Why Seaborn?

- provides a layer of abstraction hence simpler to use
- better aesthetics
- more graphs included

Seaborn Roadmap

Types of Functions

- Figure Level
- Axis Level

Main Classification

- Relational Plot
- Distribution Plot
- Categorical Plot
- Regression Plot
- Matrix Plot
- Multiplots

https://seaborn.pydata.org/api.html

1. Relational Plot

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

Plots under this section

- scatterplot
- lineplot

```
import seaborn as sns
import matplotlib.pyplot as plt
import plotly.express as px
```

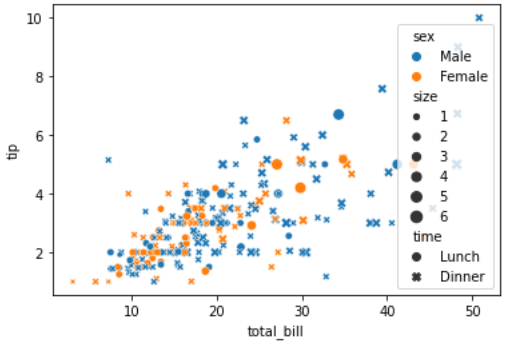
```
tips = sns.load_dataset('tips')
tips
```

	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4
...	...	...	...	...	...	...	...
239	29.03	5.92	Male	No	Sat	Dinner	3
240	27.18	2.00	Female	Yes	Sat	Dinner	2
241	22.67	2.00	Male	Yes	Sat	Dinner	2
242	17.82	1.75	Male	No	Sat	Dinner	2
243	18.78	3.00	Female	No	Thur	Dinner	2

244 rows × 7 columns

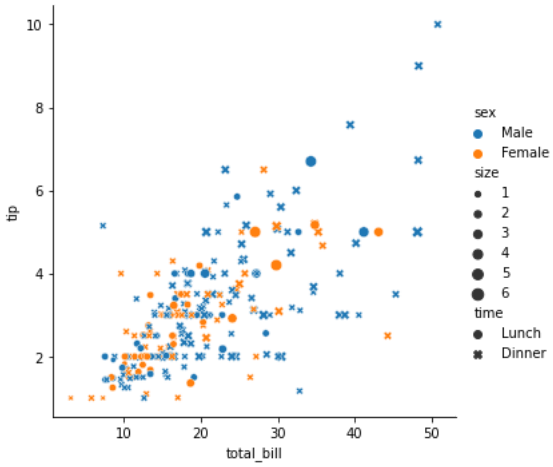
```
# scatter plot -> axes level function
sns.scatterplot(data=tips, x='total_bill', y='tip',hue='sex',style='time',size='size')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f9585634670>
```




```
# relplot -> figure level -> square shape
sns.relplot(data=tips, x='total_bill', y='tip', kind='scatter',hue='sex',style='time',size='size')
```

```
<seaborn.axisgrid.FacetGrid at 0x7f9585625820>
```




```
# scatter using relplot -> size and hue
```

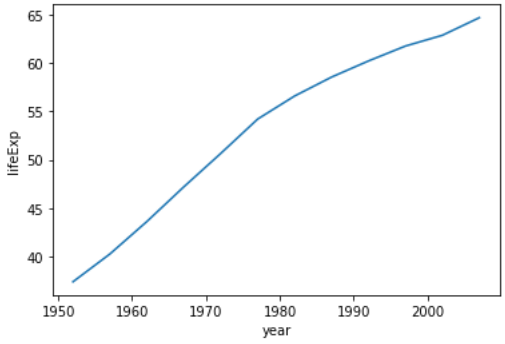
```
# style semantics
```

```
# line plot
gap = px.data.gapminder()
temp_df = gap[gap['country'] == 'India']
temp_df
```

	country	continent	year	lifeExp	pop	gdpPercap	iso_alpha	iso_nu
696	India	Asia	1952	37.373	372000000	546.565749	IND	35
697	India	Asia	1957	40.249	409000000	590.061996	IND	35
698	India	Asia	1962	43.605	454000000	658.347151	IND	35
699	India	Asia	1967	47.193	506000000	700.770611	IND	35
700	India	Asia	1972	50.651	567000000	724.032527	IND	35
701	India	Asia	1977	54.208	634000000	813.337323	IND	35
702	India	Asia	1982	56.596	708000000	855.723538	IND	35
703	India	Asia	1987	58.553	788000000	976.512676	IND	35
704	India	Asia	1992	60.223	872000000	1164.406809	IND	35
705	India	Asia	1997	61.765	959000000	1458.817442	IND	35
706	India	Asia	2002	62.879	1034172547	1746.769454	IND	35
707	India	Asia	2007	64.698	1110396331	2452.210407	IND	35

