Seaborn

Visualizing statistical relationships

Statistical analysis is a process of understanding how variables in a dataset relate to each other and how those relationships depend on other variables. Visualization can be a core component of this process because, when data are visualized properly, the human visual system can see trends and patterns that indicate a relationship.

1. Numerical Data Ploting

- relplot()
- scatterplot()
- lineplot()

2. Categorical Data Ploting

- catplot()
- boxplot()
- stripplot()
- swarmplot()
- etc...

3. Visualizing Distribution of the Data

- distplot()
- kdeplot()
- jointplot()
- rugplot()

4. Linear Regression and Relationship

- regplot()
- lmplot()

5. Controlling Ploted Figure Aesthetics

- figure styling
- axes styling
- color palettes
- etc..

In [1]:

```
import seaborn as sns
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

In [ ]:

```

```

In [3]:

```
sns.set(style = 'darkgrid')
```

In [13]:

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

Out[13]:

	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

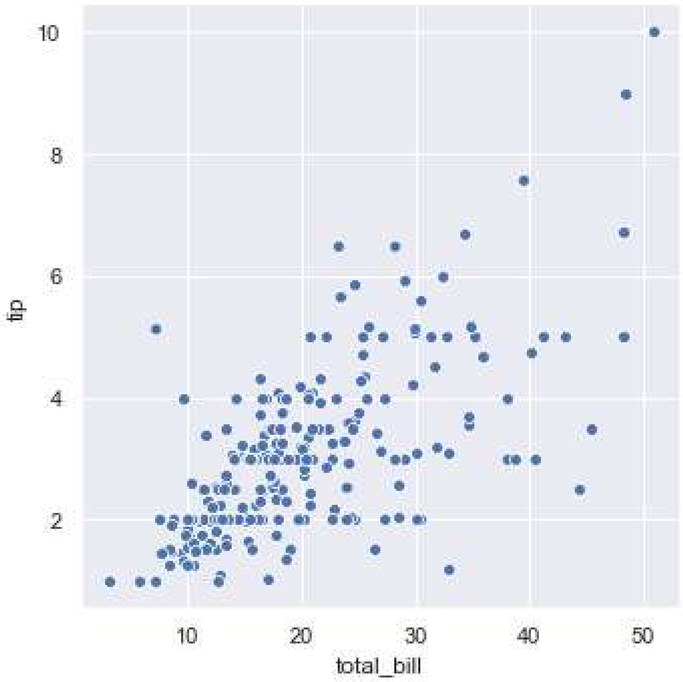
244 rows × 7 columns

In [14]:

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

Out[14]:

```
<seaborn.axisgrid.FacetGrid at 0x2078337da60>
```



In [ ]:

```

```

In [15]:

```
tips['smoker'].value_counts()
```

Out[15]:

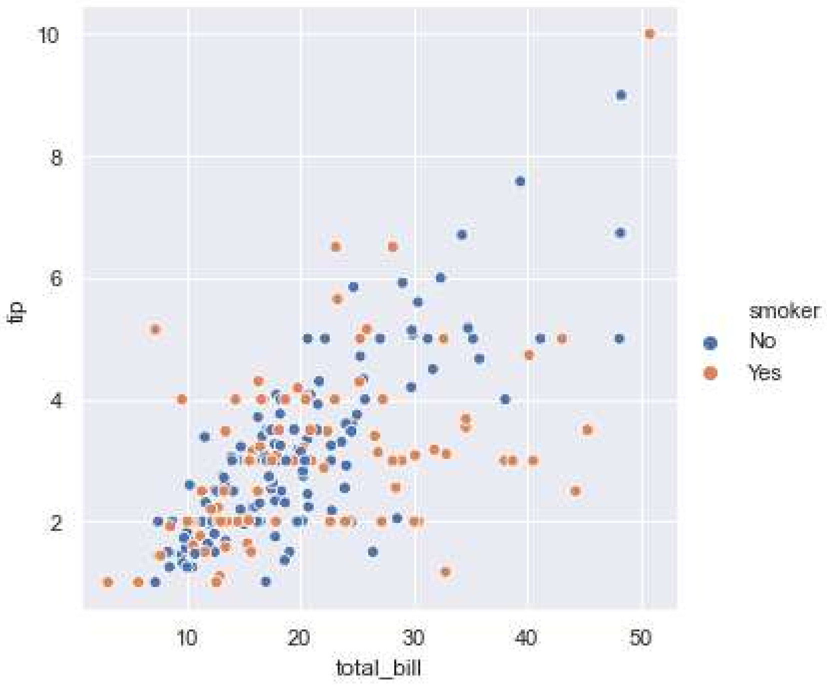
```
No     151
Yes     93
Name: smoker, dtype: int64
```

In [16]:

```
sns.relplot(x = 'total_bill', y = 'tip', data = tips, hue = 'smoker')
```

Out[16]:

```
<seaborn.axisgrid.FacetGrid at 0x207833414c0>
```

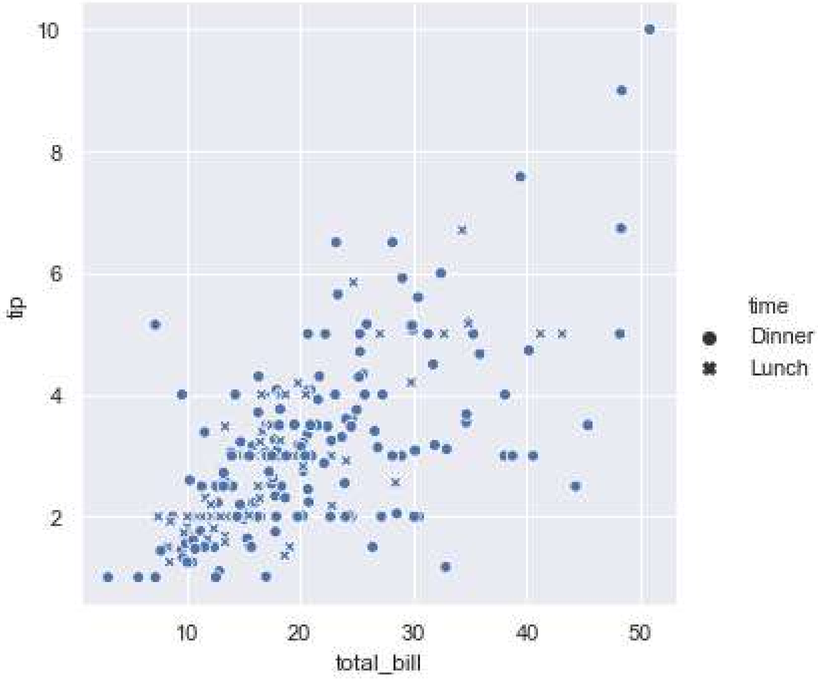


In [17]:

```
sns.relplot(x = 'total_bill', y = 'tip', data = tips, style = 'time')
```

Out[17]:

```
<seaborn.axisgrid.FacetGrid at 0x2078337d7c0>
```

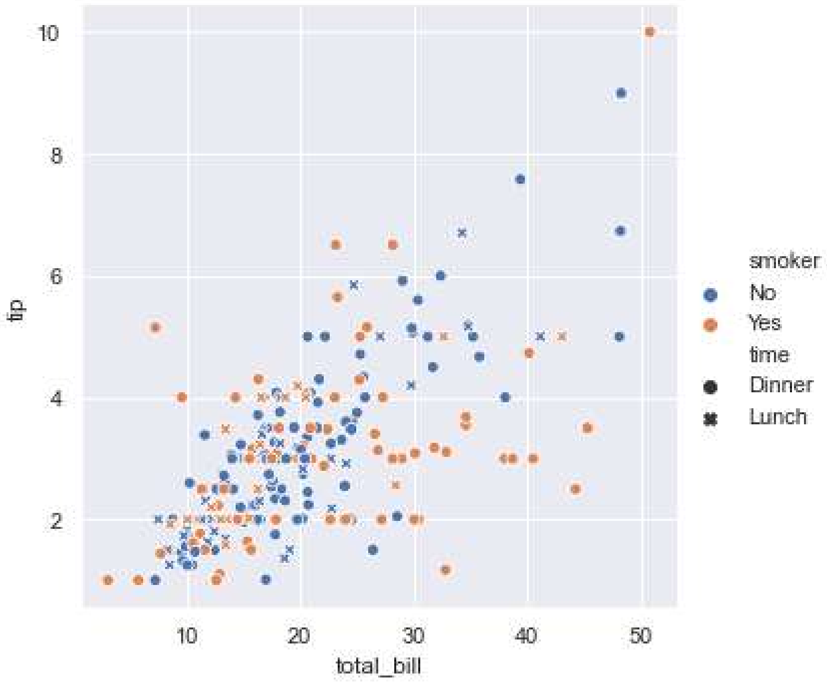


In [18]:

```
sns.relplot(x = 'total_bill', y = 'tip', data = tips, hue = 'smoker', style = 'time')
```

Out[18]:

```
<seaborn.axisgrid.FacetGrid at 0x20785548760>
```



In [19]:

```
tips
```

Out[19]:

	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

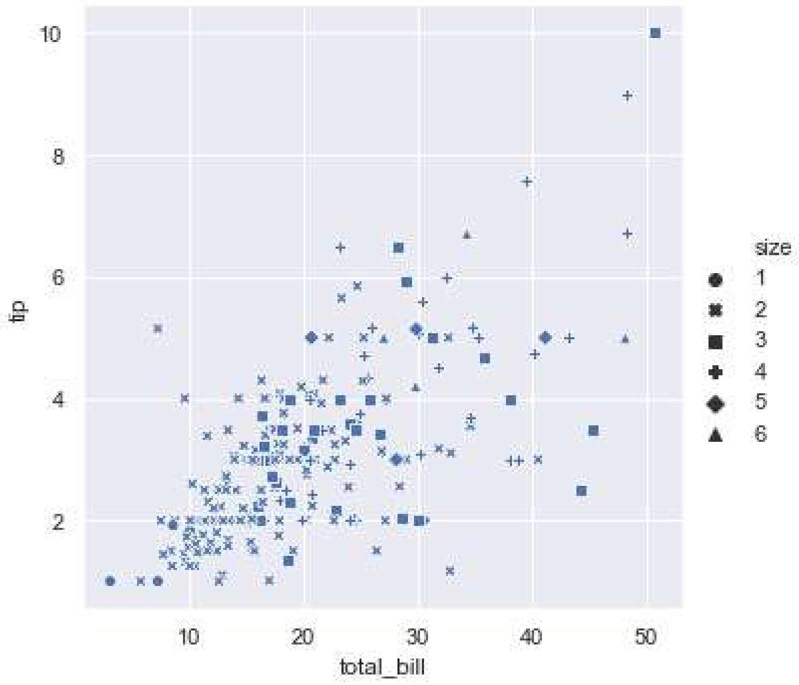
244 rows × 7 columns

In [20]:

```
sns.relplot(x = 'total_bill', y = 'tip', style = 'size', data = tips)
```

Out[20]:

```
<seaborn.axisgrid.FacetGrid at 0x207855c7310>
```



In [ ]:

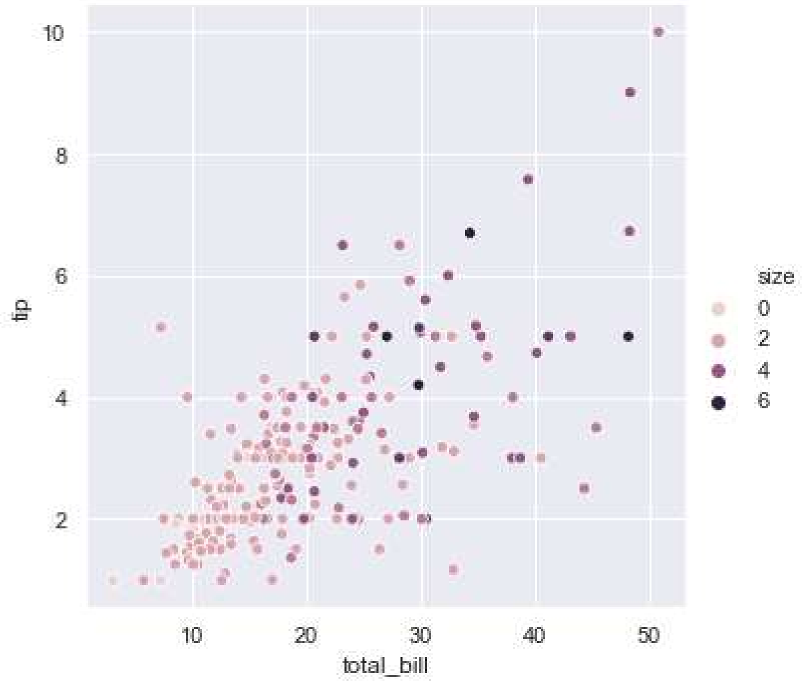
```

```

In [21]:

```
sns.relplot(x = 'total_bill', y = 'tip', hue = 'size', data = tips)
```

Out[21]:

```
<seaborn.axisgrid.FacetGrid at 0x20785537e20>
```

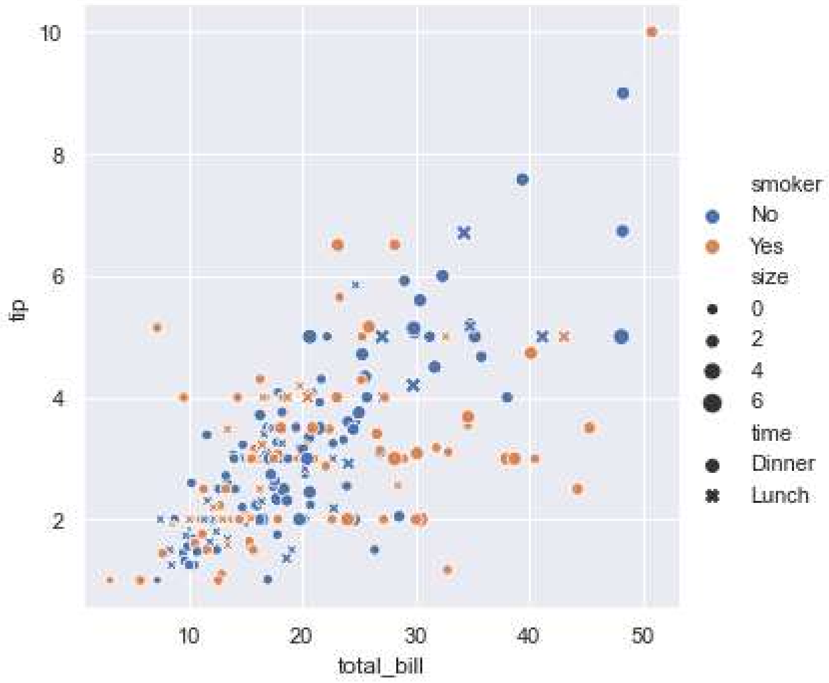


In [22]:

