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 [87]:

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

In [ ]:

```
```

In [88]:

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

In [92]:

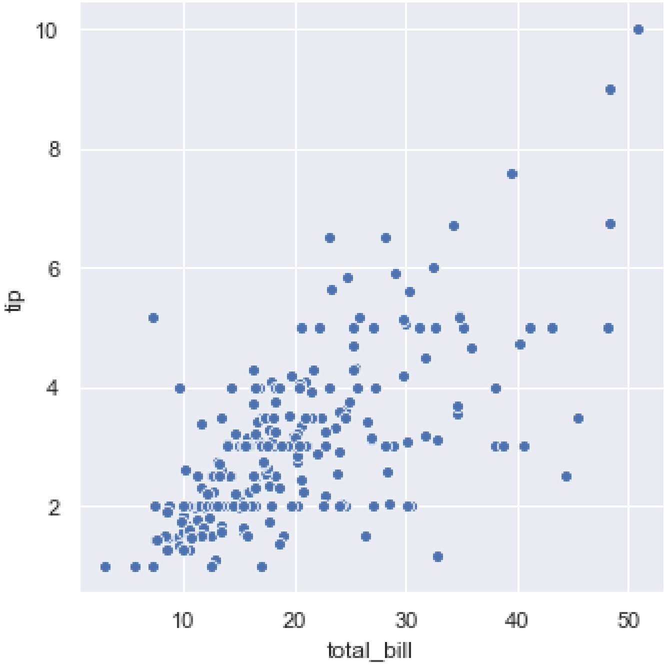
```
tips = sns.load_dataset('tips')
tips['size']
```

```
Out[92]: 0      2
         1      3
         2      3
         3      2
         4      4
         5      4
         6      2
         7      4
         8      2
         9      2
         10     2
         11     4
         12     2
         13     4
         14     2
         15     2
         16     3
         17     3
         18     3
         19     3
         20     2
         21     2
         22     2
         23     4
         24     2
         25     4
         26     2
         27     2
         28     2
         29     2
               ..
         214    3
         215    2
         216    5
         217    2
         218    2
         219    4
         220    2
         221    2
         222    1
         223    3
```

```
224    2
225    2
226    2
227    4
228    2
229    2
230    4
231    3
232    2
233    2
234    2
235    2
236    2
237    2
238    3
239    3
240    2
241    2
242    2
243    2
Name: size, Length: 244, dtype: int64
```

```
In [4]: sns.relplot(x = 'total_bill', y = 'tip', data = tips)
```

```
Out[4]: <seaborn.axisgrid.FacetGrid at 0x1a163cca58>
```



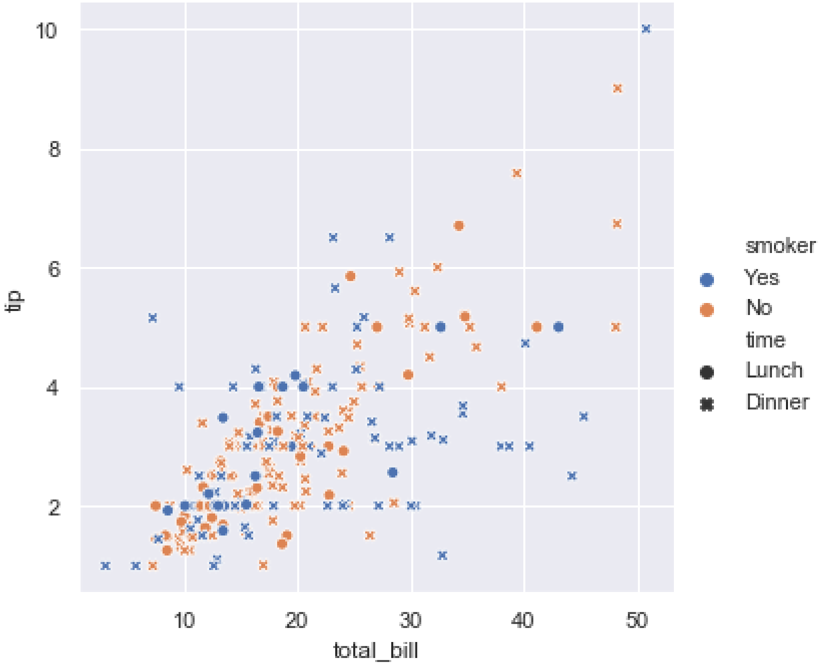
```
In [ ]:
```

```
In [90]: tips['smoker'].value_counts()
```

```
Out[90]: No      151
         Yes      93
         Name: smoker, dtype: int64
```

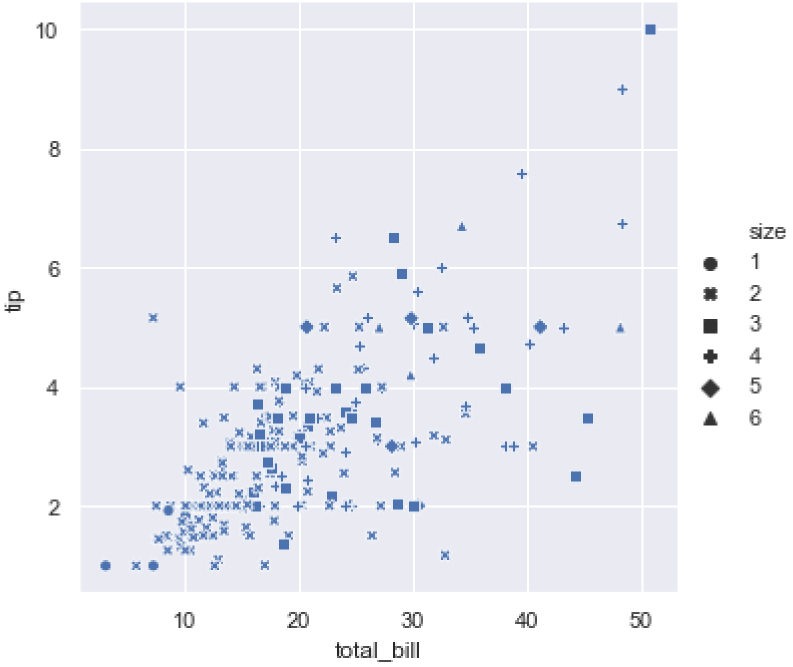
```
In [6]: sns.relplot(x = 'total_bill', y = 'tip', data = tips, hue = 'smoker', style = 'time')
```

```
Out[6]: <seaborn.axisgrid.FacetGrid at 0x1a167335f8>
```



```
In [7]: sns.relplot(x = 'total_bill', y = 'tip', style = 'size', data = tips)
```

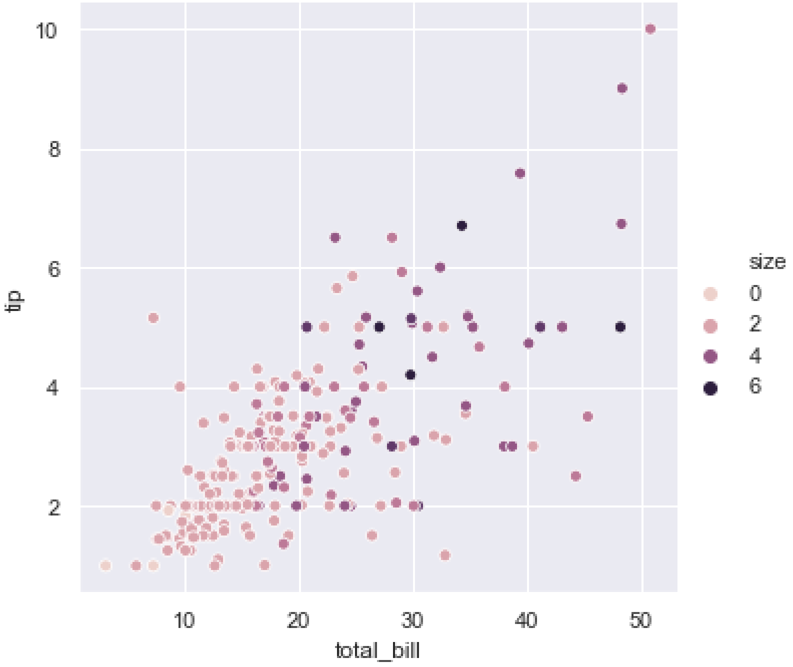
```
Out[7]: <seaborn.axisgrid.FacetGrid at 0x1a169183c8>
```



```
In [ ]:
```

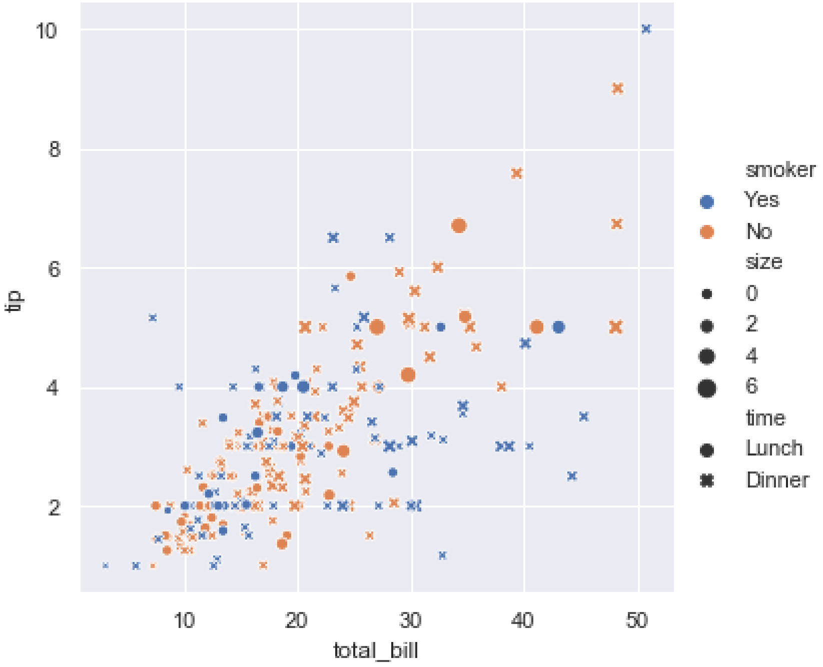
```
In [8]: sns.relplot(x = 'total_bill', y = 'tip', hue = 'size', data = tips)
```

```
Out[8]: <seaborn.axisgrid.FacetGrid at 0x1a16af3b38>
```



In [91]:
```
sns.relplot(x = 'total_bill', y = 'tip', data = tips,hue = 'smoker', style = 'time', size = 'size')
```

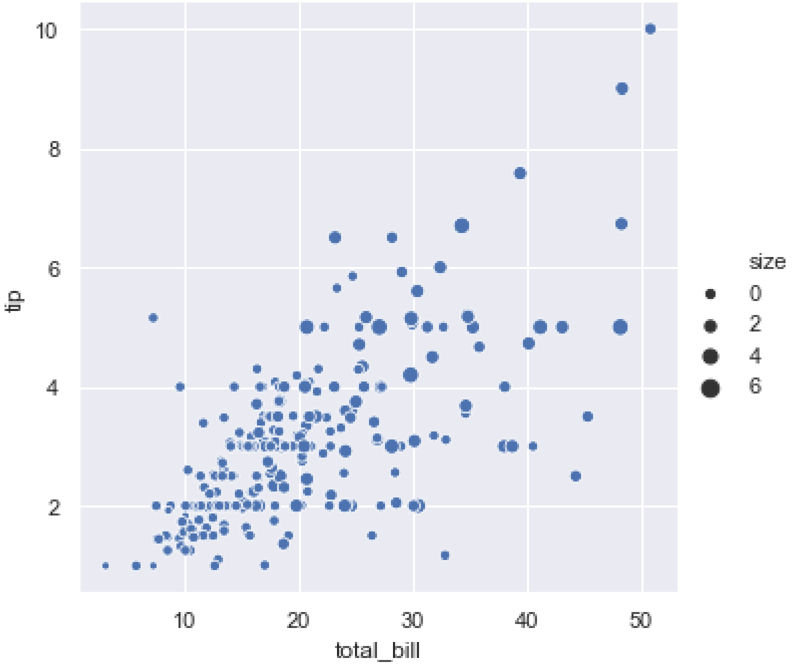
Out[91]:
```
<seaborn.axisgrid.FacetGrid at 0x1a182184a8>
```



In [ ]:

```
In [10]: sns.relplot(x = 'total_bill', y = 'tip', data = tips, size = 'size')
```

```
Out[10]: <seaborn.axisgrid.FacetGrid at 0x1a16e30860>
```



```
In [ ]:
```

```
In [11]: from numpy.random import randn
```

```
In [93]: df = pd.DataFrame(dict(time = np.arange(500), value = randn(500).cumsum()))
```