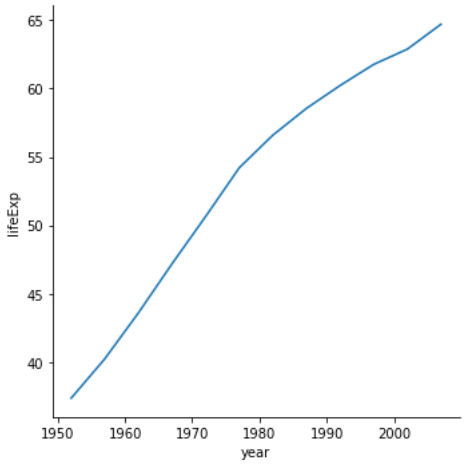
```
# axes level function
sns.lineplot(data=temp_df, x='year', y='lifeExp')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f95854b7c70>
```




```
# using relpplot
sns.relplot(data=temp_df, x='year', y='lifeExp', kind='line')
```

```
<seaborn.axisgrid.FacetGrid at 0x7f9585427a60>
```

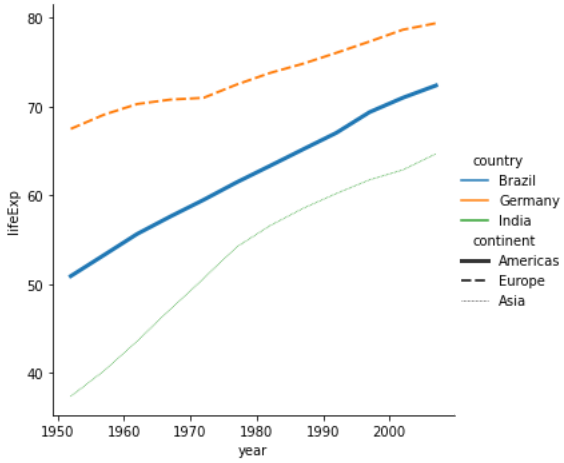



```
# hue -> style
temp_df = gap[gap['country'].isin(['India','Brazil','Germany'])]
temp_df
```

172	Brazil	Americas	1972	59.504	100840058	4985.711467	BRA	
173	Brazil	Americas	1977	61.489	114313951	6660.118654	BRA	
174	Brazil	Americas	1982	63.336	128962939	7030.835878	BRA	
175	Brazil	Americas	1987	65.205	142938076	7807.095818	BRA	
176	Brazil	Americas	1992	67.057	155975974	6950.283021	BRA	
177	Brazil	Americas	1997	69.388	168546719	7957.980824	BRA	
178	Brazil	Americas	2002	71.006	179914212	8131.212843	BRA	
179	Brazil	Americas	2007	72.390	190010647	9065.800825	BRA	
564	Germany	Europe	1952	67.500	69145952	7144.114393	DEU	4
565	Germany	Europe	1957	69.100	71019069	10187.826650	DEU	4
566	Germany	Europe	1962	70.300	73739117	12902.462910	DEU	4
567	Germany	Europe	1967	70.800	76368453	14745.625610	DEU	4
568	Germany	Europe	1972	71.000	78717088	18016.180270	DEU	4
569	Germany	Europe	1977	72.500	78160773	20512.921230	DEU	4
570	Germany	Europe	1982	73.800	78335266	22031.532740	DEU	4
571	Germany	Europe	1987	74.847	77718298	24639.185660	DEU	4
572	Germany	Europe	1992	76.070	80597764	26505.303170	DEU	4
573	Germany	Europe	1997	77.340	82011073	27788.884160	DEU	4
574	Germany	Europe	2002	78.670	82350671	30035.801980	DEU	4
575	Germany	Europe	2007	79.406	82400996	32170.374420	DEU	4
696	India	Asia	1952	37.373	372000000	546.565749	IND	3
697	India	Asia	1957	40.249	409000000	590.061996	IND	3
698	India	Asia	1962	43.605	454000000	658.347151	IND	3
699	India	Asia	1967	47.193	506000000	700.770611	IND	3
700	India	Asia	1972	50.651	567000000	724.032527	IND	3
701	India	Asia	1977	54.208	634000000	813.337323	IND	3
702	India	Asia	1982	56.596	708000000	855.723538	IND	3
703	India	Asia	1987	58.553	788000000	976.512676	IND	3
704	India	Asia	1992	60.223	872000000	1164.406809	IND	3
705	India	Asia	1997	61.765	959000000	1458.817442	IND	3
706	India	Asia	2002	62.879	1034172547	1746.769454	IND	3
707	India	Asia	2007	64.698	1110396331	2452.210407	IND	3

