata.org/api.html>

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

Seaborn is for visual exploration

The primary purpose of seaborn is to make plots and visualize data.

You can use seaborn occasionally to fit a model (e.g. linear model or logistic regression model) to your data. But keep in mind that these are simply for visual exploration. You cannot 'extract' the model (e.g. regression coefficients) from Seaborn

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

```
In [3]: penguins.head(40)
```

```
Out[3]:
```

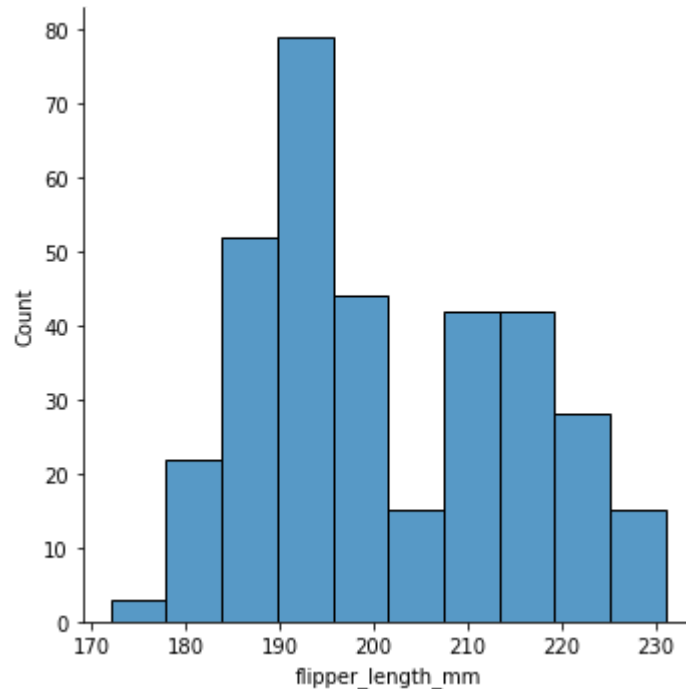
	species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	sex
0	Adelie	Torgersen	39.1	18.7	181.0	3750.0	Male
1	Adelie	Torgersen	39.5	17.4	186.0	3800.0	Female
2	Adelie	Torgersen	40.3	18.0	195.0	3250.0	Female
3	Adelie	Torgersen	NaN	NaN	NaN	NaN	NaN
4	Adelie	Torgersen	36.7	19.3	193.0	3450.0	Female
5	Adelie	Torgersen	39.3	20.6	190.0	3650.0	Male
6	Adelie	Torgersen	38.9	17.8	181.0	3625.0	Female
7	Adelie	Torgersen	39.2	19.6	195.0	4675.0	Male
8	Adelie	Torgersen	34.1	18.1	193.0	3475.0	NaN
9	Adelie	Torgersen	42.0	20.2	190.0	4250.0	NaN
10	Adelie	Torgersen	37.8	17.1	186.0	3300.0	NaN
11	Adelie	Torgersen	37.8	17.3	180.0	3700.0	NaN
12	Adelie	Torgersen	41.1	17.6	182.0	3200.0	Female
13	Adelie	Torgersen	38.6	21.2	191.0	3800.0	Male
14	Adelie	Torgersen	34.6	21.1	198.0	4400.0	Male
15	Adelie	Torgersen	36.6	17.8	185.0	3700.0	Female
16	Adelie	Torgersen	38.7	19.0	195.0	3450.0	Female
17	Adelie	Torgersen	42.5	20.7	197.0	4500.0	Male
18	Adelie	Torgersen	34.4	18.4	184.0	3325.0	Female
19	Adelie	Torgersen	46.0	21.5	194.0	4200.0	Male
20	Adelie	Biscoe	37.8	18.3	174.0	3400.0	Female
21	Adelie	Biscoe	37.7	18.7	180.0	3600.0	Male
22	Adelie	Biscoe	35.9	19.2	189.0	3800.0	Female
23	Adelie	Biscoe	38.2	18.1	185.0	3950.0	Male
24	Adelie	Biscoe	38.8	17.2	180.0	3800.0	Male
25	Adelie	Biscoe	35.3	18.9	187.0	3800.0	Female
26	Adelie	Biscoe	40.6	18.6	183.0	3550.0	Male

	species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	sex
27	Adelie	Biscoe	40.5	17.9	187.0	3200.0	Female
28	Adelie	Biscoe	37.9	18.6	172.0	3150.0	Female
29	Adelie	Biscoe	40.5	18.9	180.0	3950.0	Male
30	Adelie	Dream	39.5	16.7	178.0	3250.0	Female
31	Adelie	Dream	37.2	18.1	178.0	3900.0	Male
32	Adelie	Dream	39.5	17.8	188.0	3300.0	Female
33	Adelie	Dream	40.9	18.9	184.0	3900.0	Male
34	Adelie	Dream	36.4	17.0	195.0	3325.0	Female
35	Adelie	Dream	39.2	21.1	196.0	4150.0	Male
36	Adelie	Dream	38.8	20.0	190.0	3950.0	Male
37	Adelie	Dream	42.2	18.5	180.0	3550.0	Female
38	Adelie	Dream	37.6	19.3	181.0	3300.0	Female
39	Adelie	Dream	39.8	19.1	184.0	4650.0	Male

Univariate exploration

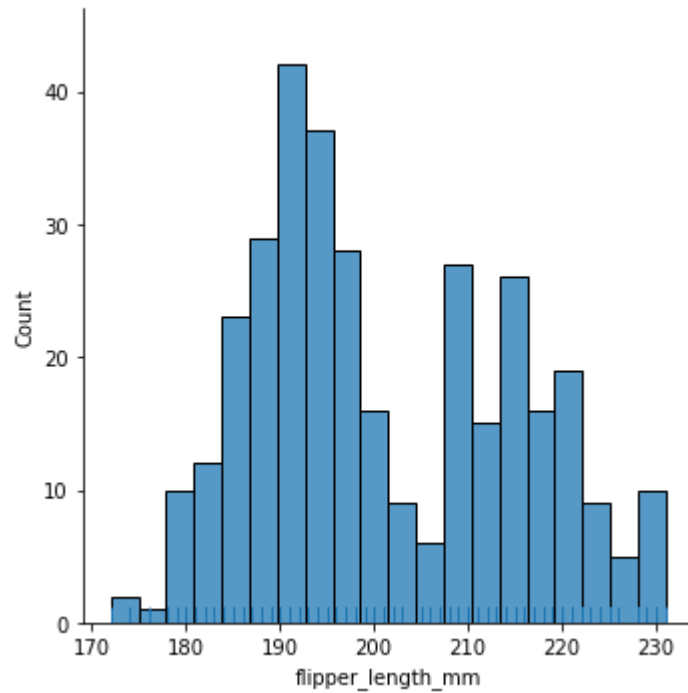
```
In [4]: sns.displot(data = penguins, x = "flipper_length_mm")  
# specify the dataframe and which variable to plot
```

```
Out[4]: <seaborn.axisgrid.FacetGrid at 0x27b392c6c10>
```



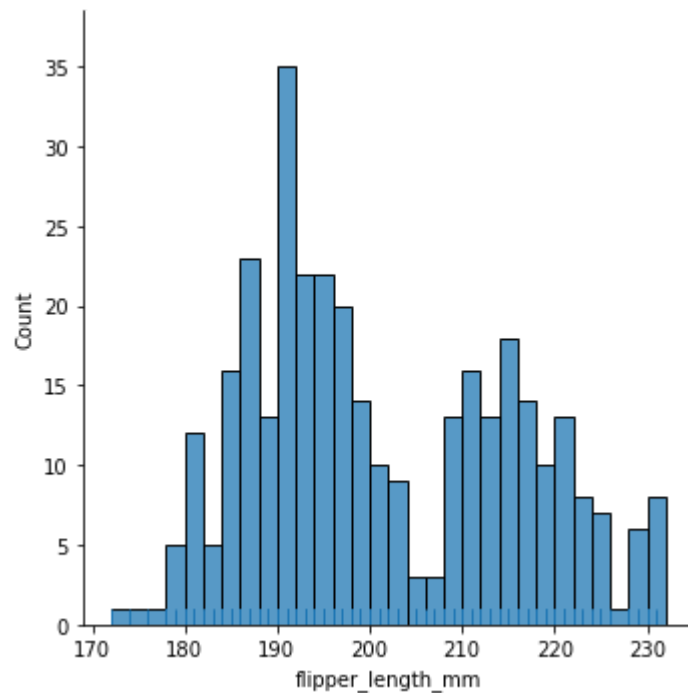
```
In [5]: sns.displot(data = penguins, x = "flipper_length_mm", bins = 20, rug=True)
# use bins to specify bins
# use rug to add a rug plot
```

```
Out[5]: <seaborn.axisgrid.FacetGrid at 0x27b39a7bc40>
```



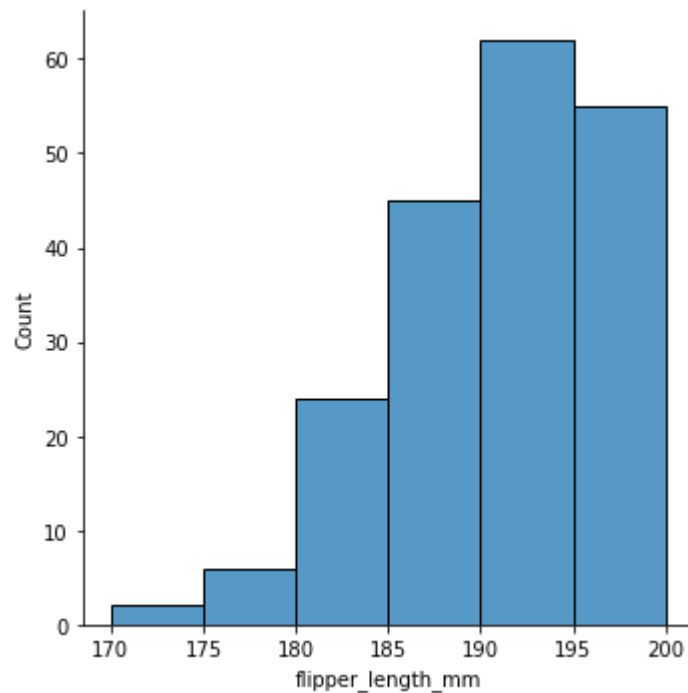
```
In [6]: sns.displot(data = penguins, x = "flipper_length_mm", binwidth = 2, rug=True)
# specify binwidth
```

```
Out[6]: <seaborn.axisgrid.FacetGrid at 0x27b39b2d310>
```



