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..

Entrée [3]:

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

Entrée [ ]:

```

```

Entrée [4]:

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

Entrée [7]:

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

Out[7]:

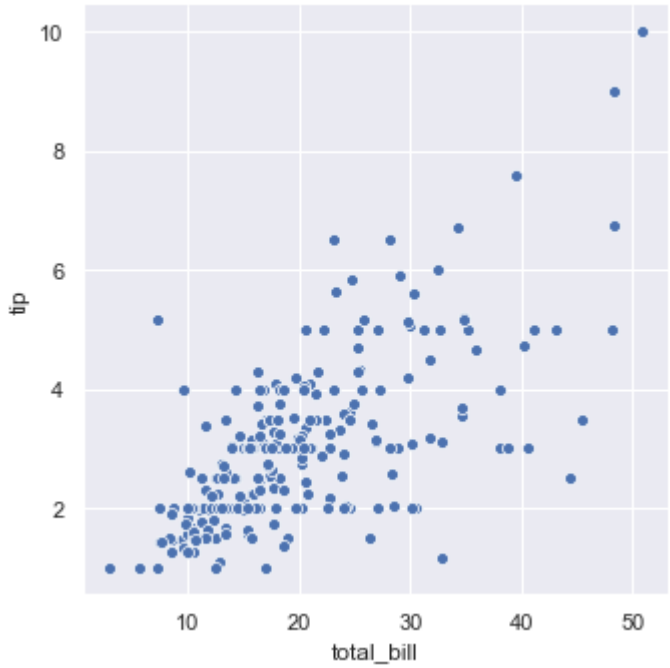
	total_bill	tip	sex	smoker	day	time	size
239	29.03	5.92	Male	No	Sat	Dinner	3
240	27.18	2.00	Female	Yes	Sat	Dinner	2
241	22.67	2.00	Male	Yes	Sat	Dinner	2
242	17.82	1.75	Male	No	Sat	Dinner	2
243	18.78	3.00	Female	No	Thur	Dinner	2

Entrée [8]:

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

Out[8]:

```
<seaborn.axisgrid.FacetGrid at 0x1f7c9fddd30>
```



Entrée [9]:

```
dir(sns.FacetGrid)
```

Out[9]:

```
['__class__',
 '__delattr__',
 '__dict__',
 '__dir__',
 '__doc__',
 '__eq__',
 '__format__',
 '__ge__',
 '__getattribute__',
 '__gt__',
 '__hash__',
 '__init__',
 '__init_subclass__',
 '__le__',
 '__lt__',
 '__module__',
 '__ne__',
 '__new__',
 '__reduce__',
 '__reduce_ex__',
 '__repr__',
 '__setattr__',
 '__sizeof__',
 '__str__',
 '__subclasshook__',
 '__weakref__',
 '_bottom_axes',
 '_clean_axis',
 '_facet_color',
 '_facet_plot',
 '_finalize_grid',
 '_get_palette',
 '_inner_axes',
 '_left_axes',
 '_legend_out',
 '_margin_titles',
 '_not_bottom_axes',
 '_not_left_axes',
 '_update_legend_data',
 'add_legend',
 'ax',
 'despine',
 'facet_axis',
 'facet_data',
 'map',
 'map_dataframe',
 'savefig',
 'set',
 'set_axis_labels',
 'set_titles',
 'set_xlabels',
 'set_xticklabels',
 'set_ylabels',
 'set_yticklabels']
```

Entrée [11]:

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

Out[11]:

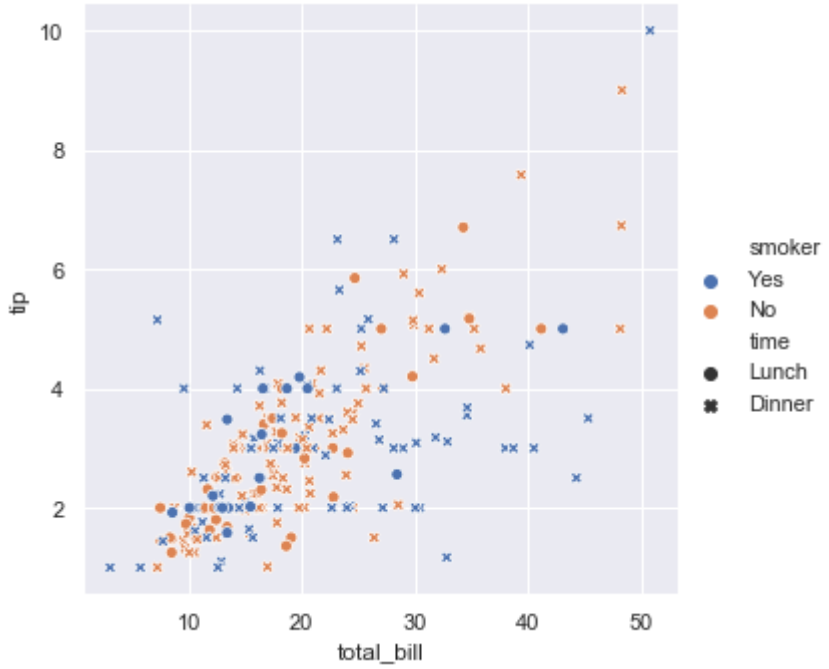
```
No     151
Yes     93
Name: smoker, dtype: int64
```

Entrée [15]:

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

Out[15]:

```
<seaborn.axisgrid.FacetGrid at 0x1f7cc3a7f28>
```

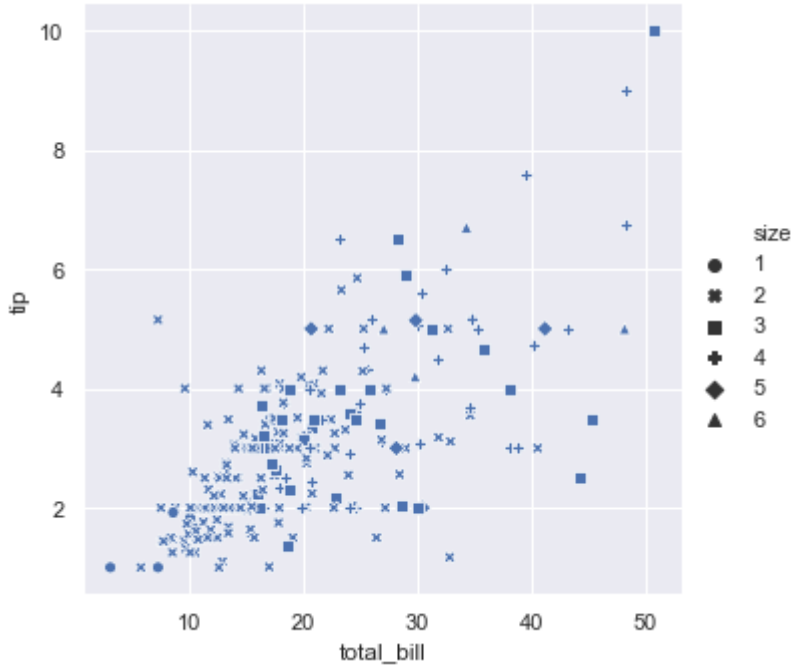


Entrée [21]:

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

Out[21]:

```
<seaborn.axisgrid.FacetGrid at 0x1f7cc72bb00>
```



Entrée [ ]:

```

```

Entrée [19]:

```
sns.relplot(x = 'total_bill', y = 'tip', hue = 'size', data = tips, palette = 'ch:r=-0.8, l
```

Out[19]:

```
<seaborn.axisgrid.FacetGrid at 0x1f7cc6404e0>
```

Entrée [20]:

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

Out[20]:

```
<seaborn.axisgrid.FacetGrid at 0x1f7cc69a240>
```

Entrée [ ]:

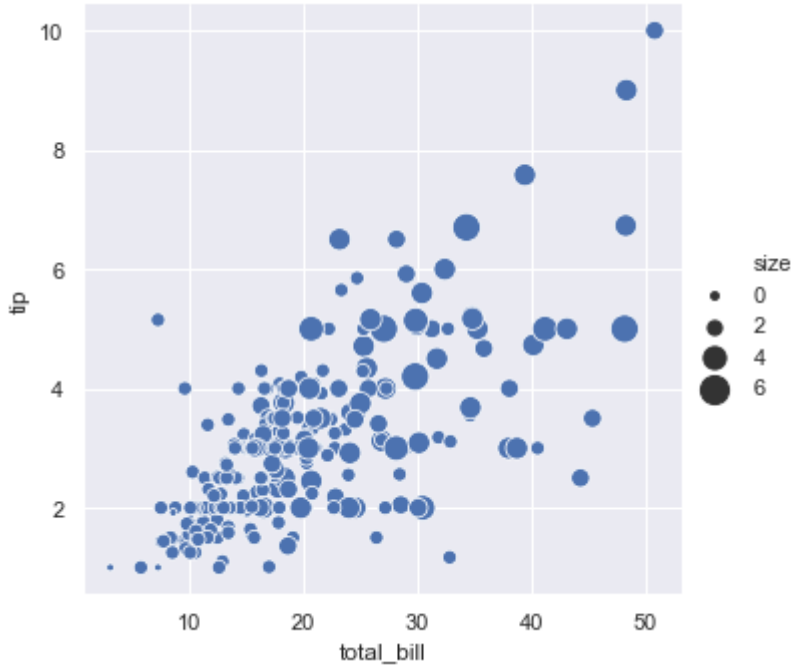
```

```

Entrée [29]:

```
sns.relplot(x = 'total_bill', y = 'tip', data = tips, size = 'size', sizes = (15, 200))
```

Out[29]:

```
<seaborn.axisgrid.FacetGrid at 0x1f7cc346f28>
```



Entrée [ ]:

```

```

Entrée [25]:

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

Entrée [26]:

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

Entrée [27]:

```
df.head()
```

Out[27]:

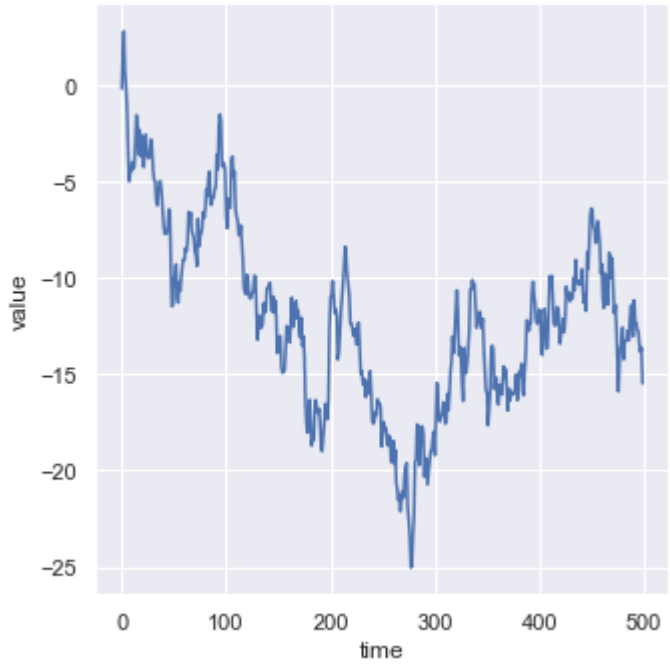
	time	value
0	0	-0.182164
1	1	1.608222
2	2	2.797099
3	3	0.714776
4	4	-0.064476

Entrée [32]:

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

Out[32]:

```
<seaborn.axisgrid.FacetGrid at 0x1f7cdde89b0>
```



Entrée [ ]:

```

```

Entrée [33]:

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

Entrée [34]:

```
df.head()
```

Out[34]:

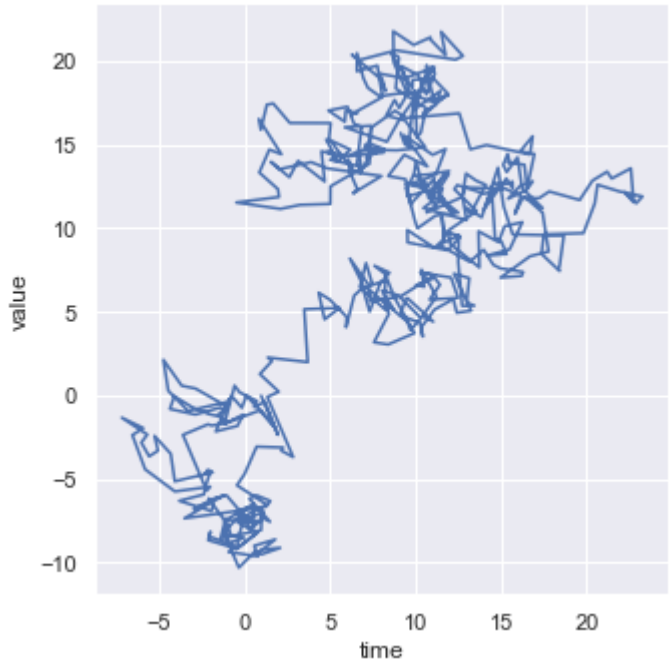
	time	value
0	0.964997	-0.035917
1	1.948768	-2.369613
2	1.186185	-0.403826
3	0.547308	-0.710893
4	0.711633	-0.354215

Entrée [36]:

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

Out[36]:

```
<seaborn.axisgrid.FacetGrid at 0x1f7cdcc7588>
```



Entrée [ ]:

```

```

Entrée [37]:

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

Out[37]:

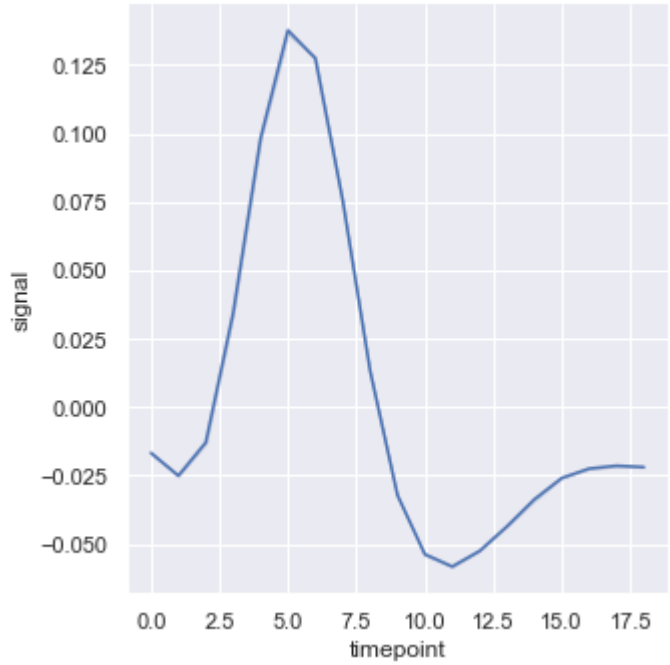
	subject	timepoint	event	region	signal
0	s13	18	stim	parietal	-0.017552
1	s5	14	stim	parietal	-0.080883
2	s12	18	stim	parietal	-0.081033
3	s11	18	stim	parietal	-0.046134
4	s10	18	stim	parietal	-0.037970

Entrée [39]:

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

Out[39]:

```
<seaborn.axisgrid.FacetGrid at 0x1f7cdab8e80>
```

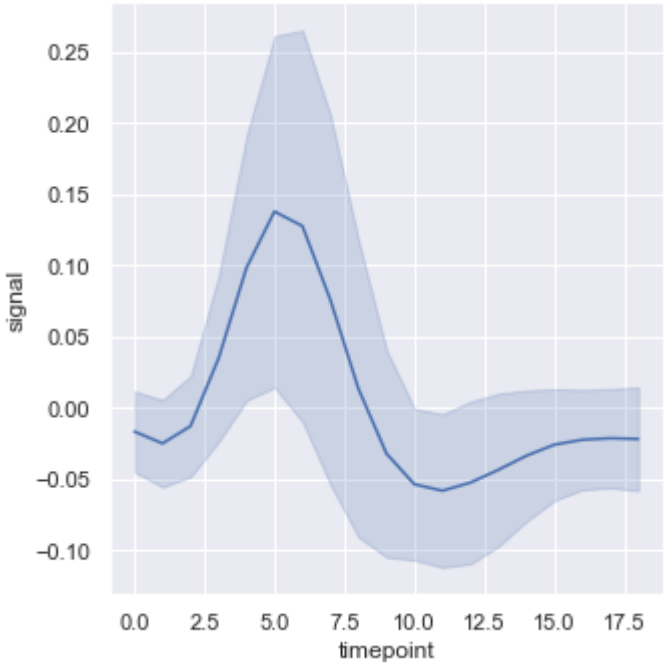


Entrée [41]:

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

Out[41]:

```
<seaborn.axisgrid.FacetGrid at 0x1f7ce3fe6a0>
```



Entrée [ ]:

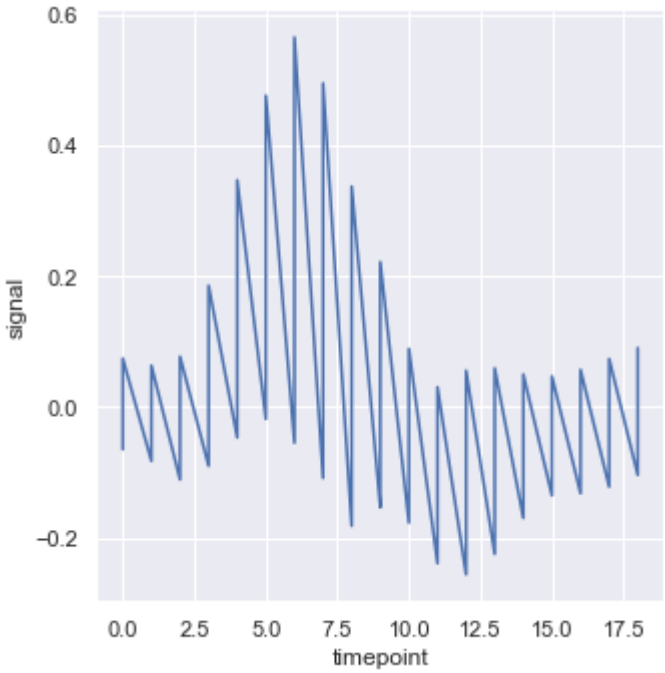
```

```

Entrée [42]:

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

Out[42]:

```
<seaborn.axisgrid.FacetGrid at 0x1f7cfd6c2e8>
```



Entrée [ ]:

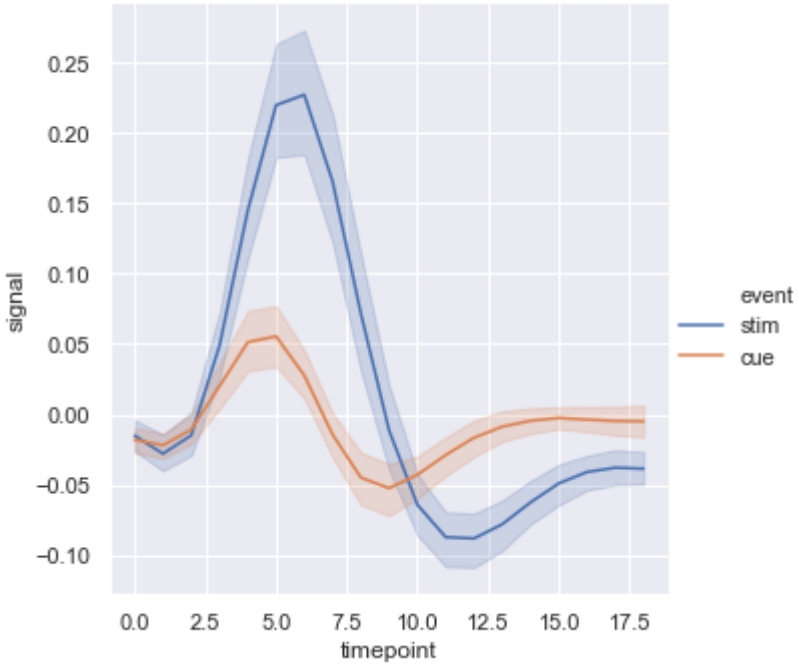
```

```

Entrée [43]:

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

Out[43]:

```
<seaborn.axisgrid.FacetGrid at 0x1f7cfd6c9e8>
```



Entrée [44]:

```
fmri.head()
```

Out[44]:

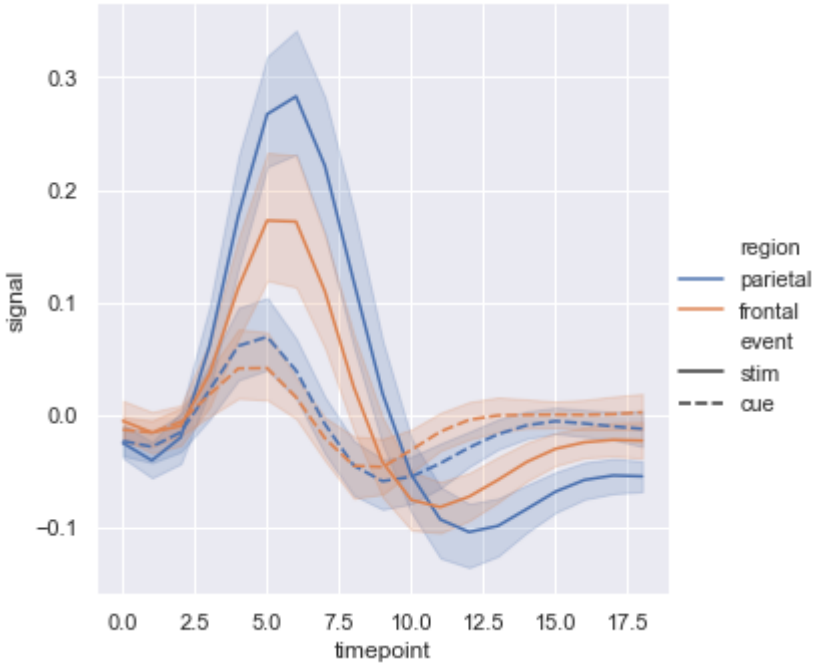
	subject	timepoint	event	region	signal
0	s13	18	stim	parietal	-0.017552
1	s5	14	stim	parietal	-0.080883
2	s12	18	stim	parietal	-0.081033
3	s11	18	stim	parietal	-0.046134
4	s10	18	stim	parietal	-0.037970

Entrée [45]:

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

Out[45]:

```
<seaborn.axisgrid.FacetGrid at 0x1f7d00cba58>
```

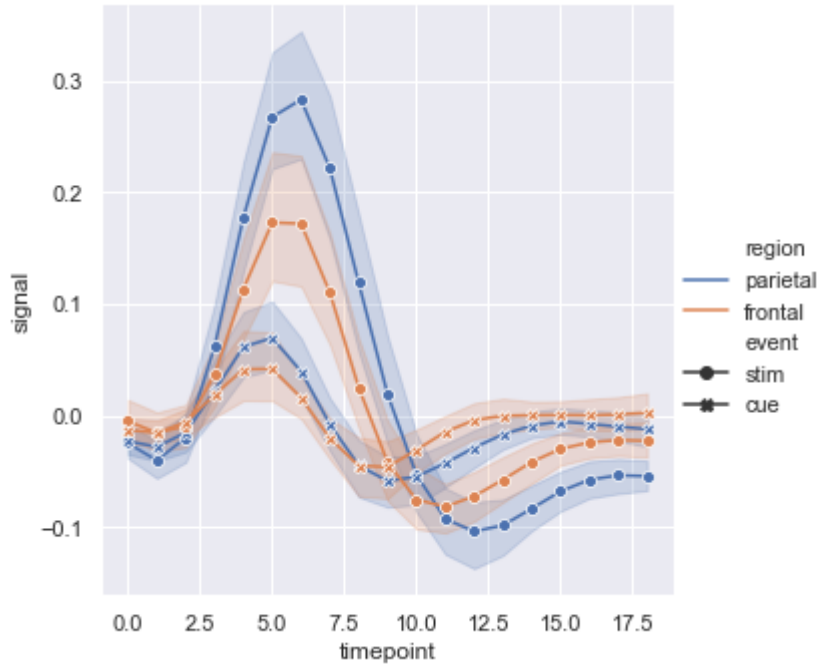


Entrée [47]:

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

Out[47]:

```
<seaborn.axisgrid.FacetGrid at 0x1f7cfb97390>
```

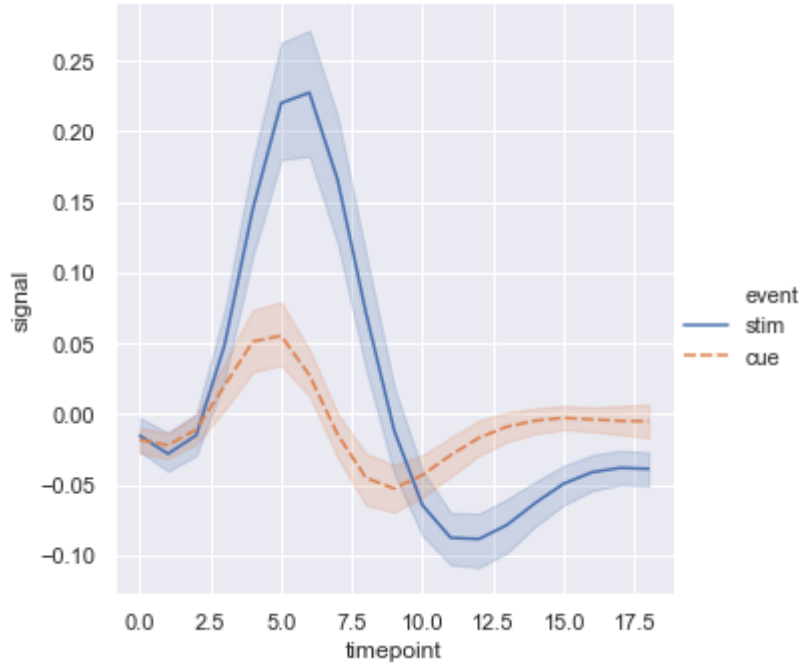


Entrée [51]:

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

Out[51]:

```
<seaborn.axisgrid.FacetGrid at 0x1f7d00ec828>
```



Entrée [ ]:

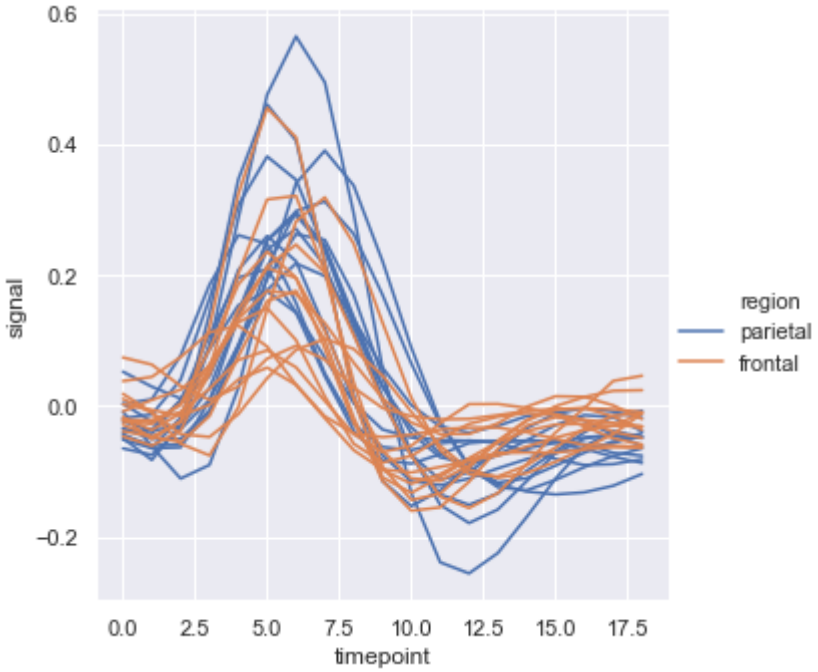
```
```

Entrée [56]:

```
sns.relplot(x = 'timepoint', y = 'signal', hue = 'region', units = 'subject', estimator = N
```

Out[56]:

```
<seaborn.axisgrid.FacetGrid at 0x1f7d153c198>
```



Entrée [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

Entrée [ ]:

```
```

Entrée [59]:

```
dots = sns.load_dataset('dots').query("align == 'dots'")
dots.head()
```

Out[59]:

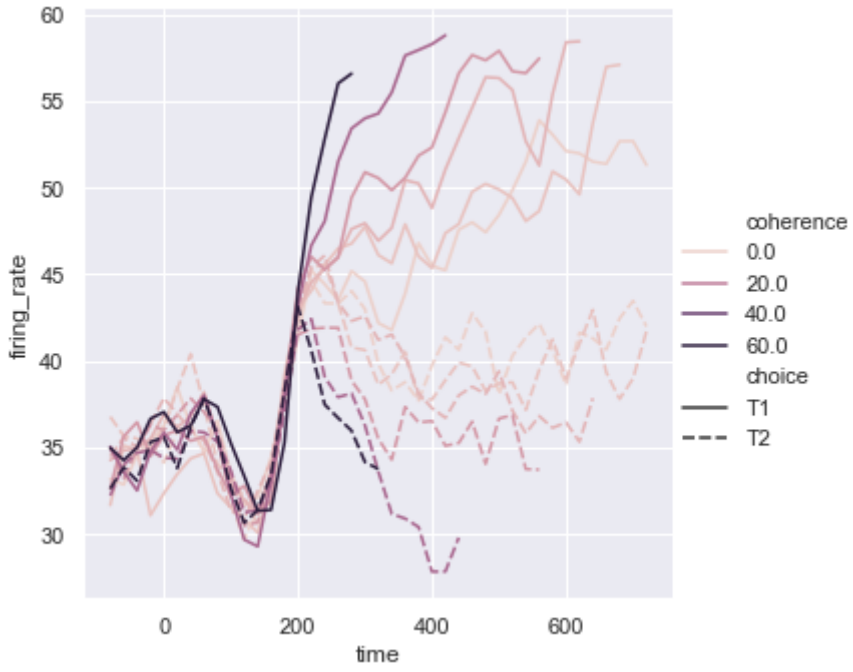
	align	choice	time	coherence	firing_rate
0	dots	T1	-80	0.0	33.189967
1	dots	T1	-80	3.2	31.691726
2	dots	T1	-80	6.4	34.279840
3	dots	T1	-80	12.8	32.631874
4	dots	T1	-80	25.6	35.060487

Entrée [62]:

```
sns.relplot(x = 'time', y = 'firing_rate', data = dots, kind = 'line', hue = 'coherence', s
```

Out[62]:

```
<seaborn.axisgrid.FacetGrid at 0x1f7d255f588>
```



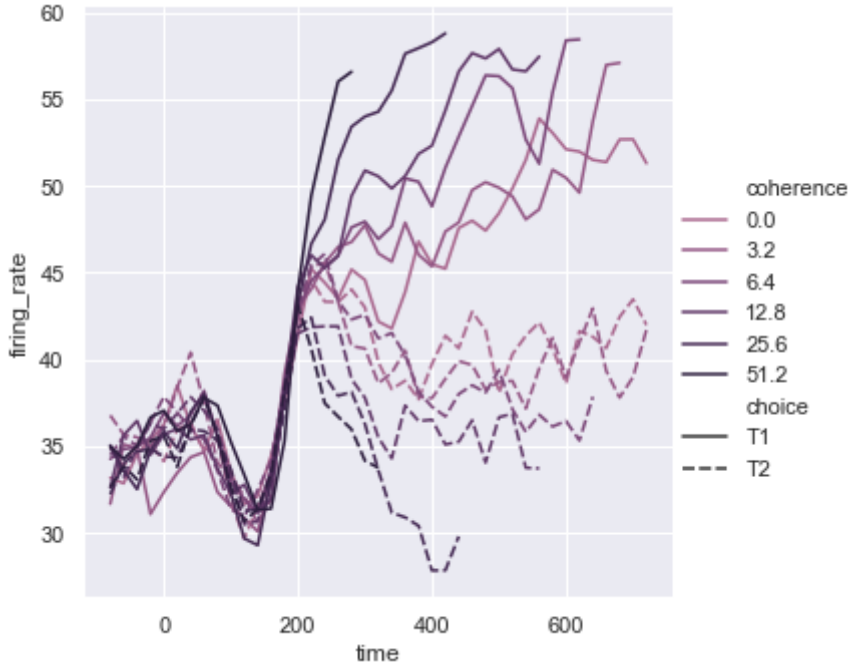
Entrée [ ]:

Entrée [64]:

```
palette = sns.cubehelix_palette(light = 0.5, n_colors=6)
sns.relplot(x = 'time', y = 'firing_rate', data = dots, kind = 'line', hue = 'coherence', s
```

Out[64]:

```
<seaborn.axisgrid.FacetGrid at 0x1f7d26d9c88>
```



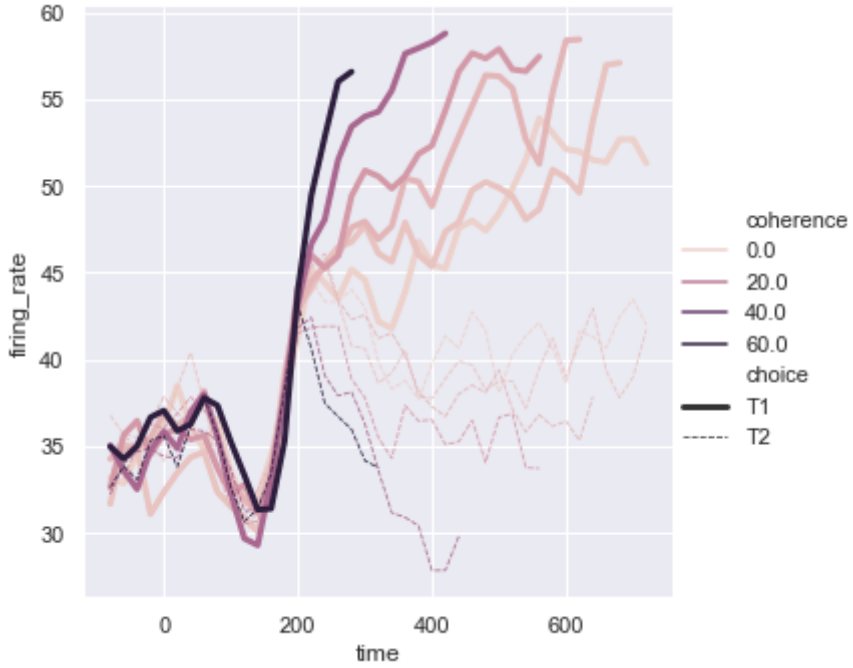
Entrée [ ]:

Entrée [71]:

```
sns.relplot(x = 'time', y = 'firing_rate', hue = 'coherence', size = 'choice', style = 'cho
```

Out[71]:

```
<seaborn.axisgrid.FacetGrid at 0x1f7d3ba7b70>
```



Entrée [ ]:

```

```

Entrée [72]:

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

Entrée [73]:

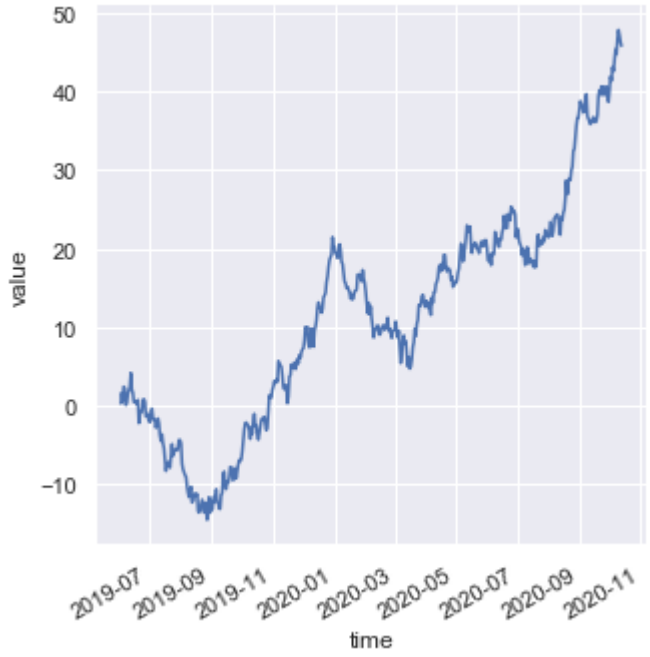
```
df.head()
```

Out[73]:

	time	value
0	2019-06-02	0.286372
1	2019-06-03	1.658994
2	2019-06-04	0.545582
3	2019-06-05	2.415458
4	2019-06-06	2.133488

Entrée [77]:

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



Entrée [78]:

```
tips.head()
```

Out[78]:

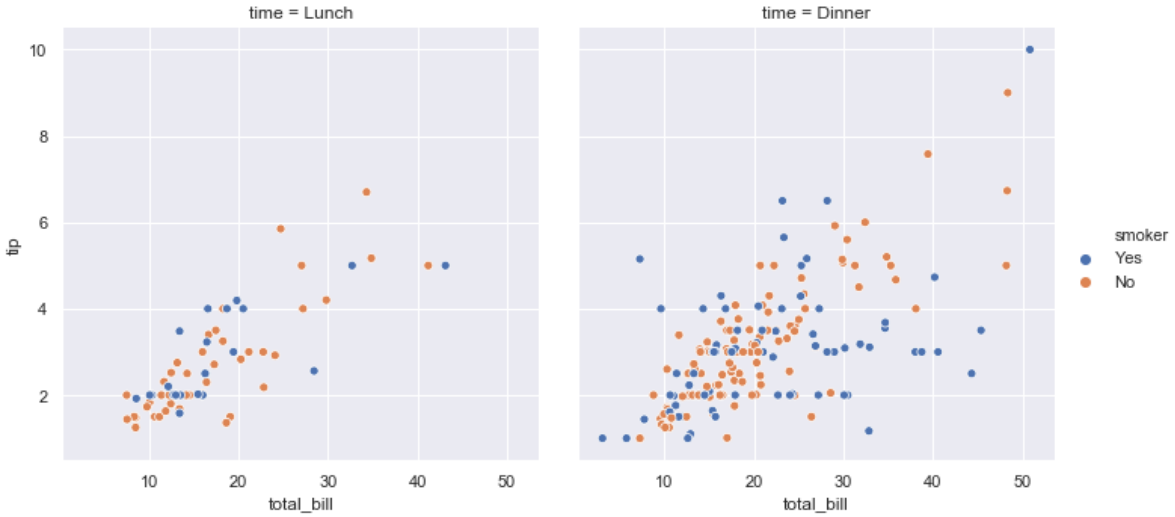
	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4

Entrée [81]:

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

Out[81]:

```
<seaborn.axisgrid.FacetGrid at 0x1f7d440e588>
```

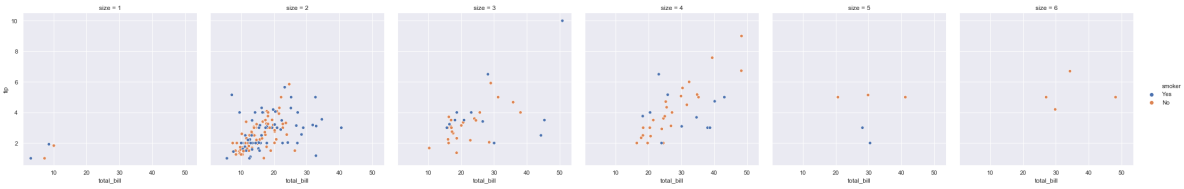


Entrée [85]:

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

Out[85]:

```
<seaborn.axisgrid.FacetGrid at 0x1f7d6f7d860>
```



Entrée [ ]:

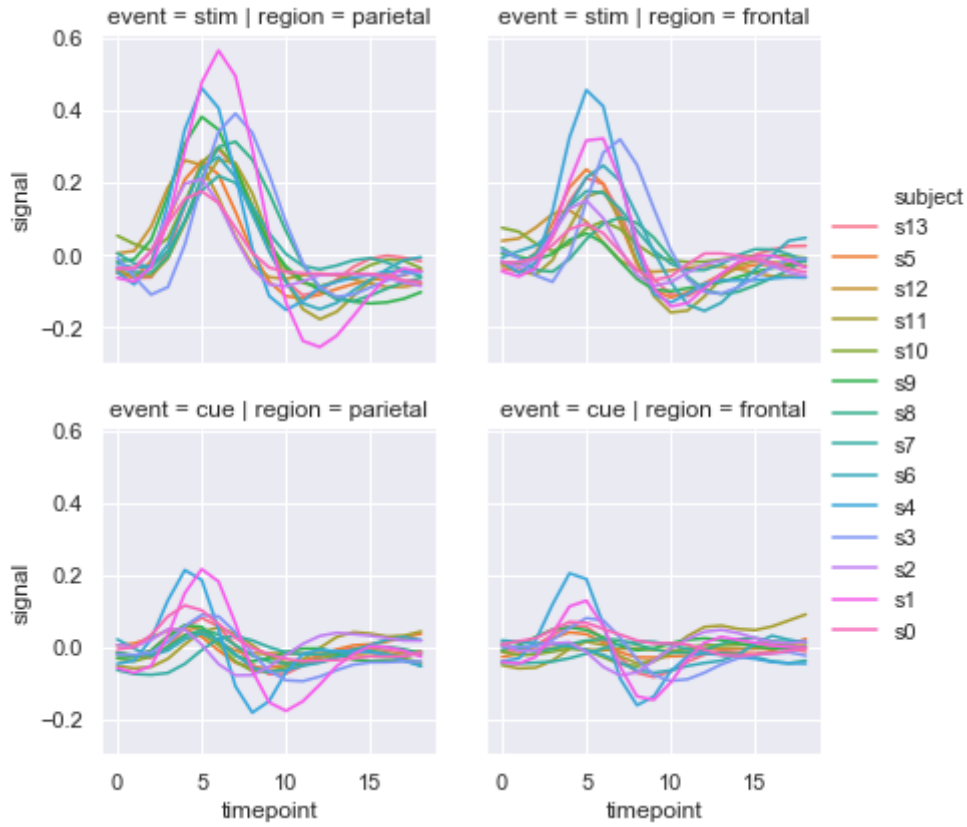
```

```

Entrée [84]:

```
sns.relplot(x = 'timepoint', y = 'signal', hue = 'subject', col = 'region', row = 'event',
```

Out[84]:

```
<seaborn.axisgrid.FacetGrid at 0x1f7d66850b8>
```



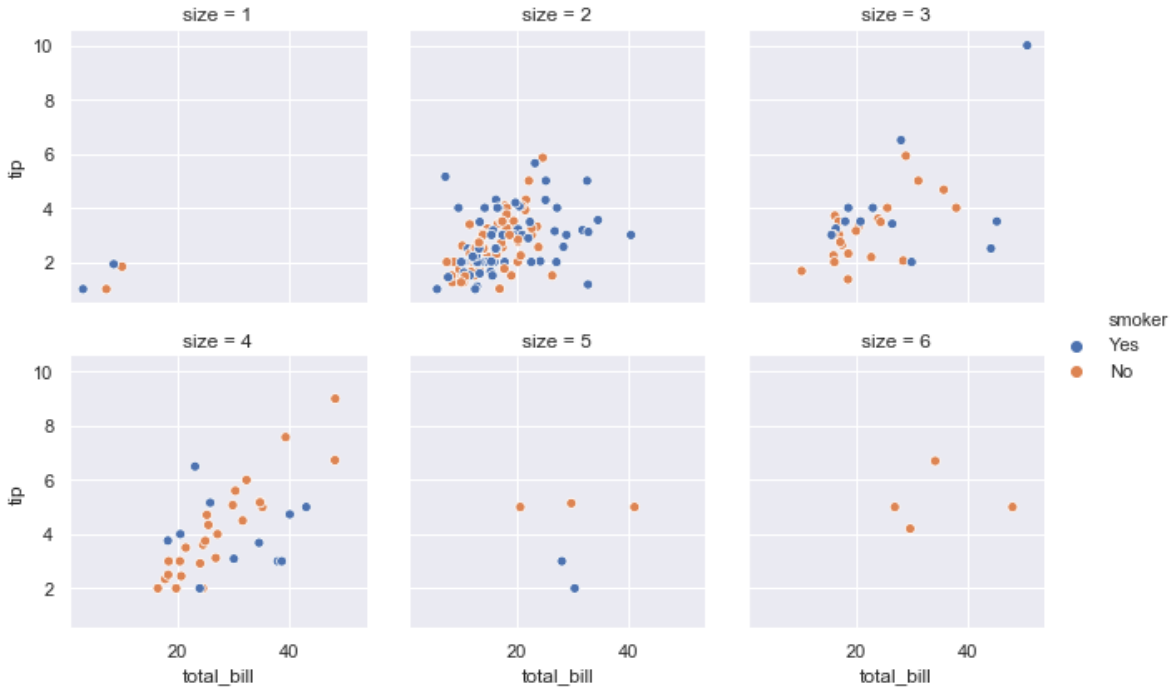
Entrée [ ]:

Entrée [87]:

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

Out[87]:

```
<seaborn.axisgrid.FacetGrid at 0x1f7d89d8588>
```



Entrée [ ]:

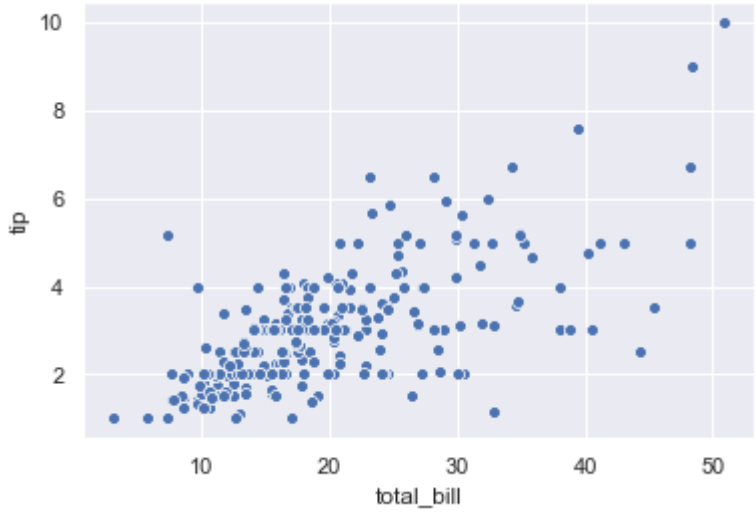
```

```

Entrée [89]:

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

Out[89]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1f7d935b278>
```



Entrée [ ]:

```

```

Entrée [91]:

```
fmri.head()
```

Out[91]:

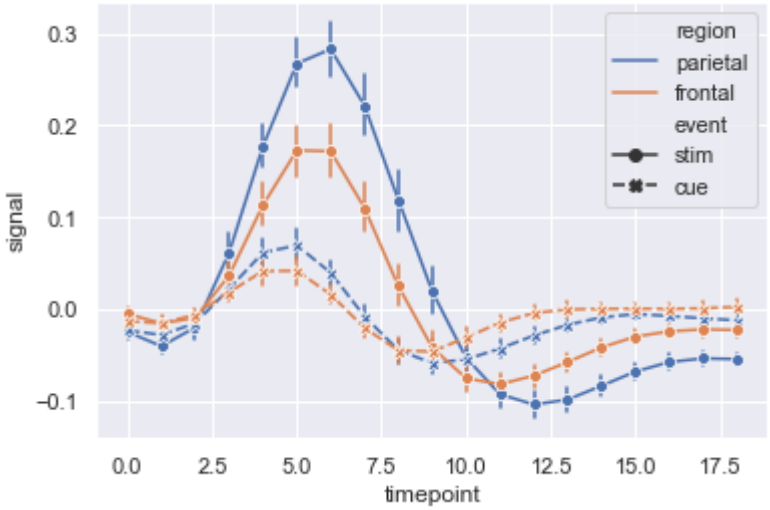
	subject	timepoint	event	region	signal
0	s13	18	stim	parietal	-0.017552
1	s5	14	stim	parietal	-0.080883
2	s12	18	stim	parietal	-0.081033
3	s11	18	stim	parietal	-0.046134
4	s10	18	stim	parietal	-0.037970

Entrée [101]:

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

Out[101]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1f7d9b9d7b8>
```