```
sns.relplot(x = 'total_bill', y = 'tip', data = tips,hue = 'smoker', style = 'time', si
```

Out[22]:

```
<seaborn.axisgrid.FacetGrid at 0x207856c6d60>
```



In [ ]:

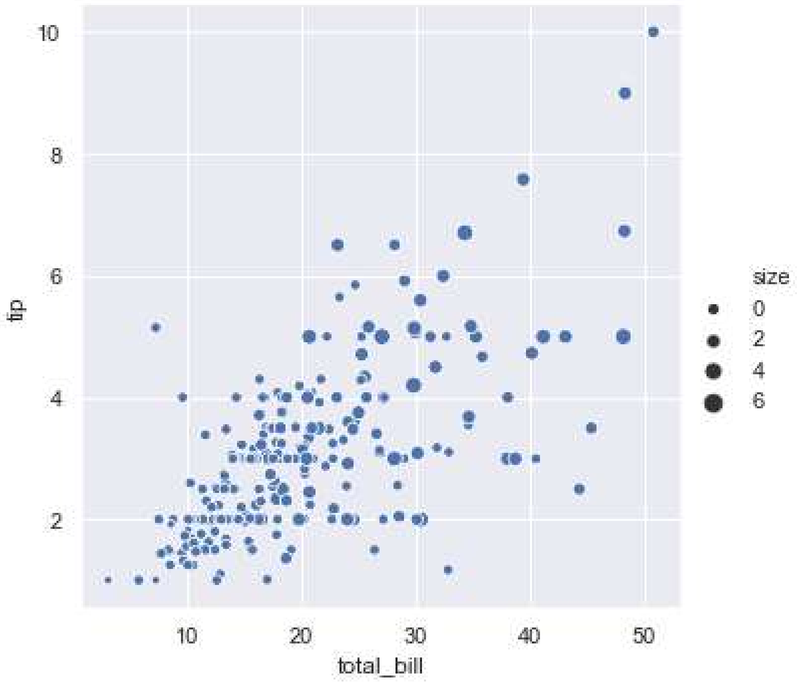
```

```

In [23]:

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

Out[23]:

```
<seaborn.axisgrid.FacetGrid at 0x207854e4df0>
```



In [ ]:

```

```

In [24]:

```
from numpy.random import randn
```

In [25]:

```
df = pd.DataFrame(dict(time = np.arange(500), value = randn(500).cumsum()))
```

In [26]:

```
df.head()
```

Out[26]:

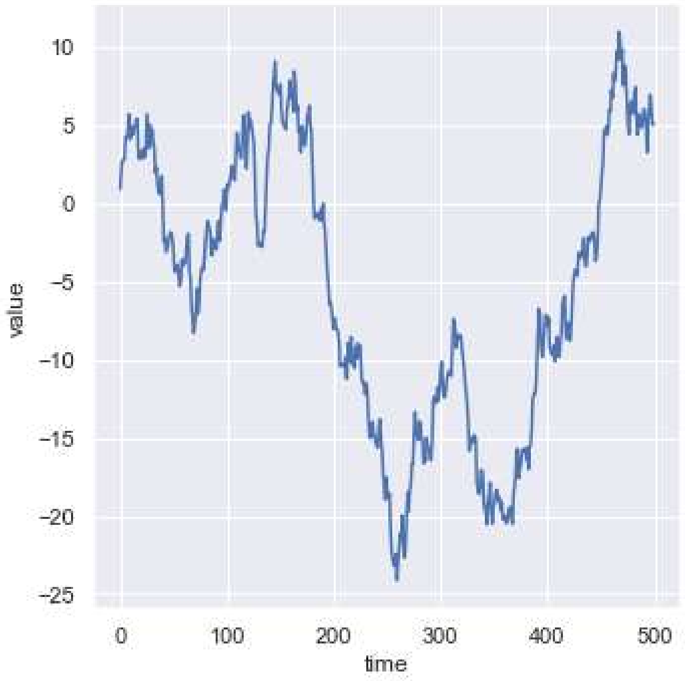
	time	value
0	0	0.903389
1	1	2.143436
2	2	2.760180
3	3	2.766698
4	4	2.834482

In [30]:

```python
sns.relplot(x = 'time', y = 'value', kind = 'line', data = df, sort = True)  # sorting
```

Out[30]:

```
<seaborn.axisgrid.FacetGrid at 0x20785835340>
```



In [ ]:

```python

```

In [31]:

```python
df = pd.DataFrame(randn(500, 2).cumsum(axis = 0), columns = ['time', 'value'])
```

In [32]:

```python
df.head()
```

Out[32]:

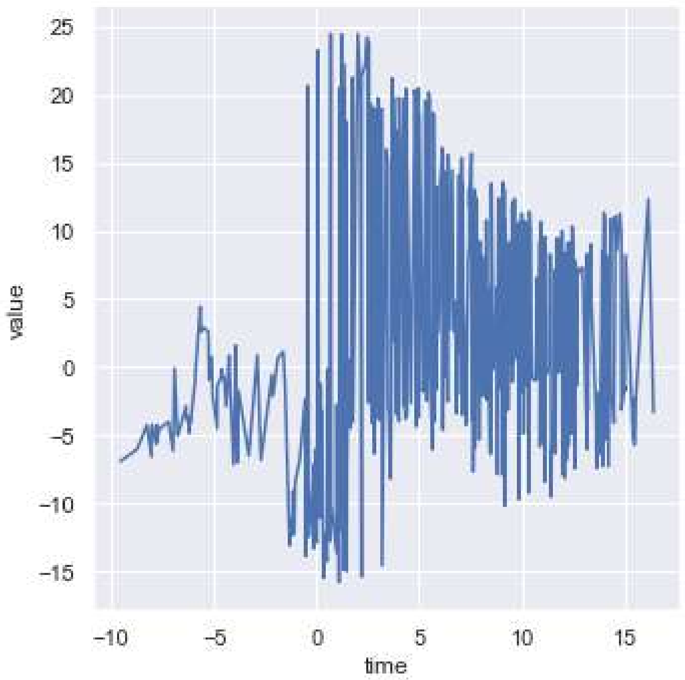
	time	value
0	0.549519	-0.167365
1	1.542254	0.554378
2	1.558775	-1.473491
3	2.983069	-1.855094
4	2.571061	-0.534342

In [33]:

```
sns.relplot(x = 'time', y = 'value', kind = 'line', data = df, sort = True)
```

Out[33]:

```
<seaborn.axisgrid.FacetGrid at 0x20785483cd0>
```



In [ ]:

```

```

In [34]:

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

Out[34]:

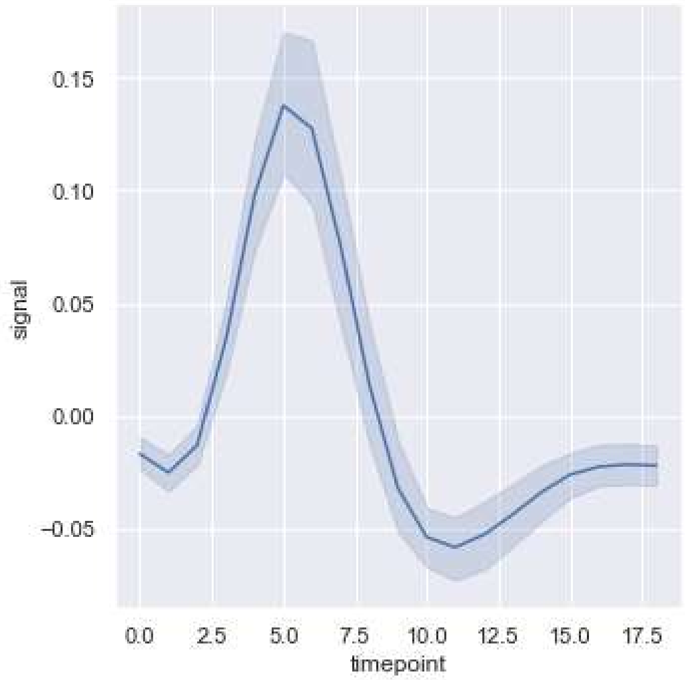
	subject	timepoint	event	region	signal
0	s13	18	stim	parietal	-0.017552
1	s5	14	stim	parietal	-0.080883
2	s12	18	stim	parietal	-0.081033
3	s11	18	stim	parietal	-0.046134
4	s10	18	stim	parietal	-0.037970

In [35]:

```
sns.relplot(x = 'timepoint', y = 'signal', kind = 'line', data = fmri)
```

Out[35]:

```
<seaborn.axisgrid.FacetGrid at 0x2078572a250>
```

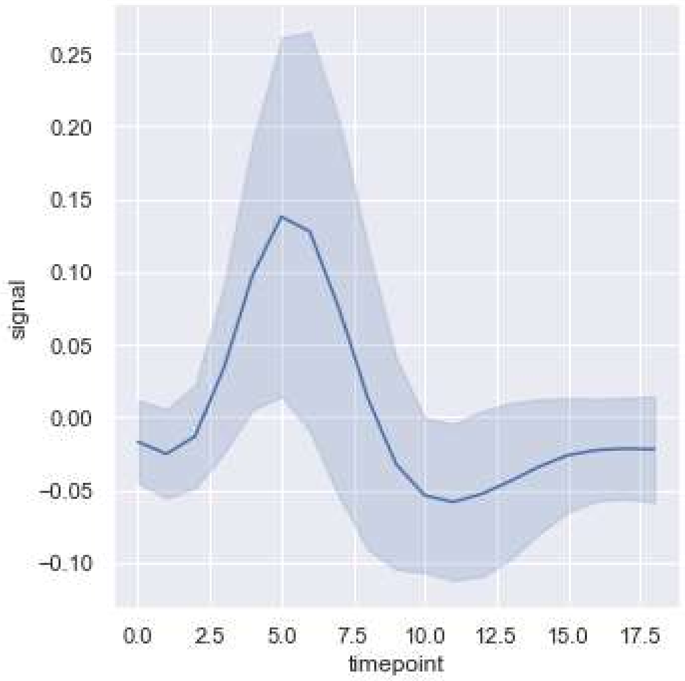


In [38]:

```
sns.relplot(x = 'timepoint', y = 'signal', kind = 'line', data = fmri, ci = 'sd')    # s
```

Out[38]:

```
<seaborn.axisgrid.FacetGrid at 0x207869a9af0>
```



In [ ]:

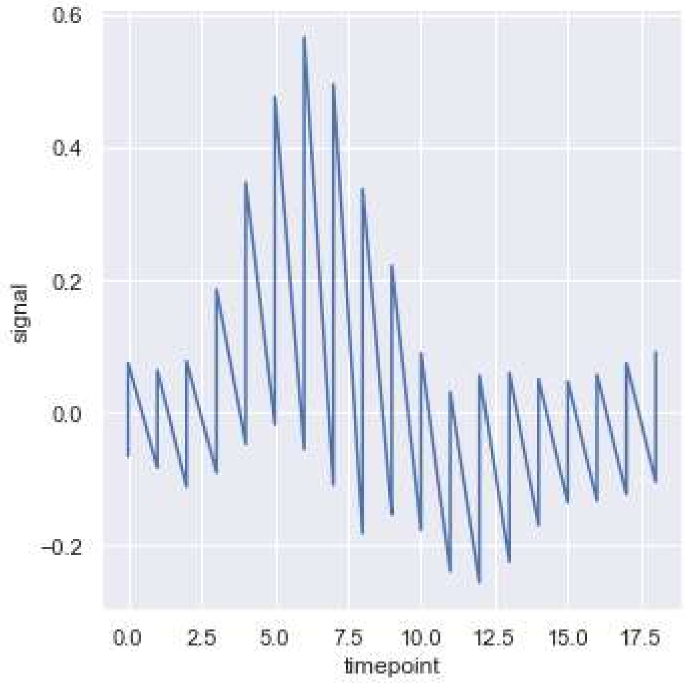
```

```

In [37]:

```
sns.relplot(x = 'timepoint', y = 'signal', estimator = None, kind = 'line', data = fmri
```

Out[37]:

```
<seaborn.axisgrid.FacetGrid at 0x20786970430>
```



In [ ]:

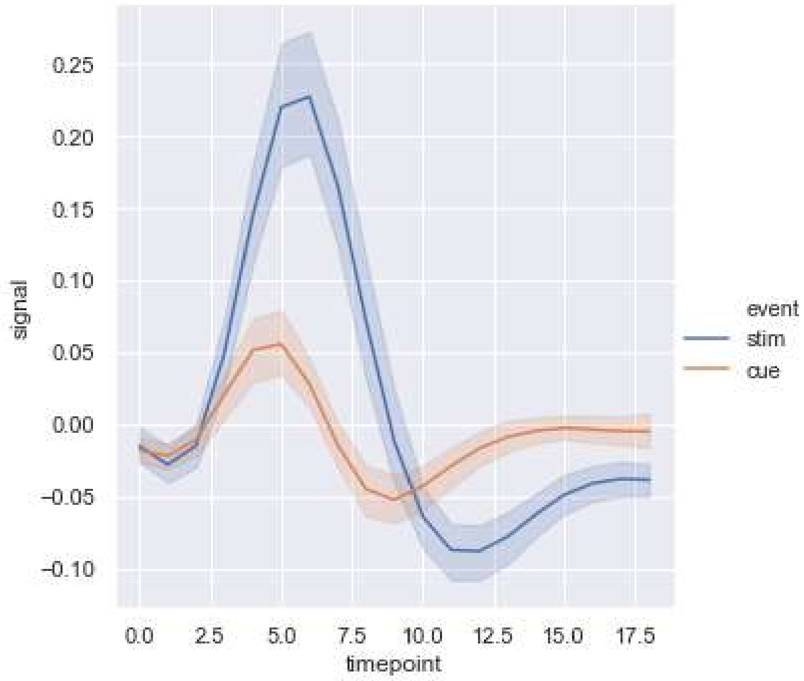
```

```

In [39]:

```
sns.relplot(x = 'timepoint', y = 'signal', hue = 'event', kind = 'line', data = fmri)
```

Out[39]:

```
<seaborn.axisgrid.FacetGrid at 0x207869a3550>
```



In [23]:

```
fmri.head()
```

Out[23]:

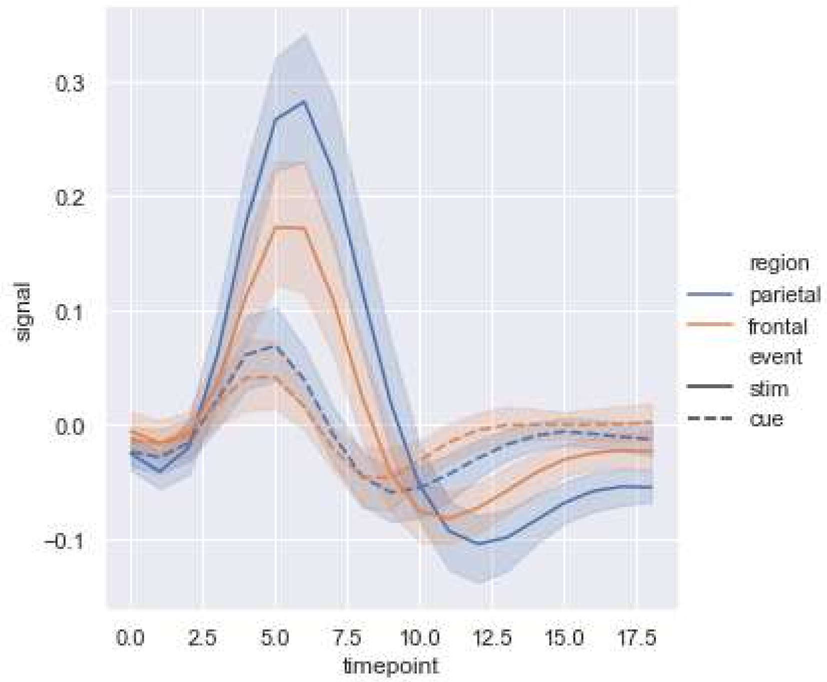
	subject	timepoint	event	region	signal
0	s13	18	stim	parietal	-0.017552
1	s5	14	stim	parietal	-0.080883
2	s12	18	stim	parietal	-0.081033
3	s11	18	stim	parietal	-0.046134
4	s10	18	stim	parietal	-0.037970

In [40]:

```
sns.relplot(x = 'timepoint', y = 'signal', hue = 'region', style = 'event', kind = 'li
```

Out[40]:

```
<seaborn.axisgrid.FacetGrid at 0x207869a9f70>
```

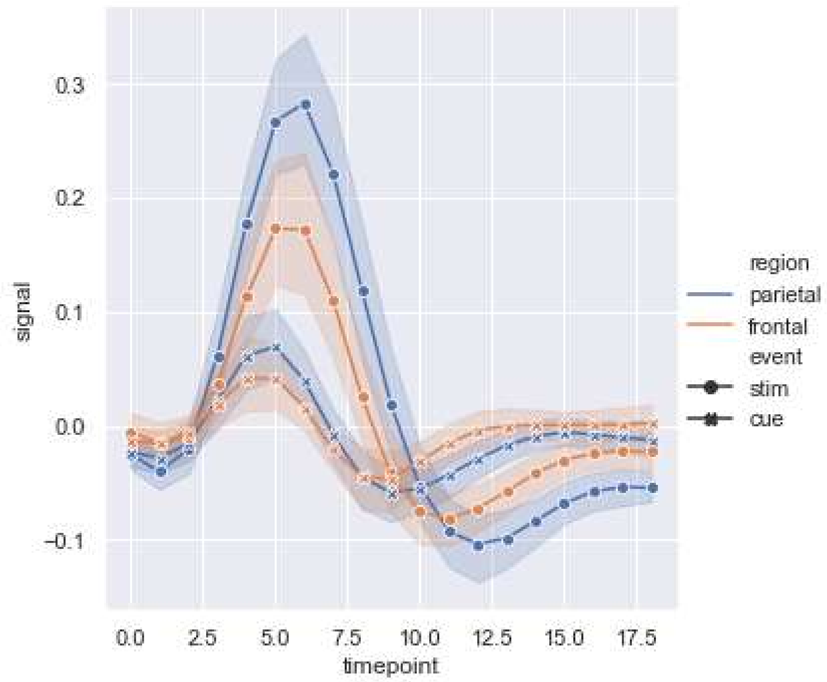


In [42]:

```
sns.relplot(x = 'timepoint', y = 'signal', hue = 'region', style = 'event', kind = 'li
```

Out[42]:

```
<seaborn.axisgrid.FacetGrid at 0x20785799490>
```

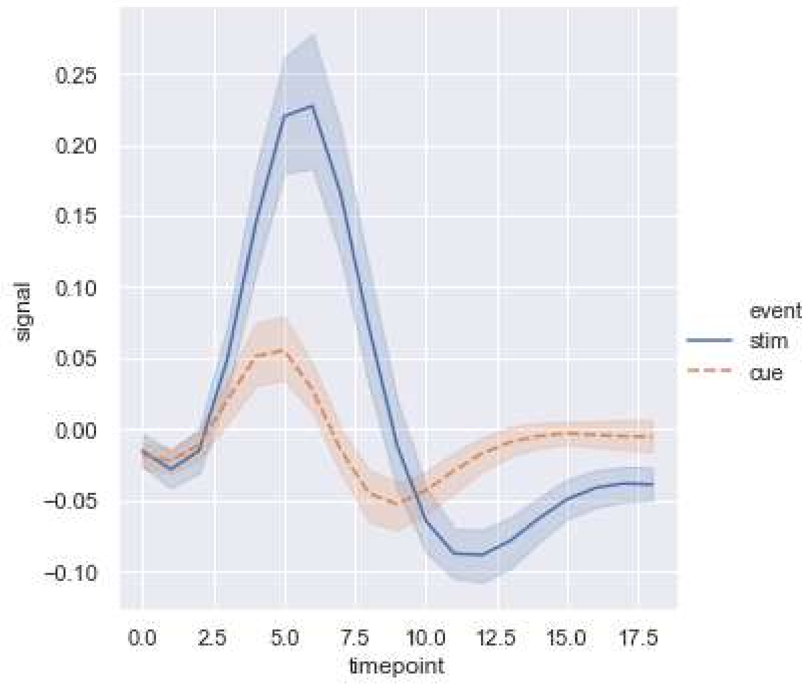


In [43]:

```
sns.relplot(x = 'timepoint', y = 'signal', hue = 'event', style = 'event', kind = 'line
```

Out[43]:

```
<seaborn.axisgrid.FacetGrid at 0x20787c34130>
```



In [44]:

```
df = pd.DataFrame(dict(time = pd.date_range('2019-06-02', periods = 500), value = randn
```

In [45]:

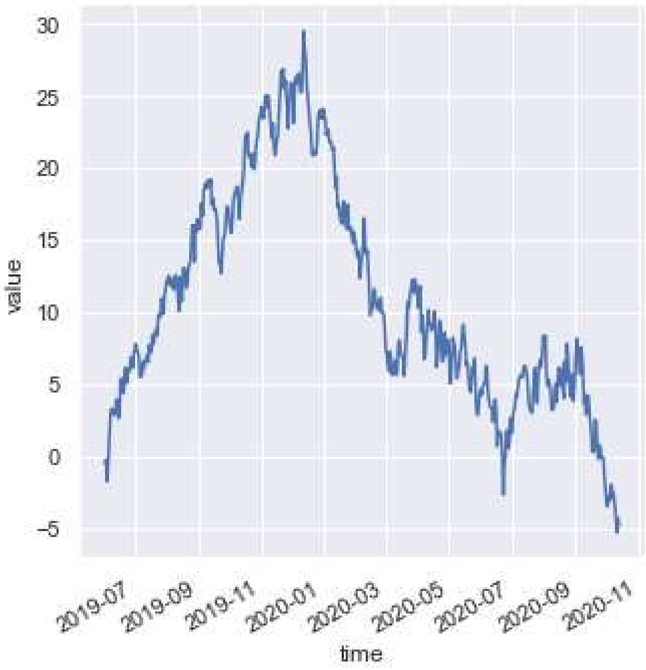
```
df.head()
```

Out[45]:

	time	value
0	2019-06-02	-0.481641
1	2019-06-03	-0.258667
2	2019-06-04	-1.755652
3	2019-06-05	-0.641708
4	2019-06-06	1.039689

In [46]:

```
g = sns.relplot(x = 'time', y = 'value', kind = 'line', data = df)
g.fig.autofmt_xdate()
```



In [47]:

```
tips.head()
```

Out[47]:

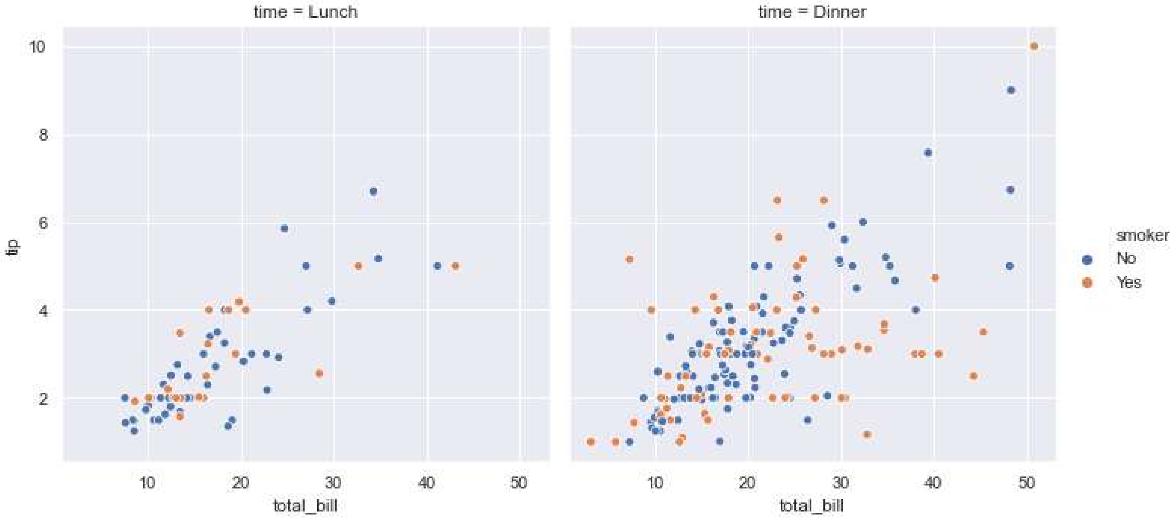
	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4

In [49]:

```
sns.relplot(x = 'total_bill', y = 'tip', hue = 'smoker', col = 'time', data = tips)
# col seperate the graph
```

Out[49]:

```
<seaborn.axisgrid.FacetGrid at 0x20787cb3c10>
```

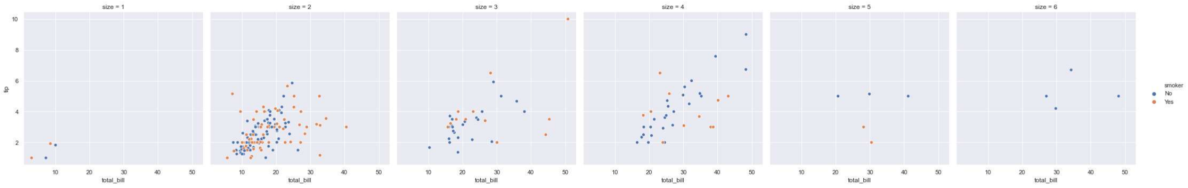


In [50]:

```
sns.relplot(x = 'total_bill', y = 'tip', hue = 'smoker', col = 'size', data = tips)
```

Out[50]:

```
<seaborn.axisgrid.FacetGrid at 0x20787e92670>
```



In [ ]:

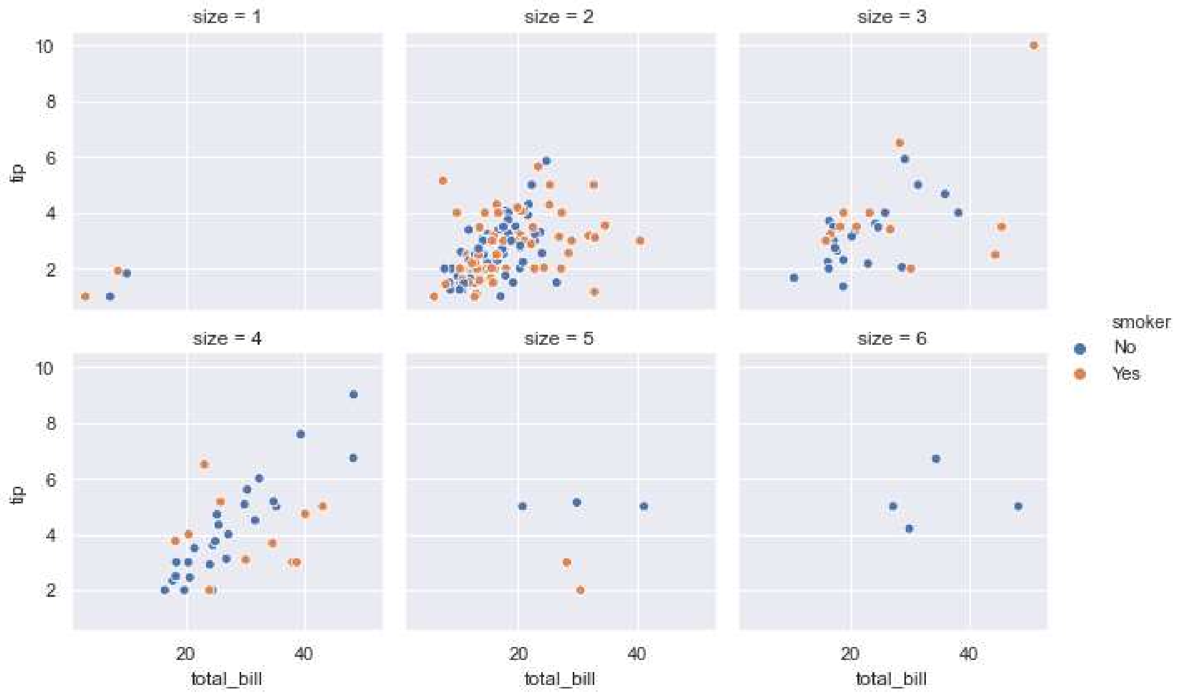
```

```

In [51]:

```
sns.relplot(x = 'total_bill', y = 'tip', hue = 'smoker', col = 'size', data = tips, col
```

Out[51]:

```
<seaborn.axisgrid.FacetGrid at 0x207886028e0>
```



In [ ]:

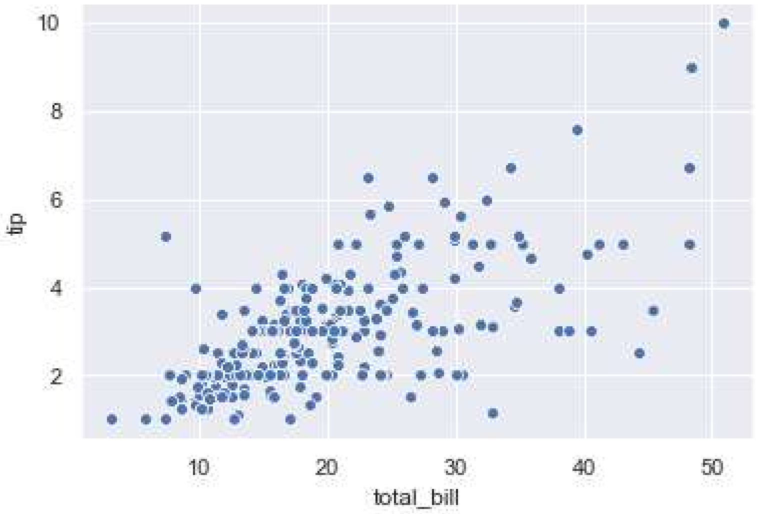
```

```

In [52]:

```
sns.scatterplot(x = 'total_bill', y = 'tip', data = tips)
```

Out[52]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x20788737730>
```



In [ ]:

```

```

In [53]:

```
fmri.head()
```

Out[53]:

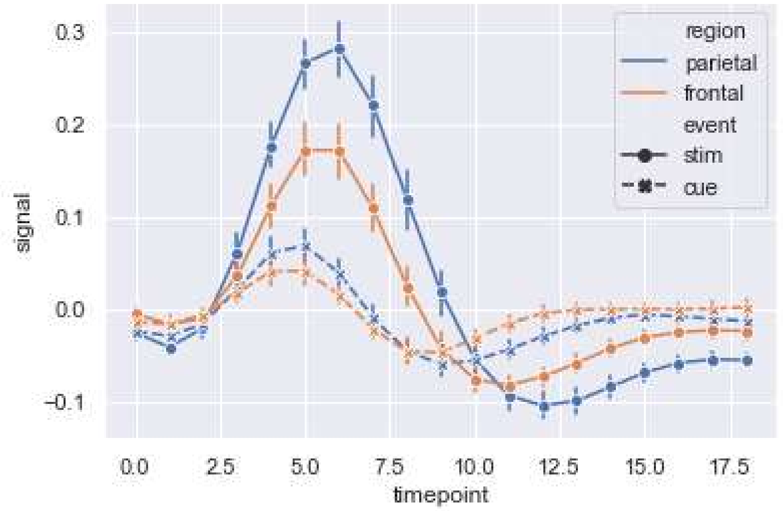
	subject	timepoint	event	region	signal
0	s13	18	stim	parietal	-0.017552
1	s5	14	stim	parietal	-0.080883
2	s12	18	stim	parietal	-0.081033
3	s11	18	stim	parietal	-0.046134
4	s10	18	stim	parietal	-0.037970

In [54]:

```
sns.lineplot(x = 'timepoint', y  = 'signal', style = 'event', hue = 'region', data = fr
```

Out[54]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x20788edb610>
```

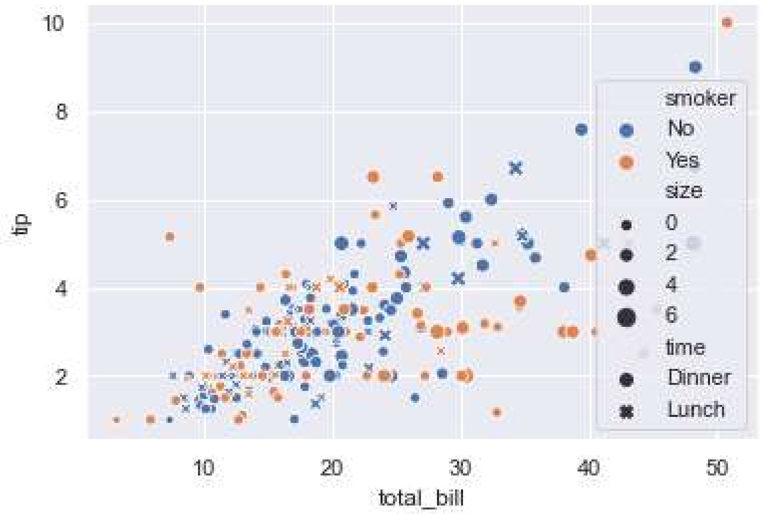


In [55]:

```
sns.scatterplot(x = 'total_bill', y = 'tip', data = tips, hue = 'smoker', size = 'size'
```

Out[55]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x20788f2faf0>
```



In [ ]:

```

```

In [56]:

```
iris = sns.load_dataset('iris')
```

In [57]:

```
iris.head()
```

Out[57]:

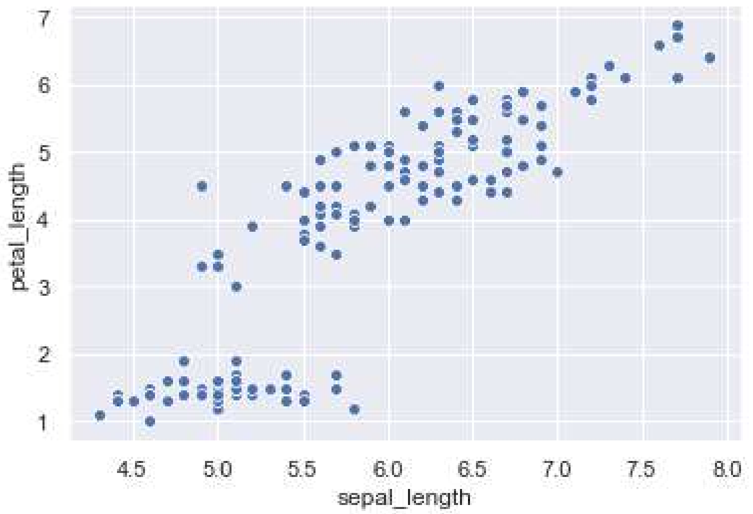
	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

In [58]:

```
sns.scatterplot(x = 'sepal_length', y = 'petal_length', data = iris)
```

Out[58]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x20787e7ebb0>
```

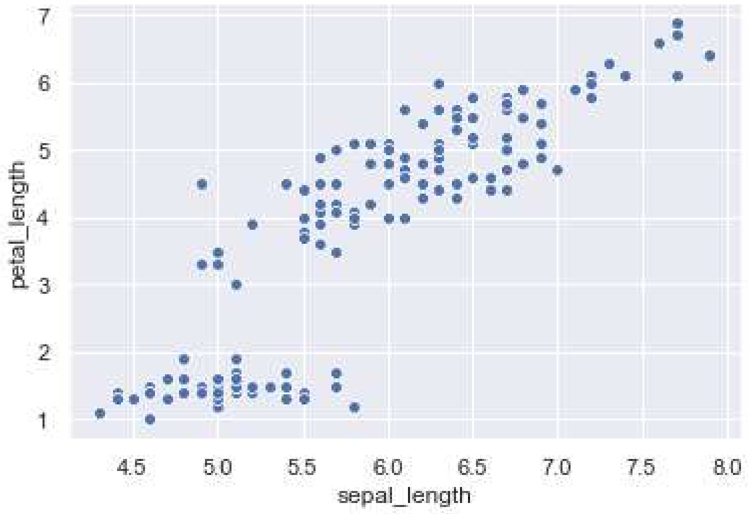


In [59]:

```
sns.scatterplot(x = iris['sepal_length'], y = iris['petal_length'])
```

Out[59]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x207890353d0>
```



2. Categorical Data Ploting

- catplot()
- boxplot()
- stripplot()
- swarmplot()
- etc...

In [60]:

```
tips.head()
```

Out[60]:

	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 [61]:

```
titanic = sns.load_dataset('titanic')
```

In [62]:

```
titanic.head()
```

Out[62]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male
0	0	3	male	22.0	1	0	7.2500	S	Third	man	True
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
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

In [63]:

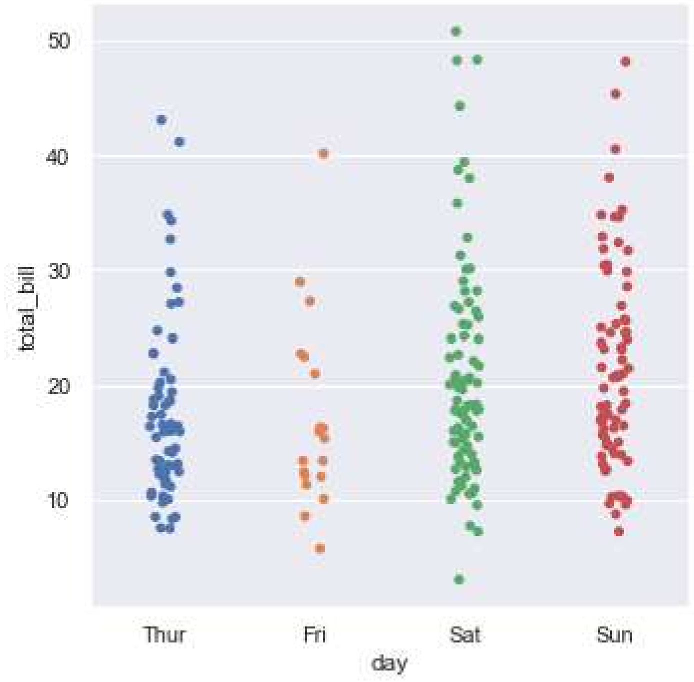
```
#catplot()
```

In [64]:

```
sns.catplot(x = 'day', y = 'total_bill', data = tips)
```

Out[64]:

```
<seaborn.axisgrid.FacetGrid at 0x20788606c70>
```

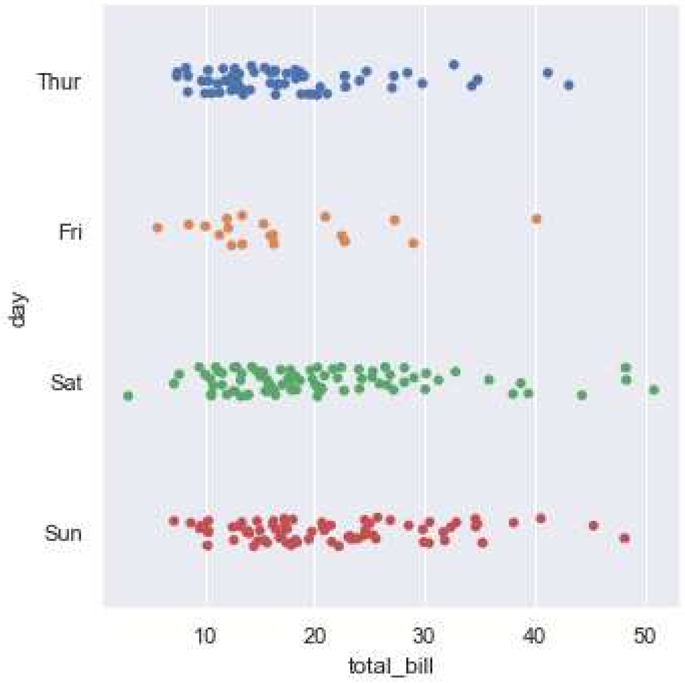


In [65]:

```
sns.catplot(y = 'day', x = 'total_bill', data = tips)
```

Out[65]:

```
<seaborn.axisgrid.FacetGrid at 0x2078a095ca0>
```

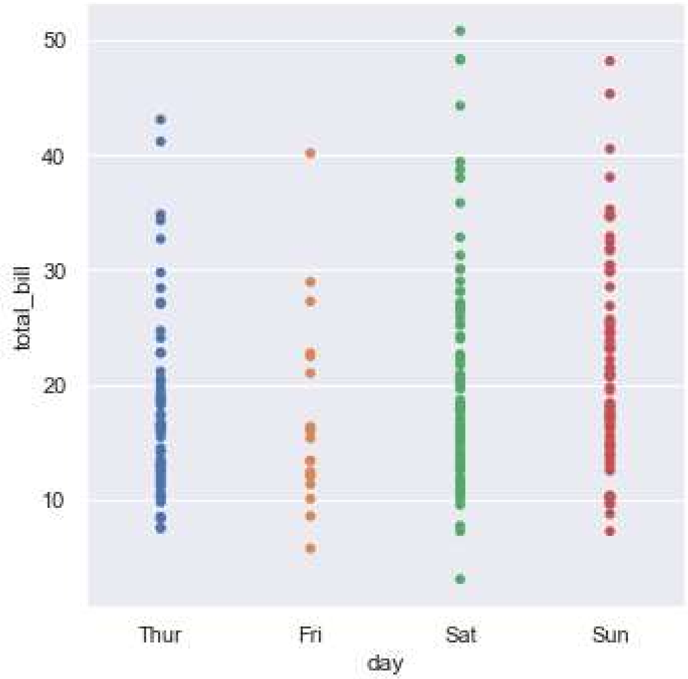


In [66]:

```
sns.catplot(x = 'day', y = 'total_bill', data = tips, jitter = False)
# jitter remove other data and plot on single line
```

Out[66]:

```
<seaborn.axisgrid.FacetGrid at 0x20787e86c70>
```



In [ ]:

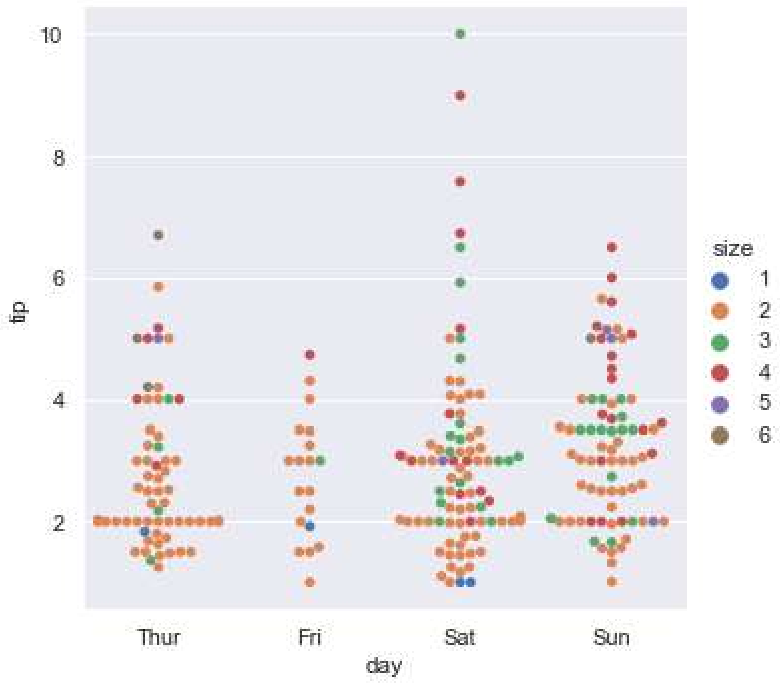
```

```

In [67]:

```
sns.catplot(x = 'day', y = 'tip', data = tips, kind = 'swarm', hue = 'size')
```

Out[67]:

```
<seaborn.axisgrid.FacetGrid at 0x20787d3cc10>
```



In [ ]:

```

```

In [69]:

```
tips
```

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
...	...	...	...	...	...	...	...
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

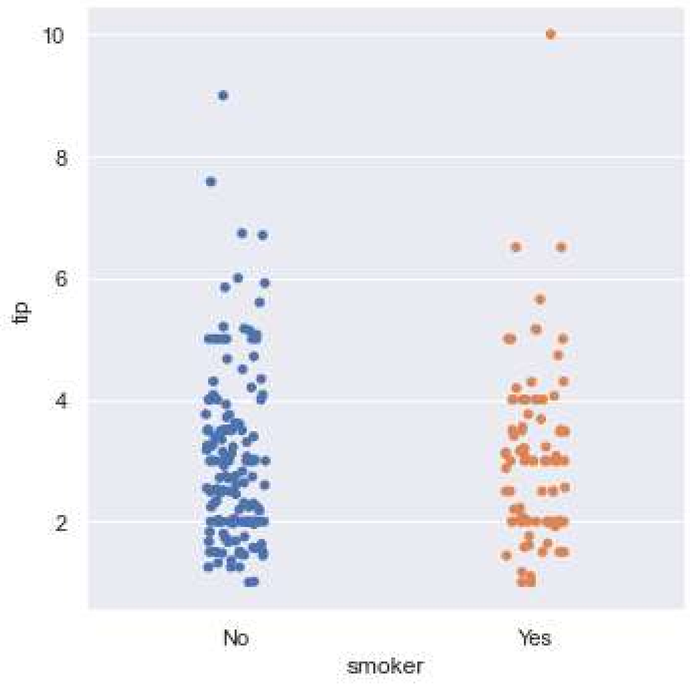
244 rows × 7 columns

In [68]:

```
sns.catplot(x = 'smoker', y = 'tip', data = tips, order= ['No', 'Yes'])
```

Out[68]:

```
<seaborn.axisgrid.FacetGrid at 0x2078a140bb0>
```



In [70]:

```
tips.head()
```

Out[70]:

	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 [ ]:

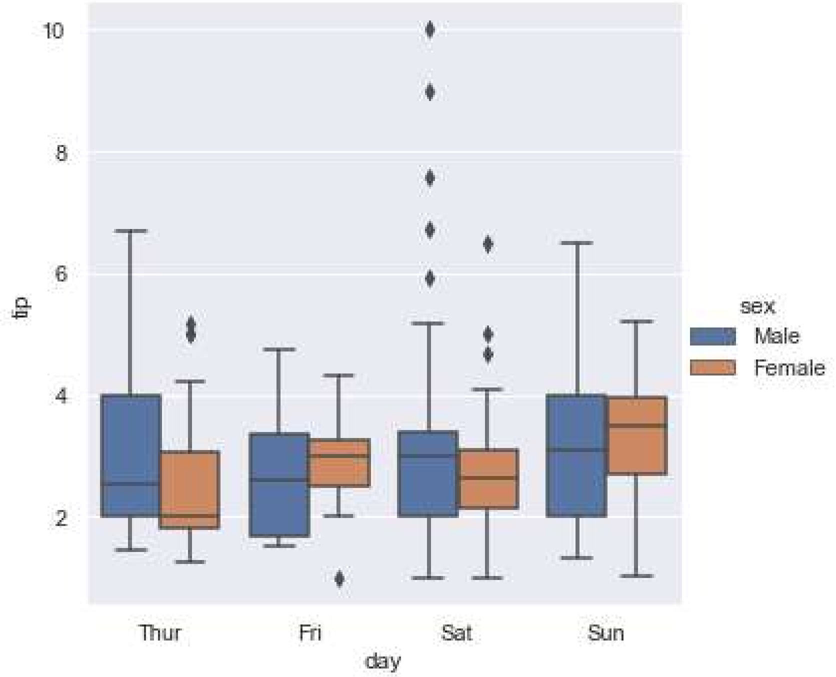
```

```

In [71]:

```
sns.catplot(x = 'day', y = 'tip', kind = 'box', data = tips, hue = 'sex')
```

Out[71]:

```
<seaborn.axisgrid.FacetGrid at 0x20787c2da30>
```

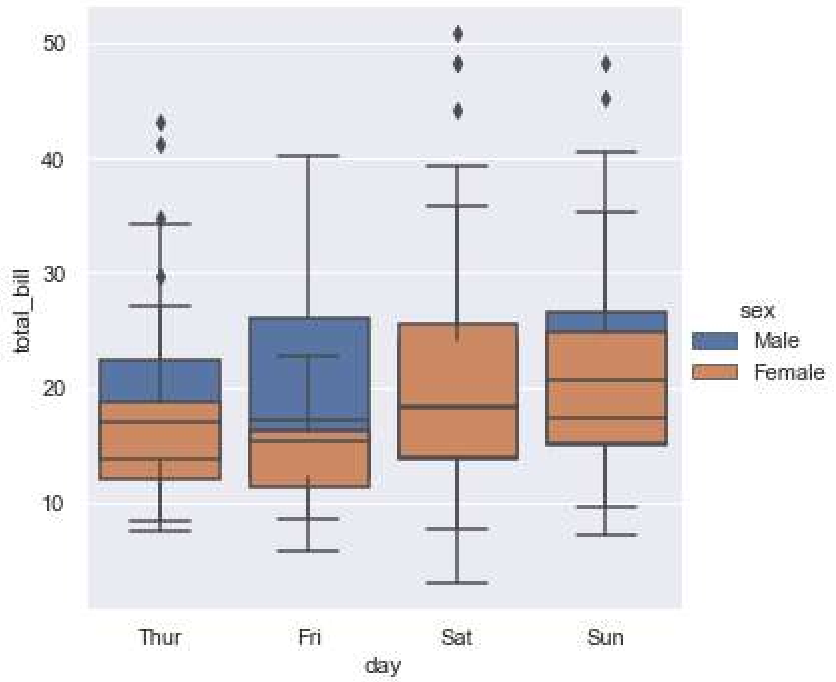


In [74]:

```python
sns.catplot(x = 'day', y = 'total_bill', kind = 'box', data = tips, hue = 'sex', dodge
# dodge
```

Out[74]:

```
<seaborn.axisgrid.FacetGrid at 0x2078a4422e0>
```



In [ ]:

```python

```

In [75]:

```python
diamonds = sns.load_dataset('diamonds')
diamonds.head()
```

Out[75]:

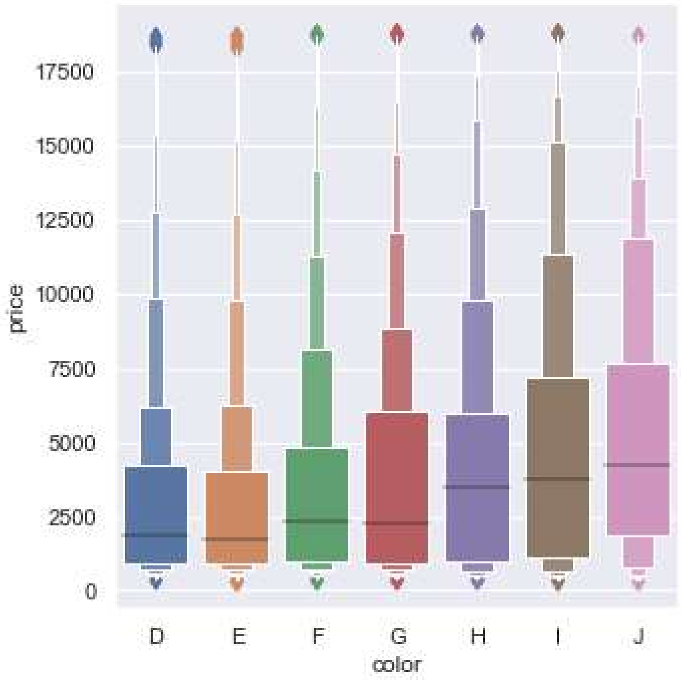
	carat	cut	color	clarity	depth	table	price	x	y	z
0	0.23	Ideal	E	SI2	61.5	55.0	326	3.95	3.98	2.43
1	0.21	Premium	E	SI1	59.8	61.0	326	3.89	3.84	2.31
2	0.23	Good	E	VS1	56.9	65.0	327	4.05	4.07	2.31
3	0.29	Premium	I	VS2	62.4	58.0	334	4.20	4.23	2.63
4	0.31	Good	J	SI2	63.3	58.0	335	4.34	4.35	2.75

In [76]:

```
sns.catplot(x = 'color', y = 'price', kind = 'boxen', data = diamonds.sort_values('colc
```

Out[76]:

```
<seaborn.axisgrid.FacetGrid at 0x2078a224430>
```

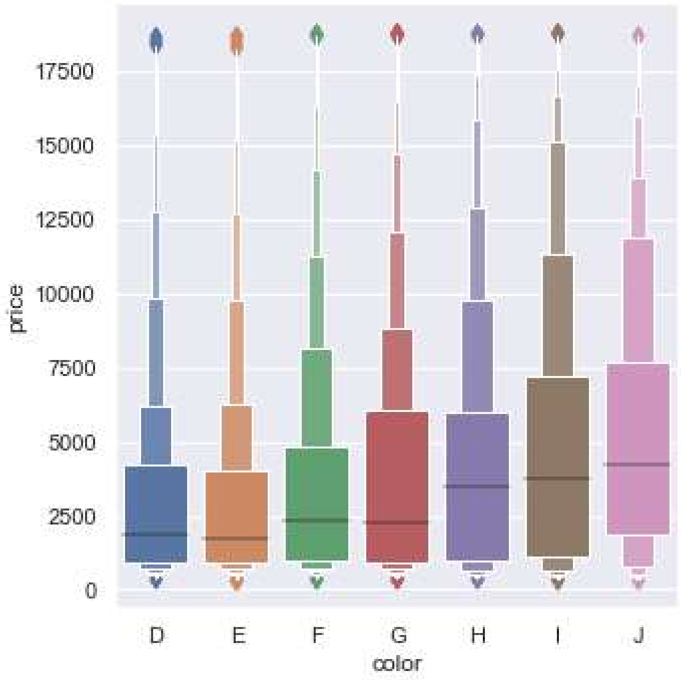


In [77]:

```
sns.catplot(x = 'color', y = 'price', kind = 'boxen', data = diamonds.sort_values('colc
```

Out[77]:

```
<seaborn.axisgrid.FacetGrid at 0x2078baa7880>
```

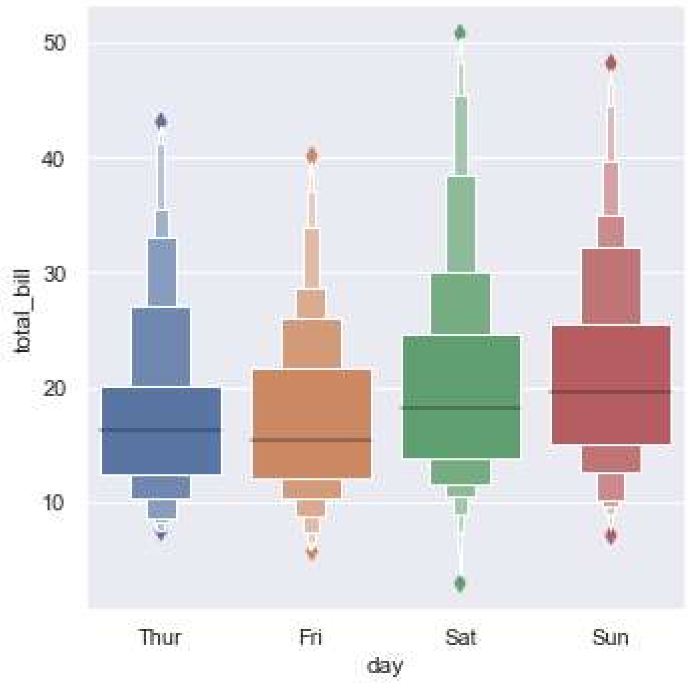


In [78]:

```
sns.catplot(x = 'day', y = 'total_bill', kind = 'boxen', data = tips, dodge = False)
```

Out[78]:

```
<seaborn.axisgrid.FacetGrid at 0x2078a1e9190>
```



In [ ]:

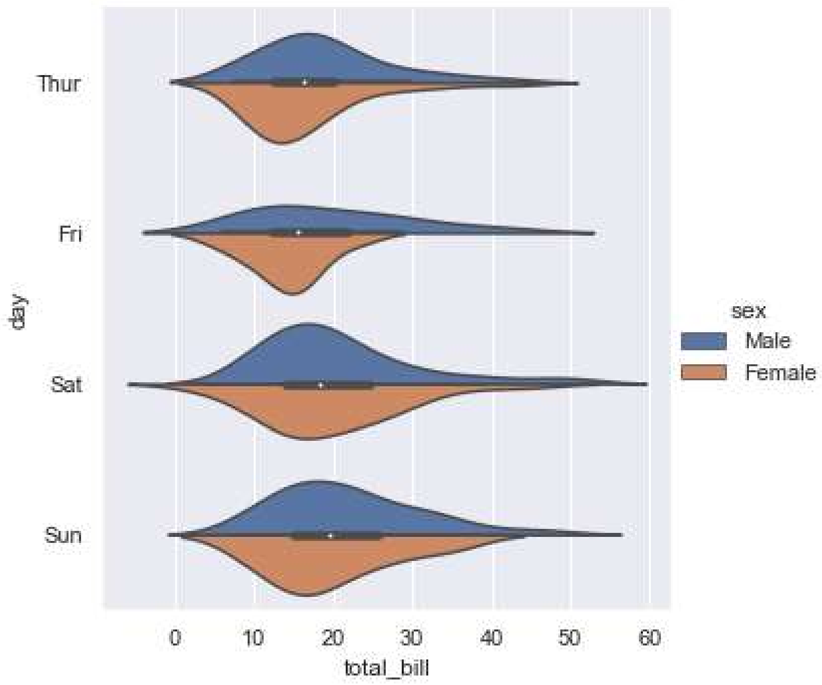
```

```

In [79]:

```
sns.catplot(x = 'total_bill', y = 'day', hue = 'sex', kind = 'violin', data = tips, spl
```