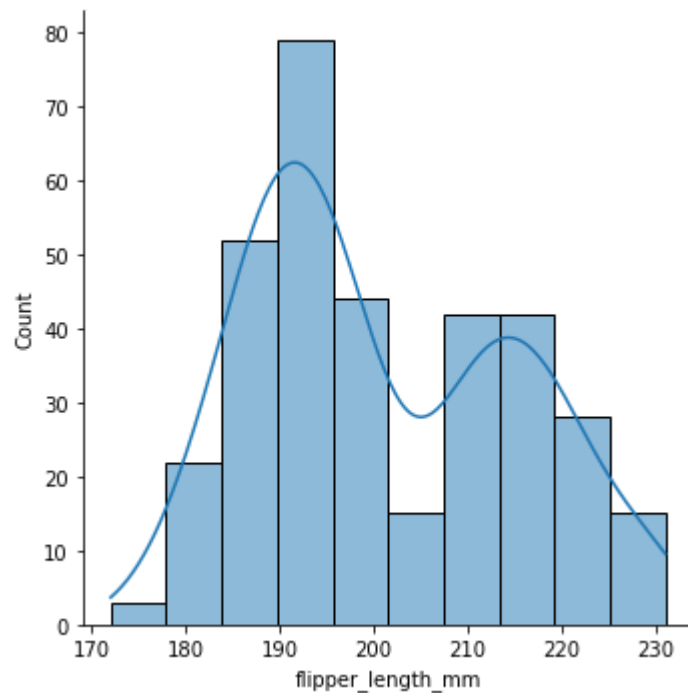
```
In [7]: sns.displot(data = penguins, x = "flipper_length_mm", bins = [170, 175, 180, 185, 190, 195, 200])  
# custom breakpoints
```

```
Out[7]: <seaborn.axisgrid.FacetGrid at 0x27b39b9ae20>
```



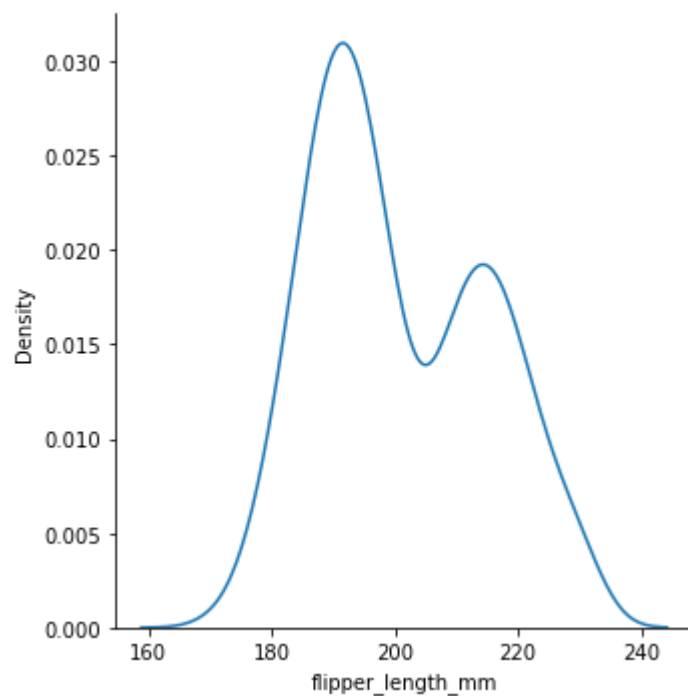

```
In [8]: sns.displot(data = penguins, x = "flipper_length_mm", kde = True)
# you can add a kernel density estimate curve
```

Out[8]: <seaborn.axisgrid.FacetGrid at 0x27b39afce20>



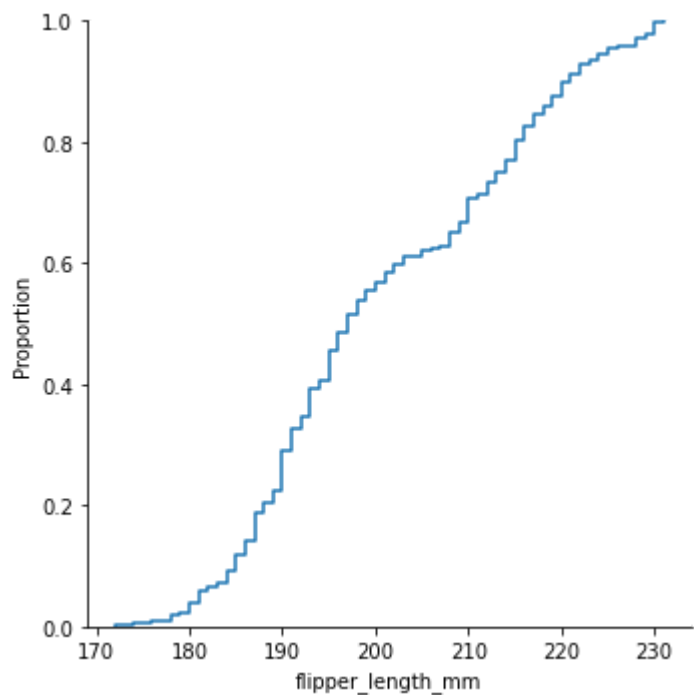
```
In [9]: sns.displot(data = penguins, x = "flipper_length_mm", kind = "kde")
```

```
Out[9]: <seaborn.axisgrid.FacetGrid at 0x27b39bc0970>
```



```
In [10]: sns.displot(data = penguins, x = "flipper_length_mm", kind = "ecdf")
```

```
Out[10]: <seaborn.axisgrid.FacetGrid at 0x27b39d608e0>
```



bivariate and multivariate plots

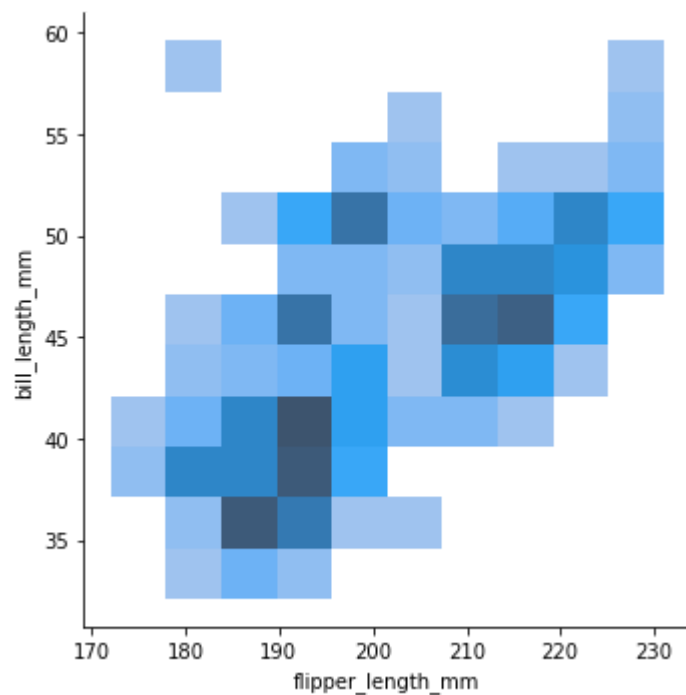
```
In [11]: penguins.head(20)
```

Out[11]:

	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
5	Adelie	Torgersen	39.3	20.6	190.0	3650.0	Male
6	Adelie	Torgersen	38.9	17.8	181.0	3625.0	Female
7	Adelie	Torgersen	39.2	19.6	195.0	4675.0	Male
8	Adelie	Torgersen	34.1	18.1	193.0	3475.0	NaN
9	Adelie	Torgersen	42.0	20.2	190.0	4250.0	NaN
10	Adelie	Torgersen	37.8	17.1	186.0	3300.0	NaN
11	Adelie	Torgersen	37.8	17.3	180.0	3700.0	NaN
12	Adelie	Torgersen	41.1	17.6	182.0	3200.0	Female
13	Adelie	Torgersen	38.6	21.2	191.0	3800.0	Male
14	Adelie	Torgersen	34.6	21.1	198.0	4400.0	Male
15	Adelie	Torgersen	36.6	17.8	185.0	3700.0	Female
16	Adelie	Torgersen	38.7	19.0	195.0	3450.0	Female
17	Adelie	Torgersen	42.5	20.7	197.0	4500.0	Male
18	Adelie	Torgersen	34.4	18.4	184.0	3325.0	Female
19	Adelie	Torgersen	46.0	21.5	194.0	4200.0	Male

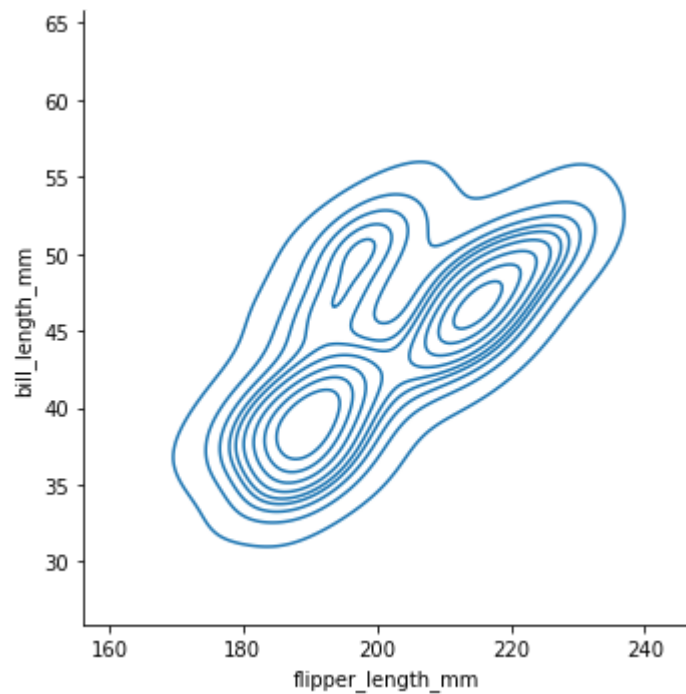
```
In [12]: sns.displot(data=penguins, x="flipper_length_mm", y="bill_length_mm")
```

```
Out[12]: <seaborn.axisgrid.FacetGrid at 0x27b39d0b0a0>
```