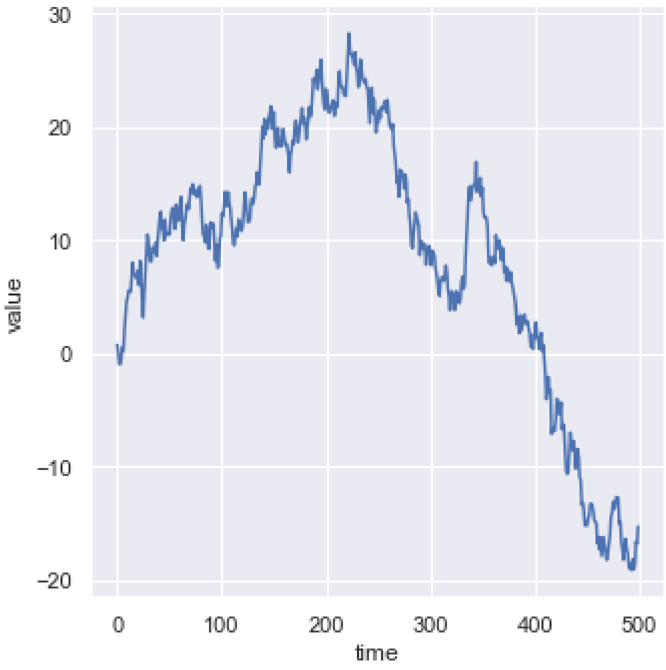
In [94]:
```
df.head()
```

Out[94]:

	time	value
0	0	0.778793
1	1	0.220757
2	2	-0.925876
3	3	-0.934216
4	4	-0.429557

In [95]:
```
sns.relplot(x = 'time', y = 'value', kind = 'line', data = df, sort = True)
```

Out[95]:
```
<seaborn.axisgrid.FacetGrid at 0x1a1eff5898>
```



In [ ]:

In [131]:
```
df = pd.DataFrame(randn(500, 2).cumsum(axis = 0), columns = ['time', 'value'])
```

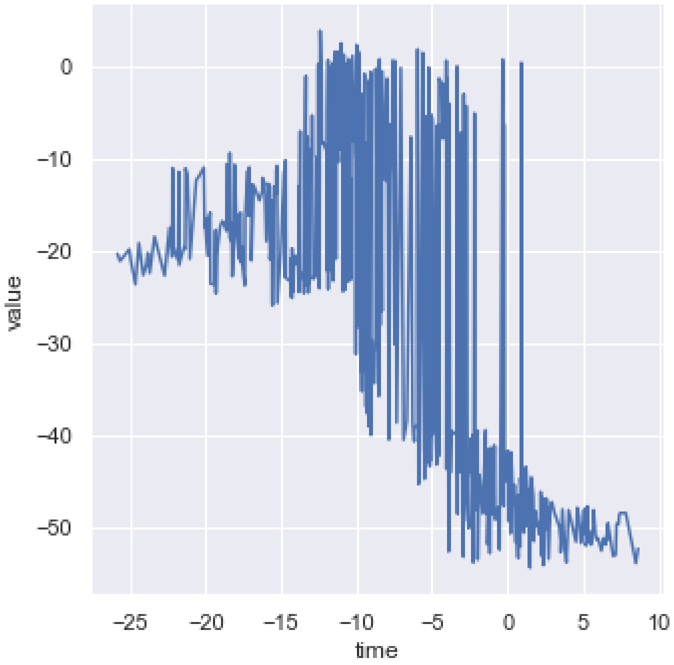
In [132]:
```
df.head()
```

Out[132]:

	time	value
0	0.903653	0.368656
1	-0.308014	0.718321
2	-3.324021	-0.020677
3	-3.974930	-1.278554
4	-4.260735	-1.982611

In [136]:
```
sns.relplot(x = 'time', y = 'value', kind = 'line', data = df, sort = True)
```

Out[136]:
```
<seaborn.axisgrid.FacetGrid at 0x1a2240d470>
```



In [ ]:

In [99]:

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

Out[99]:

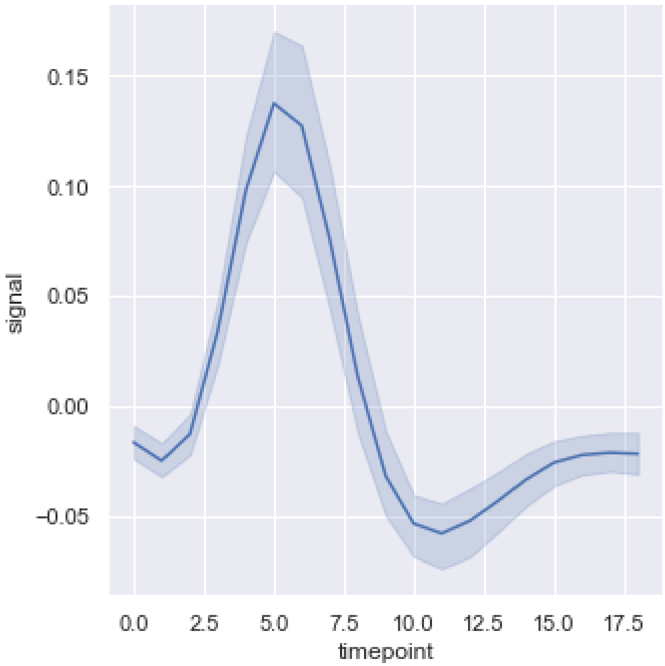
	subject	timepoint	event	region	signal
0	s13	18	stim	parietal	-0.017552
1	s5	14	stim	parietal	-0.080883
2	s12	18	stim	parietal	-0.081033
3	s11	18	stim	parietal	-0.046134
4	s10	18	stim	parietal	-0.037970

In [100]:

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

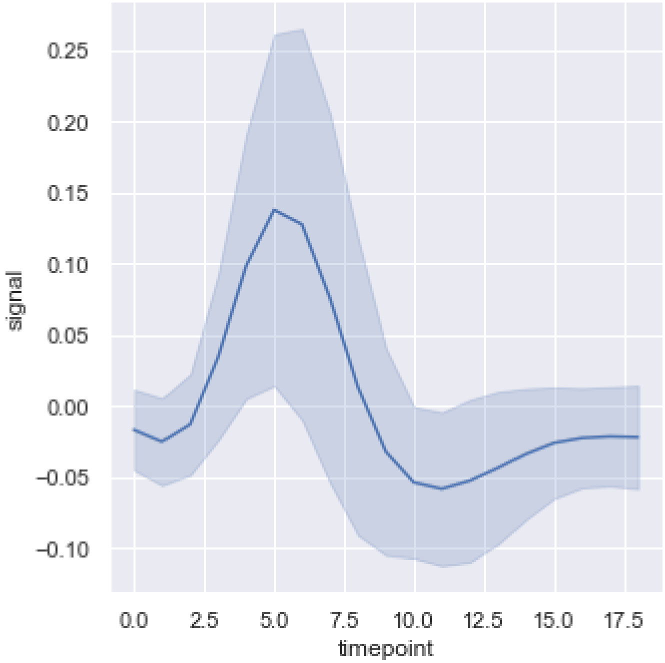
Out[100]:

```
<seaborn.axisgrid.FacetGrid at 0x1a1f6083c8>
```



```
In [101]: sns.relplot(x = 'timepoint', y = 'signal', kind = 'line', data = fmri, ci = 'sd')
```

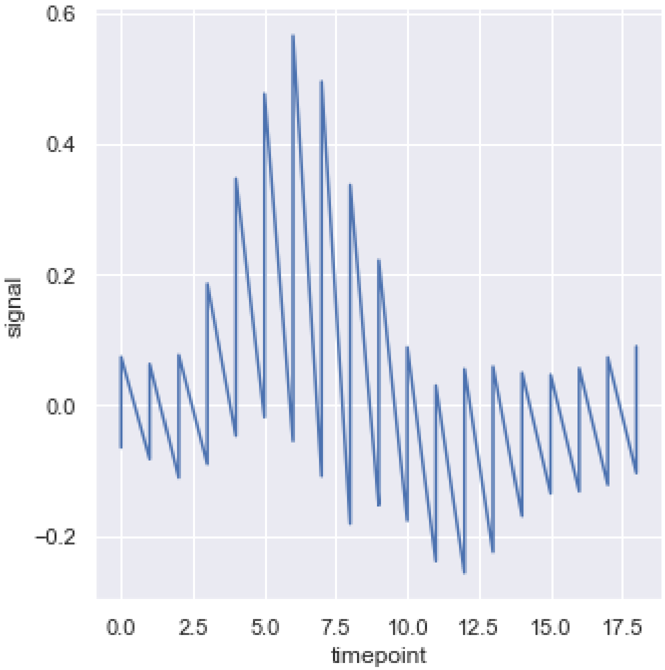
```
Out[101]: <seaborn.axisgrid.FacetGrid at 0x1a1f7a8390>
```



```
In [ ]:
```

```
In [102]: sns.relplot(x = 'timepoint', y = 'signal', estimator = None, kind = 'line', data = fmri)
```

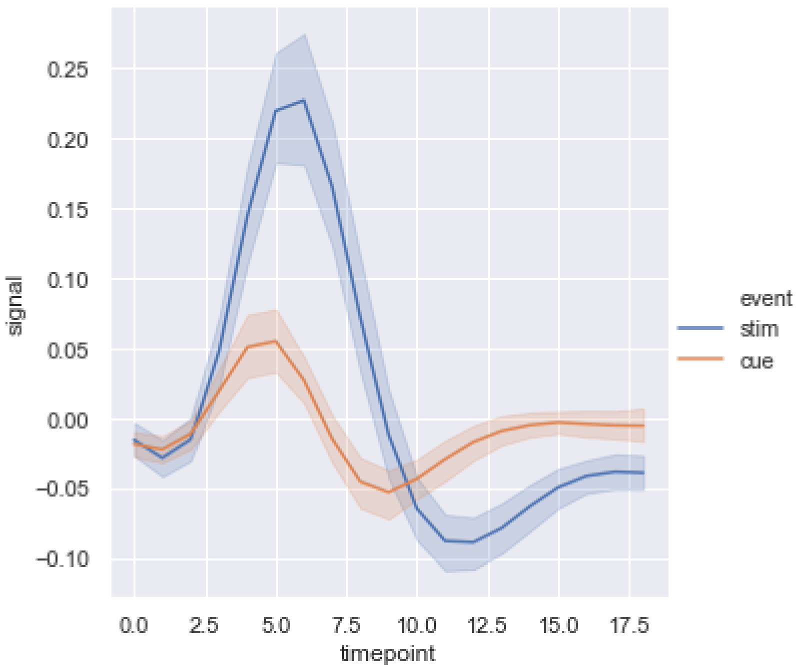
```
Out[102]: <seaborn.axisgrid.FacetGrid at 0x1a1f7a87b8>
```



```
In [ ]:
```

```
In [22]: sns.relplot(x = 'timepoint', y = 'signal', hue = 'event', kind = 'line', data = fmri)
```

```
Out[22]: <seaborn.axisgrid.FacetGrid at 0x1a17cae748>
```



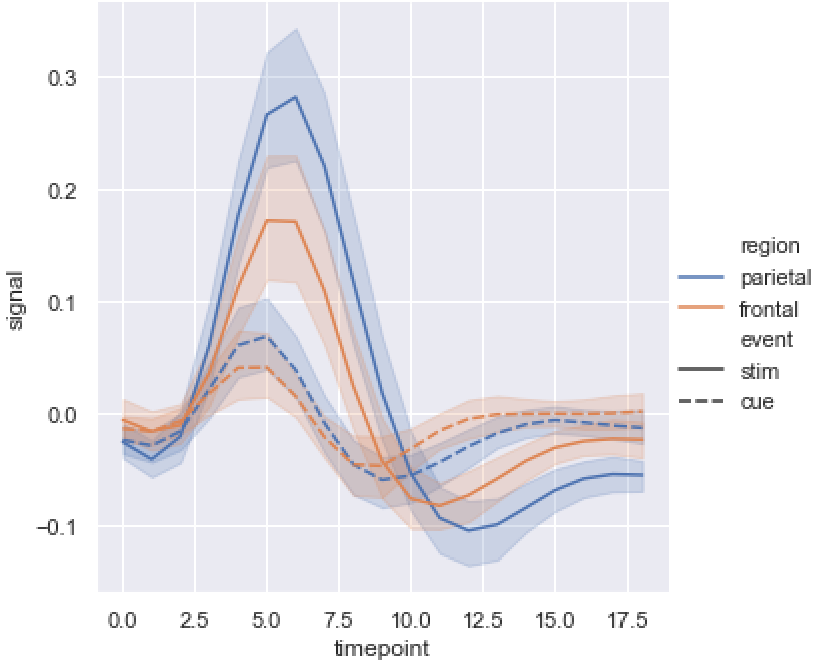
```
In [23]: fmri.head()
```

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

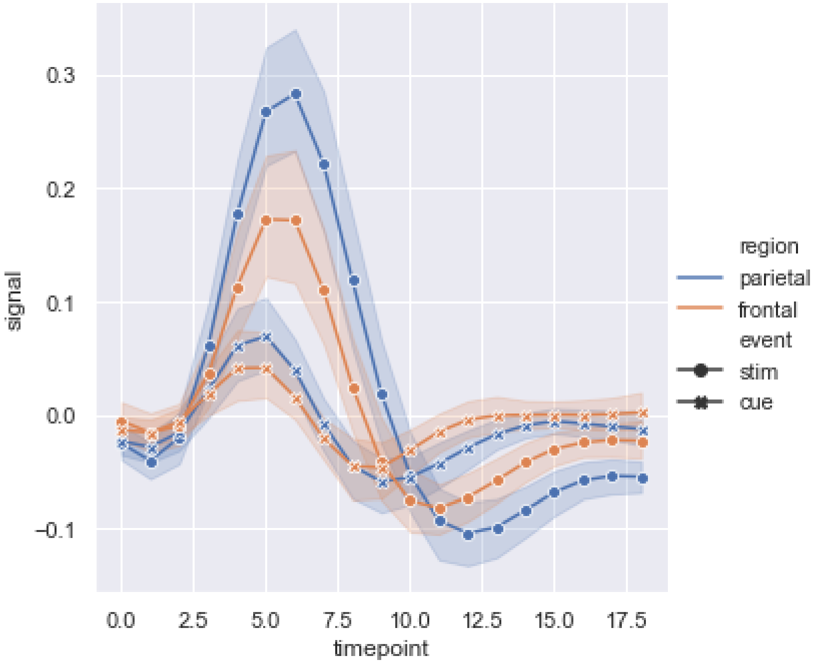
```
In [103]: sns.relplot(x = 'timepoint', y = 'signal', hue = 'region', style = 'event', kind = 'line', data = fmri)
```

```
Out[103]: <seaborn.axisgrid.FacetGrid at 0x1a1f84d0f0>
```



In [25]:
```
sns.relplot(x = 'timepoint', y = 'signal', hue = 'region', style = 'event', kind = 'line', data = fmri, markers = True
, dashes = False)
```

Out[25]:
```
<seaborn.axisgrid.FacetGrid at 0x1a18060e10>
```

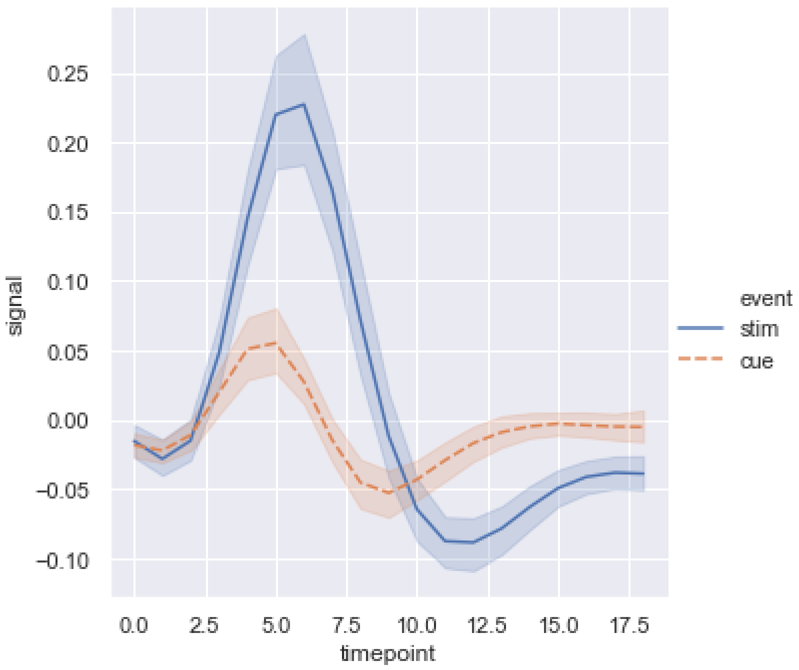


In [26]:

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

Out[26]:

```
<seaborn.axisgrid.FacetGrid at 0x1a18238668>
```



In [104]:

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

In [105]:

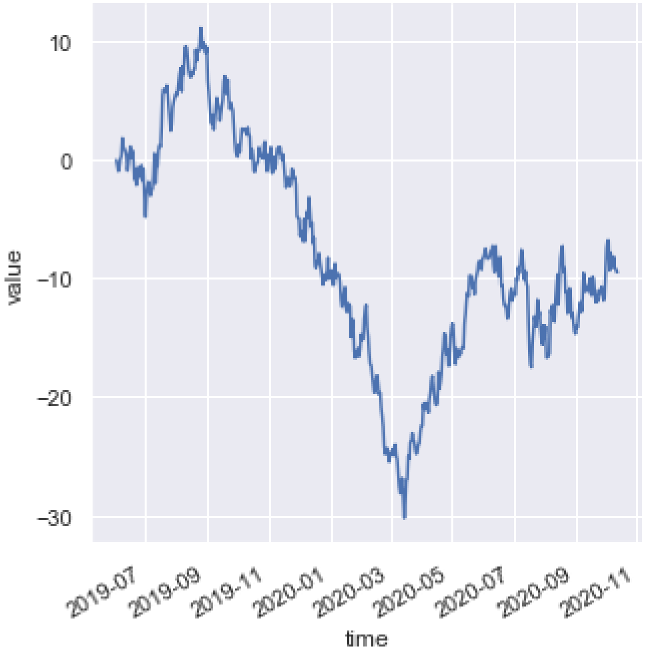
```
df.head()
```

Out[105]:

	time	value
0	2019-06-02	-0.060196
1	2019-06-03	-0.572343
2	2019-06-04	-1.002752
3	2019-06-05	-0.626627
4	2019-06-06	0.021314

In [108]:

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



In [109]:

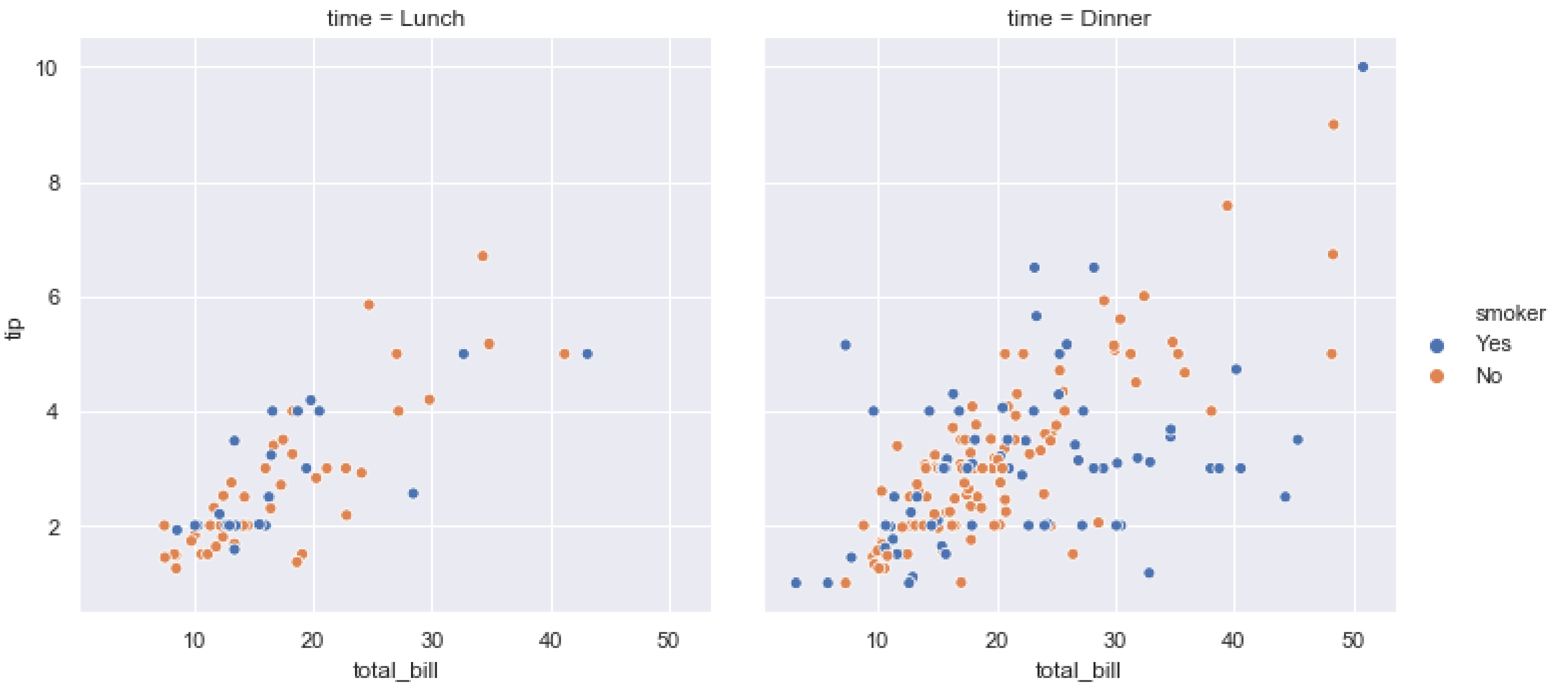
```
tips.head()
```

Out[109]:

	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

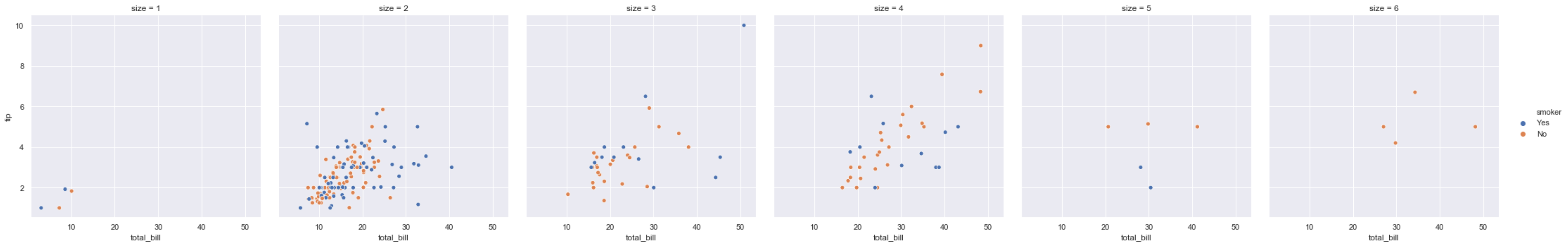
```
In [31]: sns.relplot(x = 'total_bill', y = 'tip', hue = 'smoker', col = 'time', data = tips)
```

```
Out[31]: <seaborn.axisgrid.FacetGrid at 0x1a185f4cc0>
```



```
In [32]: sns.relplot(x = 'total_bill', y = 'tip', hue = 'smoker', col = 'size', data = tips)
```

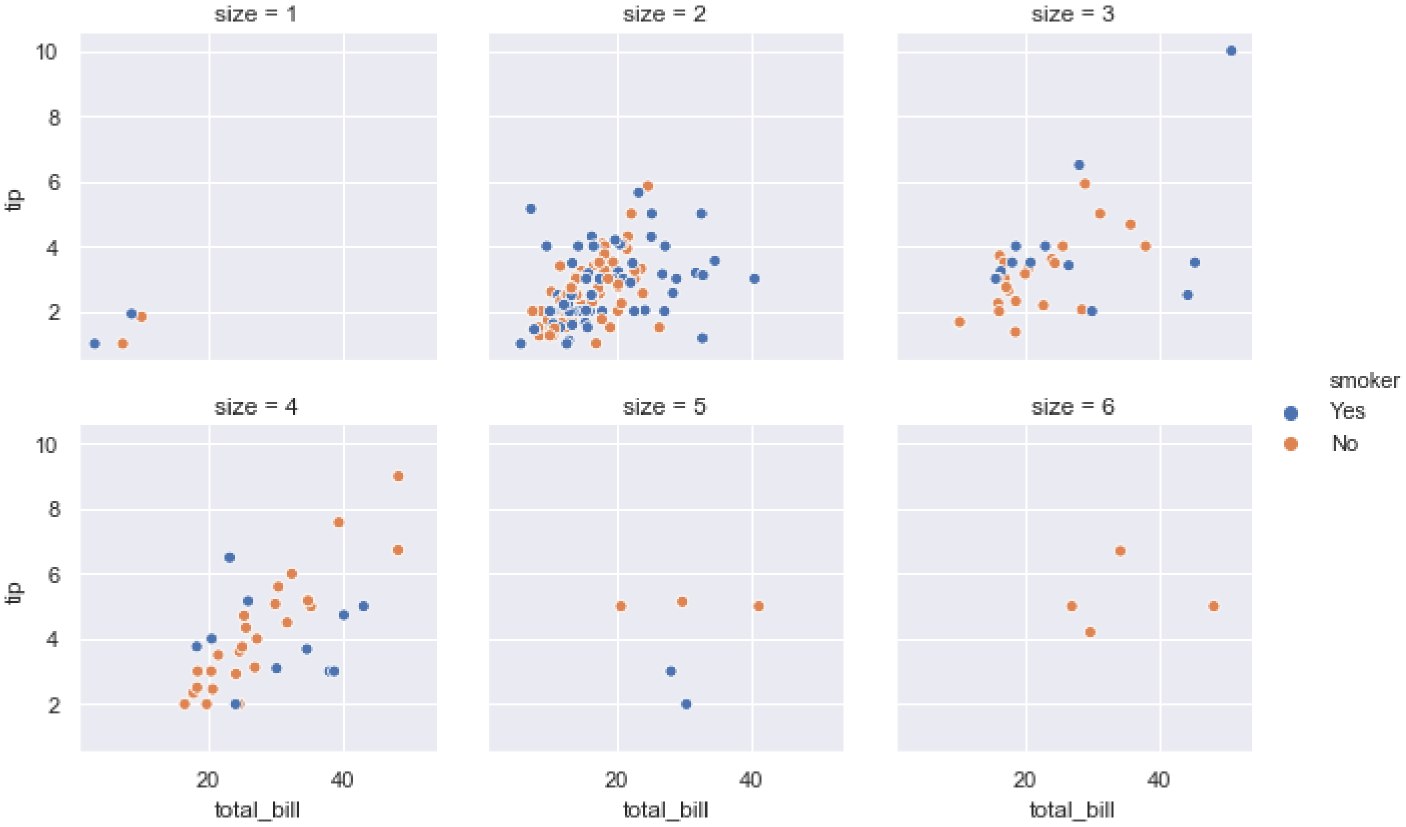
```
Out[32]: <seaborn.axisgrid.FacetGrid at 0x1a1898d048>
```



```
In [ ]:
```

In [33]:
```
sns.relplot(x = 'total_bill', y = 'tip', hue = 'smoker', col = 'size', data = tips, col_wrap=3, height=3)
```