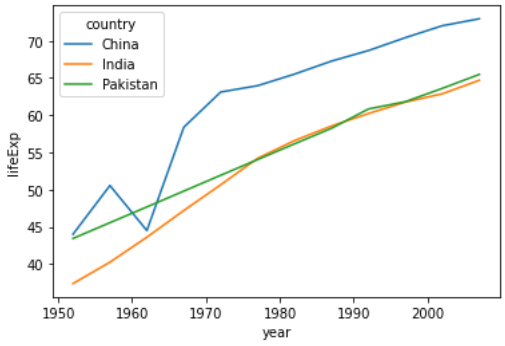
```
sns.relplot(kind='line', data=temp_df, x='year', y='lifeExp', hue='country', style='continent', size='continent')
```

```
<seaborn.axisgrid.FacetGrid at 0x7f9585258c70>
```



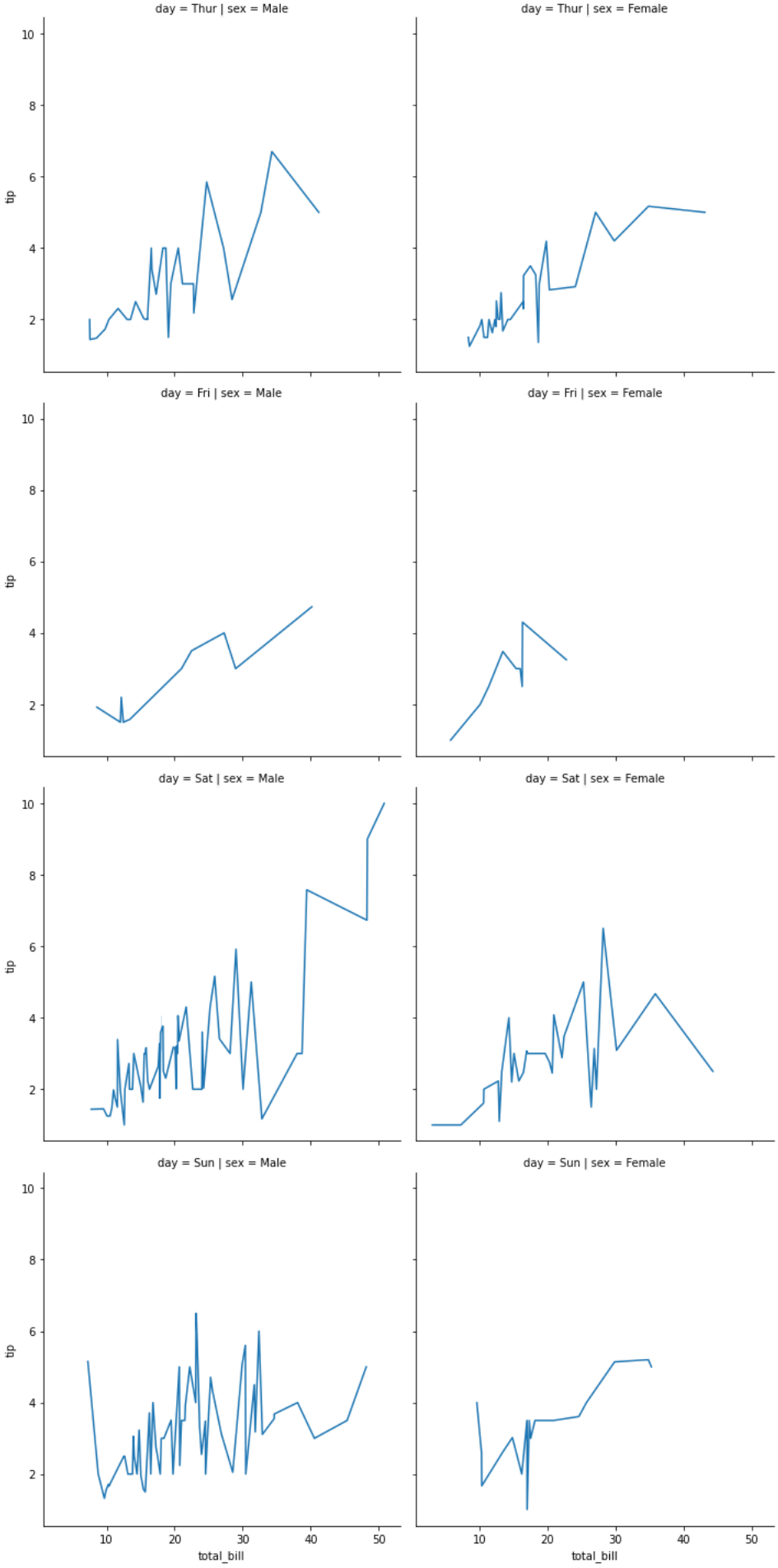

```
sns.lineplot(data=temp_df, x='year', y='lifeExp', hue='country')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f95853837c0>
```




```
# facet plot -> figure level function -> work with relplot
# it will not work with scatterplot and lineplot
sns.relplot(data=tips, x='total_bill', y='tip', kind='line', col='sex', row='day')
```