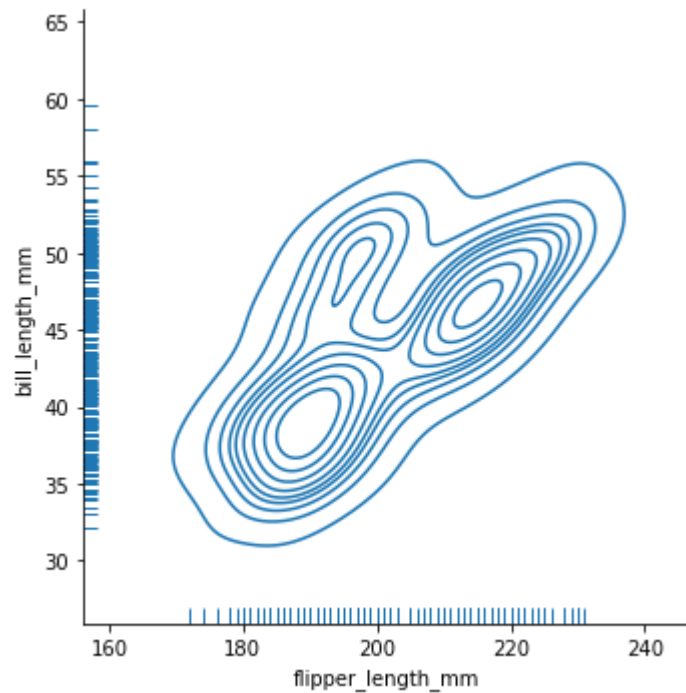
```
In [13]: sns.displot(data=penguins, x="flipper_length_mm", y="bill_length_mm", kind="kde")
```

```
Out[13]: <seaborn.axisgrid.FacetGrid at 0x27b39afc1c0>
```



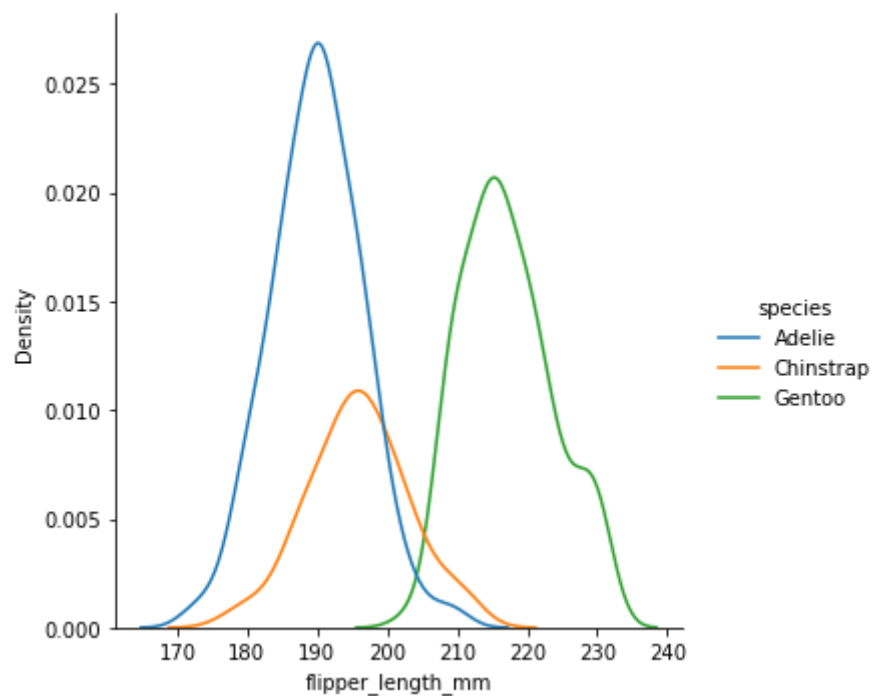
```
In [14]: sns.displot(data=penguins, x="flipper_length_mm", y="bill_length_mm", kind="kde", rug=True)
```

```
Out[14]: <seaborn.axisgrid.FacetGrid at 0x27b3af35a60>
```



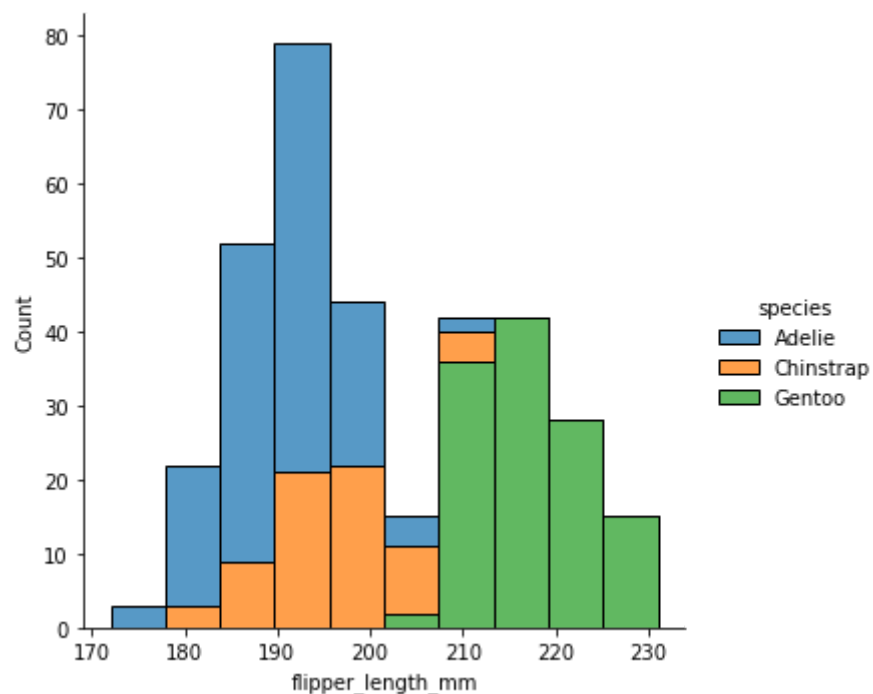
```
In [15]: sns.displot(data=penguins, x="flipper_length_mm", hue="species", kind="kde")
```

```
Out[15]: <seaborn.axisgrid.FacetGrid at 0x27b3aef7e80>
```



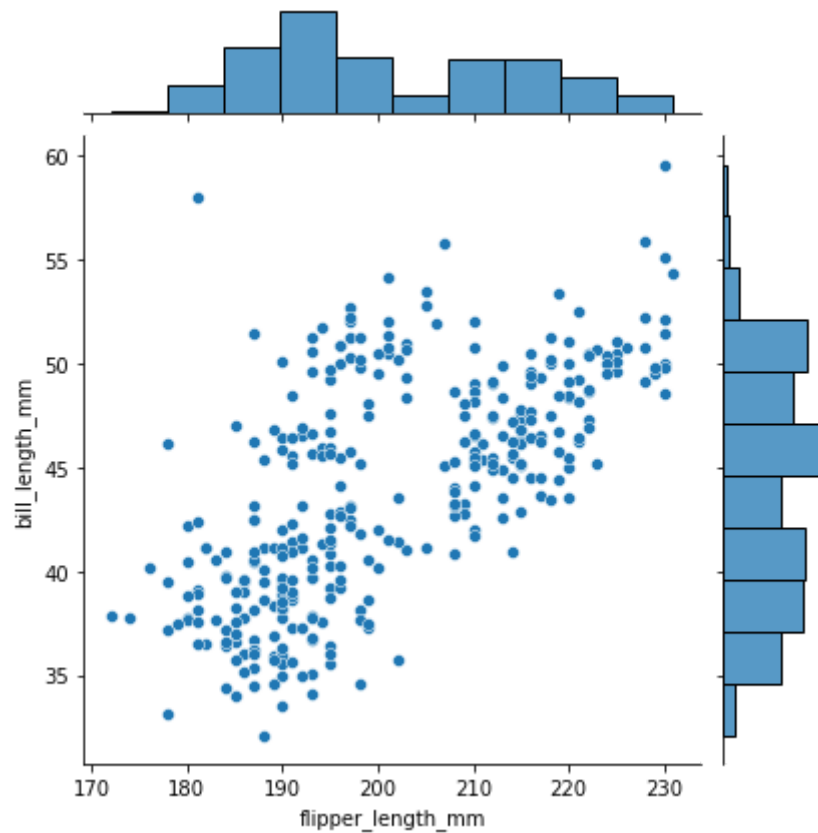

```
In [16]: sns.displot(data=penguins, x="flipper_length_mm", hue="species", multiple="stack")
```

```
Out[16]: <seaborn.axisgrid.FacetGrid at 0x27b39ed56d0>
```



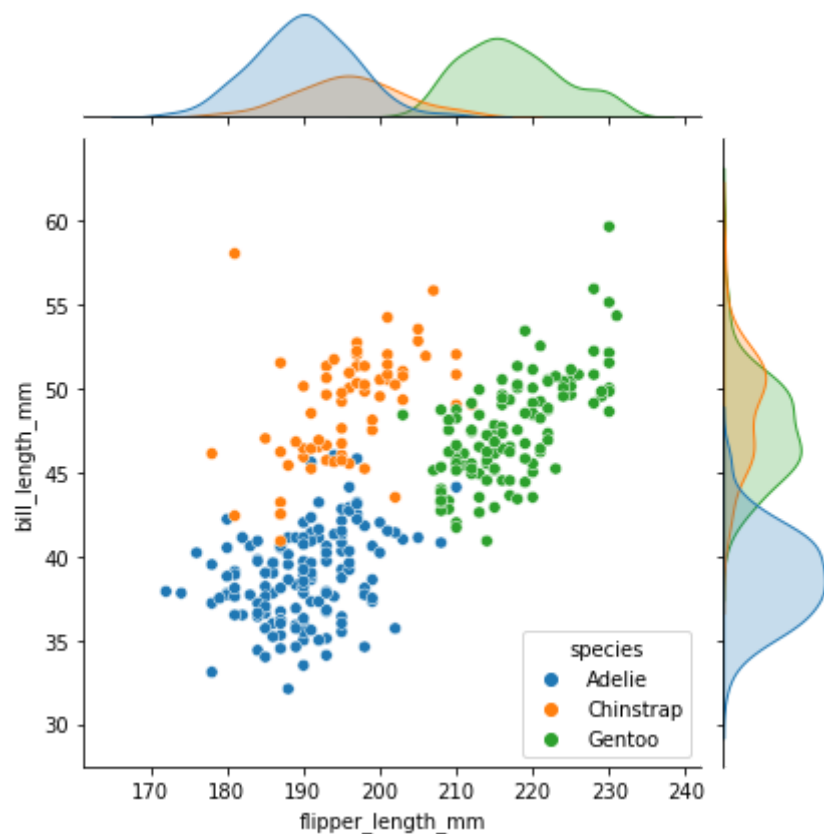
```
In [17]: sns.jointplot(data=penguins, x="flipper_length_mm", y="bill_length_mm")
```

```
Out[17]: <seaborn.axisgrid.JointGrid at 0x27b3b069b20>
```