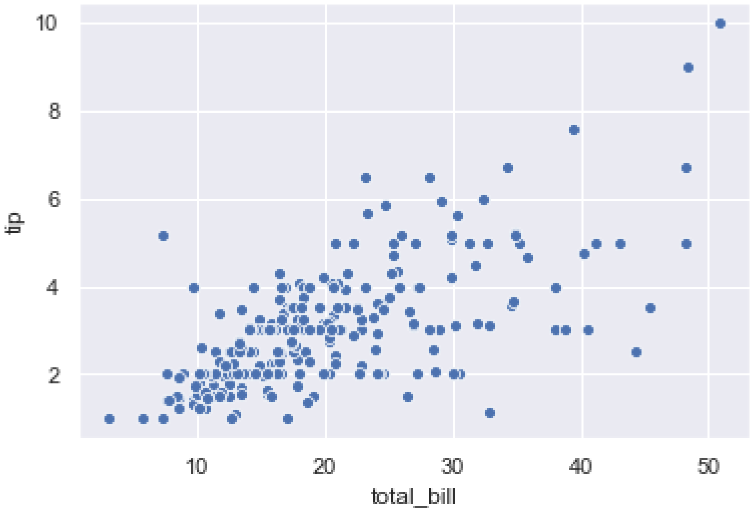
Out[33]:
```
<seaborn.axisgrid.FacetGrid at 0x1a18fa9898>
```



In [ ]:

In [137]:
```
sns.scatterplot(x = 'total_bill', y = 'tip', data = tips)
```

Out[137]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1a1f22a4a8>
```



In [ ]:
```

```

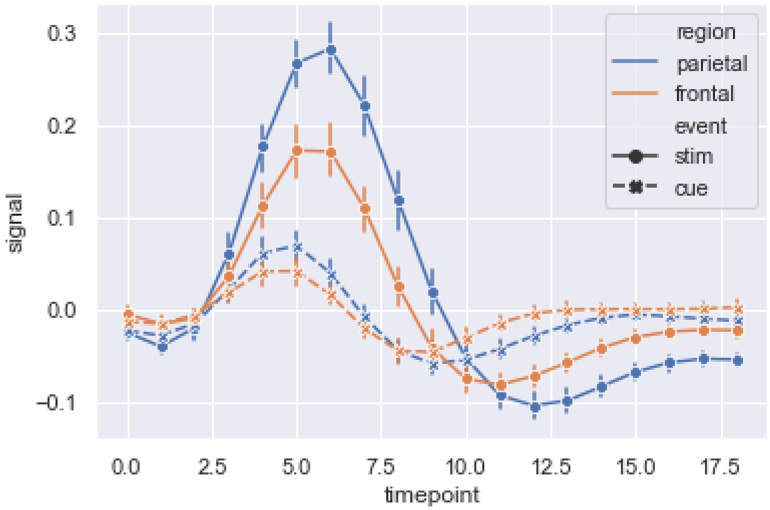
In [138]:
```
fmri.head()
```

Out[138]:

	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

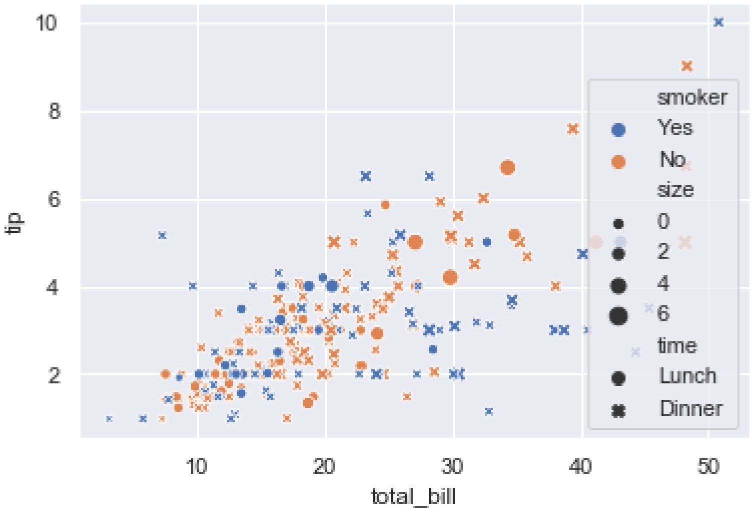
```
In [36]: sns.lineplot(x = 'timepoint', y  = 'signal', style = 'event', hue = 'region', data = fmri, markers = True, ci = 68, er
         r_style='bars')
```

```
Out[36]: <matplotlib.axes._subplots.AxesSubplot at 0x1a19a834a8>
```



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

Out[37]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1a197f6d68>
```



In [ ]:
```

```

In [139]:
```
iris = sns.load_dataset('iris')
```

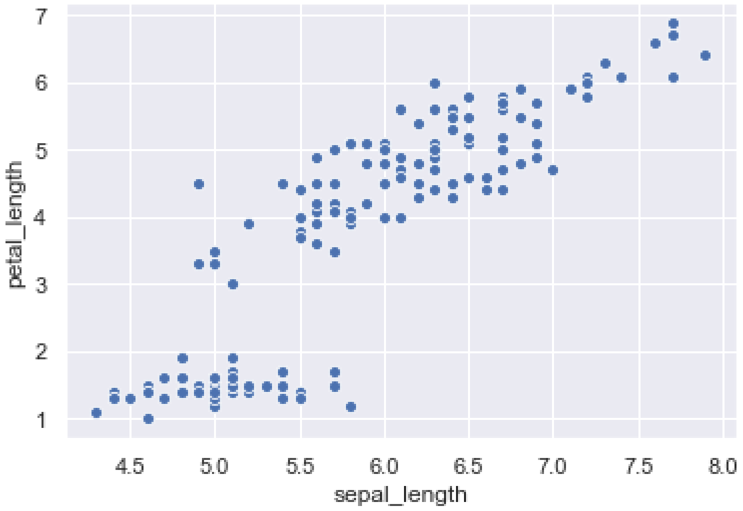
In [140]:
```
iris.head()
```

Out[140]:

	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

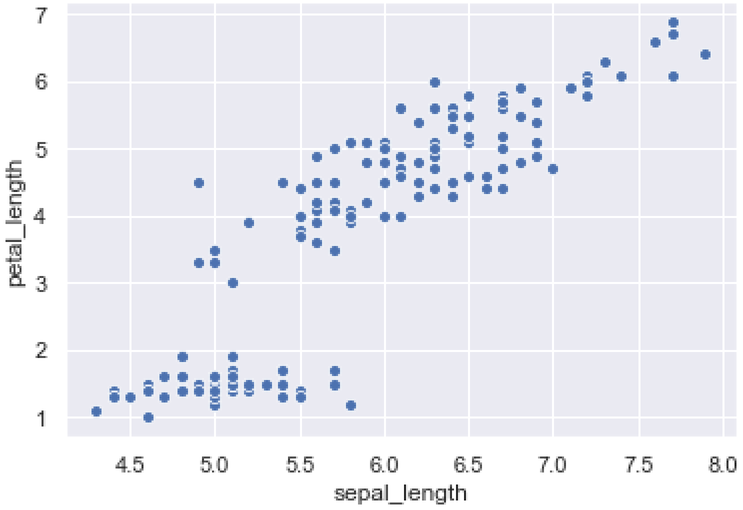
```
In [141]: sns.scatterplot(x = 'sepal_length', y = 'petal_length', data = iris)
```

```
Out[141]: <matplotlib.axes._subplots.AxesSubplot at 0x1a1f22a908>
```



```
In [142]: sns.scatterplot(x = iris['sepal_length'], y = iris['petal_length'])
```

```
Out[142]: <matplotlib.axes._subplots.AxesSubplot at 0x1a1ff3bda0>
```



2. Categorical Data Ploting

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

In [143]:

```
tips.head()
```

Out[143]:

	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 [145]:

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

In [146]:

```
titanic.head()
```

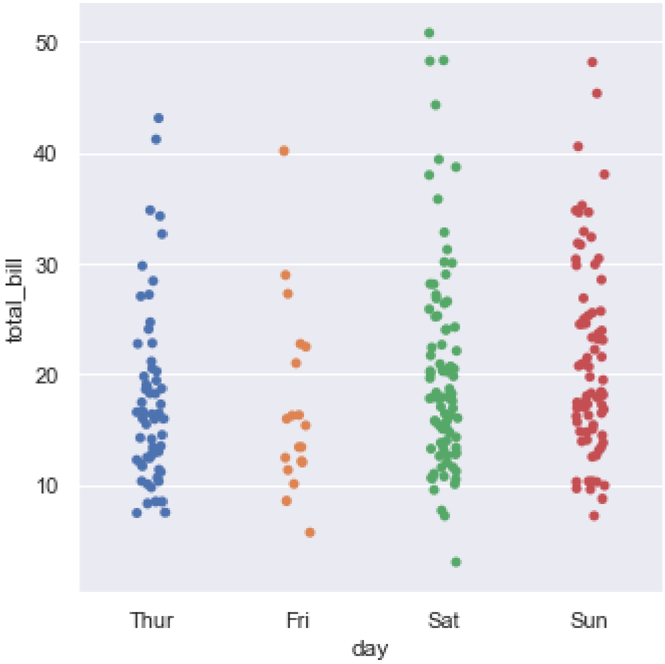
Out[146]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck	embark_town	alive	alone
0	0	3	male	22.0	1	0	7.2500	S	Third	man	True	NaN	Southampton	no	False
1	1	1	female	38.0	1	0	71.2833	C	First	woman	False	C	Cherbourg	yes	False
2	1	3	female	26.0	0	0	7.9250	S	Third	woman	False	NaN	Southampton	yes	True
3	1	1	female	35.0	1	0	53.1000	S	First	woman	False	C	Southampton	yes	False
4	0	3	male	35.0	0	0	8.0500	S	Third	man	True	NaN	Southampton	no	True

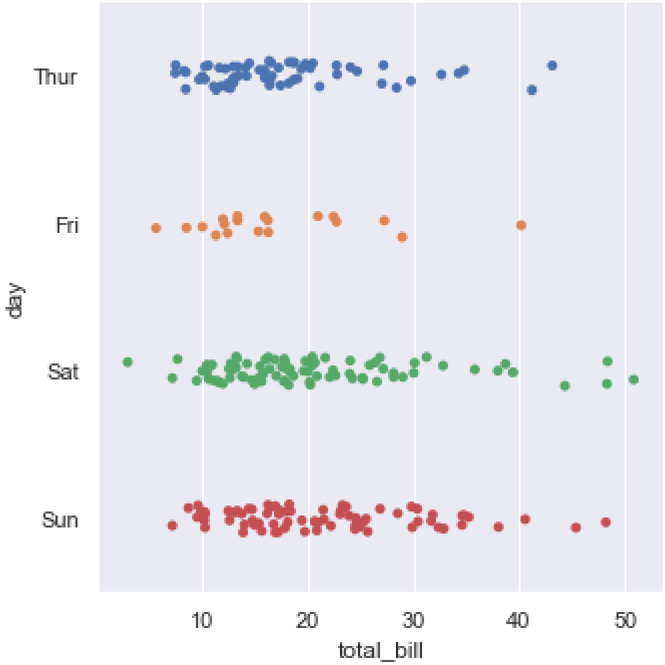
In [45]:
```
#catplot()
```

In [46]:
```
sns.catplot(x = 'day', y = 'total_bill', data = tips)
```

Out[46]:
```
<seaborn.axisgrid.FacetGrid at 0x1a19f3f208>
```

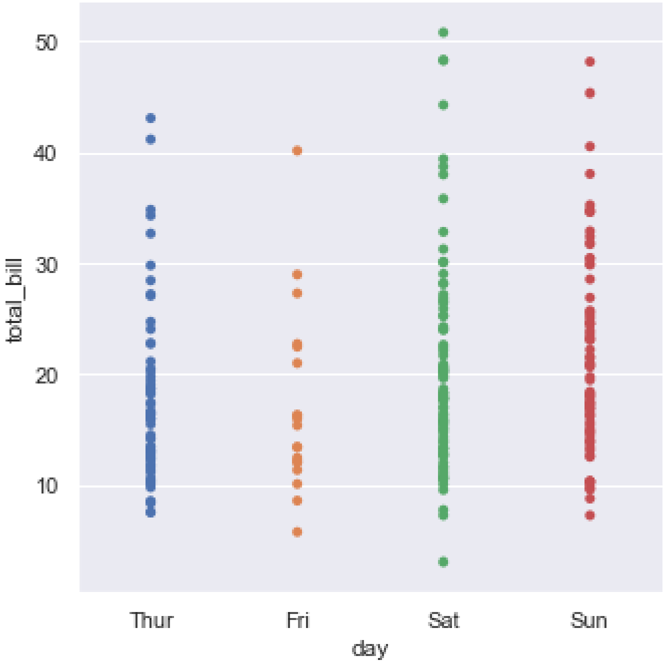


```
In [147]: sns.catplot(y = 'day', x = 'total_bill', data = tips)

Out[147]: <seaborn.axisgrid.FacetGrid at 0x1a1ea3ea20>
```



```
In [48]: sns.catplot(x = 'day', y = 'total_bill', data = tips, jitter = False)
```

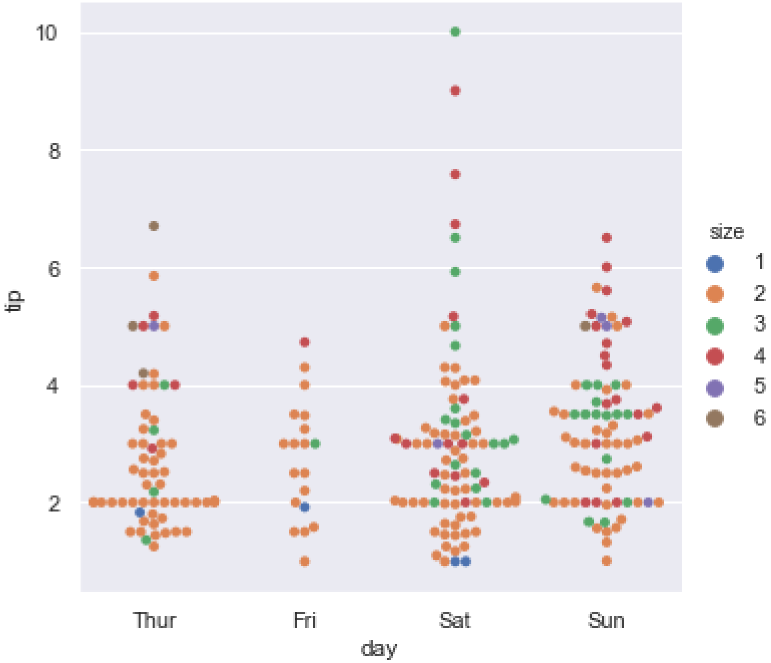
```
Out[48]: <seaborn.axisgrid.FacetGrid at 0x1a19f995c0>
```



```
In [ ]:
```

```
In [49]: sns.catplot(x = 'day', y = 'tip', data = tips, kind = 'swarm', hue = 'size')
```

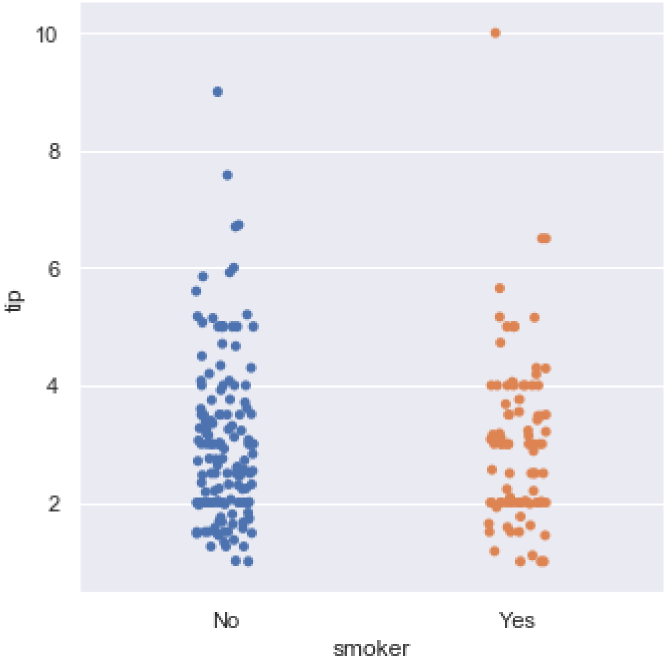
```
Out[49]: <seaborn.axisgrid.FacetGrid at 0x1a1a2d3978>
```



```
In [ ]:
```

```
In [50]: sns.catplot(x = 'smoker', y = 'tip', data = tips, order= ['No', 'Yes'])
```

```
Out[50]: <seaborn.axisgrid.FacetGrid at 0x1a1a43c1d0>
```



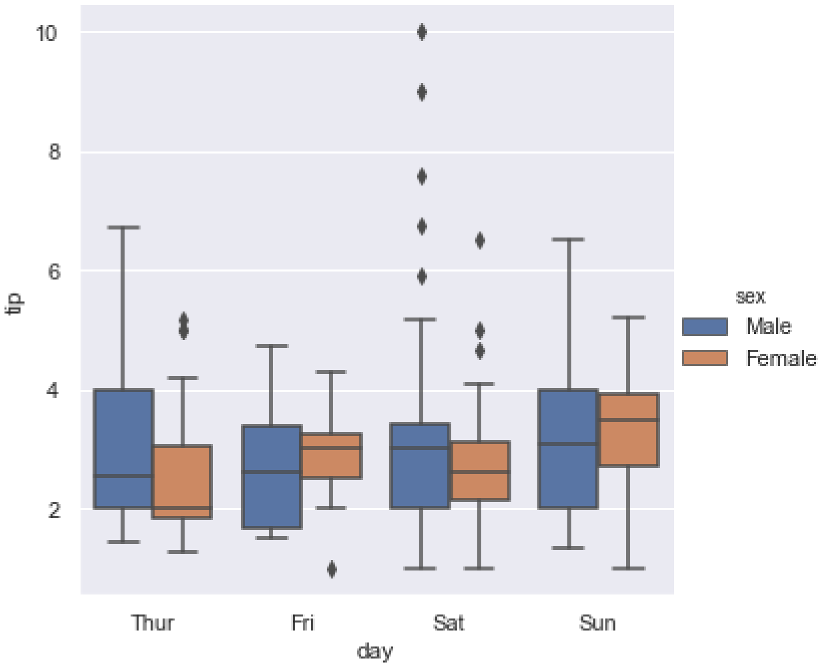
```
In [51]: tips.head()
```

Out[51]:

	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

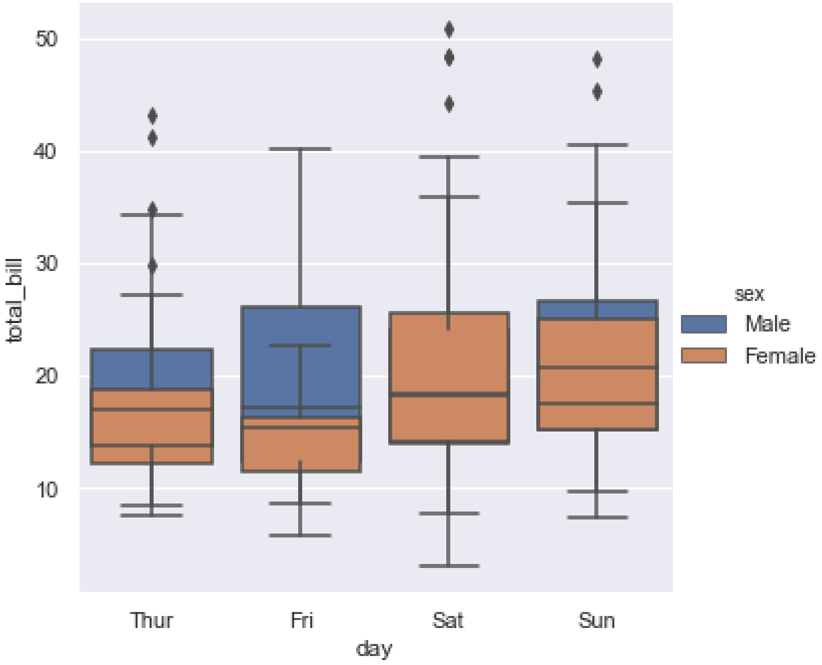
```
In [ ]:
```

```
In [52]: sns.catplot(x = 'day', y = 'tip', kind = 'box', data = tips, hue = 'sex')

Out[52]: <seaborn.axisgrid.FacetGrid at 0x1a1a78b668>
```



```
In [53]: sns.catplot(x = 'day', y = 'total_bill', kind = 'box', data = tips, hue = 'sex', dodge = False)
```

```
Out[53]: <seaborn.axisgrid.FacetGrid at 0x1a1a53c898>
```



```
In [ ]:
```

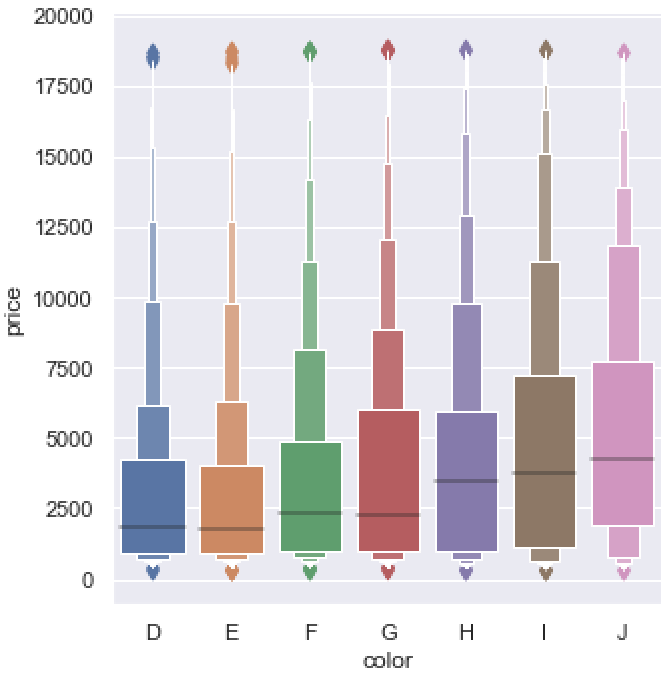
```
In [148]: diamonds = sns.load_dataset('diamonds')
          diamonds.head()
```

Out[148]:

	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

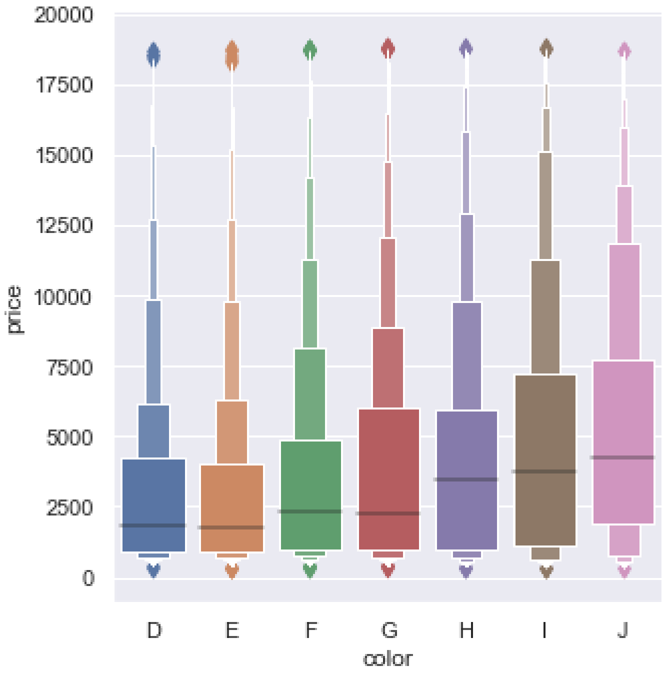
```
In [55]: sns.catplot(x = 'color', y = 'price', kind = 'boxen', data = diamonds.sort_values('color'))
```

```
Out[55]: <seaborn.axisgrid.FacetGrid at 0x1a1a77fc88>
```



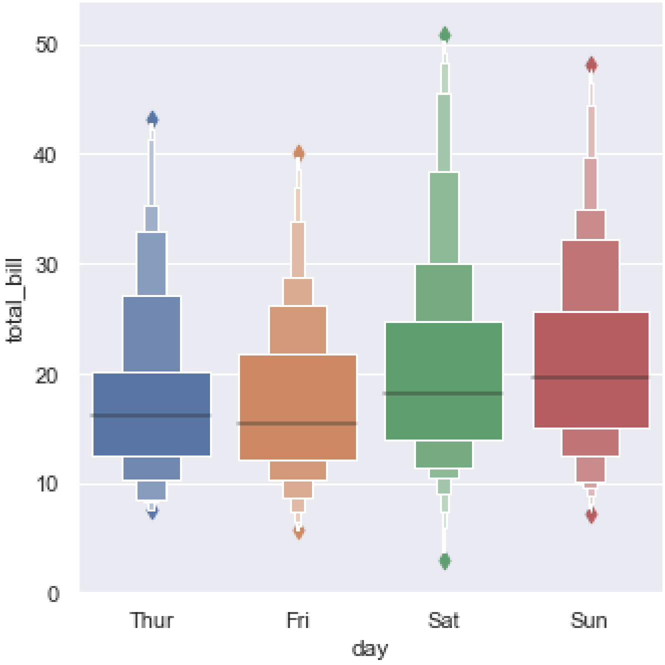
```
In [56]: sns.catplot(x = 'color', y = 'price', kind = 'boxen', data = diamonds.sort_values('color'))
```

```
Out[56]: <seaborn.axisgrid.FacetGrid at 0x1a1ac8e278>
```



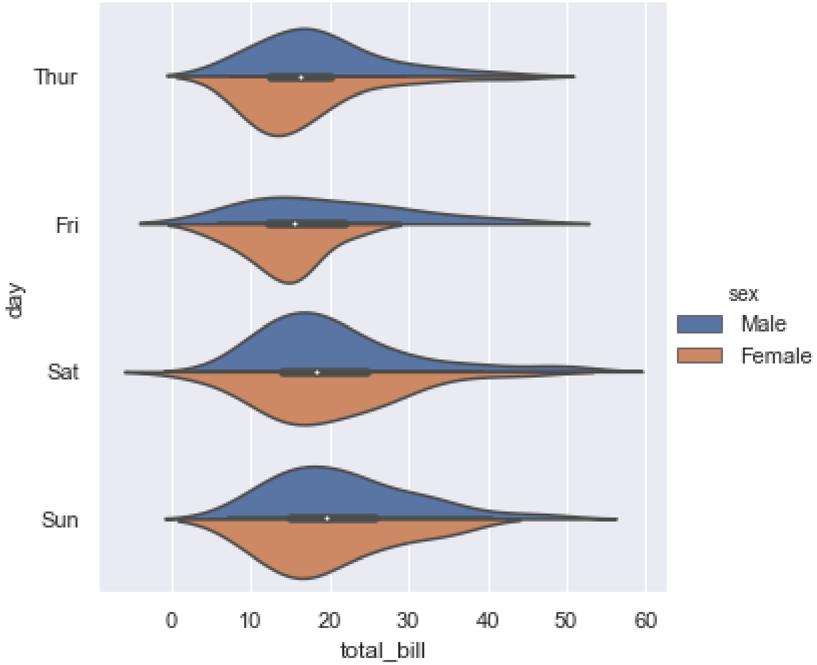
```
In [57]: sns.catplot(x = 'day', y = 'total_bill', kind = 'boxen', data = tips, dodge = False)
```

```
Out[57]: <seaborn.axisgrid.FacetGrid at 0x1a1cb035c0>
```



```
In [ ]:
```