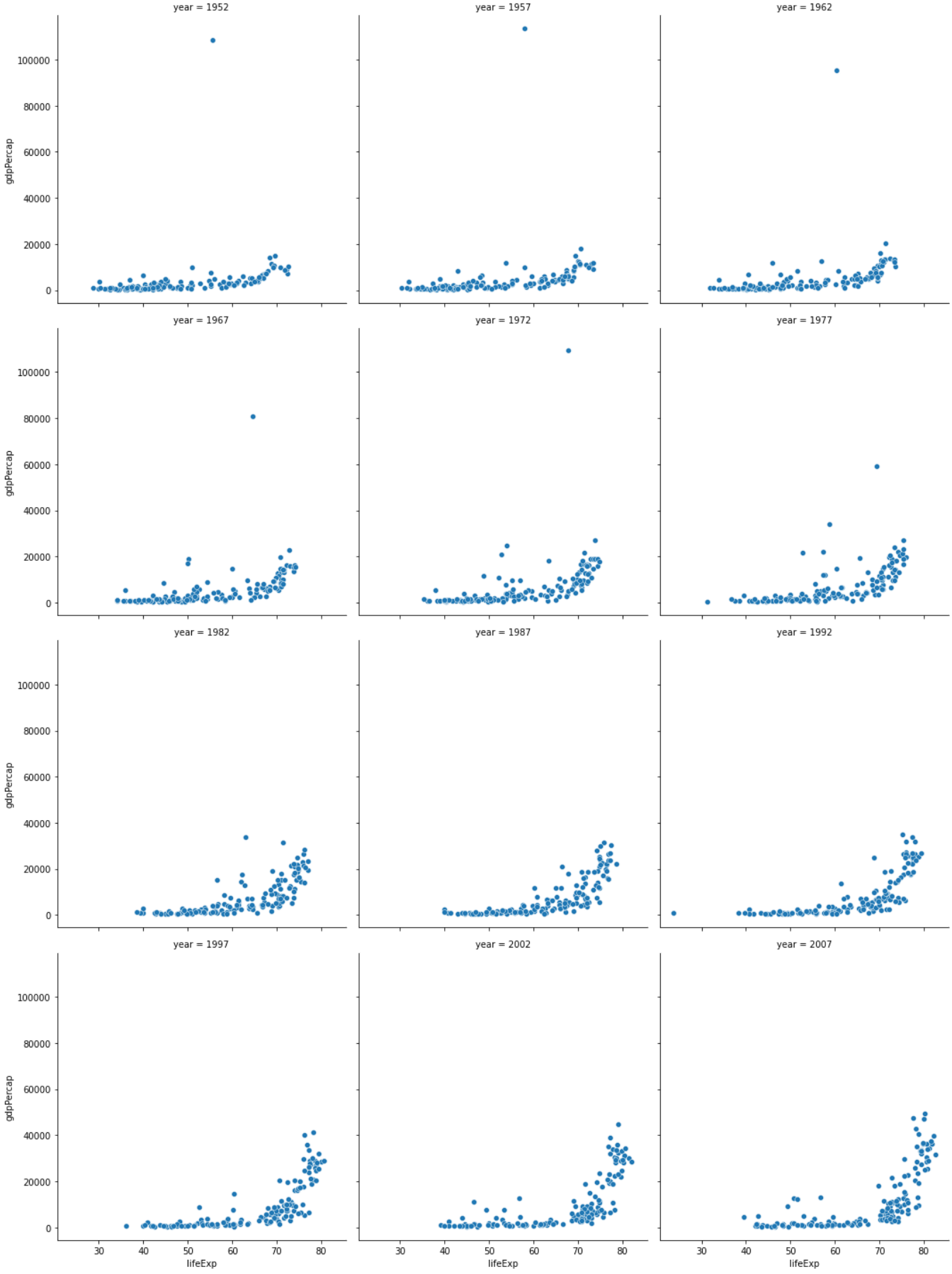
```
<seaborn.axisgrid.FacetGrid at 0x7f9584b8f8b0>
```