Out[79]:

```
<seaborn.axisgrid.FacetGrid at 0x2078a5bb7f0>
```



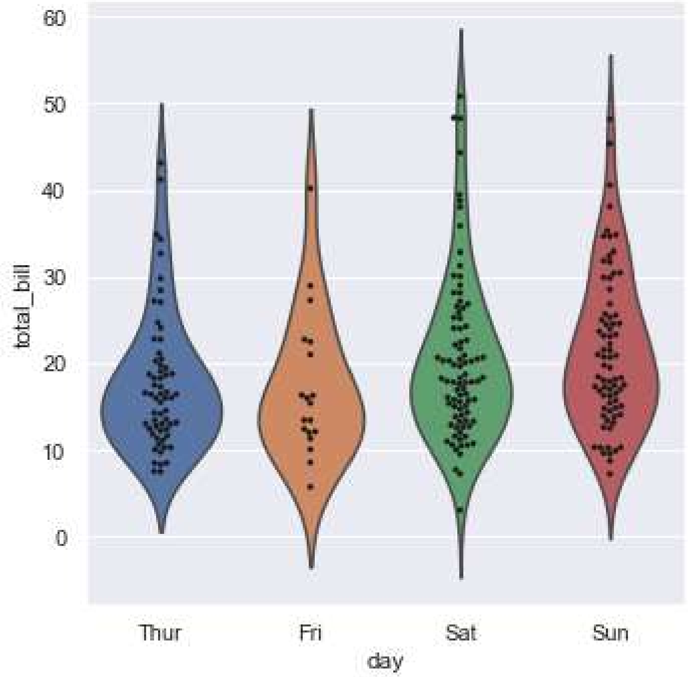
In [ ]:

In [80]:

```
g = sns.catplot(x = 'day', y = 'total_bill', kind = 'violin', inner = None, data = tips
sns.swarmplot(x = 'day', y = 'total_bill', color = 'k', size = 3, data = tips, ax = g.a
```

Out[80]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2078a1e2130>
```



In [81]:

```
titanic.head()
```

Out[81]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male
0	0	3	male	22.0	1	0	7.2500	S	Third	man	True
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
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

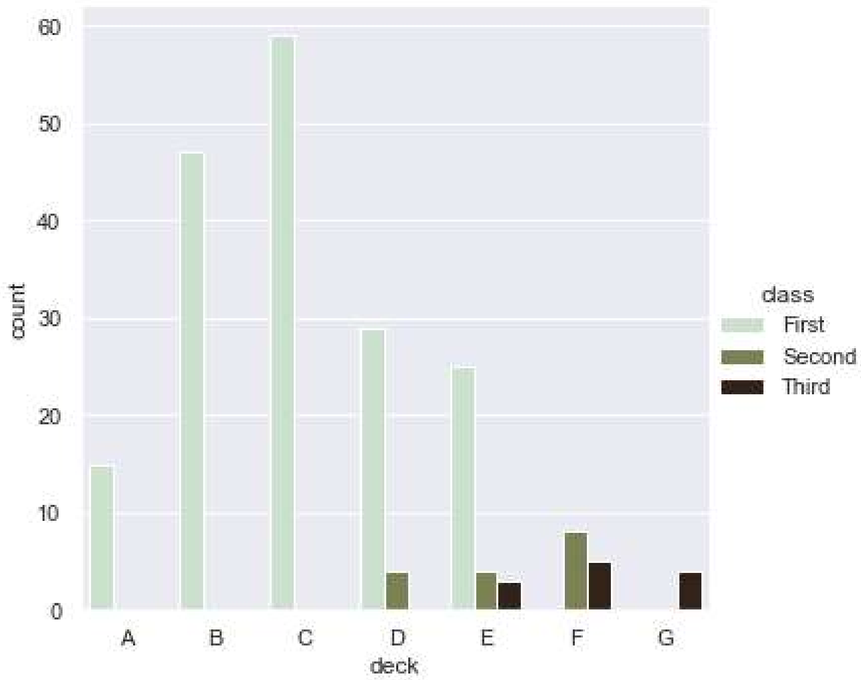
In [ ]:

In [82]:

```
sns.catplot(x = 'deck', kind = 'count', palette = 'ch:0.95', data = titanic, hue = 'cla
```

Out[82]:

```
<seaborn.axisgrid.FacetGrid at 0x2078c732df0>
```



In [ ]:

```
```

3. Visualizing Distribution of the Data

- distplot()
- kdeplot()
- jointplot()
- rugplot()

In [85]:

```
x = randn(100)
x
```

Out[85]:

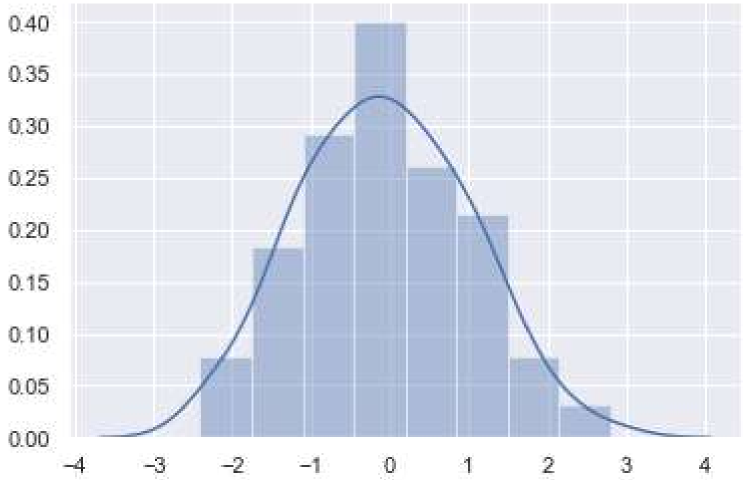
```
array([ 0.69088143, -0.11983513, -1.07511008, -1.26651175, -0.10295006,
        1.3529804 , -0.87207696,  0.50119464, -0.37710589,  0.77815484,
       -1.00132393, -0.86743377,  0.053144  ,  1.46050496, -0.25590846,
       -1.40817071, -1.09189633,  2.08725788,  0.72140114, -1.63968338,
       -0.13610849, -0.46515501,  1.14065526,  1.6890992 ,  0.37162017,
        0.341319  ,  0.54243216, -0.70008965,  1.03016697, -0.19419998,
        0.92782673,  1.58092358,  0.12924526, -0.77714922, -2.10082722,
       -0.66078449, -0.8818915 , -0.26011751,  1.19305859, -0.85099076,
       -2.02117578, -2.408257  , -1.34970639, -0.99049693, -2.15558437,
        0.42074031, -0.91091426,  0.18574755,  1.25501874,  1.14046886,
       -1.46851741,  1.40551125, -0.60410478,  1.49692786,  1.89027172,
       -0.07258838, -1.59846176,  0.83592762,  0.1023862 , -1.62899057,
       -0.06050826, -2.11577615,  0.56411863, -0.41589988,  0.52813006,
        0.64053363,  1.02225705,  0.06666744,  0.47059688,  0.40228342,
        0.13327328, -0.16858642,  0.9936799 ,  1.22947335, -1.31956358,
       -1.29453752,  0.17357164,  2.7929335 ,  1.01392013,  0.50746425,
        2.18652341, -0.27747453, -1.36288775, -0.4276486 ,  0.40890538,
       -0.10212417,  0.15604738,  0.53558841, -0.77556475, -0.80245071,
       -0.39060762,  0.98353379, -0.2474691 , -1.17689214, -0.2295814 ,
       -1.23030291, -0.58794071, -0.34480871, -0.52983624, -1.06623043])
```

In [86]:

```
sns.distplot(x, kde = True)
```

Out[86]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2078c7f07c0>
```

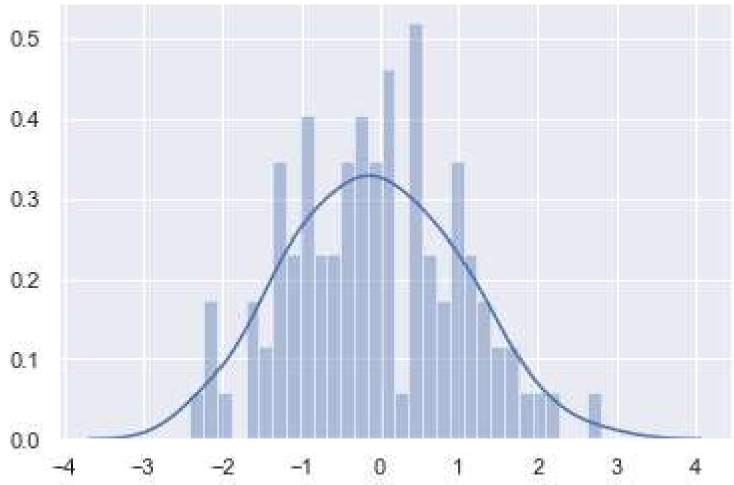


In [87]:

```
sns.distplot(x, kde = True, hist = True, rug= False, bins= 30)
```

Out[87]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2078c913a60>
```



In [88]:

```
tips.head()
```

Out[88]:

	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 [89]:

```
x = tips['total_bill']
y = tips['tip']
```

In [ ]:

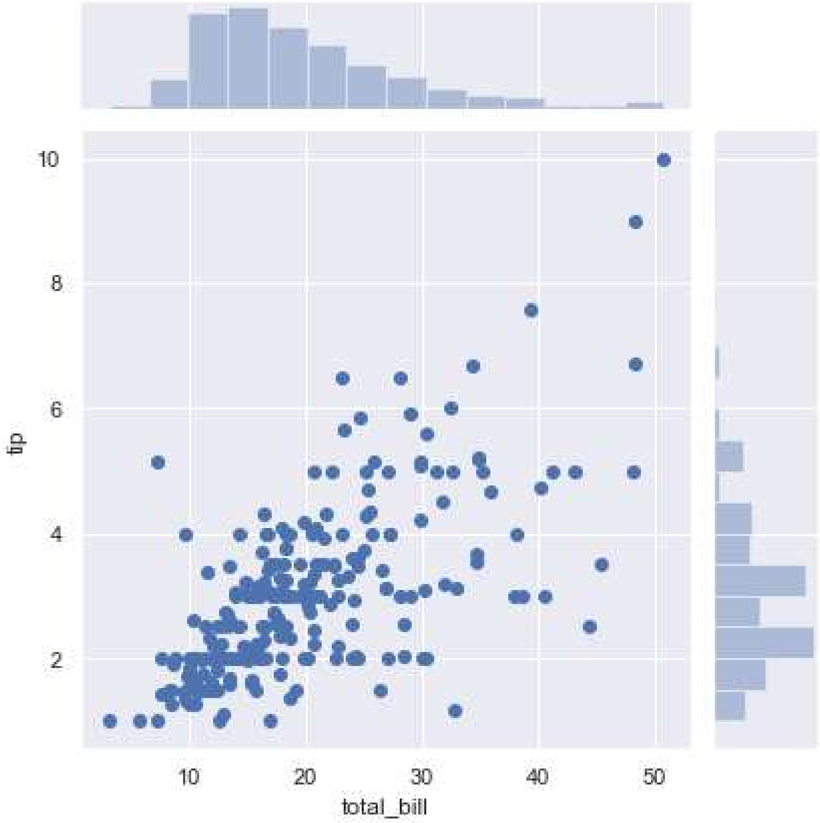
```

```

In [90]:

```
sns.jointplot(x = x, y=y)
```

Out[90]:

```
<seaborn.axisgrid.JointGrid at 0x2078c935d60>
```

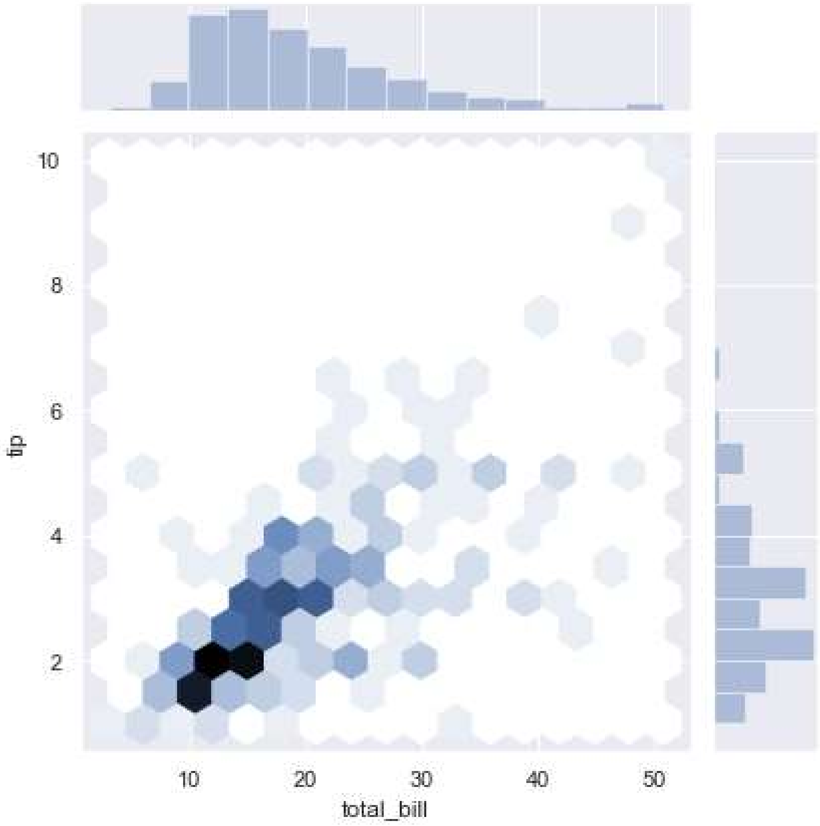


In [91]:

```
sns.set()
sns.jointplot(x = x, y=y, kind = 'hex')
```

Out[91]:

```
<seaborn.axisgrid.JointGrid at 0x2078cab77f0>
```



In [ ]:

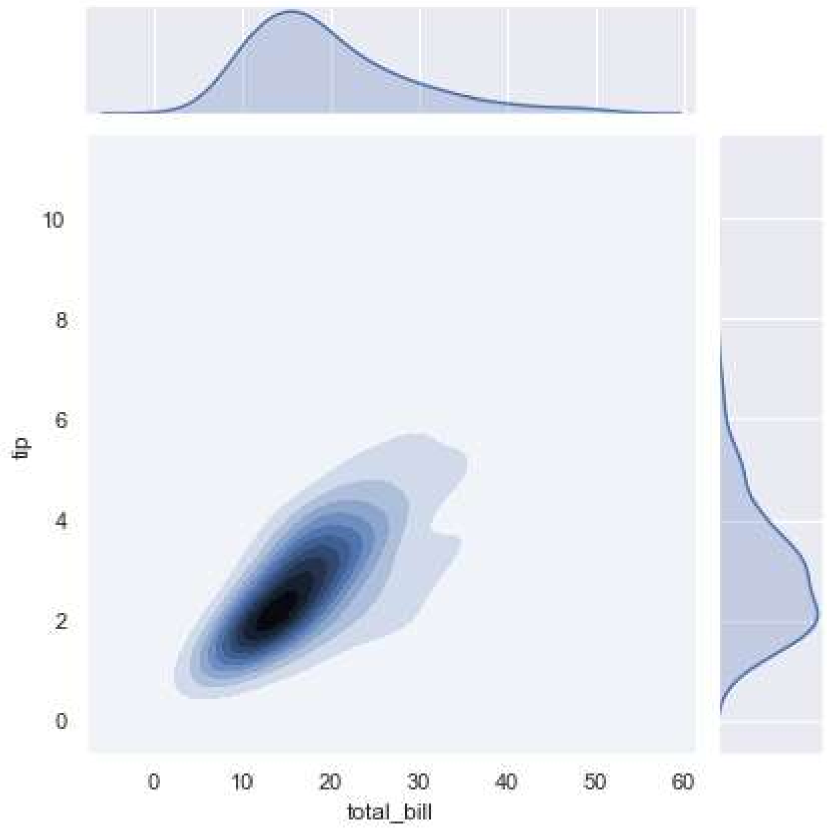
```

```

In [92]:

```
sns.jointplot(x = x, y = y, kind = 'kde')
```

Out[92]:

```
<seaborn.axisgrid.JointGrid at 0x2078cbe10d0>
```



In [ ]:

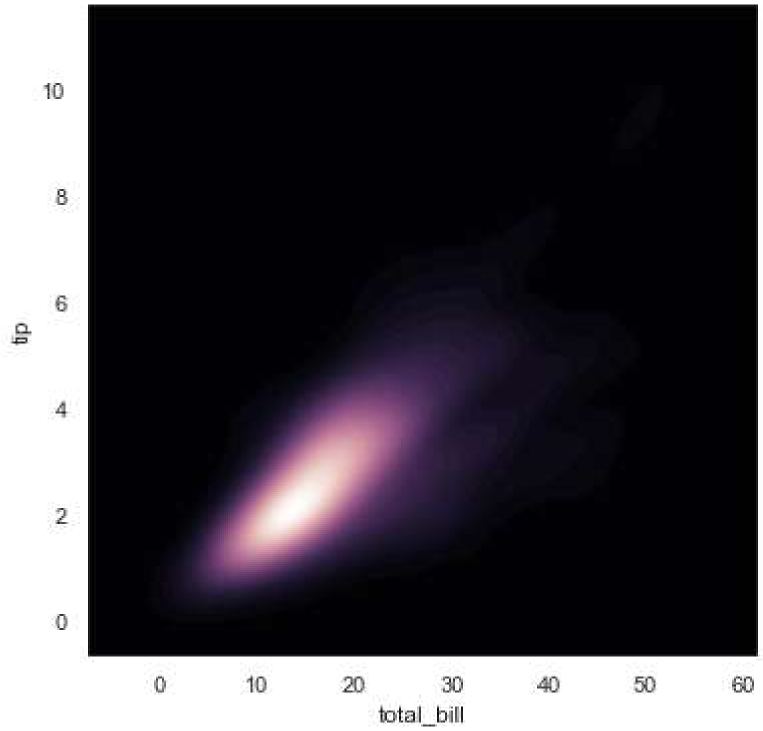
```

```

In [93]:

```
f, ax = plt.subplots(figsize = (6,6))
cmap = sns.cubehelix_palette(as_cmap = True, dark = 0, light = 1, reverse= True)
sns.kdeplot(x, y, cmap = cmap, n_levels=60, shade=True)
```

Out[93]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2078cccc160>
```

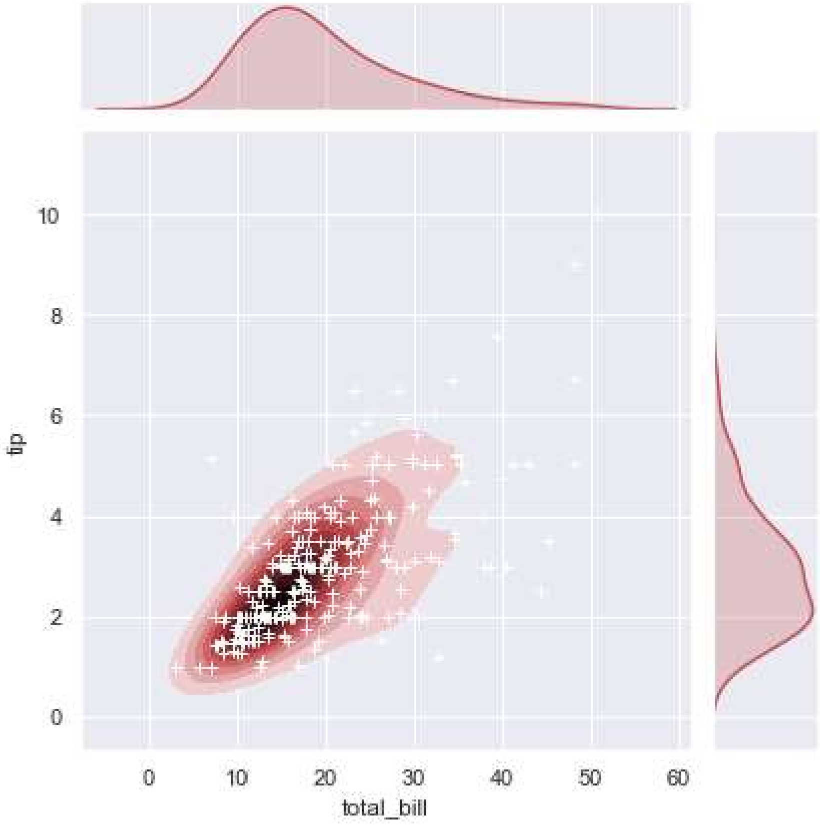


In [ ]:

```

```

In [94]:

```
g = sns.jointplot(x, y, kind = 'kde', color = 'r')
g.plot_joint(plt.scatter, c = 'w', s = 30, linewidth = 1, marker = '+')
g.ax_joint.collections[0].set_alpha(0)
```

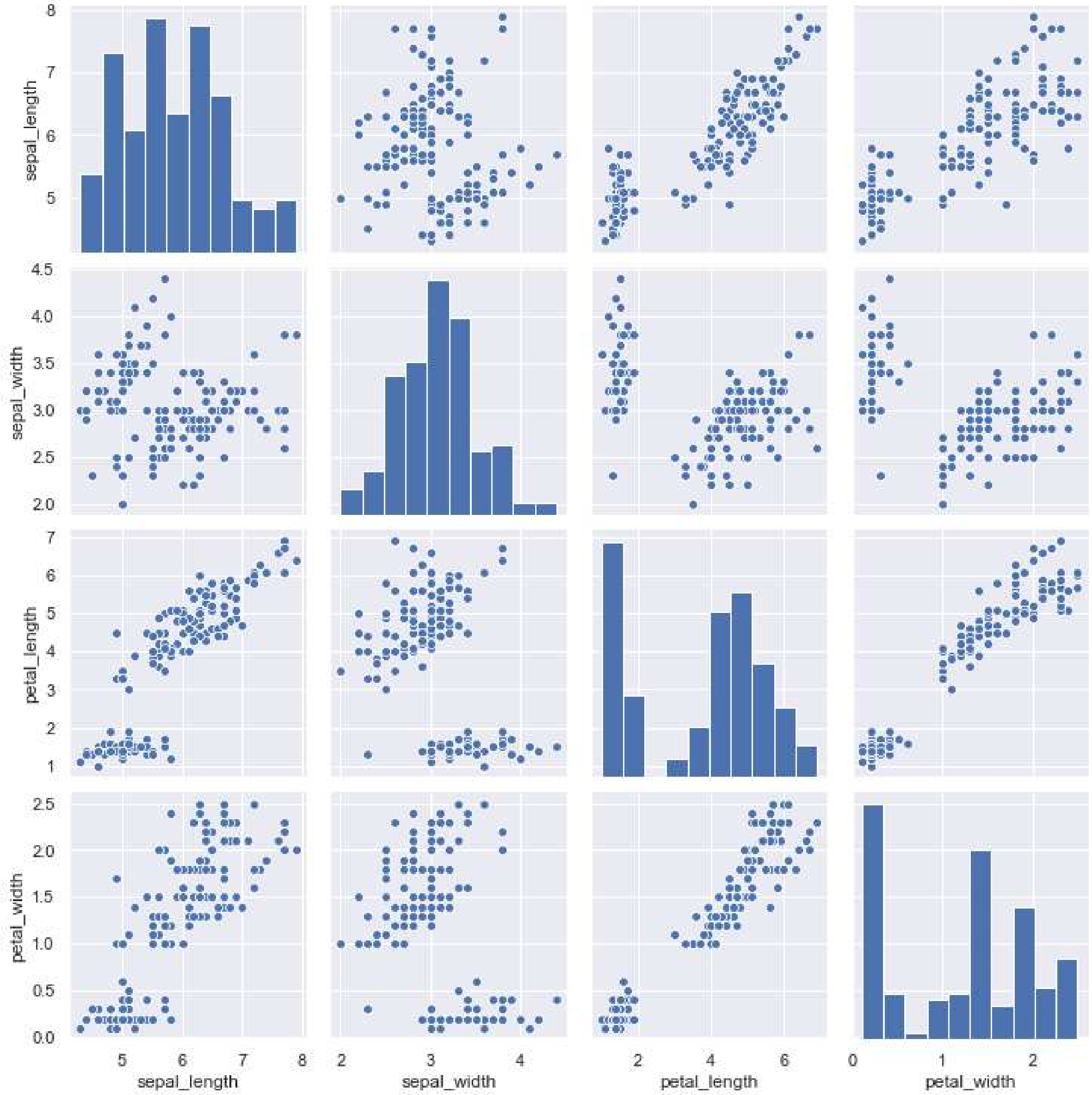


In [95]:

```
sns.pairplot(iris)

```

Out[95]:

```
<seaborn.axisgrid.PairGrid at 0x2078cea2850>
```

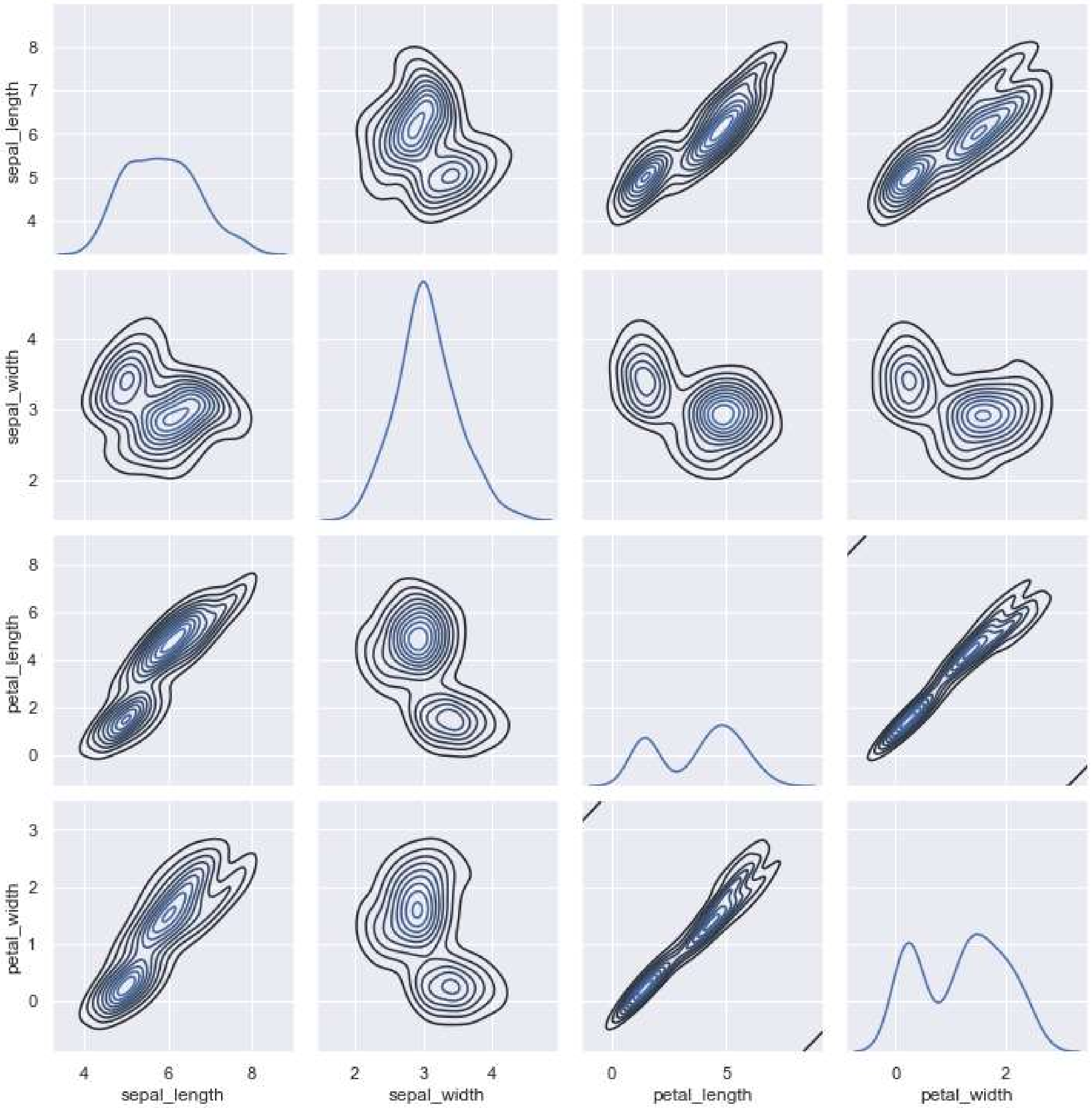


In [97]:

```
g = sns.PairGrid(iris)
g.map_diag(sns.kdeplot)
g.map_offdiag(sns.kdeplot, n_levels = 10)
# give top view

```

Out[97]:

```
<seaborn.axisgrid.PairGrid at 0x2078d9595e0>
```



In [ ]:

```

```

4. Linear Regression and Relationship

- regplot()
- lmplot()

In [99]:

```
tips.head()
```

Out[99]:

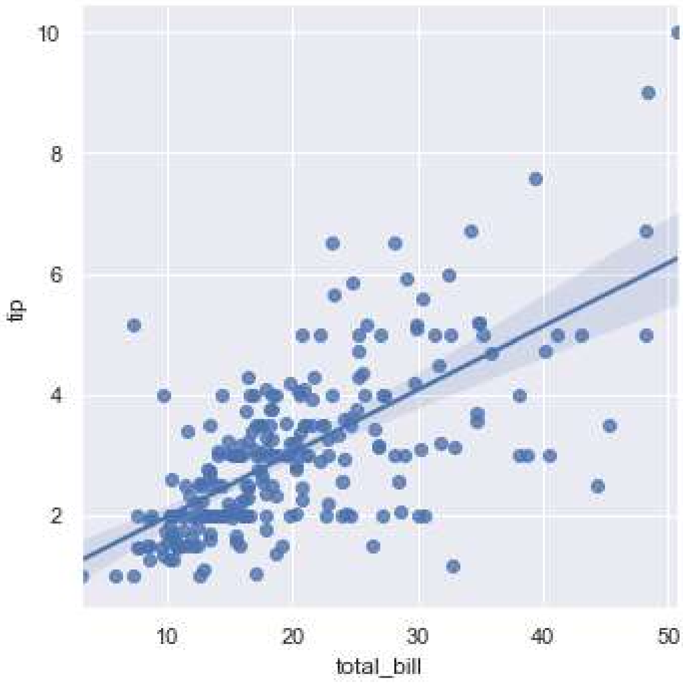
	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4

In [100]:

```
sns.lmplot(x = 'total_bill', y= 'tip', data = tips)
```

Out[100]:

```
<seaborn.axisgrid.FacetGrid at 0x2078f008040>
```



In [ ]:

```

```

In [101]:

```
data = sns.load_dataset('anscombe')
data.head()
```

Out[101]:

	dataset	x	y
0	I	10.0	8.04
1	I	8.0	6.95
2	I	13.0	7.58
3	I	9.0	8.81
4	I	11.0	8.33

In [102]:

```
data['dataset'].value_counts()
```

Out[102]:

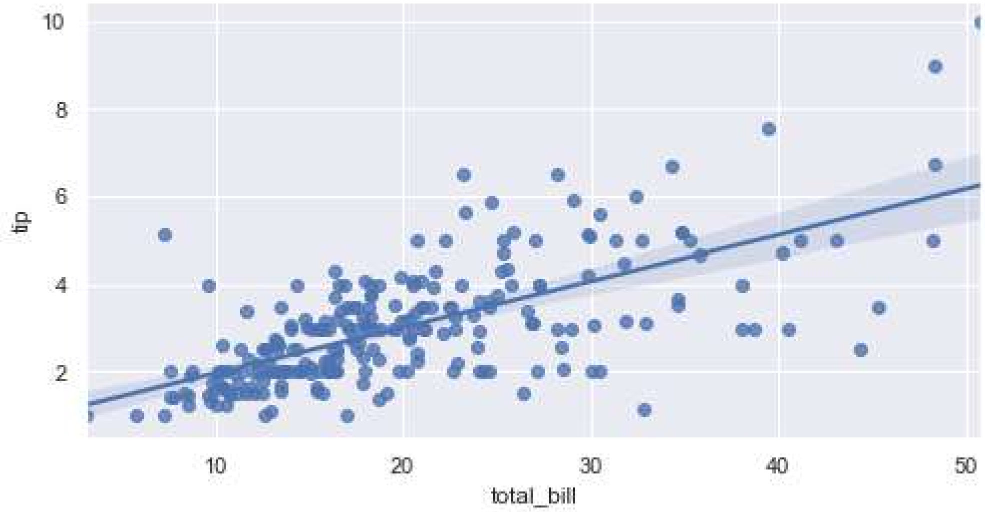
```
III    11
IV     11
II     11
I      11
Name: dataset, dtype: int64
```

In [103]:

```
f, ax = plt.subplots(figsize = (8,4))
sns.regplot(x = 'total_bill', y = 'tip', data = tips, ax = ax)
```

Out[103]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2078fbd7940>
```



In [ ]:

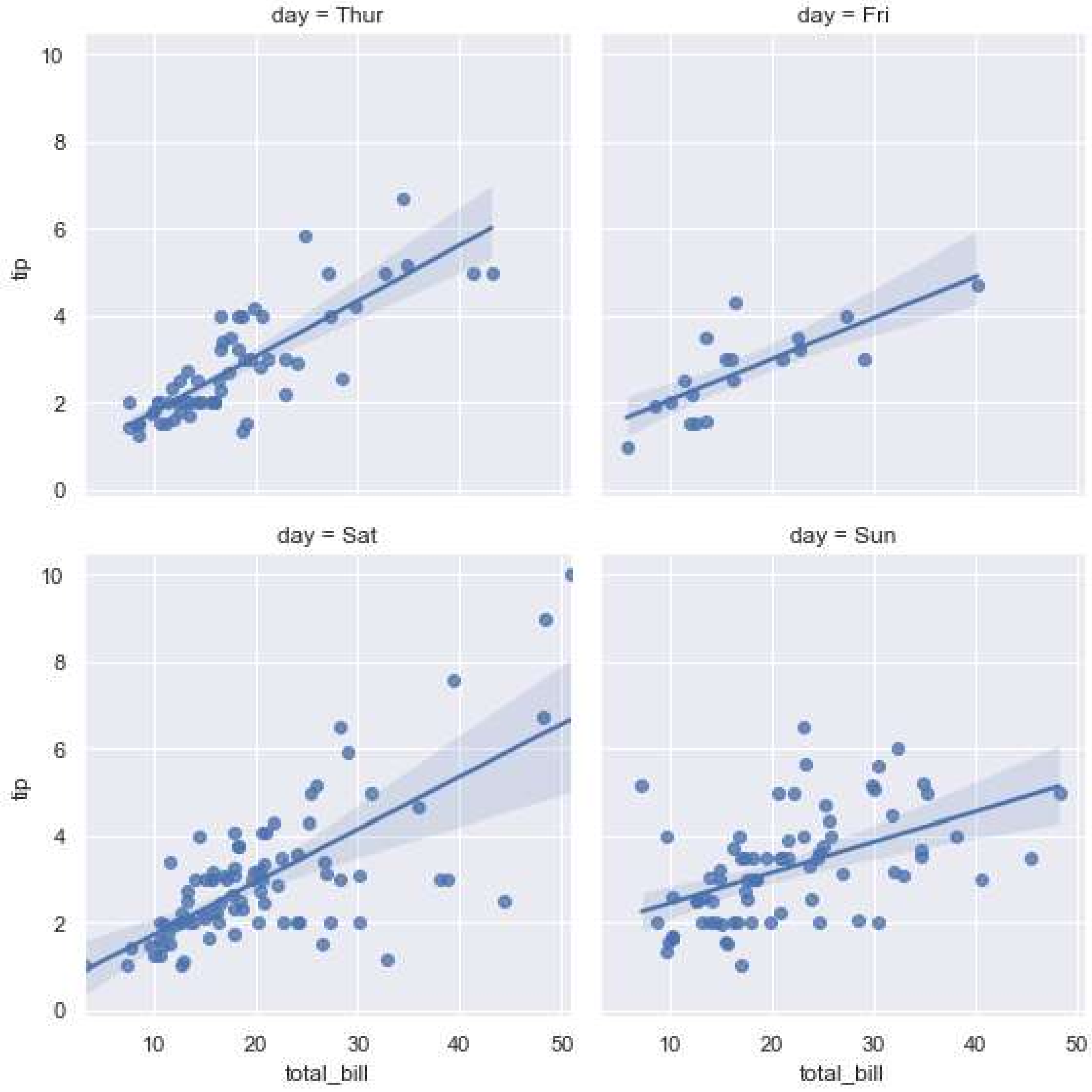
```

```

In [104]:

```
sns.lmplot(x = 'total_bill', y = 'tip', data = tips, col = 'day', col_wrap=2, height =
```

Out[104]:

```
<seaborn.axisgrid.FacetGrid at 0x20790bfdb50>
```



In [ ]:

```

```

5. Controlling Ploted Figure Aesthetics

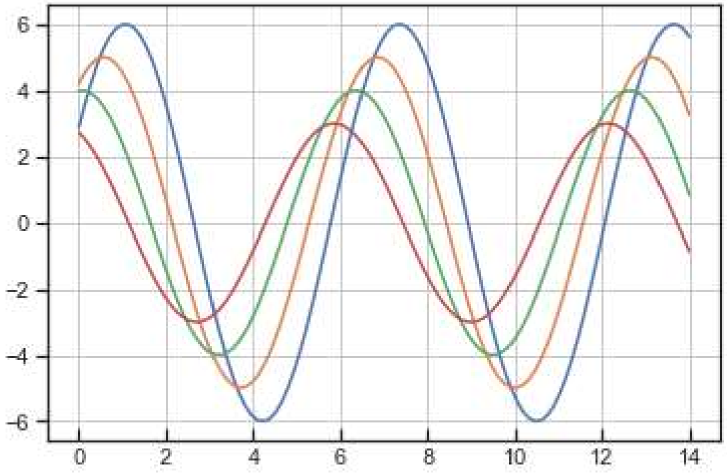
- figure styling
- axes styling
- color palettes
- etc..

In [120]:

```
def sinplot():
    x = np.linspace(0, 14, 100)
    for i in range(1, 5):
        plt.plot(x, np.sin(x+i*0.5)*(7-i))
```

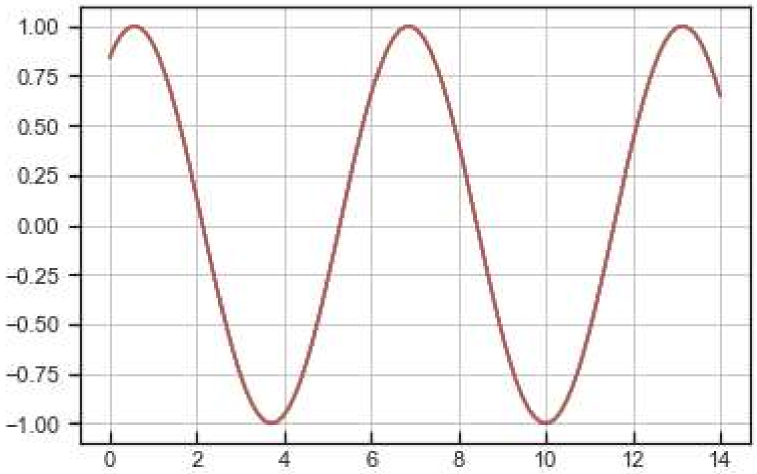
In [121]:

```
sinplot()
```



In [122]:

```
def sinplot():
    x = np.linspace(0, 14, 100)
    for i in range(1, 5):
        plt.plot(x, np.sin(x+1))
sinplot()
```

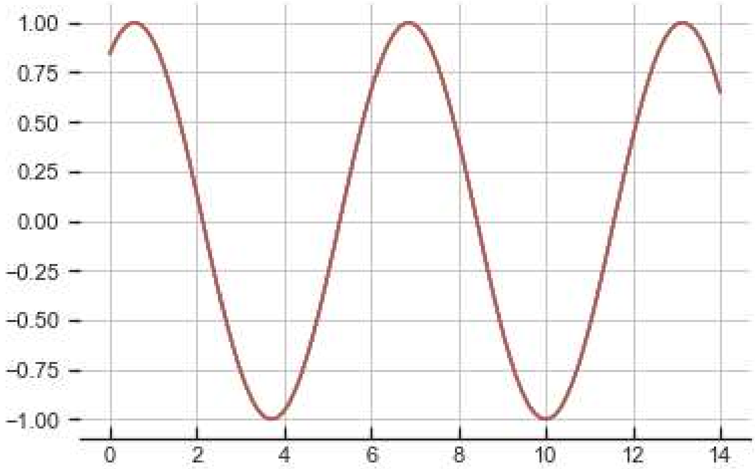


In [ ]:

```

```

In [123]:

```
sns.set_style('ticks', {'axes.grid': True, 'xtick.direction': 'in'})
sinplot()
sns.despine(left = True, bottom= False)
```



In [124]:

```
sns.axes_style()
```

Out[124]:

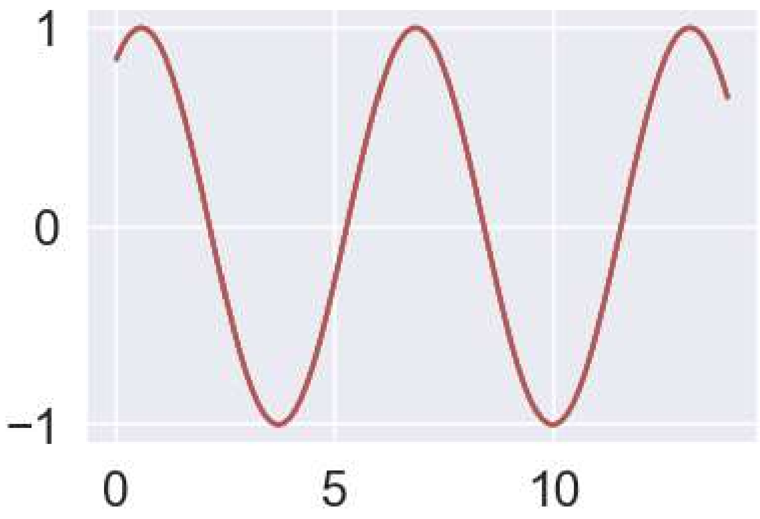
```
{'axes.facecolor': 'white',
 'axes.edgecolor': '.15',
 'axes.grid': True,
 'axes.axisbelow': True,
 'axes.labelcolor': '.15',
 'figure.facecolor': 'white',
 'grid.color': '.8',
 'grid.linestyle': '-',
 'text.color': '.15',
 'xtick.color': '.15',
 'ytick.color': '.15',
 'xtick.direction': 'in',
 'ytick.direction': 'out',
 'lines.solid_capstyle': 'round',
 'patch.edgecolor': 'w',
 'patch.force_edgecolor': True,
 'image.cmap': 'rocket',
 'font.family': ['sans-serif'],
 'font.sans-serif': ['Arial',
  'DejaVu Sans',
  'Liberation Sans',
  'Bitstream Vera Sans',
  'sans-serif'],
 'xtick.bottom': True,
 'xtick.top': False,
 'ytick.left': True,
 'ytick.right': False,
 'axes.spines.left': True,
 'axes.spines.bottom': True,
 'axes.spines.right': True,
 'axes.spines.top': True}
```

In [125]:

```
sns.set_style('darkgrid')
```

In [126]:

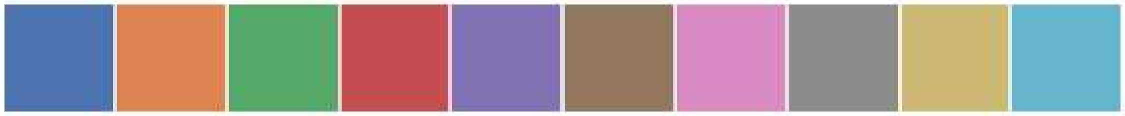
```
sns.set_context('talk', font_scale=1.5)
sinplot()
```



In [ ]:

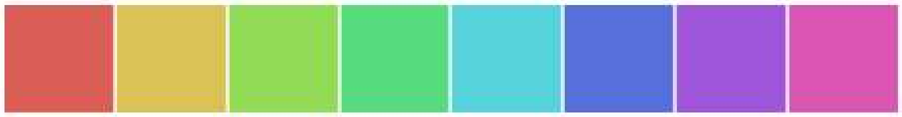
In [127]:

```
current_palettes = sns.color_palette()
sns.palplot(current_palettes)
```



In [128]:

```
sns.palplot(sns.color_palette('hls', 8))
```



In [ ]: