SEABORN

```
Seaborn is an amazing data visualization library for statistical graphics
plotting in Python. It provides beautiful default styles and colour
palettes to make statistical plots more attractive. It is built on the top
of the matplotlib library and also closely integrated to the data
structures from pandas.
```

```
Installing
```

In [1]:

```
pip install seaborn
```

```
Requirement already satisfied: seaborn in c:\users\heman\appdata\roaming\py
thon\python310\site-packages (0.12.1)
Requirement already satisfied: numpy>=1.17 in c:\users\heman\appdata\local
\programs\python\python310\lib\site-packages (from seaborn) (1.24.4)
Requirement already satisfied: pandas>=0.25 in c:\users\heman\appdata\roami
ng\python\python310\site-packages (from seaborn) (1.5.0)
Requirement already satisfied: matplotlib!=3.6.1,>=3.1 in c:\users\heman\ap
pdata\local\programs\python\python310\lib\site-packages (from seaborn) (3.
8.3)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\heman\appdata\l
ocal\programs\python\python310\lib\site-packages (from matplotlib!=3.6.1,>=
3.1->seaborn) (1.2.0)
Requirement already satisfied: cycler>=0.10 in c:\users\heman\appdata\local
\programs\python\python310\lib\site-packages (from matplotlib!=3.6.1,>=3.1-
>seaborn) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\heman\appdata
\local\programs\python\python310\lib\site-packages (from matplotlib!=3.6.1,
>=3.1->seaborn) (4.33.3)
Requirement already satisfied: kiwisolver>=1.3.1 in c:\users\heman\appdata
\local\programs\python\python310\lib\site-packages (from matplotlib!=3.6.1,
>=3.1->seaborn) (1.4.2)
Requirement already satisfied: packaging>=20.0 in c:\users\heman\appdata\lo
cal\programs\python\python310\lib\site-packages (from matplotlib!=3.6.1,>=
3.1->seaborn) (21.3)
Requirement already satisfied: pillow>=8 in c:\users\heman\appdata\local\pr
ograms\python\python310\lib\site-packages (from matplotlib!=3.6.1,>=3.1->se
aborn) (9.5.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\heman\appdata\l
ocal\programs\python\python310\lib\site-packages (from matplotlib!=3.6.1,>=
3.1->seaborn) (3.0.7)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\heman\appda
ta\local\programs\python\python310\lib\site-packages (from matplotlib!=3.6.
1,>=3.1->seaborn) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in c:\users\heman\appdata\local
\programs\python\python310\lib\site-packages (from pandas>=0.25->seaborn)
(2022.1)
Requirement already satisfied: six>=1.5 in c:\users\heman\appdata\local\pro
grams\python\python310\lib\site-packages (from python-dateutil>=2.7->matplo
tlib!=3.6.1,>=3.1->seaborn) (1.16.0)
Note: you may need to restart the kernel to use updated packages.
```

In [ ]:

```
# Also, make sure you have the following dependencies installed on your comp

# Python 3.6+
# NumPy
# SciPy
# Pandas
# Matplotlib
```

In [4]:

```
import seaborn as sns
print(sns.get_dataset_names())
```

```
['anagrams', 'anscombe', 'attention', 'brain_networks', 'car_crashes', 'dia
monds', 'dots', 'dowjones', 'exercise', 'flights', 'fmri', 'geyser', 'glu
e', 'healthexp', 'iris', 'mpg', 'penguins', 'planets', 'seaice', 'taxis',
'tips', 'titanic', 'anagrams', 'anagrams', 'anscombe', 'anscombe', 'attenti
on', 'attention', 'brain_networks', 'brain_networks', 'car_crashes', 'car_c
rashes', 'diamonds', 'diamonds', 'dots', 'dots', 'dowjones', 'dowjones', 'e
xercise', 'exercise', 'flights', 'flights', 'fmri', 'fmri', 'geyser', 'geys
er', 'glue', 'glue', 'healthexp', 'healthexp', 'iris', 'iris', 'mpg', 'mp
g', 'penguins', 'penguins', 'planets', 'planets', 'seaice', 'seaice', 'taxi
s', 'taxis', 'tips', 'tips', 'titanic', 'titanic', 'anagrams', 'anscombe',
'attention', 'brain_networks', 'car_crashes', 'diamonds', 'dots', 'dowjone
s', 'exercise', 'flights', 'fmri', 'geyser', 'glue', 'healthexp', 'iris',
'mpg', 'penguins', 'planets', 'seaice', 'taxis', 'tips', 'titanic']
```

```
In [10]: df=sns.load_dataset('car_crashes')
         df
```

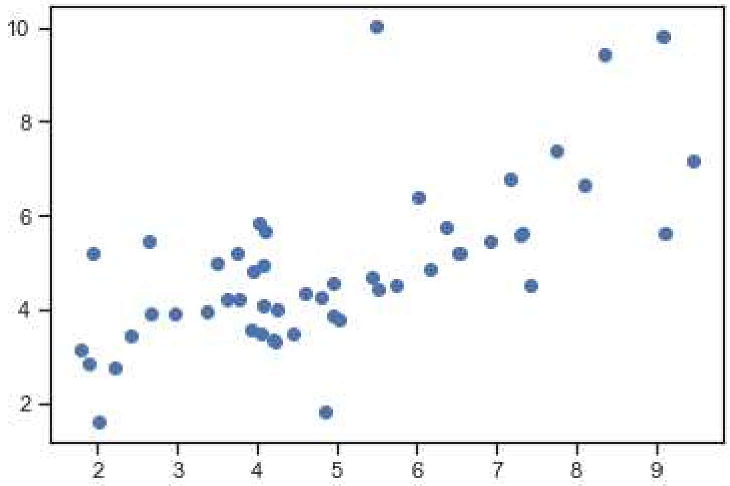
Out[10]:

	total	speeding	alcohol	not_distracted	no_previous	ins_premium	ins_losses	abbrev
0	18.8	7.332	5.640	18.048	15.040	784.55	145.08	AL
1	18.1	7.421	4.525	16.290	17.014	1053.48	133.93	AK
2	18.6	6.510	5.208	15.624	17.856	899.47	110.35	AZ
3	22.4	4.032	5.824	21.056	21.280	827.34	142.39	AR
4	12.0	4.200	3.360	10.920	10.680	878.41	165.63	CA
5	13.6	5.032	3.808	10.744	12.920	835.50	139.91	CO
6	10.8	4.968	3.888	9.396	8.856	1068.73	167.02	CT
7	16.2	6.156	4.860	14.094	16.038	1137.87	151.48	DE
8	5.9	2.006	1.593	5.900	5.900	1273.89	136.05	DC
9	17.9	3.759	5.191	16.468	16.826	1160.13	144.18	FL
10	15.6	2.964	3.900	14.820	14.508	913.15	142.80	GA
11	17.5	9.450	7.175	14.350	15.225	861.18	120.92	HI
12	15.3	5.508	4.437	13.005	14.994	641.96	82.75	ID
13	12.8	4.608	4.352	12.032	12.288	803.11	139.15	IL
14	14.5	3.625	4.205	13.775	13.775	710.46	108.92	IN
15	15.7	2.669	3.925	15.229	13.659	649.06	114.47	IA
16	17.8	4.806	4.272	13.706	15.130	780.45	133.80	KS
17	21.4	4.066	4.922	16.692	16.264	872.51	137.13	KY
18	20.5	7.175	6.765	14.965	20.090	1281.55	194.78	LA
19	15.1	5.738	4.530	13.137	12.684	661.88	96.57	ME
20	12.5	4.250	4.000	8.875	12.375	1048.78	192.70	MD
21	8.2	1.886	2.870	7.134	6.560	1011.14	135.63	MA
22	14.1	3.384	3.948	13.395	10.857	1110.61	152.26	MI
23	9.6	2.208	2.784	8.448	8.448	777.18	133.35	MN
24	17.6	2.640	5.456	1.760	17.600	896.07	155.77	MS
25	16.1	6.923	5.474	14.812	13.524	790.32	144.45	MO
26	21.4	8.346	9.416	17.976	18.190	816.21	85.15	MT
27	14.9	1.937	5.215	13.857	13.410	732.28	114.82	NE
28	14.7	5.439	4.704	13.965	14.553	1029.87	138.71	NV
29	11.6	4.060	3.480	10.092	9.628	746.54	120.21	NH
30	11.2	1.792	3.136	9.632	8.736	1301.52	159.85	NJ
31	18.4	3.496	4.968	12.328	18.032	869.85	120.75	NM
32	12.3	3.936	3.567	10.824	9.840	1234.31	150.01	NY
33	16.8	6.552	5.208	15.792	13.608	708.24	127.82	NC
34	23.9	5.497	10.038	23.661	20.554	688.75	109.72	ND
35	14.1	3.948	4.794	13.959	11.562	697.73	133.52	OH
36	19.9	6.368	5.771	18.308	18.706	881.51	178.86	OK
37	12.8	4.224	3.328	8.576	11.520	804.71	104.61	OR

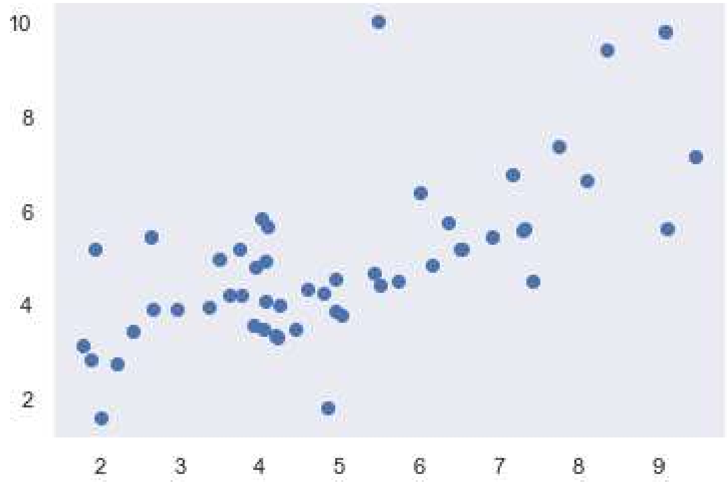
	total	speeding	alcohol	not_distracted	no_previous	ins_premium	ins_losses	abbrev
38	18.2	9.100	5.642	17.472	16.016	905.99	153.86	PA
39	11.1	3.774	4.218	10.212	8.769	1148.99	148.58	RI
40	23.9	9.082	9.799	22.944	19.359	858.97	116.29	SC
41	19.4	6.014	6.402	19.012	16.684	669.31	96.87	SD
42	19.5	4.095	5.655	15.990	15.795	767.91	155.57	TN
43	19.4	7.760	7.372	17.654	16.878	1004.75	156.83	TX
44	11.3	4.859	1.808	9.944	10.848	809.38	109.48	UT
45	13.6	4.080	4.080	13.056	12.920	716.20	109.61	VT
46	12.7	2.413	3.429	11.049	11.176	768.95	153.72	VA
47	10.6	4.452	3.498	8.692	9.116	890.03	111.62	WA
48	23.8	8.092	6.664	23.086	20.706	992.61	152.56	WV
49	13.8	4.968	4.554	5.382	11.592	670.31	106.62	WI
50	17.4	7.308	5.568	14.094	15.660	791.14	122.04	WY

In [20]:

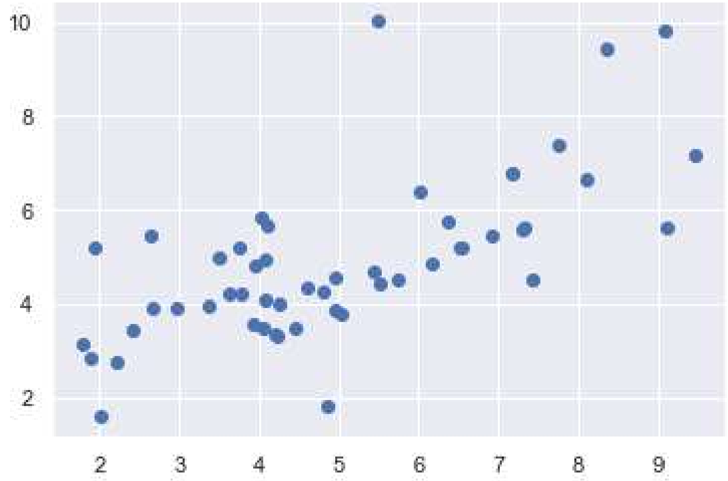
```
# set_style()

from matplotlib import pyplot as plt
plt.scatter(df.speeding,df.alcohol)
sns.set_style("ticks")
plt.show()
```



In [22]:
```
from matplotlib import pyplot as plt
plt.scatter(df.speeding,df.alcohol)
sns.set_style("dark")
plt.show()
```

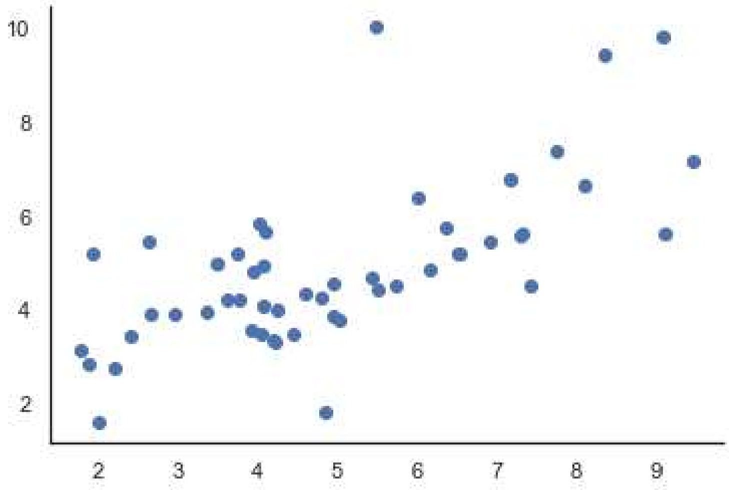


In [33]:
```
from matplotlib import pyplot as plt
plt.scatter(df.speeding,df.alcohol)
sns.set_style("darkgrid")
plt.show()
```



In [31]:

```
from matplotlib import pyplot as plt
plt.scatter(df.speeding,df.alcohol)
sns.set_style("white")
#despine
sns.despine()
plt.show()
```



In [34]:

```
param=sns.axes_style()
param
```

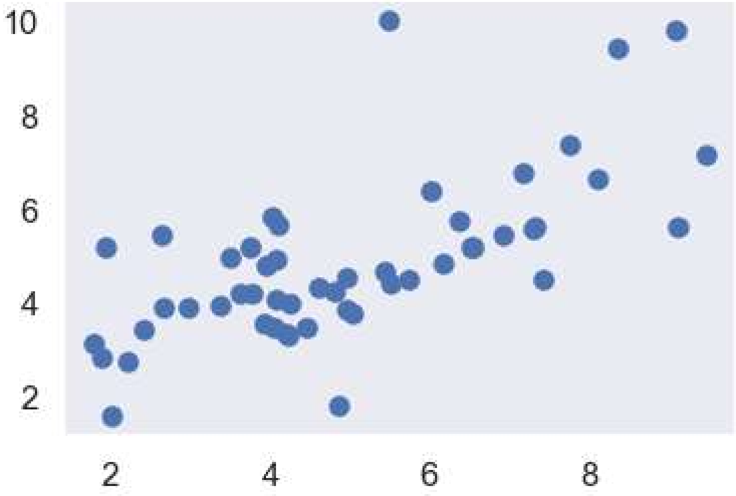
Out[34]:

```
{'axes.facecolor': '#EAEAF2',
 'axes.edgecolor': 'white',
 'axes.grid': True,
 'axes.axisbelow': True,
 'axes.labelcolor': '.15',
 'figure.facecolor': 'white',
 'grid.color': 'white',
 'grid.linestyle': '-',
 'text.color': '.15',
 'xtick.color': '.15',
 'ytick.color': '.15',
 'xtick.direction': 'out',
 'ytick.direction': 'out',
 'lines.solid_capstyle': <CapStyle.round: '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': False,
 'xtick.top': False,
 'ytick.left': False,
 'ytick.right': False,
 'axes.spines.left': True,
 'axes.spines.bottom': True,
 'axes.spines.right': True,
 'axes.spines.top': True}
```

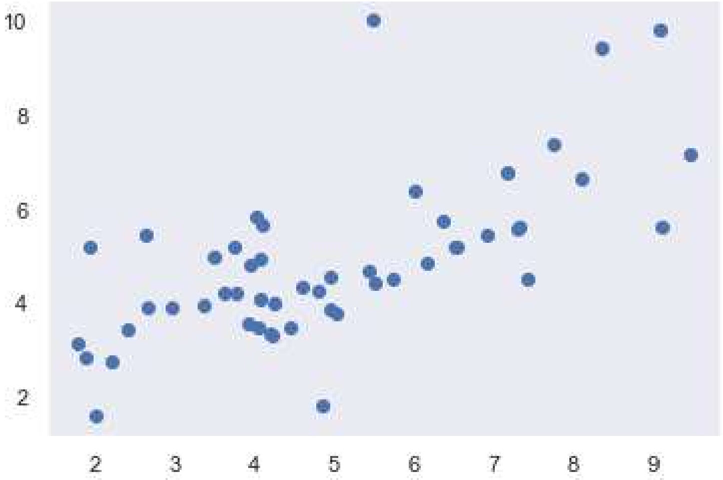
```
set_context()
#increase size when go below
```

```
Paper
Notebook
Talk
Poster
```

In [47]:
```
plt.scatter(df.speeding,df.alcohol)
sns.set_style("dark")
sns.set_context("talk")
plt.show()
```

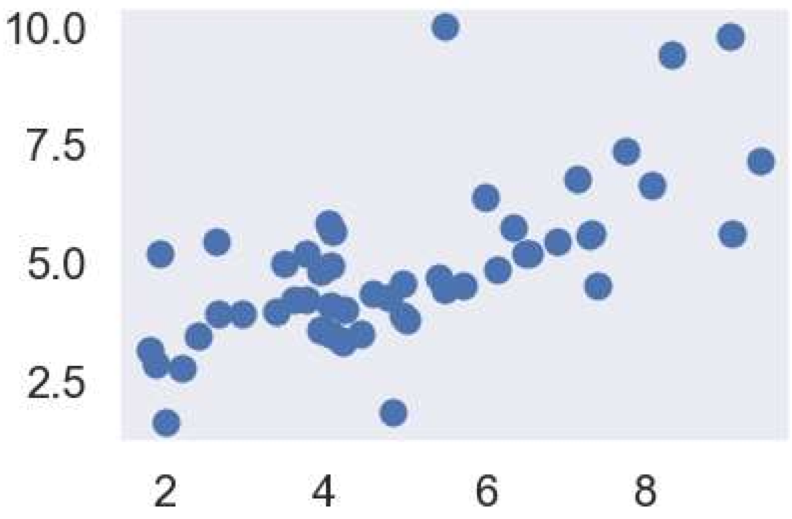


In [49]:
```
plt.scatter(df.speeding,df.alcohol)
sns.set_style("dark")
sns.set_context("notebook")
plt.show()
```



In [51]:

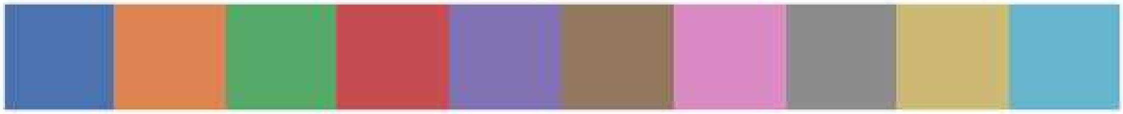
```
plt.scatter(df.speeding,df.alcohol)
sns.set_style("dark")
sns.set_context("poster")
plt.show()
```



Palette

In [52]:

```
# color_palette
sns.palplot(sns.color_palette("deep", 10))
```



In [57]:

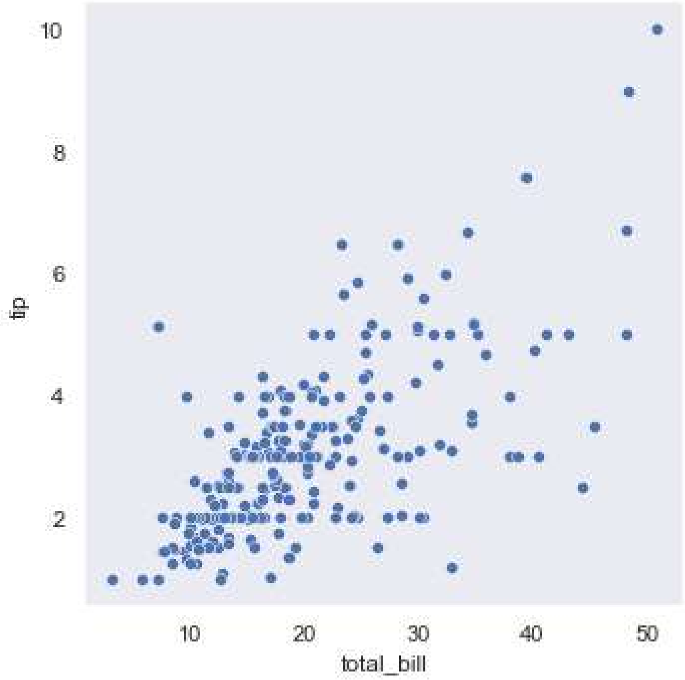
```
tips = sns.load_dataset("tips")
tips.head()
```

Out[57]:

	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 [63]:
```
sns.set_context("notebook")
sns.relplot(data=tips,x="total_bill",y="tip")
```

Out[63]:
```
<seaborn.axisgrid.FacetGrid at 0x293a979ef80>
```

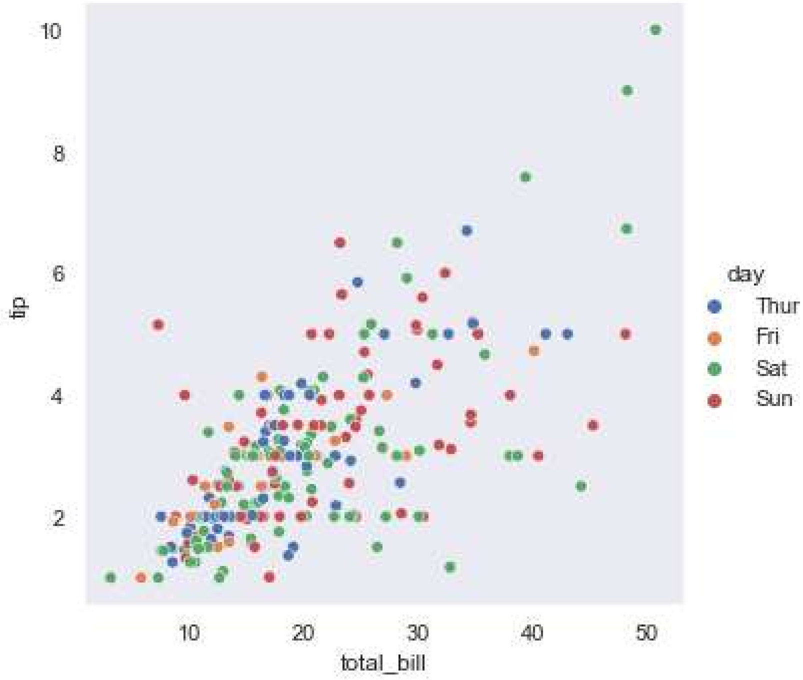


hue semantic

```
kind=line,scatter,hex,kde,...(default=>scatter)
errorbar = None to get just the line without any highlighted portion.
"errorcolor,errorwidth,errorbar"
capsize=errorbar cap size
```

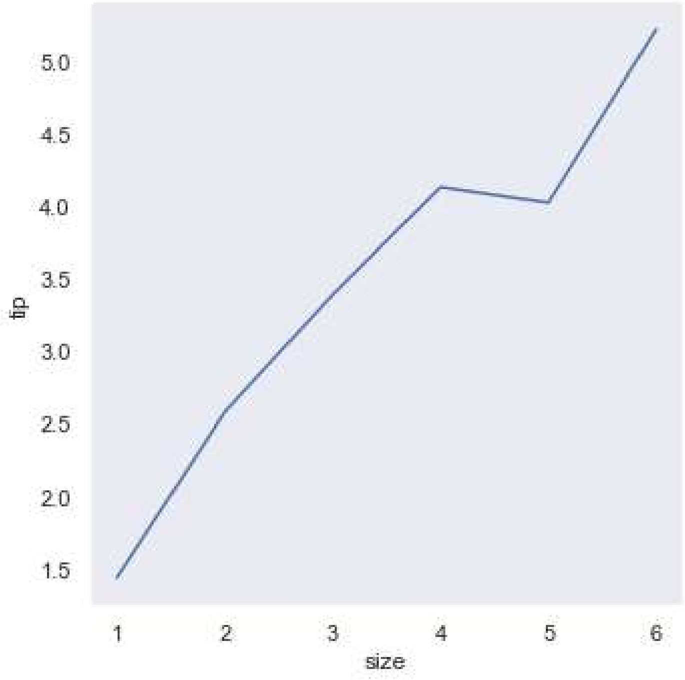
```
In [65]: #hue (third variabe=> diff in color)
         sns.relplot(data=tips, x="total_bill", y="tip", hue="day")
```

```
Out[65]: <seaborn.axisgrid.FacetGrid at 0x293a979dc90>
```



```
In [80]: sns.relplot(data=tips, x="size", y="tip",kind="line",errorbar=None)
```

```
Out[80]: <seaborn.axisgrid.FacetGrid at 0x293ab1db490>
```

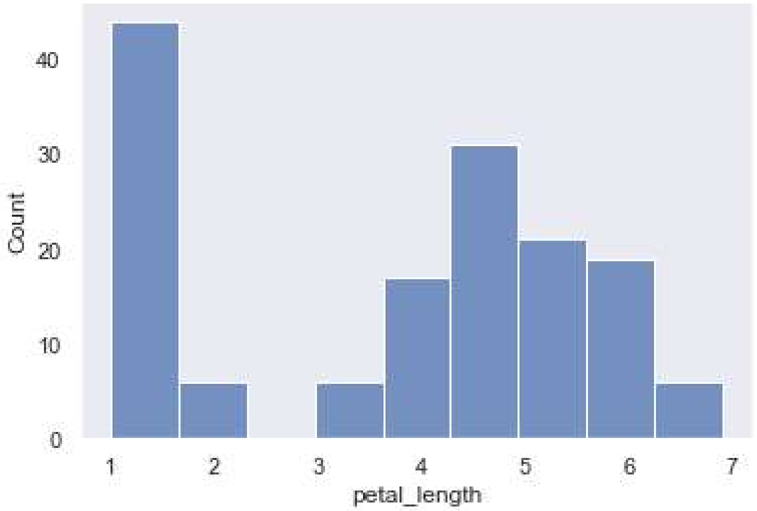


Histogram

Histograms represent the data distribution by forming bins along with the range of the data and then drawing bars to show the number of observations that fall in each bin.In Seaborn we use displot() or histplot() function to plot histograms.

```
In [77]: df = sns.load_dataset('iris')
         sns.histplot(df['petal_length'])
```

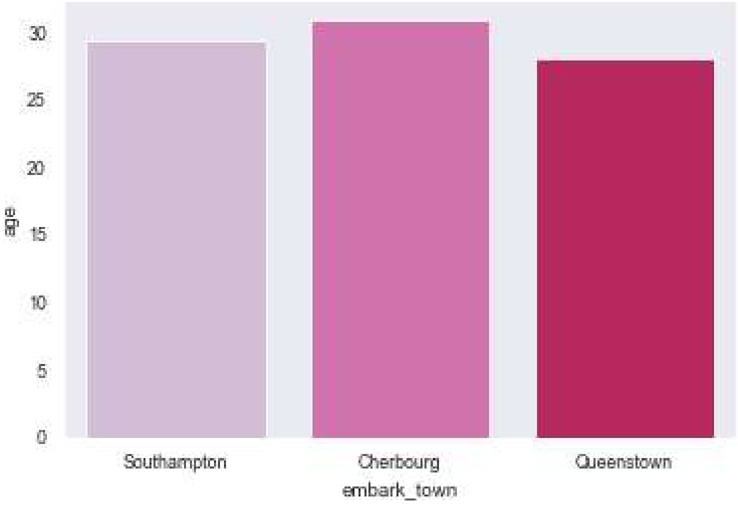
```
Out[77]: <Axes: xlabel='petal_length', ylabel='Count'>
```



Bar Plot

In [85]:

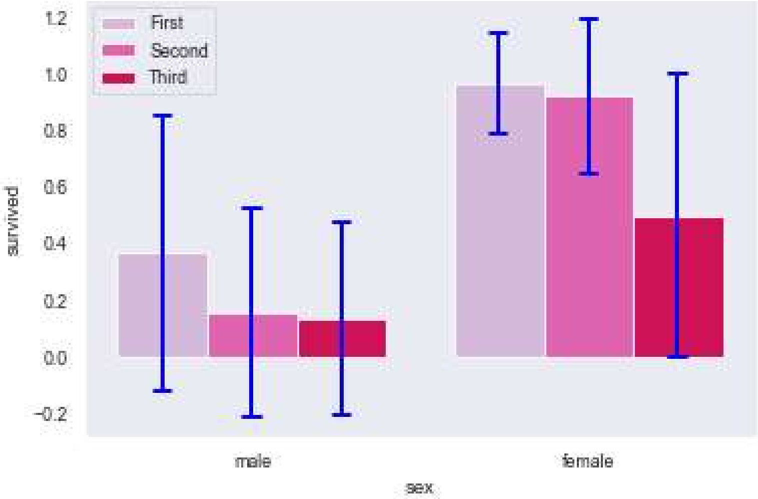
```
# Vertical barplot

sns.set_context('paper')

# load dataset
titanic = sns.load_dataset('titanic')
# create plot
sns.barplot(x = 'embark_town', y = 'age', data = titanic,
            palette = 'PuRd',errorbar=None
            )
plt.show()
print(titanic.columns)
```



```
Index(['survived', 'pclass', 'sex', 'age', 'sibsp', 'parch', 'fare',
       'embarked', 'class', 'who', 'adult_male', 'deck', 'embark_town',
       'alive', 'alone'],
      dtype='object')
```

In [102]:

```
# errorbar = None to get just the line without any highlighted portion.
# "errorcolor,errorwidth,errorbar"
# capsize=errorbar cap size

sns.barplot(x = 'sex', y = 'survived', hue = 'class', data = titanic,
            palette = 'PuRd',
            order = ['male', 'female'],
            capsize = 0.05,
            saturation = 8,
            errcolor = 'blue', errwidth = 2,
            errorbar= 'sd'
            )
plt.legend()
plt.show()
```

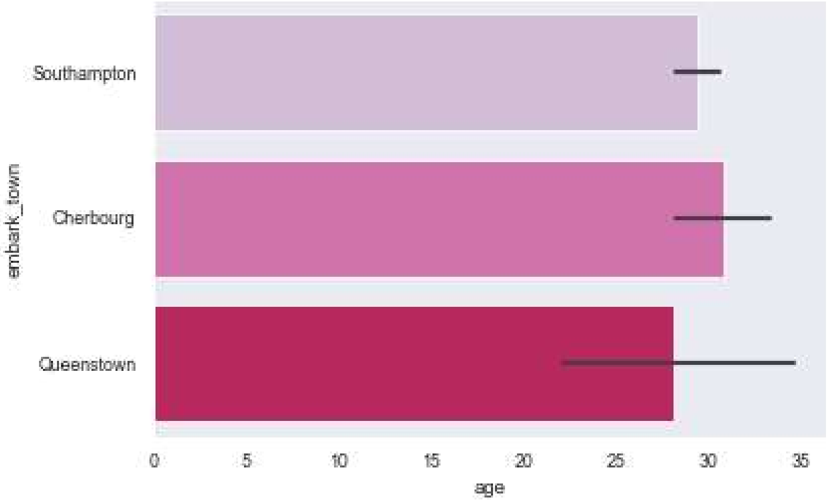


In [ ]:

```
# xaxis=  x
# yaxis=  y
# third variable=  hue
# colors=  palette
# orientation   =orient
```

In [99]:

```
#Horizontal barplot
sns.barplot(x = 'age', y = 'embark_town', data = titanic,
            palette = 'PuRd', orient = 'h',
            )
plt.show()
```

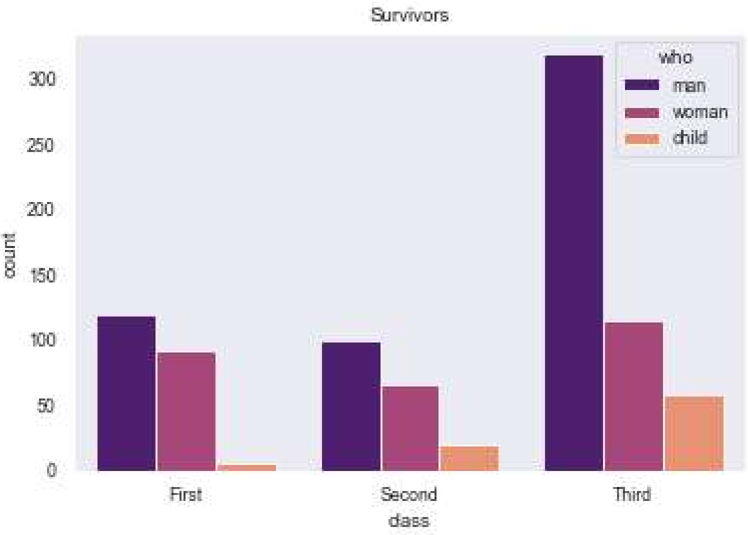


Count plot

The count plot can be thought of as a histogram across a categorical variable

In [103]:

```
sns.countplot(x = 'class', hue = 'who', data = titanic, palette = 'magma')
plt.title('Survivors')
plt.show()
```

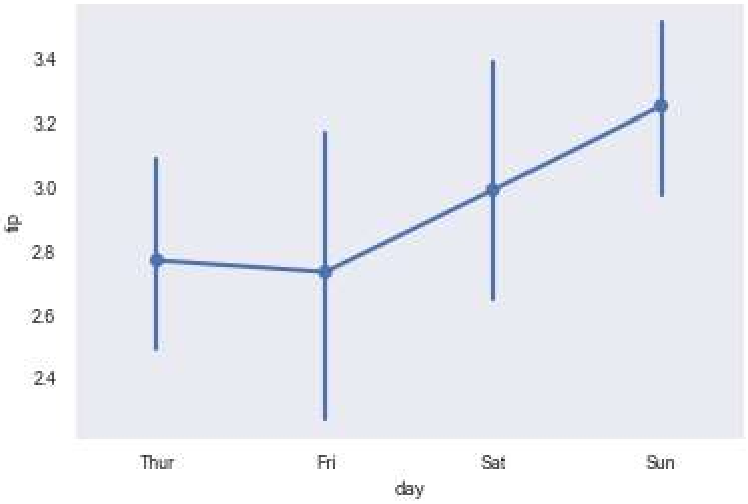


Point Plot

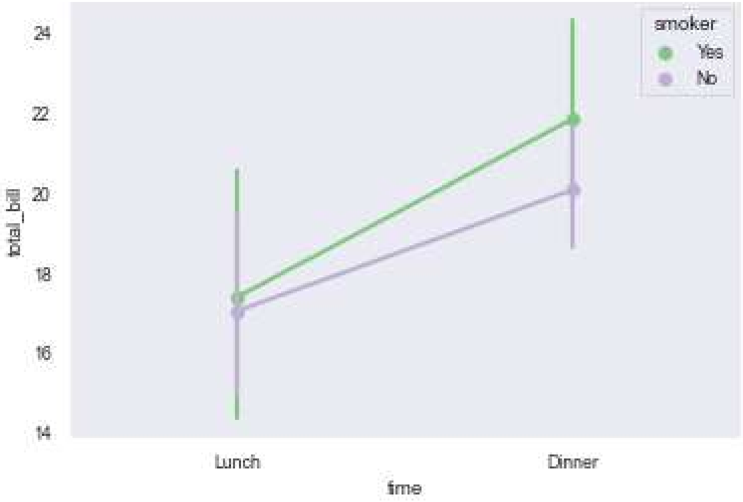
Point plot is used to show point estimates and confidence intervals using scatter plot glyphs. A point plot represents an estimate of central tendency for a numeric variable by the position of scatter plot points and provides some indication of the uncertainty around that estimate using error bars

In [104]:

```
data = sns.load_dataset("tips")
sns.pointplot(x="day", y="tip", data=data)
plt.show()
```



```
In [106]: sns.pointplot(x="time", y="total_bill", hue="smoker",
                              data=data, palette="Accent")

Out[106]: <Axes: xlabel='time', ylabel='total_bill'>
```



Joint Plot

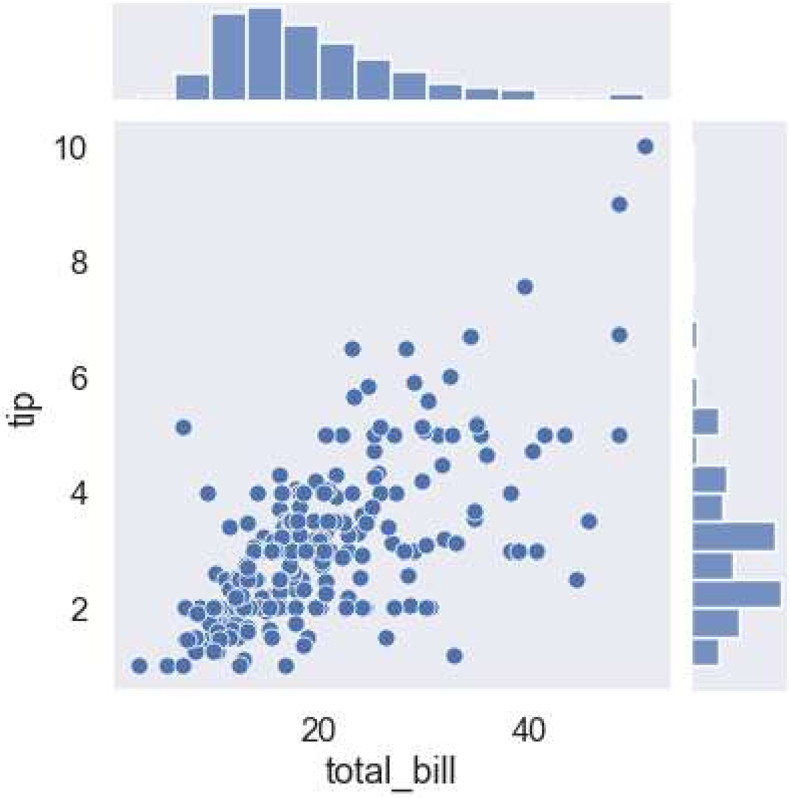
Joint Plot draws a plot of two variables with bivariate and univariate graphs. It uses the Scatter Plot and Histogram. Joint Plot can also display data using Kernel Density Estimate (KDE) and Hexagons.

In [110]:

```
# hex, reg, kde
sns.set_context("talk")
sns.set_style("dark")
tips=sns.load_dataset('tips')
sns.jointplot(x='total_bill', y='tip',data=tips)
```

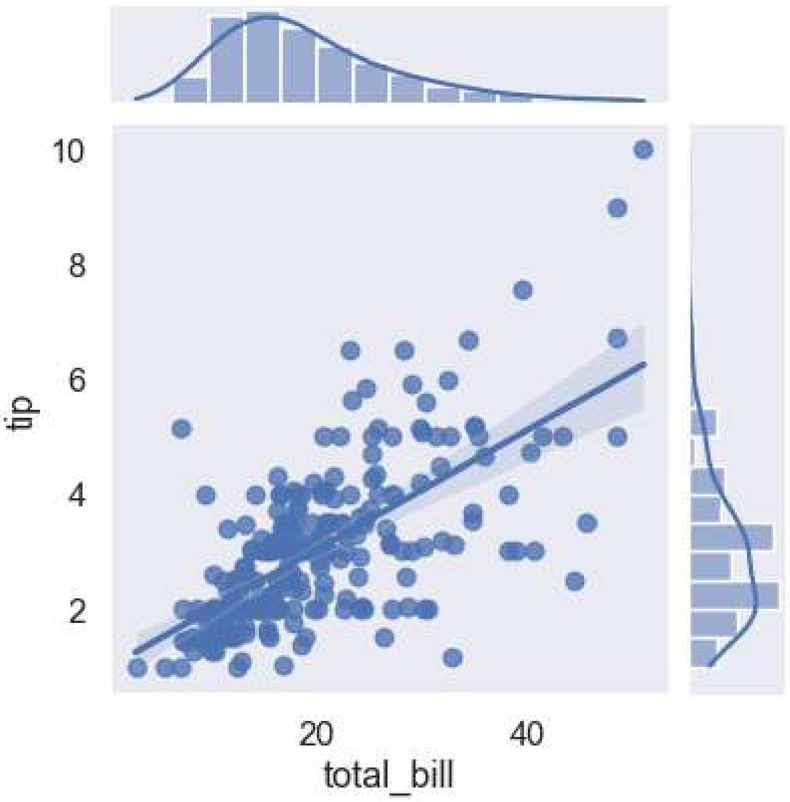
Out[110]:

```
<seaborn.axisgrid.JointGrid at 0x293b0f63dc0>
```



In [111]:
```
# Add regression line to scatter plot and kernel density estimate to histogr
sns.jointplot(x='total_bill', y='tip', data=tips, kind='reg')
```

Out[111]:
```
<seaborn.axisgrid.JointGrid at 0x293b10c8790>
```

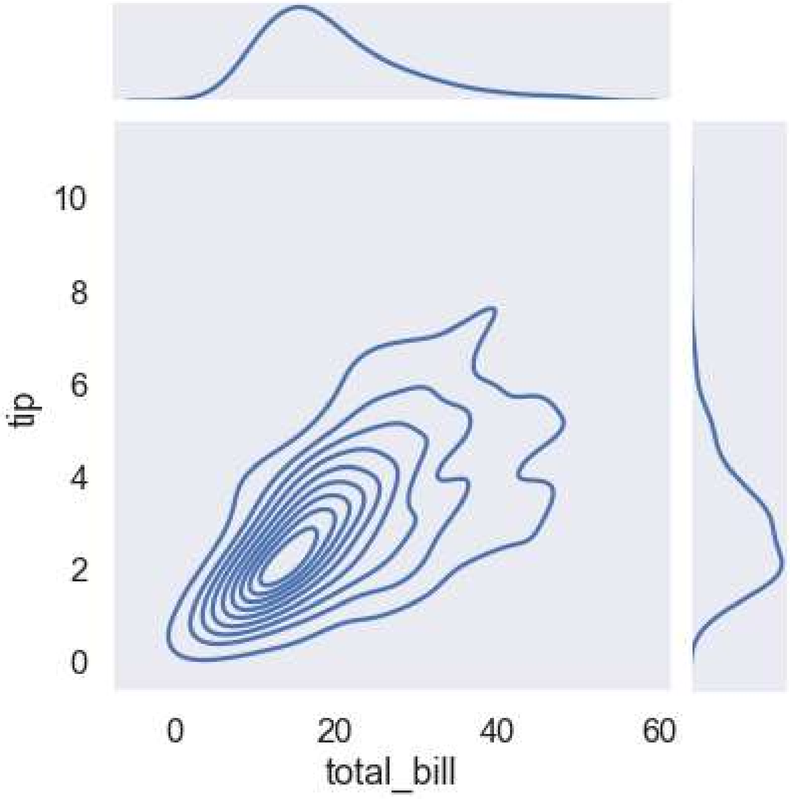


In [114]:

```
# Display kernel density estimate instead of scatter plot and histogram
sns.set_style("dark")
sns.jointplot(x='total_bill', y='tip', data=tips, kind='kde')
```

Out[114]:

```
<seaborn.axisgrid.JointGrid at 0x293b10b5840>
```

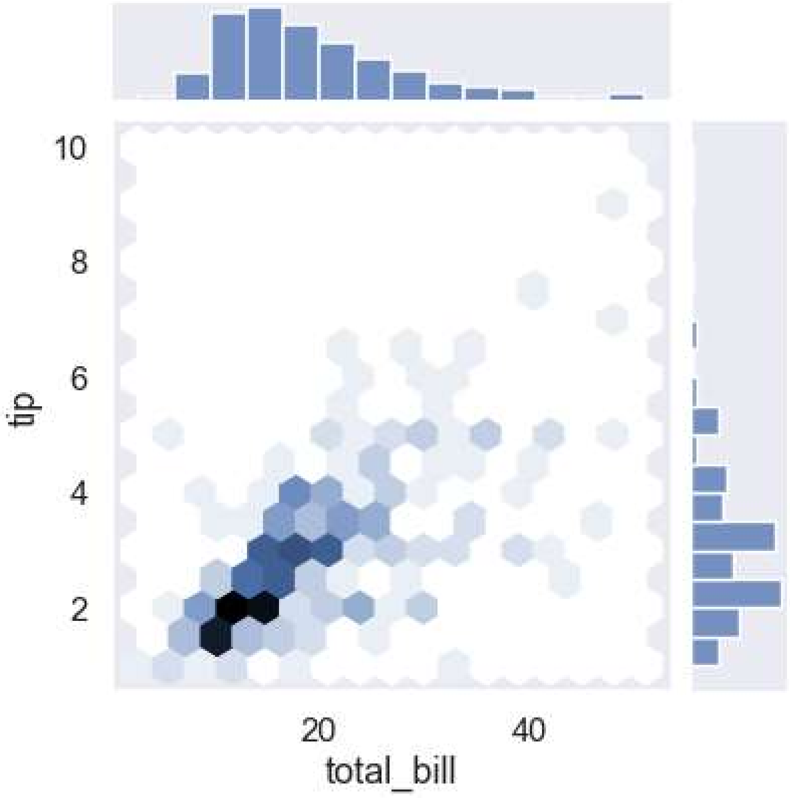


In [117]:

```
# Display hexagons instead of points in scatter plot
sns.jointplot(x='total_bill', y='tip', data=tips, kind='hex')
```

Out[117]:

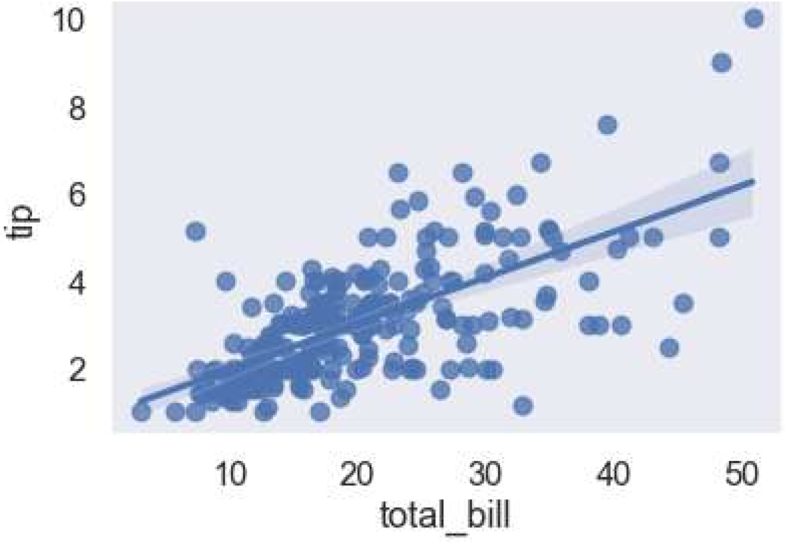
```
<seaborn.axisgrid.JointGrid at 0x293affae8c0>
```



Regplot

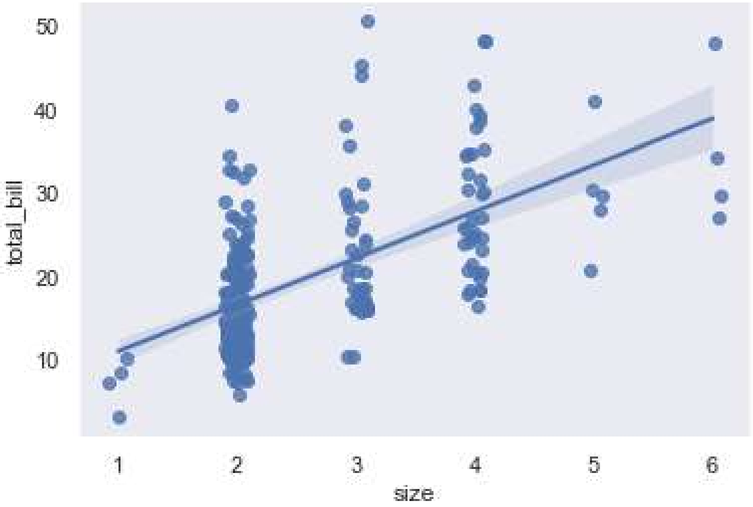
```
Regplot is one of the functions in Seaborn that are used to visualize the
linear relationship as determined through regression. Also, you‘ll see a
slightly shaded portion around the regression line which indicates how
much the pints are scattered around a certain area. Here are few of the
examples

Now we will plot a discrete x variable and add some jitter. Here you can
see that the areas where points are more densely populated have less
shaded portion around the regression line and shaded portion is more
spread where the points are more scattered.
```

```
In [118]: tips = sns.load_dataset("tips")
          ax = sns.regplot(x="total_bill", y="tip", data=tips)
```

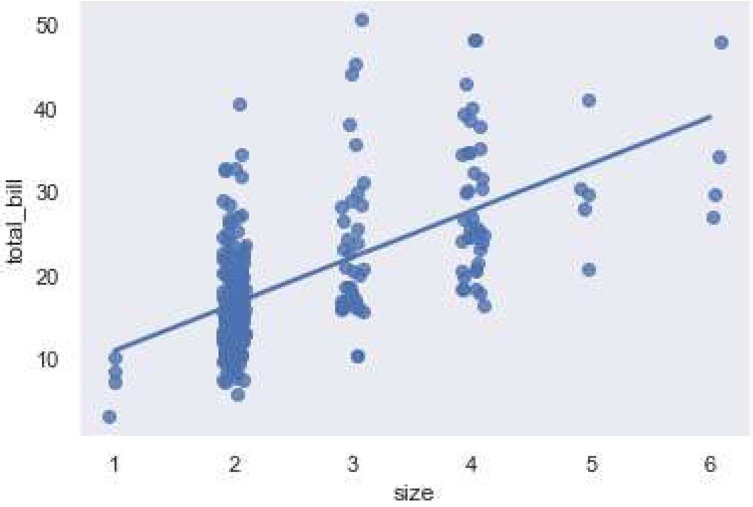


```
In [120]: sns.set_context("notebook")
          sns.regplot(x="size", y="total_bill", data=tips, x_jitter=0.1)
```

```
Out[120]: <Axes: xlabel='size', ylabel='total_bill'>
```



```
In [122]: sns.regplot(x="size", y="total_bill", data=tips, x_jitter=0.1,ci=None
                  )

Out[122]: <Axes: xlabel='size', ylabel='total_bill'>
```


total_bill
size