+ Code + Text

```
# col wrap
sns.relplot(data=gap, x='lifeExp', y='gdpPercap', kind='scatter', col='year', col_wrap=3)
```

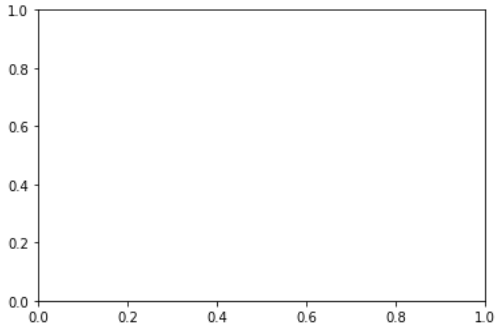
```
<seaborn.axisgrid.FacetGrid at 0x7f95844efb80>
```




```
sns.scatterplot(data=tips, x='total_bill', y='tip', col='sex', row='day')
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-199-13fa0b1b528e> in <module>
----> 1 sns.scatterplot(data=tips, x='total_bill', y='tip', col='sex', row='day')

8 frames
/usr/local/lib/python3.8/dist-packages/matplotlib/artist.py in _update_property(self, k, v)
    999                 func = getattr(self, 'set_' + k, None)
   1000                 if not callable(func):
-> 1001                     raise AttributeError('{!r} object has no property {!r}'
   1002                                          .format(type(self).__name__, k))
   1003                 return func(v)

AttributeError: 'PathCollection' object has no property 'col'
```

SEARCH STACK OVERFLOW



2. 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?

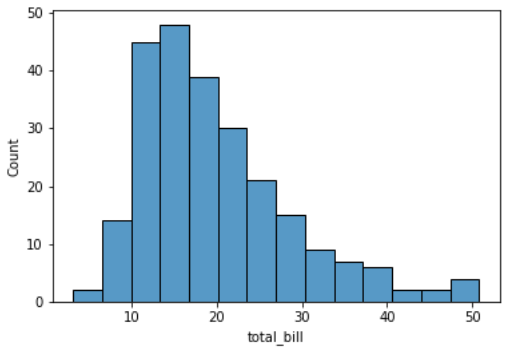
Plots under distribution plot

- histplot
- kdeplot
- rugplot

```
# figure level -> displot
# axes level -> histplot -> kdeplot -> rugplot
```

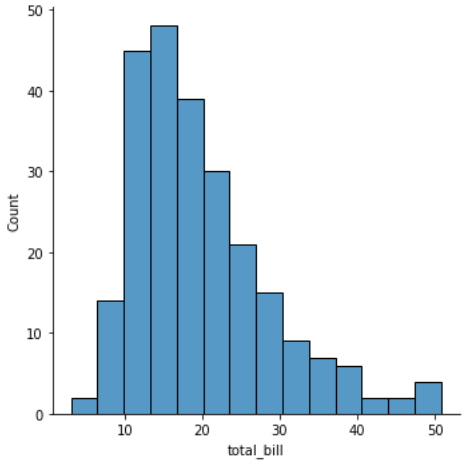
```
# plotting univariate histogram
sns.histplot(data=tips, x='total_bill')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f9583d7dbb0>
```



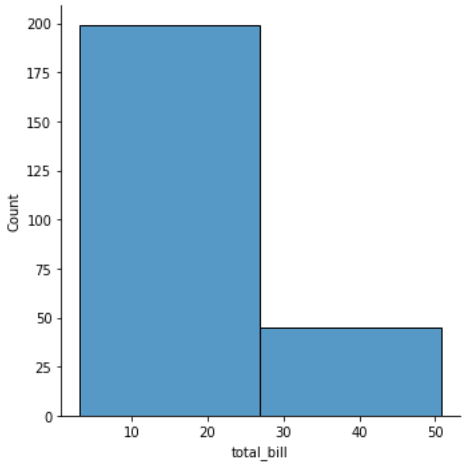

```
sns.displot(data=tips, x='total_bill', kind='hist')
```

```
<seaborn.axisgrid.FacetGrid at 0x7f9583ebc7f0>
```



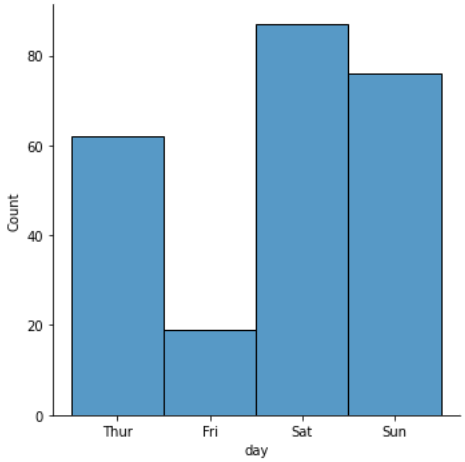

```
# bins parameter
sns.displot(data=tips, x='total_bill', kind='hist',bins=2)
```

```
<seaborn.axisgrid.FacetGrid at 0x7f9583c93280>
```



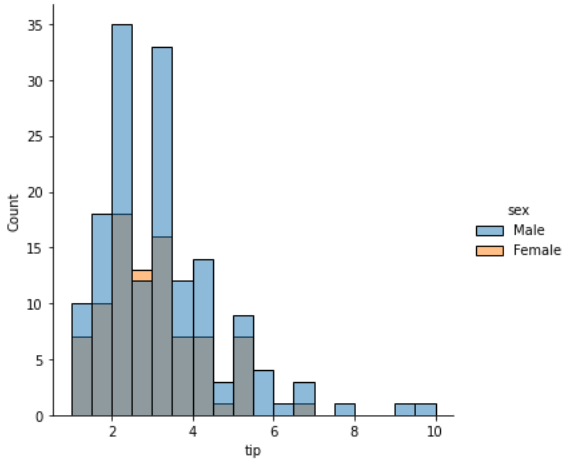

```
# It’s also possible to visualize the distribution of a categorical variable using the logic of a histogram.
# Discrete bins are automatically set for categorical variables

# countplot
sns.displot(data=tips, x='day', kind='hist')
```

```
<seaborn.axisgrid.FacetGrid at 0x7f958517d9d0>
```



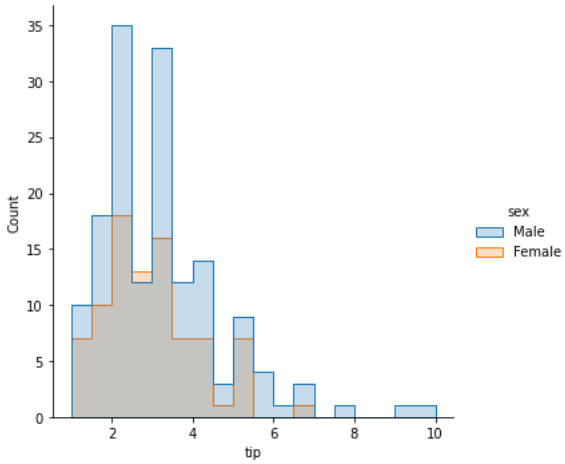

```
# hue parameter
sns.displot(data=tips, x='tip', kind='hist',hue='sex')
```

```
<seaborn.axisgrid.FacetGrid at 0x7f9583d05280>
```




```
# element -> step
sns.displot(data=tips, x='tip', kind='hist',hue='sex',element='step')
```

```
<seaborn.axisgrid.FacetGrid at 0x7f9583c2cfa0>
```



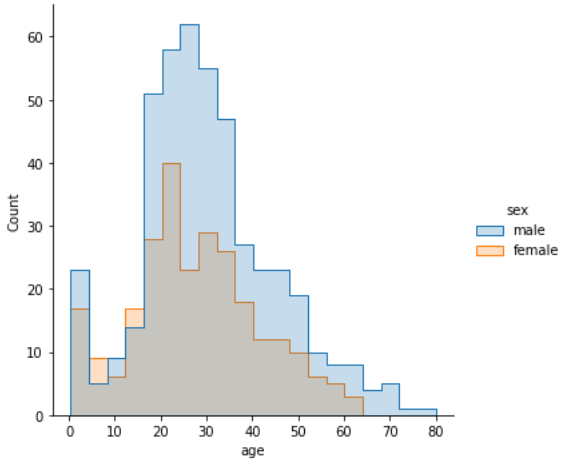

```
titanic = sns.load_dataset('titanic')
titanic
```

