서울사이버대학교 데이터리터러시 13주차 실습 코드

학습용 데이터 다루기

In [1]:
```
import warnings
```

In [2]:
```
warnings.filterwarnings('ignore')
```

In [3]:
```
import pandas as pd
```

In [4]:
```
import seaborn as sns
```

In [5]:
```
iris = sns.load_dataset('iris')
```

In [6]:
```
iris.head()
```

Out[6]:

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

In [7]:
```
iris.head(10)
```

Out[7]:

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa
5	5.4	3.9	1.7	0.4	setosa
6	4.6	3.4	1.4	0.3	setosa
7	5.0	3.4	1.5	0.2	setosa
8	4.4	2.9	1.4	0.2	setosa
9	4.9	3.1	1.5	0.1	setosa

In [8]:
```
iris.tail()
```

Out[8]:

	sepal_length	sepal_width	petal_length	petal_width	species
145	6.7	3.0	5.2	2.3	virginica
146	6.3	2.5	5.0	1.9	virginica
147	6.5	3.0	5.2	2.0	virginica

	sepal_length	sepal_width	petal_length	petal_width	species
148	6.2	3.4	5.4	2.3	virginica
149	5.9	3.0	5.1	1.8	virginica

In [9]:
```
type(iris)
```

Out[9]:
```
pandas.core.frame.DataFrame
```

In [10]:
```
iris.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   sepal_length  150 non-null    float64
 1   sepal_width   150 non-null    float64
 2   petal_length  150 non-null    float64
 3   petal_width   150 non-null    float64
 4   species       150 non-null    object 
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
```

In [11]:
```
iris.shape
```

Out[11]:
```
(150, 5)
```

In [12]:
```
iris.describe()
```

Out[12]:

	sepal_length	sepal_width	petal_length	petal_width
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.057333	3.758000	1.199333
std	0.828066	0.435866	1.765298	0.762238
min	4.300000	2.000000	1.000000	0.100000
25%	5.100000	2.800000	1.600000	0.300000
50%	5.800000	3.000000	4.350000	1.300000
75%	6.400000	3.300000	5.100000	1.800000
max	7.900000	4.400000	6.900000	2.500000

In [13]:
```
iris.describe(include = 'all')
```

Out[13]:

	sepal_length	sepal_width	petal_length	petal_width	species
count	150.000000	150.000000	150.000000	150.000000	150
unique	NaN	NaN	NaN	NaN	3
top	NaN	NaN	NaN	NaN	virginica
freq	NaN	NaN	NaN	NaN	50
mean	5.843333	3.057333	3.758000	1.199333	NaN
std	0.828066	0.435866	1.765298	0.762238	NaN
min	4.300000	2.000000	1.000000	0.100000	NaN

	sepal_length	sepal_width	petal_length	petal_width	species
25%	5.100000	2.800000	1.600000	0.300000	NaN
50%	5.800000	3.000000	4.350000	1.300000	NaN
75%	6.400000	3.300000	5.100000	1.800000	NaN
max	7.900000	4.400000	6.900000	2.500000	NaN

In [14]:
```
iris['species'].value_counts()
```

Out[14]:
```
virginica     50
setosa        50
versicolor    50
Name: species, dtype: int64
```

In [15]:
```
iris.species.value_counts()
```

Out[15]:
```
virginica     50
setosa        50
versicolor    50
Name: species, dtype: int64
```

In [16]:
```
iris.mean()
```

Out[16]:
```
sepal_length    5.843333
sepal_width     3.057333
petal_length    3.758000
petal_width     1.199333
dtype: float64
```

In [17]:
```
iris['sepal_length'].mean()
```

Out[17]:
```
5.843333333333335
```

In [18]:
```
iris.sepal_length.mean()
```

Out[18]:
```
5.843333333333335
```

In [19]:
```
iris.median()
```

Out[19]:
```
sepal_length    5.80
sepal_width     3.00
petal_length    4.35
petal_width     1.30
dtype: float64
```

In [20]:
```
iris['sepal_length'].median()
```

Out[20]:
```
5.8
```

In [21]:
```
iris.max()
```

Out[21]:
```
sepal_length          7.9
sepal_width           4.4
petal_length          6.9
petal_width           2.5
species         virginica
dtype: object
```

In [22]:
```
iris['sepal_length'].max()
```

Out[22]:
```
7.9
```

In [23]:
```
iris.min()
```

Out[23]:
```
sepal_length       4.3
sepal_width          2
petal_length         1
petal_width        0.1
species         setosa
dtype: object
```

In [24]:
```
iris['sepal_length'].min()
```

Out[24]: 4.3

In [25]:
```
iris.corr()
```

Out[25]:

	sepal_length	sepal_width	petal_length	petal_width
sepal_length	1.000000	-0.117570	0.871754	0.817941
sepal_width	-0.117570	1.000000	-0.428440	-0.366126
petal_length	0.871754	-0.428440	1.000000	0.962865
petal_width	0.817941	-0.366126	0.962865	1.000000

In [26]:
```
a = [[1, 2, 3, 4], [2, 4, 6, 8], [10, 9, 8, 7], [20, 60, 30, 30]]
```

In [27]:
```
df = pd.DataFrame(a)
```

In [28]:
```
df
```

Out[28]:

	0	1	2	3
0	1	2	3	4
1	2	4	6	8
2	10	9	8	7
3	20	60	30	30

In [29]:
```
df = df.T
```

In [30]:
```
df
```

Out[30]:

	0	1	2	3
0	1	2	10	20
1	2	4	9	60
2	3	6	8	30
3	4	8	7	30

In [31]:
```
df.corr()
```

Out[31]:

	0	1	2	3
0	1.0	1.0	-1.0	0.0

	0	1	2	3
1	1.0	1.0	-1.0	0.0
2	-1.0	-1.0	1.0	0.0
3	0.0	0.0	0.0	1.0

In [32]:
```
iris[['sepal_length', 'petal_length', 'petal_width']].corr()
```

Out[32]:

	sepal_length	petal_length	petal_width
sepal_length	1.000000	0.871754	0.817941
petal_length	0.871754	1.000000	0.962865
petal_width	0.817941	0.962865	1.000000

In [33]:
```
sns.get_dataset_names()
```

Out[33]:
```
['anagrams',
 'anscombe',
 'attention',
 'brain_networks',
 'car_crashes',
 'diamonds',
 'dots',
 'exercise',
 'flights',
 'fmri',
 'gammas',
 'geyser',
 'iris',
 'mpg',
 'penguins',
 'planets',
 'tips',
 'titanic']
```

In [34]:
```
tips = sns.load_dataset('tips')
```

In [35]:
```
tips.head()
```

Out[35]:

	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 [36]:
```
tips.tail()
```

Out[36]:

	total_bill	tip	sex	smoker	day	time	size
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

	total_bill	tip	sex	smoker	day	time	size
242	17.82	1.75	Male	No	Sat	Dinner	2
243	18.78	3.00	Female	No	Thur	Dinner	2

In [37]:
```
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: 7.3 KB
```

In [38]:
```
tips.describe()
```

Out[38]:

	total_bill	tip	size
count	244.000000	244.000000	244.000000
mean	19.785943	2.998279	2.569672
std	8.902412	1.383638	0.951100
min	3.070000	1.000000	1.000000
25%	13.347500	2.000000	2.000000
50%	17.795000	2.900000	2.000000
75%	24.127500	3.562500	3.000000
max	50.810000	10.000000	6.000000

In [39]:
```
tips.describe(include = 'all')
```

Out[39]:

	total_bill	tip	sex	smoker	day	time	size
count	244.000000	244.000000	244	244	244	244	244.000000
unique	NaN	NaN	2	2	4	2	NaN
top	NaN	NaN	Male	No	Sat	Dinner	NaN
freq	NaN	NaN	157	151	87	176	NaN
mean	19.785943	2.998279	NaN	NaN	NaN	NaN	2.569672
std	8.902412	1.383638	NaN	NaN	NaN	NaN	0.951100
min	3.070000	1.000000	NaN	NaN	NaN	NaN	1.000000
25%	13.347500	2.000000	NaN	NaN	NaN	NaN	2.000000
50%	17.795000	2.900000	NaN	NaN	NaN	NaN	2.000000
75%	24.127500	3.562500	NaN	NaN	NaN	NaN	3.000000
max	50.810000	10.000000	NaN	NaN	NaN	NaN	6.000000

In [40]:
```
tips['tip'].value_counts()
```

Out[40]:
```
2.00    33
3.00    23
4.00    12
5.00    10
2.50    10
        ..
2.83     1
1.58     1
3.71     1
3.35     1
2.18     1
Name: tip, Length: 123, dtype: int64
```

In [41]:
```
tips.corr()
```

Out[41]:

	total_bill	tip	size
total_bill	1.000000	0.675734	0.598315
tip	0.675734	1.000000	0.489299
size	0.598315	0.489299	1.000000

In [42]:
```
tips[['total_bill', 'tip']].corr()
```

Out[42]:

	total_bill	tip
total_bill	1.000000	0.675734
tip	0.675734	1.000000

In [43]:
```
tt = sns.load_dataset('titanic')
```

In [44]:
```
tt.head()
```

Out[44]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	decl
0	0	3	male	22.0	1	0	7.2500	S	Third	man	True	NaN
1	1	1	female	38.0	1	0	71.2833	C	First	woman	False	C
2	1	3	female	26.0	0	0	7.9250	S	Third	woman	False	NaN
3	1	1	female	35.0	1	0	53.1000	S	First	woman	False	C
4	0	3	male	35.0	0	0	8.0500	S	Third	man	True	NaN

In [45]:
```
tt.head(10)
```

Out[45]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	d
0	0	3	male	22.0	1	0	7.2500	S	Third	man	True	N
1	1	1	female	38.0	1	0	71.2833	C	First	woman	False	
2	1	3	female	26.0	0	0	7.9250	S	Third	woman	False	N
3	1	1	female	35.0	1	0	53.1000	S	First	woman	False	

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	d
4	0	3	male	35.0	0	0	8.0500	S	Third	man	True	N
5	0	3	male	NaN	0	0	8.4583	Q	Third	man	True	N
6	0	1	male	54.0	0	0	51.8625	S	First	man	True	
7	0	3	male	2.0	3	1	21.0750	S	Third	child	False	N
8	1	3	female	27.0	0	2	11.1333	S	Third	woman	False	N
9	1	2	female	14.0	1	0	30.0708	C	Second	child	False	N

In [46]:
```
tt.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 15 columns):
 #   Column       Non-Null Count  Dtype   
---  ------       --------------  -----   
 0   survived     891 non-null    int64   
 1   pclass       891 non-null    int64   
 2   sex          891 non-null    object  
 3   age          714 non-null    float64 
 4   sibsp        891 non-null    int64   
 5   parch        891 non-null    int64   
 6   fare         891 non-null    float64 
 7   embarked     889 non-null    object  
 8   class        891 non-null    category
 9   who          891 non-null    object  
 10  adult_male   891 non-null    bool    
 11  deck         203 non-null    category
 12  embark_town  889 non-null    object  
 13  alive        891 non-null    object  
 14  alone        891 non-null    bool    
dtypes: bool(2), category(2), float64(2), int64(4), object(5)
memory usage: 80.6+ KB
```

In [47]:
```
tt.describe()
```

Out[47]:

	survived	pclass	age	sibsp	parch	fare
count	891.000000	891.000000	714.000000	891.000000	891.000000	891.000000
mean	0.383838	2.308642	29.699118	0.523008	0.381594	32.204208
std	0.486592	0.836071	14.526497	1.102743	0.806057	49.693429
min	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
25%	0.000000	2.000000	20.125000	0.000000	0.000000	7.910400
50%	0.000000	3.000000	28.000000	0.000000	0.000000	14.454200
75%	1.000000	3.000000	38.000000	1.000000	0.000000	31.000000
max	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200

In [48]:
```
tt['age'].mean()
```

Out[48]:
```
29.69911764705882
```

In [49]:
```
tt['age'].median()
```

Out[49]: 28.0

In [50]:
```
m = tt['age'].median()
```

In [51]:
```
tt['age'] = tt['age'].fillna(m)
```

In [52]:
```
tt.describe()
```

Out[52]:

	survived	pclass	age	sibsp	parch	fare
count	891.000000	891.000000	891.000000	891.000000	891.000000	891.000000
mean	0.383838	2.308642	29.361582	0.523008	0.381594	32.204208
std	0.486592	0.836071	13.019697	1.102743	0.806057	49.693429
min	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
25%	0.000000	2.000000	22.000000	0.000000	0.000000	7.910400
50%	0.000000	3.000000	28.000000	0.000000	0.000000	14.454200
75%	1.000000	3.000000	35.000000	1.000000	0.000000	31.000000
max	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200

In [53]:
```
tt.head(10)
```

Out[53]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	de
0	0	3	male	22.0	1	0	7.2500	S	Third	man	True	N
1	1	1	female	38.0	1	0	71.2833	C	First	woman	False	
2	1	3	female	26.0	0	0	7.9250	S	Third	woman	False	N
3	1	1	female	35.0	1	0	53.1000	S	First	woman	False	
4	0	3	male	35.0	0	0	8.0500	S	Third	man	True	N
5	0	3	male	28.0	0	0	8.4583	Q	Third	man	True	N
6	0	1	male	54.0	0	0	51.8625	S	First	man	True	
7	0	3	male	2.0	3	1	21.0750	S	Third	child	False	N
8	1	3	female	27.0	0	2	11.1333	S	Third	woman	False	N
9	1	2	female	14.0	1	0	30.0708	C	Second	child	False	N

In [54]:
```
tt.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 15 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   survived     891 non-null    int64
 1   pclass       891 non-null    int64
 2   sex          891 non-null    object
 3   age          891 non-null    float64
 4   sibsp        891 non-null    int64
 5   parch        891 non-null    int64
 6   fare         891 non-null    float64
 7   embarked     889 non-null    object
```

```
 8   class        891 non-null    category
 9   who          891 non-null    object
 10  adult_male   891 non-null    bool
 11  deck         203 non-null    category
 12  embark_town  889 non-null    object
 13  alive        891 non-null    object
 14  alone        891 non-null    bool
dtypes: bool(2), category(2), float64(2), int64(4), object(5)
memory usage: 80.6+ KB
```

matlplotlib으로 데이터 시각화하기

In [1]:
```
import warnings
warnings.filterwarnings('ignore')
```

In [2]:
```
import numpy as np
```

In [3]:
```
import pandas as pd
```

In [4]:
```
import seaborn as sns
```

In [5]:
```
import matplotlib.pyplot as plt
```

In [6]:
```
iris = sns.load_dataset('iris')
```

In [7]:
```
iris.head()
```

Out[7]:

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

In [8]:
```
iris.groupby(iris['species']).mean()
```

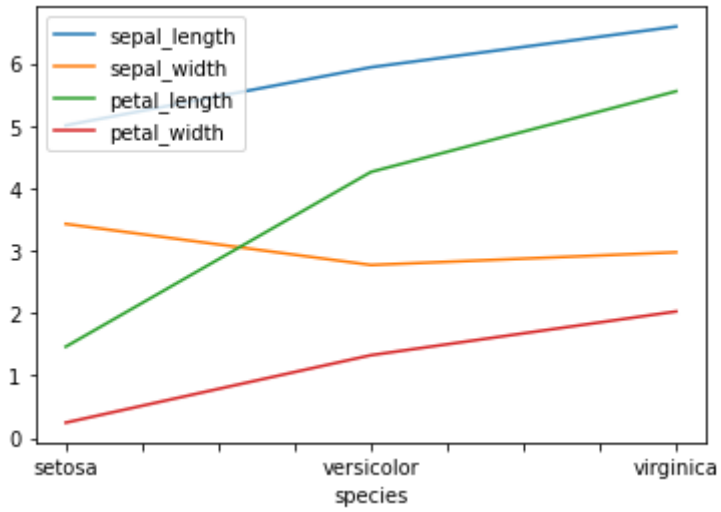
Out[8]:

species	sepal_length	sepal_width	petal_length	petal_width
setosa	5.006	3.428	1.462	0.246
versicolor	5.936	2.770	4.260	1.326
virginica	6.588	2.974	5.552	2.026

In [9]:
```
df = iris.groupby(iris['species']).mean()
```

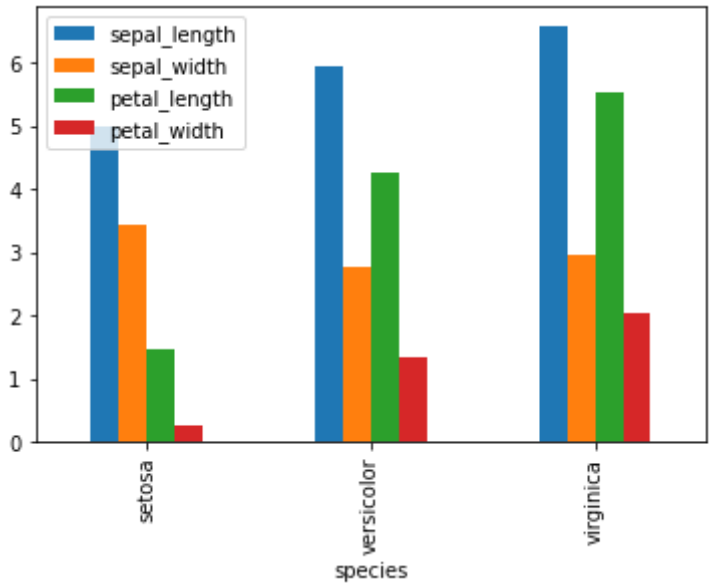
In [10]:
```
df.plot()
```

Out[10]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1f39051fc40>
```



In [11]:
```
df.plot.bar()
```

Out[11]: <matplotlib.axes._subplots.AxesSubplot at 0x1f390cbb880>



In [12]:
```
df2 = iris.groupby(iris['species']).median()
```

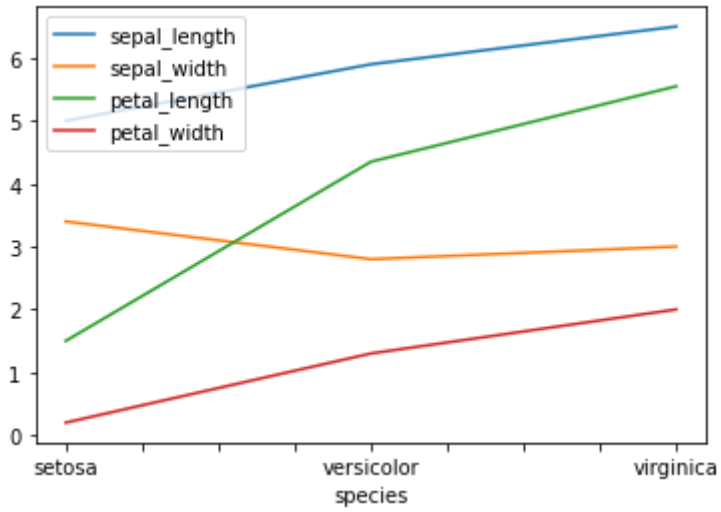
In [13]:
```
df2
```

Out[13]:

species	sepal_length	sepal_width	petal_length	petal_width
setosa	5.0	3.4	1.50	0.2
versicolor	5.9	2.8	4.35	1.3
virginica	6.5	3.0	5.55	2.0

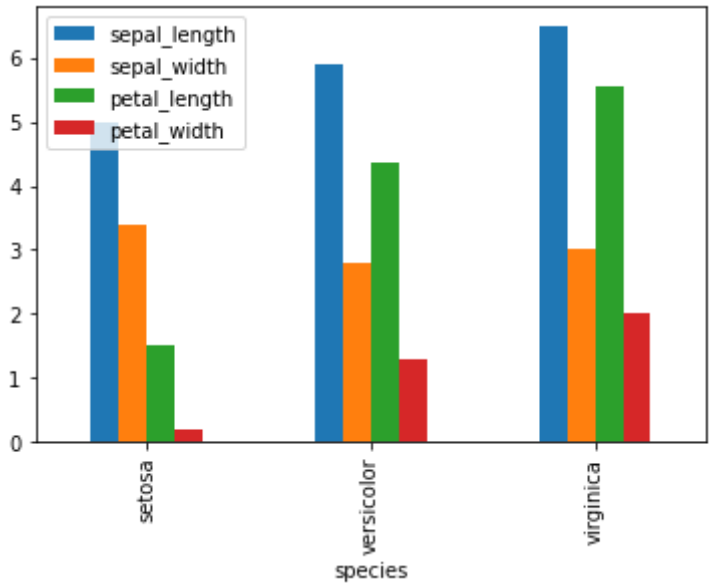
In [15]:
```
df2.plot()
```

Out[15]: <matplotlib.axes._subplots.AxesSubplot at 0x1f390d4be20>



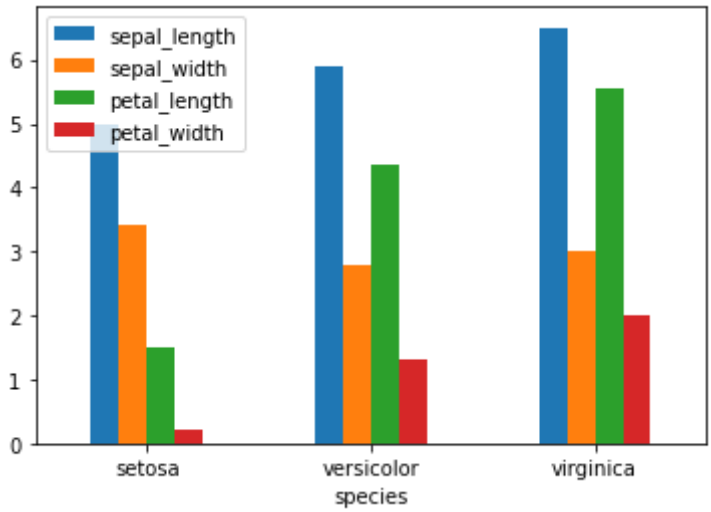
In [16]:
```
df2.plot.bar()
```

Out[16]: <matplotlib.axes._subplots.AxesSubplot at 0x1f390f9bd90>



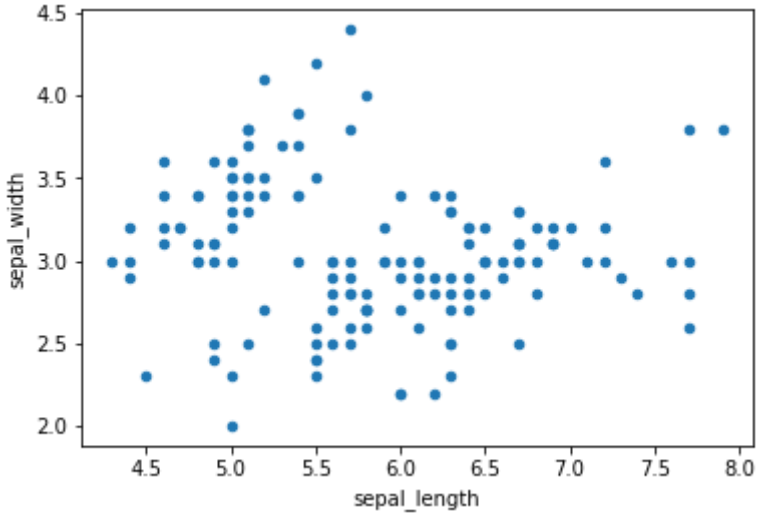
In [17]:
```
df2.plot.bar(rot = 0)
```

Out[17]: <matplotlib.axes._subplots.AxesSubplot at 0x1f39101fa60>



In [18]:
```
iris.plot.scatter(x = 'sepal_length', y = 'sepal_width')
```

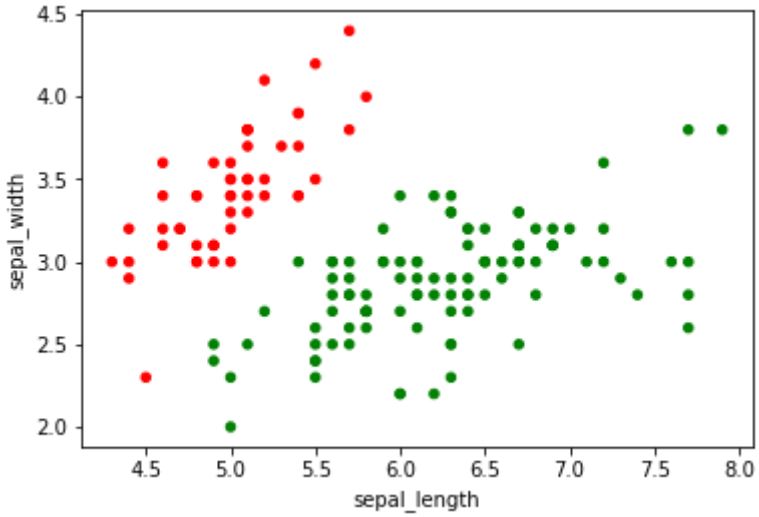
Out[18]: <matplotlib.axes._subplots.AxesSubplot at 0x1f3910940d0>



In [19]:
```
x = np.where(iris['species'] == 'setosa', 'red', 'green')
```

In [20]:
```
iris.plot.scatter(x = 'sepal_length', y = 'sepal_width', c = x)
```

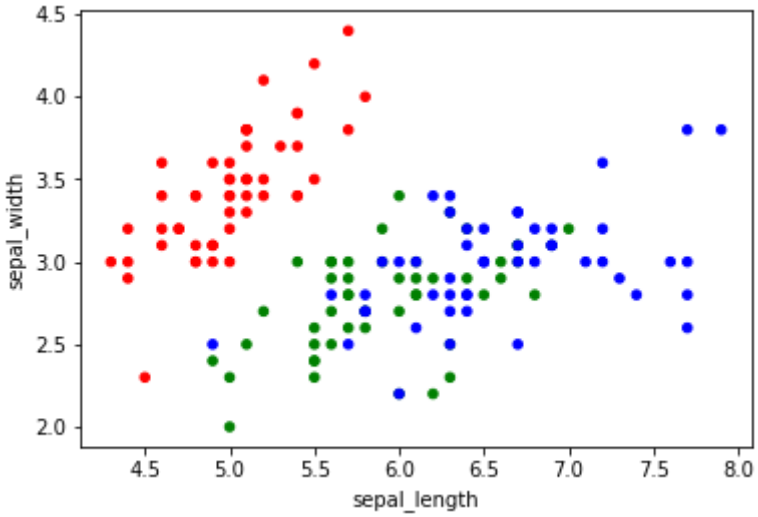
Out[20]: <matplotlib.axes._subplots.AxesSubplot at 0x1f391100730>



In [21]:
```
x = np.where(iris['species'] == 'setosa', 'red',
    np.where(iris['species'] == 'versicolor', 'green', 'blue'))
```

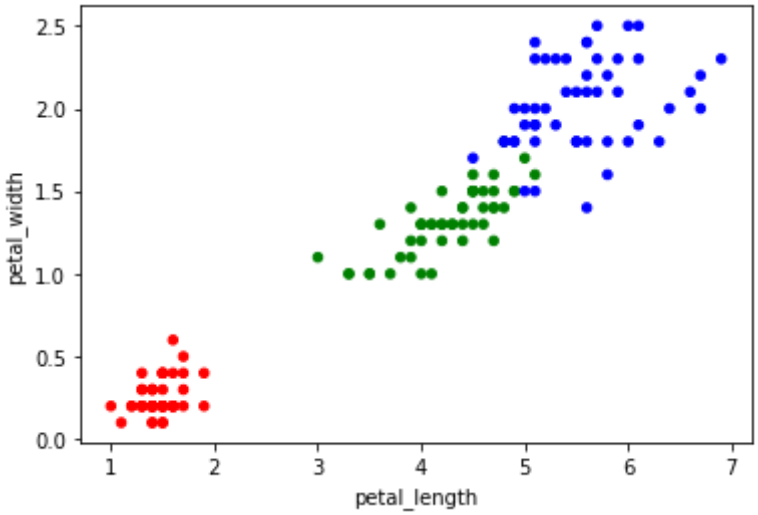
In [22]:
```
iris.plot.scatter(x = 'sepal_length', y = 'sepal_width', c = x)
```

Out[22]: <matplotlib.axes._subplots.AxesSubplot at 0x1f39112e940>



In [23]:
```
iris.plot.scatter(x = 'petal_length', y = 'petal_width', c = x)
```

Out[23]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1f3911ac8b0>
```



In [24]:
```
tips = sns.load_dataset('tips')
```

In [25]:
```
tips.head()
```

Out[25]:

	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 [26]:
```
tips.groupby(tips['time']).mean()
```

Out[26]:

time	total_bill	tip	size
Lunch	17.168676	2.728088	2.411765
Dinner	20.797159	3.102670	2.630682

In [27]:
```
tips.groupby(tips['day']).mean()
```

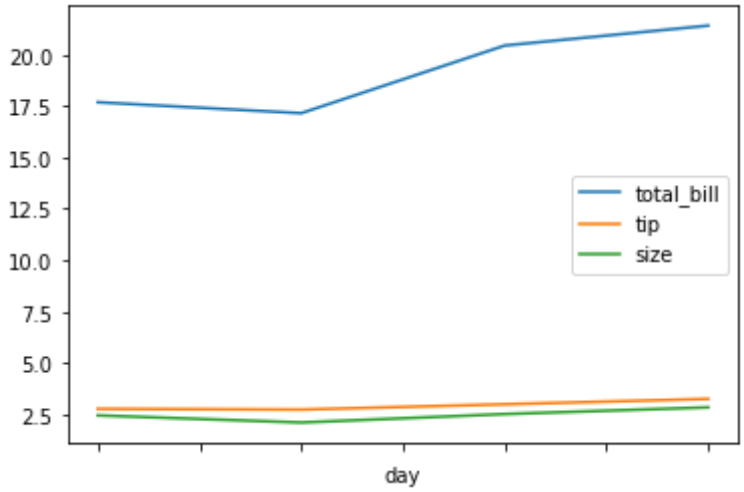
Out[27]:

day	total_bill	tip	size
Thur	17.682742	2.771452	2.451613
Fri	17.151579	2.734737	2.105263
Sat	20.441379	2.993103	2.517241
Sun	21.410000	3.255132	2.842105

In [28]:
```
df3 = tips.groupby(tips['day']).mean()
```

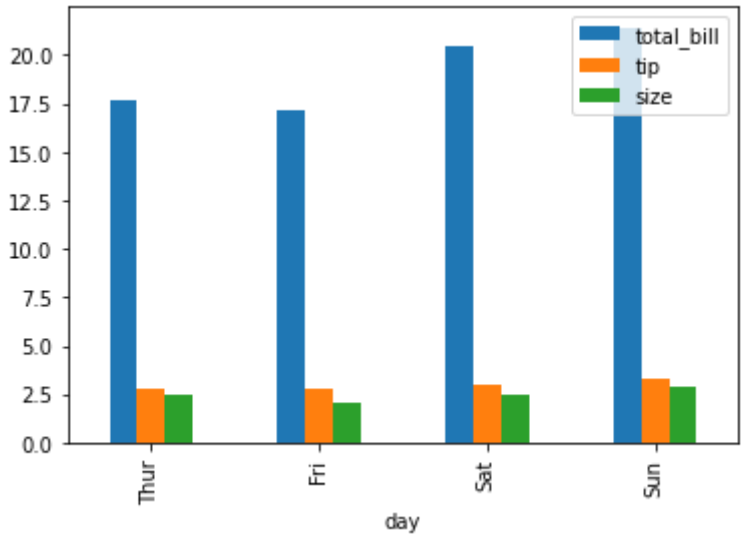
In [29]:
```
df3.plot()
```

Out[29]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1f390ca50d0>
```



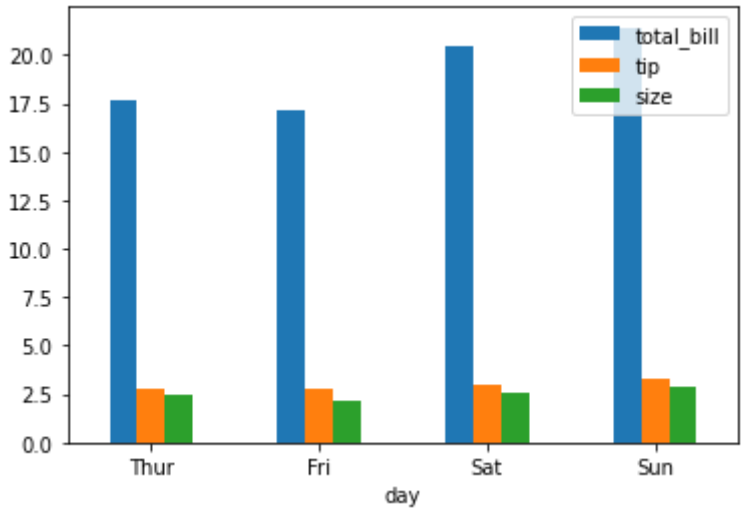
In [30]:
```
df3.plot.bar()
```

Out[30]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1f391016f40>
```



In [31]:
```
df3.plot.bar(rot = 0)
```

Out[31]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1f3912d4fa0>
```



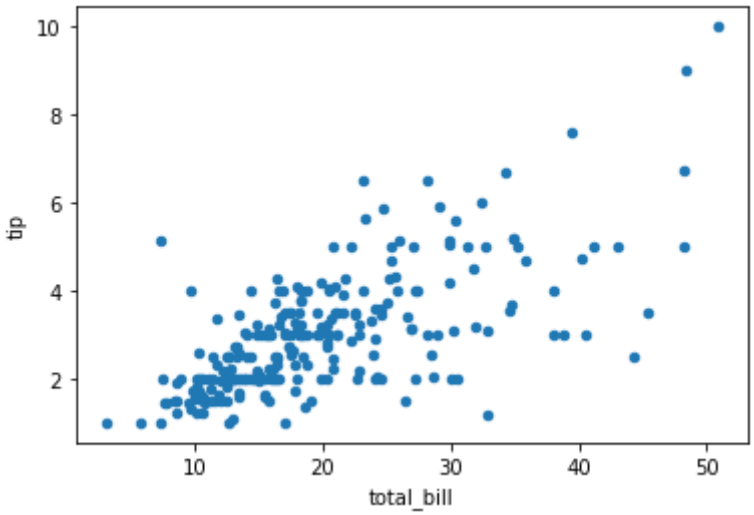
In [32]:
```
tips.corr()
```

Out[32]:

	total_bill	tip	size
total_bill	1.000000	0.675734	0.598315
tip	0.675734	1.000000	0.489299
size	0.598315	0.489299	1.000000

In [33]:
```
tips.plot.scatter(x = 'total_bill', y = 'tip')
```

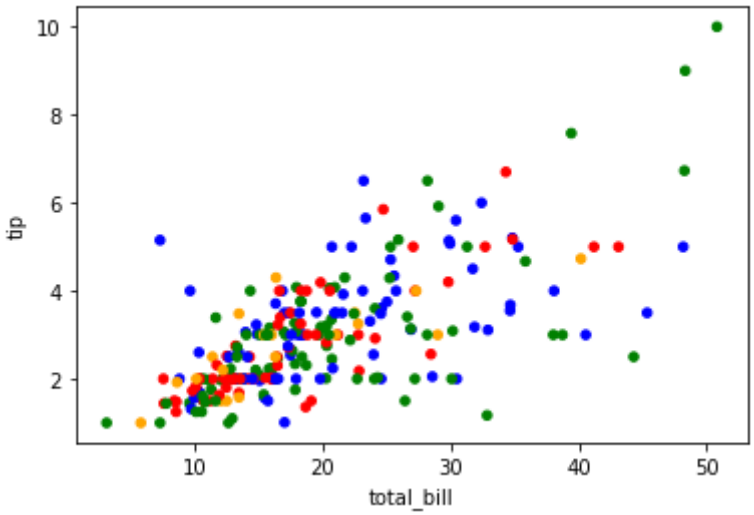
Out[33]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1f39135fb80>
```



In [34]:
```
x = np.where(tips['day'] == 'Thur', 'red',
    np.where(tips['day'] == 'Fri', 'orange',
    np.where(tips['day'] == 'Sat', 'green', 'blue')))
```

In [35]:
```
tips.plot.scatter(x = 'total_bill', y = 'tip', c = x)
```

Out[35]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1f39139d730>
```


10
8
6
4
2
tip
10
20
30
40
50
total_bill

seaborn으로 데이터 시각화하기

In [1]:
```
import warnings
warnings.filterwarnings('ignore')
```

In [2]:
```
import pandas as pd
```

In [3]:
```
import seaborn as sns
```

In [4]:
```
iris = sns.load_dataset('iris')
```

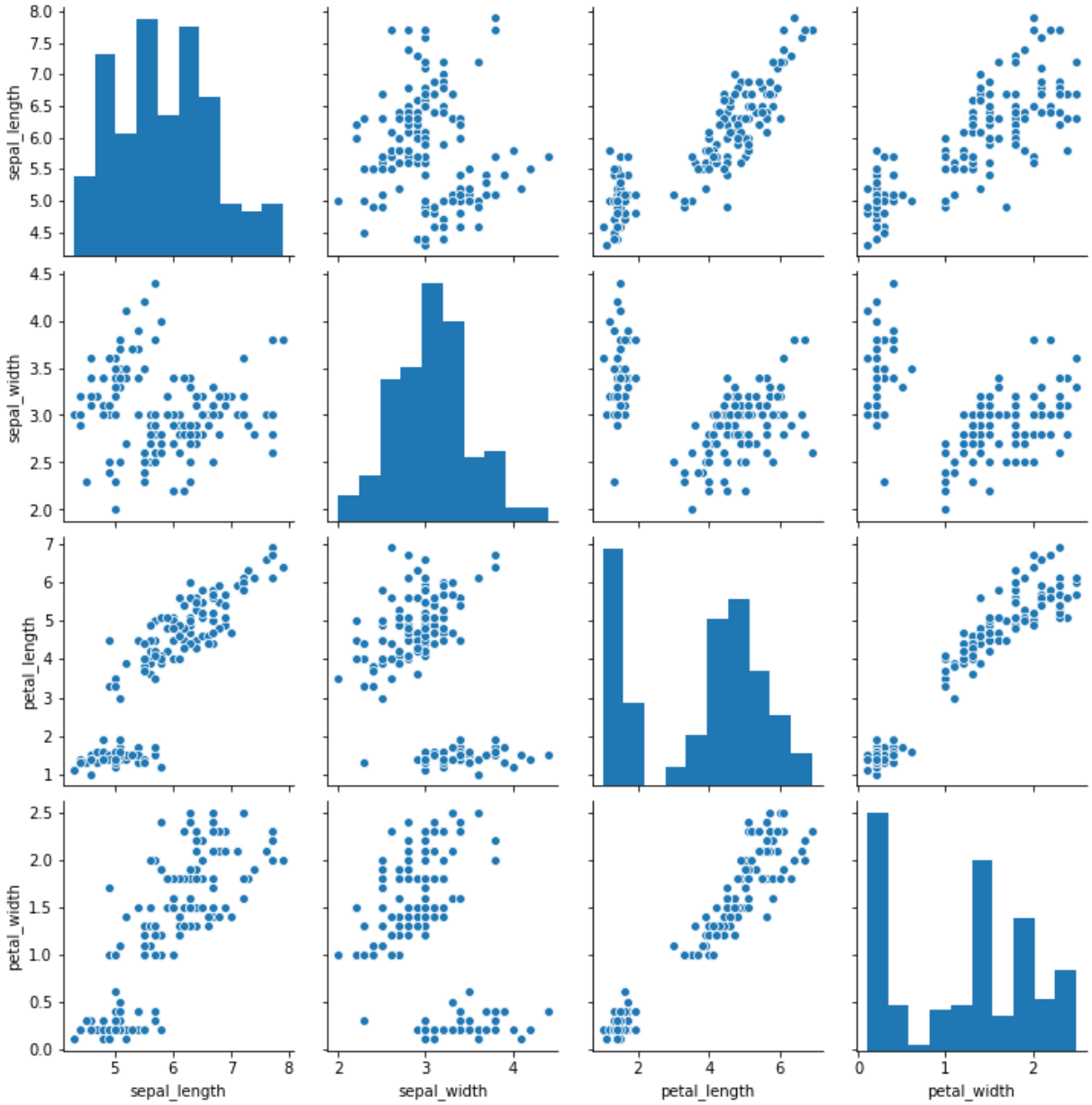
In [5]:
```
iris.head()
```

Out[5]:

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

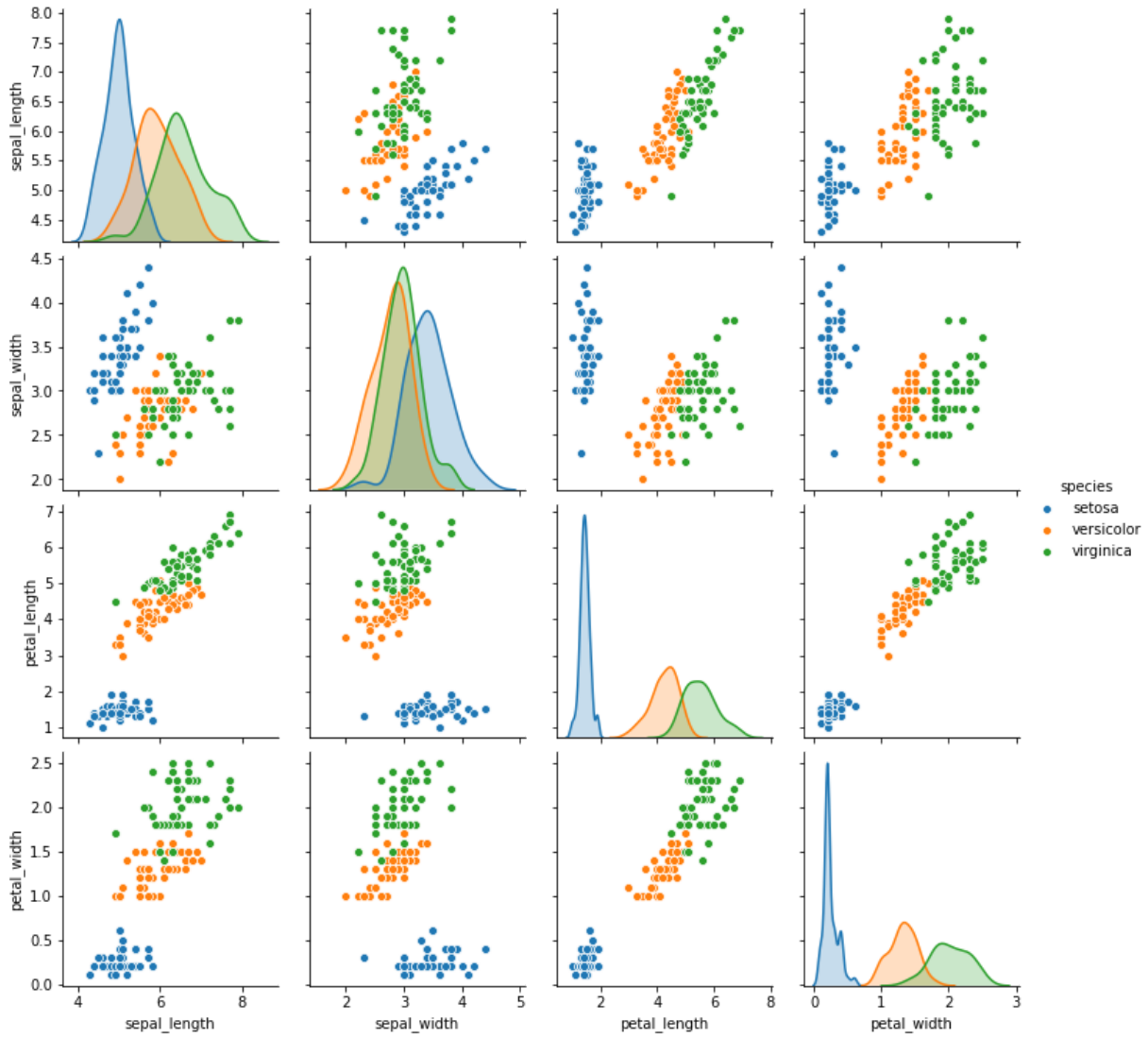
In [6]:
```
sns.pairplot(data = iris)
```

Out[6]:
```
<seaborn.axisgrid.PairGrid at 0x18cc08dbfa0>
```



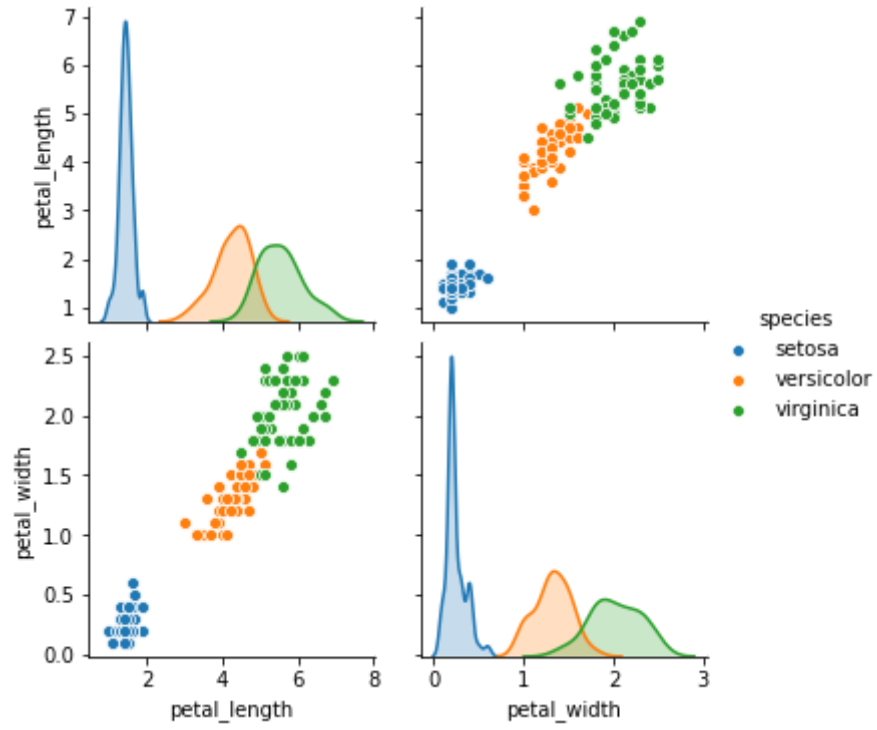
In [7]:
```
sns.pairplot(data = iris, hue = 'species')
```

Out[7]:
```
<seaborn.axisgrid.PairGrid at 0x18cc64b63d0>
```



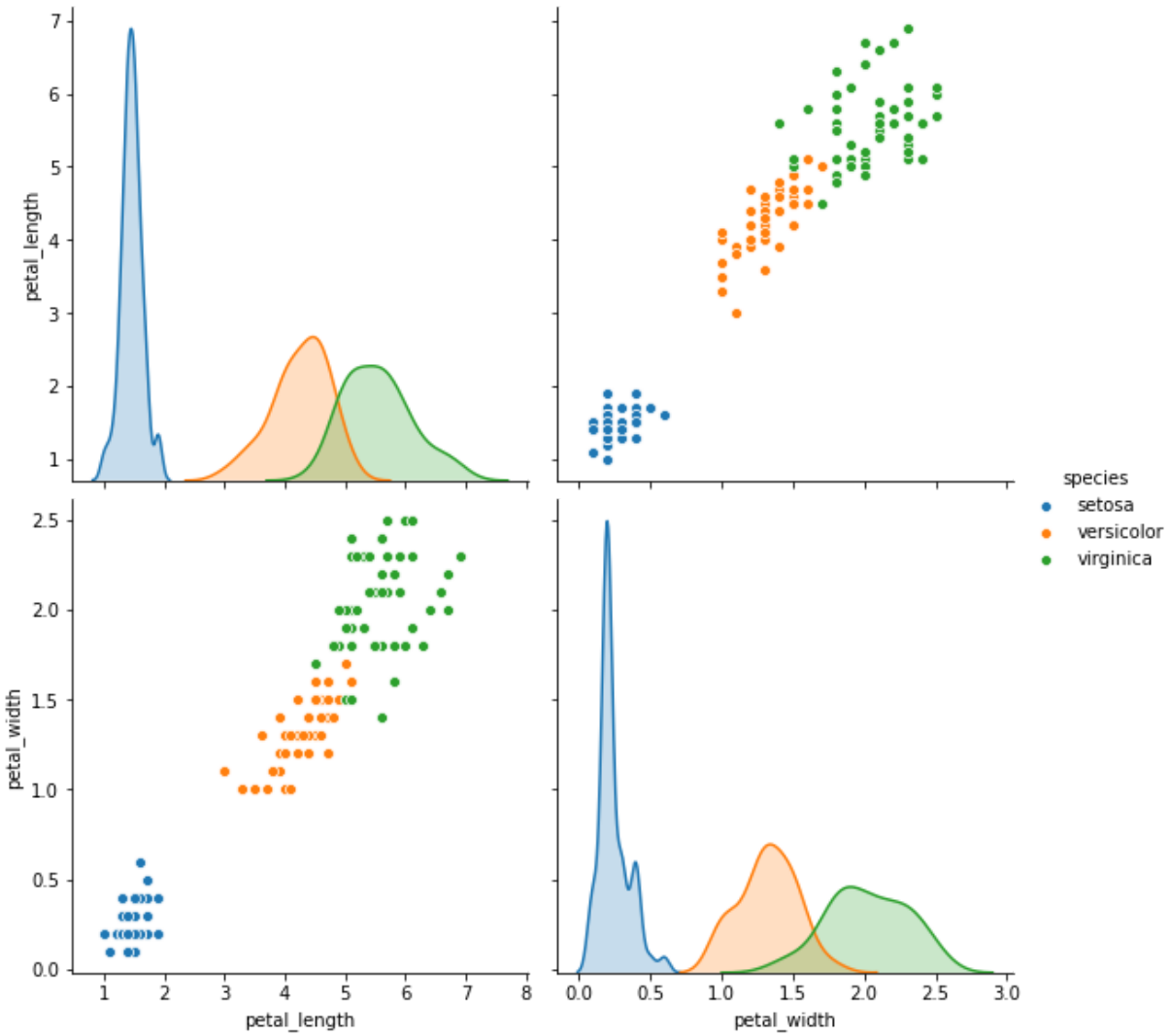
In [8]:
```
sns.pairplot(data = iris, hue = 'species', vars = ['petal_length', 'petal_width'])
```

Out[8]:
```
<seaborn.axisgrid.PairGrid at 0x18cc6f9c250>
```



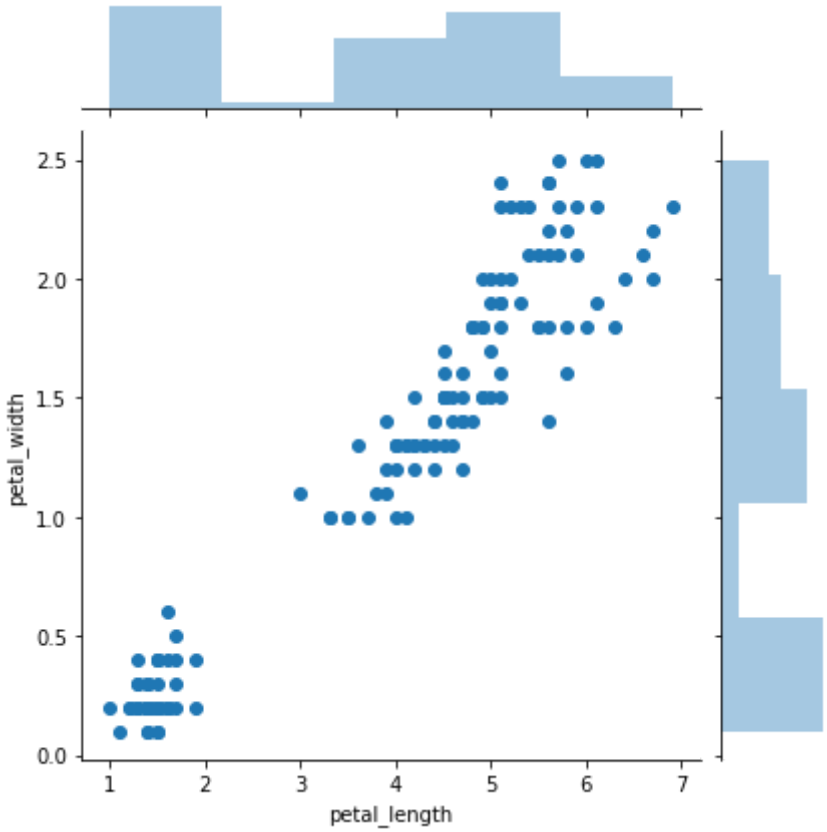
In [10]:
```
sns.pairplot(data = iris, hue = 'species', vars = ['petal_length', 'petal_width'], he
```

Out[10]: <seaborn.axisgrid.PairGrid at 0x18cc772b1c0>



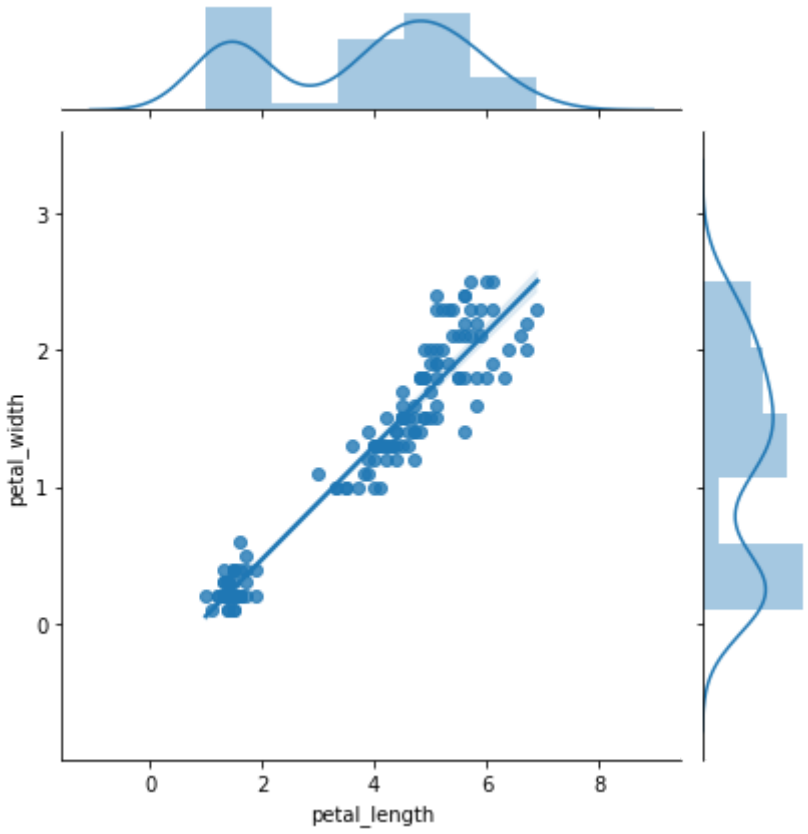
In [11]:
```
sns.jointplot(x = 'petal_length', y = 'petal_width', data = iris, kind = 'scatter')
```

Out[11]: <seaborn.axisgrid.JointGrid at 0x18cc703cd00>



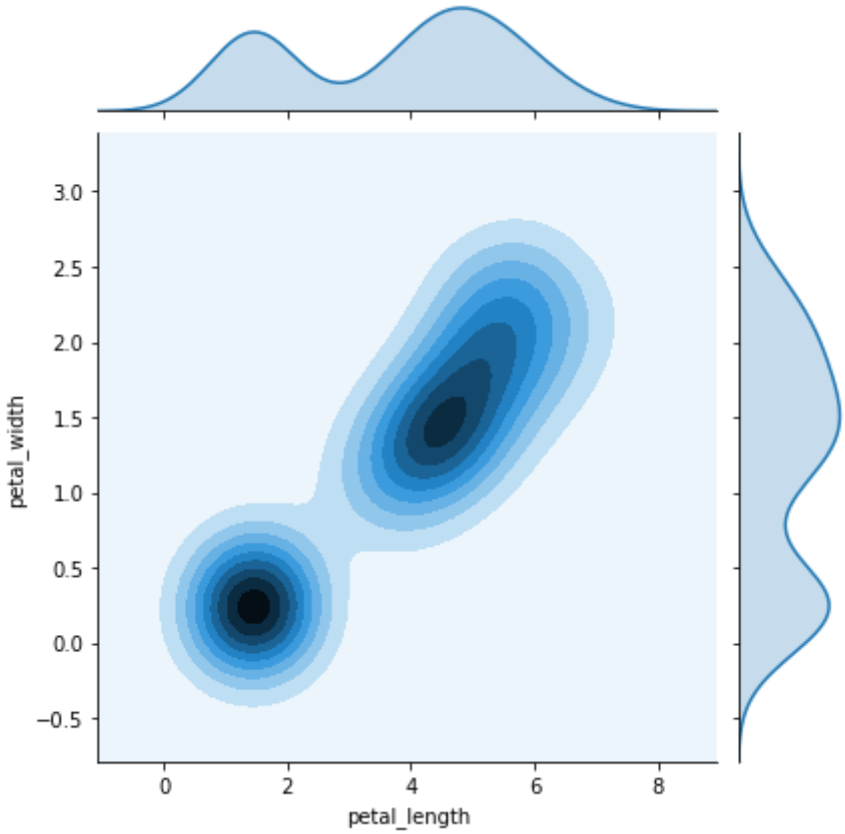
In [12]:
```
sns.jointplot(x = 'petal_length', y = 'petal_width', data = iris, kind = 'reg')
```

Out[12]: <seaborn.axisgrid.JointGrid at 0x18cc8f3aa30>



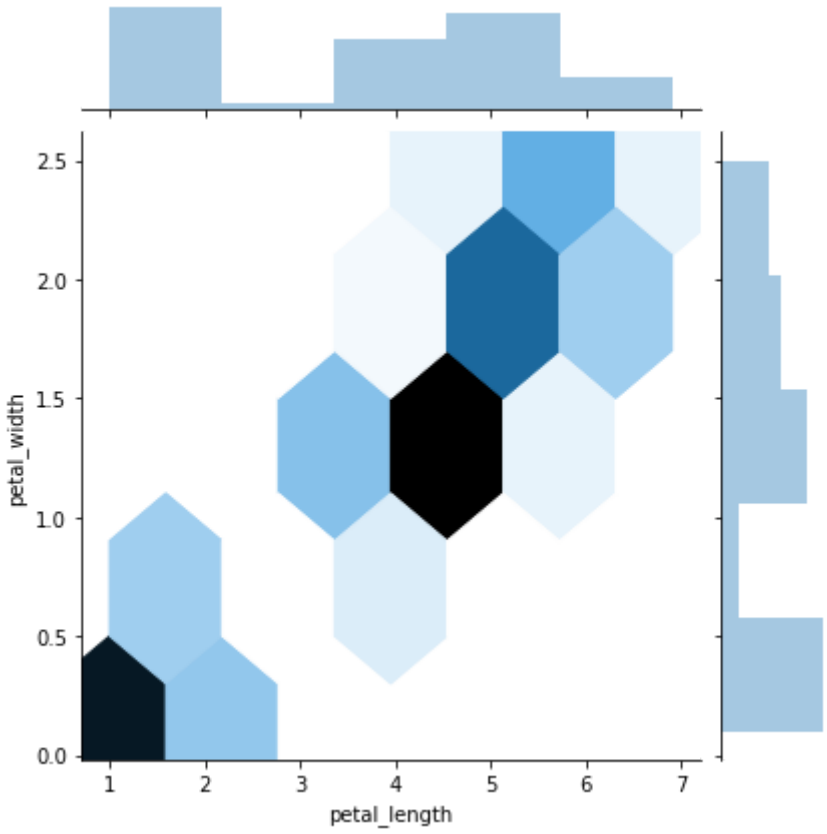
In [13]:
```
sns.jointplot(x = 'petal_length', y = 'petal_width', data = iris, kind = 'kde')
```

Out[13]: <seaborn.axisgrid.JointGrid at 0x18cc9034310>



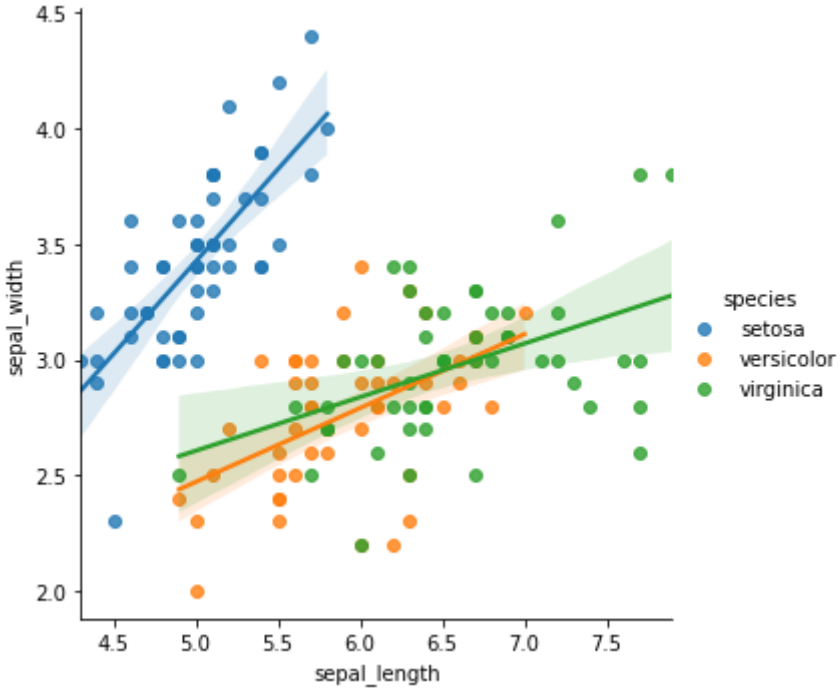
In [14]:
```
sns.jointplot(x = 'petal_length', y = 'petal_width', data = iris, kind = 'hex')
```

Out[14]: <seaborn.axisgrid.JointGrid at 0x18cc708e100>



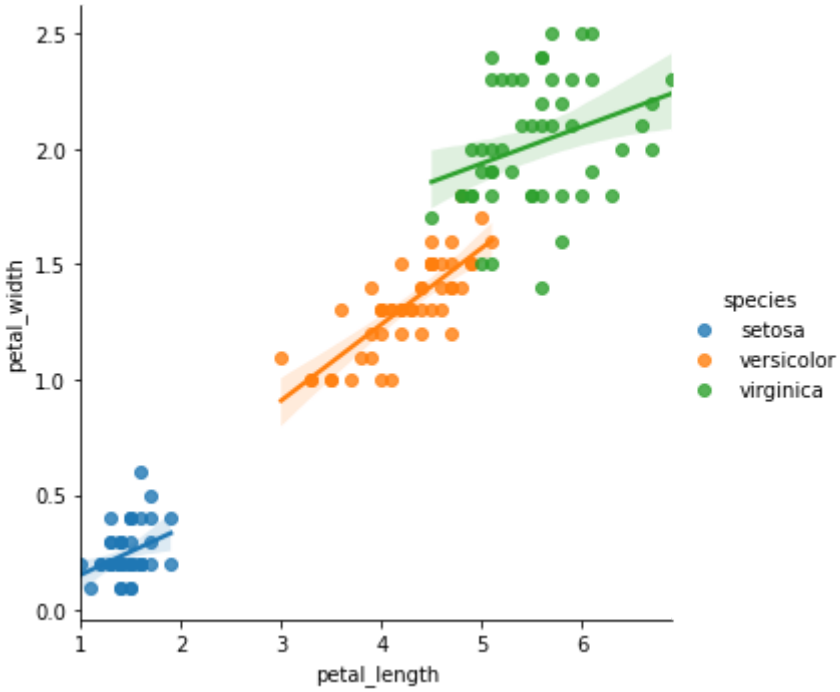
In [15]:
```
sns.lmplot(x = 'sepal_length', y = 'sepal_width', hue = 'species', data = iris)
```

Out[15]: <seaborn.axisgrid.FacetGrid at 0x18cc7147880>



In [16]:
```
sns.lmplot(x = 'petal_length', y = 'petal_width', hue = 'species', data = iris)
```

Out[16]:
```
<seaborn.axisgrid.FacetGrid at 0x18cc71cc310>
```



In [17]:
```
iris.corr()
```

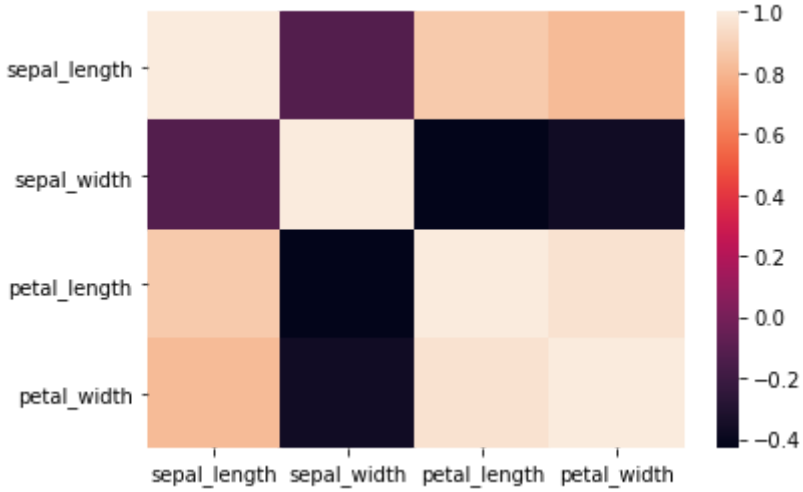
Out[17]:

	sepal_length	sepal_width	petal_length	petal_width
sepal_length	1.000000	-0.117570	0.871754	0.817941
sepal_width	-0.117570	1.000000	-0.428440	-0.366126
petal_length	0.871754	-0.428440	1.000000	0.962865
petal_width	0.817941	-0.366126	0.962865	1.000000

In [18]:
```
sns.heatmap(iris.corr())
```

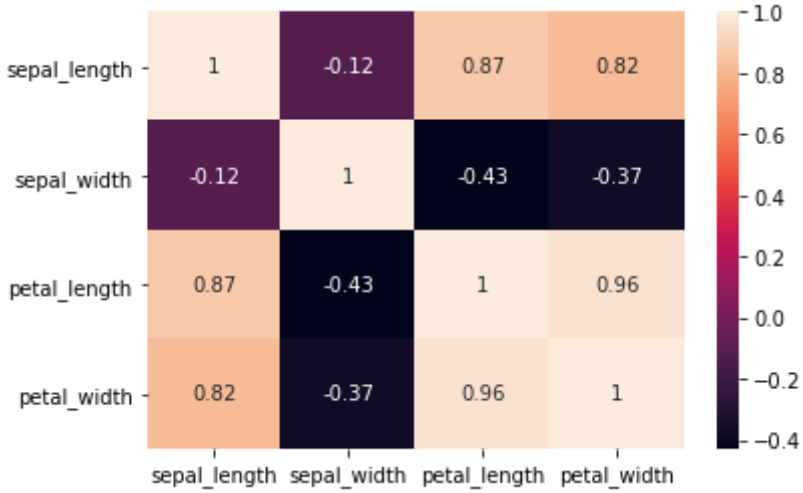
```
<matplotlib.axes._subplots.AxesSubplot at 0x18cc725da30>
```

Out[18]:



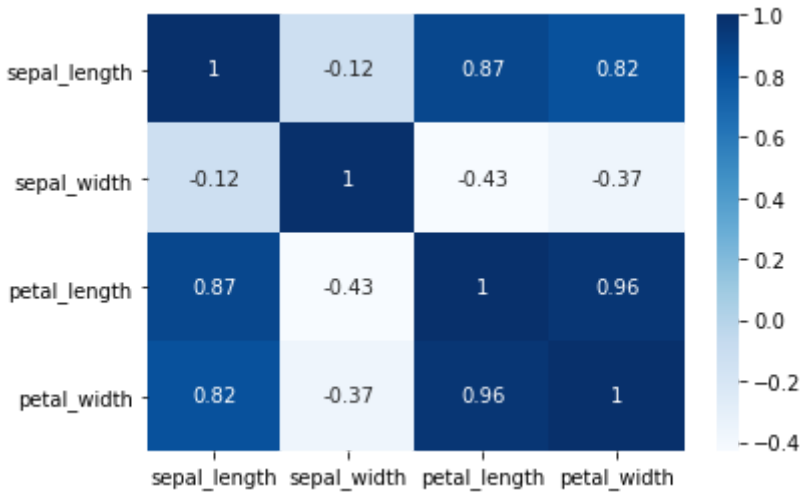
In [19]:
```
sns.heatmap(iris.corr(), annot = True)
```

Out[19]: <matplotlib.axes._subplots.AxesSubplot at 0x18cc8b518e0>



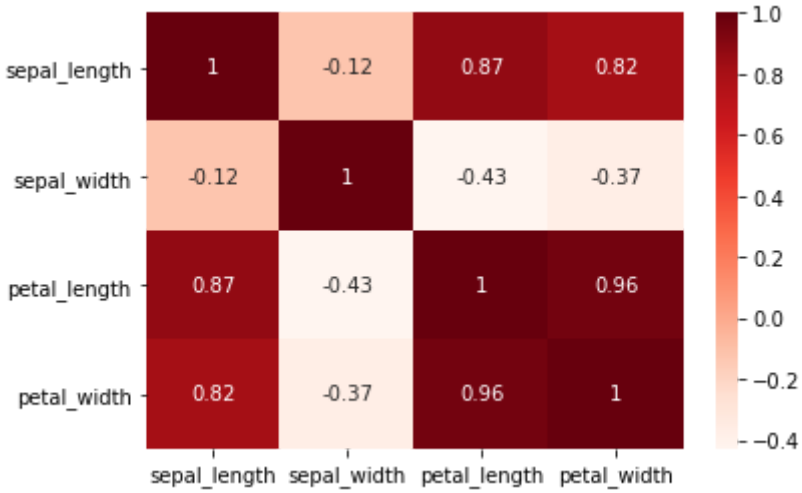
In [20]:
```
sns.heatmap(iris.corr(), annot = True, cmap = 'Blues')
```

Out[20]: <matplotlib.axes._subplots.AxesSubplot at 0x18cc8bf4580>



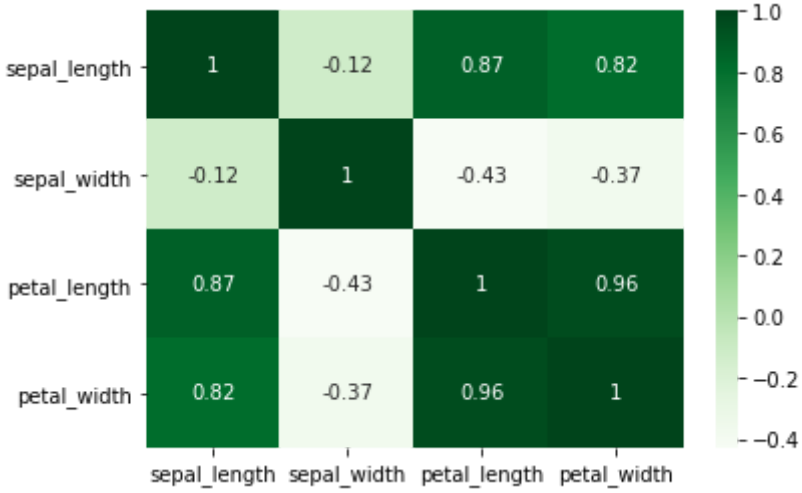
In [21]:
```
sns.heatmap(iris.corr(), annot = True, cmap = 'Reds')
```

Out[21]: <matplotlib.axes._subplots.AxesSubplot at 0x18cc8c9ee80>



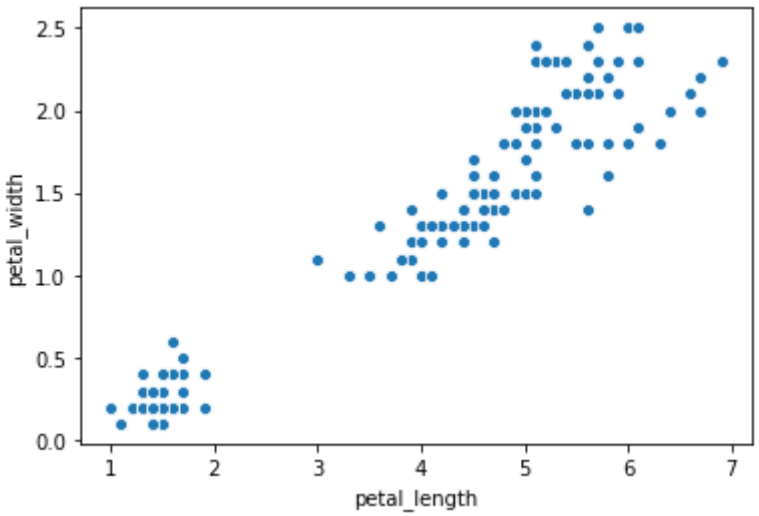
In [22]:
```
sns.heatmap(iris.corr(), annot = True, cmap = 'Greens')
```

Out[22]: <matplotlib.axes._subplots.AxesSubplot at 0x18cc91e9820>



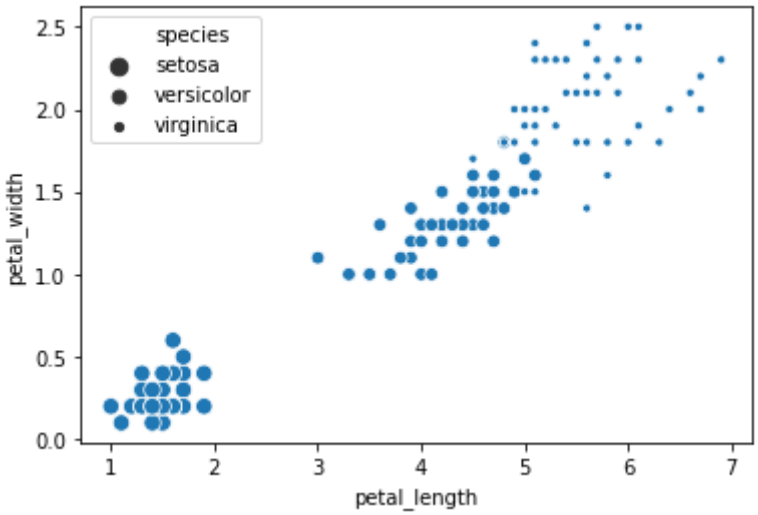
In [23]:
```
sns.scatterplot(x = 'petal_length', y = 'petal_width', data = iris)
```

Out[23]: <matplotlib.axes._subplots.AxesSubplot at 0x18cc9290bb0>



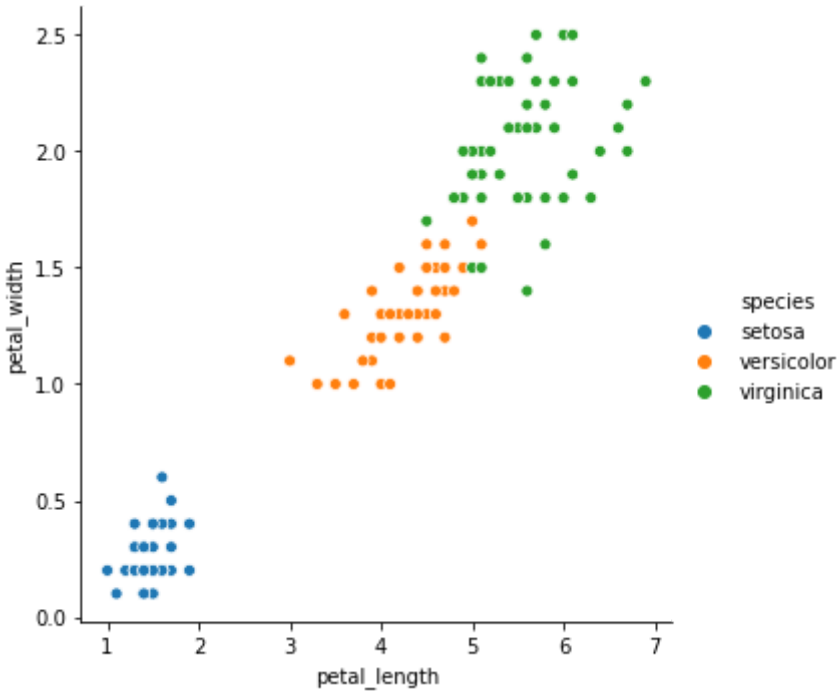
In [24]:
```
sns.scatterplot(x = 'petal_length', y = 'petal_width', size = 'species', data = iris)
```

Out[24]: <matplotlib.axes._subplots.AxesSubplot at 0x18cc9249610>



In [25]:
```
sns.relplot(x = 'petal_length', y = 'petal_width', hue = 'species', data = iris)
```

Out[25]:
```
<seaborn.axisgrid.FacetGrid at 0x18cca2f4280>
```



In [26]:
```
tips = sns.load_dataset('tips')
```

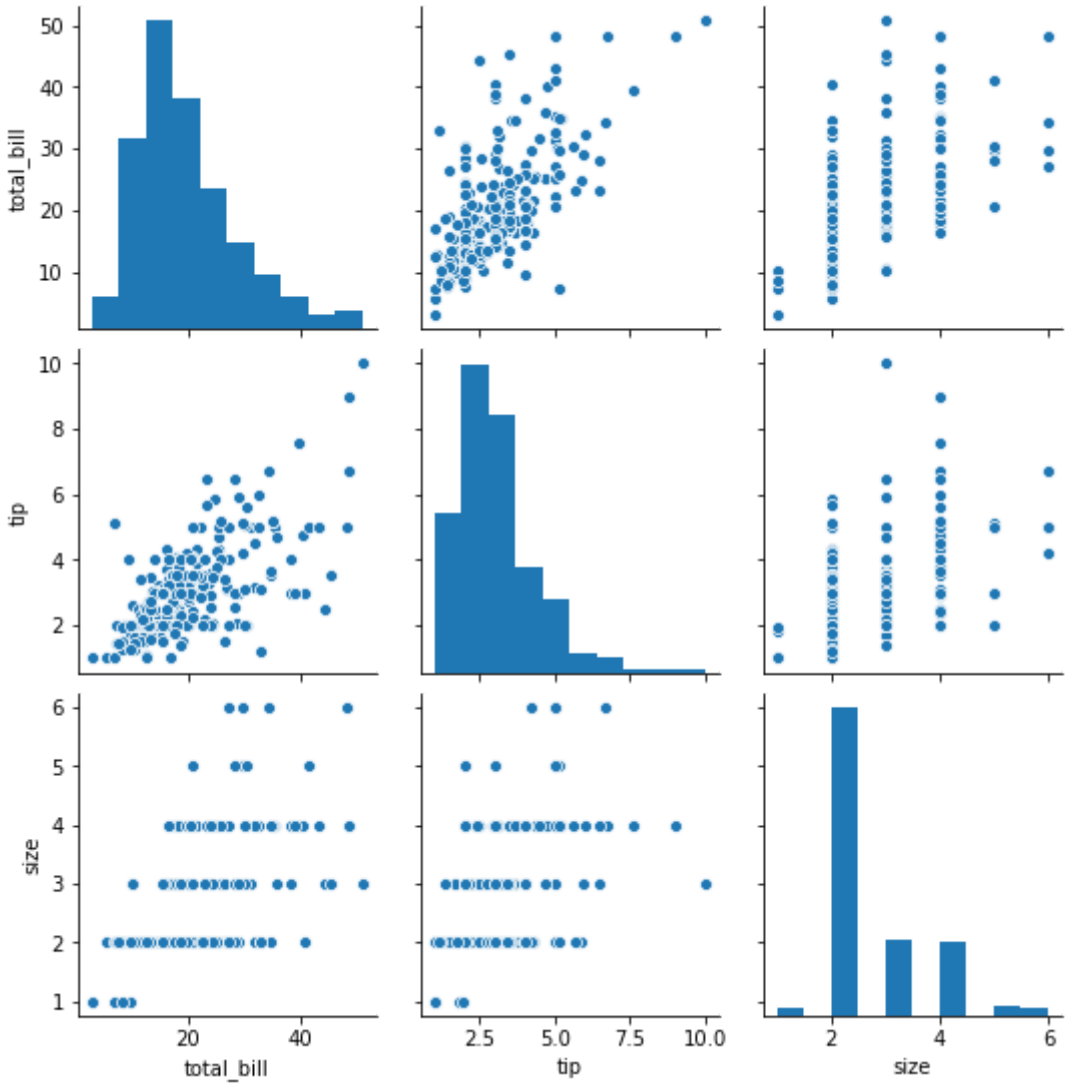
In [27]:
```
tips.head()
```

Out[27]:

	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

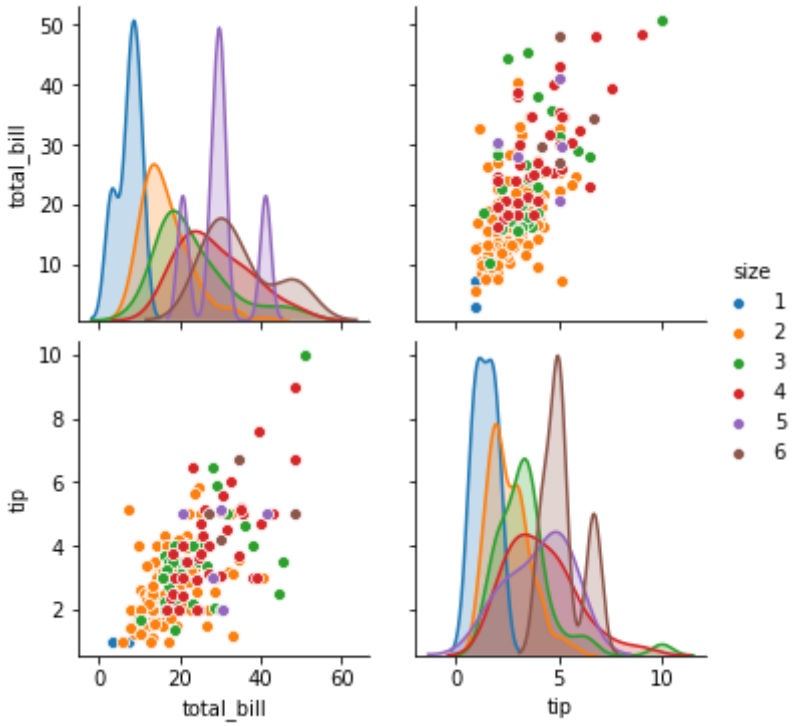
In [28]:
```
sns.pairplot(tips)
```

Out[28]:
```
<seaborn.axisgrid.PairGrid at 0x18cca38b040>
```



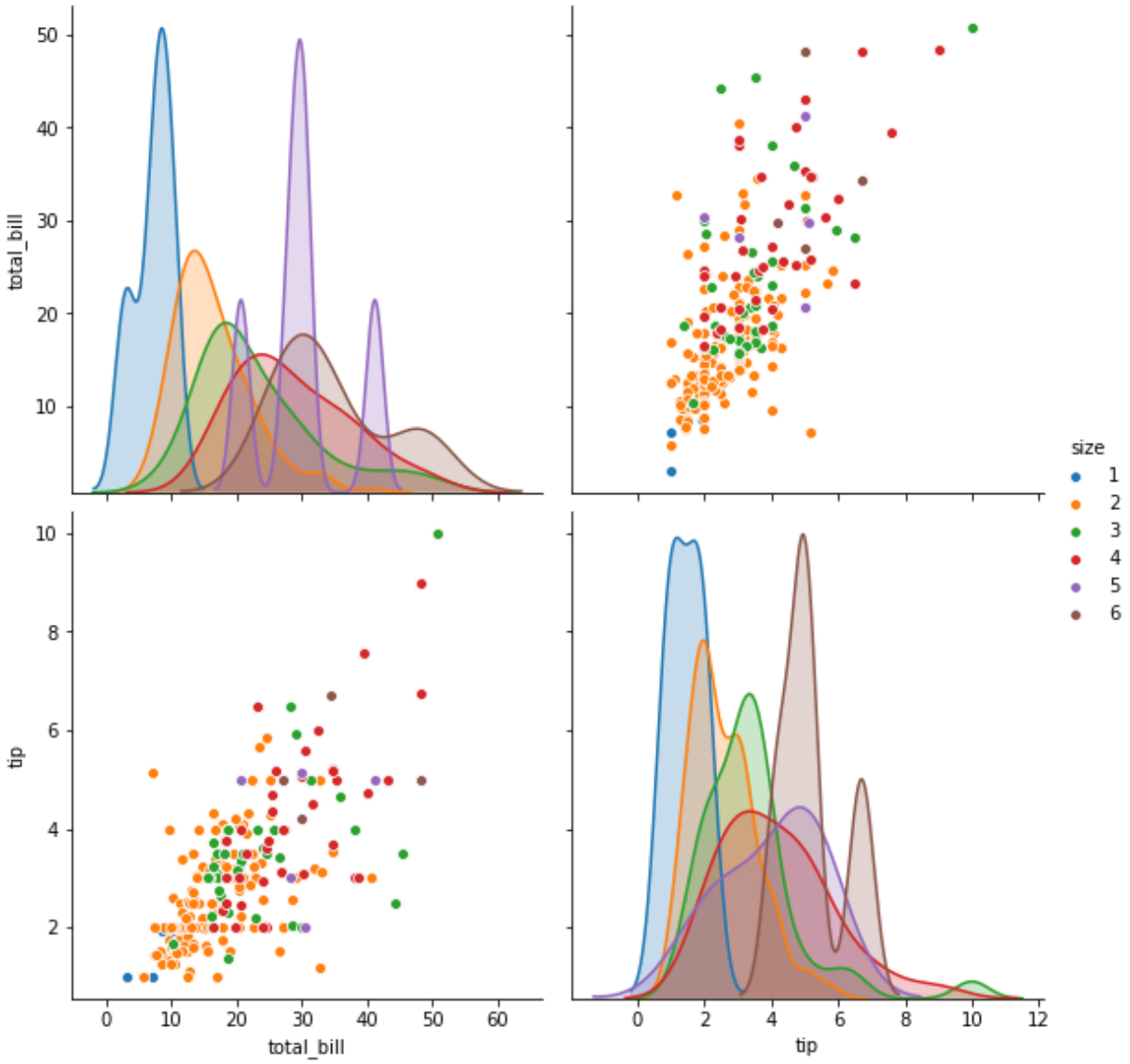
In [29]:
```
sns.pairplot(tips, hue = 'size')
```

Out[29]: <seaborn.axisgrid.PairGrid at 0x18cca7ecc70>



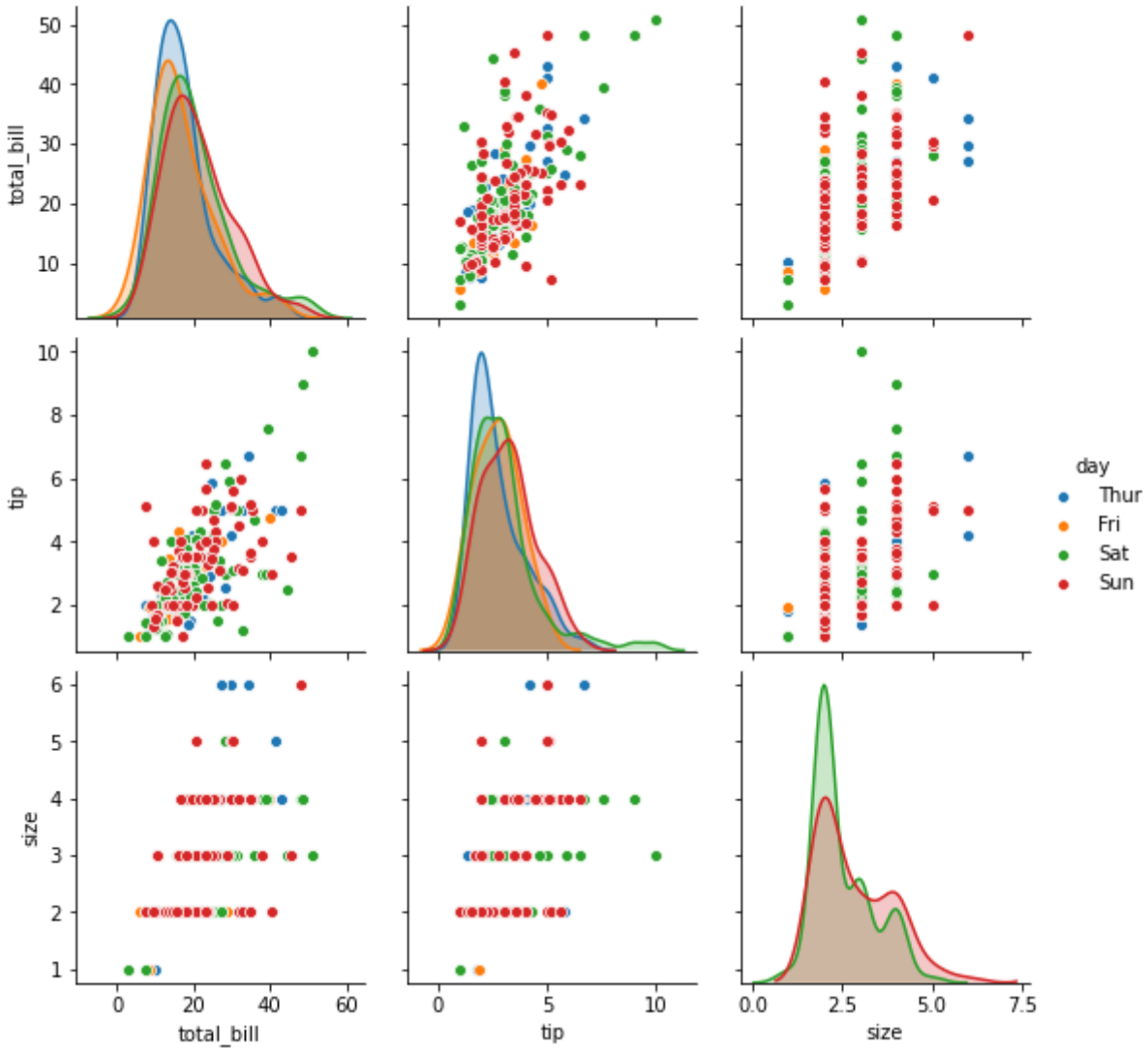
In [30]:
```
sns.pairplot(tips, hue = 'size', height = 4)
```

Out[30]: <seaborn.axisgrid.PairGrid at 0x18ccab4af70>



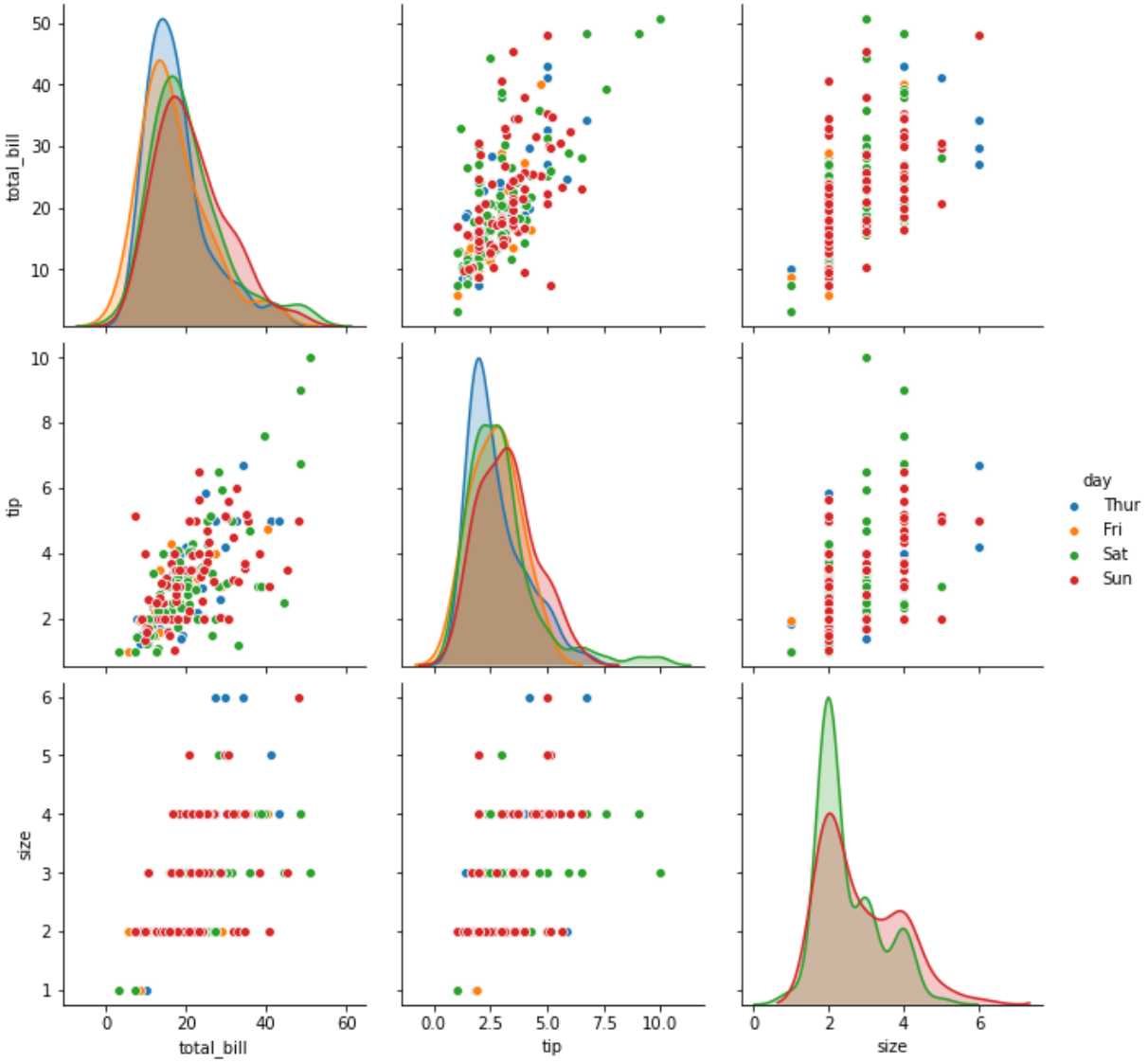
In [31]:
```
sns.pairplot(tips, hue = 'day')
```

Out[31]: <seaborn.axisgrid.PairGrid at 0x18ccab46c70>



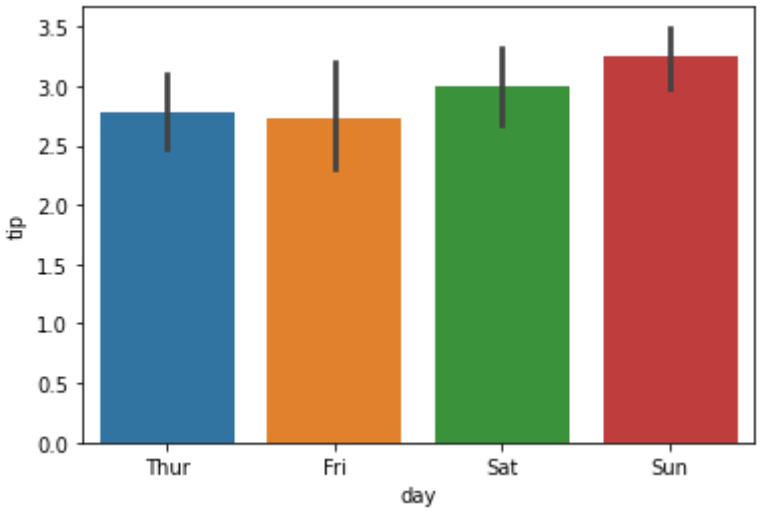
In [32]:
```
sns.pairplot(tips, hue = 'day', height = 3)
```

Out[32]:
```
<seaborn.axisgrid.PairGrid at 0x18ccb68b670>
```



In [33]:
```
sns.barplot(x = 'day', y = 'tip', data = tips)
```

Out[33]: <matplotlib.axes._subplots.AxesSubplot at 0x18cca7ce2b0>

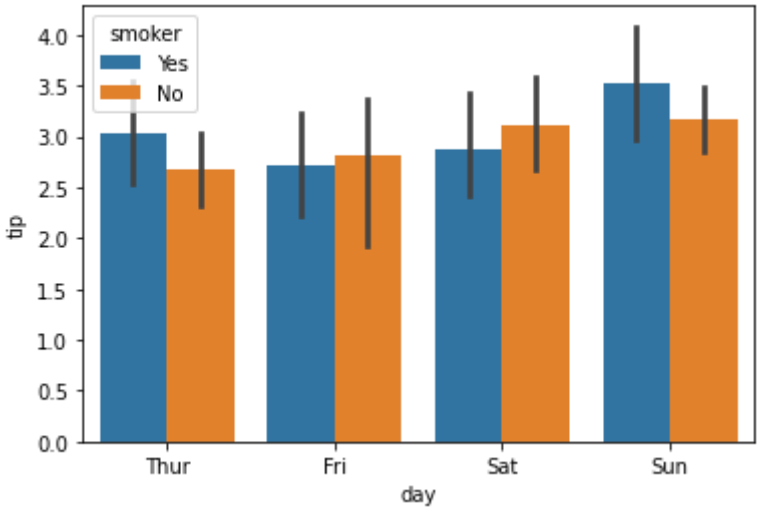


In [34]:
```
tips['day'].value_counts()
```

Out[34]:
```
Sat     87
Sun     76
Thur    62
Fri     19
Name: day, dtype: int64
```

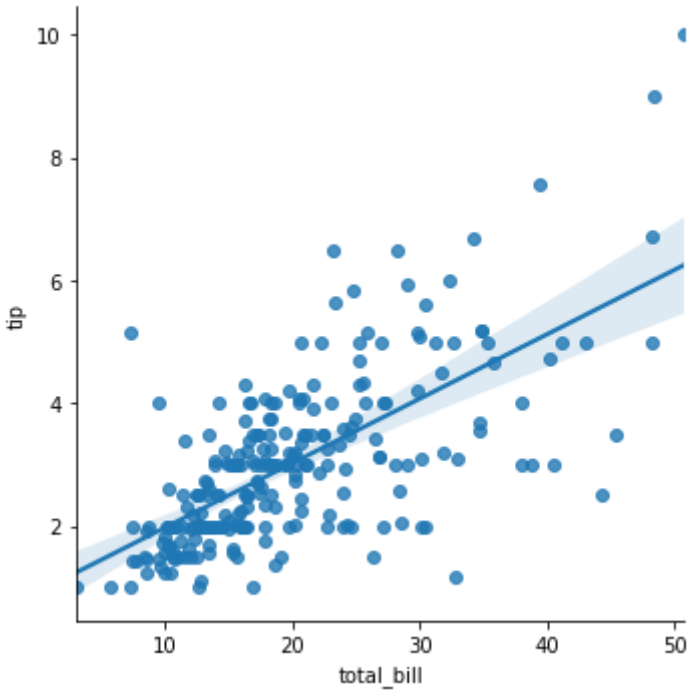
In [35]:
```
sns.barplot(x = 'day', y = 'tip', hue = 'smoker', data = tips)
```

Out[35]: <matplotlib.axes._subplots.AxesSubplot at 0x18ccbfaf4c0>



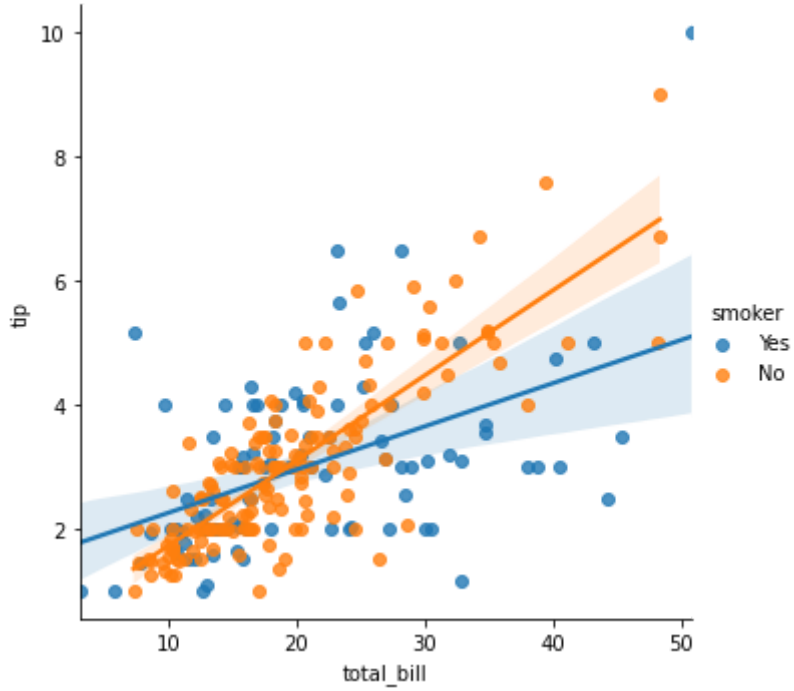
In [36]:
```
sns.lmplot(x = 'total_bill', y = 'tip', data = tips)
```

Out[36]: <seaborn.axisgrid.FacetGrid at 0x18cccfe8c10>



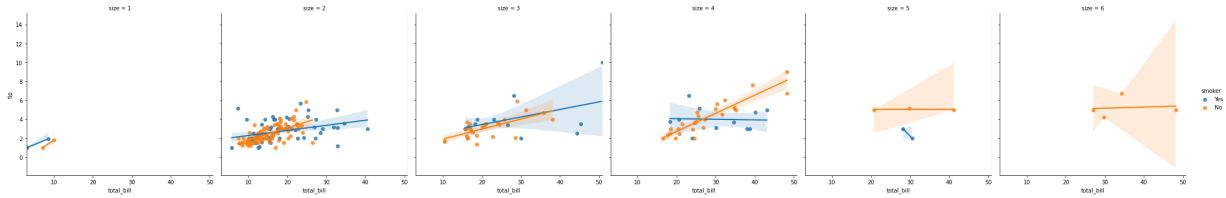
In [37]:
```
sns.lmplot(x = 'total_bill', y = 'tip', hue = 'smoker', data = tips)
```

Out[37]: <seaborn.axisgrid.FacetGrid at 0x18ccd054100>



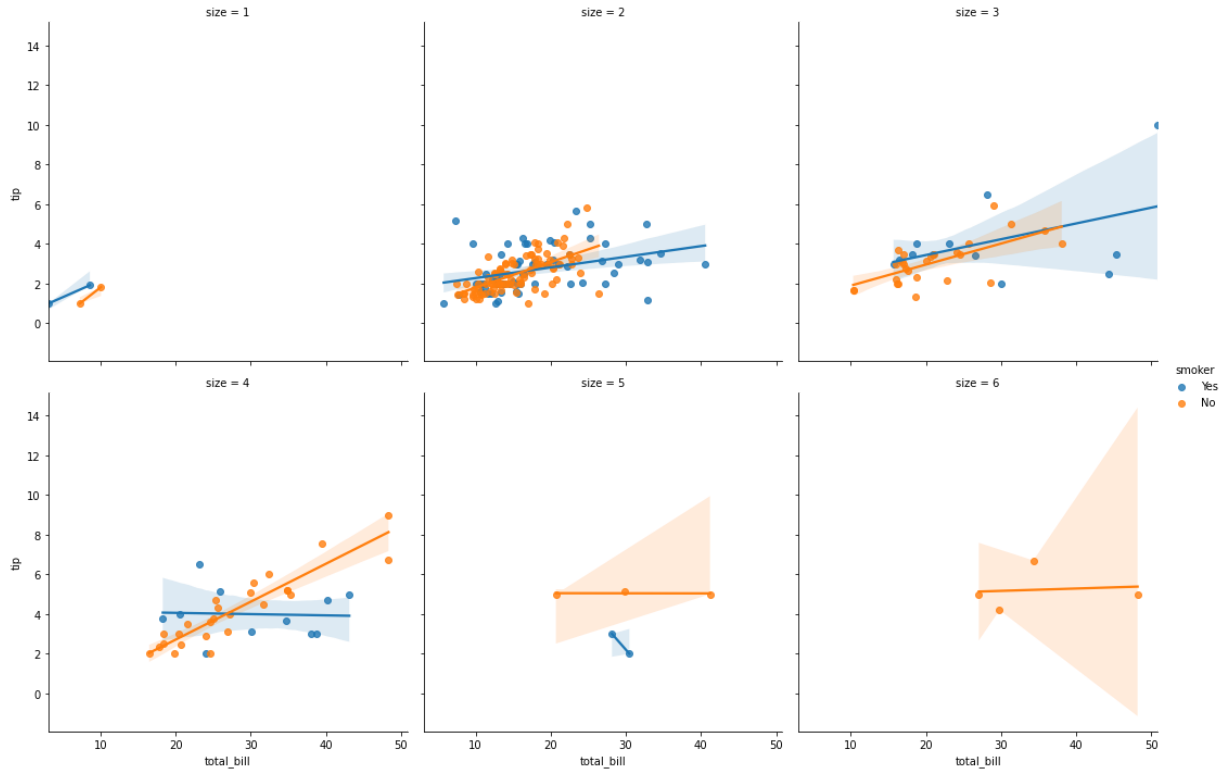
In [38]:
```
sns.lmplot(x = 'total_bill', y = 'tip', hue = 'smoker', col = 'size', data = tips)
```

Out[38]: <seaborn.axisgrid.FacetGrid at 0x18ccd0c8160>



In [39]:
```
sns.lmplot(x = 'total_bill', y = 'tip', hue = 'smoker', col = 'size', col_wrap = 3, d
```

Out[39]: <seaborn.axisgrid.FacetGrid at 0x18ccd5e7340>



In [40]:
```
pivot = tips.pivot_table(index = 'day', columns = 'smoker', values = 'tip')
```

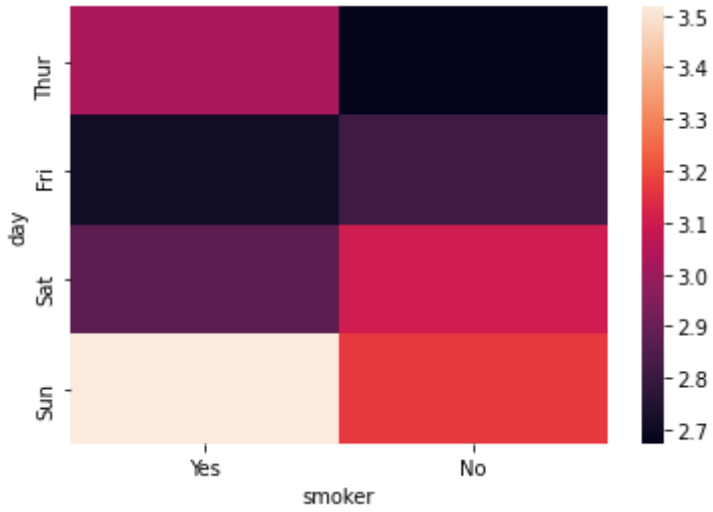
In [41]:
```
pivot
```

Out[41]:

smoker	Yes	No
day		
Thur	3.030000	2.673778
Fri	2.714000	2.812500
Sat	2.875476	3.102889
Sun	3.516842	3.167895

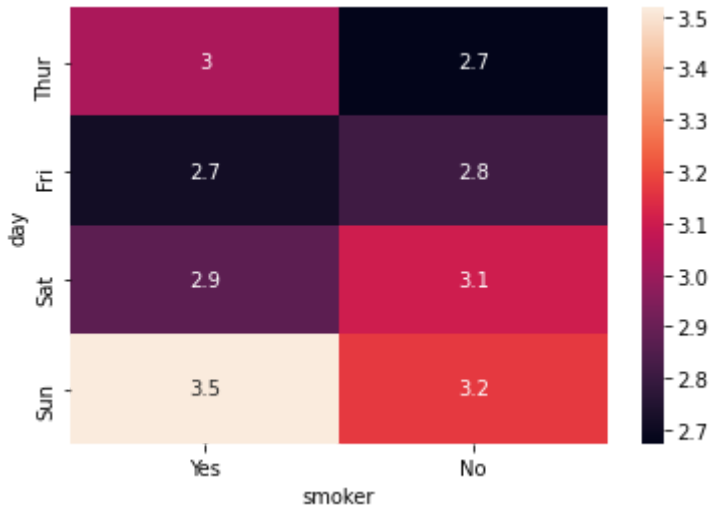
In [42]:
```
sns.heatmap(pivot)
```

Out[42]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x18cce126820>
```



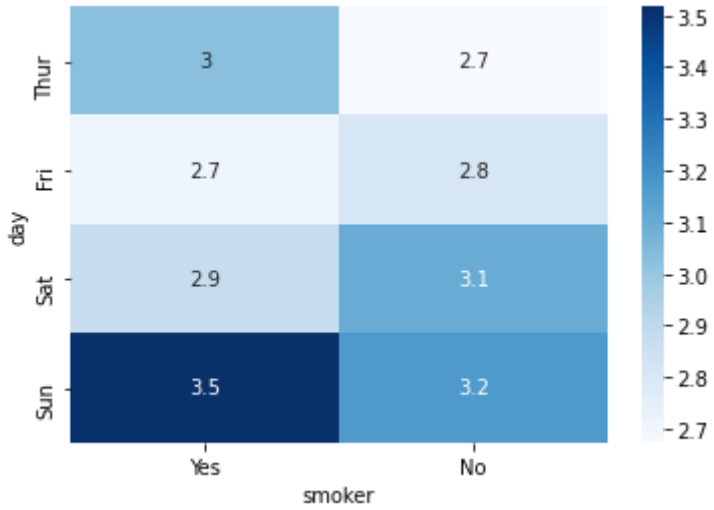
In [43]:
```
sns.heatmap(pivot, annot = True)
```

Out[43]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x18cce825760>
```



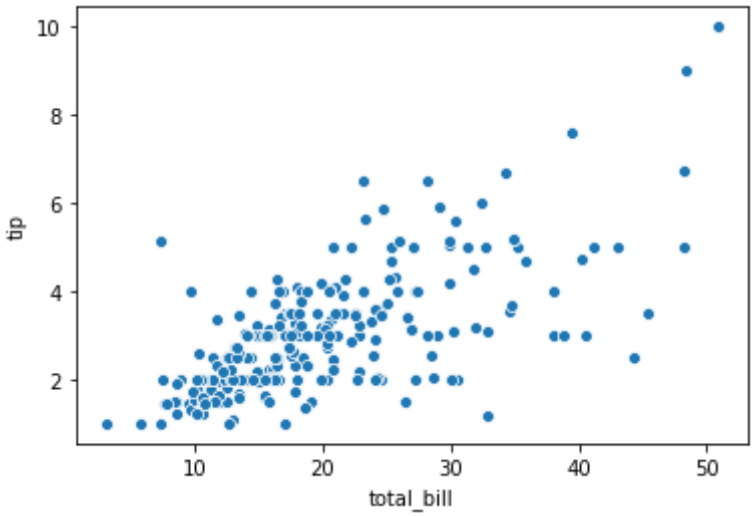
In [44]:
```
sns.heatmap(pivot, annot = True, cmap = 'Blues')
```

Out[44]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x18cce8b4e20>
```



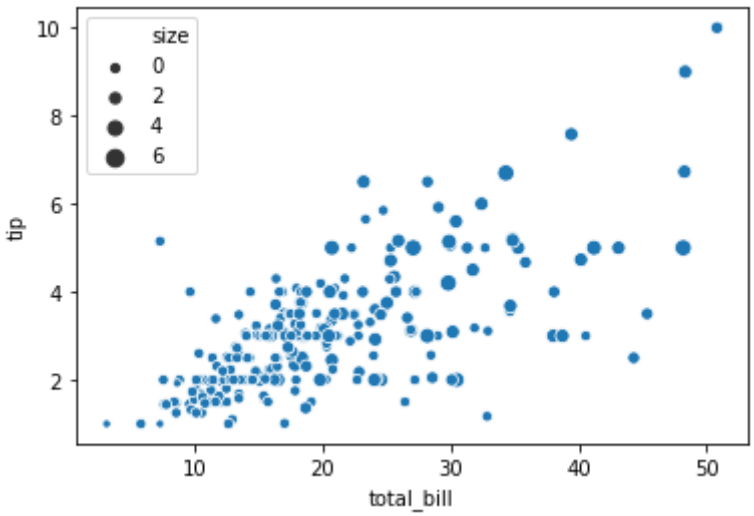
In [45]:
```
sns.scatterplot(x = 'total_bill', y = 'tip', data = tips)
```

Out[45]: <matplotlib.axes._subplots.AxesSubplot at 0x18cce8d5f10>



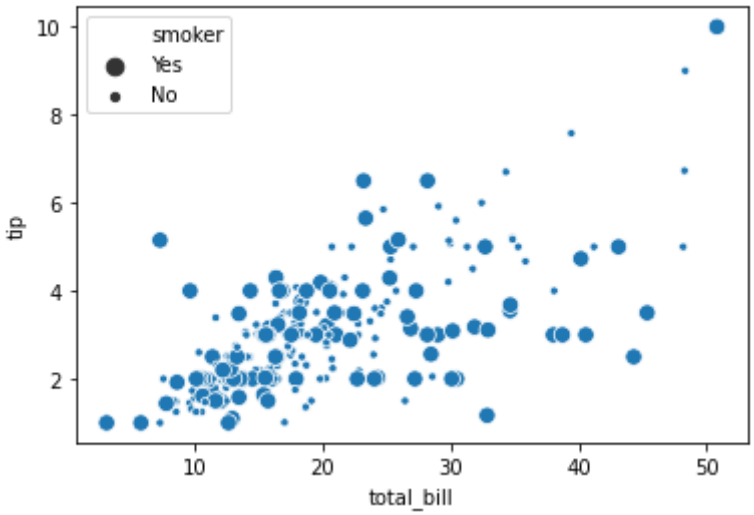
In [46]:
```
sns.scatterplot(x = 'total_bill', y = 'tip', size = 'size', data = tips)
```

Out[46]: <matplotlib.axes._subplots.AxesSubplot at 0x18cce8aaac0>



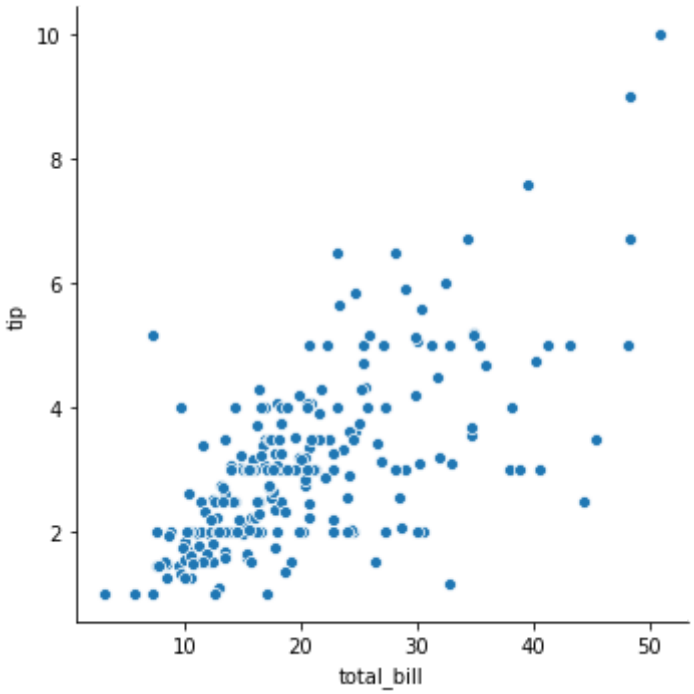
In [47]:
```
sns.scatterplot(x = 'total_bill', y = 'tip', size = 'smoker', data = tips)
```

Out[47]: <matplotlib.axes._subplots.AxesSubplot at 0x18ccea0ff40>



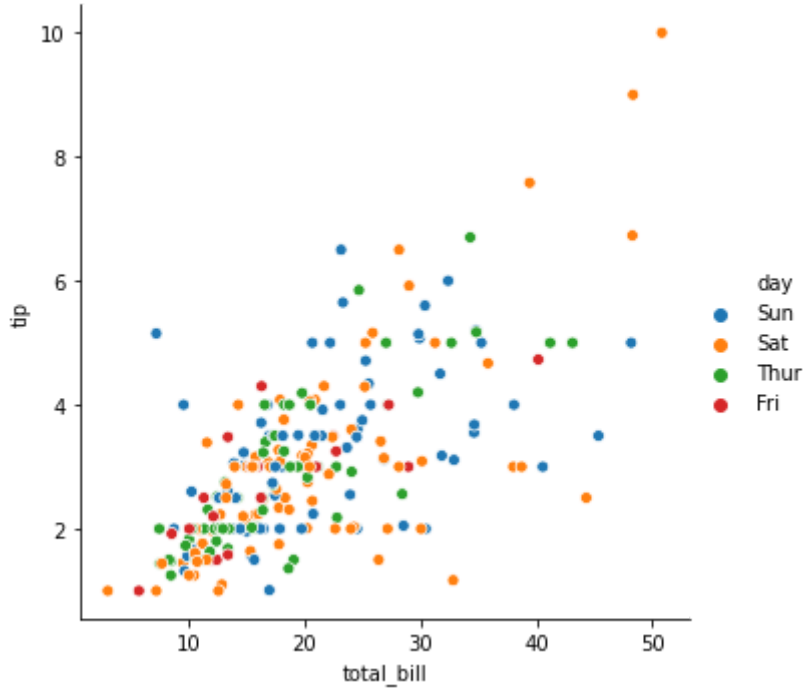
In [48]:
```
sns.relplot(x = 'total_bill', y = 'tip', data = tips)
```

Out[48]: <seaborn.axisgrid.FacetGrid at 0x18ccea4eb80>



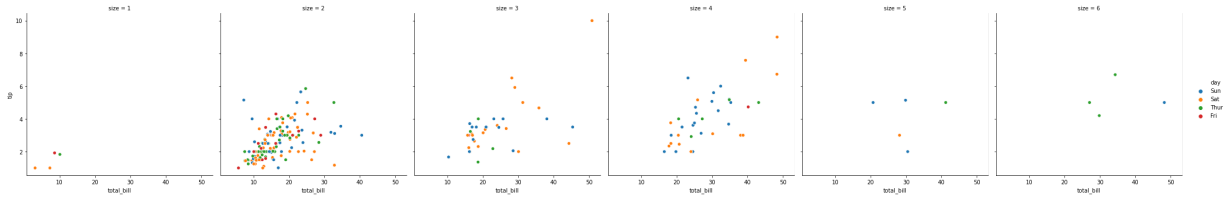
In [49]:
```
sns.relplot(x = 'total_bill', y = 'tip', hue = 'day', data = tips)
```

Out[49]: <seaborn.axisgrid.FacetGrid at 0x18cce9e03d0>



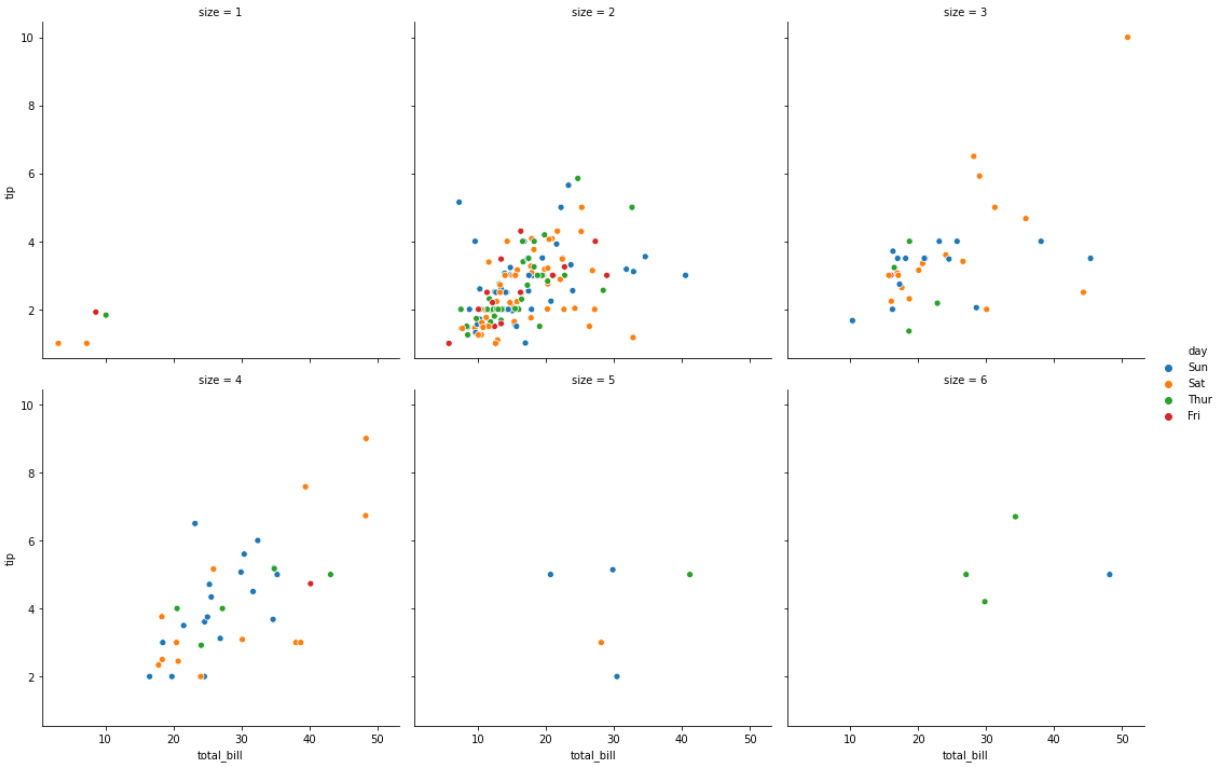
In [50]:
```
sns.relplot(x = 'total_bill', y = 'tip', hue = 'day', col = 'size', data = tips)
```

Out[50]: <seaborn.axisgrid.FacetGrid at 0x18ccd0b8bb0>



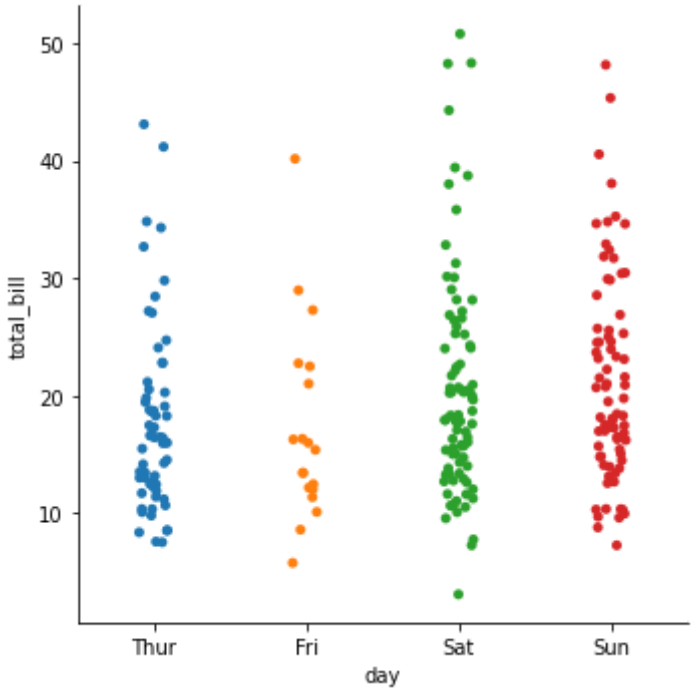
In [51]:
```
sns.relplot(x = 'total_bill', y = 'tip', hue = 'day', col = 'size', col_wrap = 3, dat
```

Out[51]: <seaborn.axisgrid.FacetGrid at 0x18ccb794760>



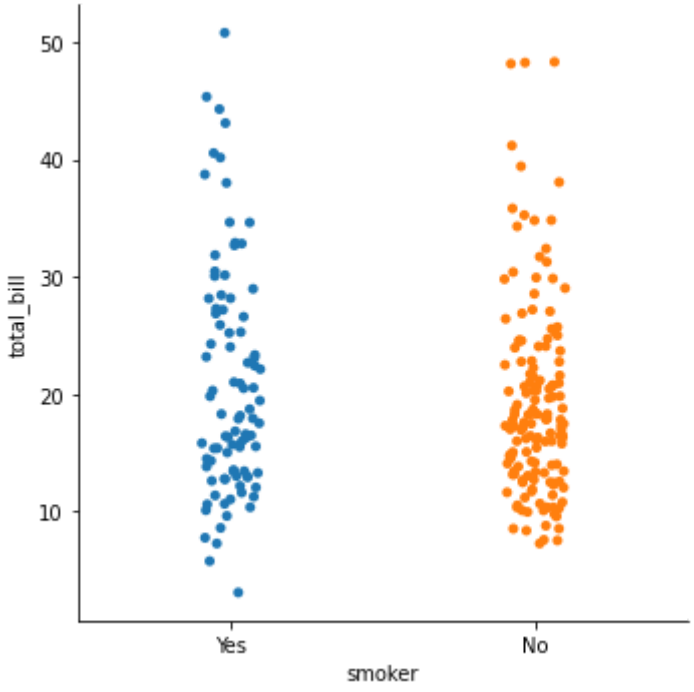
In [52]:
```
sns.catplot(x = 'day', y = 'total_bill', data = tips)
```

Out[52]: <seaborn.axisgrid.FacetGrid at 0x18ccb794ca0>



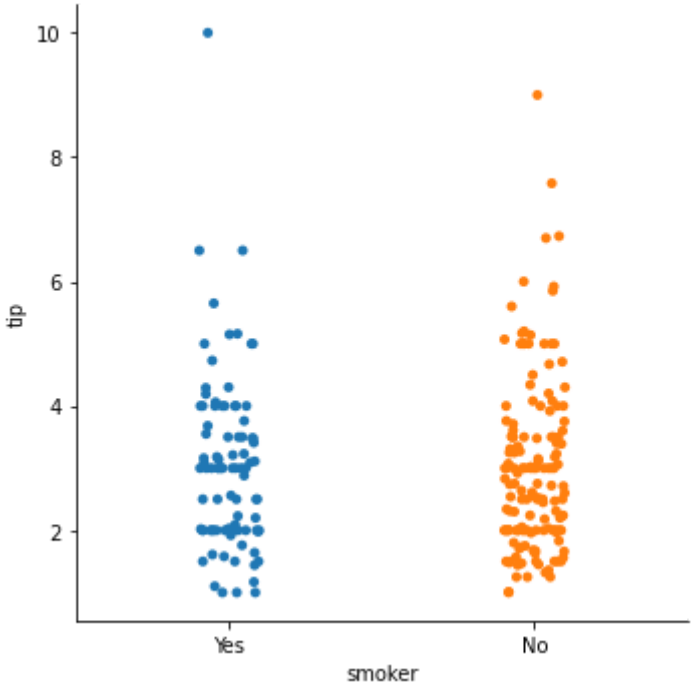
In [53]:
```
sns.catplot(x = 'smoker', y = 'total_bill', data = tips)
```

Out[53]: <seaborn.axisgrid.FacetGrid at 0x18ccff65d30>



In [54]:
```
sns.catplot(x = 'smoker', y = 'tip', data = tips)
```

Out[54]: <seaborn.axisgrid.FacetGrid at 0x18cd06a54c0>


10
8
6
4
2
tip
Yes
No
smoker