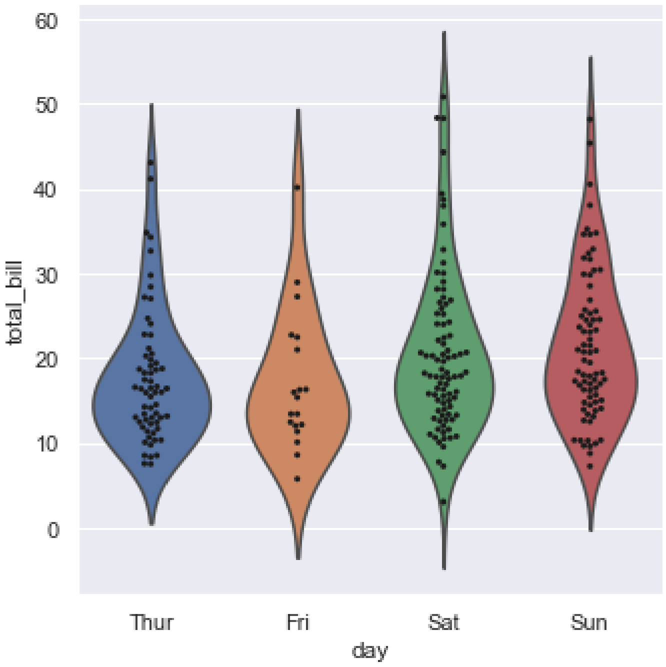
```
In [58]: sns.catplot(x = 'total_bill', y = 'day', hue = 'sex', kind = 'violin', data = tips, split = True,)

Out[58]: <seaborn.axisgrid.FacetGrid at 0x1a1caf9860>
```



```
In [ ]:
```

In [59]:
```
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.ax)
```

Out[59]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1a1cb1b550>
```



In [149]:
```
titanic.head()
```

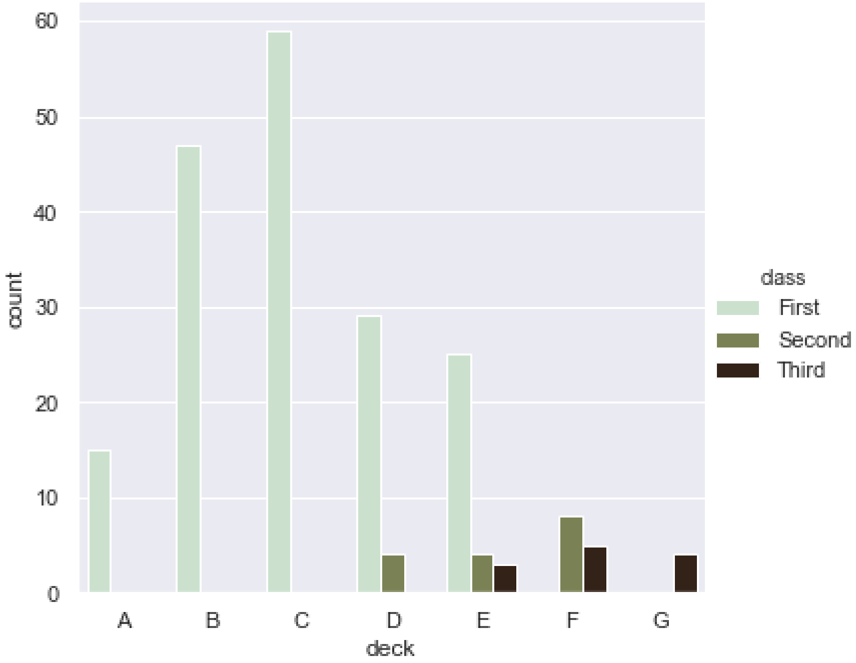
Out[149]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck	embark_town	alive	alone
0	0	3	male	22.0	1	0	7.2500	S	Third	man	True	NaN	Southampton	no	False
1	1	1	female	38.0	1	0	71.2833	C	First	woman	False	C	Cherbourg	yes	False
2	1	3	female	26.0	0	0	7.9250	S	Third	woman	False	NaN	Southampton	yes	True
3	1	1	female	35.0	1	0	53.1000	S	First	woman	False	C	Southampton	yes	False
4	0	3	male	35.0	0	0	8.0500	S	Third	man	True	NaN	Southampton	no	True

In [ ]:

```
In [61]: sns.catplot(x = 'deck', kind = 'count', palette = 'ch:0.95', data = titanic, hue = 'class')
```

```
Out[61]: <seaborn.axisgrid.FacetGrid at 0x1a1bddb908>
```



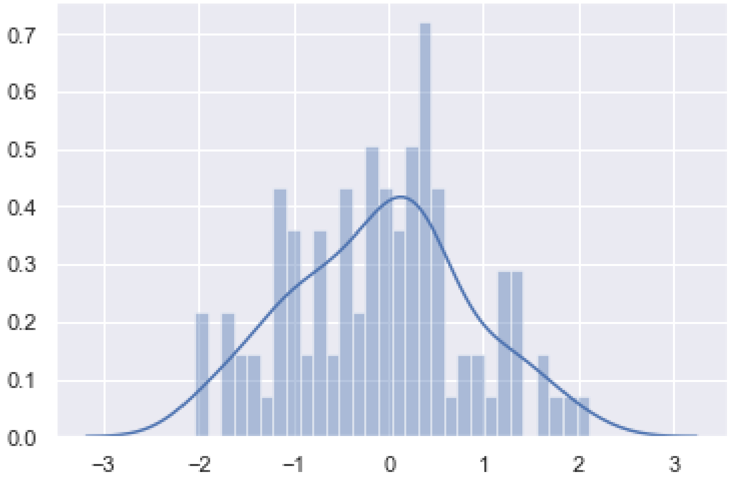
```
In [ ]:
```

3. Visualizing Distribution of the Data

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

```
In [150]: x = randn(100)
In [152]: sns.distplot(x, kde = True, hist = True, rug= False, bins= 30)
```

```
Out[152]: <matplotlib.axes._subplots.AxesSubplot at 0x1a22decf98>
```



```
In [153]: tips.head()
```

Out[153]:

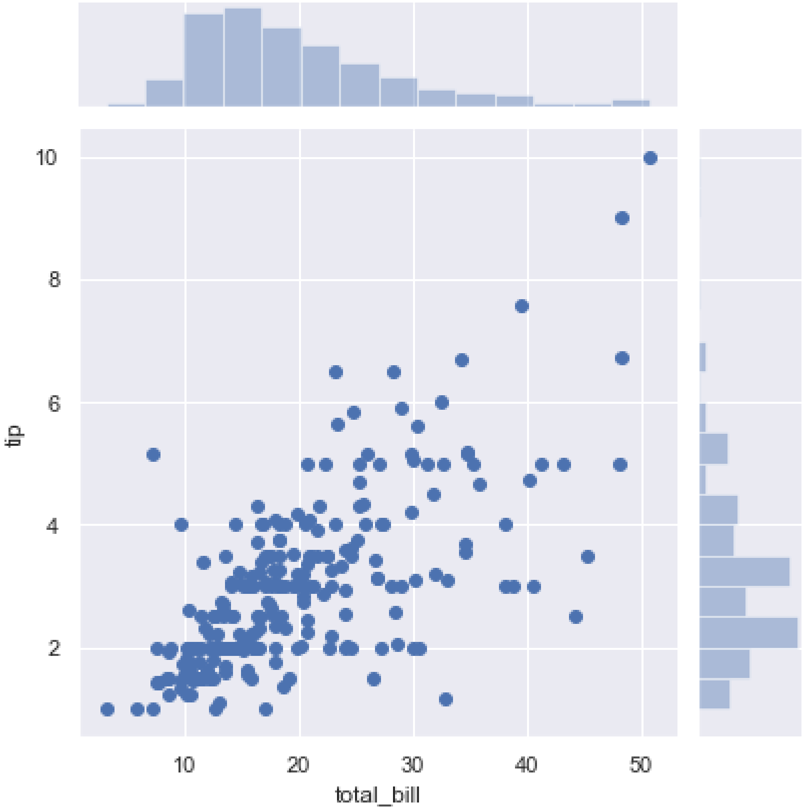
	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4

```
In [65]: x = tips['total_bill']
         y = tips['tip']
```

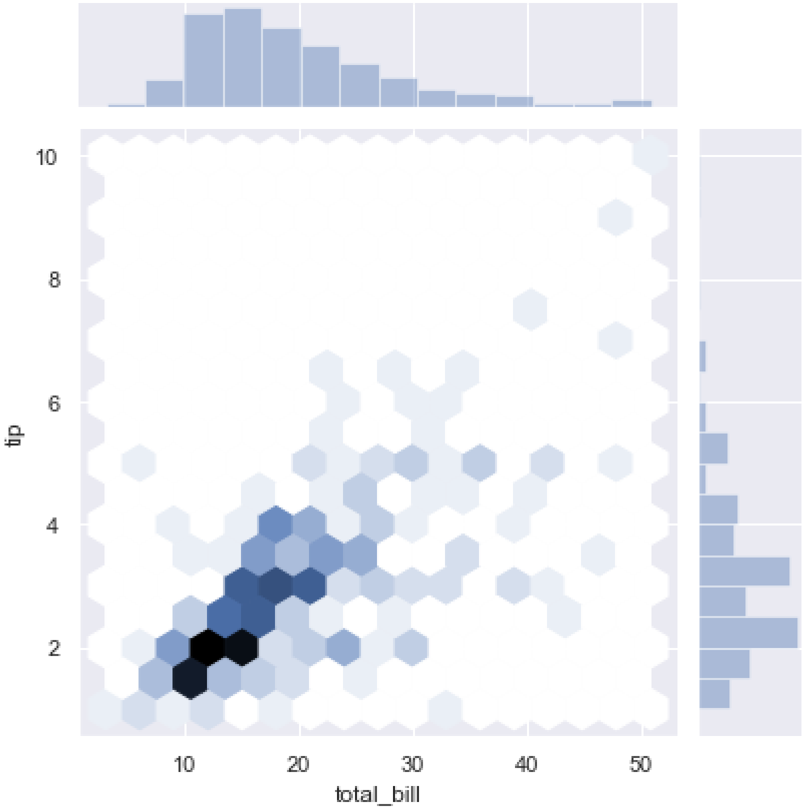
```
In [ ]:
```

```
In [66]: sns.jointplot(x = x, y=y)
```

```
Out[66]: <seaborn.axisgrid.JointGrid at 0x1a1c477ef0>
```



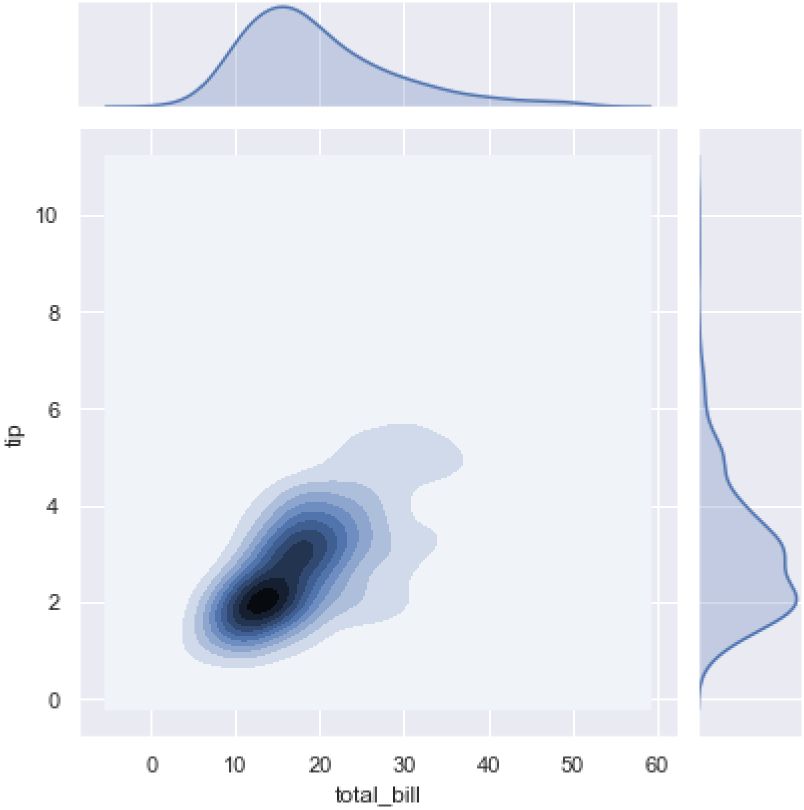
```
In [67]: sns.set()
         sns.jointplot(x = x, y=y, kind = 'hex')