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who
0	0	3	male	22.0	1	0	7.2500	S	Third	man
1	1	1	female	38.0	1	0	71.2833	C	First	woman
2	1	3	female	26.0	0	0	7.9250	S	Third	woman
3	1	1	female	35.0	1	0	53.1000	S	First	woman
4	0	3	male	35.0	0	0	8.0500	S	Third	man
...	...	...	...	...	...	...	...	...	...	...
886	0	2	male	27.0	0	0	13.0000	S	Second	man
887	1	1	female	19.0	0	0	30.0000	S	First	woman
888	0	3	female	NaN	1	2	23.4500	S	Third	woman
889	1	1	male	26.0	0	0	30.0000	C	First	man
890	0	3	male	32.0	0	0	7.7500	Q	Third	man

891 rows × 15 columns

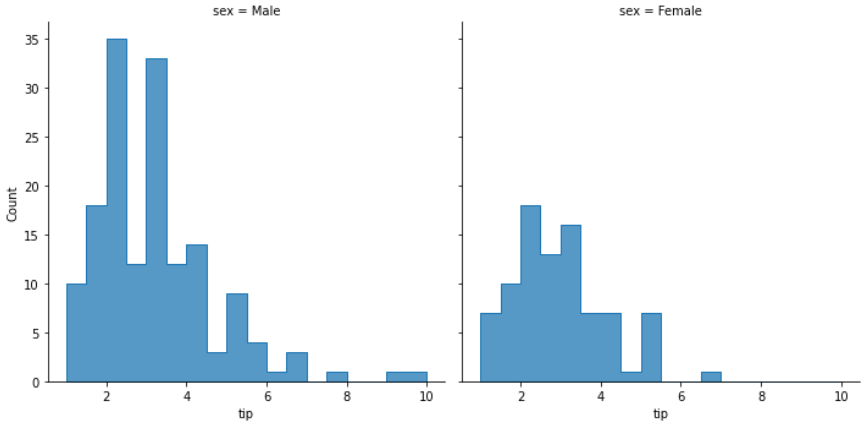
```
sns.displot(data=titanic, x='age', kind='hist',element='step',hue='sex')
```

```
<seaborn.axisgrid.FacetGrid at 0x7f9583c1b9d0>
```



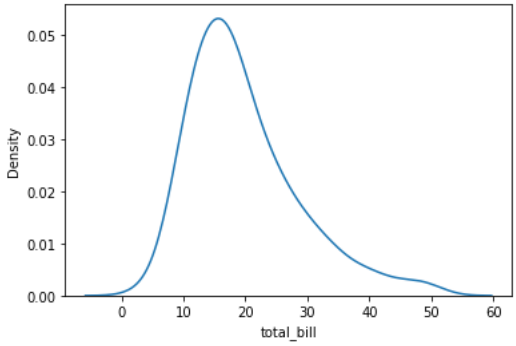

```
# faceting using col and row -> not work on histplot function
sns.displot(data=tips, x='tip', kind='hist',col='sex',element='step')
```

```
<seaborn.axisgrid.FacetGrid at 0x7f9583904850>
```



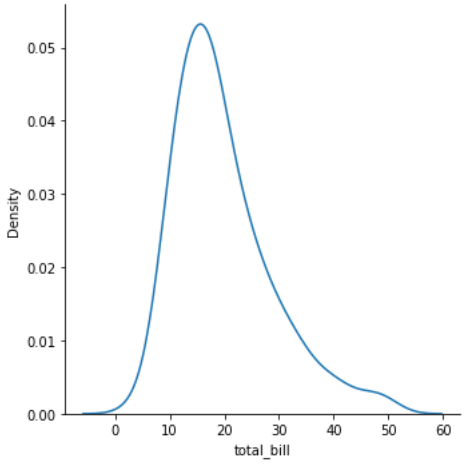

```
# kdeplot
# Rather than using discrete bins, a KDE plot smooths the observations with a Gaussian kernel, producing a continuous densit
sns.kdeplot(data=tips,x='total_bill')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f958384a160>
```



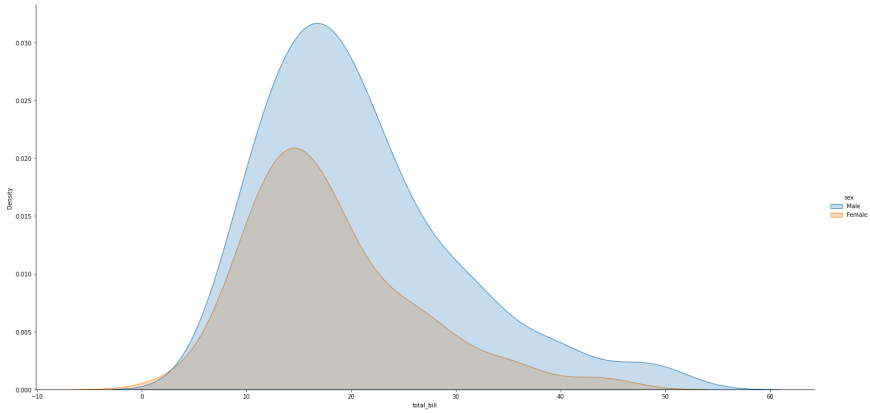

```
sns.displot(data=tips,x='total_bill',kind='kde')
```

```
<seaborn.axisgrid.FacetGrid at 0x7f95838c2790>
```



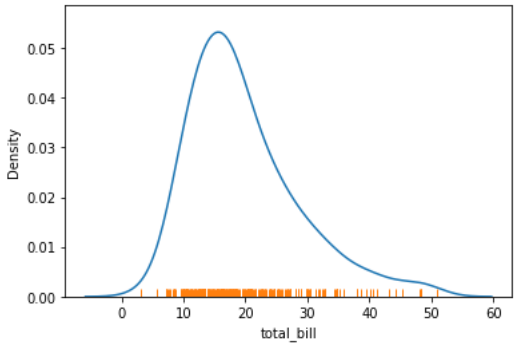

```
# hue -> fill
sns.displot(data=tips,x='total_bill',kind='kde',hue='sex',fill=True,height=10,aspect=2)
```

```
<seaborn.axisgrid.FacetGrid at 0x7f95821a35b0>
```



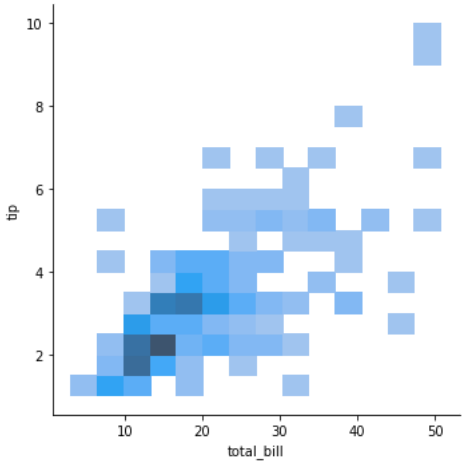

```
# Rugplot

# Plot marginal distributions by drawing ticks along the x and y axes.

# This function is intended to complement other plots by showing the location of individual observations in an unobtrusive w
sns.kdeplot(data=tips,x='total_bill')
sns.rugplot(data=tips,x='total_bill')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f95836ff0d0>
```



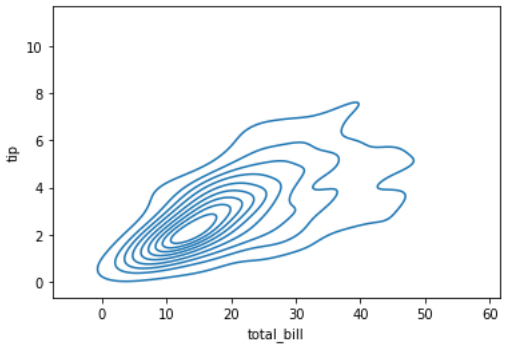
```
# Bivariate histogram
# A bivariate histogram bins the data within rectangles that tile the plot
# and then shows the count of observations within each rectangle with the fill color

# sns.histplot(data=tips, x='total_bill', y='tip')
sns.displot(data=tips, x='total_bill', y='tip',kind='hist')
```

```
<seaborn.axisgrid.FacetGrid at 0x7f958362fc10>
```




```
# Bivariate Kdeplot
# a bivariate KDE plot smoothes the (x, y) observations with a 2D Gaussian
sns.kdeplot(data=tips, x='total_bill', y='tip')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f95835b60d0>
```



2. Matrix Plot

- Heatmap
- Clustermap

```
# Heatmap

# Plot rectangular data as a color-encoded matrix
temp_df = gap.pivot(index='country',columns='year',values='lifeExp')

# axes level function
plt.figure(figsize=(15,15))
sns.heatmap(temp_df)
```