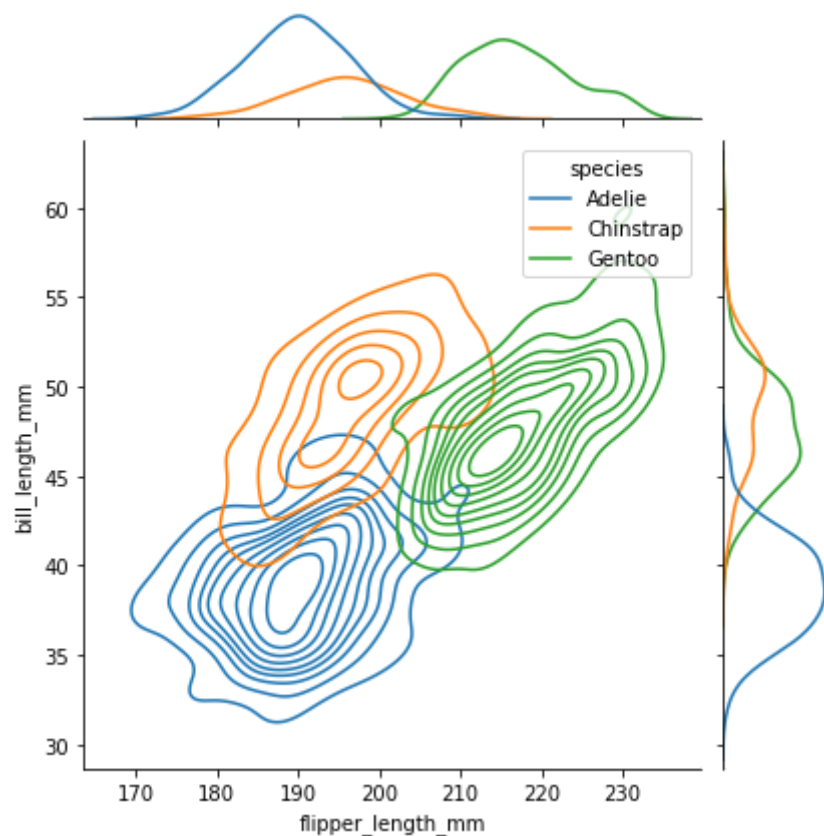
```
In [18]: sns.jointplot(data=penguins, x="flipper_length_mm", y="bill_length_mm", hue = "species")
```

```
Out[18]: <seaborn.axisgrid.JointGrid at 0x27b3b177e20>
```



```
In [19]: sns.jointplot(data=penguins, x="flipper_length_mm", y="bill_length_mm", hue = "species", kind = "
```

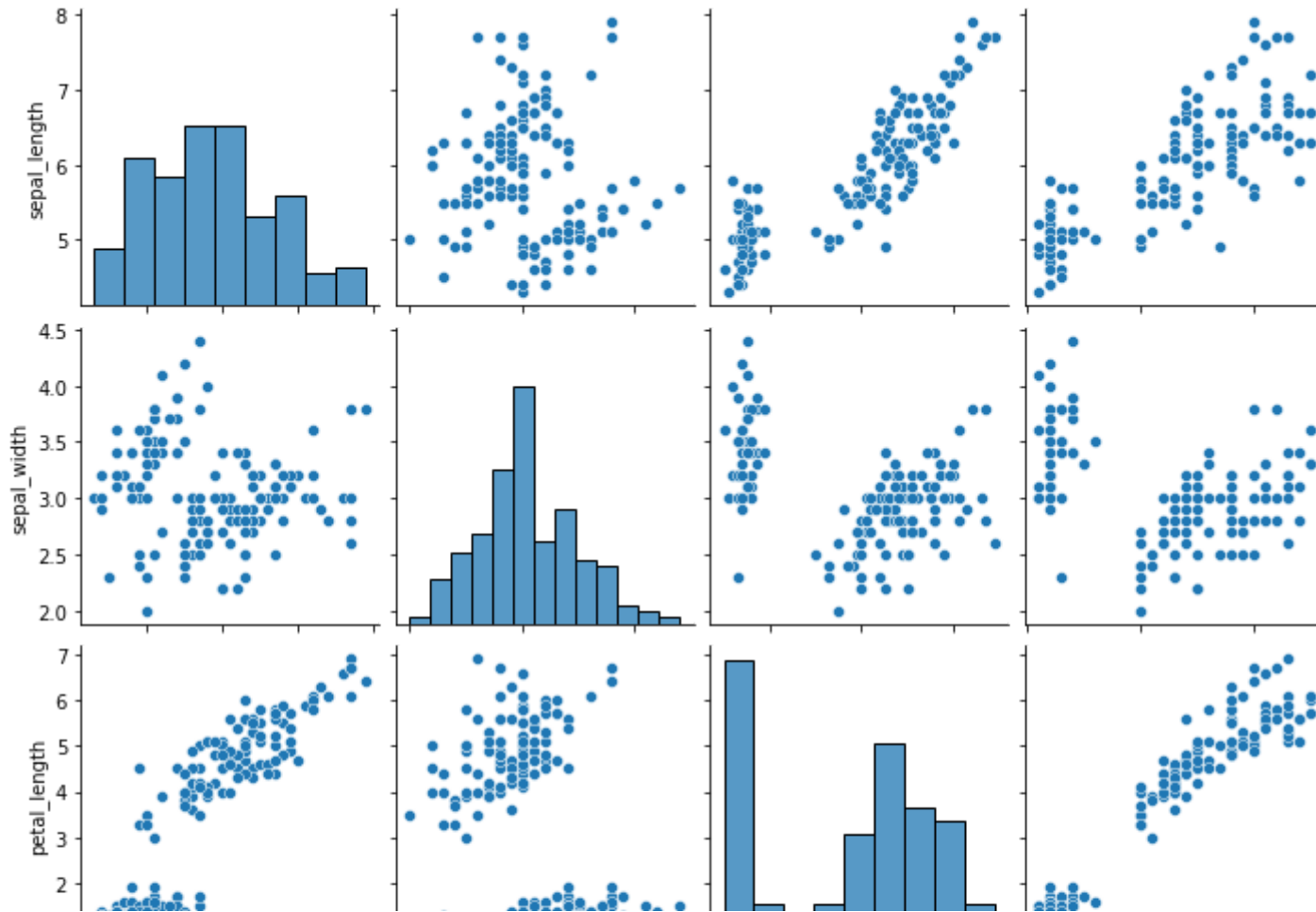
```
Out[19]: <seaborn.axisgrid.JointGrid at 0x27b3b3bad30>
```

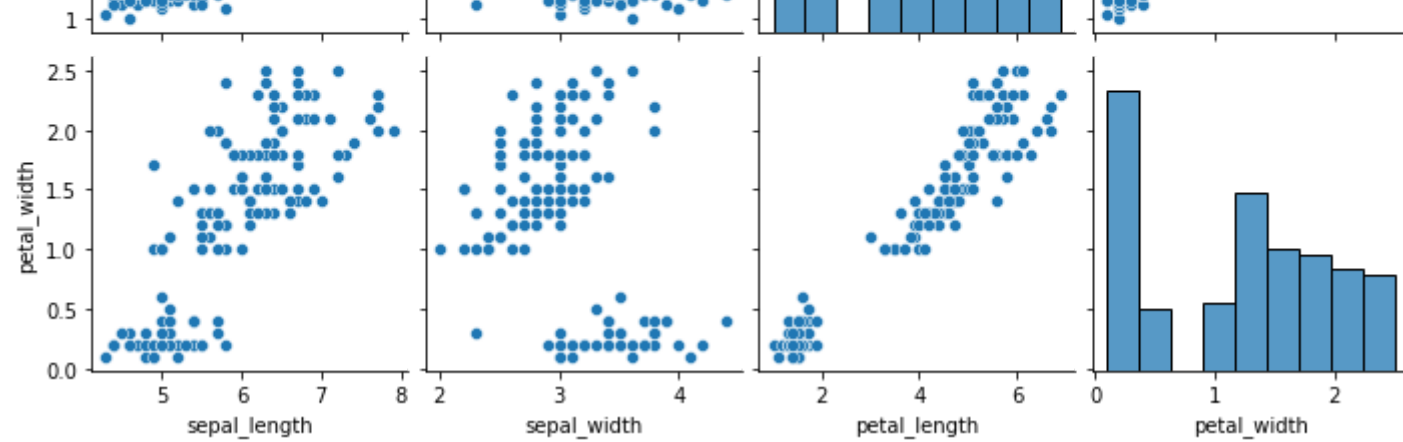


Pair-wise plots for multiple numeric data variables

```
In [20]: # good ol' iris data  
iris = sns.load_dataset("iris")  
sns.pairplot(iris)
```

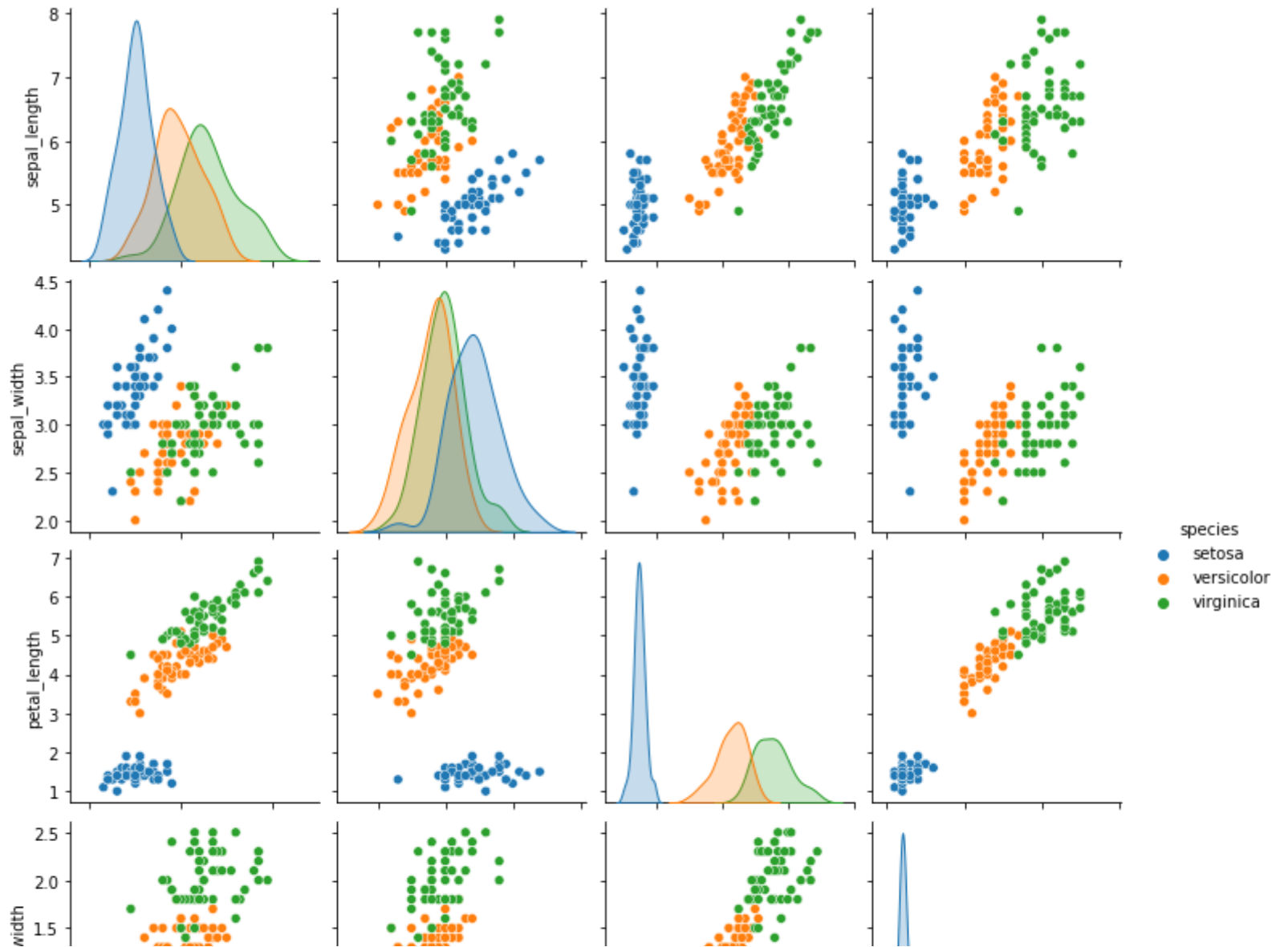
```
Out[20]: <seaborn.axisgrid.PairGrid at 0x27b3c50b7c0>
```

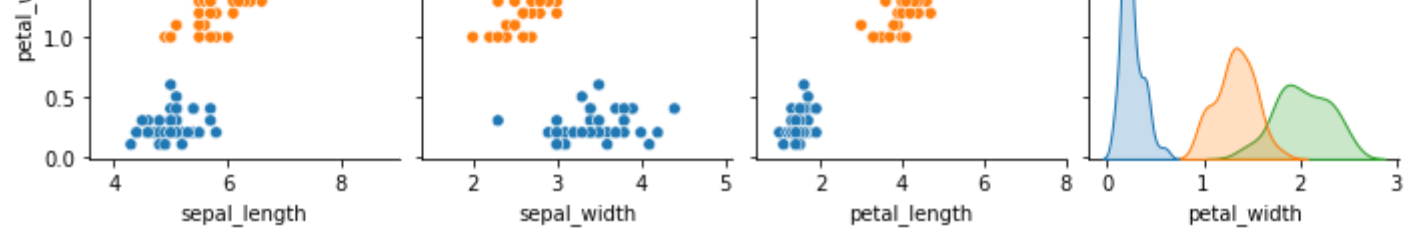




```
In [21]: sns.pairplot(iris, hue="species")
```

```
Out[21]: <seaborn.axisgrid.PairGrid at 0x27b3b4bed90>
```





Univariate plots separated by category

```
In [22]: tips = sns.load_dataset("tips")
tips.head(10)
# tips data, contains numeric vars: total_bill, tip, size
# categorical vars: sex, smoker, day, time
```

```
Out[22]:
```

	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

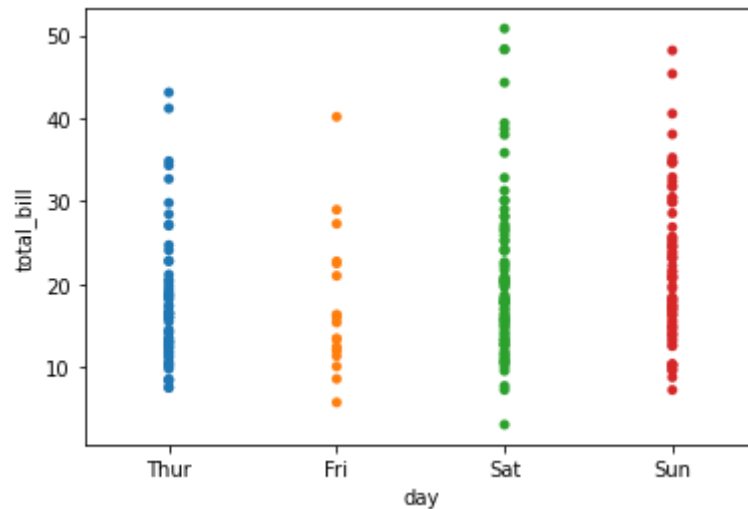
```
In [23]: tips.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 244 entries, 0 to 243
Data columns (total 7 columns):
 #   Column        Non-Null Count  Dtype  
---  -
 0   total_bill    244 non-null    float64
 1   tip           244 non-null    float64
 2   sex           244 non-null    category
 3   smoker        244 non-null    category
```

```
4    day          244 non-null    category
5    time          244 non-null    category
6    size          244 non-null    int64
dtypes: category(4), float64(2), int64(1)
memory usage: 7.4 KB
```

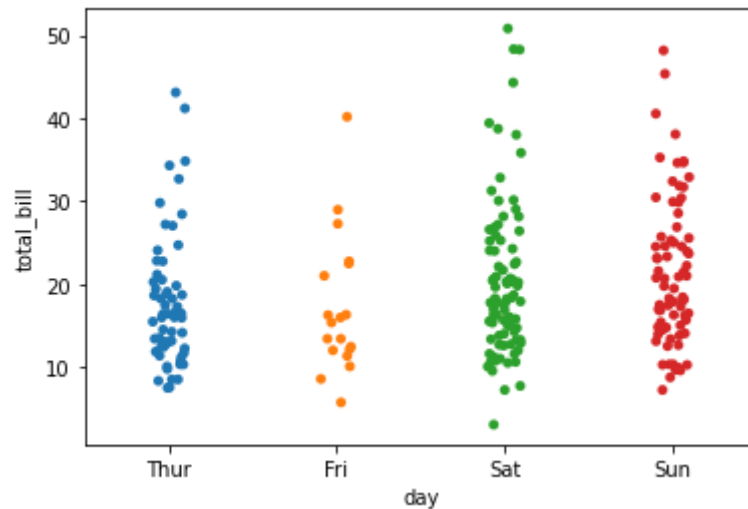
```
sns.stripplot(x="day", y="total_bill", data=tips, jitter = False)
# Like a dotplot, but the dots are plotted on top of each other
```

```
Out[24]: <AxesSubplot:xlabel='day', ylabel='total_bill'>
```



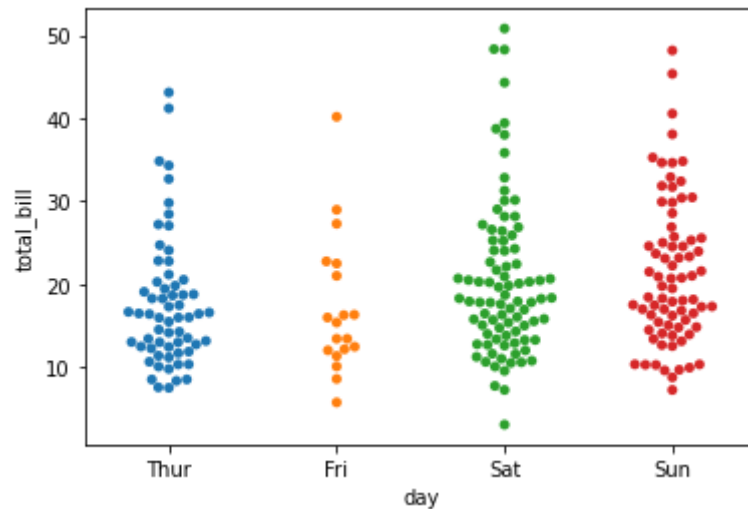
```
In [25]: sns.stripplot(x="day", y="total_bill", data=tips, jitter=True)
# adding jitter allows us to see where points were overlapping
```

```
Out[25]: <AxesSubplot:xlabel='day', ylabel='total_bill'>
```



```
In [26]: sns.swarmplot(x="day", y="total_bill", data=tips)
# swarmplots are like symmetric dotplots
```

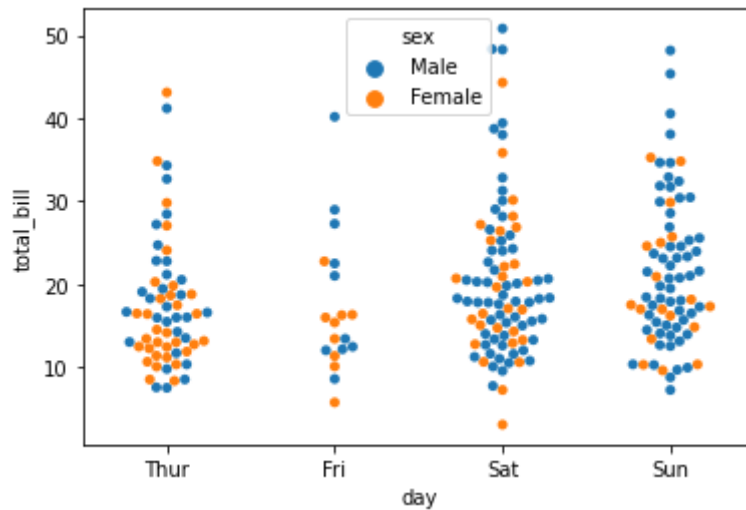
```
Out[26]: <AxesSubplot:xlabel='day', ylabel='total_bill'>
```



In [27]:

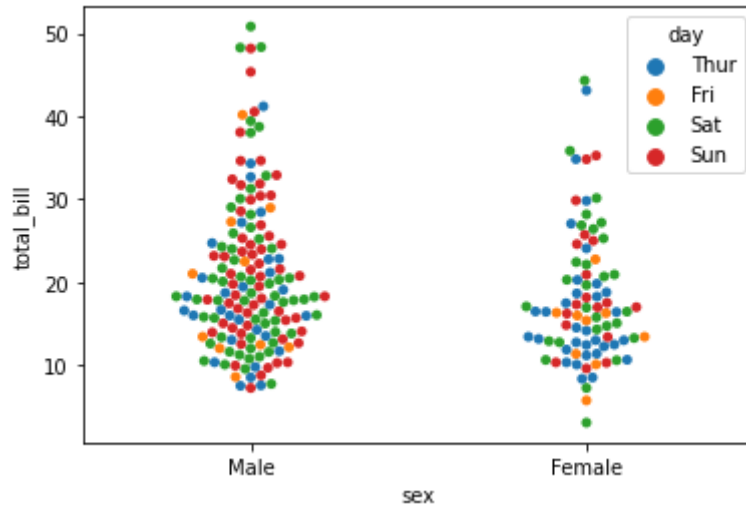
```
sns.swarmplot(x="day", y="total_bill", hue="sex", data=tips)  
# you can change the color of the point based on another categorical variable  
# seaborn automatically adds a legend
```

Out[27]: <AxesSubplot:xlabel='day', ylabel='total_bill'>



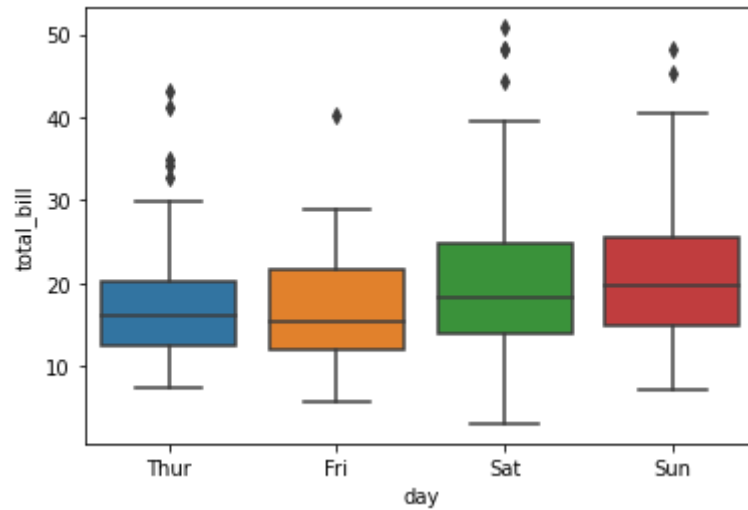
```
In [28]: sns.swarmplot(x="sex", y="total_bill", hue="day", data=tips)
# this plot is harder on the eyes, but contains the same info as above
# the data is separated based on sex and colored based on the day
# but the colors aren't helping me
```

```
Out[28]: <AxesSubplot:xlabel='sex', ylabel='total_bill'>
```



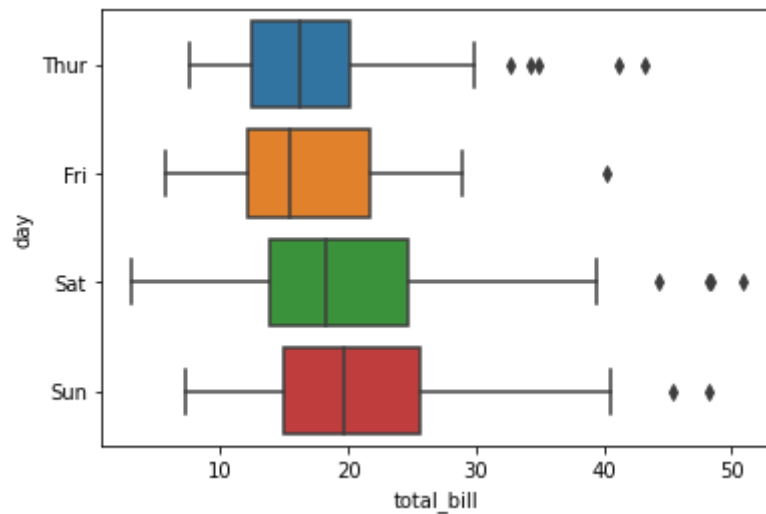
```
In [29]: sns.boxplot(x="day", y="total_bill", data=tips) # boxplots
```

```
Out[29]: <AxesSubplot:xlabel='day', ylabel='total_bill'>
```



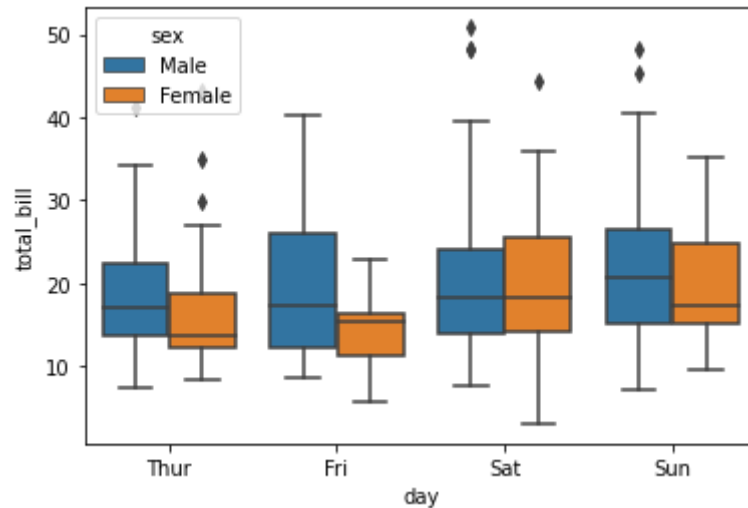

```
In [30]: sns.boxplot(y="day", x="total_bill", data=tips) # boxplots
```

```
Out[30]: <AxesSubplot:xlabel='total_bill', ylabel='day'>
```



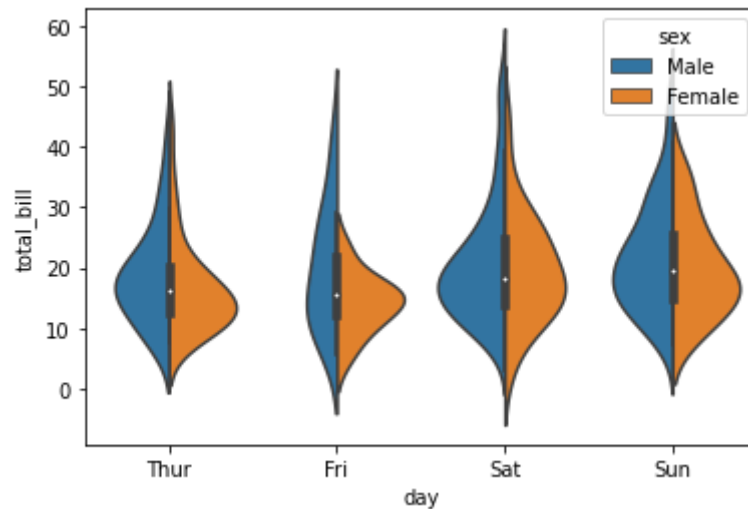
```
In [31]: sns.boxplot(x="day", y="total_bill", hue="sex", data=tips) # boxplots
```

```
Out[31]: <AxesSubplot:xlabel='day', ylabel='total_bill'>
```



```
In [32]: sns.violinplot(x="day", y="total_bill", hue="sex", data=tips, split=True)  
# i'm not a huge fan of these plots
```

```
Out[32]: <AxesSubplot:xlabel='day', ylabel='total_bill'>
```



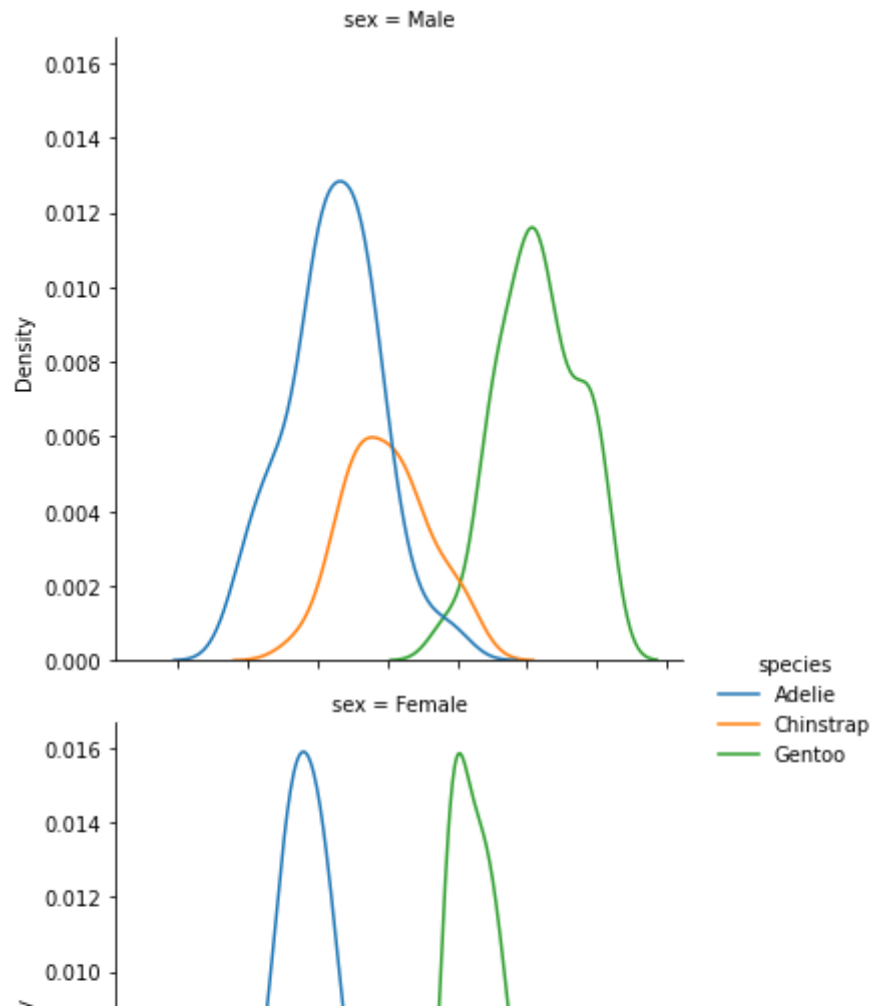
facet grids to make several plots

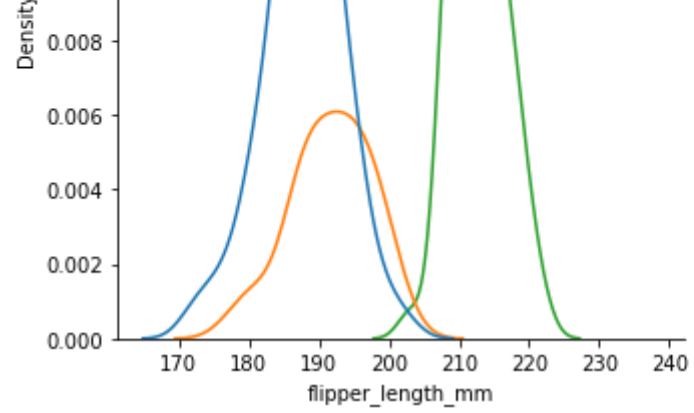
In [33]:

```
# specify the column or row to create a grid based on that variable  
sns.displot(data=penguins, x="flipper_length_mm", hue="species", row="sex", kind="kde")
```

Out[33]:

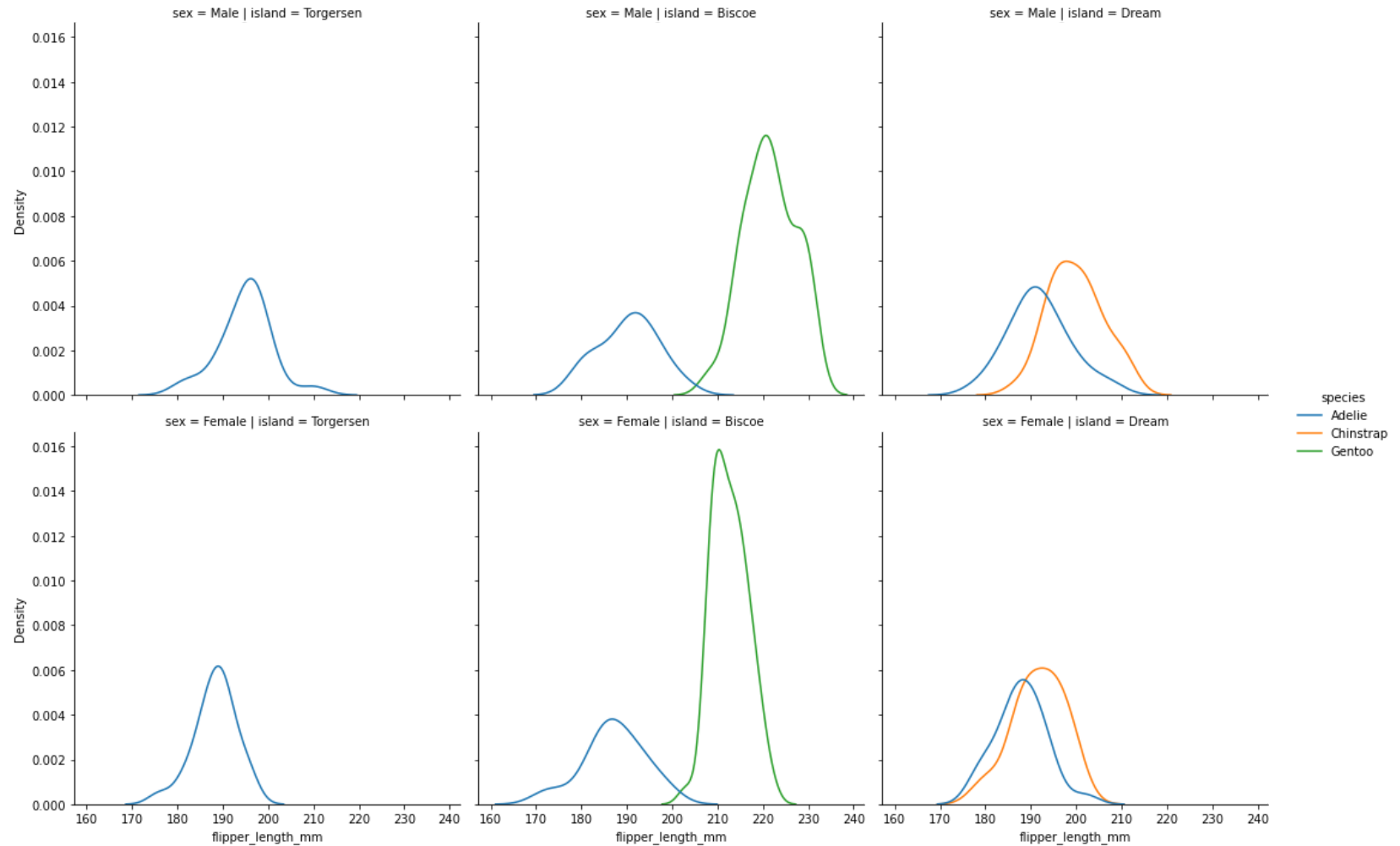
<seaborn.axisgrid.FacetGrid at 0x27b3dfcc730>





```
In [34]: sns.displot(data=penguins, x="flipper_length_mm", hue="species", row="sex", col = "island", kind=
```

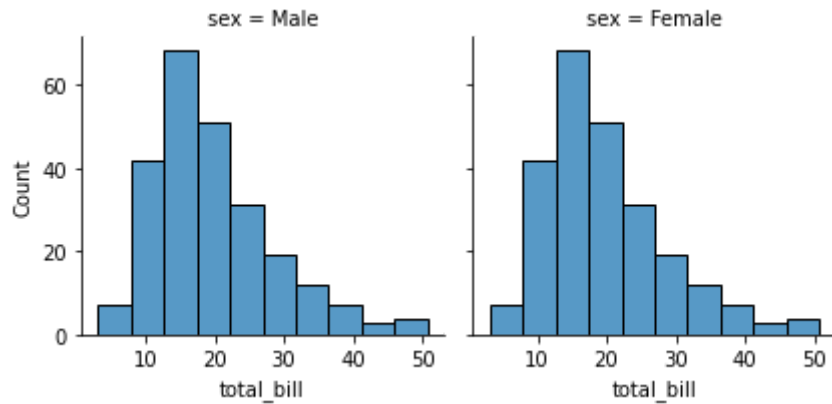
```
Out[34]: <seaborn.axisgrid.FacetGrid at 0x27b3f111190>
```



In [35]:

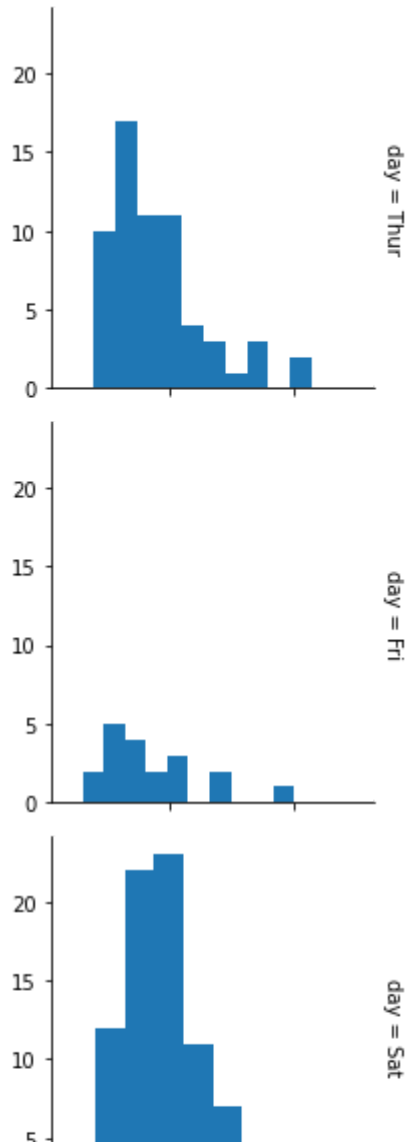
```
g = sns.FacetGrid(tips, col="sex", margin_titles=True) # define the facet grid
# the facet grid will subset the data based on the categorical variable you provide it
g.map(sns.histplot, data = tips, x = "total_bill", bins=10)
# you then 'map' a plot command (e.g. sns.distplot, or plt.hist) to the facet grid
# be sure to pass the appropriate arguments
```

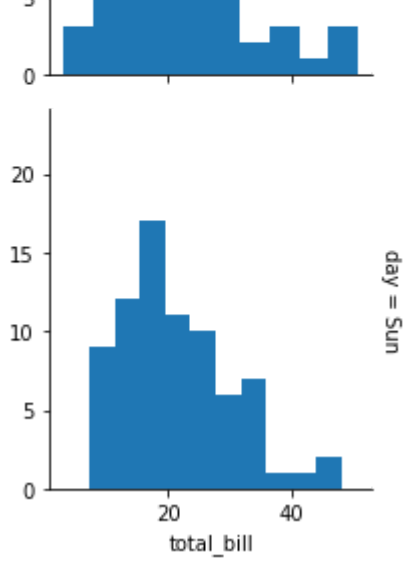
Out[35]: <seaborn.axisgrid.FacetGrid at 0x27b3f142d60>



```
In [36]: g = sns.FacetGrid(tips, row = 'day', margin_titles=True)
g.map(plt.hist, "total_bill")
```

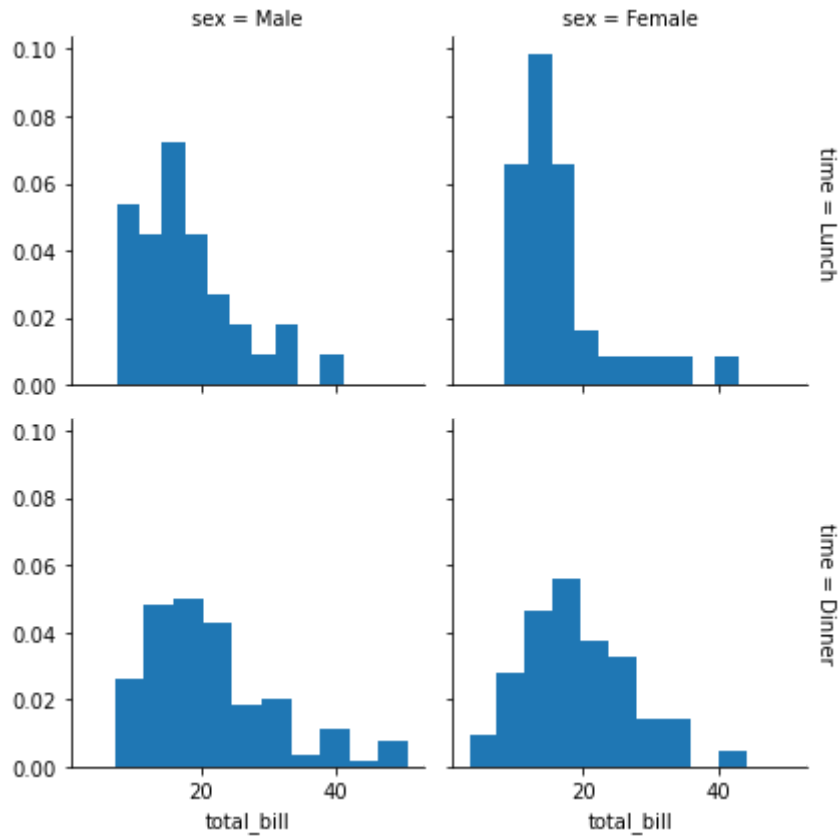
```
Out[36]: <seaborn.axisgrid.FacetGrid at 0x27b3f911940>
```





```
In [37]: g = sns.FacetGrid(tips, row = 'time', col='sex', margin_titles=True) # you can even have a 2D fac
g.map(plt.hist, "total_bill", density = True, bins=10)
```

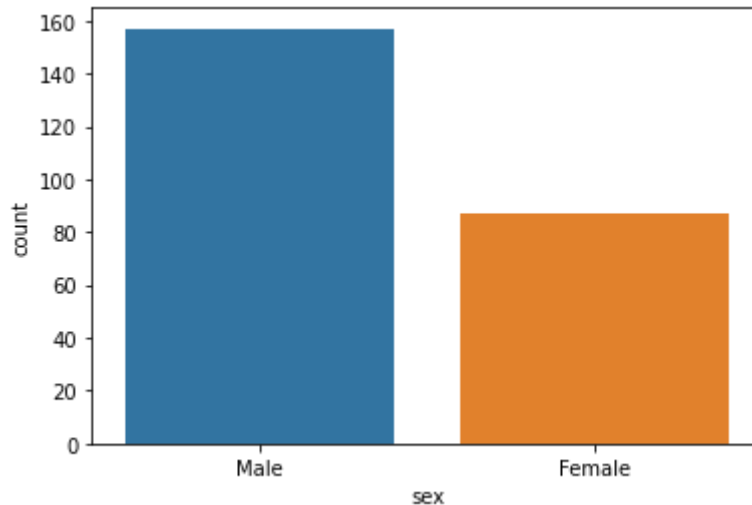
Out[37]: <seaborn.axisgrid.FacetGrid at 0x27b3f9f5430>



bar charts (count plots) for data that is only categorical

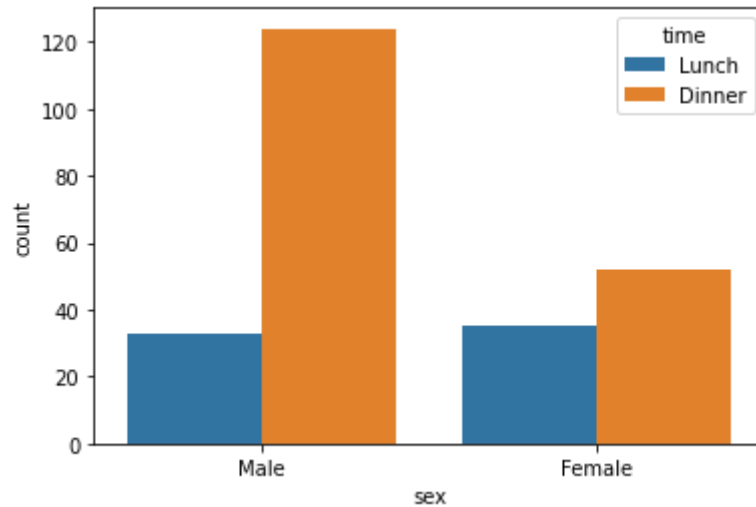
In [38]: `sns.countplot(x="sex",data=tips)`

Out[38]: `<AxesSubplot:xlabel='sex', ylabel='count'>`



```
In [39]: sns.countplot(x="sex",hue = 'time', data=tips)
```

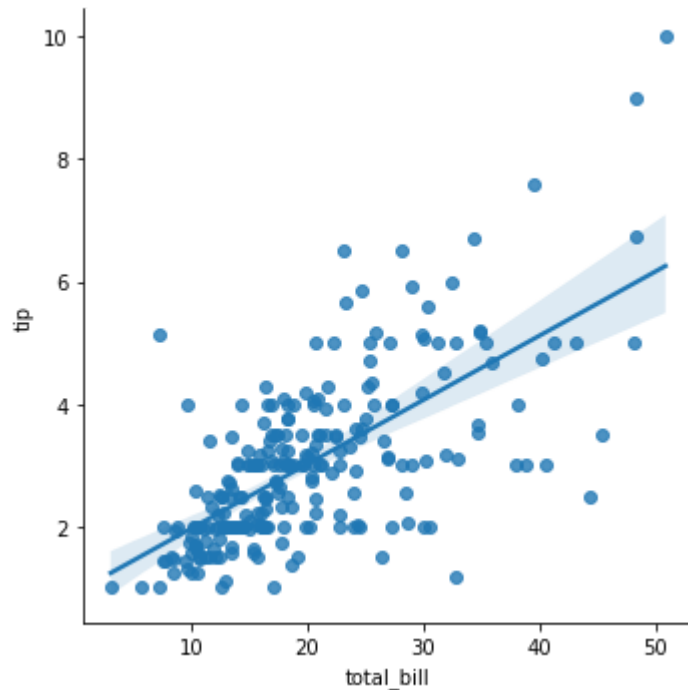
```
Out[39]: <AxesSubplot:xlabel='sex', ylabel='count'>
```



fitting basic statistical models

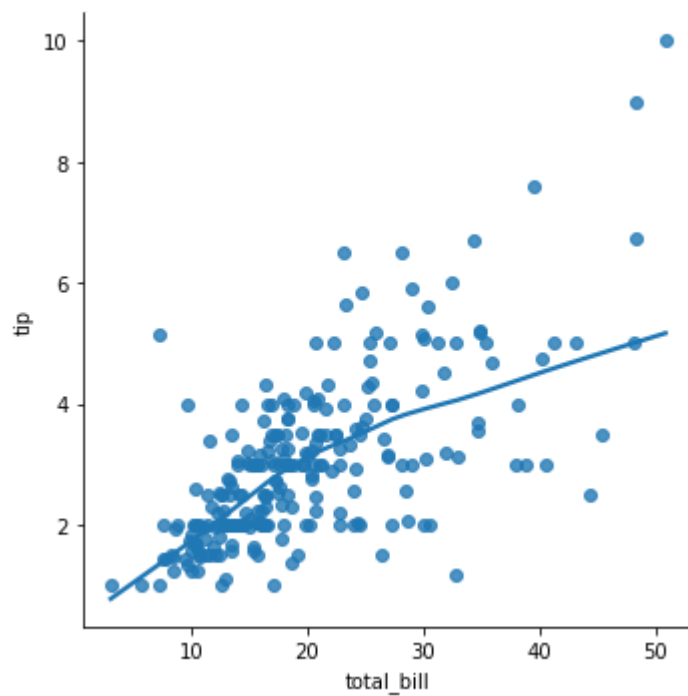
```
In [40]: sns.lmplot(x="total_bill", y="tip", data=tips)
```

```
Out[40]: <seaborn.axisgrid.FacetGrid at 0x27b40031e80>
```



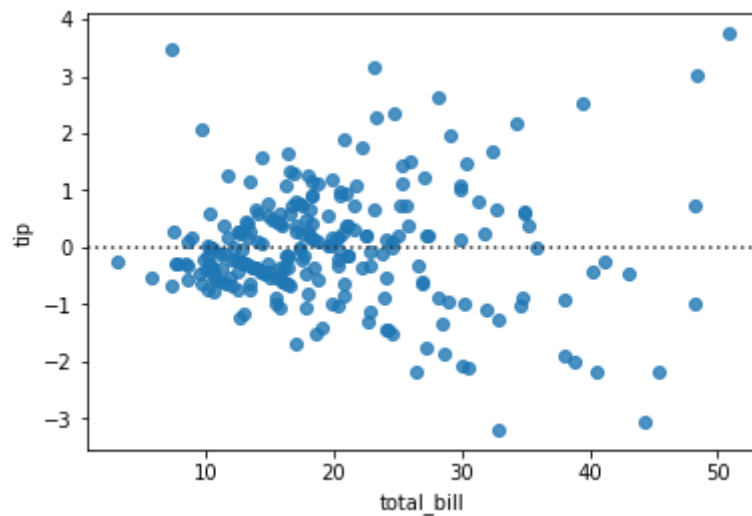
```
In [41]: sns.lmplot(x="total_bill", y="tip", data=tips, lowess=True)
```

```
Out[41]: <seaborn.axisgrid.FacetGrid at 0x27b40049f10>
```



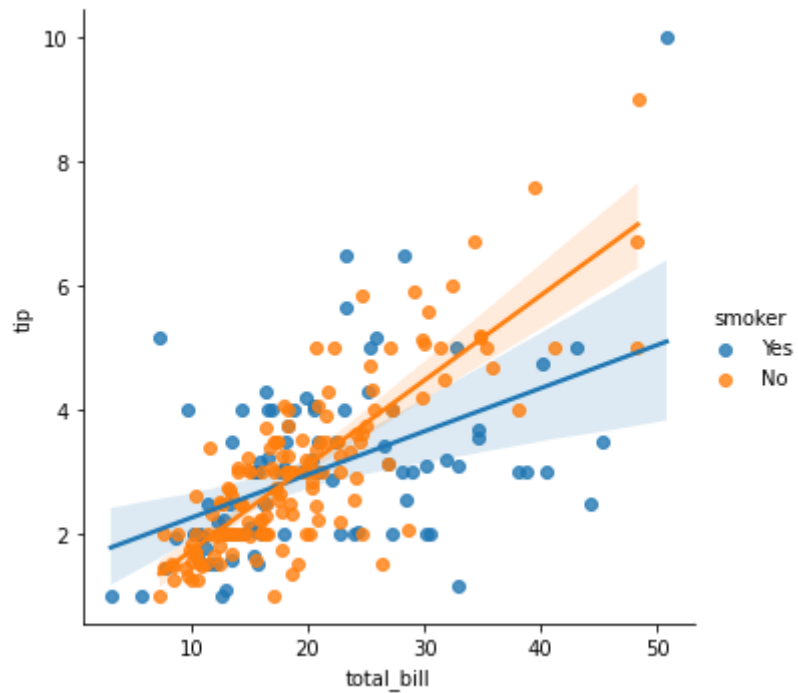
```
In [42]: sns.residplot(x="total_bill", y="tip", data=tips)
```

```
Out[42]: <AxesSubplot:xlabel='total_bill', ylabel='tip'>
```



```
In [43]: sns.lmplot(x = "total_bill", y = "tip", hue = "smoker", data=tips)
```

```
Out[43]: <seaborn.axisgrid.FacetGrid at 0x27b411762e0>
```



```
In [ ]:
```