Out[67]: <seaborn.axisgrid.JointGrid at 0x1a1c3a34a8>
```



```
In [ ]:
```

```
In [68]: sns.jointplot(x = x, y = y, kind = 'kde')
```

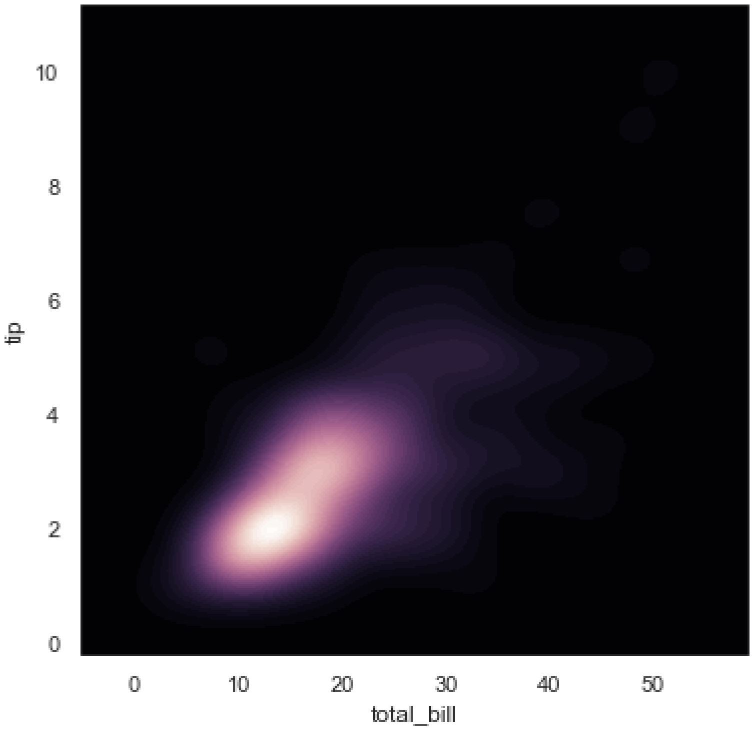
```
Out[68]: <seaborn.axisgrid.JointGrid at 0x1a1cf7b630>
```



```
In [ ]:
```

In [69]:
```
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)
```

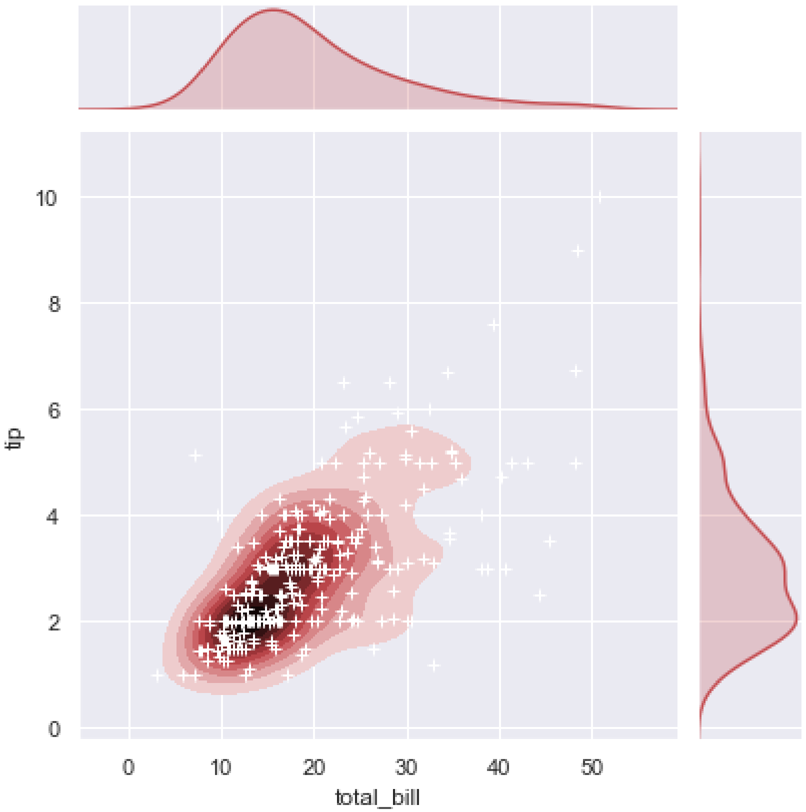
Out[69]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1a1d2a9208>
```



In [ ]:
```
```

In [70]:

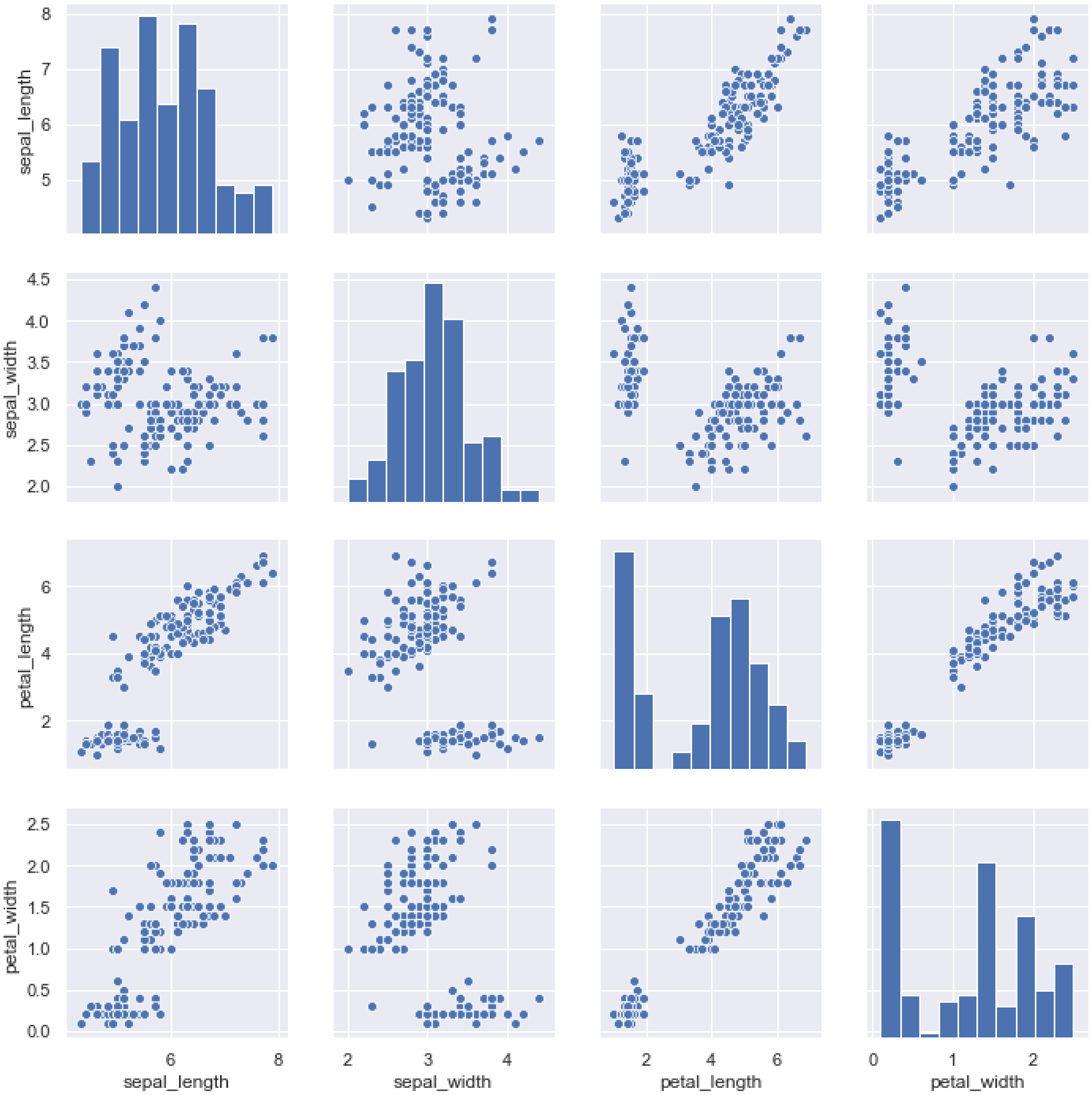
```
g = sns.jointplot(x, y, kind = 'kde', color = 'r')
g.plot_joint(plt.scatter, c = 'w', s = 30, linewidth = 1, marker = '+')
g.ax_joint.collections[0].set_alpha(0)
```



In [71]:

```
sns.pairplot(iris)
```

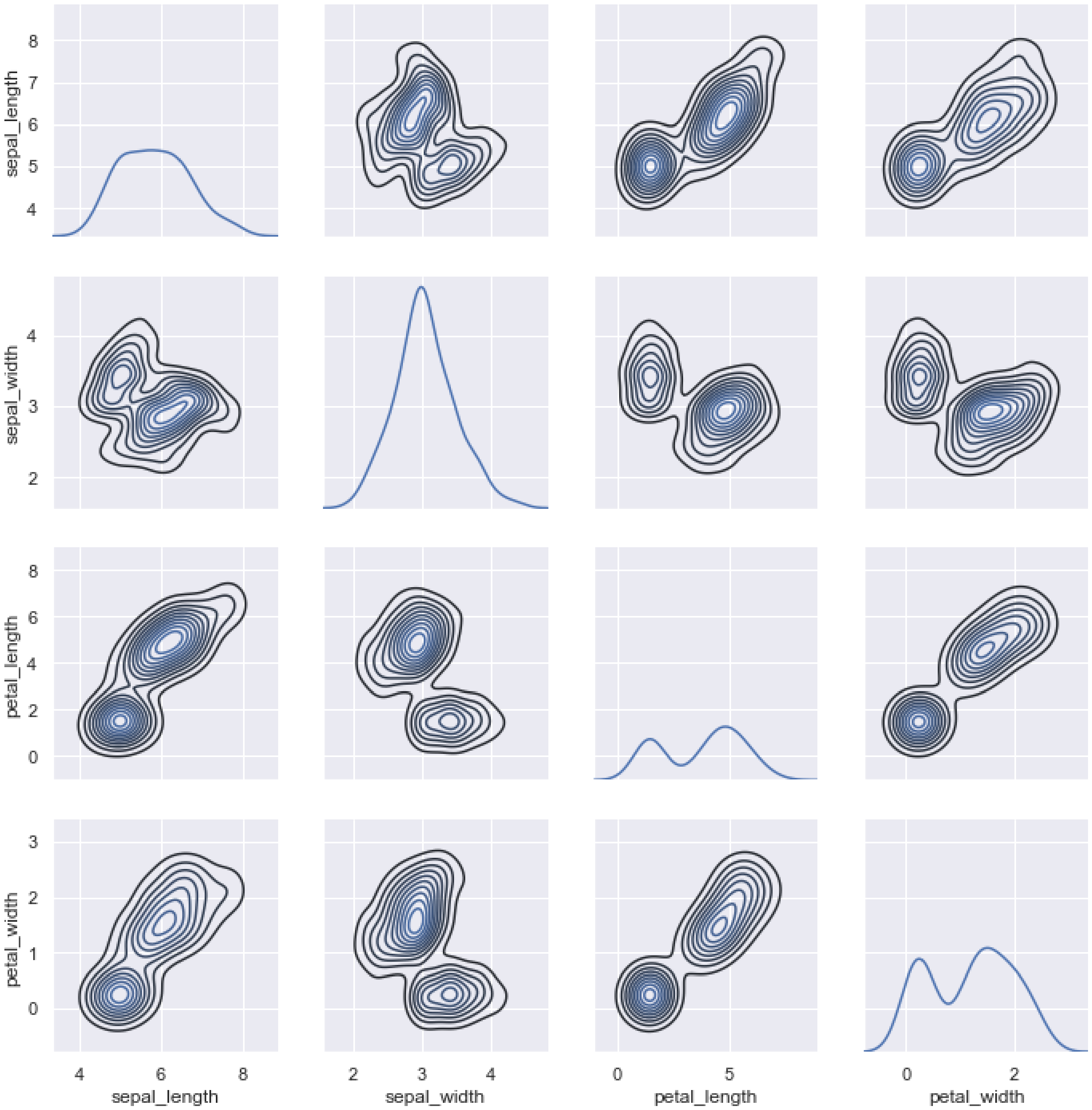
```
Out[71]: <seaborn.axisgrid.PairGrid at 0x1a1d77e4a8>
```


sepal_length
8
7
6
5
sepal_width
4.5
4.0
3.5
3.0
2.5
2.0
petal_length
6
4
2
petal_width
2.5
2.0
1.5
1.0
0.5
0.0
6
8
sepal_length
2
3
4
sepal_width
2
4
6
petal_length
0
1
2
petal_width

In [72]:

```
g = sns.PairGrid(iris)
g.map_diag(sns.kdeplot)
g.map_offdiag(sns.kdeplot, n_levels = 10)
```

```
Out[72]: <seaborn.axisgrid.PairGrid at 0x1a1bd57438>
```


sepal_length
8
7
6
5
4
sepal_width
4
3
2
petal_length
8
6
4
2
0
petal_width
3
2
1
0
4
6
8
sepal_length
2
3
4
sepal_width
0
5
petal_length
0
2
petal_width

```
In [ ]:
```

4. Linear Regression and Relationship

- regplot()
- lmplot()

```
In [154]: tips.head()
```

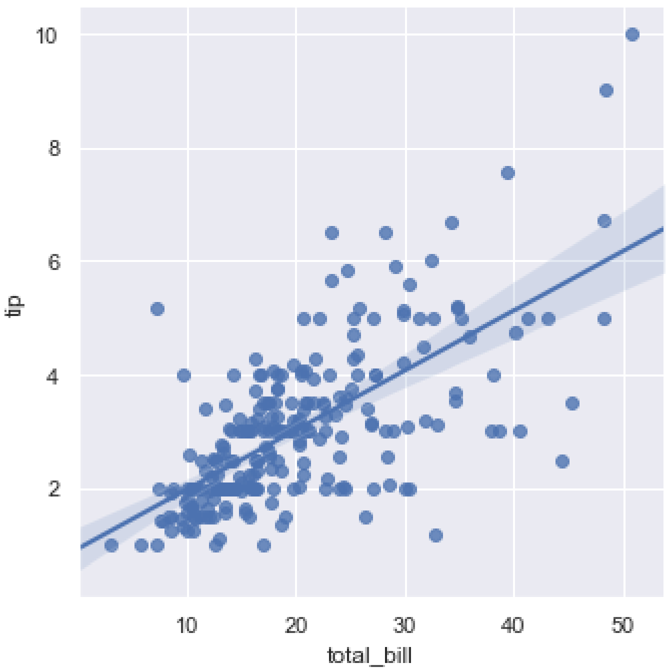
```
Out[154]:
   total_bill   tip     sex  smoker  day    time  size
0       16.99  1.01  Female      No  Sun  Dinner     2
1       10.34  1.66    Male      No  Sun  Dinner     3
2       21.01  3.50    Male      No  Sun  Dinner     3
3       23.68  3.31    Male      No  Sun  Dinner     2
4       24.59  3.61  Female      No  Sun  Dinner     4
```

In [74]:

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

Out[74]:

```
<seaborn.axisgrid.FacetGrid at 0x1a1e6e55c0>
```



In [ ]:

```

```

In [155]:

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

Out[155]:

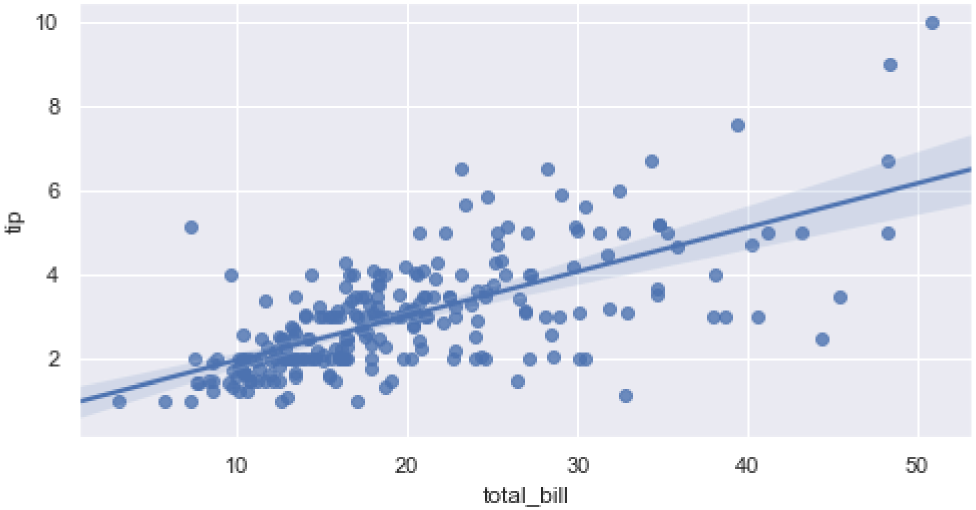
	dataset	x	y
0	I	10.0	8.04
1	I	8.0	6.95
2	I	13.0	7.58
3	I	9.0	8.81
4	I	11.0	8.33

In [156]:
```
data['dataset'].value_counts()
```

Out[156]:
```
III     11
IV      11
II      11
I       11
Name: dataset, dtype: int64
```

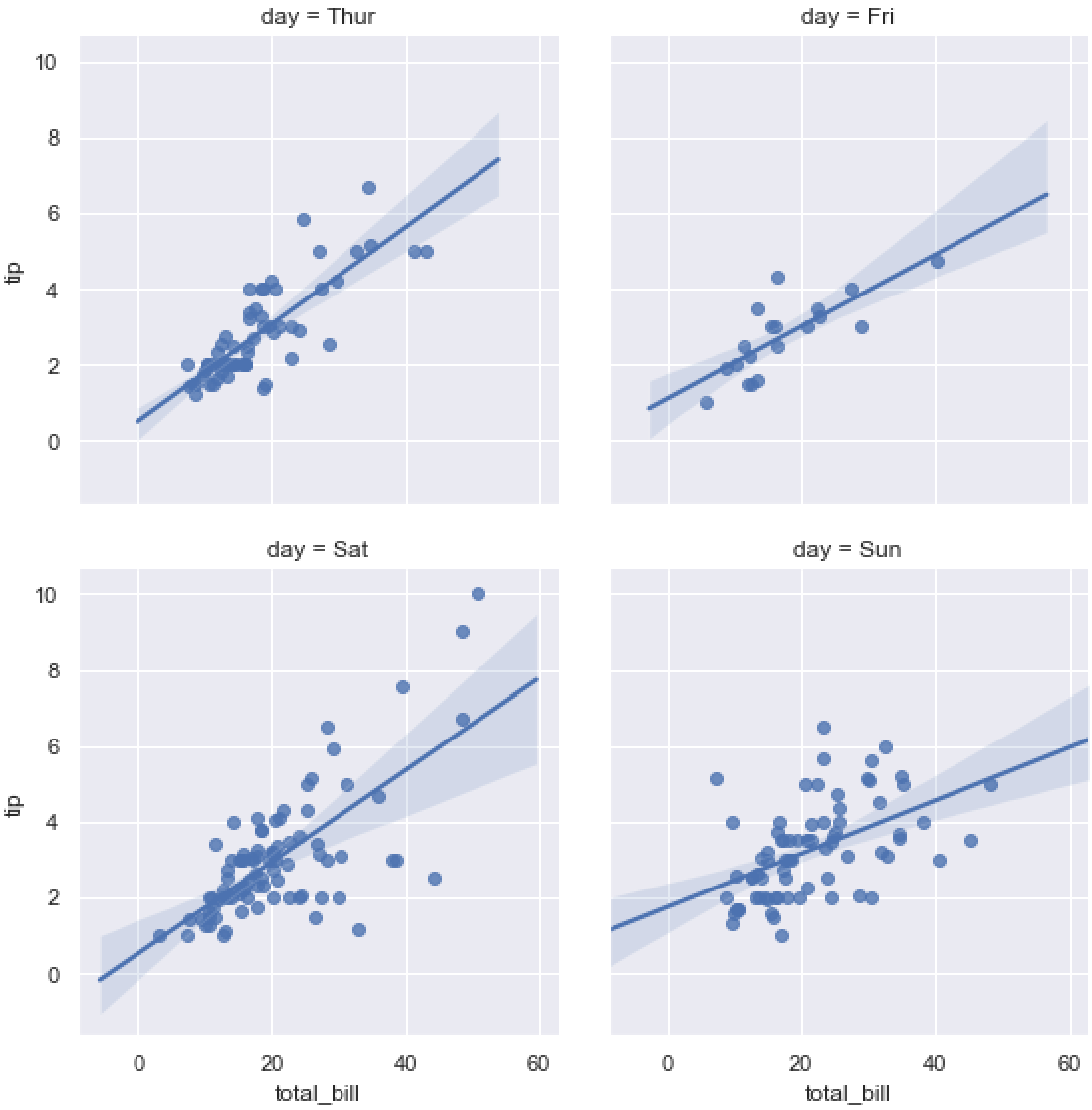
In [157]:
```
f, ax = plt.subplots(figsize = (8,4))
sns.regplot(x = 'total_bill', y = 'tip', data = tips, ax = ax)
```

Out[157]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1a24e0dcc0>
```



In [ ]:

```
In [78]: sns.lmplot(x = 'total_bill', y = 'tip', data = tips, col = 'day', col_wrap=2, height = 4)

Out[78]: <seaborn.axisgrid.FacetGrid at 0x1a1d716ba8>
```



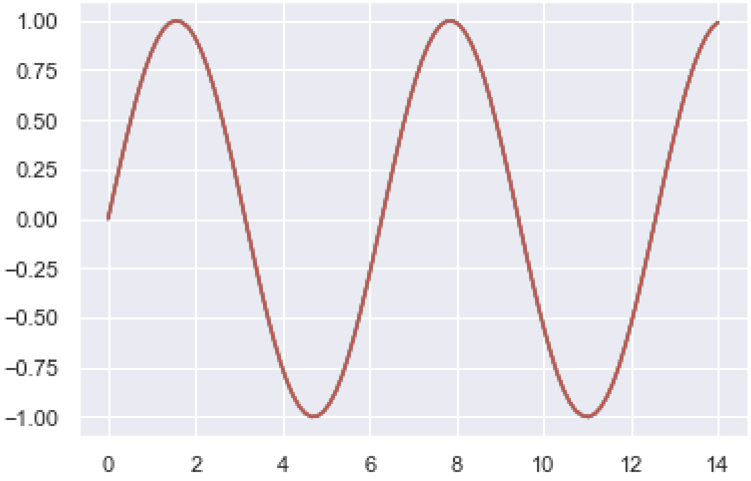
```
In [ ]:
```

5. Controlling Ploted Figure Aesthetics

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

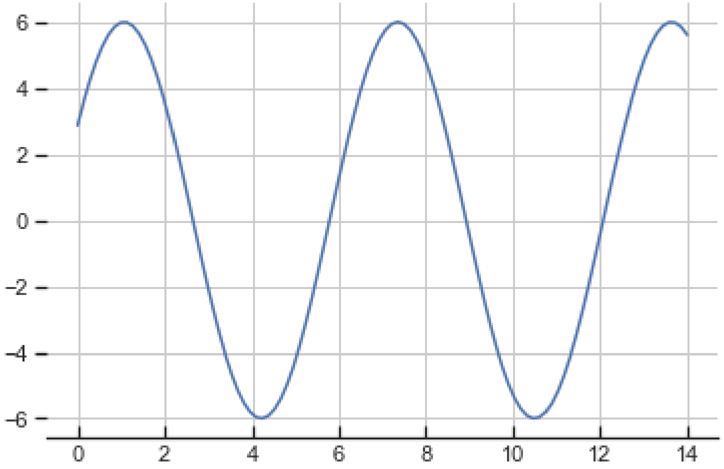
```
In [161]: def sinplot():
              x = np.linspace(0, 14, 100)
              for i in range(1, 5):
                  plt.plot(x, np.sin(x+i*0.5)*(7-i))
```

```
In [162]: sinplot()
```



```
In [ ]:
```

In [81]:
```
sns.set_style('ticks', {'axes.grid': True, 'xtick.direction': 'in'})
sinplot()
sns.despine(left = True, bottom= False)
```



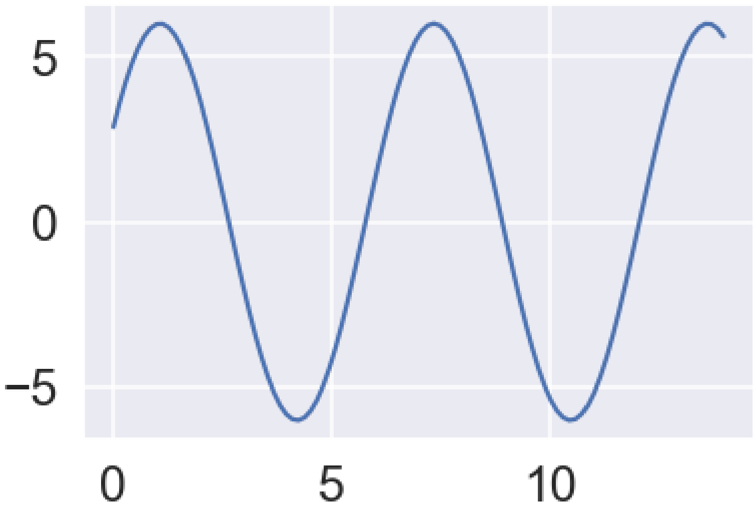
In [82]:
```
sns.axes_style()
```

Out[82]:
```
{'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',
 'image.cmap': 'rocket',
 'font.family': ['sans-serif'],
 'font.sans-serif': ['Arial',
  'DejaVu Sans',
  'Liberation Sans',
  'Bitstream Vera Sans',
  'sans-serif'],
 'patch.force_edgecolor': True,
 '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 [83]:
```
sns.set_style('darkgrid')
```

In [84]:

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

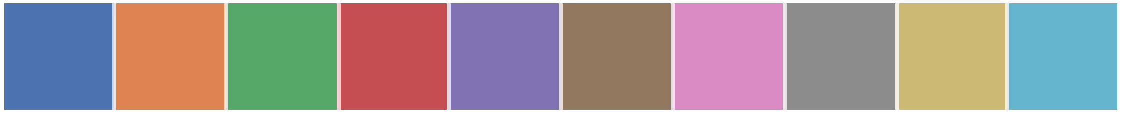


In [ ]:

```

```

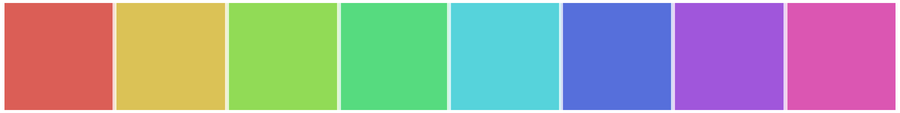
In [85]:

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



In [86]:

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



```
In [ ]:
```