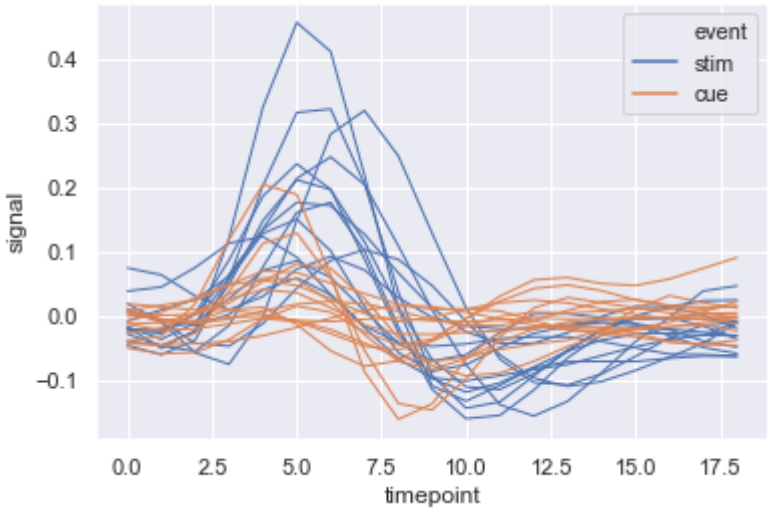


Entrée [102]:

```
sns.lineplot(x = 'timepoint', y = 'signal', hue = 'event', units = 'subject', estimator = N
```

Out[102]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1f7d9d736a0>
```



Entrée [103]:

```
dots.head()
```

Out[103]:

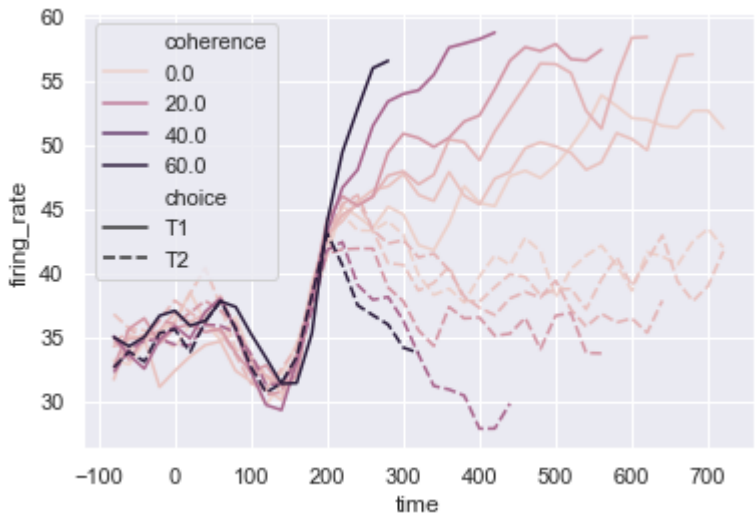
	align	choice	time	coherence	firing_rate
0	dots	T1	-80	0.0	33.189967
1	dots	T1	-80	3.2	31.691726
2	dots	T1	-80	6.4	34.279840
3	dots	T1	-80	12.8	32.631874
4	dots	T1	-80	25.6	35.060487

Entrée [105]:

```
sns.lineplot(x = 'time', y = 'firing_rate', hue = 'coherence', style = 'choice', data = dot
```

Out[105]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1f7d9c7b9b0>
```



Entrée [ ]:

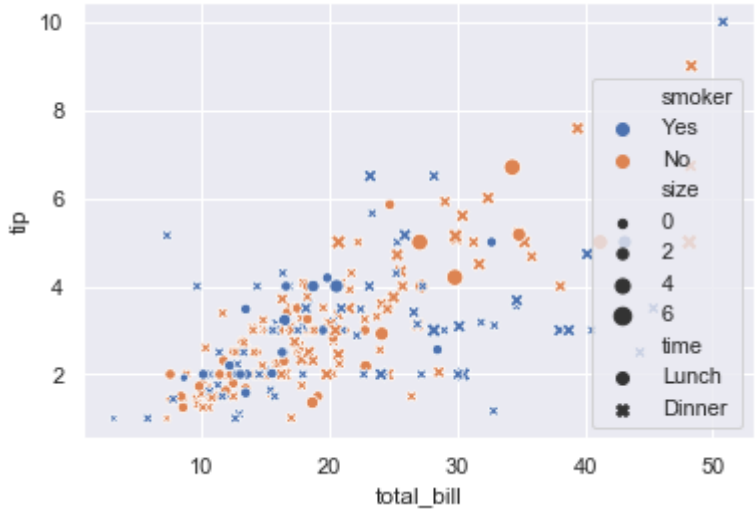
```

```

Entrée [111]:

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

Out[111]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1f7daf2b860>
```



Entrée [ ]:

```

```

Entrée [112]:

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

Entrée [113]:

```
iris.head()
```

Out[113]:

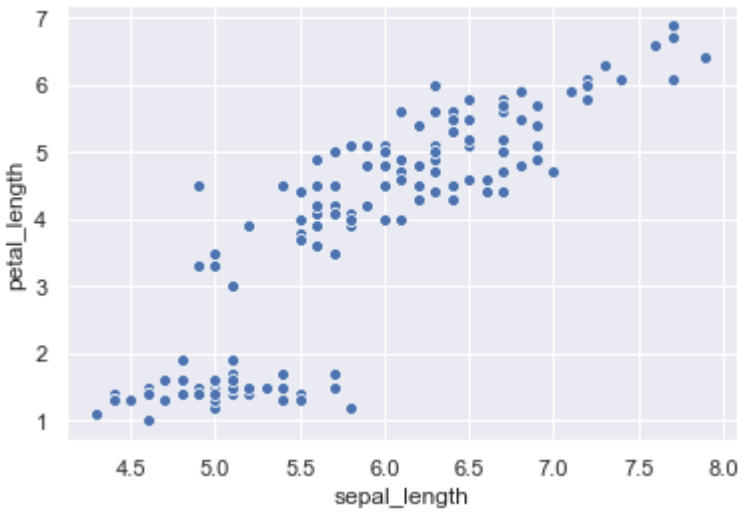
	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

Entrée [114]:

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

Out[114]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1f7db1172b0>
```

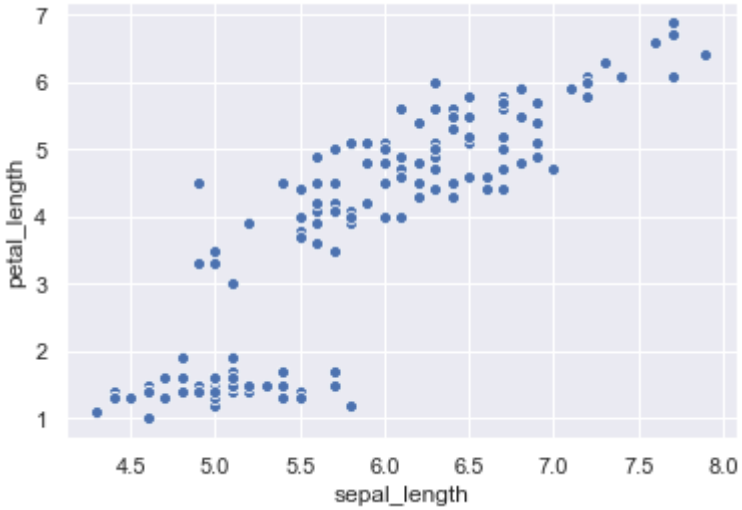


Entrée [115]:

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

Out[115]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1f7db12bb00>
```



2. Categorical Data Ploting

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

Entrée [116]:

```
tips.head()
```

Out[116]:

	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

Entrée [117]:

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

Entrée [118]:

```
titanic.head()
```

Out[118]:

	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

Entrée [119]:

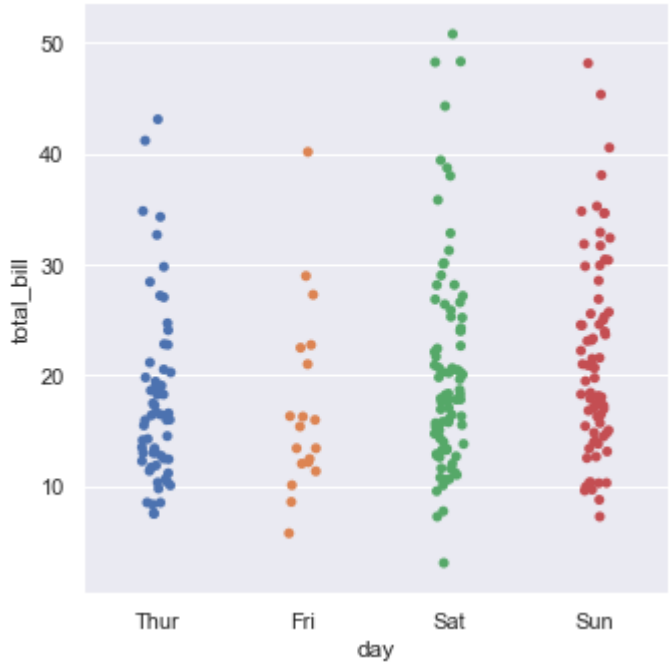
```
#catplot()
```

Entrée [121]:

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

Out[121]:

```
<seaborn.axisgrid.FacetGrid at 0x1f7db1c3128>
```

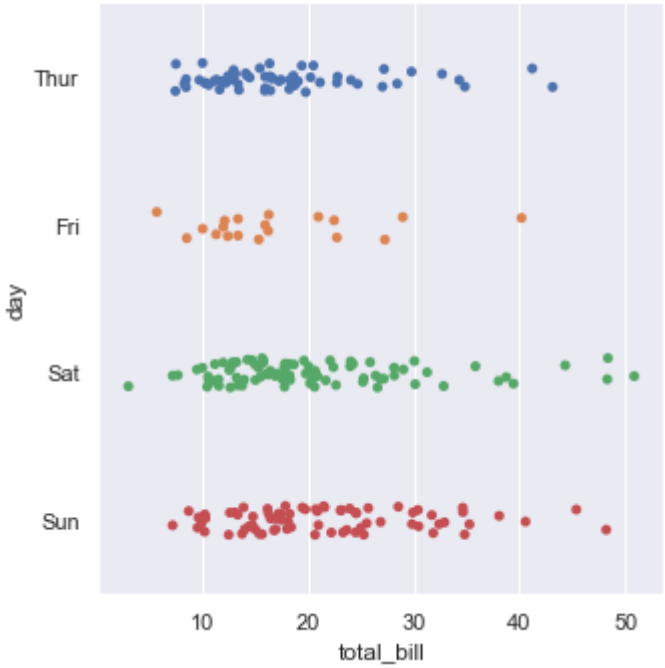


Entrée [122]:

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

Out[122]:

```
<seaborn.axisgrid.FacetGrid at 0x1f7daffc860>
```

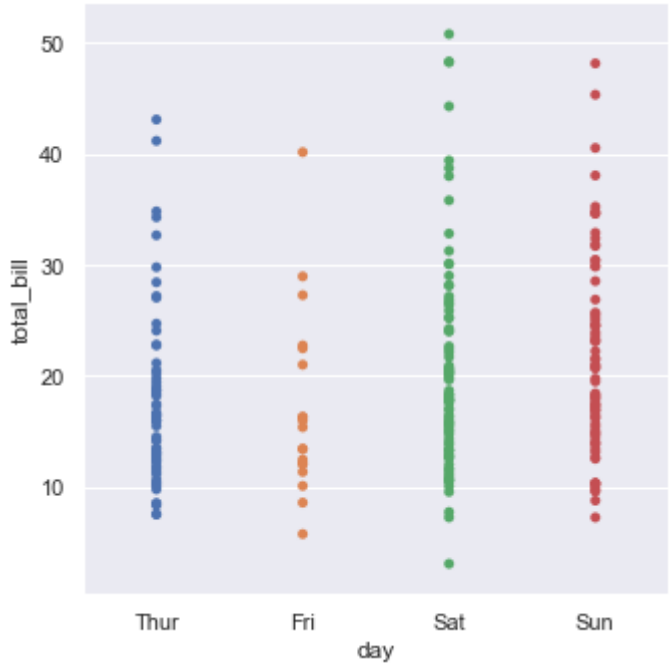


Entrée [124]:

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

Out[124]:

```
<seaborn.axisgrid.FacetGrid at 0x1f7db5f69e8>
```



Entrée [ ]:

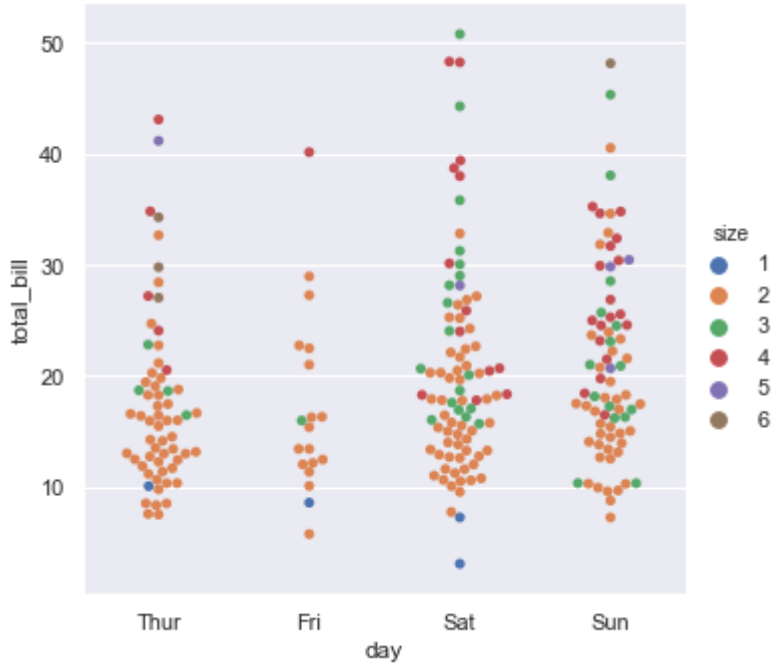
```

```

Entrée [130]:

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

Out[130]:

```
<seaborn.axisgrid.FacetGrid at 0x1f7dbc6c780>
```



Entrée [ ]:

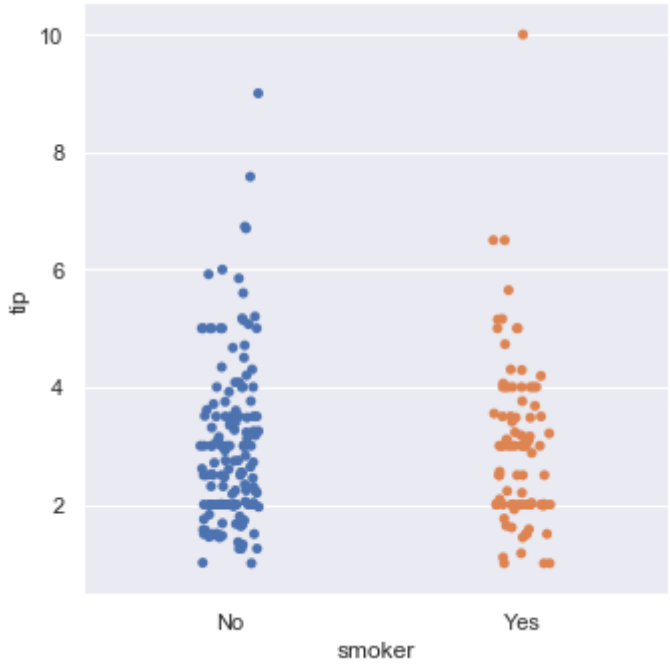
```

```

Entrée [136]:

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

Out[136]:

```
<seaborn.axisgrid.FacetGrid at 0x1f7dbd51ef0>
```



Entrée [132]:

```
tips.head()
```

Out[132]:

	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

Entrée [ ]:

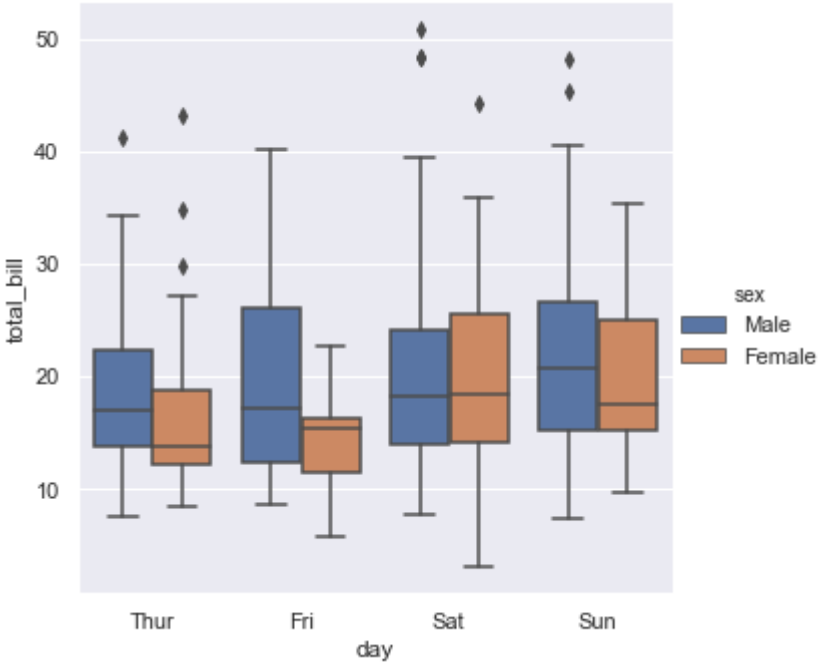
```
```

Entrée [147]:

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

Out[147]:

```
<seaborn.axisgrid.FacetGrid at 0x1f7dd06beb8>
```

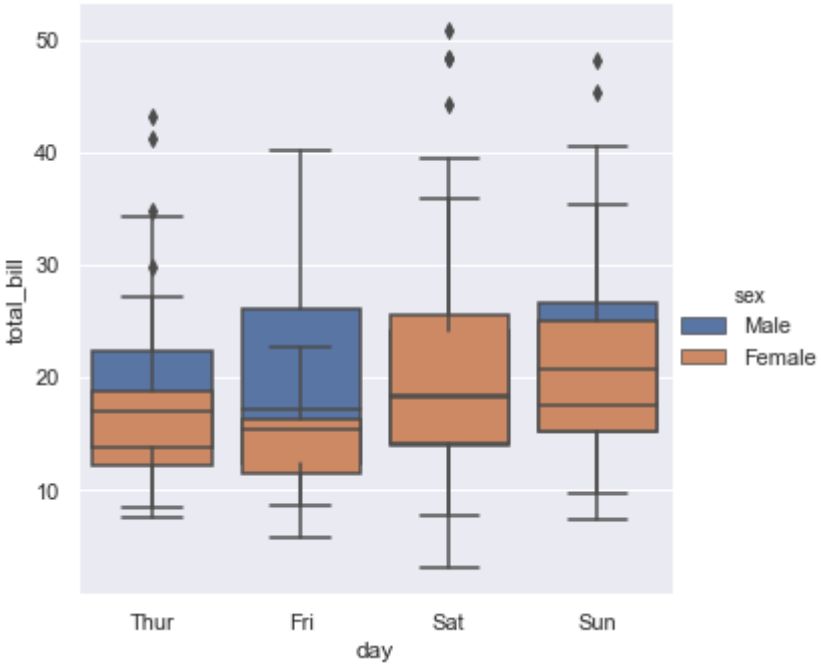


Entrée [146]:

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

Out[146]:

```
<seaborn.axisgrid.FacetGrid at 0x1f7de1c7a58>
```



Entrée [ ]:

```

```

Entrée [152]:

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

Out[152]:

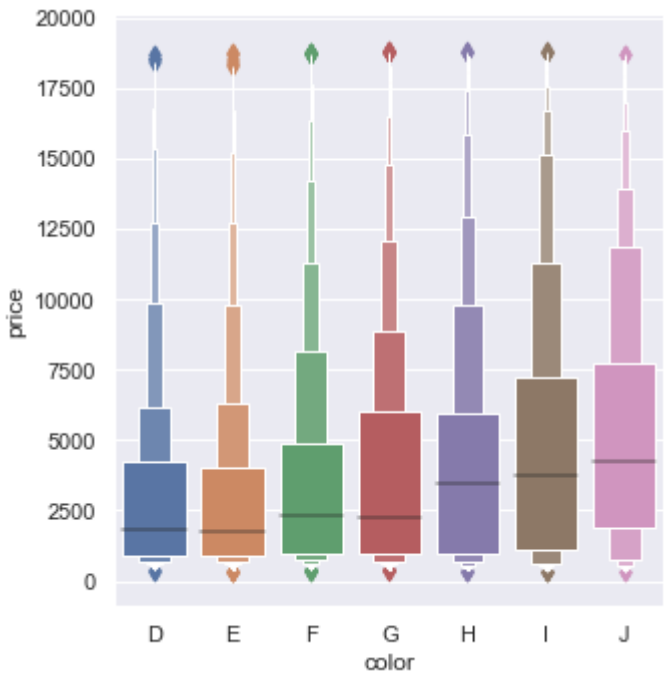
	carat	cut	color	clarity	depth	table	price	x	y	z
0	0.23	Ideal	E	SI2	61.5	55.0	326	3.95	3.98	2.43
1	0.21	Premium	E	SI1	59.8	61.0	326	3.89	3.84	2.31
2	0.23	Good	E	VS1	56.9	65.0	327	4.05	4.07	2.31
3	0.29	Premium	I	VS2	62.4	58.0	334	4.20	4.23	2.63
4	0.31	Good	J	SI2	63.3	58.0	335	4.34	4.35	2.75

Entrée [149]:

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

Out[149]:

```
<seaborn.axisgrid.FacetGrid at 0x1f7de346978>
```

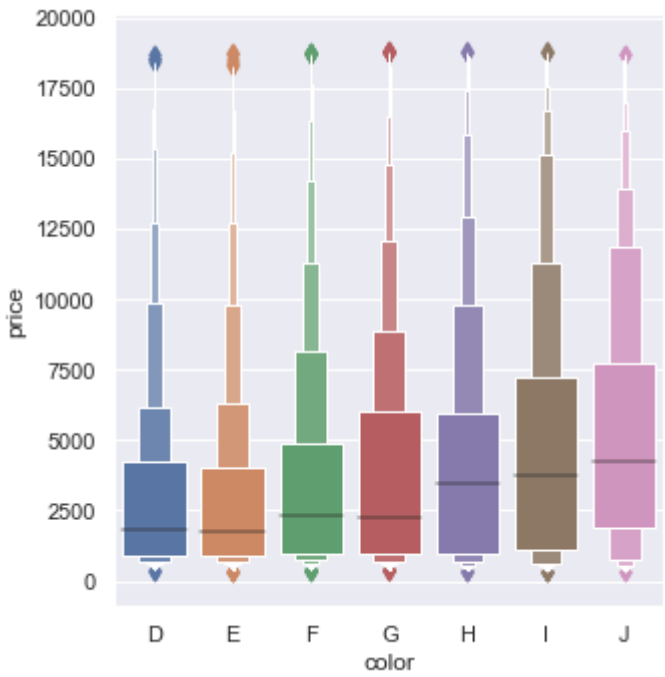


Entrée [156]:

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

Out[156]:

```
<seaborn.axisgrid.FacetGrid at 0x1f7de2ff2b0>
```

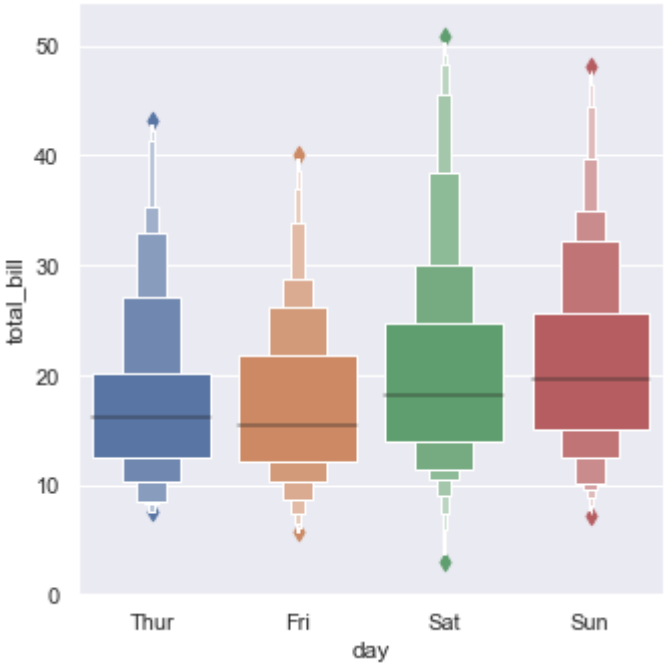


Entrée [159]:

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

Out[159]:

```
<seaborn.axisgrid.FacetGrid at 0x1f7deb941d0>
```



Entrée [ ]:

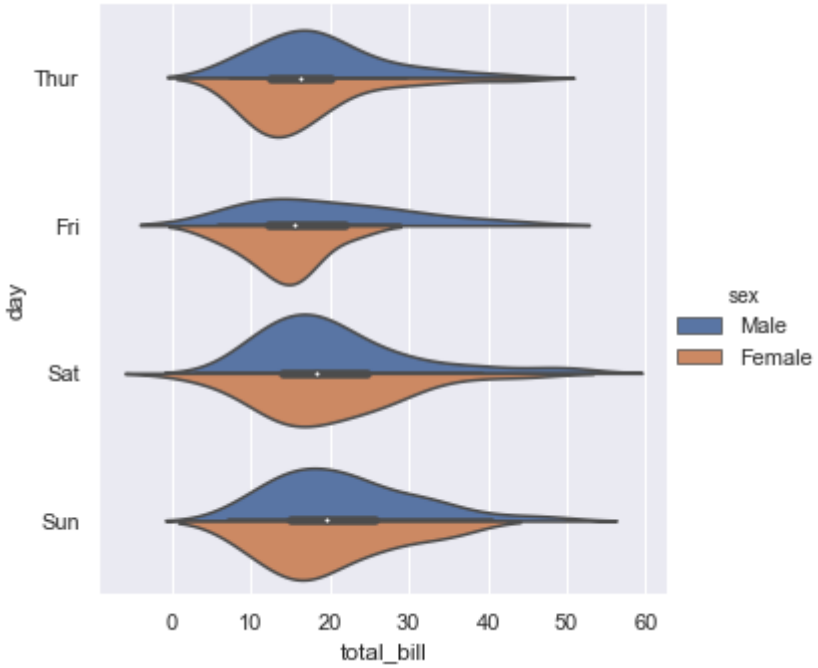
```

```

Entrée [164]:

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

Out[164]:

```
<seaborn.axisgrid.FacetGrid at 0x1f7dfca52b0>
```



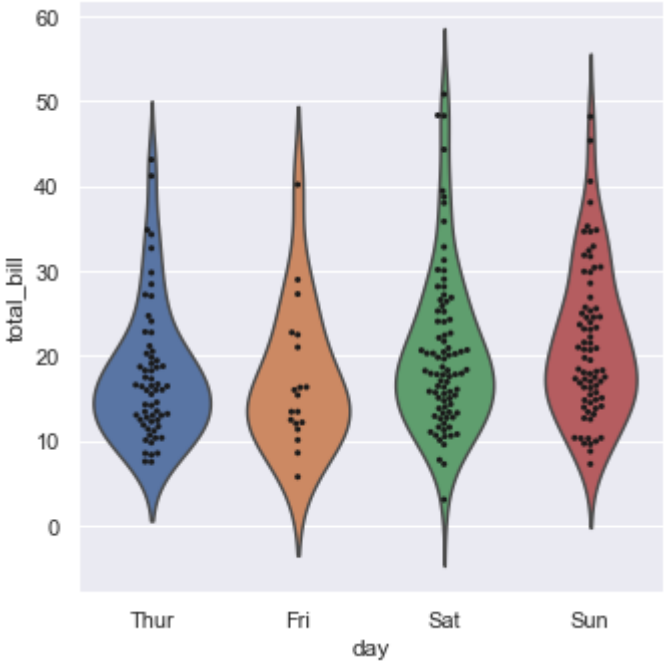
Entrée [ ]:

Entrée [167]:

```
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[167]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1f7de1dfa58>
```



Entrée [168]:

```
titanic.head()
```

Out[168]:

	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

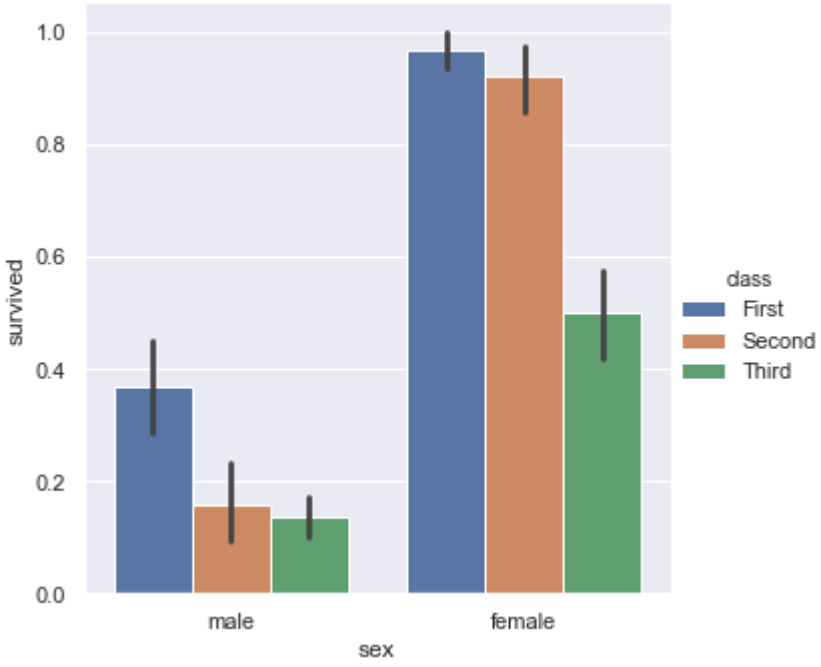
Entrée [169]:

```
sns.catplot(x = 'sex', y = 'survived', hue = 'class', kind = 'bar', data = titanic)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\scipy\stats\stats.py:1713: Future
Warning: Using a non-tuple sequence for multidimensional indexing is depreca
ted; use `arr[tuple(seq)]` instead of `arr[seq]`. In the future this will be
interpreted as an array index, `arr[np.array(seq)]`, which will result eithe
r in an error or a different result.
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval
```

Out[169]:

```
<seaborn.axisgrid.FacetGrid at 0x1f7dfafbc50>
```



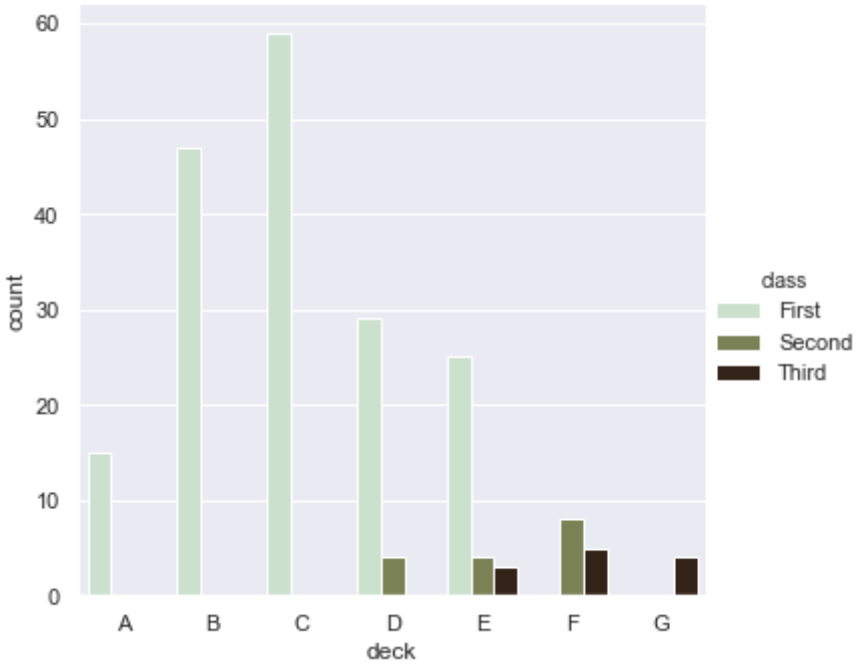
Entrée [ ]:

Entrée [173]:

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

Out[173]:

```
<seaborn.axisgrid.FacetGrid at 0x1f7e01a8e10>
```



Entrée [ ]:

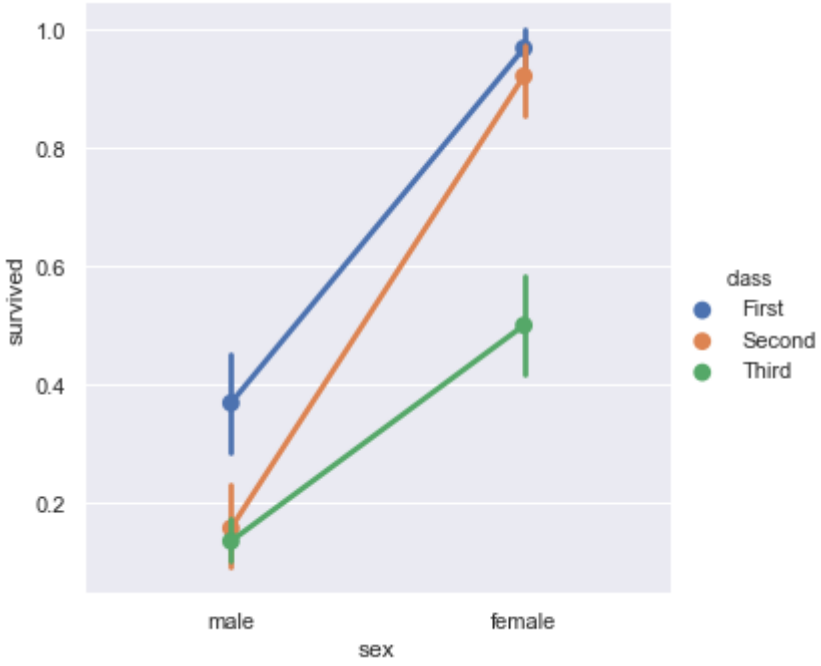
Entrée [174]:

```
sns.catplot(x = 'sex', y = 'survived', hue = 'class', kind = 'point', data = titanic)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\scipy\stats\stats.py:1713: Future
Warning: Using a non-tuple sequence for multidimensional indexing is depreca
ted; use `arr[tuple(seq)]` instead of `arr[seq]`. In the future this will be
interpreted as an array index, `arr[np.array(seq)]`, which will result eithe
r in an error or a different result.
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval
```

Out[174]:

```
<seaborn.axisgrid.FacetGrid at 0x1f7e0227128>
```



3. Visualizing Distribution of the Data

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

Entrée [175]:

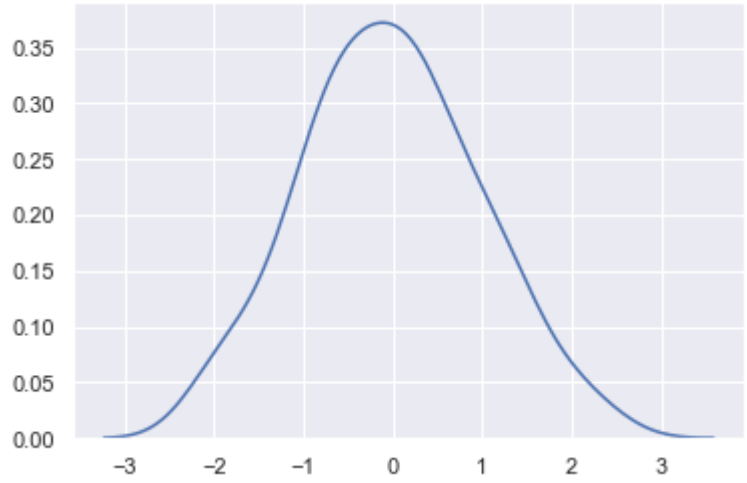
```
x = randn(100)
```

Entrée [182]:

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

Out[182]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1f7e0695198>
```



Entrée [ ]:

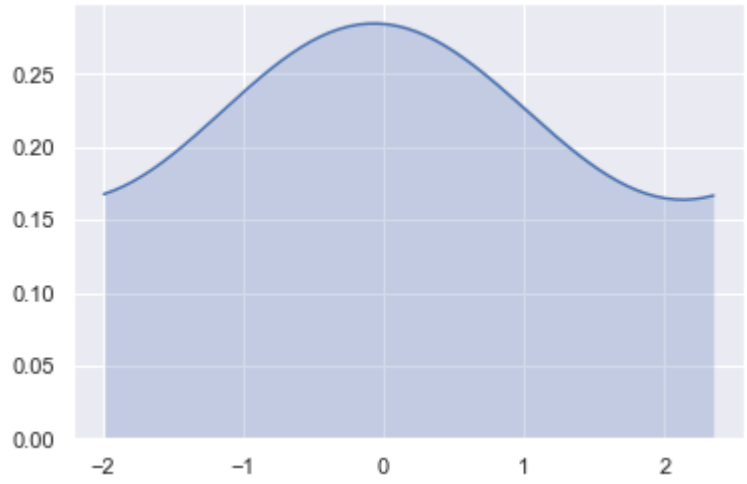
```

```

Entrée [191]:

```
sns.kdeplot(x, shade=True, cbar = True, bw = 1, cut = 0)
```

Out[191]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1f7e09204a8>
```



Entrée [ ]:

```

```

Entrée [193]:

```
tips.head()
```

Out[193]:

	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

Entrée [194]:

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

Entrée [ ]:

```

```

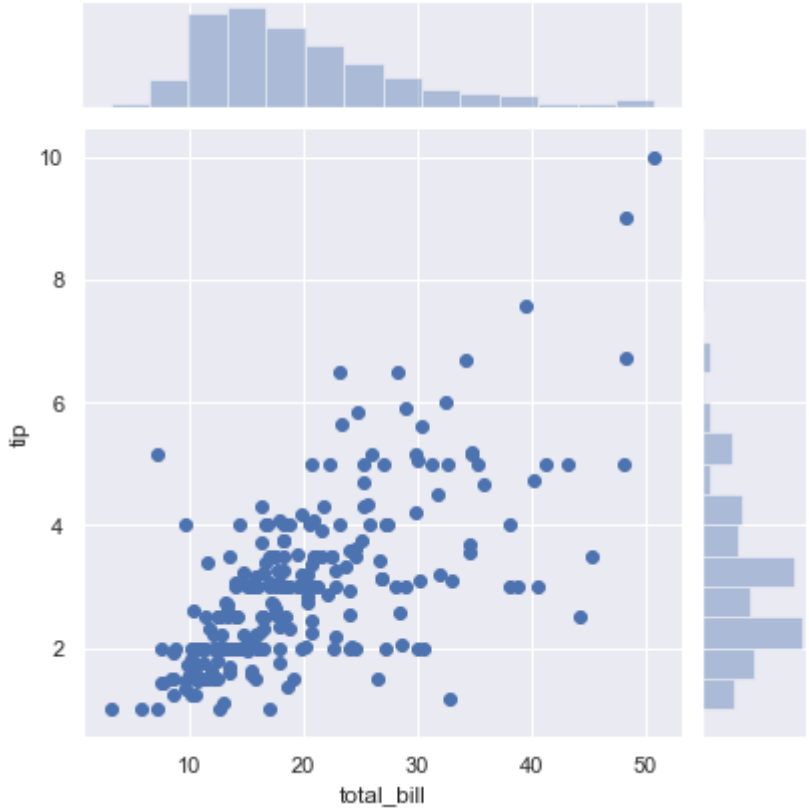
Entrée [195]:

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

```
C:\ProgramData\Anaconda3\lib\site-packages\scipy\stats\stats.py:1713: Future
Warning: Using a non-tuple sequence for multidimensional indexing is depreca
ted; use `arr[tuple(seq)]` instead of `arr[seq]`. In the future this will be
interpreted as an array index, `arr[np.array(seq)]`, which will result eithe
r in an error or a different result.
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval
```

Out[195]:

```
<seaborn.axisgrid.JointGrid at 0x1f7e0a83a20>
```

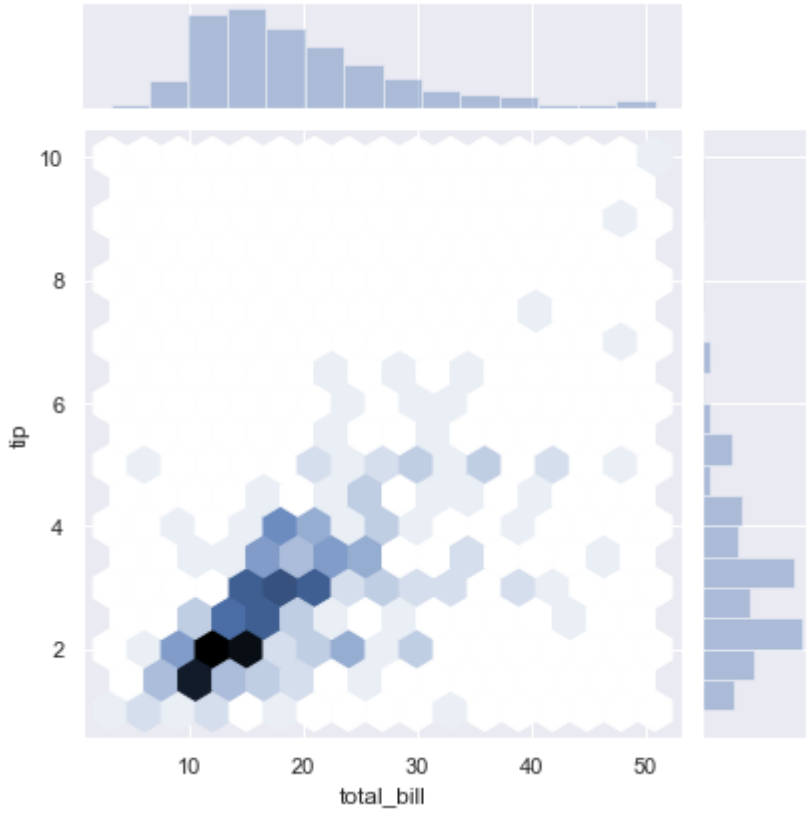


Entrée [199]:

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

Out[199]:

```
<seaborn.axisgrid.JointGrid at 0x1f7e3d318d0>
```



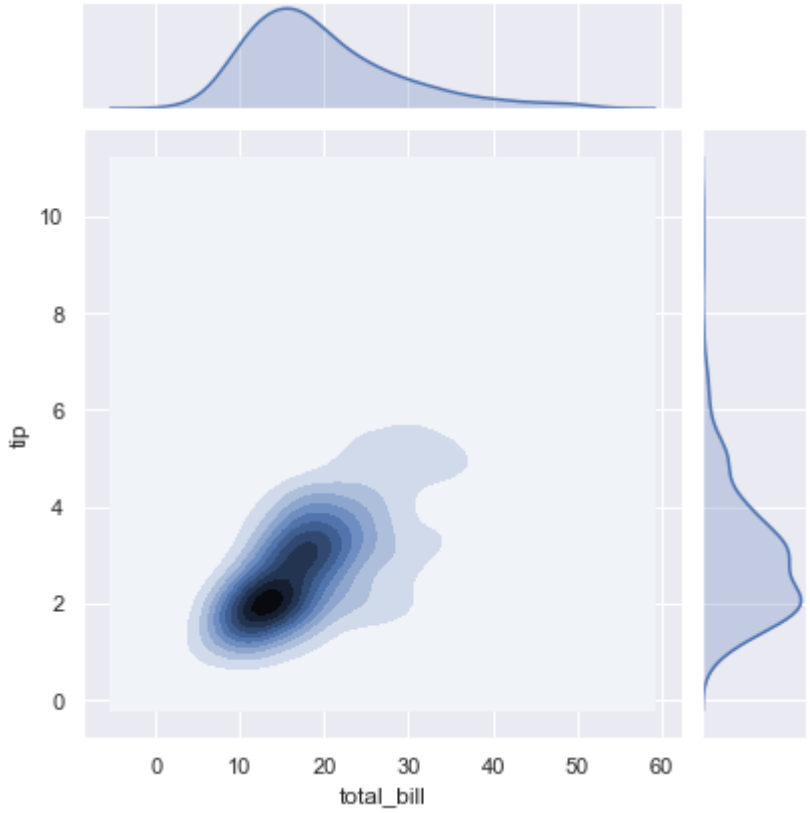
Entrée [ ]:

Entrée [200]:

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

Out[200]:

```
<seaborn.axisgrid.JointGrid at 0x1f7e2cb3978>
```



Entrée [ ]:

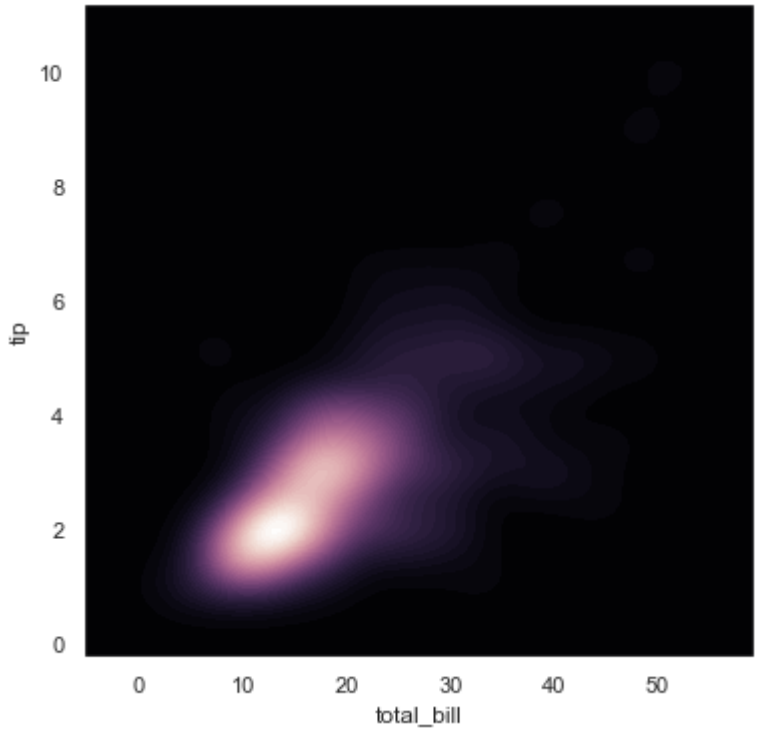
Entrée [201]:

```
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)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\scipy\stats\stats.py:1713: Future
Warning: Using a non-tuple sequence for multidimensional indexing is depreca
ted; use `arr[tuple(seq)]` instead of `arr[seq]`. In the future this will be
interpreted as an array index, `arr[np.array(seq)]`, which will result eithe
r in an error or a different result.
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval
```

Out[201]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1f7e4015550>
```

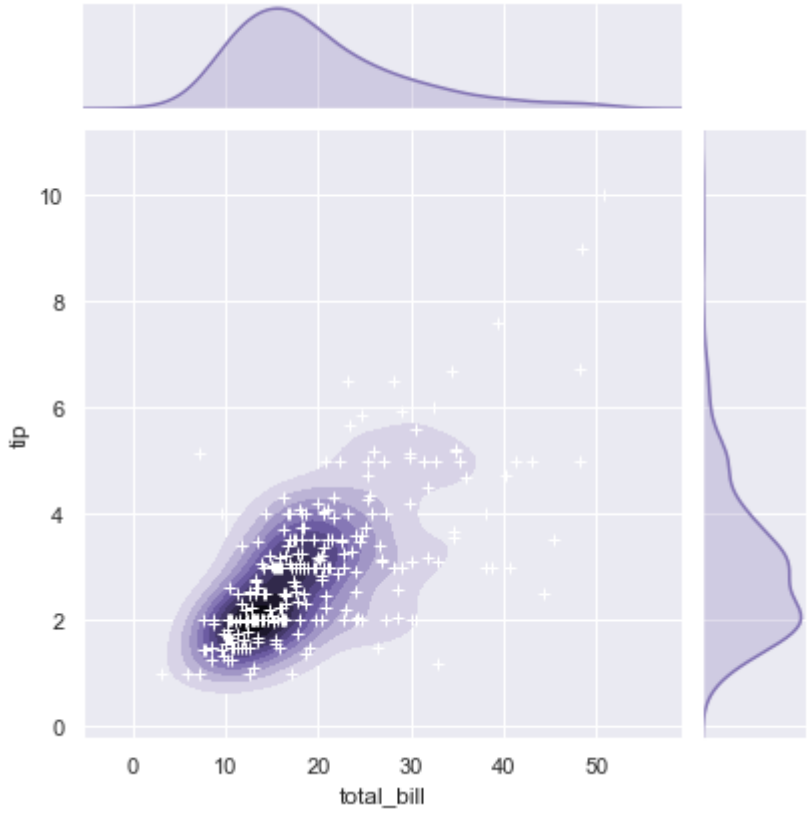


Entrée [ ]:

Entrée [203]:

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

```
C:\ProgramData\Anaconda3\lib\site-packages\scipy\stats\stats.py:1713: Future
Warning: Using a non-tuple sequence for multidimensional indexing is depreca
ted; use `arr[tuple(seq)]` instead of `arr[seq]`. In the future this will be
interpreted as an array index, `arr[np.array(seq)]`, which will result eithe
r in an error or a different result.
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval
```

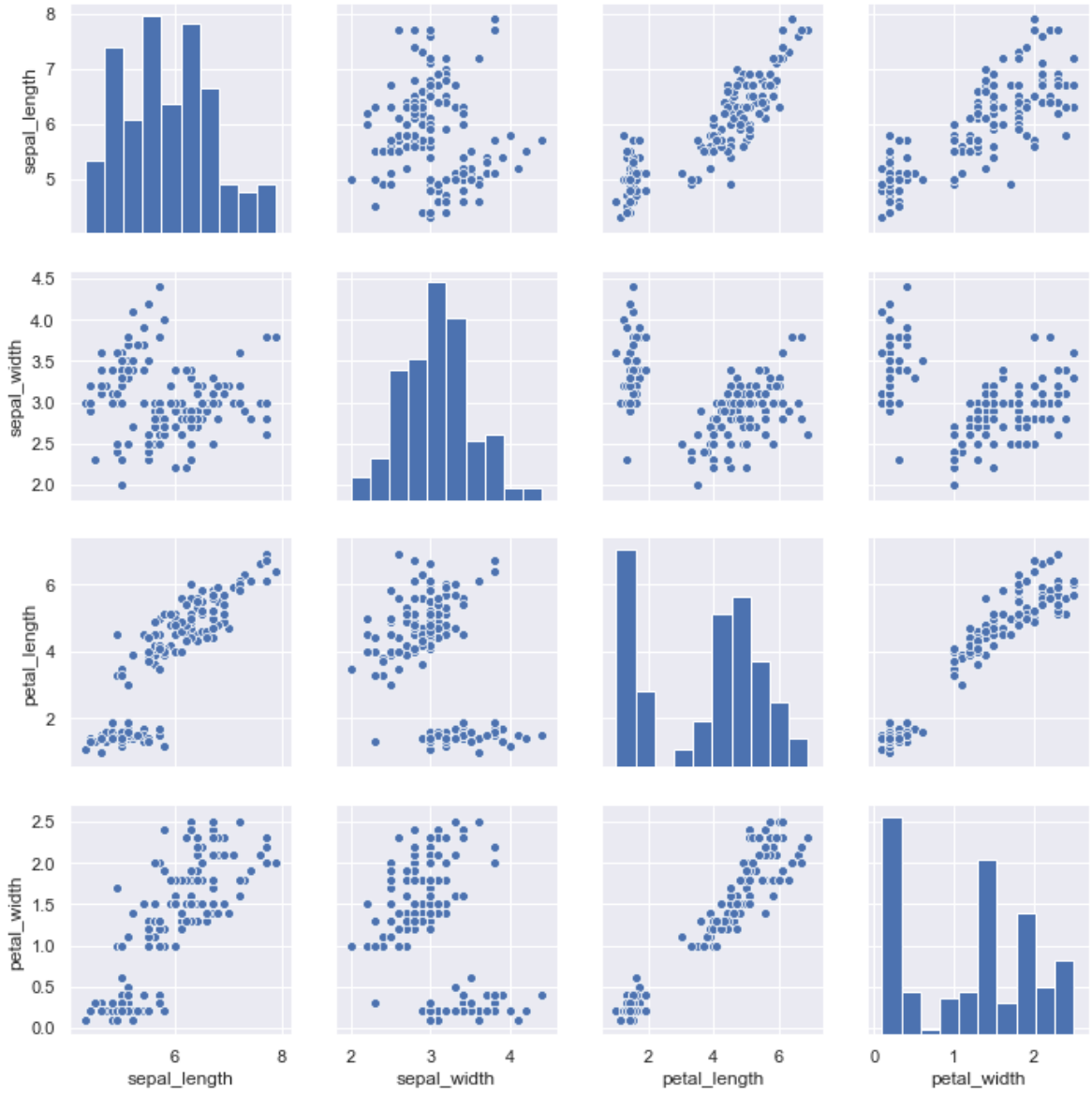


Entrée [205]:

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

Out[205]:

```
<seaborn.axisgrid.PairGrid at 0x1f7e42bad30>
```

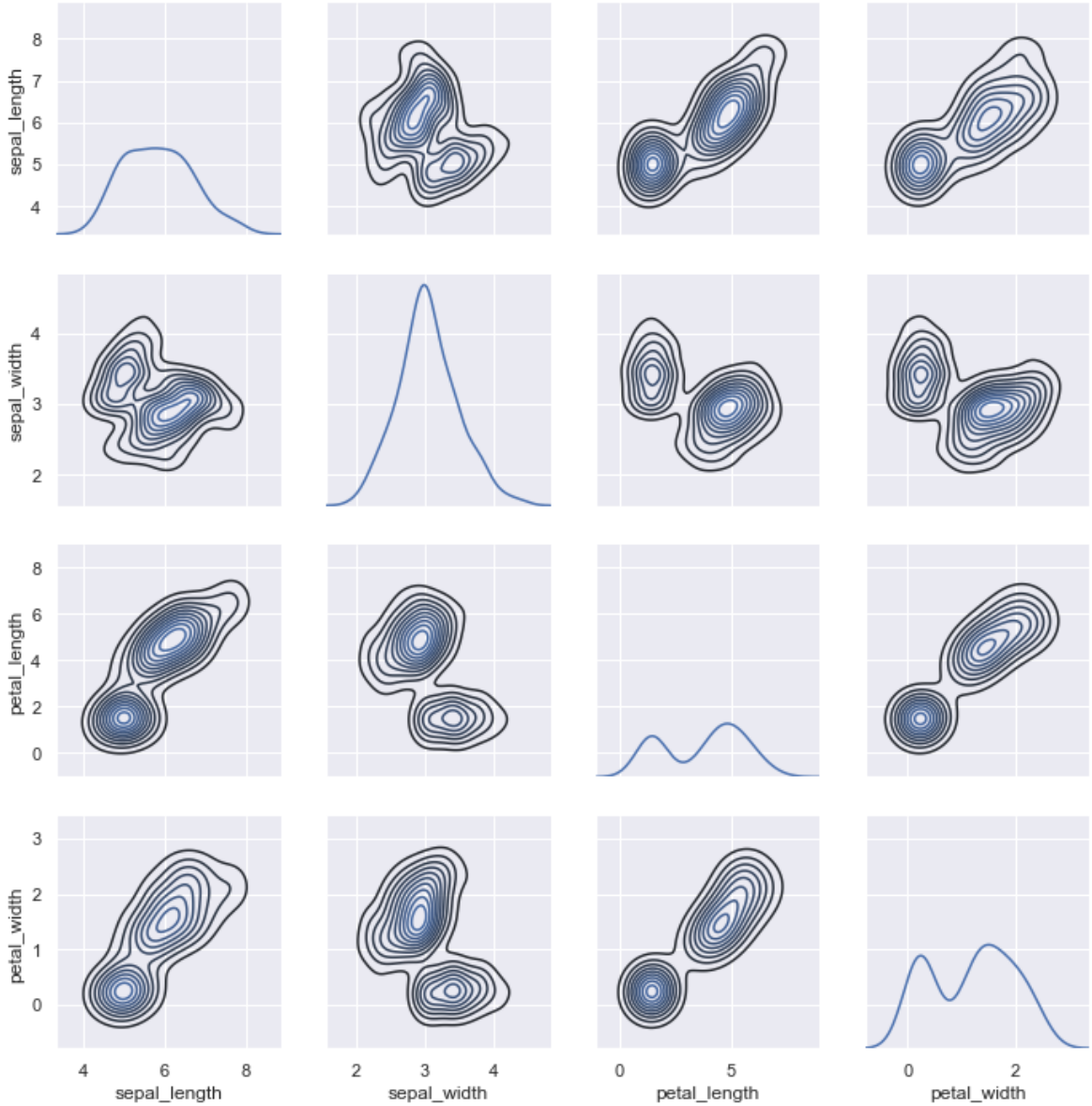


Entrée [207]:

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

Out[207]:

```
<seaborn.axisgrid.PairGrid at 0x1f7e6397fd0>
```



Entrée [ ]:

4. Linear Regression and Relationship

- regplot()
- lmplot()

Entrée [208]:

```
tips.head()
```

Out[208]:

	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

Entrée [ ]:

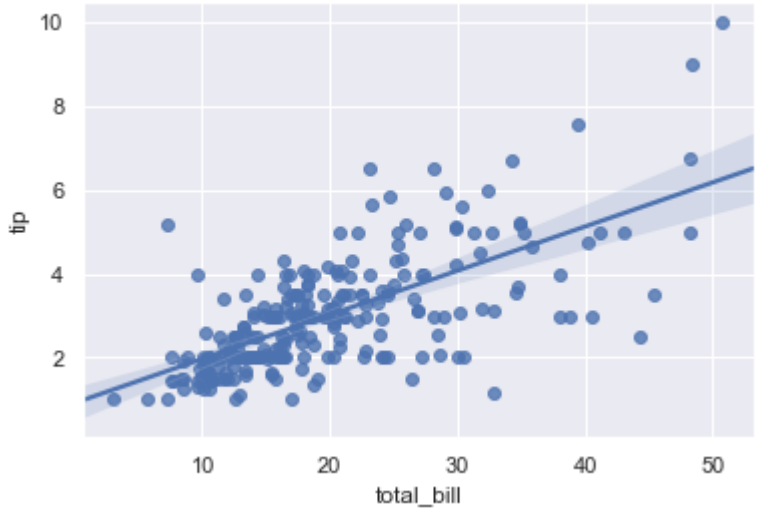
```

```

Entrée [209]:

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

Out[209]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1f7e6acaf60>
```

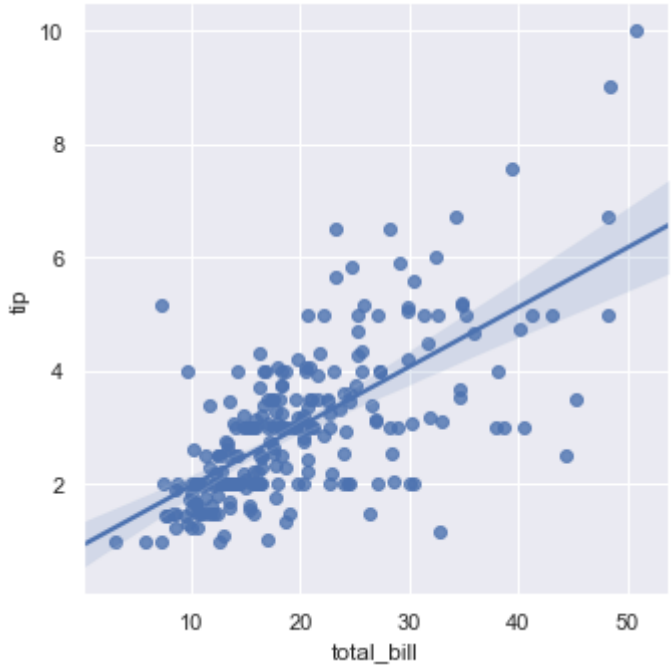


Entrée [212]:

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

Out[212]:

```
<seaborn.axisgrid.FacetGrid at 0x1f7e6b9e5c0>
```



Entrée [ ]:

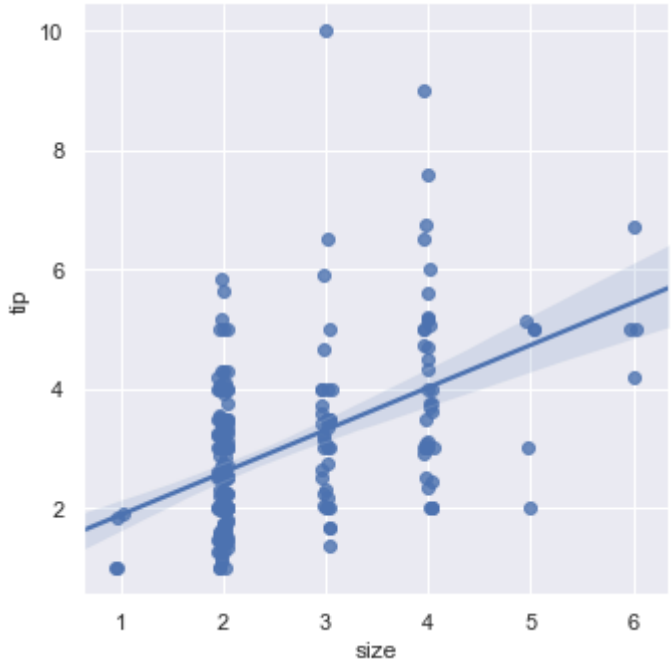
```

```

Entrée [215]:

```
sns.lmplot(x = 'size', y = 'tip', data = tips, x_jitter = 0.05)
```

Out[215]:

```
<seaborn.axisgrid.FacetGrid at 0x1f7e6aa9a58>
```



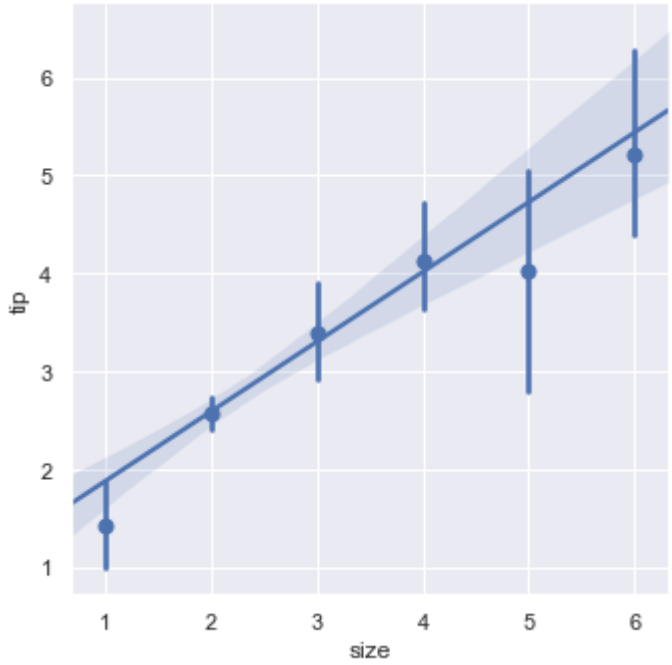
Entrée [ ]:

Entrée [218]:

```
sns.lmplot(x = 'size', y = 'tip', data = tips, x_estimator = np.mean)
```

Out[218]:

```
<seaborn.axisgrid.FacetGrid at 0x1f7e6fc7f28>
```



Entrée [220]:

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

Out[220]:

	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

Entrée [221]:

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

Out[221]:

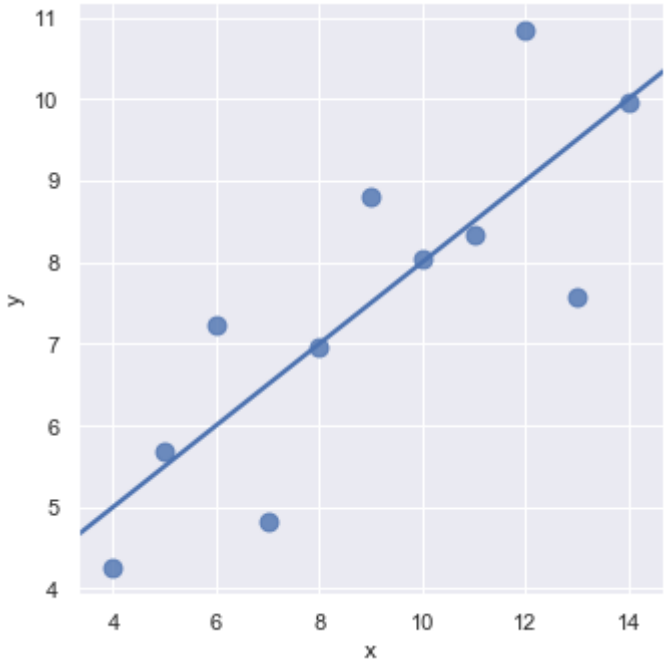
```
III    11
II     11
IV     11
I      11
Name: dataset, dtype: int64
```

Entrée [222]:

```
sns.lmplot(x = 'x', y = 'y', data = data.query("dataset == 'I'"), ci = None, scatter_kws={'
```

Out[222]:

```
<seaborn.axisgrid.FacetGrid at 0x1f7e6f57c88>
```

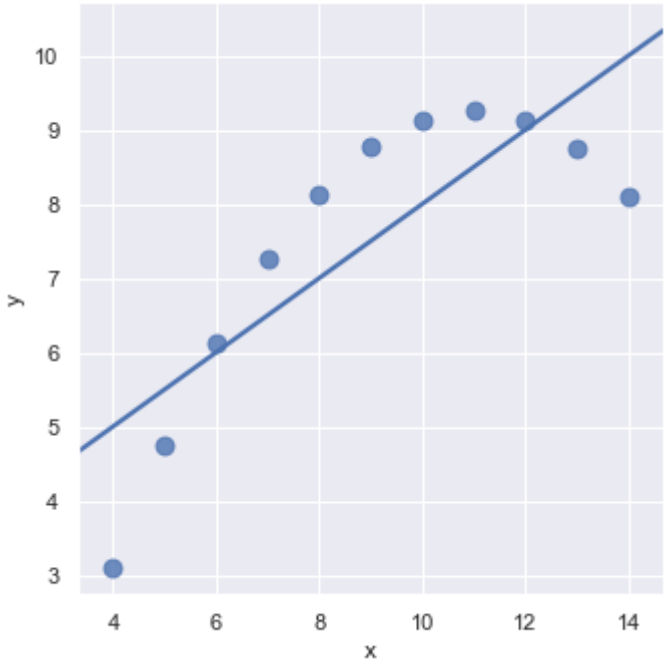


Entrée [226]:

```
sns.lmplot(x = 'x', y = 'y', data = data.query("dataset == 'II'"), ci = None, scatter_kws={
```

Out[226]:

```
<seaborn.axisgrid.FacetGrid at 0x1f7e728f5f8>
```



Entrée [ ]:

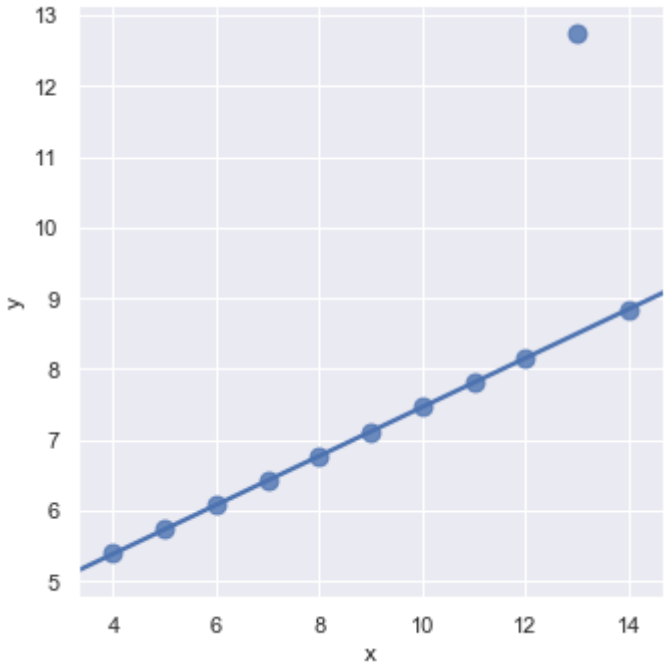
```

```

Entrée [228]:

```
sns.lmplot(x = 'x', y = 'y', data = data.query("dataset == 'III'"), ci = None, scatter_kws=
```

Out[228]:

```
<seaborn.axisgrid.FacetGrid at 0x1f7e8321f60>
```



Entrée [ ]:

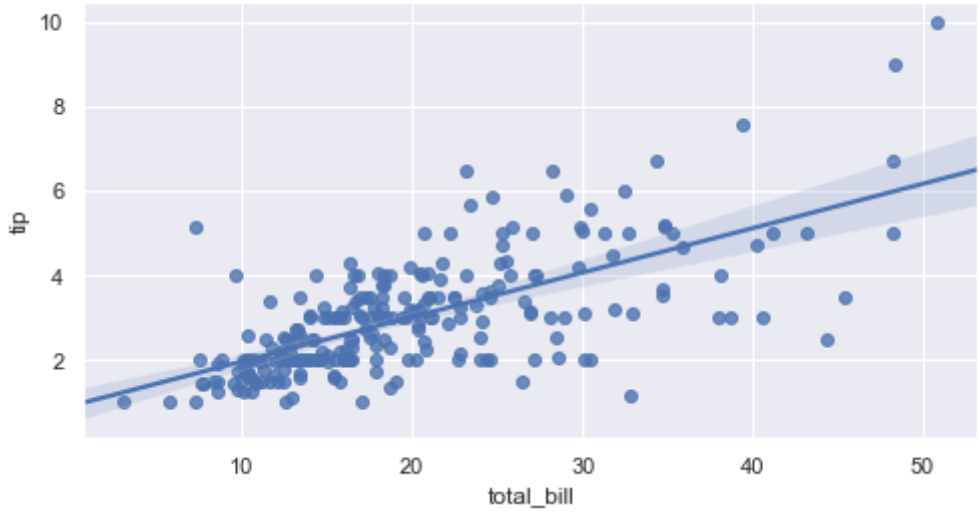
Entrée [ ]:

Entrée [240]:

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

Out[240]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1f7e8e29cf8>
```



Entrée [ ]:

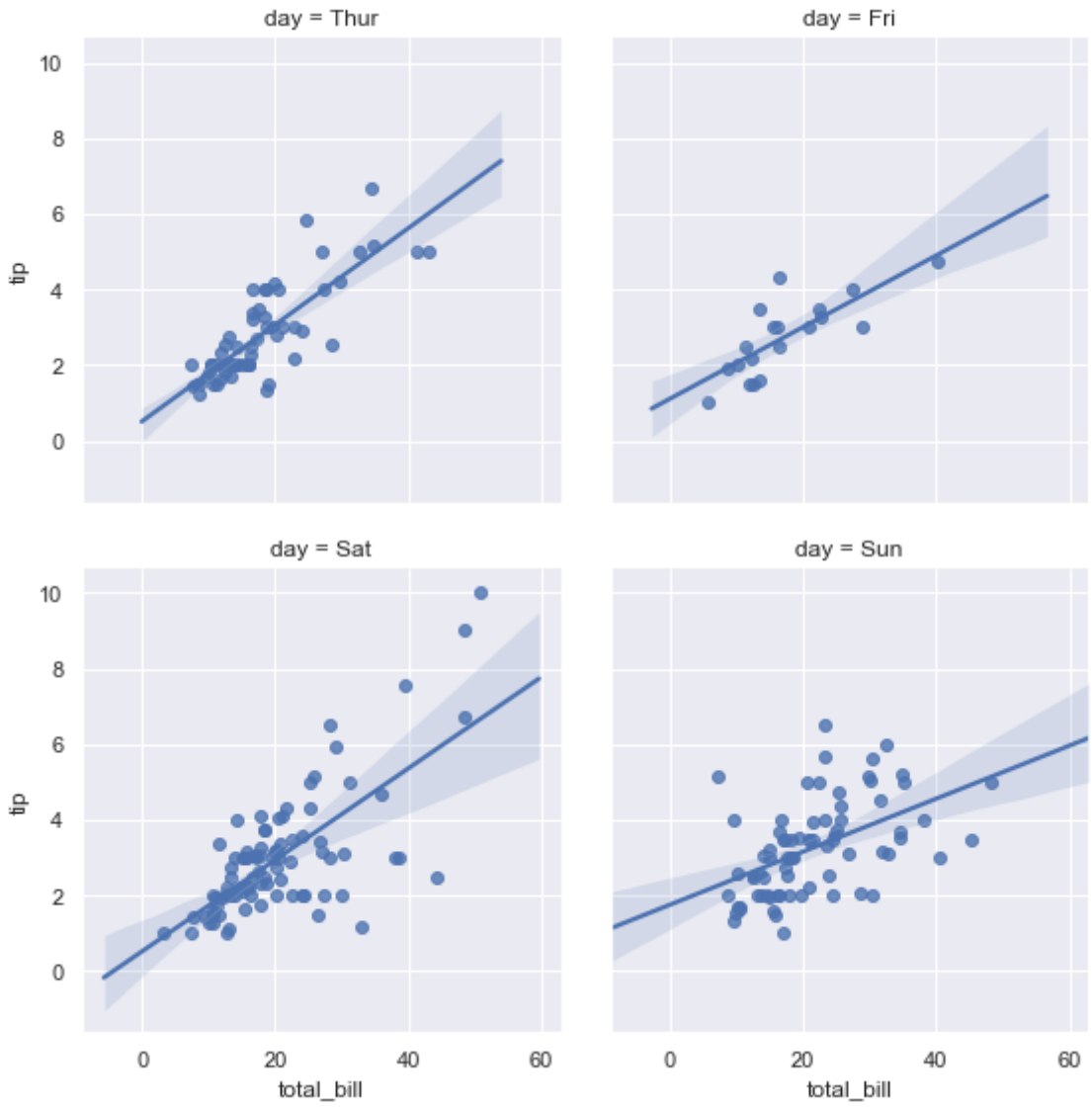
```

```

Entrée [244]:

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

Out[244]:

```
<seaborn.axisgrid.FacetGrid at 0x1f7ea9ce2e8>
```



Entrée [ ]:

```
```

5. Controlling Ploted Figure Aesthetics

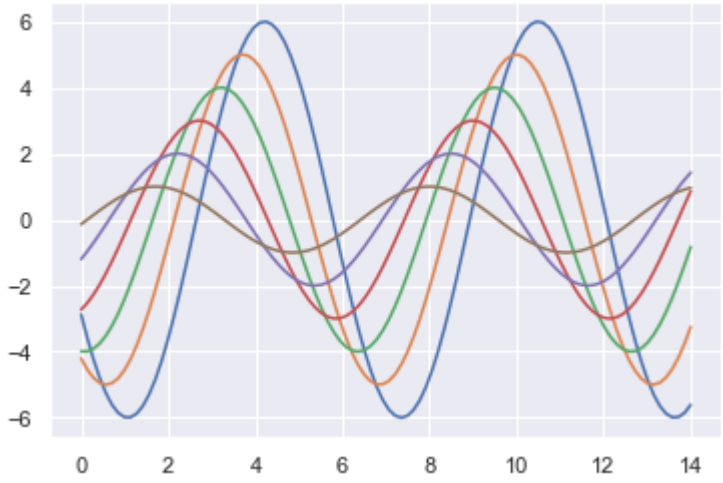
- figure styling
- axes styling
- color palettes
- etc..

Entrée [245]:

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

Entrée [247]:

```
sinplot(-1)
```

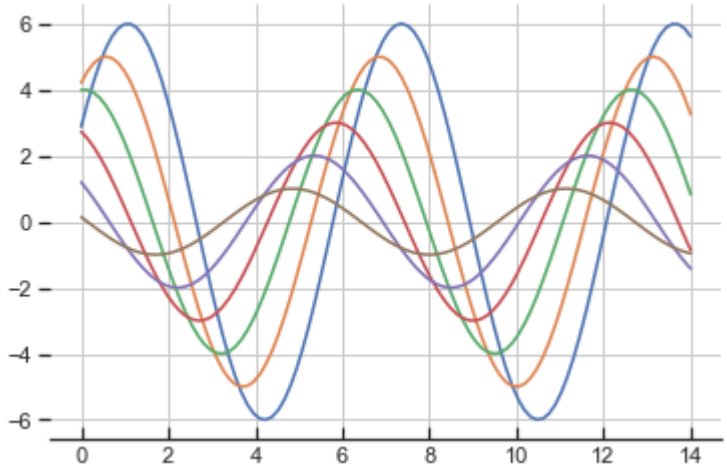


Entrée [ ]:

```
```

Entrée [260]:

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



Entrée [257]:

```
sns.axes_style()
```

Out[257]:

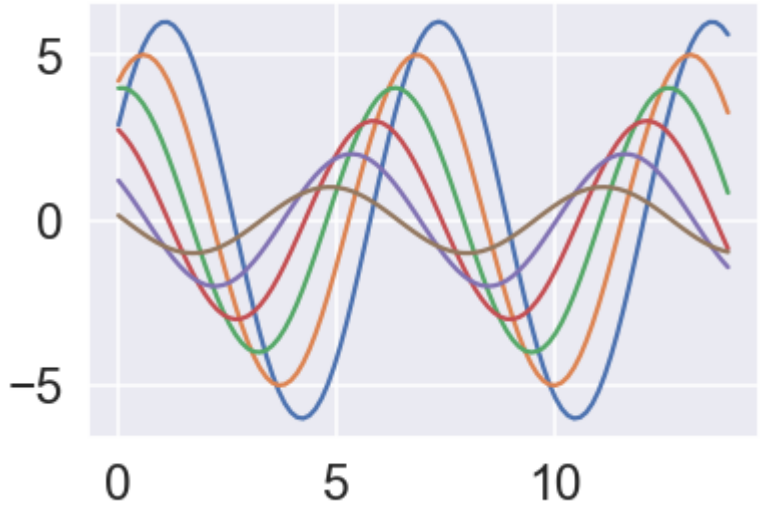
```
{'axes.facecolor': 'white',
 'axes.edgecolor': '.15',
 'axes.grid': False,
 'axes.axisbelow': True,
 'axes.labelcolor': '.15',
 'figure.facecolor': 'white',
 'grid.color': '.8',
 'grid.linestyle': '-',
 'text.color': '.15',
 'xtick.color': '.15',
 'ytick.color': '.15',
 'xtick.direction': 'out',
 '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}
```

Entrée [262]:

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

Entrée [267]:

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



Entrée [ ]:

```

```

Entrée [270]:

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

Entrée [271]:

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

Entrée [ ]:

```

```

Entrée [ ]:

```

```

Entrée [ ]:

Entrée [ ]:

Entrée [ ]:

Entrée [ ]:

Entrée [ ]:

Entrée [ ]:

Entrée [ ]:

Entrée [ ]:

Entrée [ ]:

Entrée [ ]:

Entrée [ ]:

Entrée [ ]:

Entrée [ ]:

Entrée [ ]:

Entrée [ ]:

Entrée [ ]:

Entrée [ ]:

Entrée [ ]:

Entrée [ ]:

Entrée [ ]:

Entrée [ ]:

Entrée [ ]:

Entrée [ ]:

Entrée [ ]:

Entrée [ ]:

Entrée [ ]:

Entrée [ ]:

Entrée [ ]:

Entrée [ ]:

Entrée [ ]:

Entrée [ ]:

Entrée [ ]:

Entrée [ ]:

Entrée [ ]:

Entrée [ ]:

Entrée [ ]:

Entrée [ ]:

Entrée [ ]:

Entrée [ ]:

Entrée [ ]:

Entrée [ ]:

Entrée [ ]:

Entrée [ ]:

Entrée [ ]:

Entrée [ ]:

Entrée [ ]:

Entrée [ ]:

Entrée [ ]:

Entrée [ ]:

Entrée [ ]:

Entrée [ ]:

Entrée [ ]:

Entrée [ ]:

Entrée [ ]:

Entrée [ ]:

Entrée [ ]:

Entrée [ ]:

Entrée [ ]:

Entrée [ ]:

Entrée [ ]:

Entrée [ ]:

Entrée [ ]:

Entrée [ ]:

Entrée [ ]:

Entrée [ ]:

Entrée [ ]:

Entrée [ ]:

Entrée [ ]:

Entrée [ ]:

Entrée [ ]:

Entrée [ ]:

Entrée [ ]:

Entrée [ ]:

Entrée [ ]:

Entrée [ ]:

Entrée [ ]:

Entrée [ ]:

Entrée [ ]: