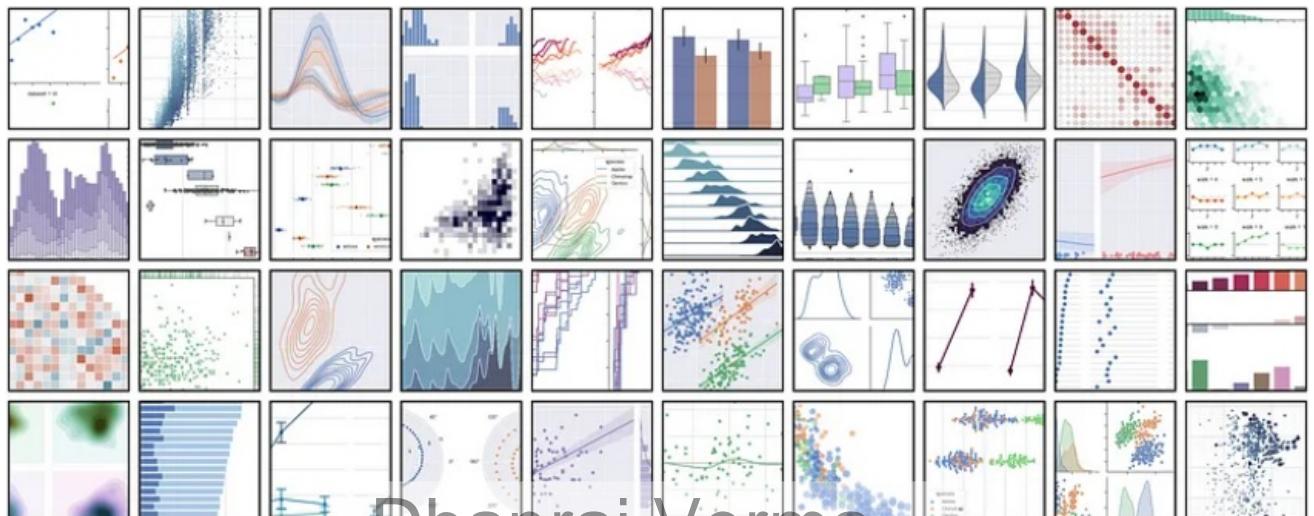


Seaborn



Dhanraj Verma

What is seaborn in python?

Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics. Seaborn is built on top of matplotlib and integrates closely with pandas data structures. It helps you explore and understand your data.

Seaborn provides a variety of plot types, including:

- Relational plots: These plots show the relationship between two variables. Examples include scatter plots, line plots, and bar charts.
- Categorical plots: These plots show the distribution of a categorical variable. Examples include histograms and box plots.
- Distribution plots: These plots show the distribution of a continuous variable. Examples include histograms and kernel density estimates.
- Regression plots: These plots show the relationship between a dependent variable and one or more independent variables. Examples include linear regression plots and logistic regression plots.
- Matrix plots: These plots show the relationship between multiple variables. Examples include heatmaps and correlation matrices.

Seaborn also provides a number of functions for customizing the appearance of your plots. For example, you can change the colors, fonts, and sizes of the elements in your plot.

Relational Plot

- to see the statistical relation between 2 or more variable
- Bivariate Analysis

Plots under this section

- scatterplot
- lineplot

```
In [1]: import seaborn as sns  
import matplotlib.pyplot as plt  
import plotly.express as px
```

```
In [2]: tips = sns.load_dataset('tips')
```

```
tips.head()
```

```
Out[2]:
```

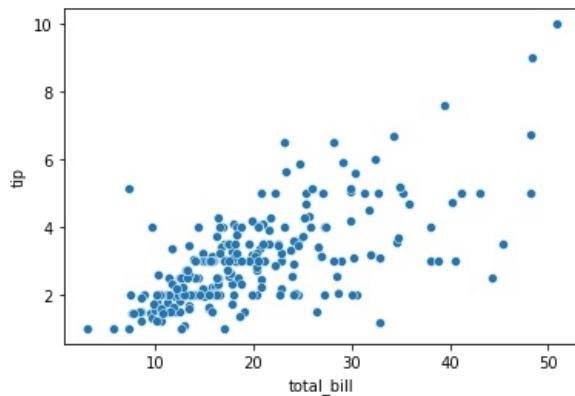
	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4

```
In [8]:
```

```
# scatter -> axis level function
sns.color_palette("pastel")
sns.scatterplot(x='total_bill', y='tip', data=tips)
```

```
Out[8]:
```

```
<AxesSubplot:xlabel='total_bill', ylabel='tip'>
```

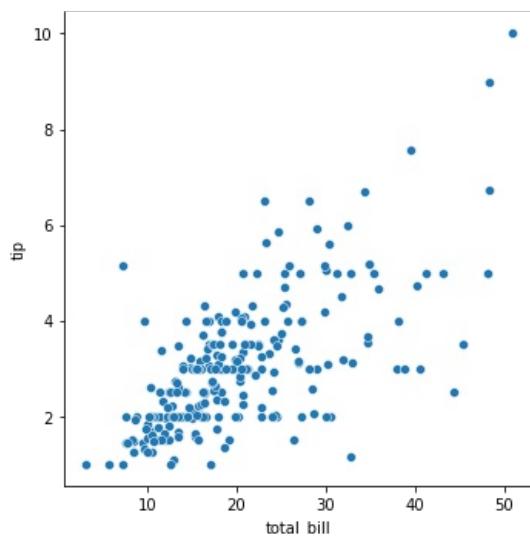


```
In [10]:
```

```
# scatter -> figure level function
sns.relplot(x='total_bill', y='tip', data=tips, kind='scatter')
```

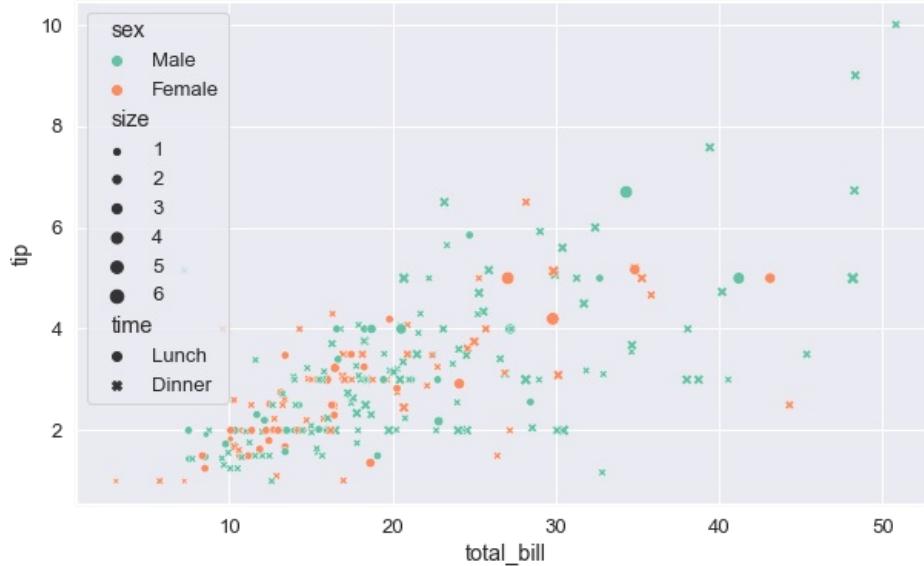
```
Out[10]:
```

```
<seaborn.axisgrid.FacetGrid at 0x1c7288bbf70>
```



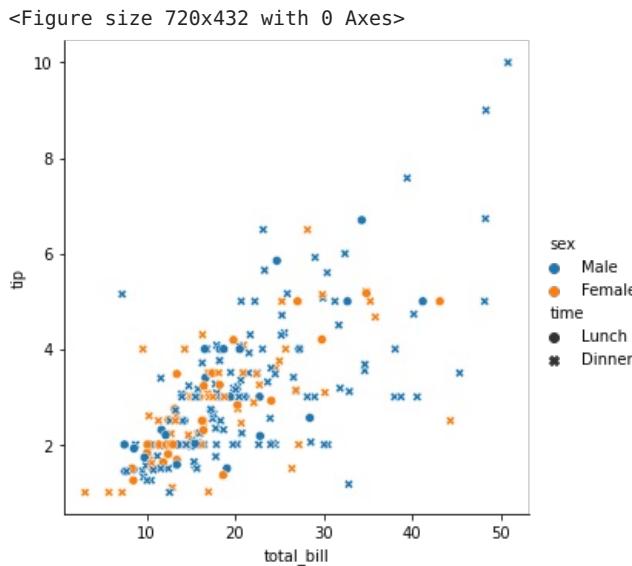
```
In [7]:
```

```
plt.figure(figsize=(10, 6))
sns.set_theme(style='darkgrid', palette='Set2', font_scale=1.2)
sns.scatterplot(x='total_bill', y='tip', data=tips,
                hue='sex', style='time', size='size')
plt.show()
```



```
In [20]: sns.relplot(x='total_bill', y='tip', data=tips,
                 kind='scatter', hue='sex', style='time')
```

```
Out[20]: <seaborn.axisgrid.FacetGrid at 0x1c72a1629d0>
```



Line plot

```
In [3]: df = px.data.gapminder()
```

```
In [36]: df.head()
```

	country	continent	year	lifeExp	pop	gdpPercap	iso_alpha	iso_num
0	Afghanistan	Asia	1952	28.801	8425333	779.445314	AFG	4
1	Afghanistan	Asia	1957	30.332	9240934	820.853030	AFG	4
2	Afghanistan	Asia	1962	31.997	10267083	853.100710	AFG	4
3	Afghanistan	Asia	1967	34.020	11537966	836.197138	AFG	4
4	Afghanistan	Asia	1972	36.088	13079460	739.981106	AFG	4

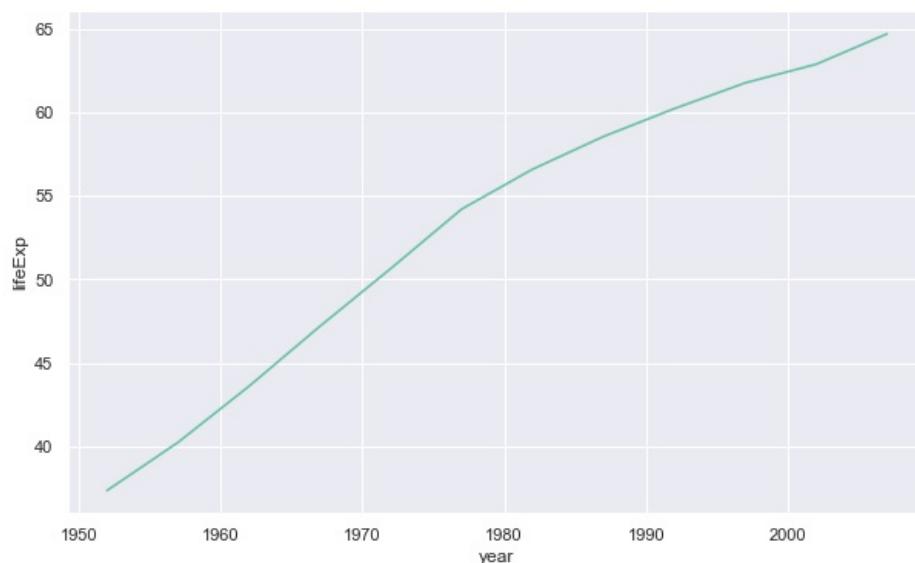
```
In [4]: df1 = df[df['country'] == 'India']
df1
```

Out[4]:

	country	continent	year	lifeExp	pop	gdpPercap	iso_alpha	iso_num
696	India	Asia	1952	37.373	372000000	546.565749	IND	356
697	India	Asia	1957	40.249	409000000	590.061996	IND	356
698	India	Asia	1962	43.605	454000000	658.347151	IND	356
699	India	Asia	1967	47.193	506000000	700.770611	IND	356
700	India	Asia	1972	50.651	567000000	724.032527	IND	356
701	India	Asia	1977	54.208	634000000	813.337323	IND	356
702	India	Asia	1982	56.596	708000000	855.723538	IND	356
703	India	Asia	1987	58.553	788000000	976.512676	IND	356
704	India	Asia	1992	60.223	872000000	1164.406809	IND	356
705	India	Asia	1997	61.765	959000000	1458.817442	IND	356
706	India	Asia	2002	62.879	1034172547	1746.769454	IND	356
707	India	Asia	2007	64.698	1110396331	2452.210407	IND	356

In [44]:

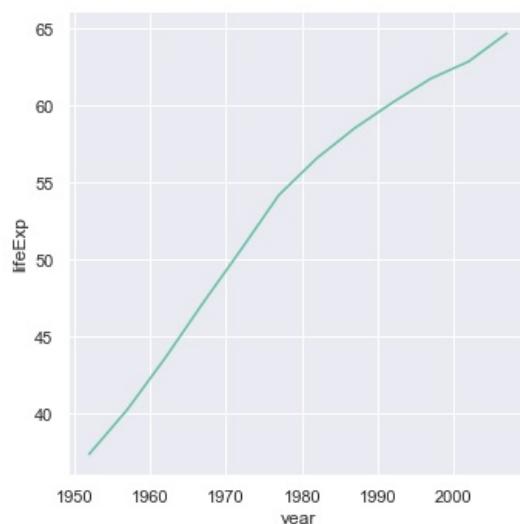
```
# axis level func
plt.figure(figsize=(10, 6))
sns.lineplot(x='year', y='lifeExp', data=df1)
plt.show()
```



In [46]:

```
# figure level
sns.relplot(x='year', y='lifeExp', data=df1, kind='line')
```

Out[46]:



In [5]:

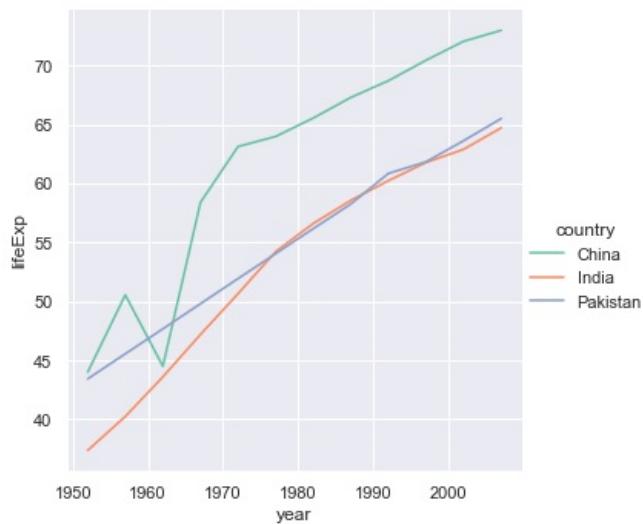
```
df2 = df[df['country'].isin(['India', 'Pakistan', 'China'])]
df2.head()
```

```
Out[5]:
```

	country	continent	year	lifeExp	pop	gdpPercap	iso_alpha	iso_num
288	China	Asia	1952	44.00000	556263527	400.448611	CHN	156
289	China	Asia	1957	50.54896	637408000	575.987001	CHN	156
290	China	Asia	1962	44.50136	665770000	487.674018	CHN	156
291	China	Asia	1967	58.38112	754550000	612.705693	CHN	156
292	China	Asia	1972	63.11888	862030000	676.900092	CHN	156

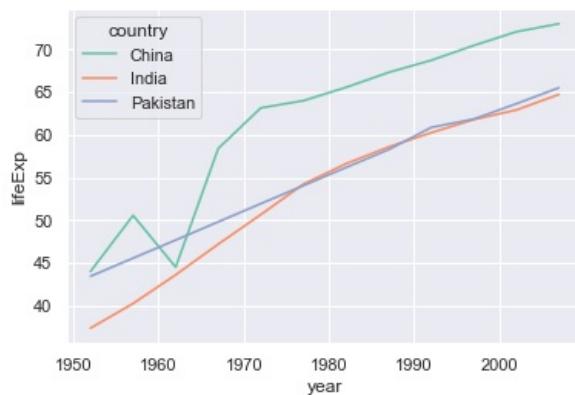
```
In [56]: sns.relplot(x='year', y='lifeExp', data=df2, hue='country', kind='line')
```

```
Out[56]: <seaborn.axisgrid.FacetGrid at 0x1c72e3f54c0>
```



```
In [57]: sns.lineplot(x='year', y='lifeExp', data=df2, hue='country')
```

```
Out[57]: <AxesSubplot:xlabel='year', ylabel='lifeExp'>
```



```
In [6]: df3 = df[df['country'].isin(['India', 'Brazil', 'Germany'])]
df3
```

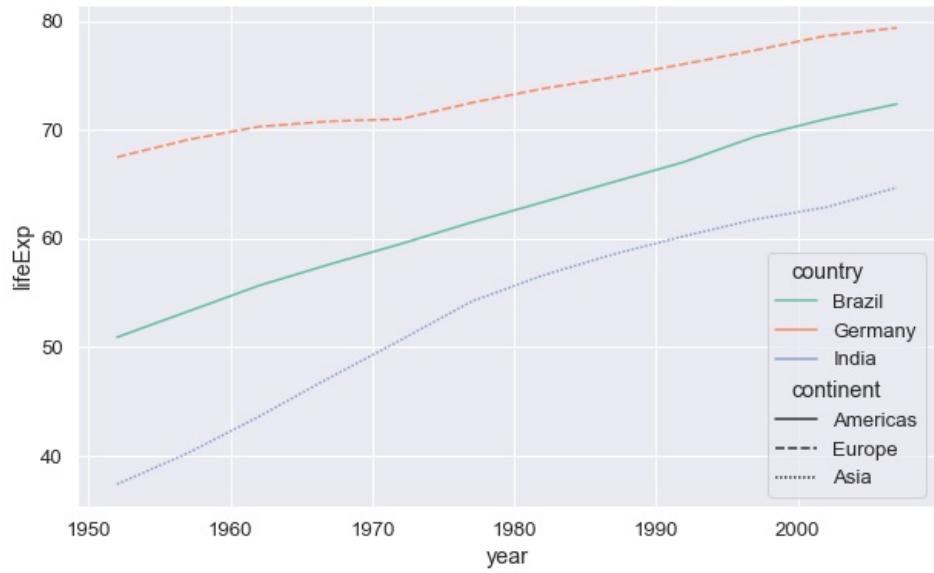
Out[6]:

	country	continent	year	lifeExp	pop	gdpPercap	iso_alpha	iso_num
168	Brazil	Americas	1952	50.917	56602560	2108.944355	BRA	76
169	Brazil	Americas	1957	53.285	65551171	2487.365989	BRA	76
170	Brazil	Americas	1962	55.665	76039390	3336.585802	BRA	76
171	Brazil	Americas	1967	57.632	88049823	3429.864357	BRA	76
172	Brazil	Americas	1972	59.504	100840058	4985.711467	BRA	76
173	Brazil	Americas	1977	61.489	114313951	6660.118654	BRA	76
174	Brazil	Americas	1982	63.336	128962939	7030.835878	BRA	76
175	Brazil	Americas	1987	65.205	142938076	7807.095818	BRA	76
176	Brazil	Americas	1992	67.057	155975974	6950.283021	BRA	76
177	Brazil	Americas	1997	69.388	168546719	7957.980824	BRA	76
178	Brazil	Americas	2002	71.006	179914212	8131.212843	BRA	76
179	Brazil	Americas	2007	72.390	190010647	9065.800825	BRA	76
564	Germany	Europe	1952	67.500	69145952	7144.114393	DEU	276
565	Germany	Europe	1957	69.100	71019069	10187.826650	DEU	276
566	Germany	Europe	1962	70.300	73739117	12902.462910	DEU	276
567	Germany	Europe	1967	70.800	76368453	14745.625610	DEU	276
568	Germany	Europe	1972	71.000	78717088	18016.180270	DEU	276
569	Germany	Europe	1977	72.500	78160773	20512.921230	DEU	276
570	Germany	Europe	1982	73.800	78335266	22031.532740	DEU	276
571	Germany	Europe	1987	74.847	77718298	24639.185660	DEU	276
572	Germany	Europe	1992	76.070	80597764	26505.303170	DEU	276
573	Germany	Europe	1997	77.340	82011073	27788.884160	DEU	276
574	Germany	Europe	2002	78.670	82350671	30035.801980	DEU	276
575	Germany	Europe	2007	79.406	82400996	32170.374420	DEU	276
696	India	Asia	1952	37.373	372000000	546.565749	IND	356
697	India	Asia	1957	40.249	409000000	590.061996	IND	356
698	India	Asia	1962	43.605	454000000	658.347151	IND	356
699	India	Asia	1967	47.193	506000000	700.770611	IND	356
700	India	Asia	1972	50.651	567000000	724.032527	IND	356
701	India	Asia	1977	54.208	634000000	813.337323	IND	356
702	India	Asia	1982	56.596	708000000	855.723538	IND	356
703	India	Asia	1987	58.553	788000000	976.512676	IND	356
704	India	Asia	1992	60.223	872000000	1164.406809	IND	356
705	India	Asia	1997	61.765	959000000	1458.817442	IND	356
706	India	Asia	2002	62.879	1034172547	1746.769454	IND	356
707	India	Asia	2007	64.698	1110396331	2452.210407	IND	356

In [64]:

```
plt.figure(figsize=(10, 6))
sns.lineplot(x='year', y='lifeExp', data=df3, hue='country', style='continent')
```

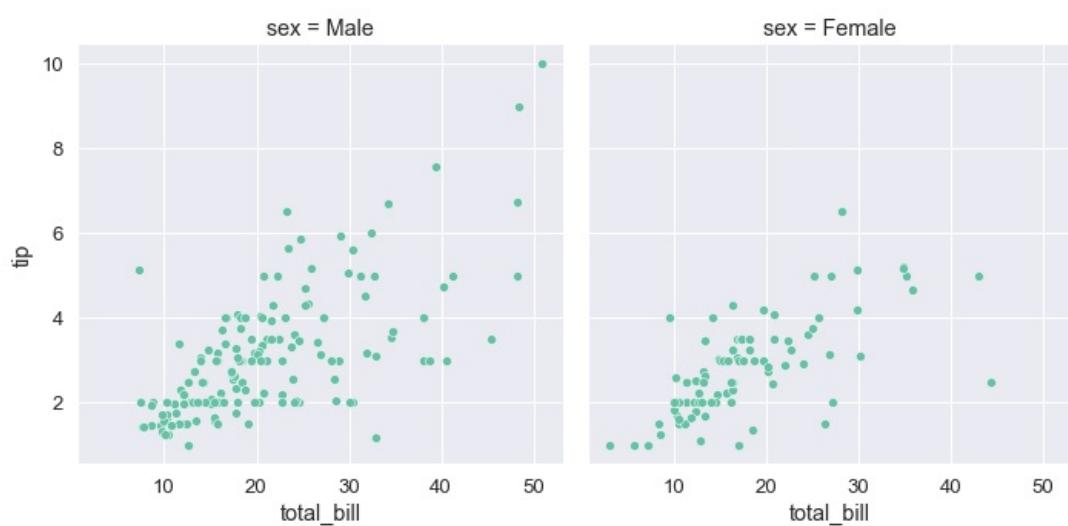
Out[64]: <AxesSubplot:xlabel='year', ylabel='lifeExp'>



RelPlot

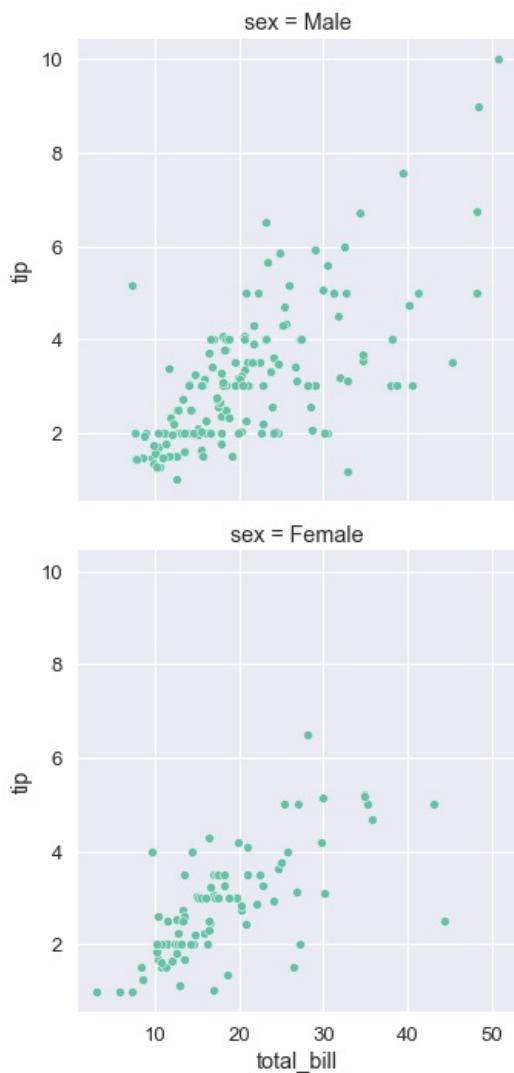
```
In [67]: sns.relplot(x='total_bill', y='tip', data=tips, kind='scatter', col='sex')
```

```
Out[67]: <seaborn.axisgrid.FacetGrid at 0x1c72e33bfd0>
```

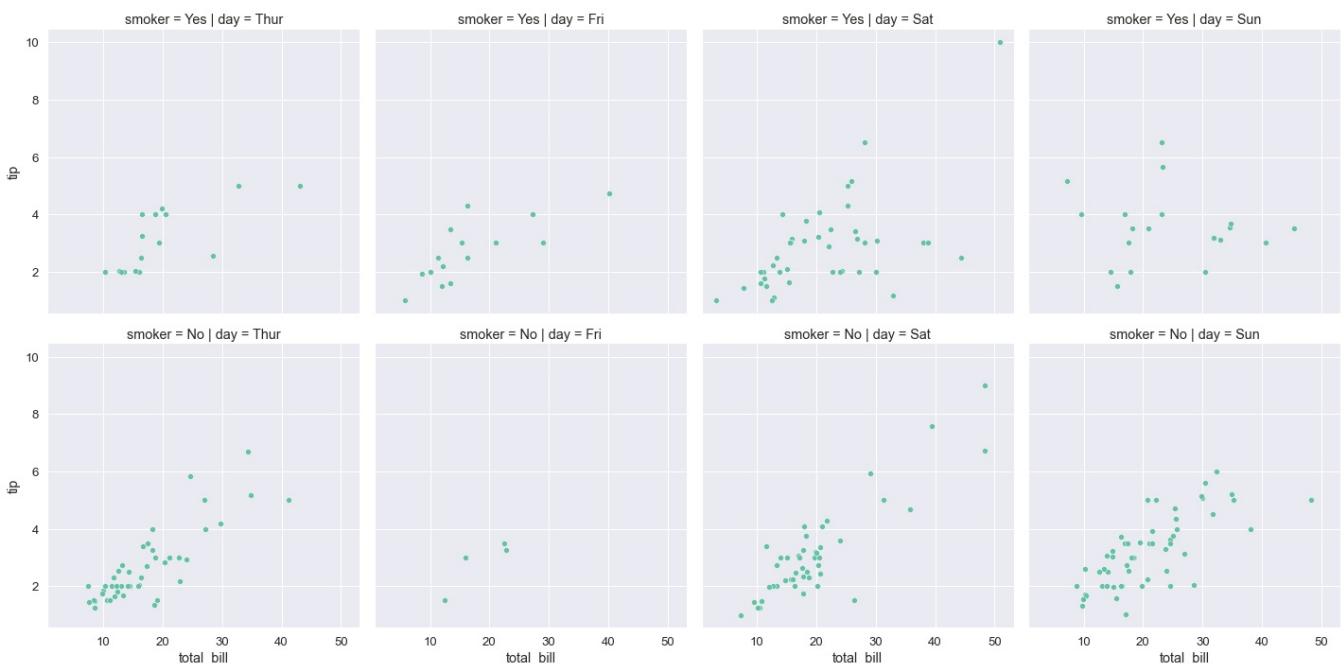


```
In [69]: sns.relplot(x='total_bill', y='tip', data=tips, kind='scatter', row='sex')
```

```
Out[69]: <seaborn.axisgrid.FacetGrid at 0x1c72e85f520>
```

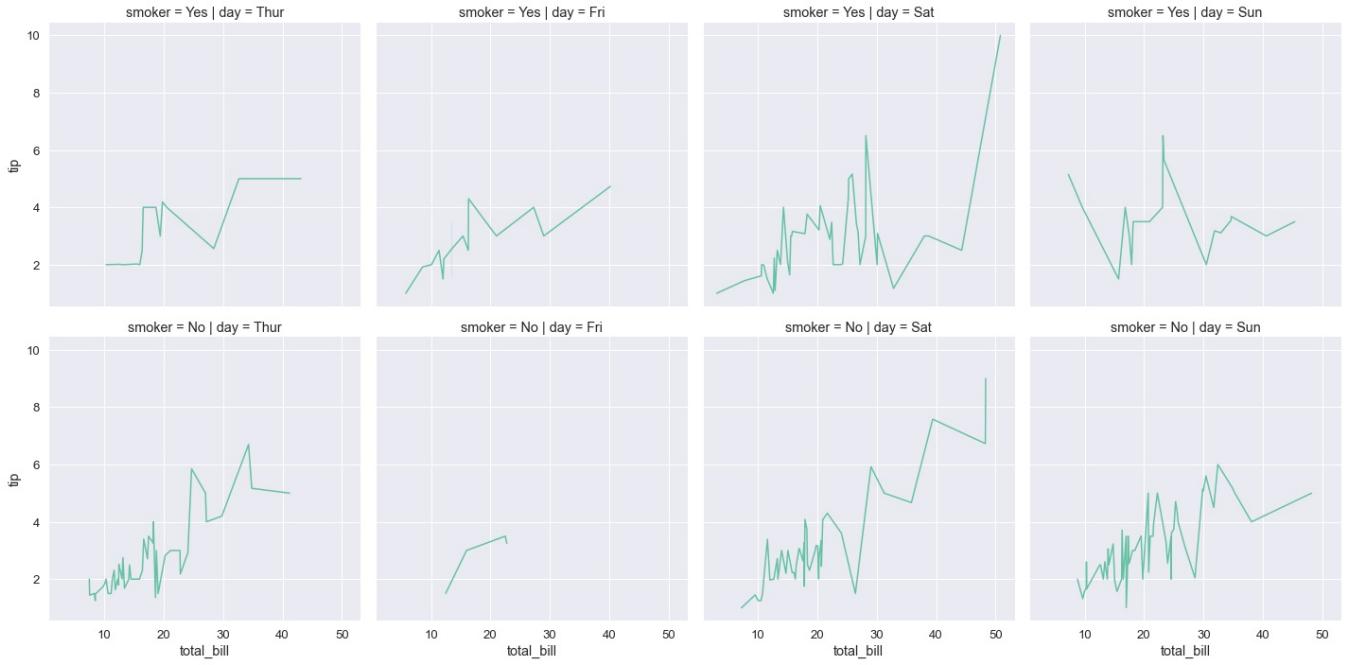


```
In [74]: # its only work with relplot
sns.relplot(x='total_bill', y='tip', data=tips,
             kind='scatter', col='day', row='smoker')
plt.show()
```



```
In [75]: sns.relplot(x='total_bill', y='tip', data=tips,  
                 kind='line', col='day', row='smoker')
```

```
plt.show()
```



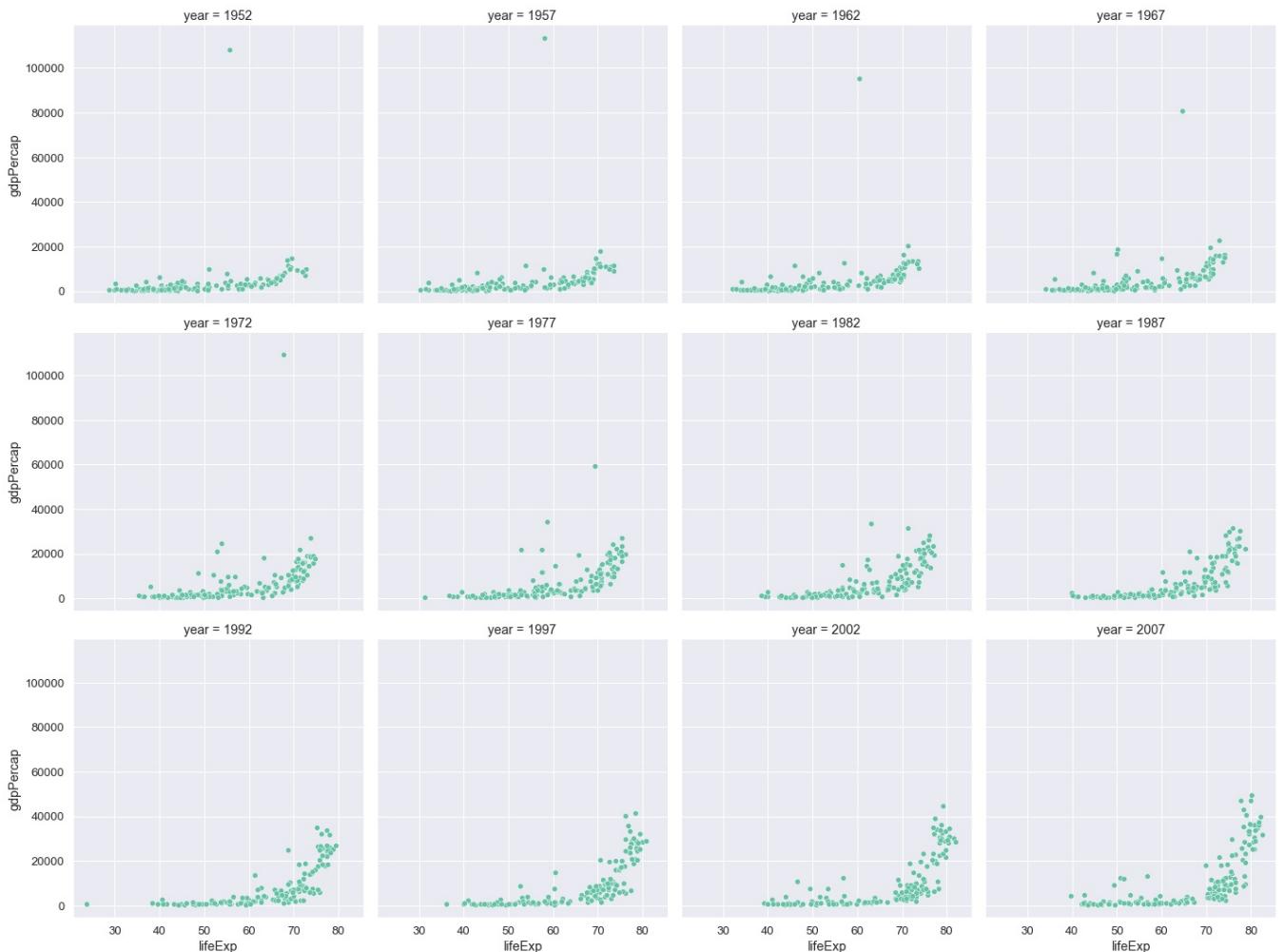
```
In [76]: sns.relplot(data=df, x='lifeExp', y='gdpPercap', kind='scatter', col='year')
```

```
Out[76]: <seaborn.axisgrid.FacetGrid at 0x1c731dd9820>
```



```
In [77]: sns.relplot(data=df, x='lifeExp', y='gdpPercap', kind='scatter', col='year', col_wrap=4)
```

```
Out[77]: <seaborn.axisgrid.FacetGrid at 0x1c732cb3b20>
```



****Distribution Plots****

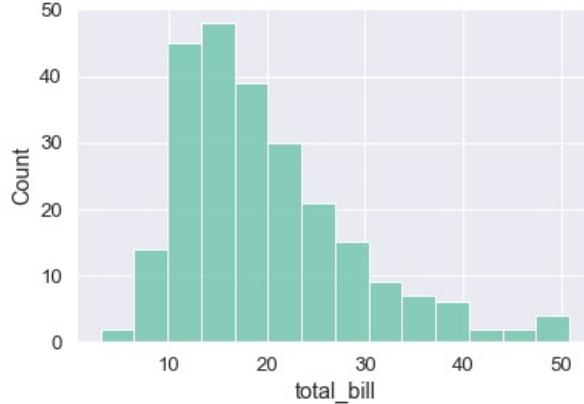
- used for univariate analysis
- used to find out the distribution
- Range of the observation
- Central Tendency
- is the data bimodal?
- Are there outliers?

Plots under distribution plot

- histplot
- kdeplot
- rugplot

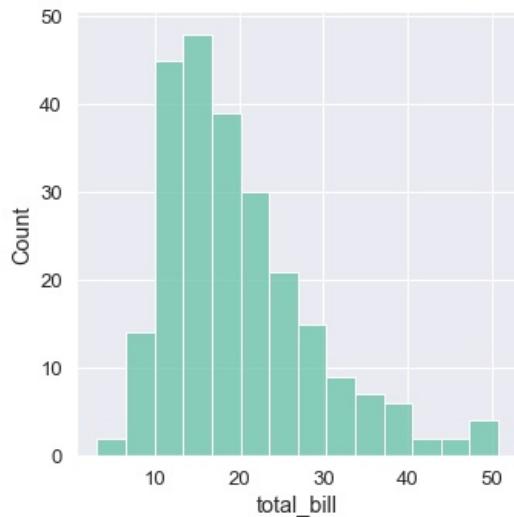
```
In [79]: # hist
sns.histplot(data=tips, x='total_bill')

Out[79]: <AxesSubplot:xlabel='total_bill', ylabel='Count'>
```



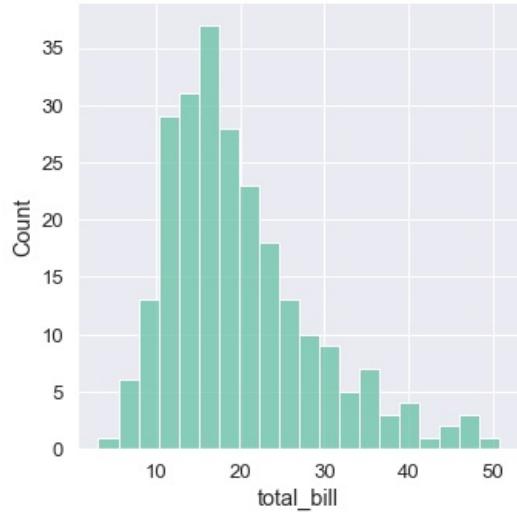
```
In [83]: sns.displot(data=tips, x='total_bill', kind='hist')

Out[83]: <seaborn.axisgrid.FacetGrid at 0x1c734c230a0>
```



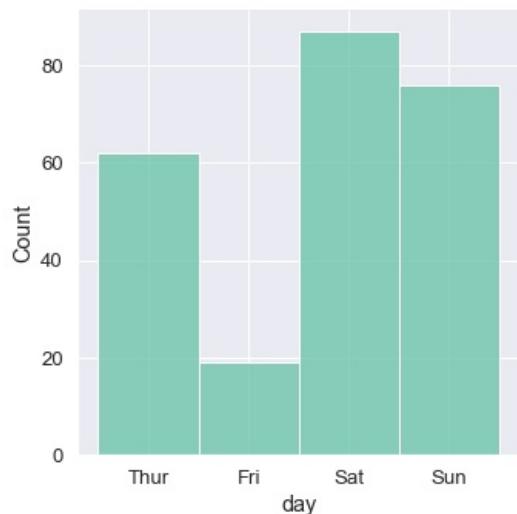
```
In [81]: sns.displot(data=tips, x='total_bill', bins=20)

Out[81]: <seaborn.axisgrid.FacetGrid at 0x1c73532bd90>
```



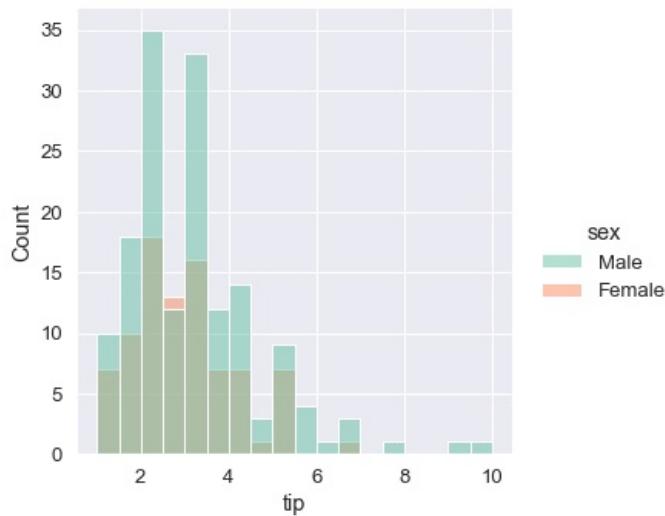
```
In [84]: sns.displot(data=tips, x='day', kind='hist')
```

```
Out[84]: <seaborn.axisgrid.FacetGrid at 0x1c73532b5b0>
```



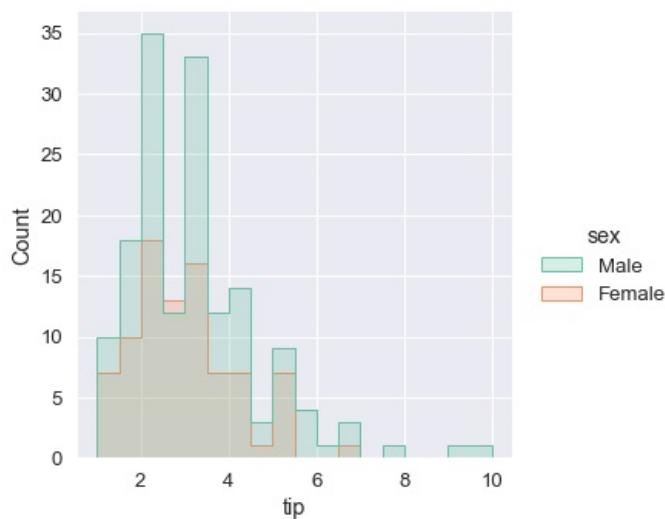
```
In [85]: sns.displot(data=tips, x='tip', kind='hist', hue='sex')
```

```
Out[85]: <seaborn.axisgrid.FacetGrid at 0x1c731e4a9d0>
```



```
In [86]: sns.displot(data=tips, x='tip', kind='hist', hue='sex', element='step')
```

```
Out[86]: <seaborn.axisgrid.FacetGrid at 0x1c7313d5670>
```

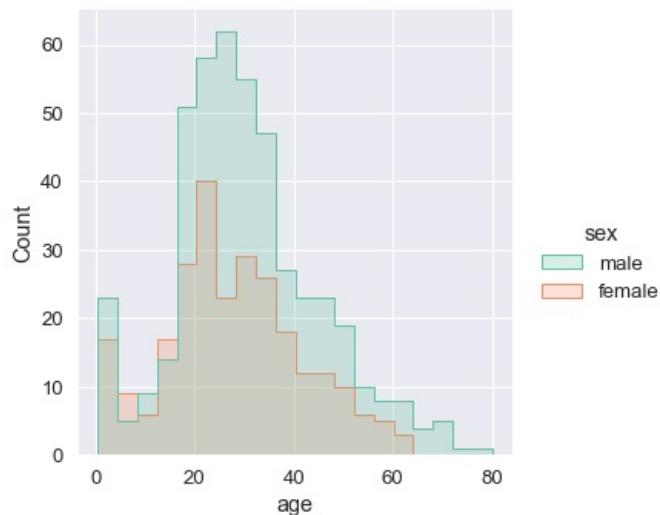


```
In [88]: titanic = sns.load_dataset('titanic')
titanic.head()
```

	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

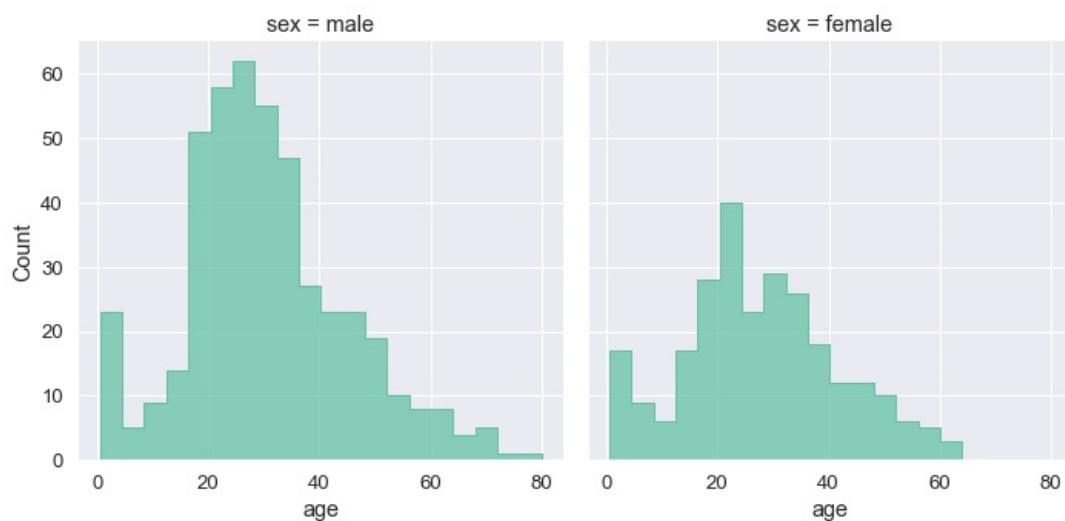
```
In [91]: sns.displot(data=titanic, x='age', kind='hist', hue='sex', element='step')
```

```
Out[91]: <seaborn.axisgrid.FacetGrid at 0x1c73136ef10>
```



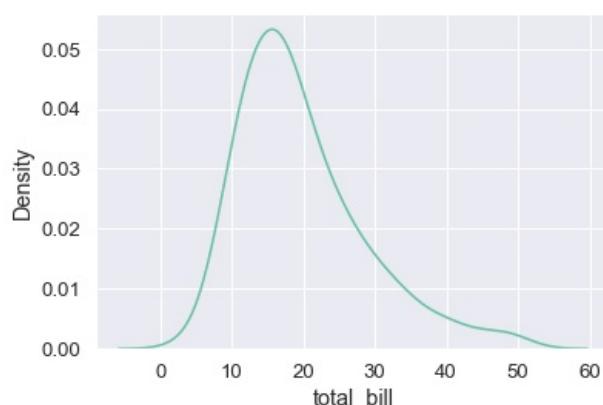
```
In [93]: sns.displot(data=titanic, x='age', kind='hist', col='sex', element='step')
```

```
Out[93]: <seaborn.axisgrid.FacetGrid at 0x1c7313b2850>
```



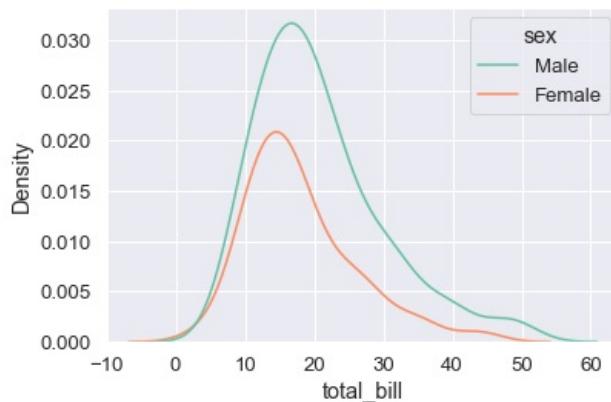
```
In [95]: sns.kdeplot(data=tips, x='total_bill')
```

```
Out[95]: <AxesSubplot:xlabel='total_bill', ylabel='Density'>
```



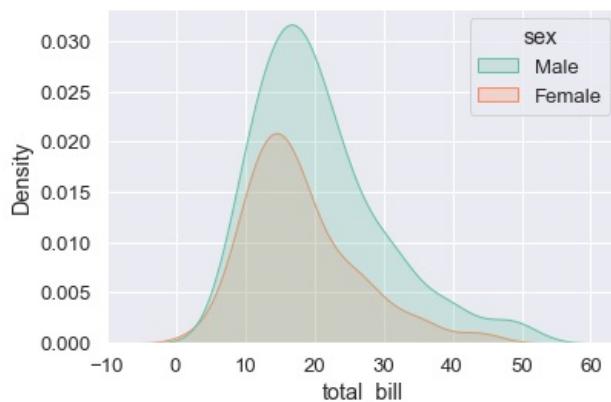
```
In [96]: sns.kdeplot(data=tips, x='total_bill', hue='sex')
```

```
Out[96]: <AxesSubplot:xlabel='total_bill', ylabel='Density'>
```



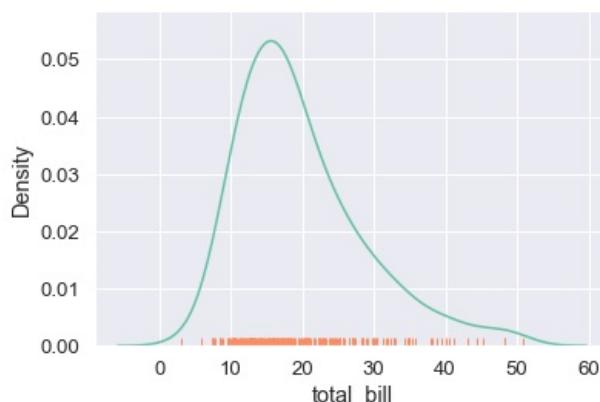
```
In [97]: sns.kdeplot(data=tips, x='total_bill', hue='sex', fill=True)
```

```
Out[97]: <AxesSubplot:xlabel='total_bill', ylabel='Density'>
```



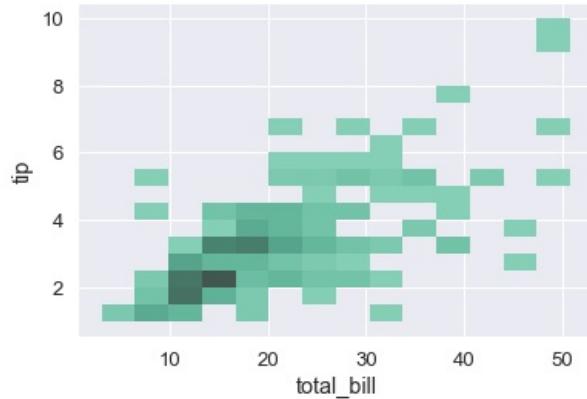
```
In [98]: # rugplot  
sns.kdeplot(data=tips, x='total_bill')  
sns.rugplot(data=tips, x='total_bill')
```

```
Out[98]: <AxesSubplot:xlabel='total_bill', ylabel='Density'>
```



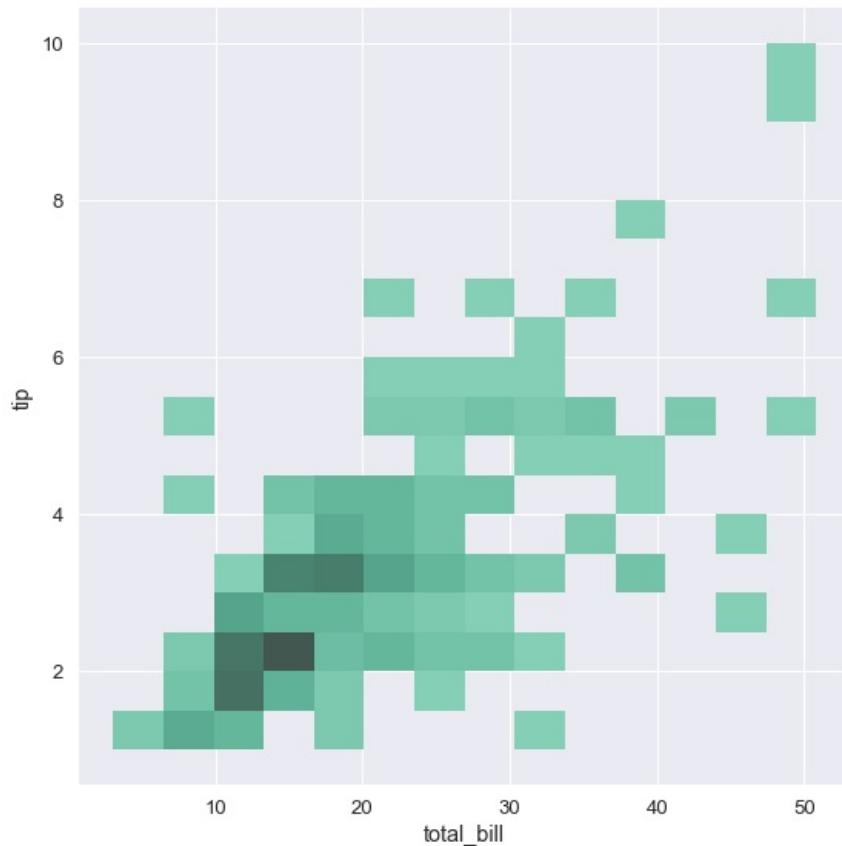
```
In [8]: # bivariate histogram  
sns.histplot(data=tips, x='total_bill', y='tip')
```

```
Out[8]: <AxesSubplot:xlabel='total_bill', ylabel='tip'>
```



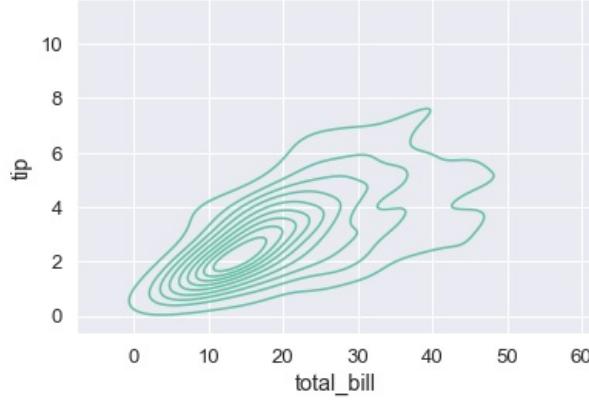
```
In [41]: sns.displot(data=tips, x='total_bill', y='tip', kind='hist', height=8)
```

```
Out[41]: <seaborn.axisgrid.FacetGrid at 0x1feba0ea0d0>
```



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

```
Out[10]: <AxesSubplot:xlabel='total_bill', ylabel='tip'>
```



Metrics

```
In [11]: df4 = df.pivot(index='country', columns='year', values='lifeExp')
df4
```

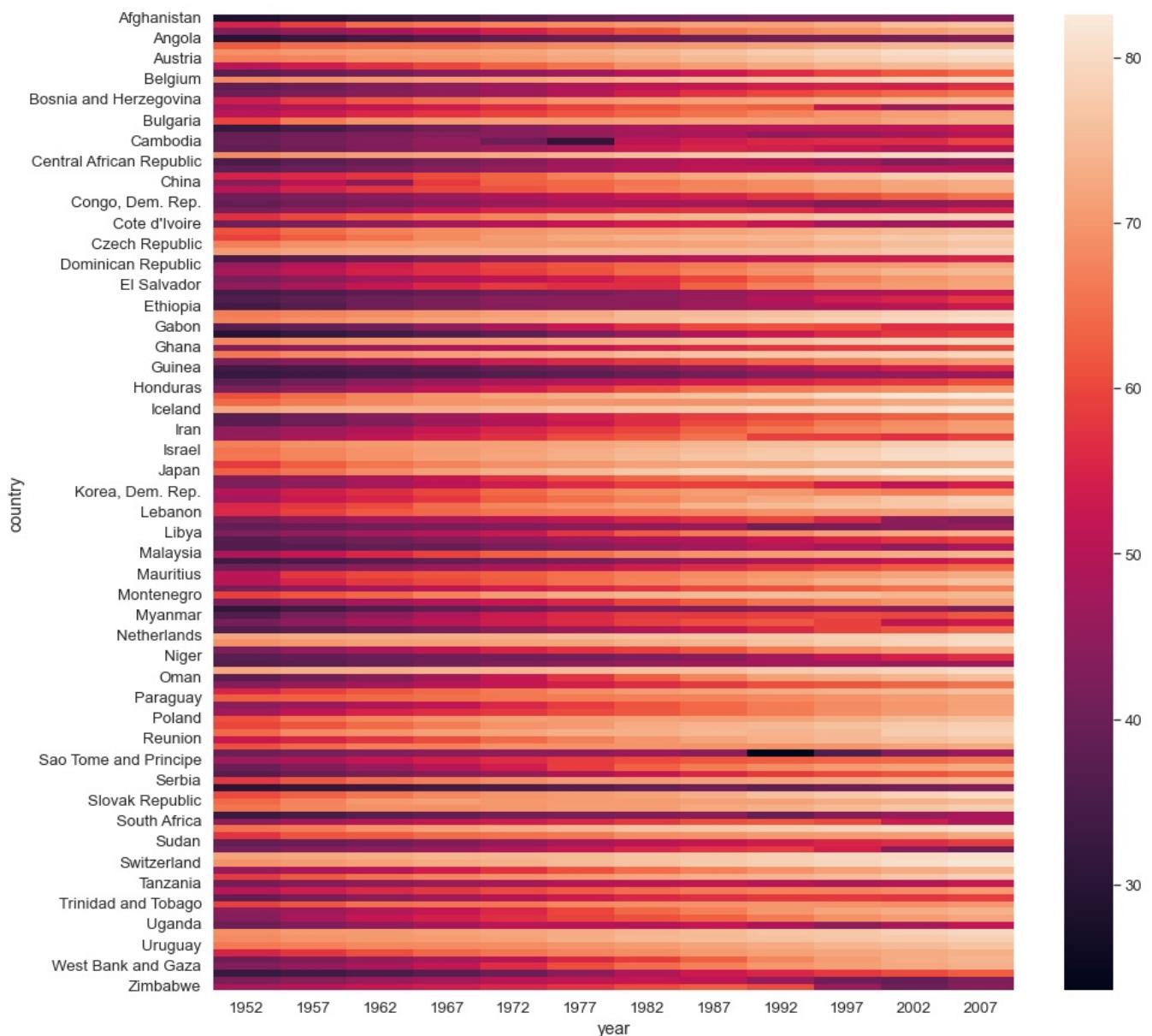
```
Out[11]:
```

	year	1952	1957	1962	1967	1972	1977	1982	1987	1992	1997	2002	2007
country													
Afghanistan	28.801	30.332	31.997	34.020	36.088	38.438	39.854	40.822	41.674	41.763	42.129	43.828	
Albania	55.230	59.280	64.820	66.220	67.690	68.930	70.420	72.000	71.581	72.950	75.651	76.423	
Algeria	43.077	45.685	48.303	51.407	54.518	58.014	61.368	65.799	67.744	69.152	70.994	72.301	
Angola	30.015	31.999	34.000	35.985	37.928	39.483	39.942	39.906	40.647	40.963	41.003	42.731	
Argentina	62.485	64.399	65.142	65.634	67.065	68.481	69.942	70.774	71.868	73.275	74.340	75.320	
...
Vietnam	40.412	42.887	45.363	47.838	50.254	55.764	58.816	62.820	67.662	70.672	73.017	74.249	
West Bank and Gaza	43.160	45.671	48.127	51.631	56.532	60.765	64.406	67.046	69.718	71.096	72.370	73.422	
Yemen, Rep.	32.548	33.970	35.180	36.984	39.848	44.175	49.113	52.922	55.599	58.020	60.308	62.698	
Zambia	42.038	44.077	46.023	47.768	50.107	51.386	51.821	50.821	46.100	40.238	39.193	42.384	
Zimbabwe	48.451	50.469	52.358	53.995	55.635	57.674	60.363	62.351	60.377	46.809	39.989	43.487	

142 rows × 12 columns

```
In [14]: plt.figure(figsize=(15, 15))
sns.heatmap(df4)
```

```
Out[14]: <AxesSubplot:xlabel='year', ylabel='country'>
```



```
In [15]: df5 = df[df['continent'] == 'Europe']
df5.head()
```

```
Out[15]:   country continent  year  lifeExp      pop  gdpPercap  iso_alpha  iso_num
12  Albania    Europe  1952  55.23  1282697  1601.056136     ALB       8
13  Albania    Europe  1957  59.28  1476505  1942.284244     ALB       8
14  Albania    Europe  1962  64.82  1728137  2312.888958     ALB       8
15  Albania    Europe  1967  66.22  1984060  2760.196931     ALB       8
16  Albania    Europe  1972  67.69  2263554  3313.422188     ALB       8
```

```
In [16]: df6 = df5.pivot(index='country', columns='year', values='lifeExp')
df6
```

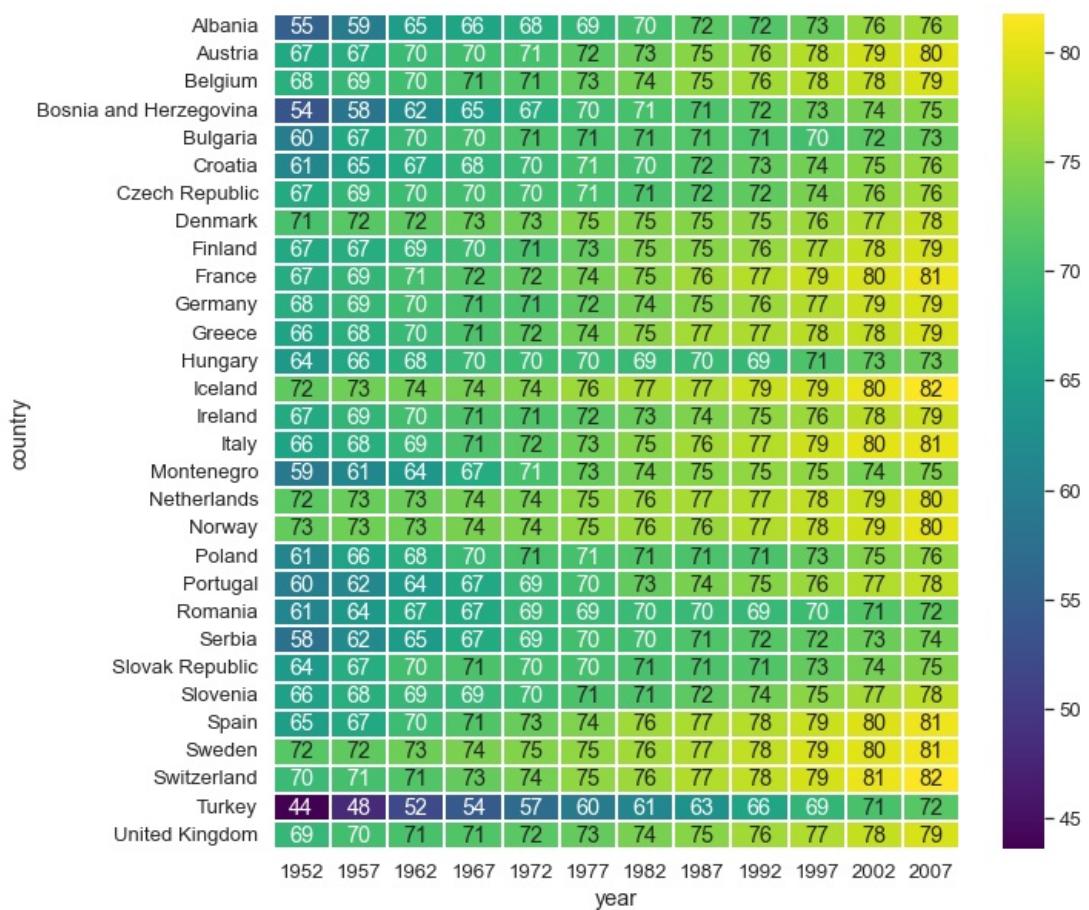
Out[16]:

	year	1952	1957	1962	1967	1972	1977	1982	1987	1992	1997	2002	2007
country													
Albania	55.230	59.280	64.820	66.220	67.690	68.930	70.420	72.000	71.581	72.950	75.651	76.423	
Austria	66.800	67.480	69.540	70.140	70.630	72.170	73.180	74.940	76.040	77.510	78.980	79.829	
Belgium	68.000	69.240	70.250	70.940	71.440	72.800	73.930	75.350	76.460	77.530	78.320	79.441	
Bosnia and Herzegovina	53.820	58.450	61.930	64.790	67.450	69.860	70.690	71.140	72.178	73.244	74.090	74.852	
Bulgaria	59.600	66.610	69.510	70.420	70.900	70.810	71.080	71.340	71.190	70.320	72.140	73.005	
Croatia	61.210	64.770	67.130	68.500	69.610	70.640	70.460	71.520	72.527	73.680	74.876	75.748	
Czech Republic	66.870	69.030	69.900	70.380	70.290	70.710	70.960	71.580	72.400	74.010	75.510	76.486	
Denmark	70.780	71.810	72.350	72.960	73.470	74.690	74.630	74.800	75.330	76.110	77.180	78.332	
Finland	66.550	67.490	68.750	69.830	70.870	72.520	74.550	74.830	75.700	77.130	78.370	79.313	
France	67.410	68.930	70.510	71.550	72.380	73.830	74.890	76.340	77.460	78.640	79.590	80.657	
Germany	67.500	69.100	70.300	70.800	71.000	72.500	73.800	74.847	76.070	77.340	78.670	79.406	
Greece	65.860	67.860	69.510	71.000	72.340	73.680	75.240	76.670	77.030	77.869	78.256	79.483	
Hungary	64.030	66.410	67.960	69.500	69.760	69.950	69.390	69.580	69.170	71.040	72.590	73.338	
Iceland	72.490	73.470	73.680	73.730	74.460	76.110	76.990	77.230	78.770	78.950	80.500	81.757	
Ireland	66.910	68.900	70.290	71.080	71.280	72.030	73.100	74.360	75.467	76.122	77.783	78.885	
Italy	65.940	67.810	69.240	71.060	72.190	73.480	74.980	76.420	77.440	78.820	80.240	80.546	
Montenegro	59.164	61.448	63.728	67.178	70.636	73.066	74.101	74.865	75.435	75.445	73.981	74.543	
Netherlands	72.130	72.990	73.230	73.820	73.750	75.240	76.050	76.830	77.420	78.030	78.530	79.762	
Norway	72.670	73.440	73.470	74.080	74.340	75.370	75.970	75.890	77.320	78.320	79.050	80.196	
Poland	61.310	65.770	67.640	69.610	70.850	70.670	71.320	70.980	70.990	72.750	74.670	75.563	
Portugal	59.820	61.510	64.390	66.600	69.260	70.410	72.770	74.060	74.860	75.970	77.290	78.098	
Romania	61.050	64.100	66.800	66.800	69.210	69.460	69.660	69.530	69.360	69.720	71.322	72.476	
Serbia	57.996	61.685	64.531	66.914	68.700	70.300	70.162	71.218	71.659	72.232	73.213	74.002	
Slovak Republic	64.360	67.450	70.330	70.980	70.350	70.450	70.800	71.080	71.380	72.710	73.800	74.663	
Slovenia	65.570	67.850	69.150	69.180	69.820	70.970	71.063	72.250	73.640	75.130	76.660	77.926	
Spain	64.940	66.660	69.690	71.440	73.060	74.390	76.300	76.900	77.570	78.770	79.780	80.941	
Sweden	71.860	72.490	73.370	74.160	74.720	75.440	76.420	77.190	78.160	79.390	80.040	80.884	
Switzerland	69.620	70.560	71.320	72.770	73.780	75.390	76.210	77.410	78.030	79.370	80.620	81.701	
Turkey	43.585	48.079	52.098	54.336	57.005	59.507	61.036	63.108	66.146	68.835	70.845	71.777	
United Kingdom	69.180	70.420	70.760	71.360	72.010	72.760	74.040	75.007	76.420	77.218	78.471	79.425	

In [27]:

```
plt.figure(figsize=(10, 10))
sns.set_theme(style='darkgrid', palette='Set2', font_scale=1.2)
sns.heatmap(df6, annot=True, linewidths=0.1, cmap='viridis')
```

Out[27]: <AxesSubplot:xlabel='year', ylabel='country'>

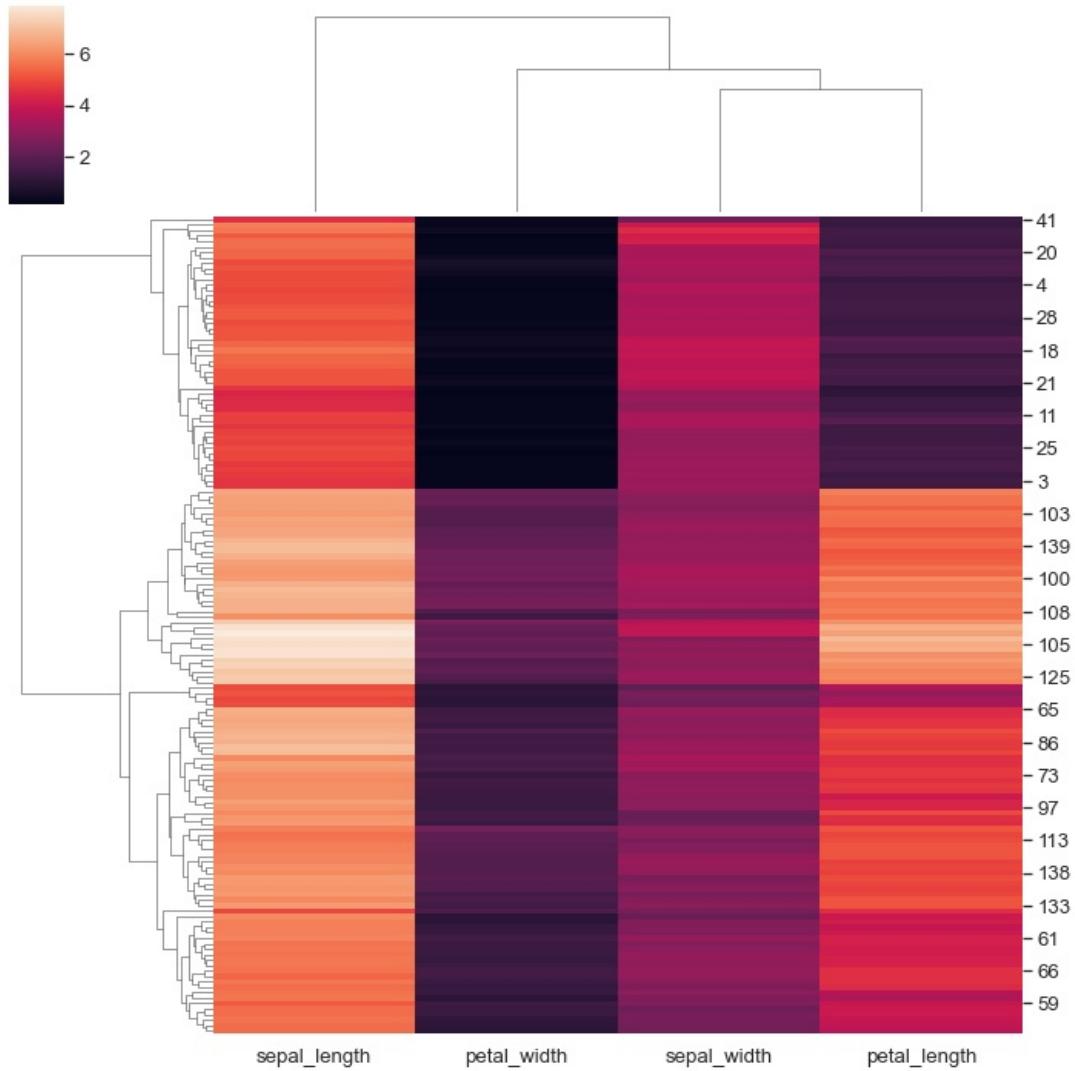


```
In [29]: iris = sns.load_dataset('iris')
iris.head()
```

```
Out[29]:   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 [34]: sns.clustermap(iris.iloc[:, [0, 1, 2, 3]])
<seaborn.matrix.ClusterGrid at 0x1feb77c6520>
```

```
Out[34]:
```

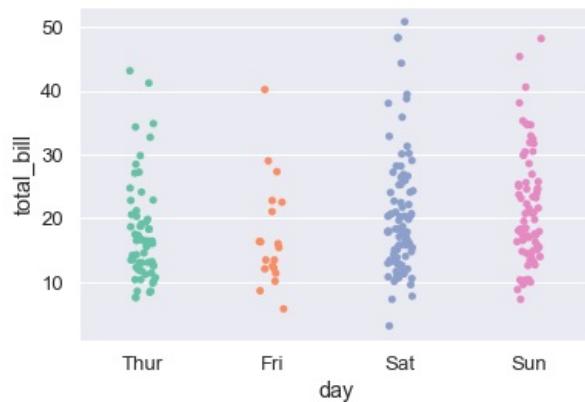


Categorical plot

Strip plot

```
In [35]: # scatterplot in category
sns.stripplot(data=tips, x='day', y='total_bill')
```

```
Out[35]: <AxesSubplot:xlabel='day', ylabel='total_bill'>
```



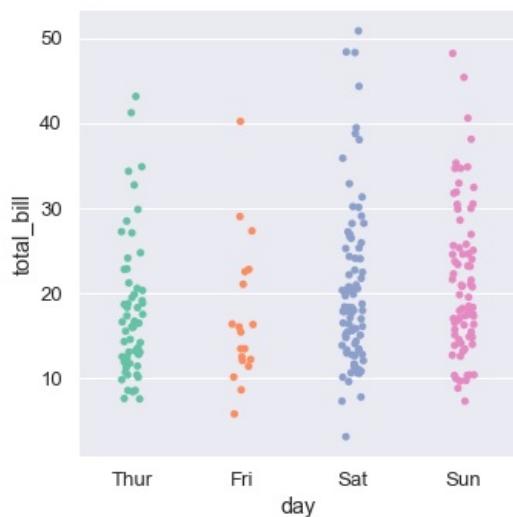
```
In [44]: sns.stripplot(data=tips, x='day', y='total_bill', jitter=False)
# without noise
```

```
Out[44]: <AxesSubplot:xlabel='day', ylabel='total_bill'>
```



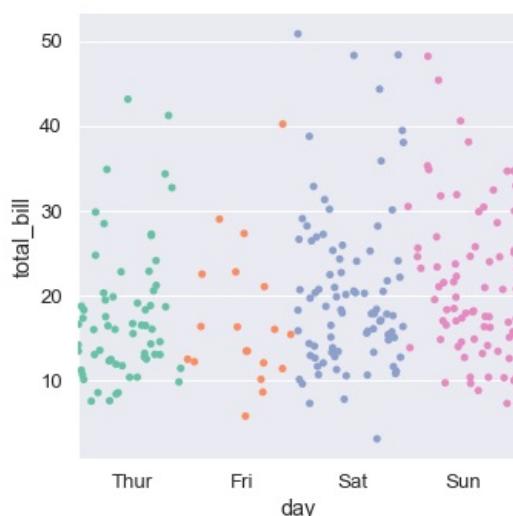
```
In [45]: # figure level
sns.catplot(data=tips, x='day', y='total_bill', kind='strip')
```

```
Out[45]: <seaborn.axisgrid.FacetGrid at 0x1febb2f5910>
```



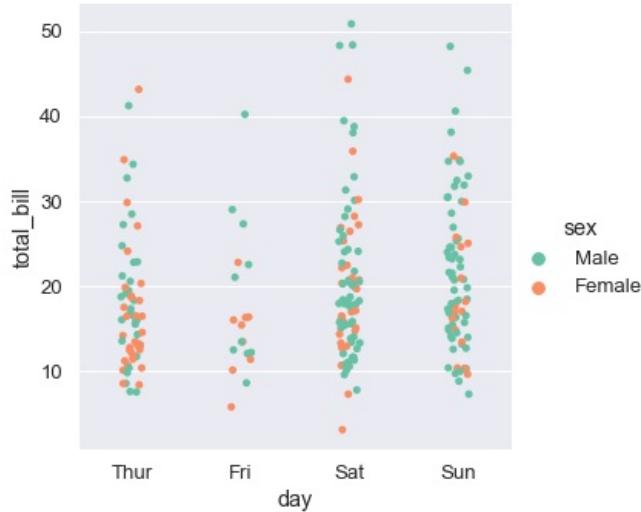
```
In [46]: sns.catplot(data=tips, x='day', y='total_bill', kind='strip', jitter=0.5)
```

```
Out[46]: <seaborn.axisgrid.FacetGrid at 0x1febbae729a0>
```



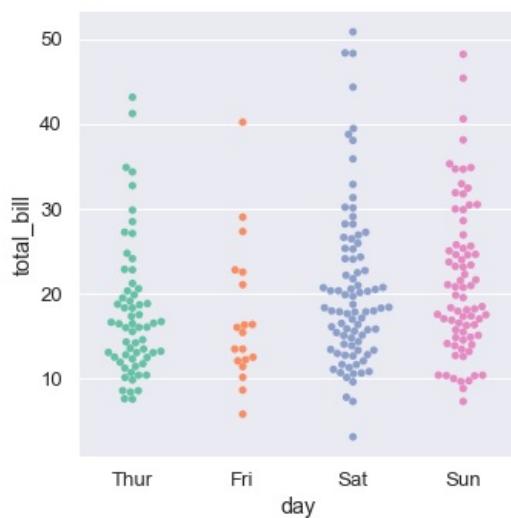
```
In [47]: sns.catplot(data=tips, x='day', y='total_bill', kind='strip', hue='sex')
```

```
Out[47]: <seaborn.axisgrid.FacetGrid at 0x1febb43a8b0>
```



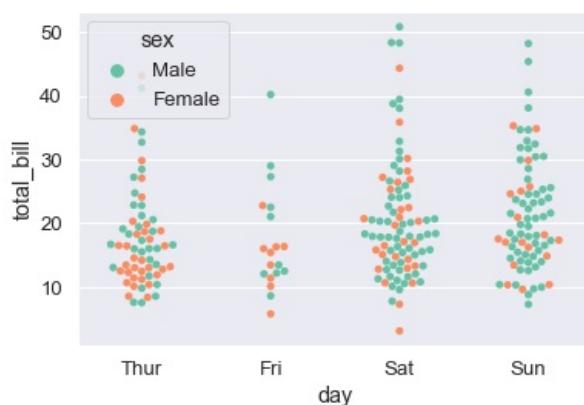
```
In [48]: sns.catplot(data=tips, x='day', y='total_bill', kind='swarm')
```

```
Out[48]: <seaborn.axisgrid.FacetGrid at 0x1febb502100>
```



```
In [50]: sns.swarmplot(data=tips, x='day', y='total_bill', hue='sex')
```

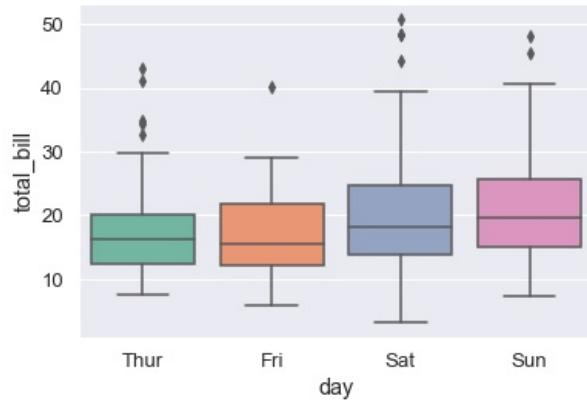
```
Out[50]: <AxesSubplot:xlabel='day', ylabel='total_bill'>
```



box plot

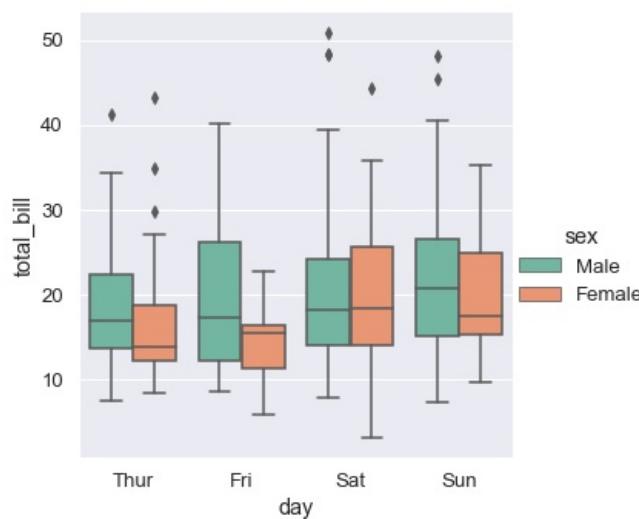
```
In [52]: sns.boxplot(data=tips, x='day', y='total_bill')
```

```
Out[52]: <AxesSubplot:xlabel='day', ylabel='total_bill'>
```



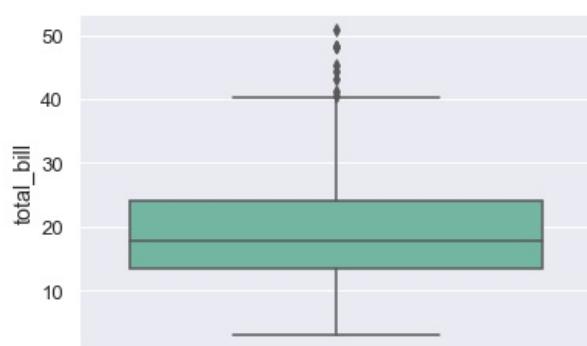
```
In [54]: sns.catplot(data=tips, x='day', y='total_bill', kind='box', hue='sex')
```

```
Out[54]: <seaborn.axisgrid.FacetGrid at 0x1feba8a8580>
```



```
In [55]: sns.boxplot(data=tips, y='total_bill')
```

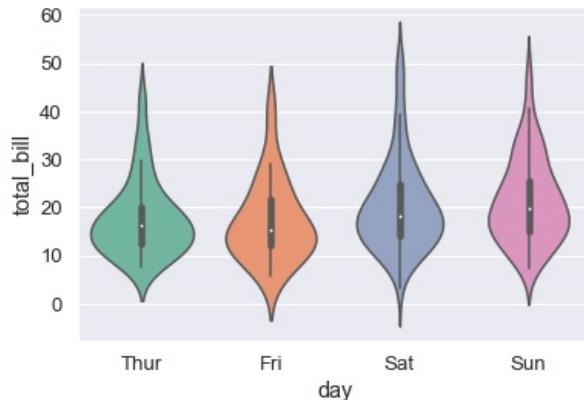
```
Out[55]: <AxesSubplot:ylabel='total_bill'>
```



Violin plot

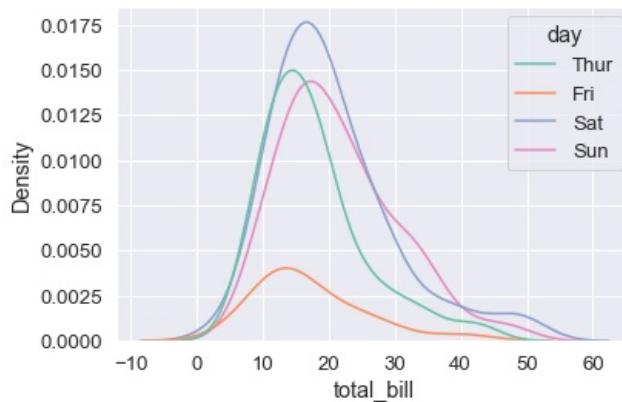
```
In [56]: sns.violinplot(data=tips, x='day', y='total_bill')
```

```
Out[56]: <AxesSubplot:xlabel='day', ylabel='total_bill'>
```



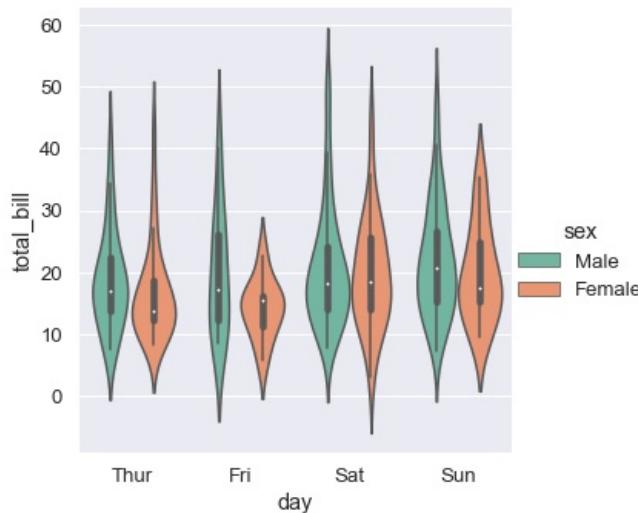
```
In [58]: sns.kdeplot(data=tips, x='total_bill', hue='day')
```

```
Out[58]: <AxesSubplot:xlabel='total_bill', ylabel='Density'>
```



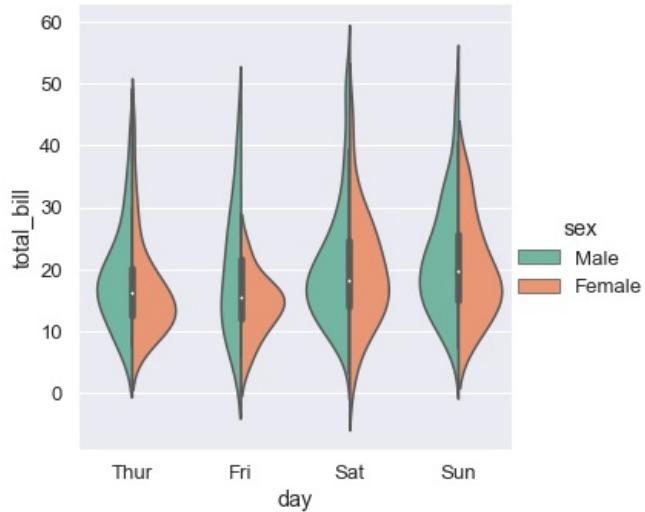
```
In [59]: sns.catplot(data=tips, x='day', y='total_bill', kind='violin', hue='sex')
```

```
Out[59]: <seaborn.axisgrid.FacetGrid at 0x1febc885b50>
```



```
In [60]: sns.catplot(data=tips, x='day', y='total_bill', kind='violin', hue='sex', split=True)
```

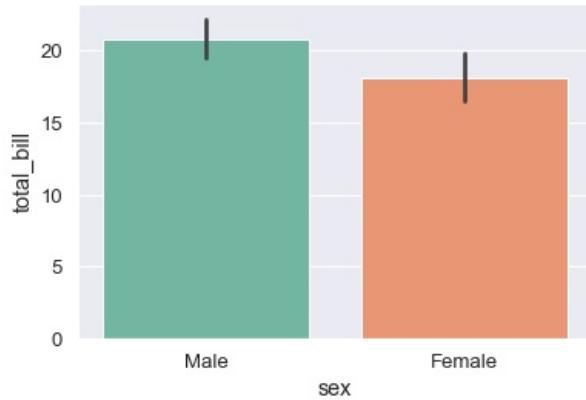
```
Out[60]: <seaborn.axisgrid.FacetGrid at 0x1fec11bdbe0>
```



Bar plot

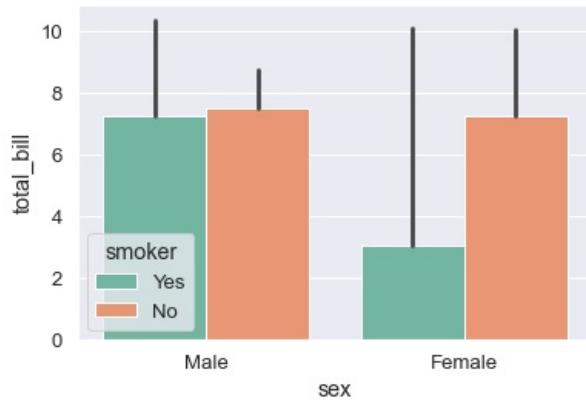
```
In [61]: sns.barplot(data=tips, x='sex', y='total_bill')
# avg total bill
```

```
Out[61]: <AxesSubplot:xlabel='sex', ylabel='total_bill'>
```



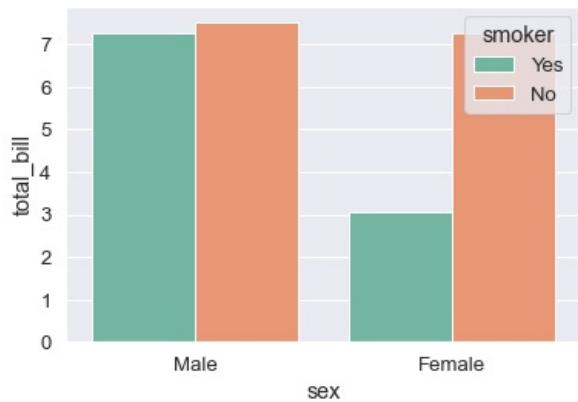
```
In [63]: import numpy as np
sns.barplot(data=tips, x='sex', y='total_bill', hue='smoker', estimator=np.min)
```

```
Out[63]: <AxesSubplot:xlabel='sex', ylabel='total_bill'>
```



```
In [71]: sns.barplot(data=tips, x='sex', y='total_bill',
hue='smoker', estimator=np.min, ci=None)
```

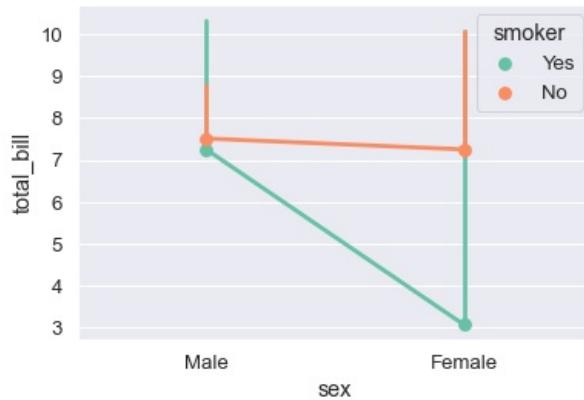
```
Out[71]: <AxesSubplot:xlabel='sex', ylabel='total_bill'>
```



Point plot

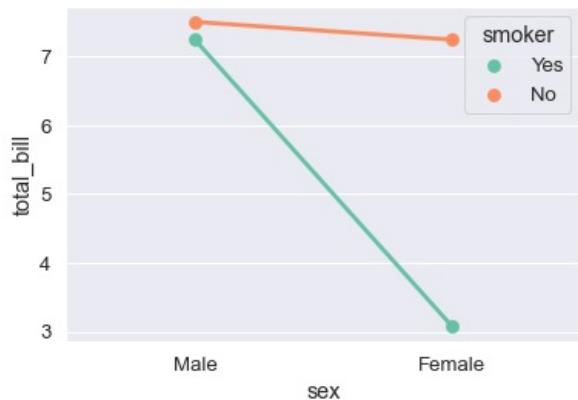
```
In [64]: sns.pointplot(data=tips, x='sex', y='total_bill',
                     hue='smoker', estimator=np.min)
```

```
Out[64]: <AxesSubplot:xlabel='sex', ylabel='total_bill'>
```



```
In [72]: sns.pointplot(data=tips, x='sex', y='total_bill',
                     hue='smoker', estimator=np.min, ci=None)
```

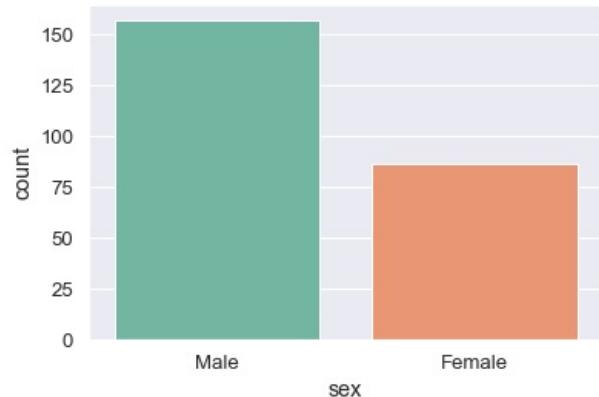
```
Out[72]: <AxesSubplot:xlabel='sex', ylabel='total_bill'>
```



Count plot

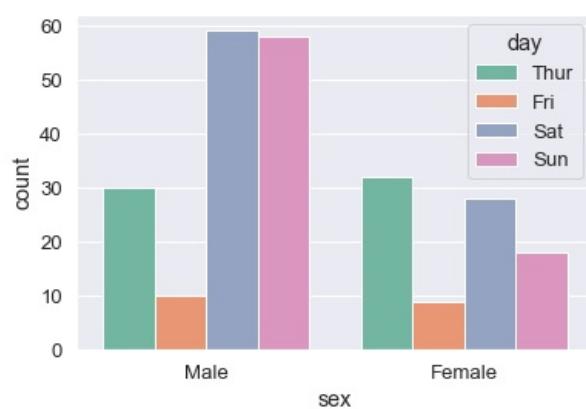
```
In [65]: sns.countplot(data=tips, x='sex')
```

```
Out[65]: <AxesSubplot:xlabel='sex', ylabel='count'>
```



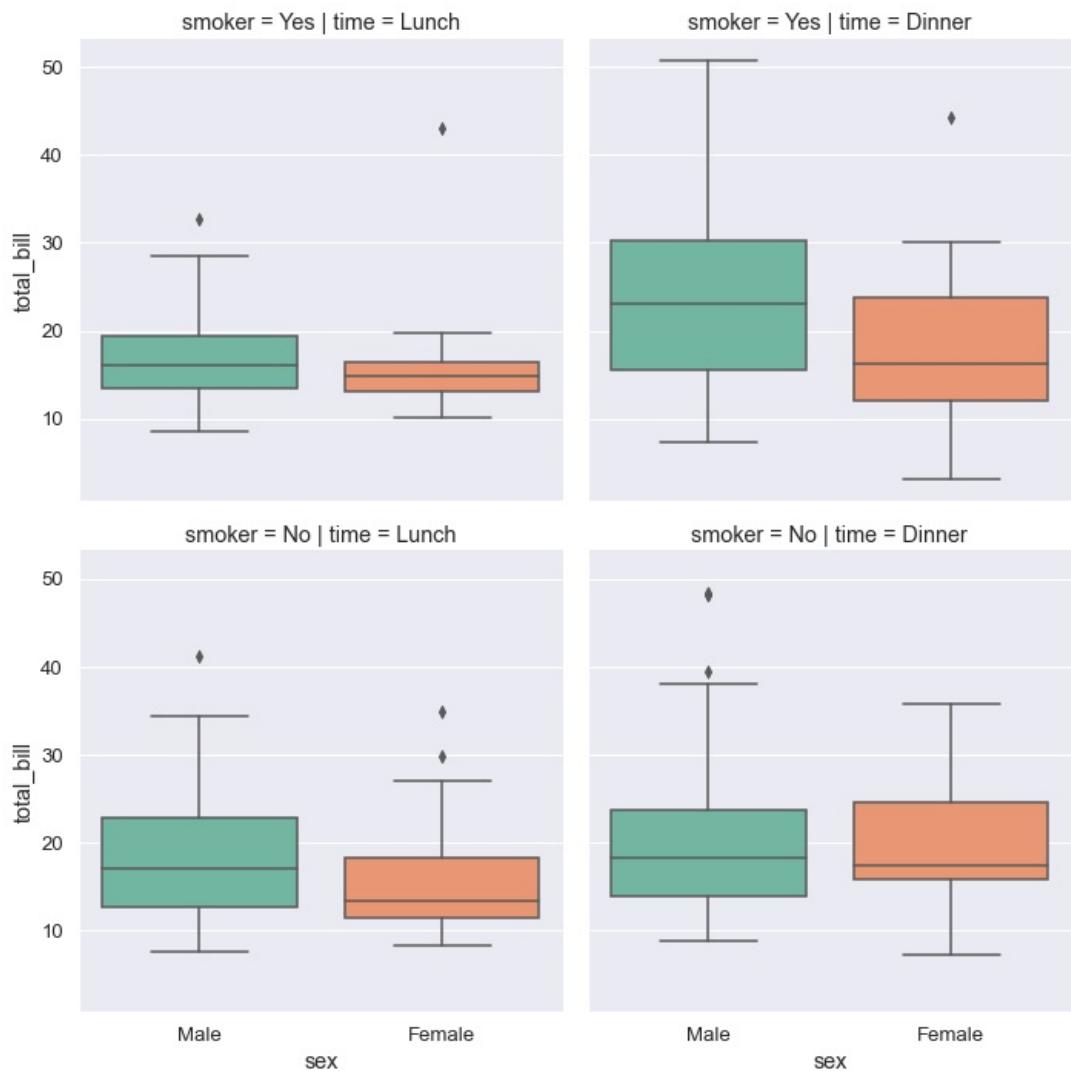
```
In [66]: sns.countplot(data=tips, x='sex', hue='day')
```

```
Out[66]: <AxesSubplot:xlabel='sex', ylabel='count'>
```



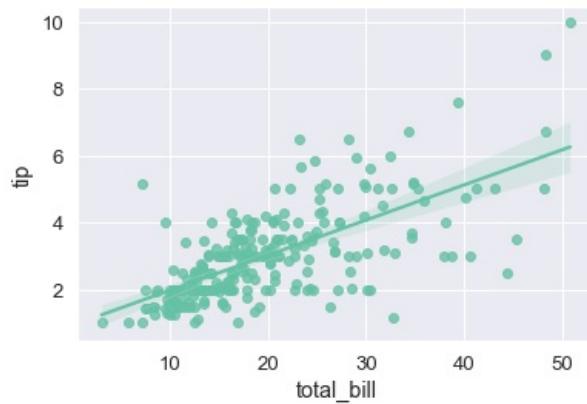
```
In [70]: sns.catplot(data=tips, x='sex', y='total_bill',  
                   col='time', row='smoker', kind='box')
```

```
Out[70]: <seaborn.axisgrid.FacetGrid at 0x1fec15a8130>
```



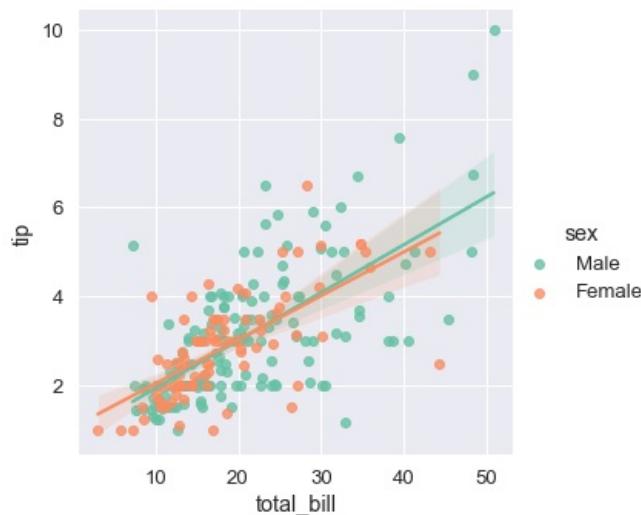
```
In [73]: sns.regplot(data=tips, x='total_bill', y='tip')
```

```
Out[73]: <AxesSubplot:xlabel='total_bill', ylabel='tip'>
```



```
In [74]: sns.lmplot(data=tips, x='total_bill', y='tip', hue='sex')
```

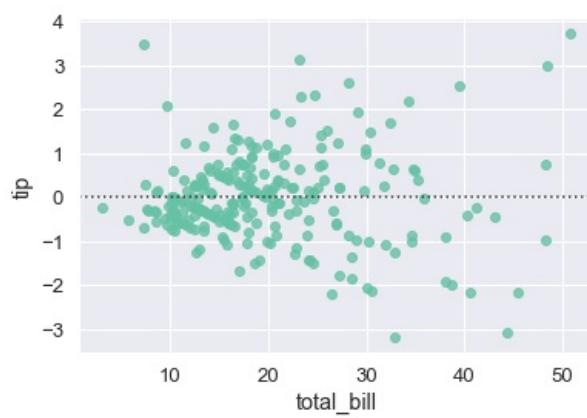
```
Out[74]: <seaborn.axisgrid.FacetGrid at 0x1fec2e289a0>
```



Resid plot

```
In [75]: sns.residplot(data=tips, x='total_bill', y='tip')
```

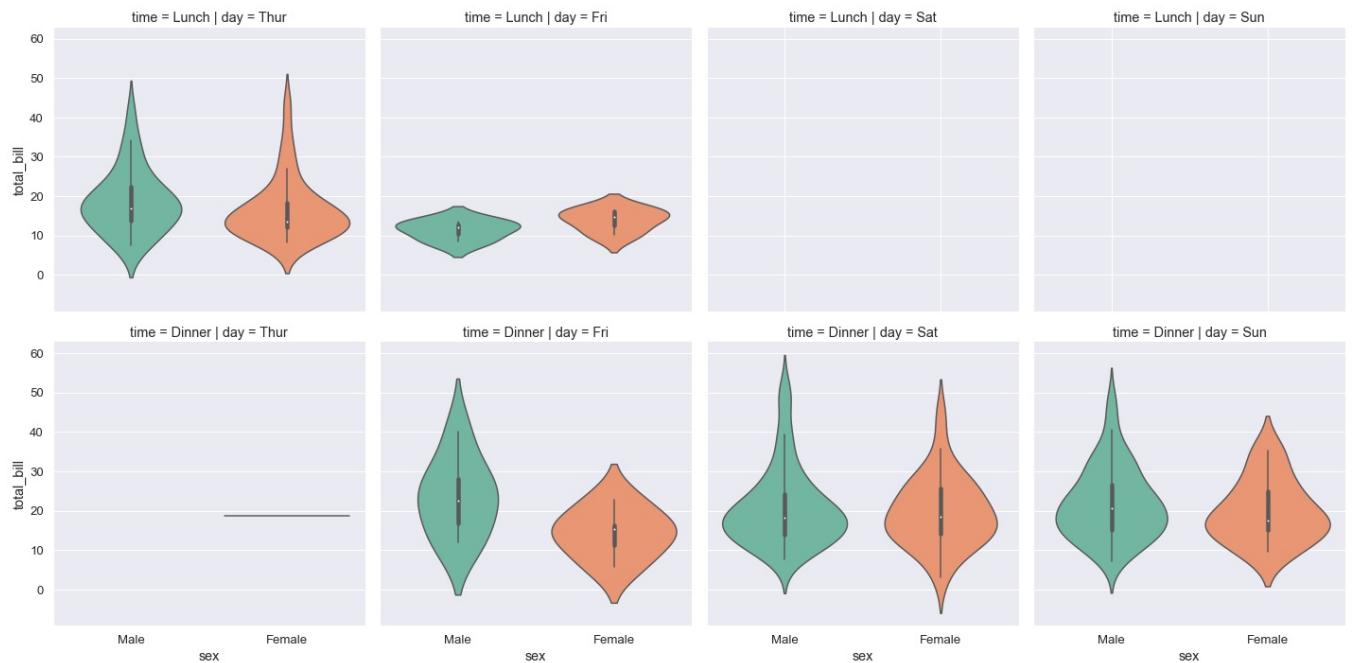
```
Out[75]: <AxesSubplot:xlabel='total_bill', ylabel='tip'>
```



FacitGrid

```
In [77]: sns.catplot(data=tips, x='sex', y='total_bill',
                 kind='violin', col='day', row='time')
```

```
Out[77]: <seaborn.axisgrid.FacetGrid at 0x1fec1b56d60>
```



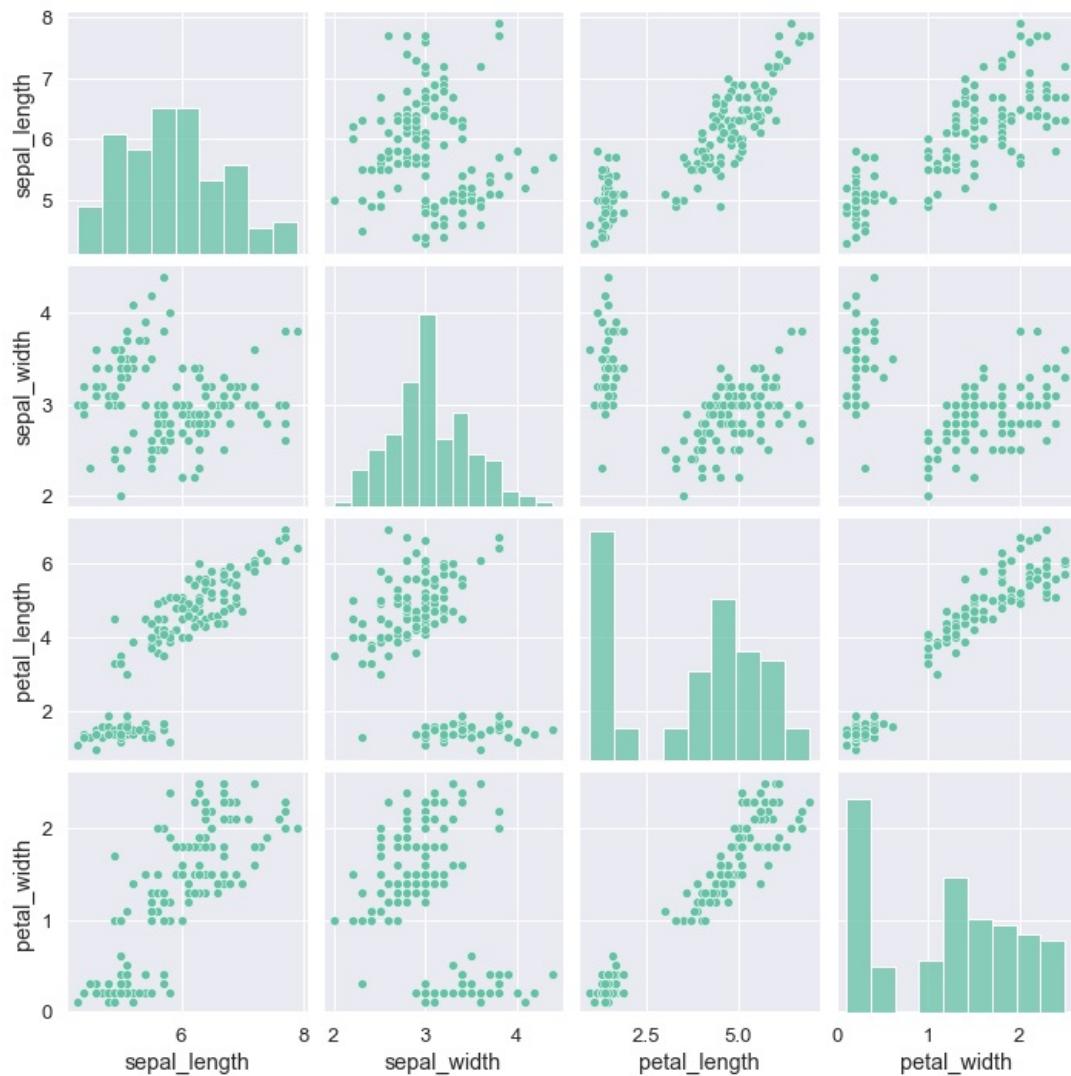
```
In [84]: fig = sns.FacetGrid(data=tips, col='day', row='time', hue='smoker')
fig.map(sns.scatterplot, 'tip', 'total_bill')
fig.add_legend()
```

```
Out[84]: <seaborn.axisgrid.FacetGrid at 0x1fec6ed6400>
```



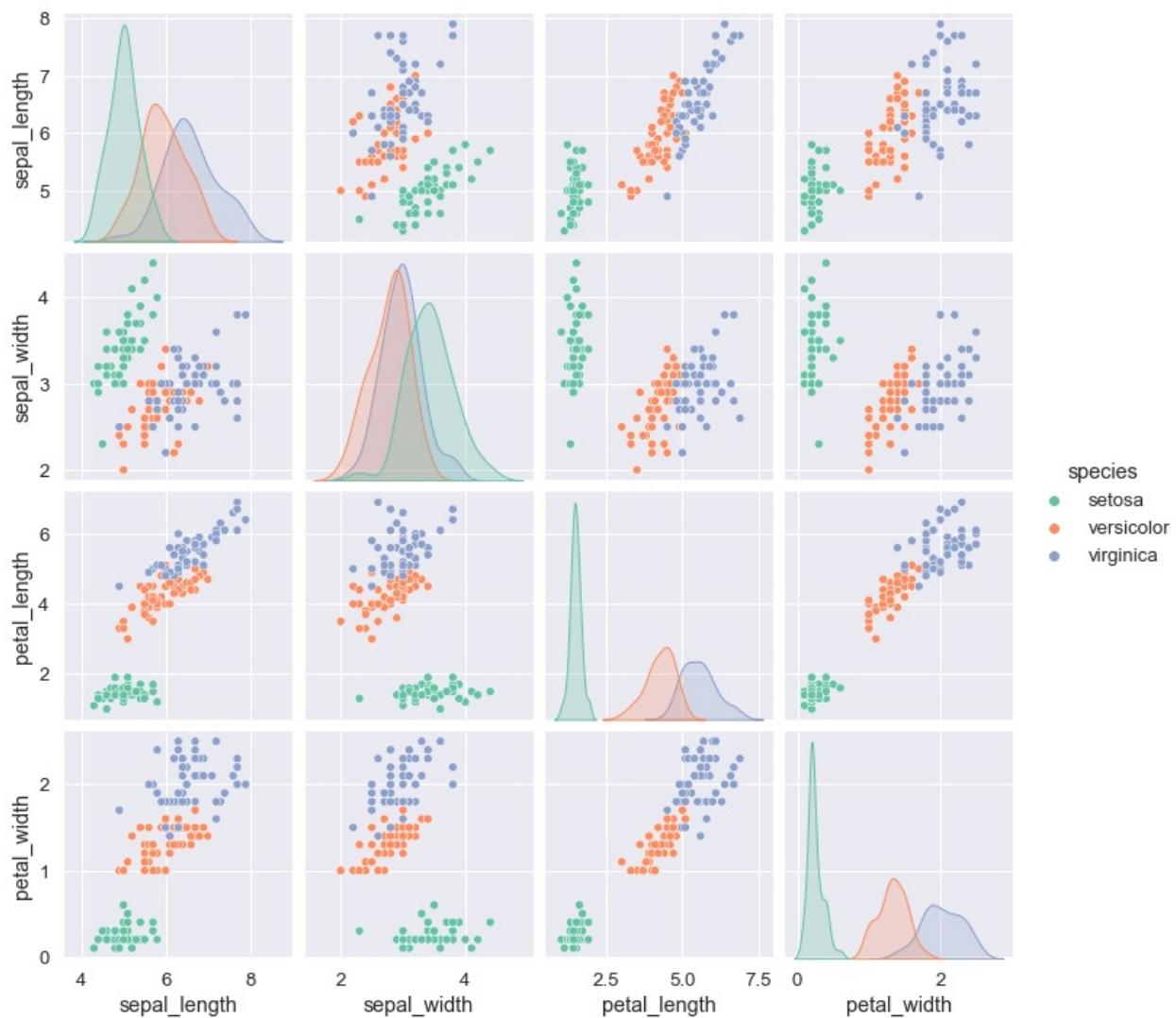
```
In [85]: sns.pairplot(iris)
```

```
Out[85]: <seaborn.axisgrid.PairGrid at 0x1fec3b77e80>
```



In [86]: `sns.pairplot(iris, hue='species')`

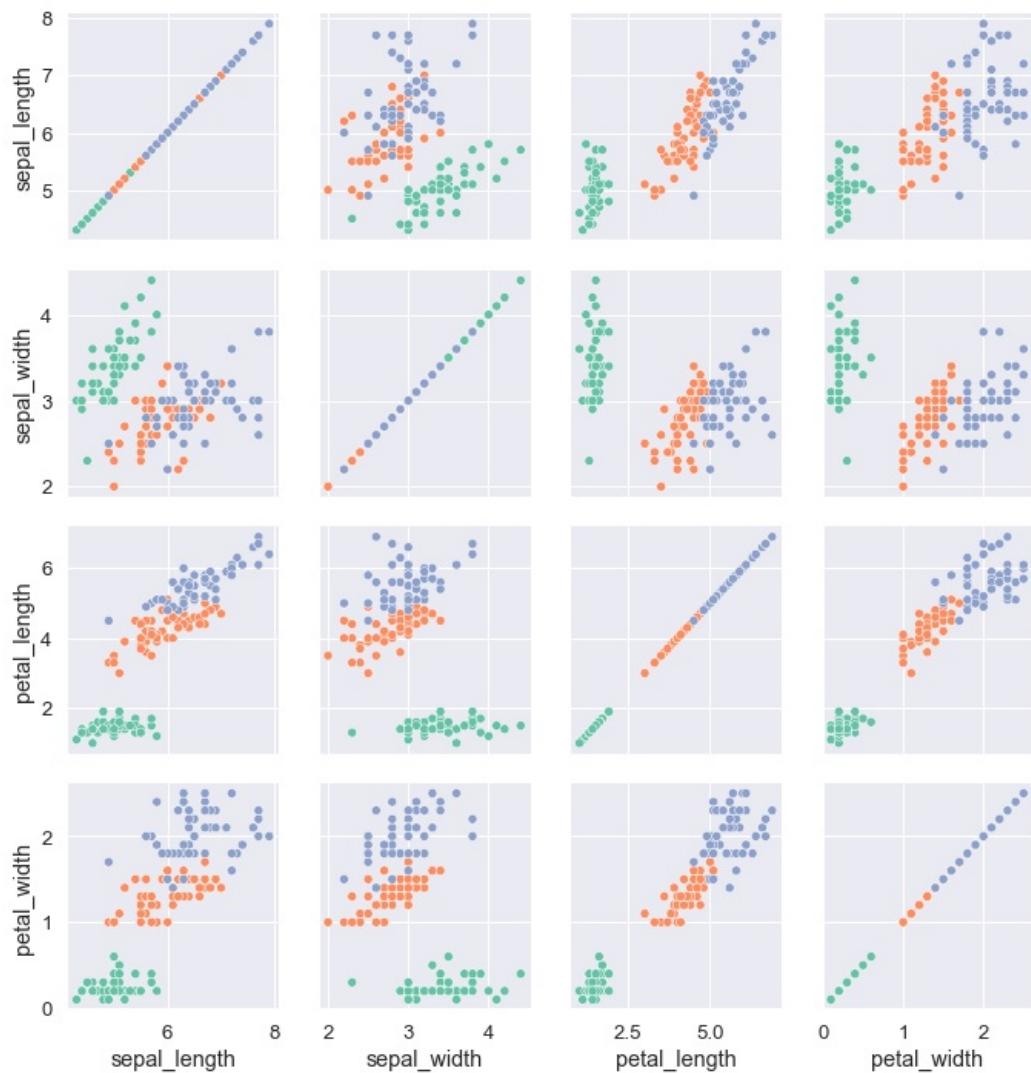
Out[86]: <seaborn.axisgrid.PairGrid at 0x1fec49a4610>



pair grid

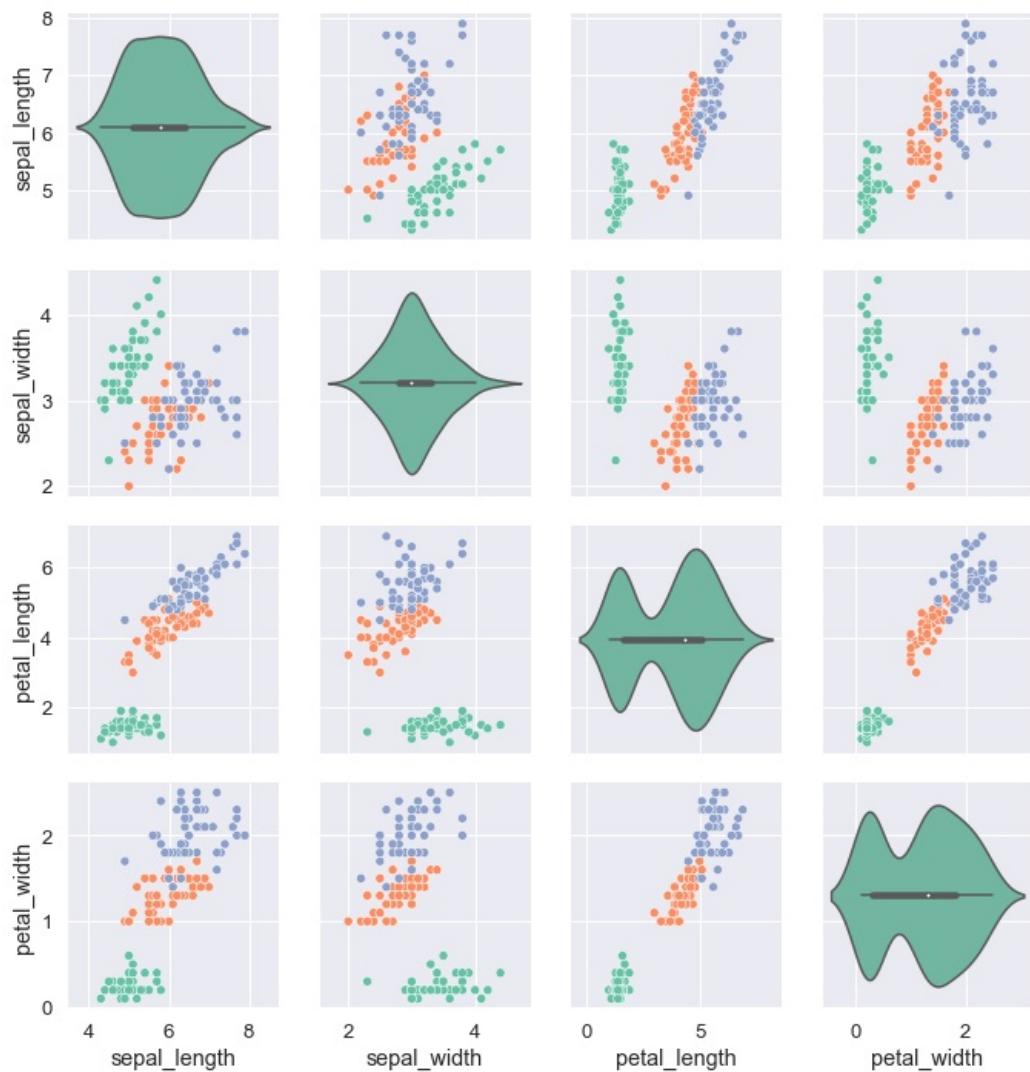
```
In [88]: g = sns.PairGrid(data=iris, hue='species')
g.map(sns.scatterplot)

Out[88]: <seaborn.axisgrid.PairGrid at 0x1fec8d953a0>
```



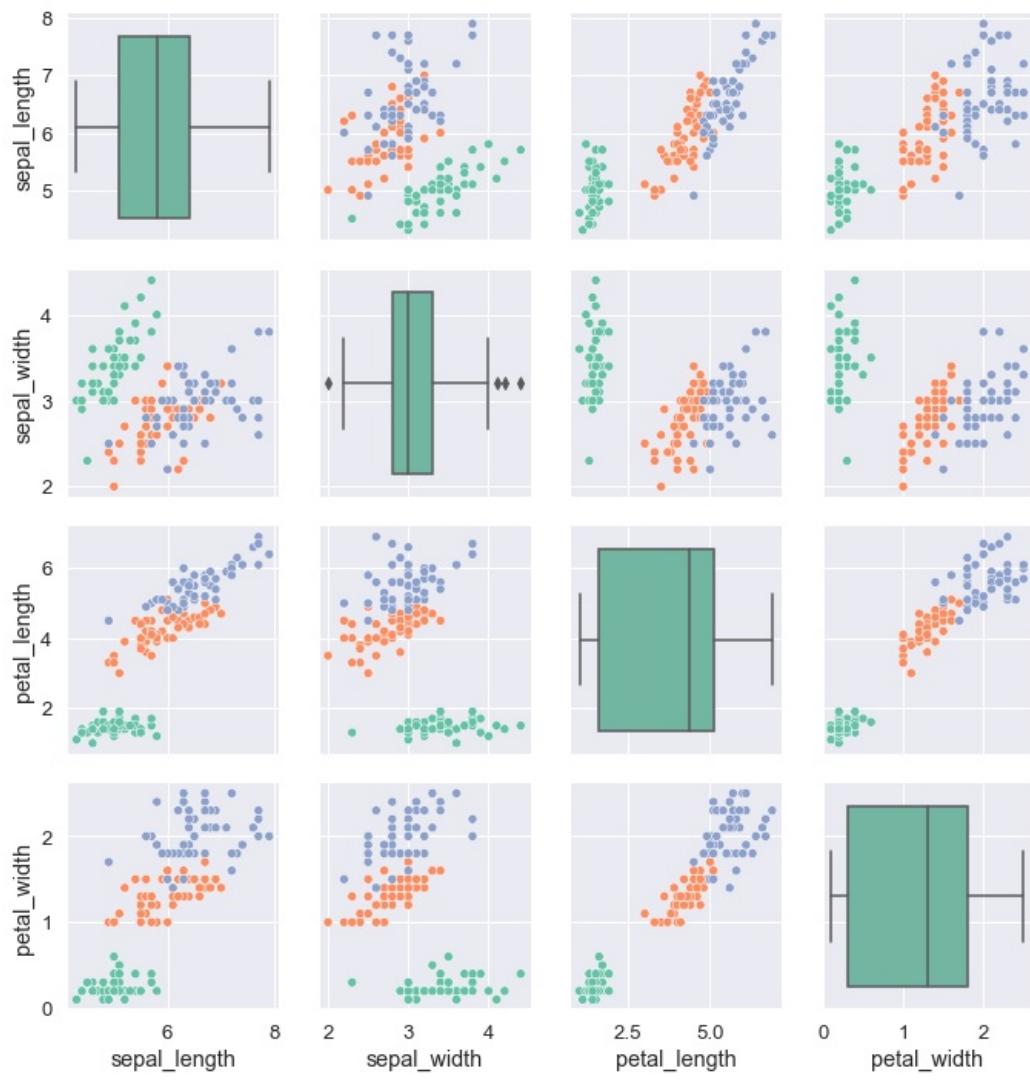
```
In [91]: g = sns.PairGrid(data=iris, hue='species')
g.map_diag(sns.violinplot)
g.map_offdiag(sns.scatterplot)
```

```
Out[91]: <seaborn.axisgrid.PairGrid at 0x1fecbc00280>
```



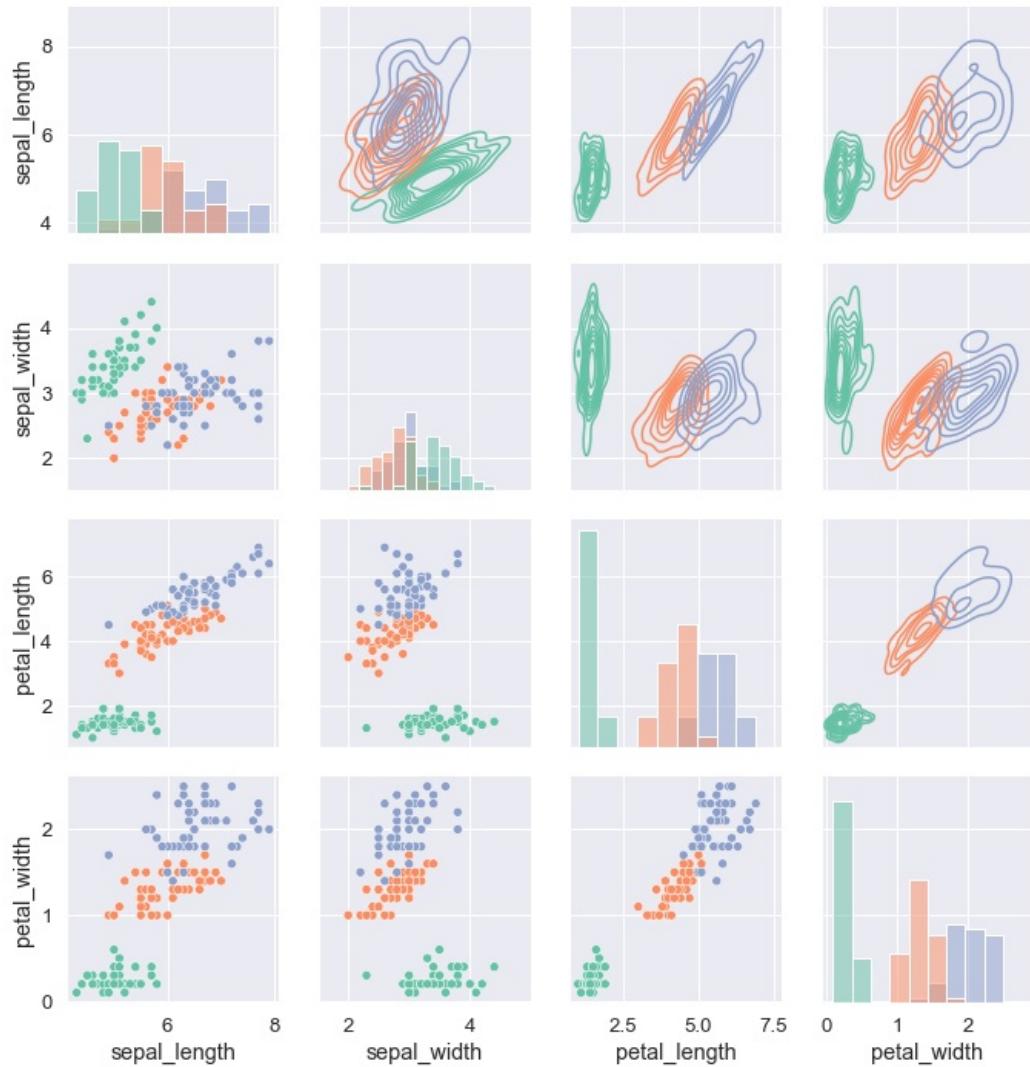
```
In [92]: g = sns.PairGrid(data=iris, hue='species')
g.map_diag(sns.boxplot)
g.map_offdiag(sns.scatterplot)
```

```
Out[92]: <seaborn.axisgrid.PairGrid at 0x1fecc5c4a60>
```



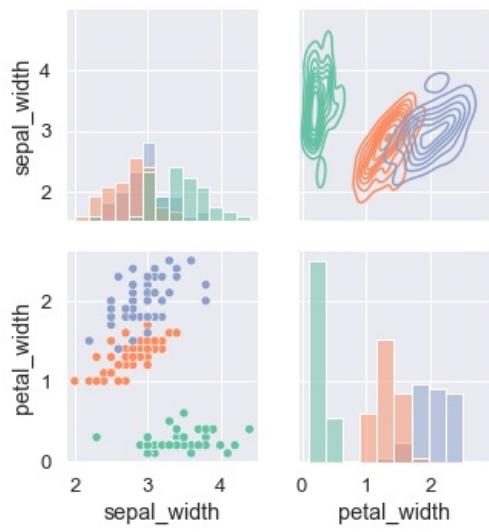
```
In [93]: g = sns.PairGrid(data=iris, hue='species')
g.map_diag(sns.histplot)
g.map_upper(sns.kdeplot)
g.map_lower(sns.scatterplot)
```

```
Out[93]: <seaborn.axisgrid.PairGrid at 0x1fecdf21640>
```



```
In [94]: g = sns.PairGrid(data=iris, hue='species', vars=['sepal_width', 'petal_width'])
g.map_diag(sns.histplot)
g.map_upper(sns.kdeplot)
g.map_lower(sns.scatterplot)
```

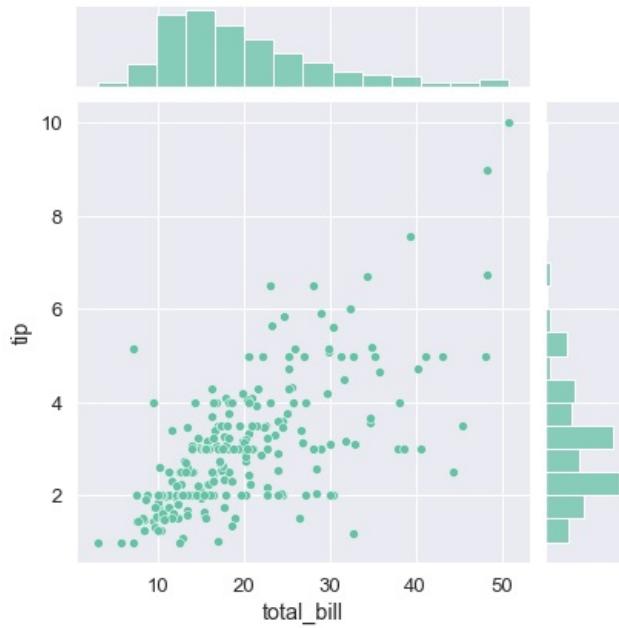
```
Out[94]: <seaborn.axisgrid.PairGrid at 0x1fece8fb20>
```



Joint plot

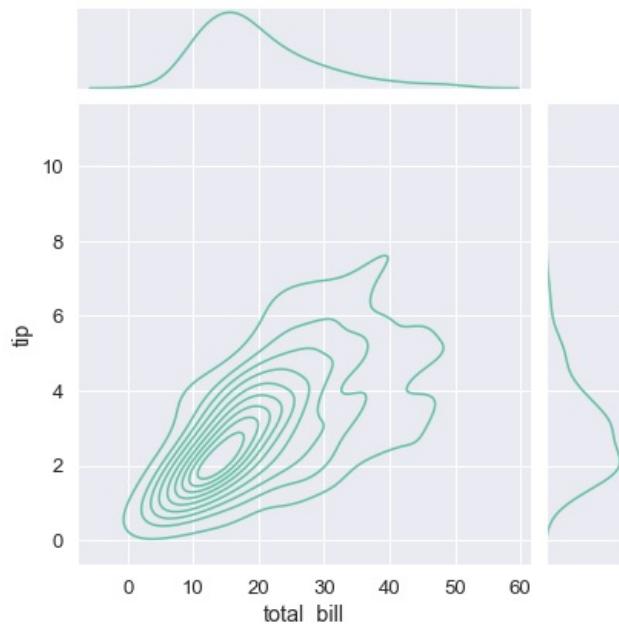
```
In [95]: sns.jointplot(data=tips, x='total_bill', y='tip')
```

```
Out[95]: <seaborn.axisgrid.JointGrid at 0x1feced59e50>
```



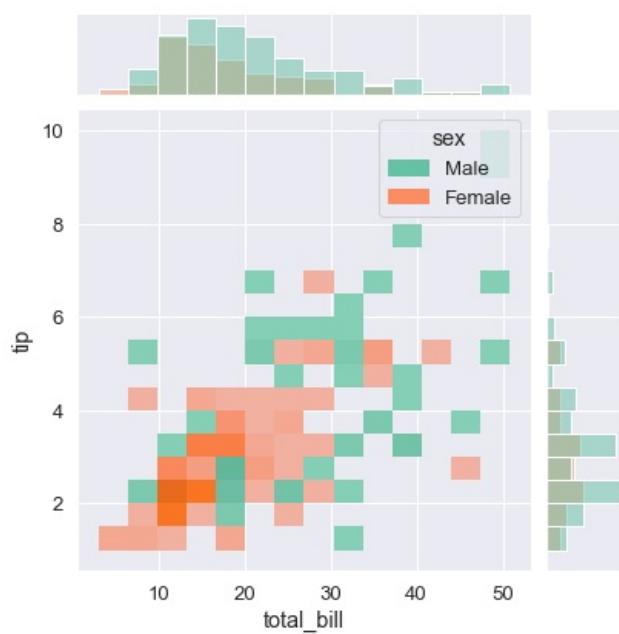
```
In [96]: sns.jointplot(data=tips, x='total_bill', y='tip', kind='kde')
```

```
Out[96]: <seaborn.axisgrid.JointGrid at 0x1fed01d26a0>
```



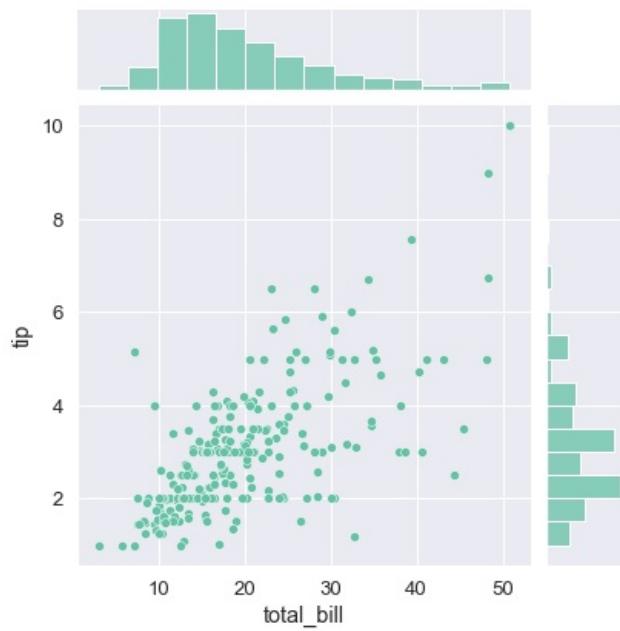
```
In [98]: sns.jointplot(data=tips, x='total_bill', y='tip', kind='hist', hue='sex')
```

```
Out[98]: <seaborn.axisgrid.JointGrid at 0x1fed04fb970>
```



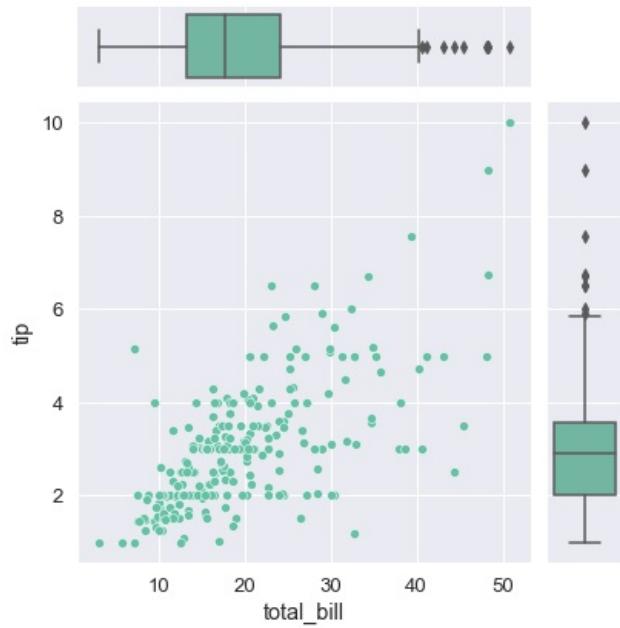
```
In [101... g = sns.JointGrid(data=tips, x='total_bill', y='tip')  
g.plot(sns.scatterplot, sns.histplot)
```

```
Out[101]: <seaborn.axisgrid.JointGrid at 0x1fed1a1f0d0>
```



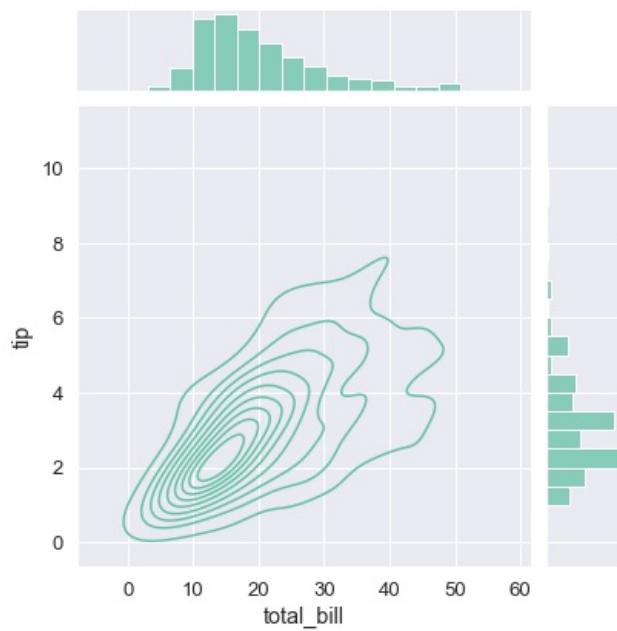
```
In [102]: g = sns.JointGrid(data=tips, x='total_bill', y='tip')
g.plot(sns.scatterplot, sns.boxplot)
```

```
Out[102]: <seaborn.axisgrid.JointGrid at 0x1fed1b1fa0>
```



```
In [103]: g = sns.JointGrid(data=tips, x='total_bill', y='tip')
g.plot(sns.kdeplot, sns.histplot)
```

```
Out[103]: <seaborn.axisgrid.JointGrid at 0x1fed1d118b0>
```



Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js