# Muhammad Danial

muhammaddanialarain@gmail.com

## What is Seaborn?

Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics

## 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

Differnce between Figure Level and Axis Level?

Axes-level functions take an explicit ax argument and return an Axes object. For figure-level functions, these need to have overall control over the figure plotted.

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
```

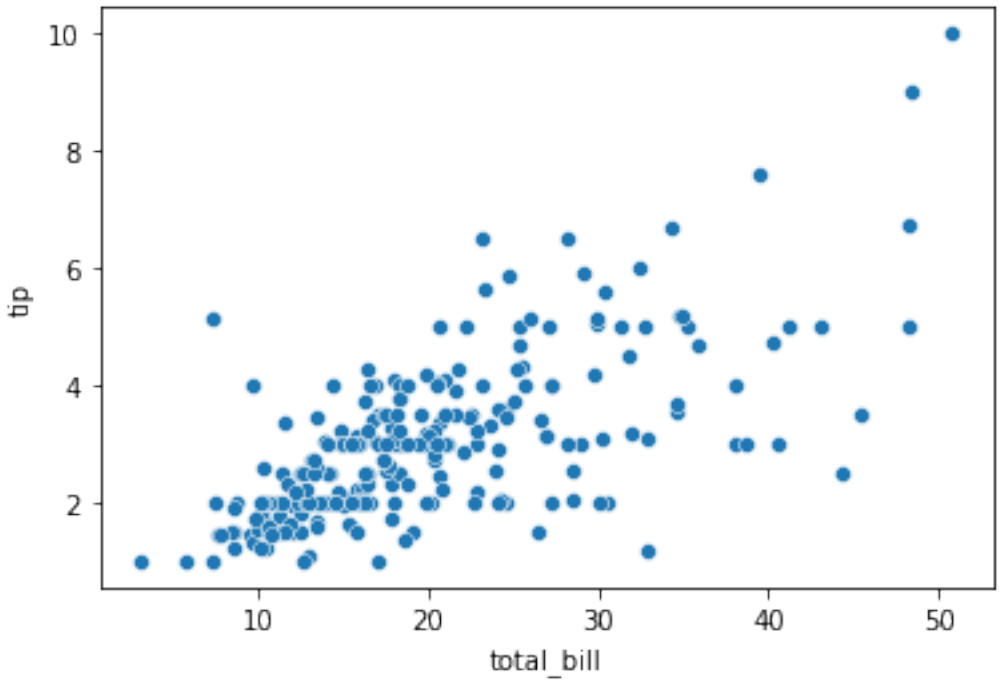
| | total_bill | tip | sex | smoker | day | time | size |
|---|---|---|---|---|---|---|---|
| 0 | 16.99 | 1.01 | Female | No | Sun | Dinner | 2 |
| 1 | 10.34 | 1.66 | Male | No | Sun | Dinner | 3 |
| 2 | 21.01 | 3.50 | Male | No | Sun | Dinner | 3 |
| 3 | 23.68 | 3.31 | Male | No | Sun | Dinner | 2 |
| 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 x 7 columns]

scatter plot --> axis level function

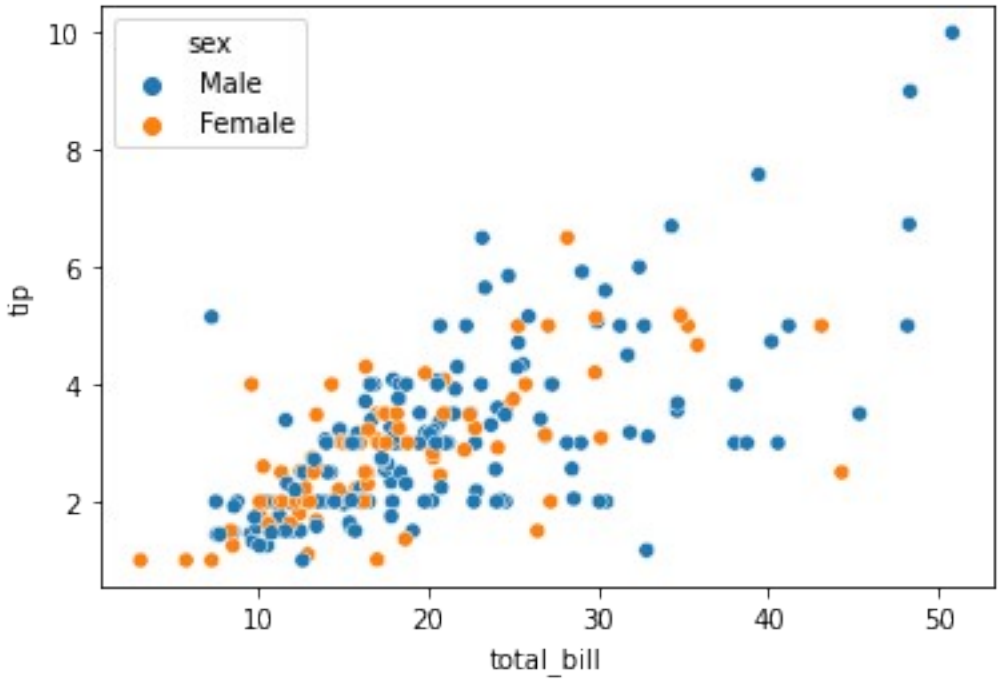
```
sns.scatterplot(data=tips,x='total_bill',y='tip')
```

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



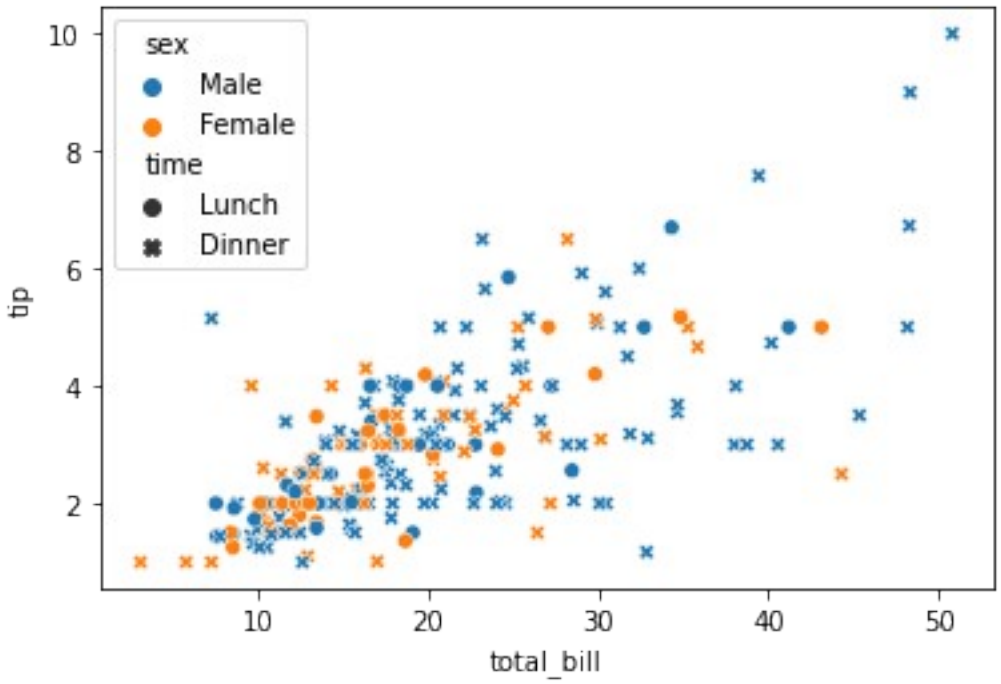

```
sns.scatterplot(data=tips,x='total_bill',y='tip',hue='sex')
```

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




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

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



```
sns.scatterplot(data=tips,x='total_bill',y='tip',hue='sex',style='time
',size='size')
```

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

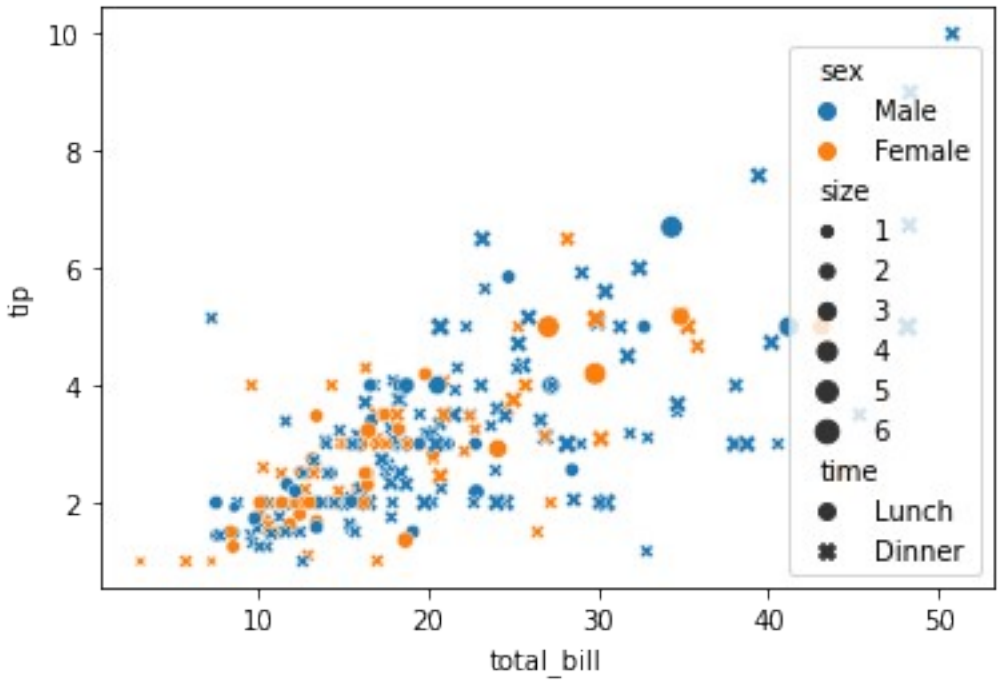
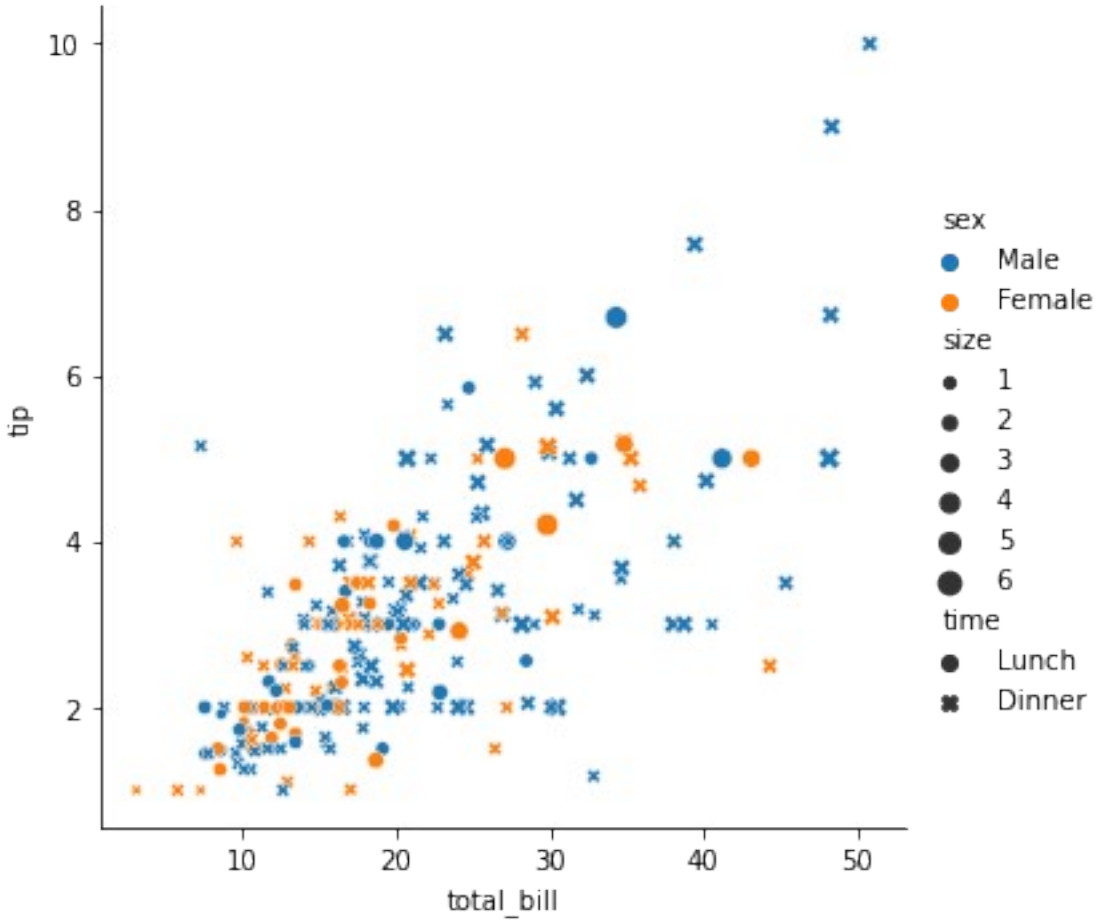



figure function

scatter using relplot --> size and hue

```
sns.relplot(data=tips,x='total_bill',y='tip',kind='scatter',hue='sex',
style='time',size='size')
```

```
<seaborn.axisgrid.FacetGrid at 0x28bdb1327f0>
```




line plot

```
gap=px.data.gapminder()
temp_df=gap[gap['country']=='Pakistan']
temp_df
```

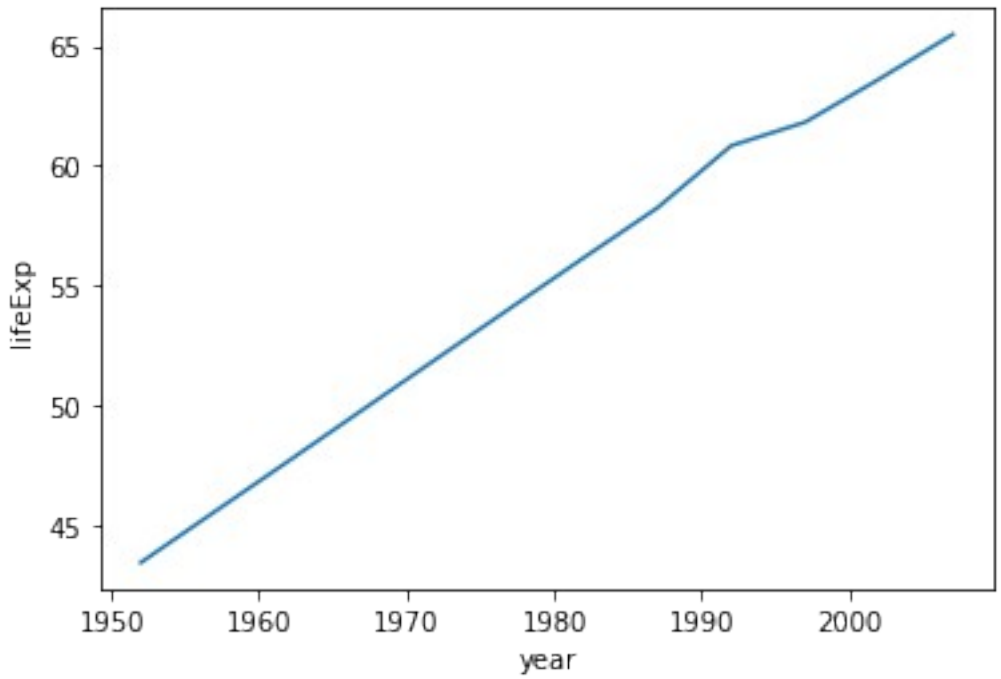
```
       country continent  year  lifeExp        pop     gdpPercap
iso_alpha  \
1164  Pakistan      Asia  1952   43.436   41346560   684.597144
PAK
1165  Pakistan      Asia  1957   45.557   46679944   747.083529
PAK
1166  Pakistan      Asia  1962   47.670   53100671   803.342742
PAK
1167  Pakistan      Asia  1967   49.800   60641899   942.408259
PAK
1168  Pakistan      Asia  1972   51.929   69325921  1049.938981
PAK
1169  Pakistan      Asia  1977   54.043   78152686  1175.921193
PAK
1170  Pakistan      Asia  1982   56.158   91462088  1443.429832
PAK
```

```
1171  Pakistan      Asia  1987   58.245  105186881  1704.686583
PAK
1172  Pakistan      Asia  1992   60.838  120065004  1971.829464
PAK
1173  Pakistan      Asia  1997   61.818  135564834  2049.350521
PAK
1174  Pakistan      Asia  2002   63.610  153403524  2092.712441
PAK
1175  Pakistan      Asia  2007   65.483  169270617  2605.947580
PAK
```

```
      iso_num
1164      586
1165      586
1166      586
1167      586
1168      586
1169      586
1170      586
1171      586
1172      586
1173      586
1174      586
1175      586
```

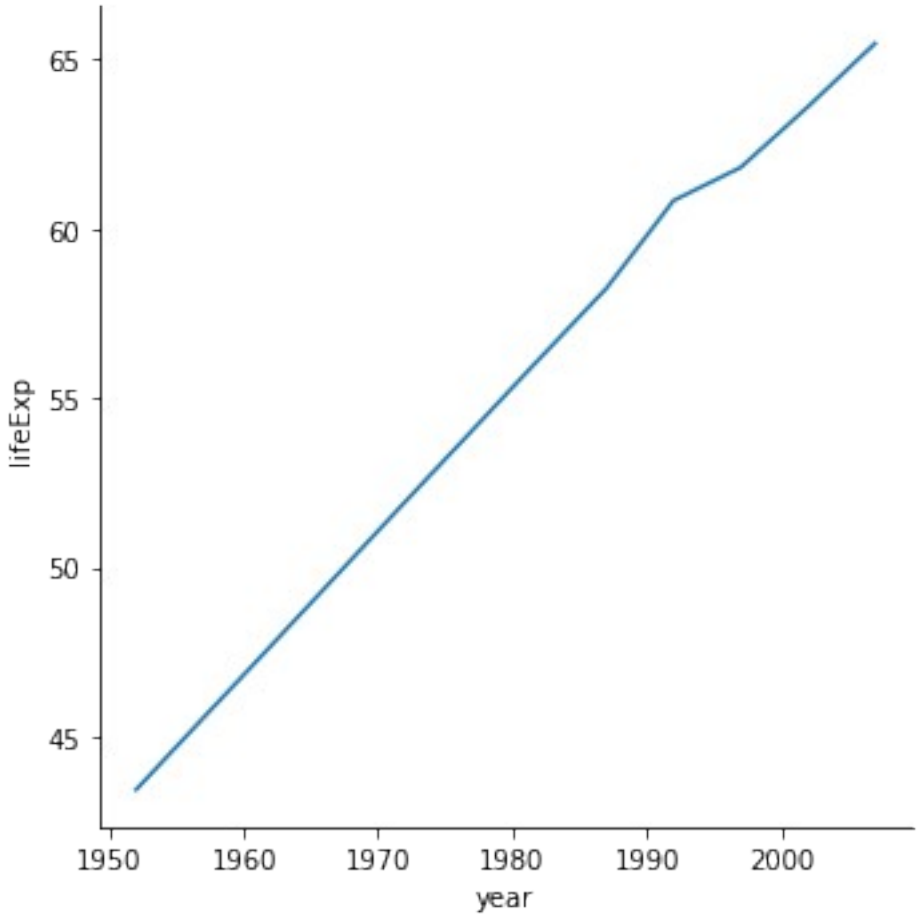
```
# axis level function
sns.lineplot(data=temp_df,x='year',y='lifeExp')
```

```
<AxesSubplot:xlabel='year', ylabel='lifeExp'>
```



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

```
<seaborn.axisgrid.FacetGrid at 0x28bdaf67a00>
```




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

```
       country continent  year  lifeExp        pop     gdpPercap
iso_alpha  \
168     Brazil  Americas  1952   50.917   56602560   2108.944355
BRA
169     Brazil  Americas  1957   53.285   65551171   2487.365989
BRA
170     Brazil  Americas  1962   55.665   76039390   3336.585802
BRA
171     Brazil  Americas  1967   57.632   88049823   3429.864357
BRA
172     Brazil  Americas  1972   59.504  100840058   4985.711467
BRA
```

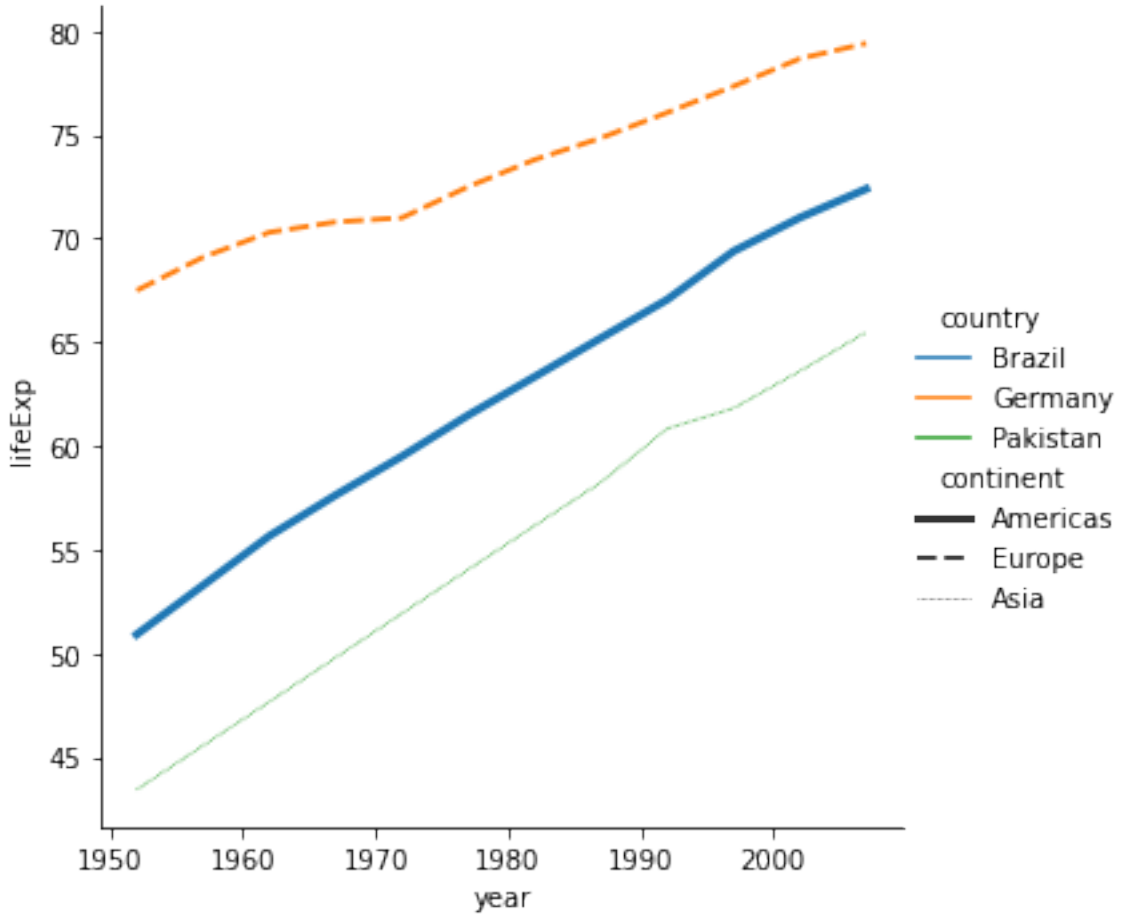
| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 173 | Brazil | Americas | 1977 | 61.489 | 114313951 | 6660.118654 | BRA |
| 174 | Brazil | Americas | 1982 | 63.336 | 128962939 | 7030.835878 | BRA |
| 175 | Brazil | Americas | 1987 | 65.205 | 142938076 | 7807.095818 | BRA |
| 176 | Brazil | Americas | 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 |
| 565 | Germany | Europe | 1957 | 69.100 | 71019069 | 10187.826650 | DEU |
| 566 | Germany | Europe | 1962 | 70.300 | 73739117 | 12902.462910 | DEU |
| 567 | Germany | Europe | 1967 | 70.800 | 76368453 | 14745.625610 | DEU |
| 568 | Germany | Europe | 1972 | 71.000 | 78717088 | 18016.180270 | DEU |
| 569 | Germany | Europe | 1977 | 72.500 | 78160773 | 20512.921230 | DEU |
| 570 | Germany | Europe | 1982 | 73.800 | 78335266 | 22031.532740 | DEU |
| 571 | Germany | Europe | 1987 | 74.847 | 77718298 | 24639.185660 | DEU |
| 572 | Germany | Europe | 1992 | 76.070 | 80597764 | 26505.303170 | DEU |
| 573 | Germany | Europe | 1997 | 77.340 | 82011073 | 27788.884160 | DEU |
| 574 | Germany | Europe | 2002 | 78.670 | 82350671 | 30035.801980 | DEU |
| 575 | Germany | Europe | 2007 | 79.406 | 82400996 | 32170.374420 | DEU |
| 1164 | Pakistan | Asia | 1952 | 43.436 | 41346560 | 684.597144 | PAK |
| 1165 | Pakistan | Asia | 1957 | 45.557 | 46679944 | 747.083529 | PAK |
| 1166 | Pakistan | Asia | 1962 | 47.670 | 53100671 | 803.342742 | PAK |
| 1167 | Pakistan | Asia | 1967 | 49.800 | 60641899 | 942.408259 | PAK |
| 1168 | Pakistan | Asia | 1972 | 51.929 | 69325921 | 1049.938981 | PAK |
| 1169 | Pakistan | Asia | 1977 | 54.043 | 78152686 | 1175.921193 | |

```
PAK
1170  Pakistan      Asia  1982   56.158   91462088   1443.429832
PAK
1171  Pakistan      Asia  1987   58.245  105186881   1704.686583
PAK
1172  Pakistan      Asia  1992   60.838  120065004   1971.829464
PAK
1173  Pakistan      Asia  1997   61.818  135564834   2049.350521
PAK
1174  Pakistan      Asia  2002   63.610  153403524   2092.712441
PAK
1175  Pakistan      Asia  2007   65.483  169270617   2605.947580
PAK

      iso_num
168        76
169        76
170        76
171        76
172        76
173        76
174        76
175        76
176        76
177        76
178        76
179        76
564       276
565       276
566       276
567       276
568       276
569       276
570       276
571       276
572       276
573       276
574       276
575       276
1164      586
1165      586
1166      586
1167      586
1168      586
1169      586
1170      586
1171      586
1172      586
1173      586
```

```
1174      586
1175      586
```

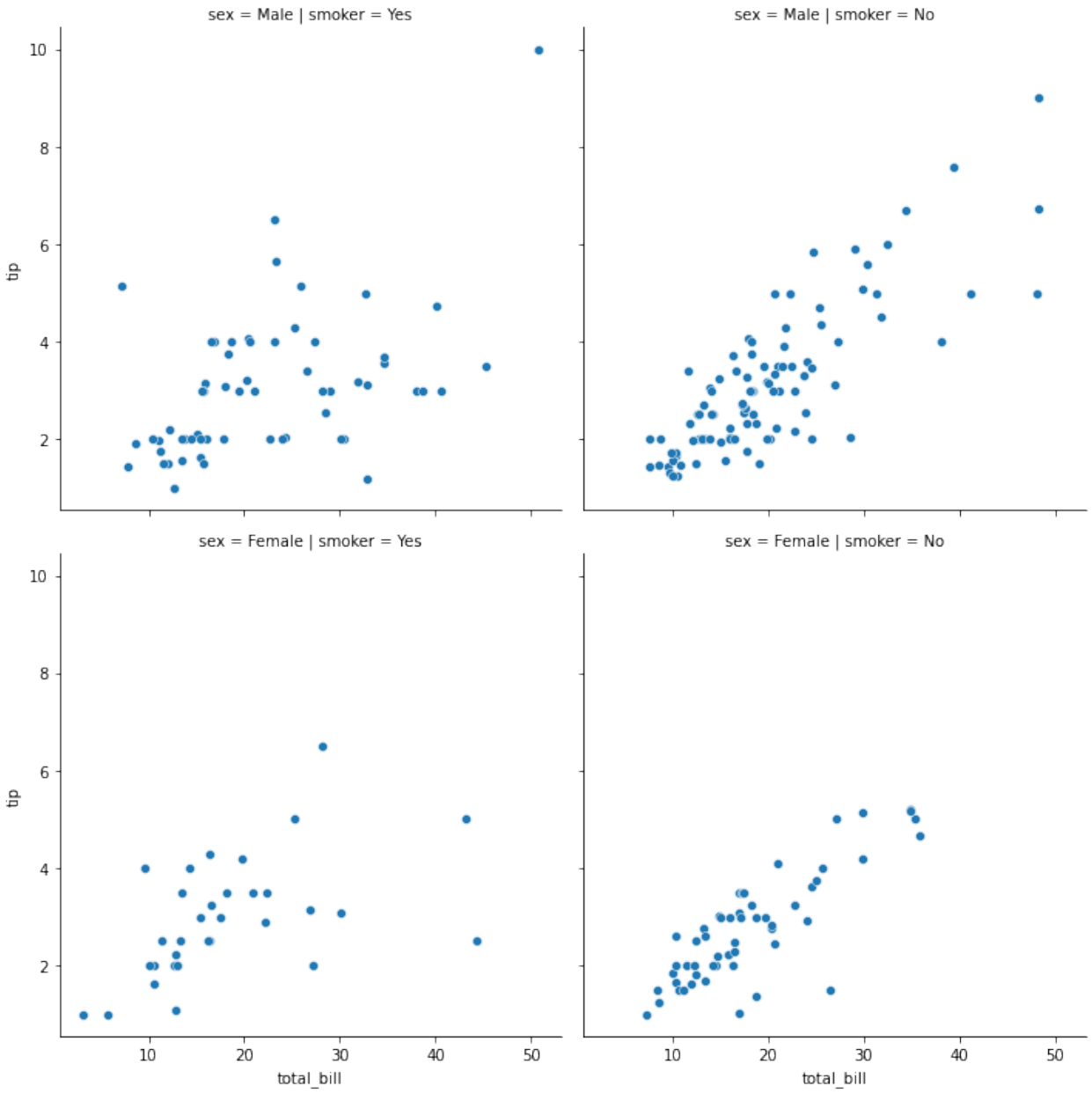
```
sns.relplot(kind='line',data=temp_df,x='year',y='lifeExp',hue='country
',style='continent',size='continent')
```

```
<seaborn.axisgrid.FacetGrid at 0x28bdb6e7b20>
```



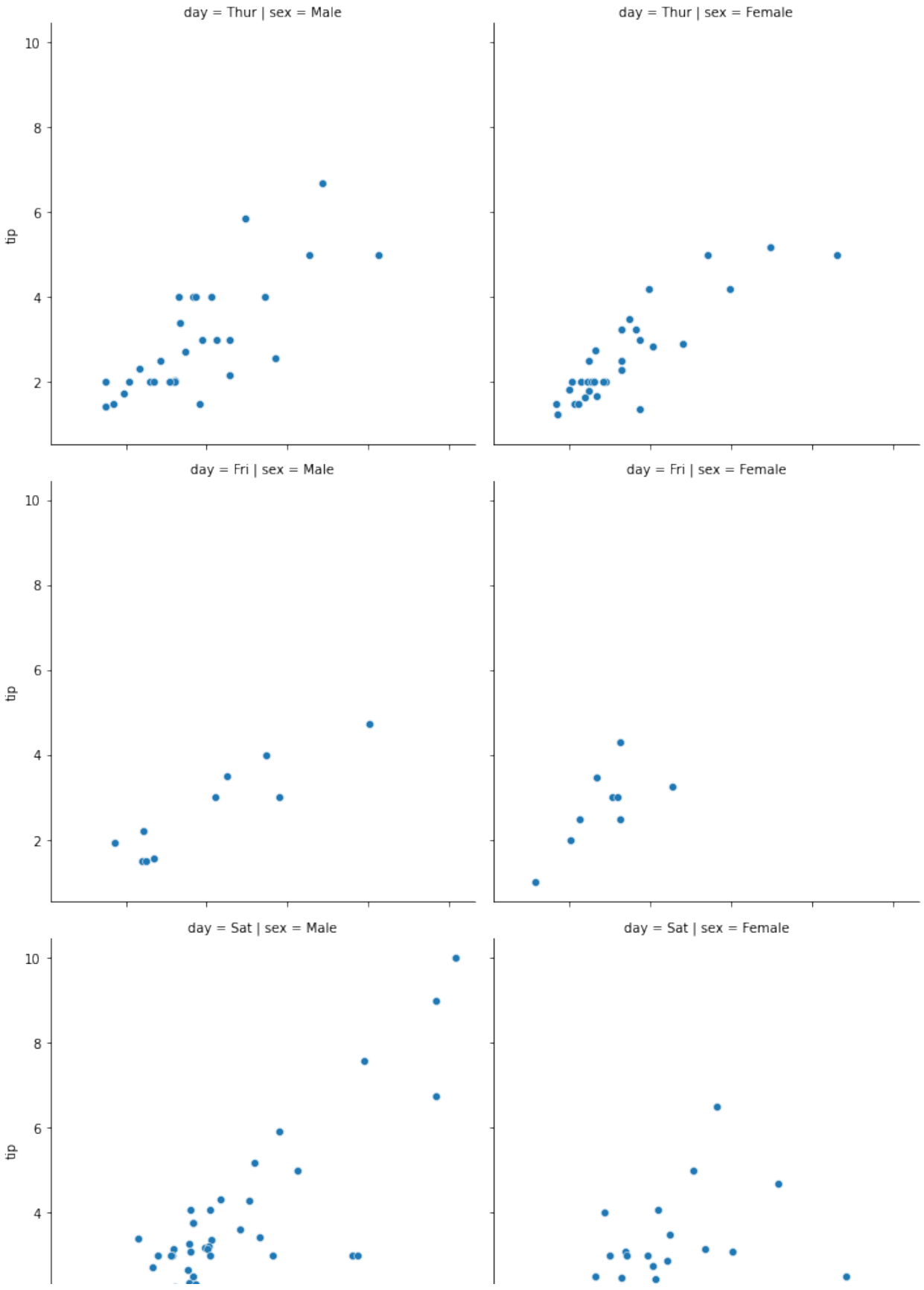

```
# facet plot --> figure level --> work with relplot
sns.relplot(data=tips,x='total_bill',y='tip',kind='scatter',col='smoke
r',row='sex')
```

```
<seaborn.axisgrid.FacetGrid at 0x28bdb6eb220>
```



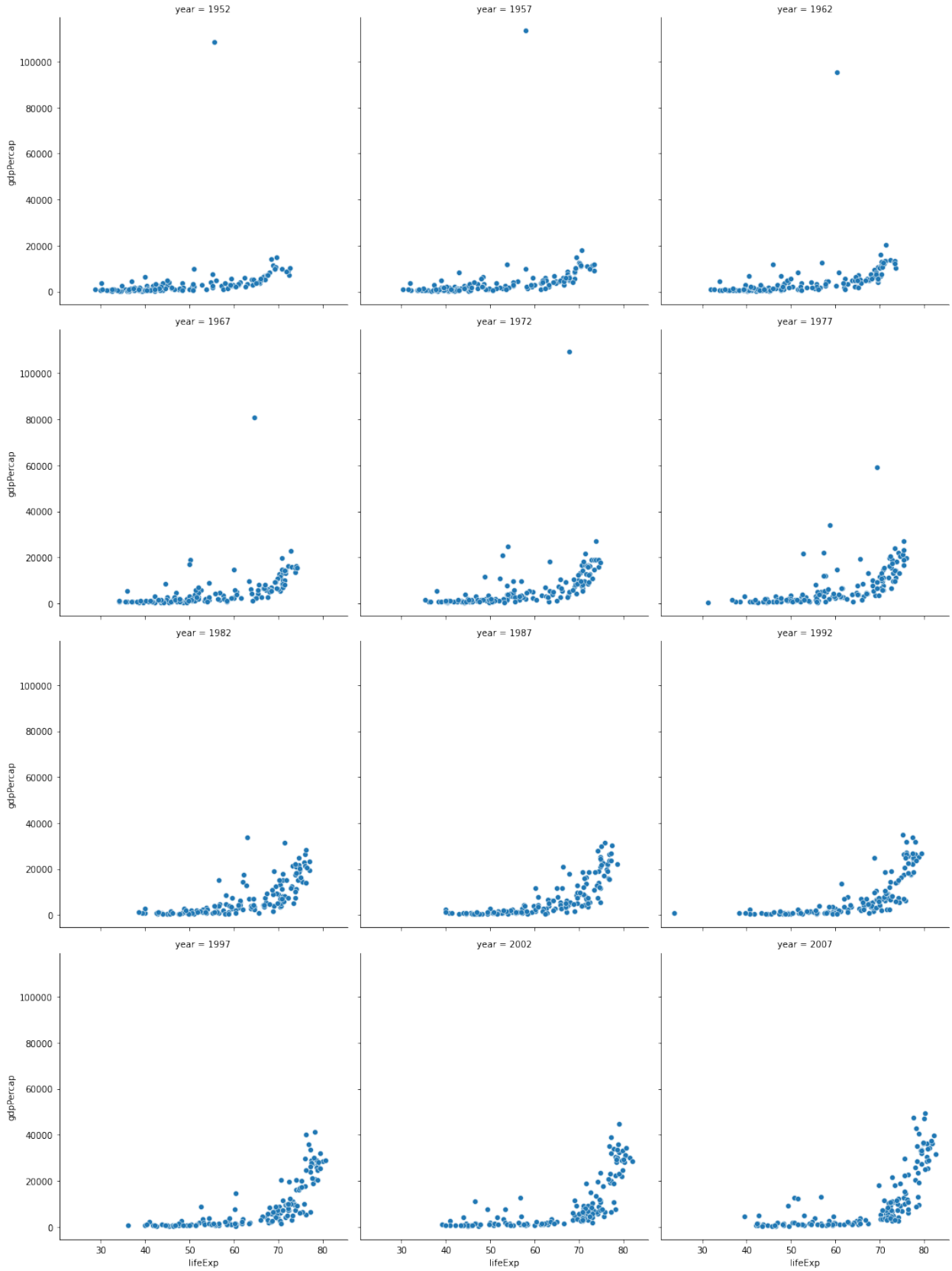

```
# facet plot
sns.relplot(data=tips,x='total_bill',y='tip',kind='scatter',col='sex',
row='day')
```

```
<seaborn.axisgrid.FacetGrid at 0x28bdb240220>
```


day = Thur | sex = Male
day = Thur | sex = Female
day = Fri | sex = Male
day = Fri | sex = Female
day = Sat | sex = Male
day = Sat | sex = Female
tip
10
8
6
4
2

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

```
<seaborn.axisgrid.FacetGrid at 0x28bdb6e7c40>
```


year = 1952
year = 1957
year = 1962
year = 1967
year = 1972
year = 1977
year = 1982
year = 1987
year = 1992
year = 1997
year = 2002
year = 2007
gdpPercap
lifeExp
0
20000
40000
60000
80000
100000
30
40
50
60
70
80

## 2. Distribution Plots

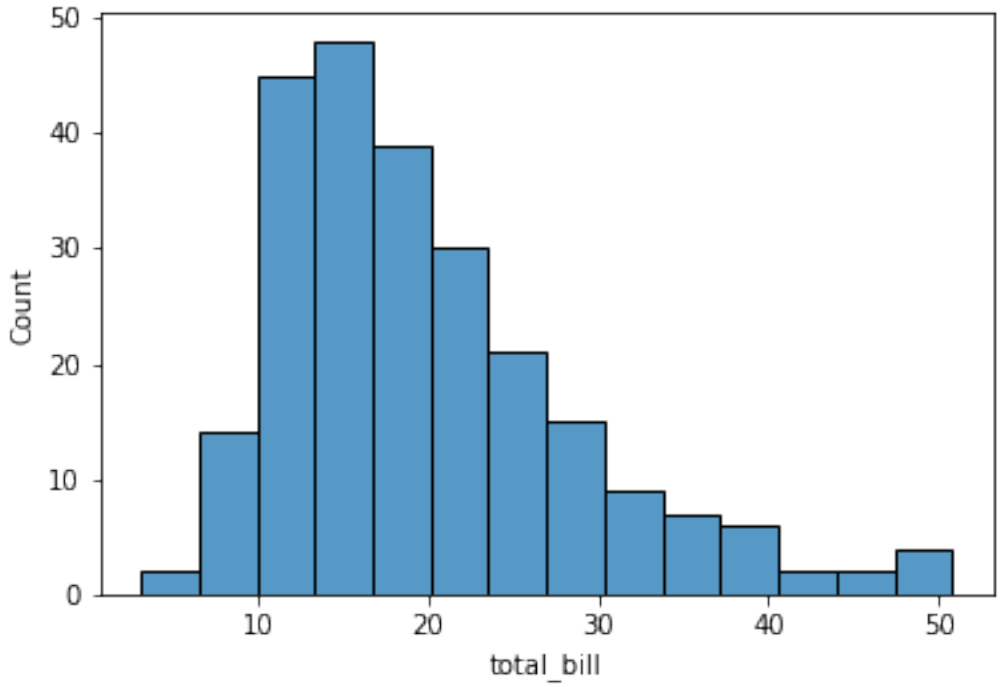
- used for univariate analysis
- used to find out the distribution
- Range of the observation
- Central Tendency
- is the data bimodal?
- Are there outliers?

Plots under distribution plot

- histplot
- kdeplot
- rugplot

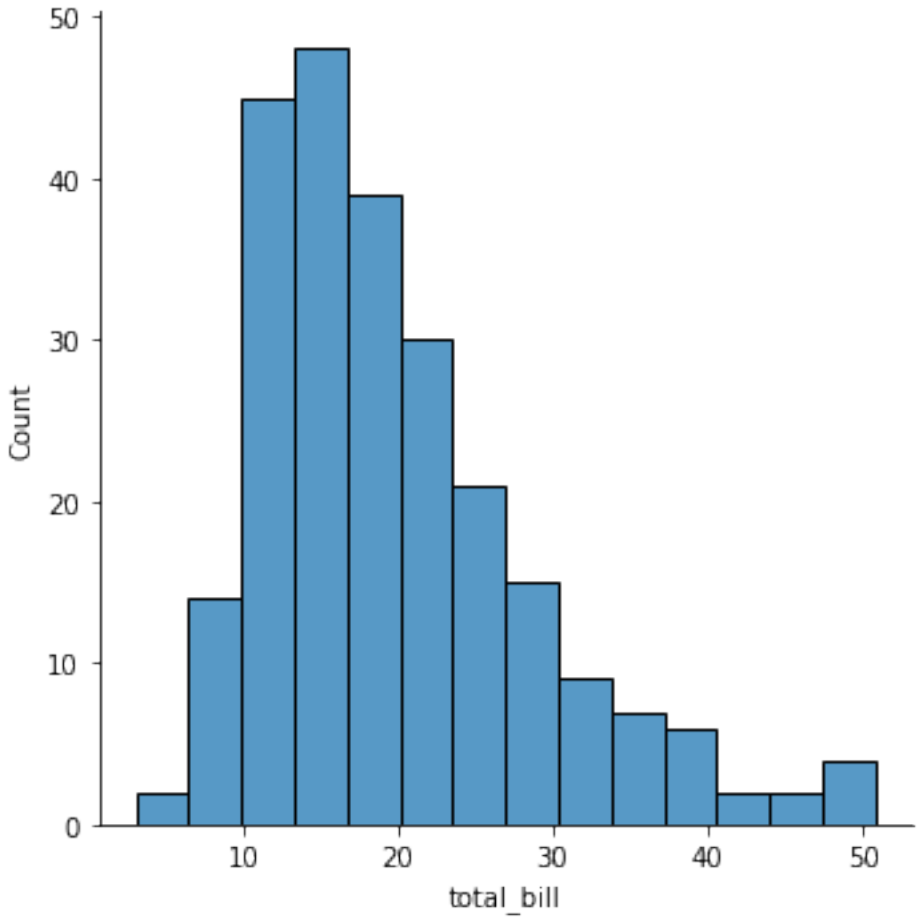
```
# figure level --> displot
# axis level--> histplot --> kdeplot --> rugplot

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

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



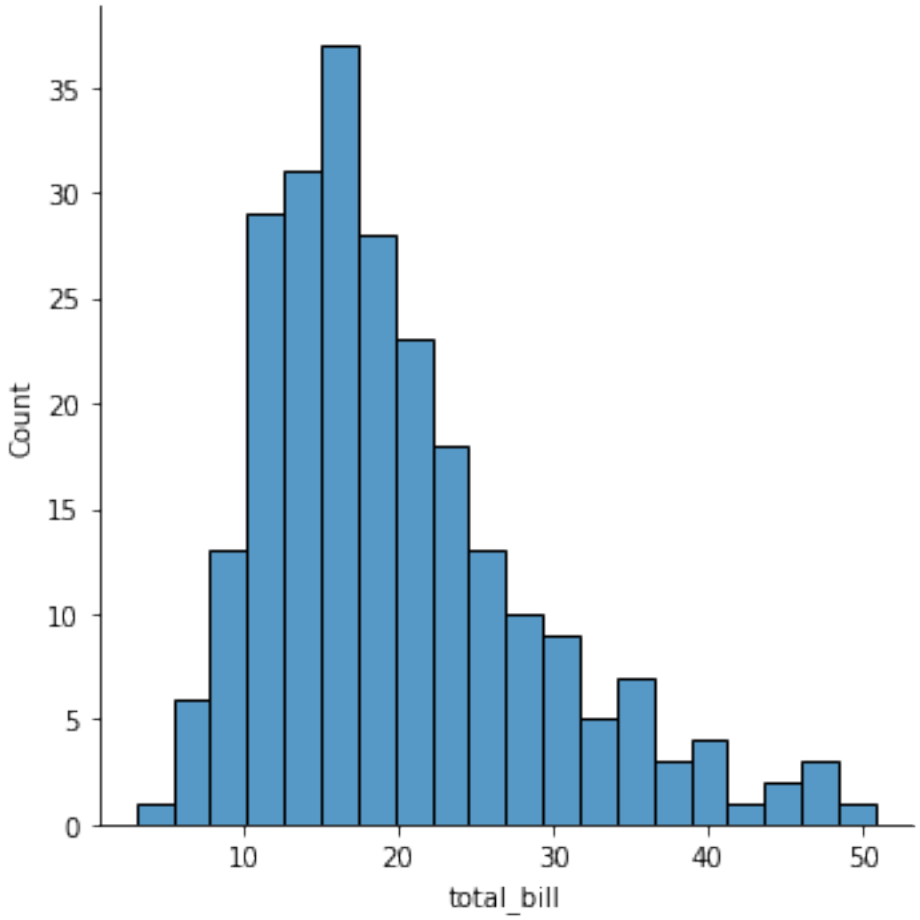

```
sns.displot(data=tips,x='total_bill',kind='hist')
```

```
<seaborn.axisgrid.FacetGrid at 0x28bde27c340>
```



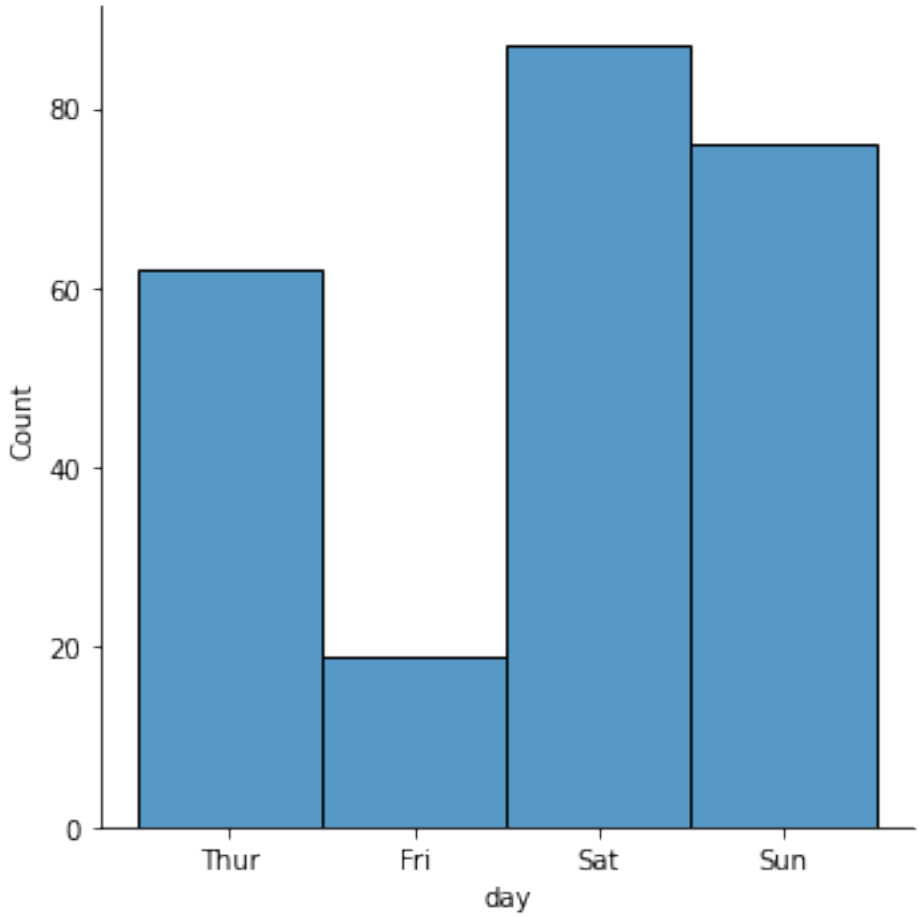

```
# bins parameter
sns.displot(data=tips,x='total_bill',kind='hist',bins=20)
```

```
<seaborn.axisgrid.FacetGrid at 0x28bde909bb0>
```



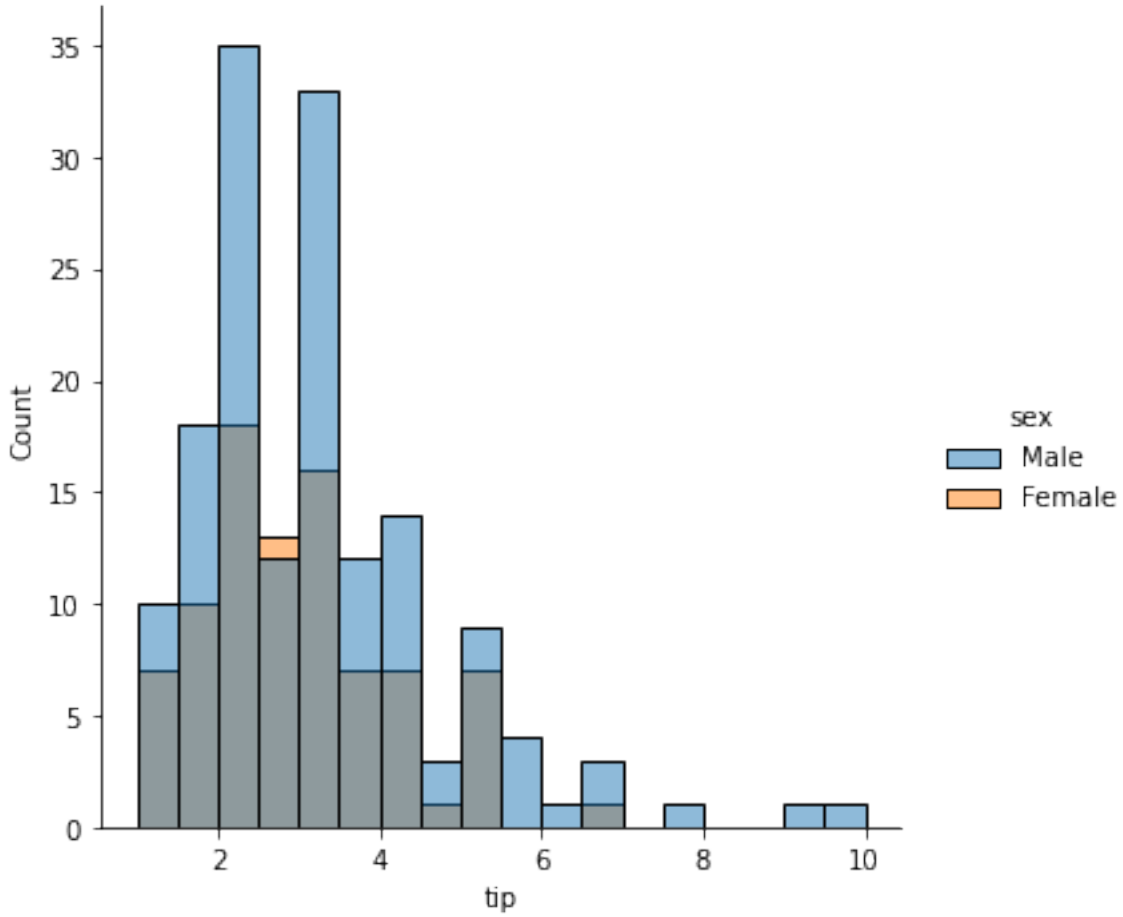

```
# It’s also possible to visualize the distribution of a categorical
variable using the logic of a histogram.
# Discrete bins are automatically set for categorical variables

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

```
<seaborn.axisgrid.FacetGrid at 0x28bde912f40>
```



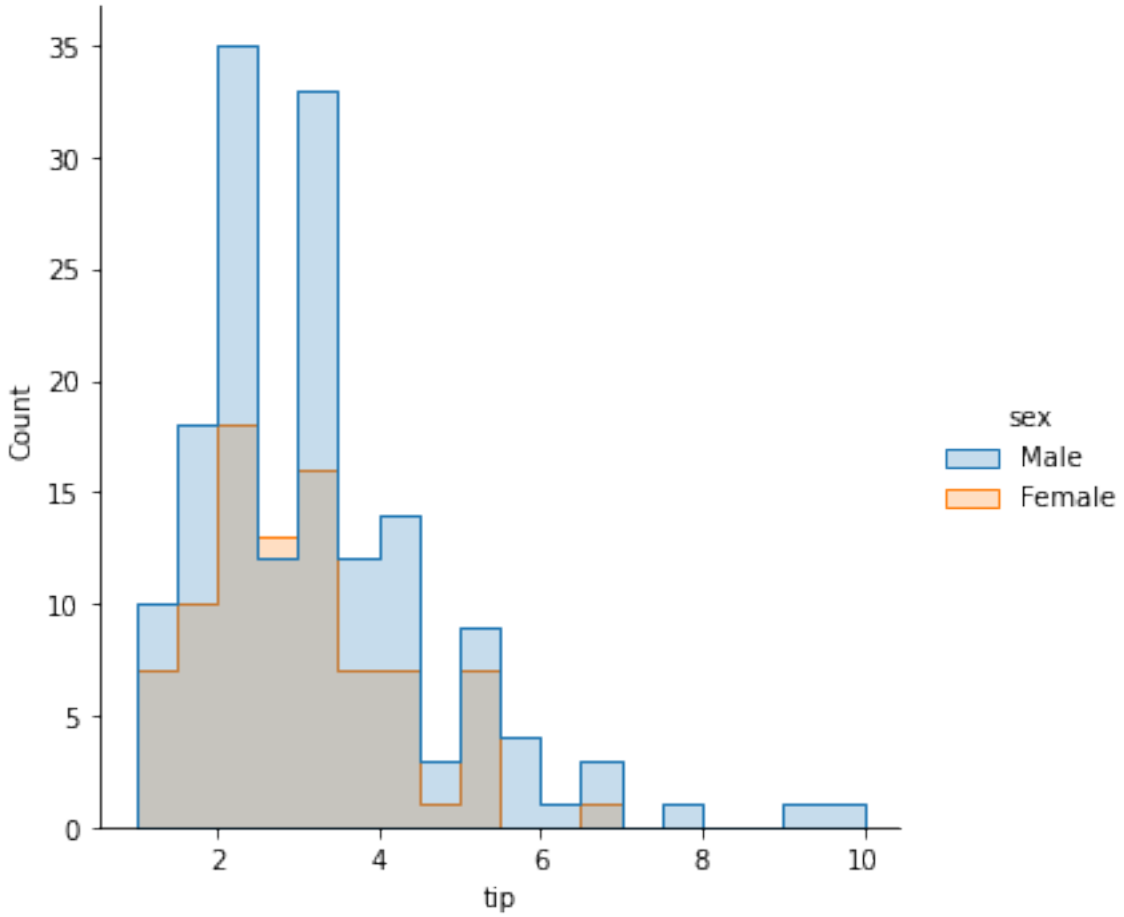

```
# hue
sns.displot(data=tips,x='tip',hue='sex',kind='hist')
```

```
<seaborn.axisgrid.FacetGrid at 0x28bdcac3400>
```




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

```
<seaborn.axisgrid.FacetGrid at 0x28bdfb142b0>
```



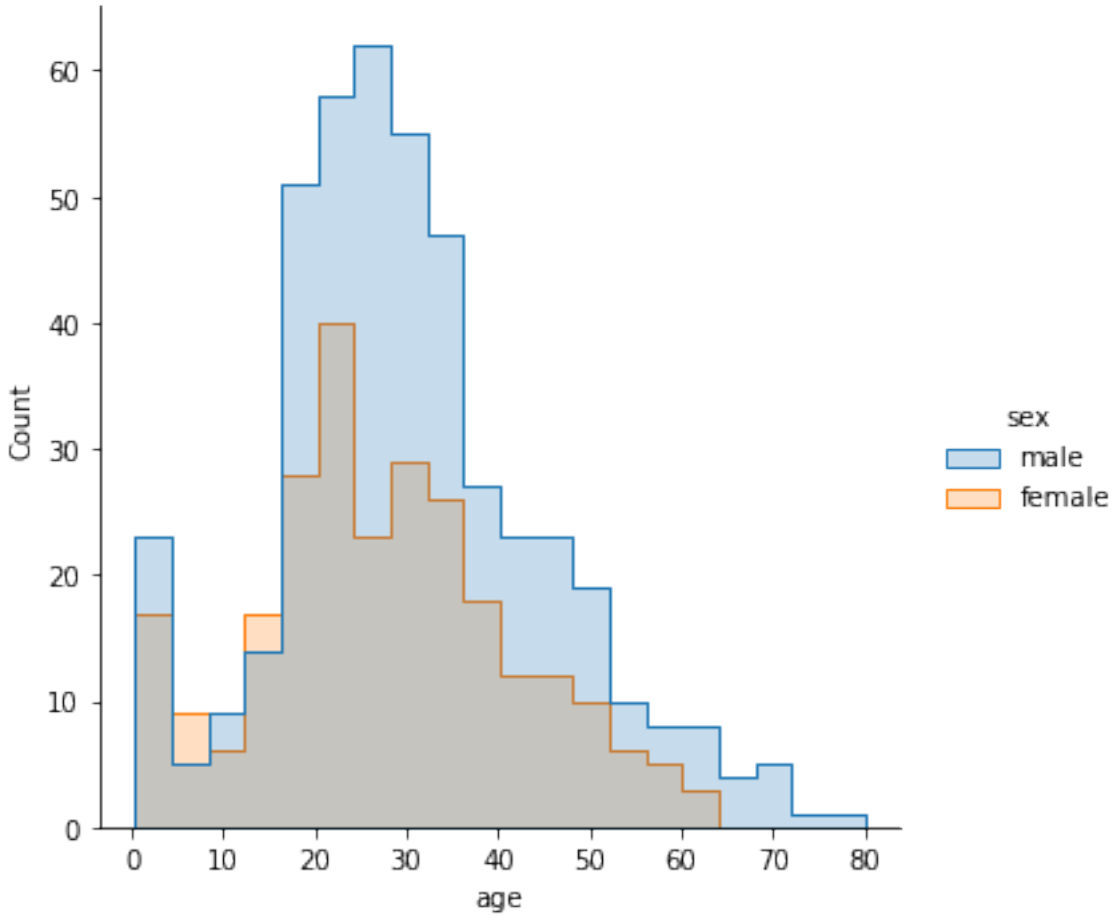

```
titanic=sns.load_dataset('titanic')
titanic
```

```
     survived  pclass     sex   age  sibsp  parch     fare embarked
class  \
0           0       3    male  22.0      1      0   7.2500        S
Third
1           1       1  female  38.0      1      0  71.2833        C
First
2           1       3  female  26.0      0      0   7.9250        S
Third
3           1       1  female  35.0      1      0  53.1000        S
First
4           0       3    male  35.0      0      0   8.0500        S
Third
..        ...     ...     ...   ...    ...    ...      ...       ...
...
886         0       2    male  27.0      0      0  13.0000        S
Second
887         1       1  female  19.0      0      0  30.0000        S
First
```

```
888         0       3  female   NaN      1      2  23.4500         S
Third
889         1       1    male  26.0      0      0  30.0000         C
First
890         0       3    male  32.0      0      0   7.7500         Q
Third

       who  adult_male deck  embark_town alive  alone
0      man        True  NaN  Southampton    no  False
1    woman       False    C    Cherbourg   yes  False
2    woman       False  NaN  Southampton   yes   True
3    woman       False    C  Southampton   yes  False
4      man        True  NaN  Southampton    no   True
..     ...         ...  ...          ...   ...    ...
886    man        True  NaN  Southampton    no   True
887  woman       False    B  Southampton   yes   True
888  woman       False  NaN  Southampton    no  False
889    man        True    C    Cherbourg   yes   True
890    man        True  NaN   Queenstown    no   True

[891 rows x 15 columns]
```

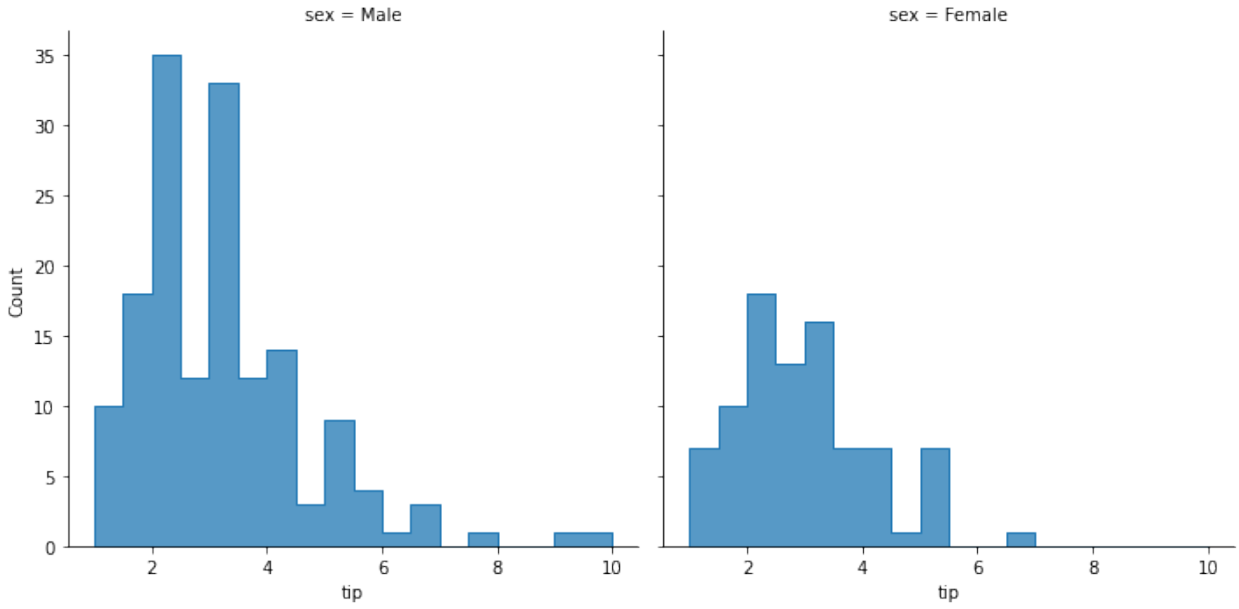
```
# element --> step
sns.displot(data=titanic,x='age',kind='hist',hue='sex',element='step')
```

```
<seaborn.axisgrid.FacetGrid at 0x28bdfba7a60>
```



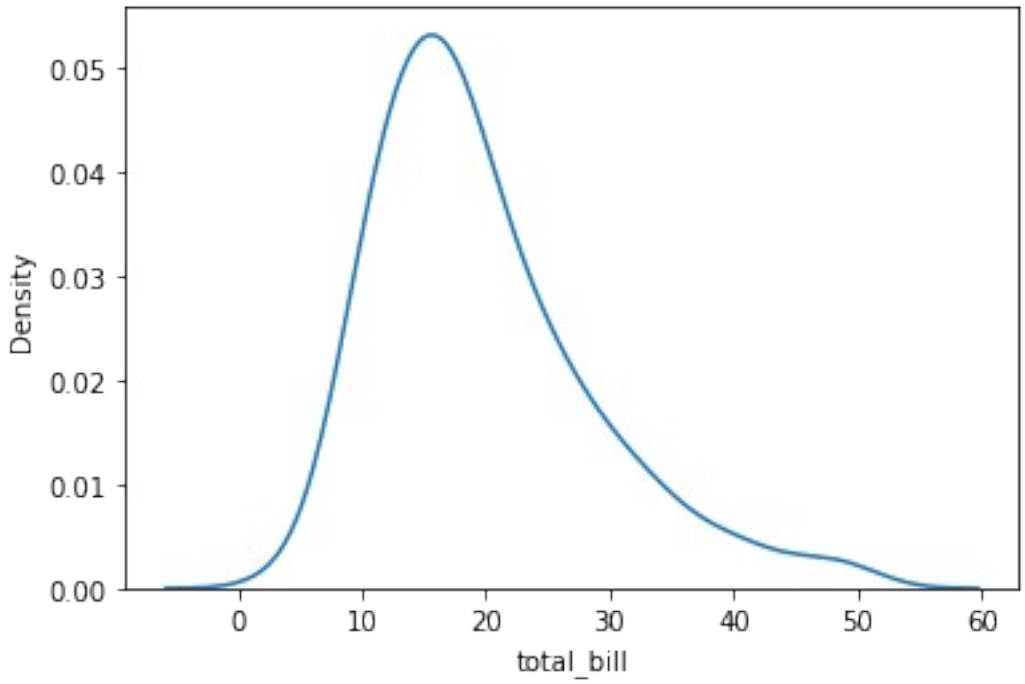

```
# faceting using col and row
sns.displot(data=tips,x='tip',kind='hist',col='sex',element='step')
```

```
<seaborn.axisgrid.FacetGrid at 0x28bdfae8460>
```



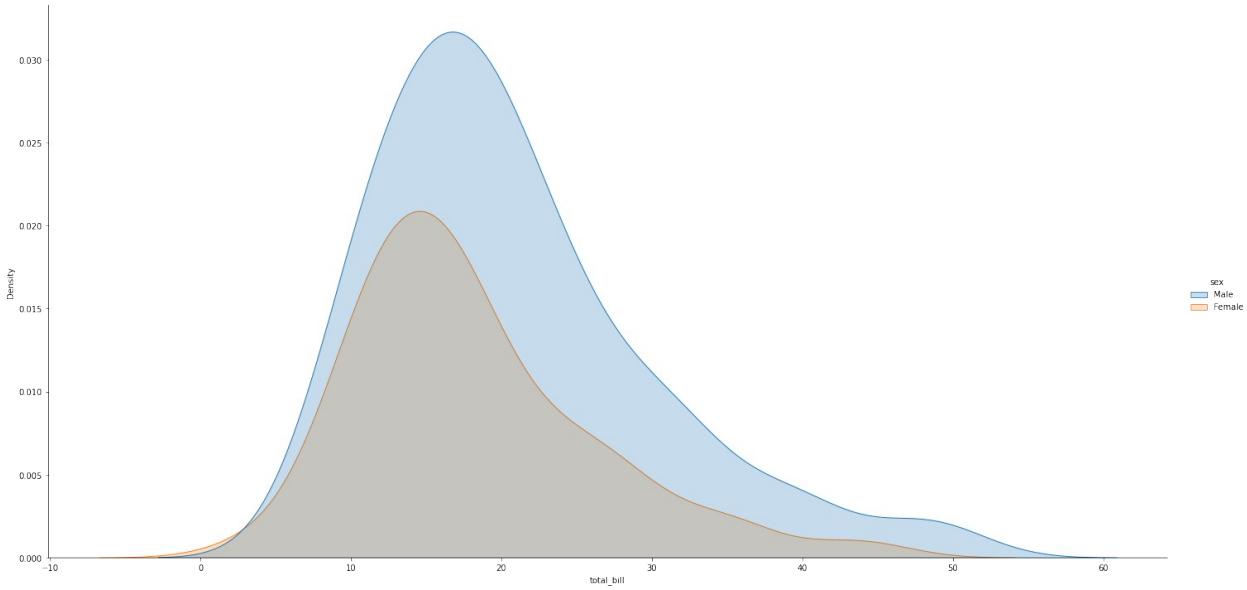

```
# kdeplot
# Rather than using discrete bins, a KDE plot smooths the observations
with a Gaussian kernel, producing a continuous density estimate
sns.kdeplot(data=tips,x='total_bill')
```

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



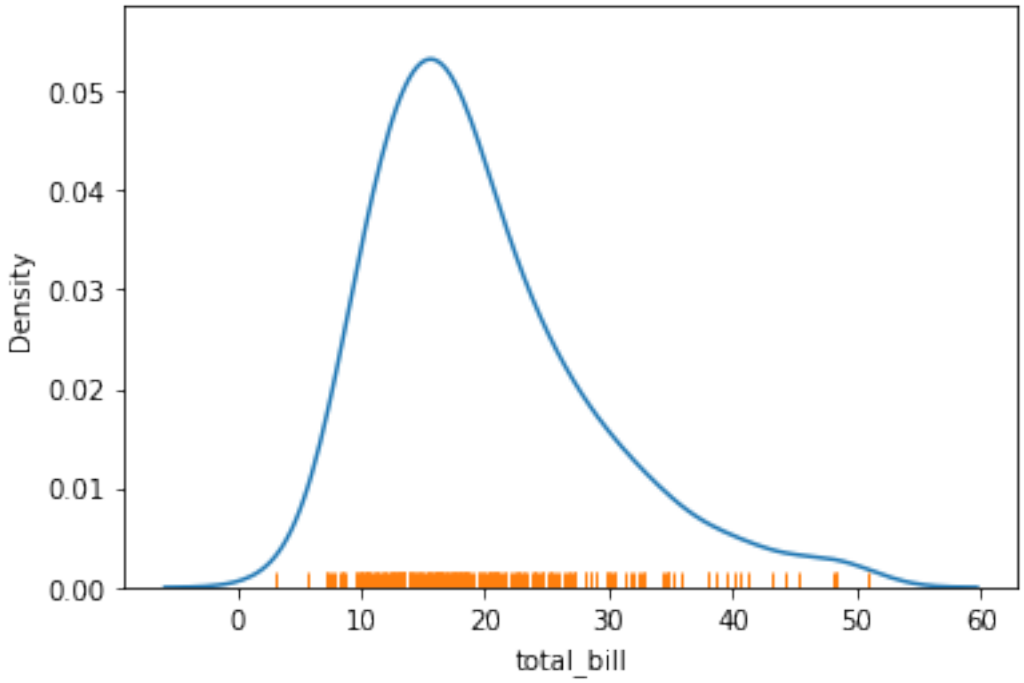
```
# hue -> fill
sns.displot(data=tips,x='total_bill',kind='kde',hue='sex',fill=True,he
ight=10,aspect=2)
```

```
<seaborn.axisgrid.FacetGrid at 0x28bdfcdd100>
```



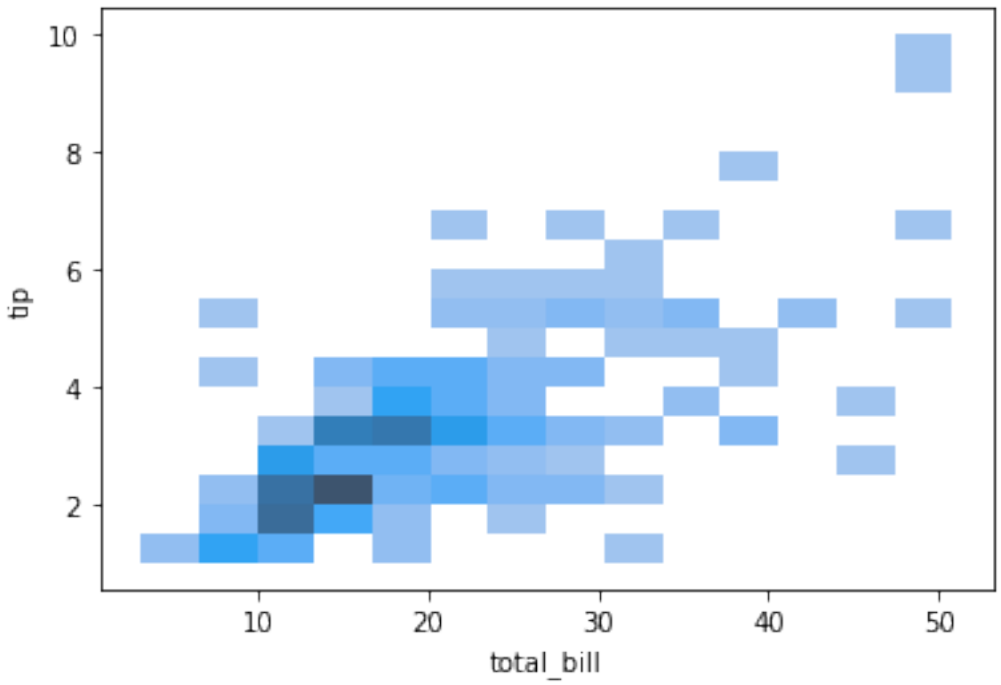

```
# Rugplot

# 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 way.
sns.kdeplot(data=tips,x='total_bill')
sns.rugplot(data=tips,x='total_bill')
```

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



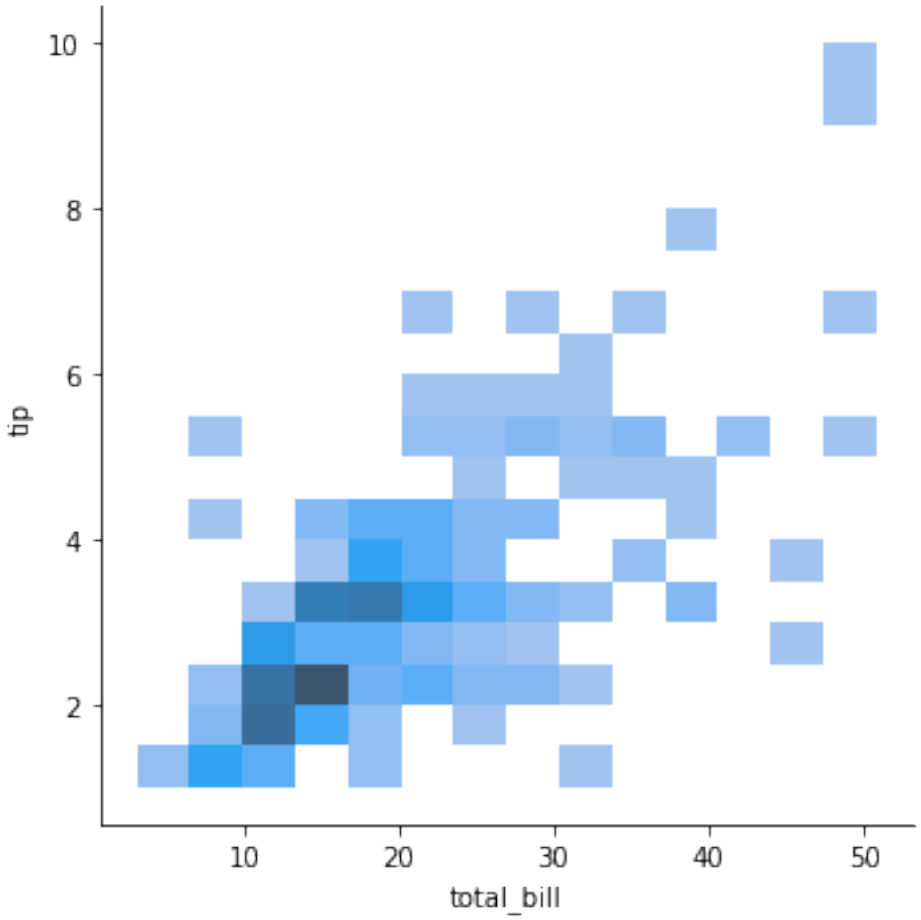

```
# Bivariate histogram
# A bivariate histogram bins the data within rectangles that tile the
plot
# and then shows the count of observations within each rectangle with
the fill color

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

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



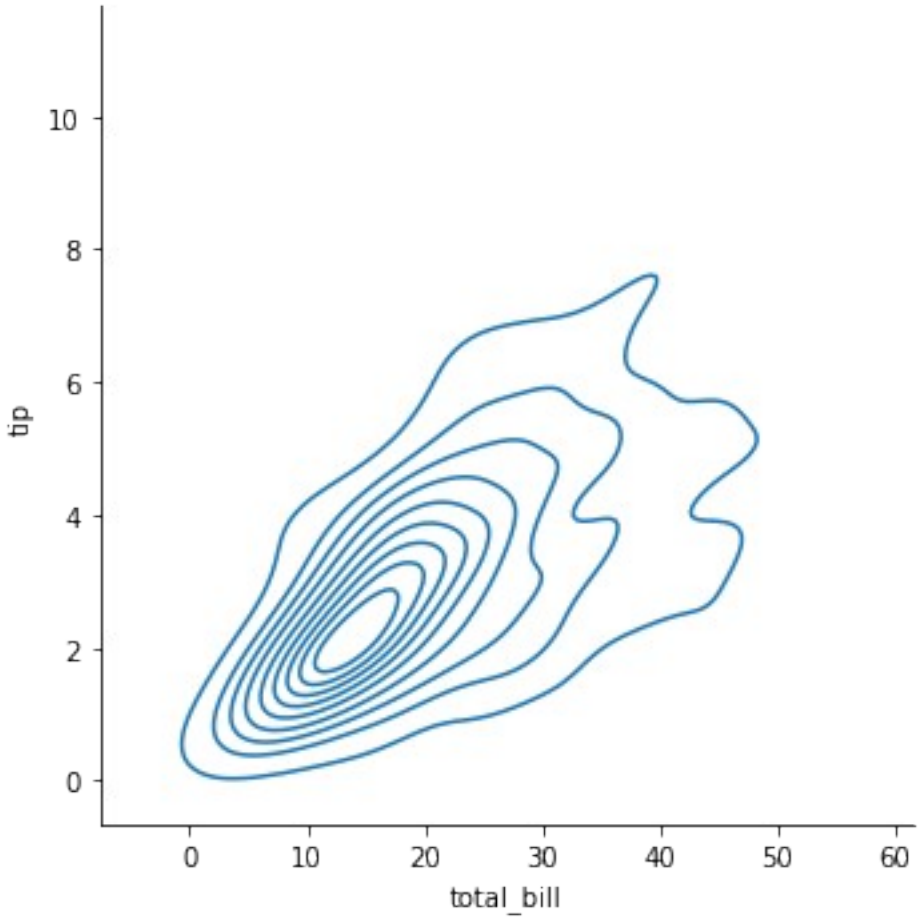

```
sns.displot(data=tips, x='total_bill', y='tip',kind='hist')
```

```
<seaborn.axisgrid.FacetGrid at 0x28be23a0a90>
```




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

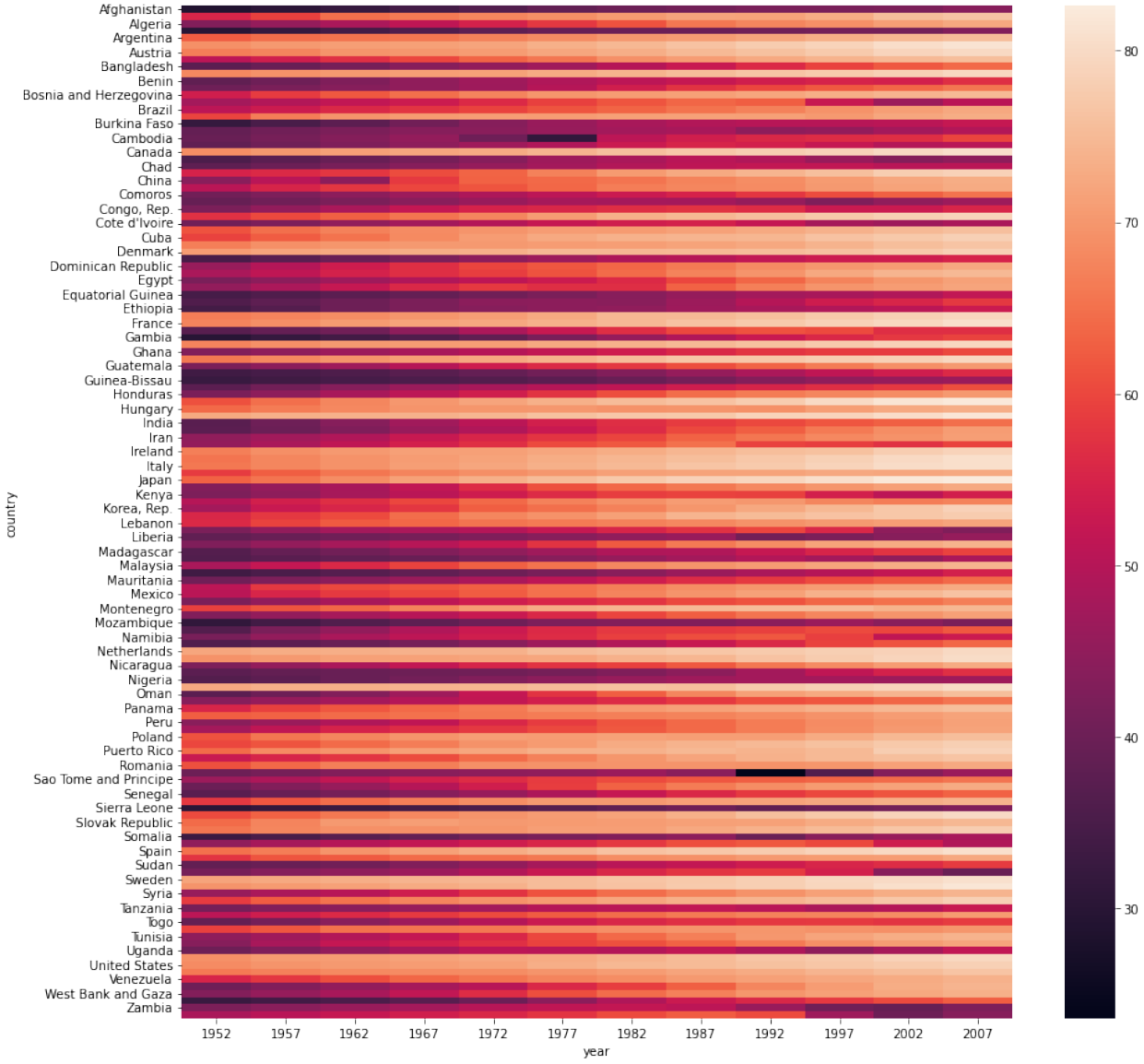
```
<seaborn.axisgrid.FacetGrid at 0x28be22e3df0>
```




## 2. Matrix Plot

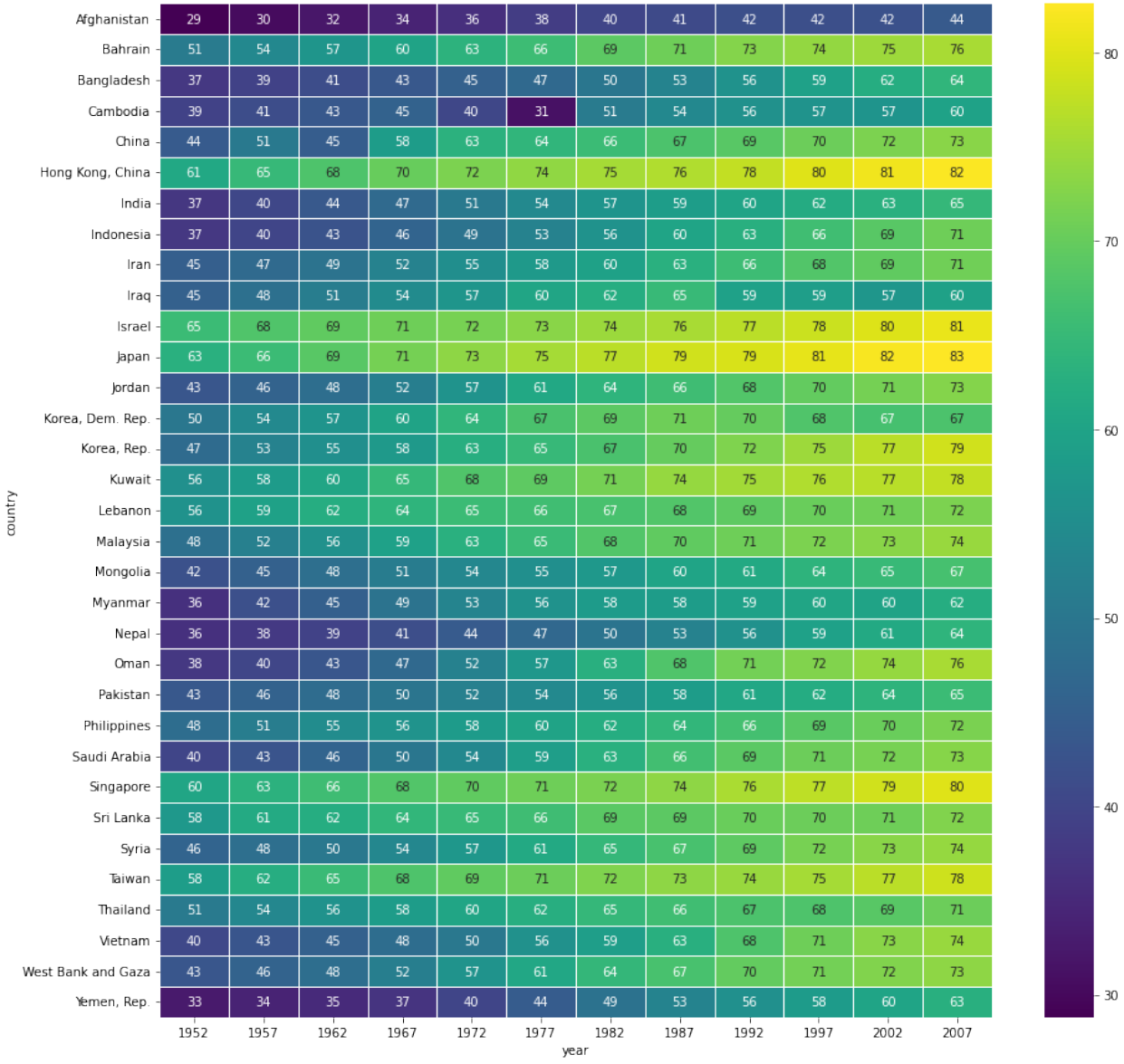
- Heatmap
- Clustermap

```
# 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)

<AxesSubplot:xlabel='year', ylabel='country'>
```




```
# annot
temp_df=gap[gap['continent']=='Asia'].pivot(index='country',columns='y
ear',values='lifeExp')
plt.figure(figsize=(15,15))
sns.heatmap(temp_df,annot=True,linewidths=0.5,cmap='viridis')
```

```
<AxesSubplot:xlabel='year', ylabel='country'>
```



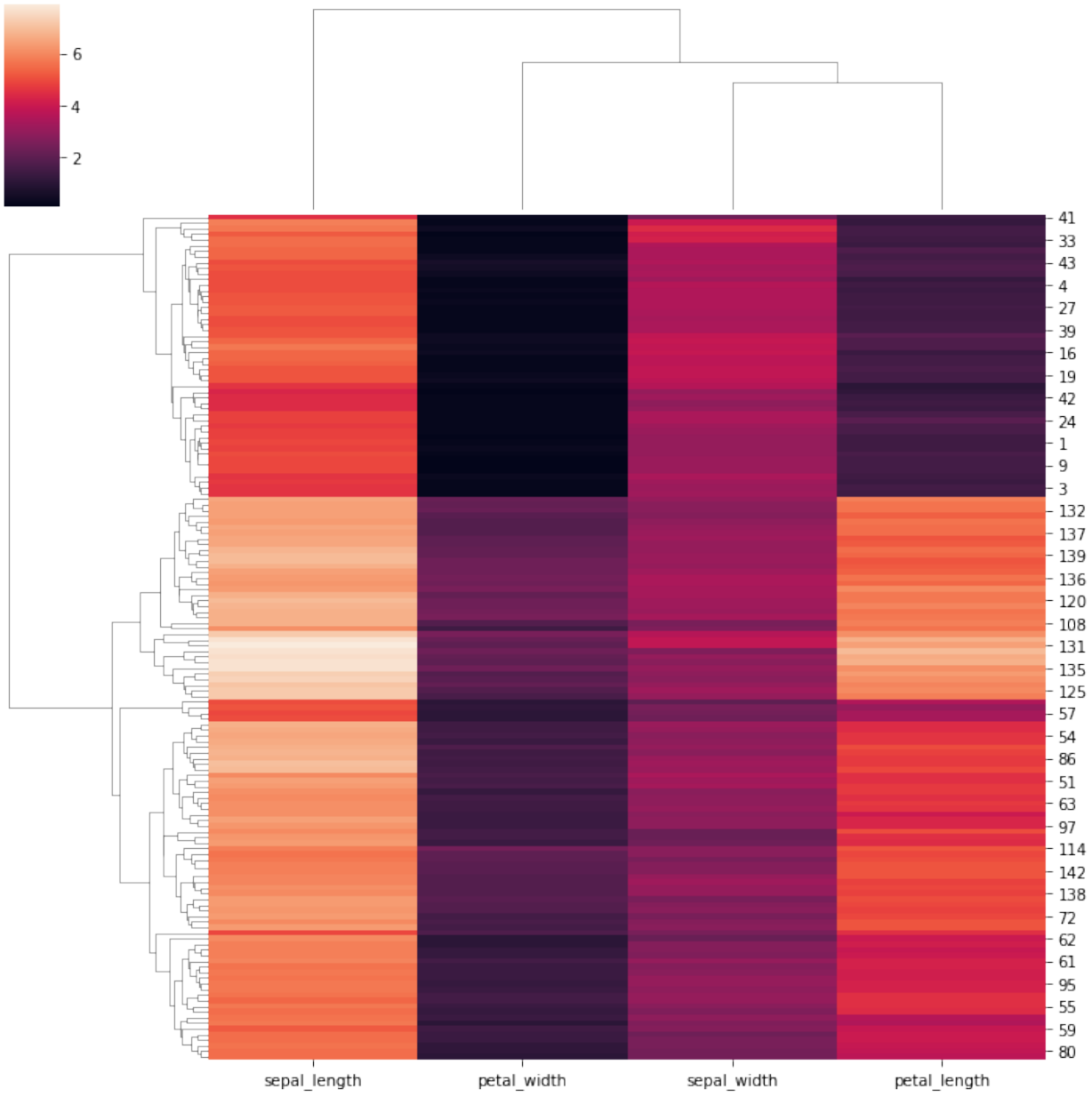

```
# Clustermap

# Plot a matrix dataset as a hierarchically-clustered heatmap.

# This function requires scipy to be available.
iris=px.data.iris()
iris
```

```
     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
..               ...           ...            ...           ...        ...
145              6.7           3.0            5.2           2.3  virginica
146              6.3           2.5            5.0           1.9  virginica
147              6.5           3.0            5.2           2.0  virginica
148              6.2           3.4            5.4           2.3  virginica
149              5.9           3.0            5.1           1.8  virginica


     species_id
0             1
1             1
2             1
3             1
4             1
..          ...
145           3
146           3
147           3
148           3
149           3

[150 rows x 6 columns]
```

```
sns.clustermap(iris.iloc[:,[0,1,2,3]])
```

```
<seaborn.matrix.ClusterGrid at 0x28be42ead90>
```




# Categorical Plots

## Categorical Scatter Plot

- Stripplot
- Swarmplot

## Categorical Distribution Plots

- Boxplot
- Violinplot

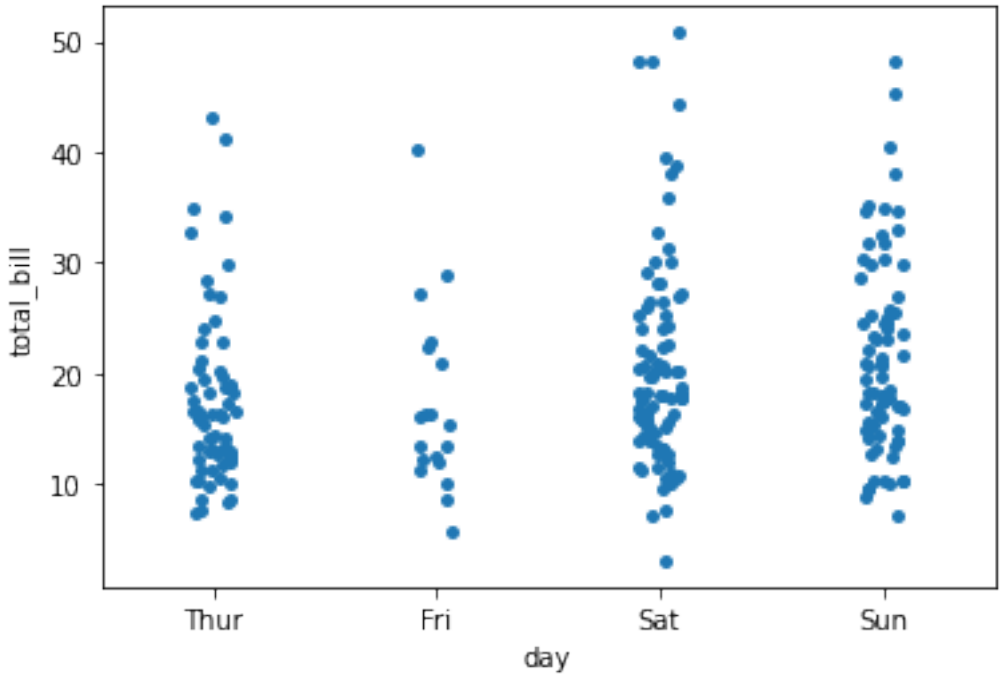
## Categorical Estimate Plot -> for central tendency

- Barplot
- Pointplot
- Countplot

## Figure level function -> `catplot`

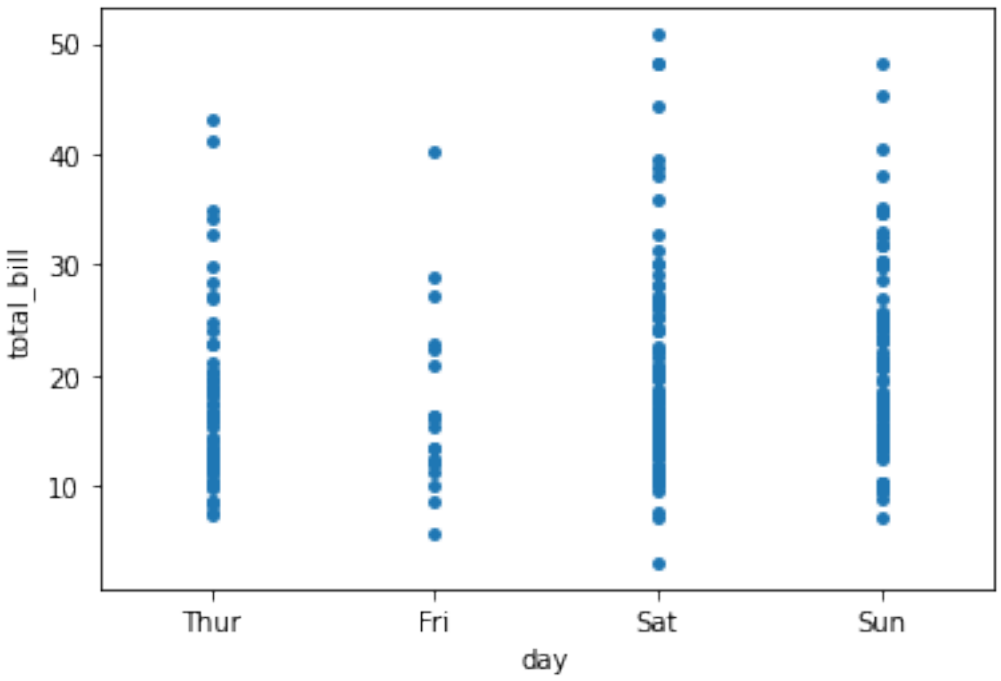
```
# strip plot
# axes level
sns.stripplot(data=tips,x='day',y='total_bill')
```

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



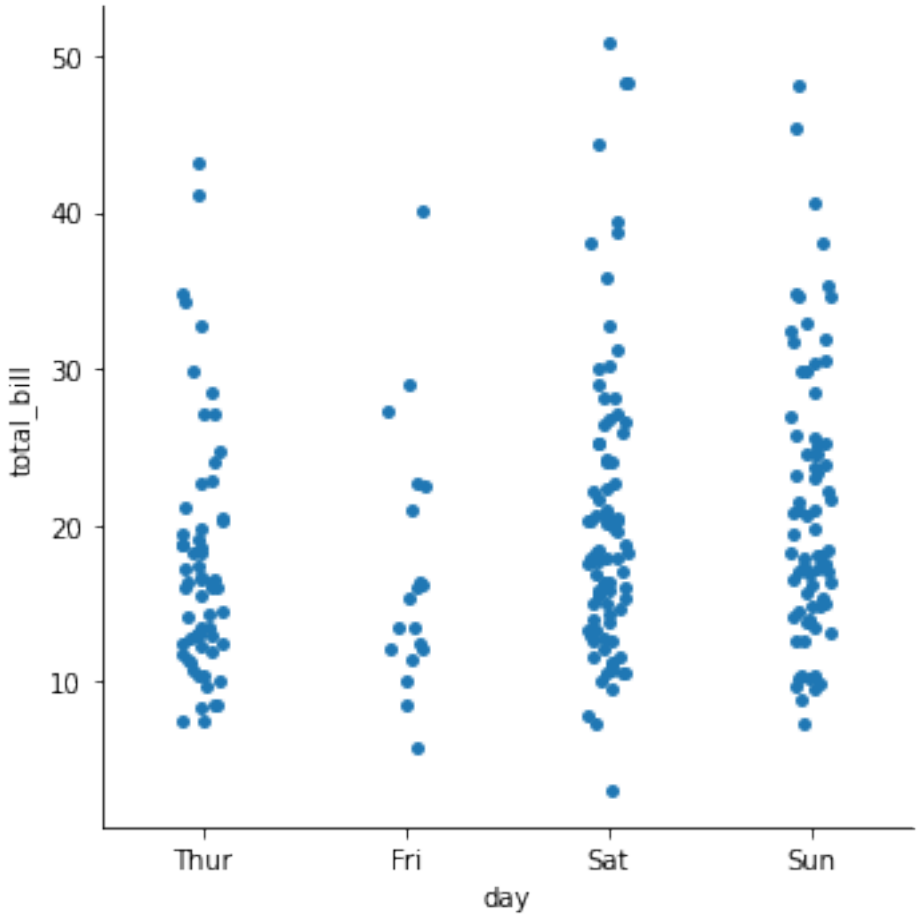

```
sns.stripplot(data=tips,x='day',y='total_bill',jitter=False)
```

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



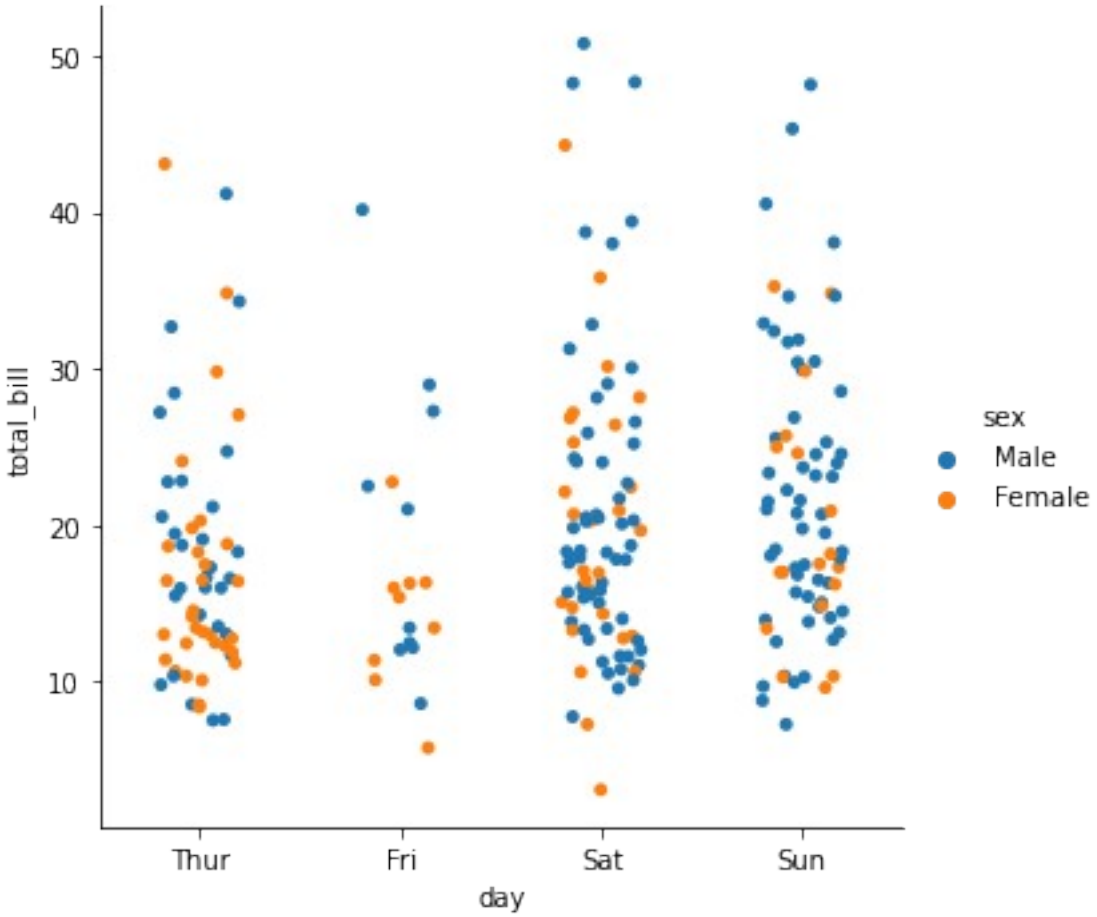

```
# using catplot
# figure level
sns.catplot(data=tips,x='day',y='total_bill',kind='strip')
```

```
<seaborn.axisgrid.FacetGrid at 0x28bde1856a0>
```



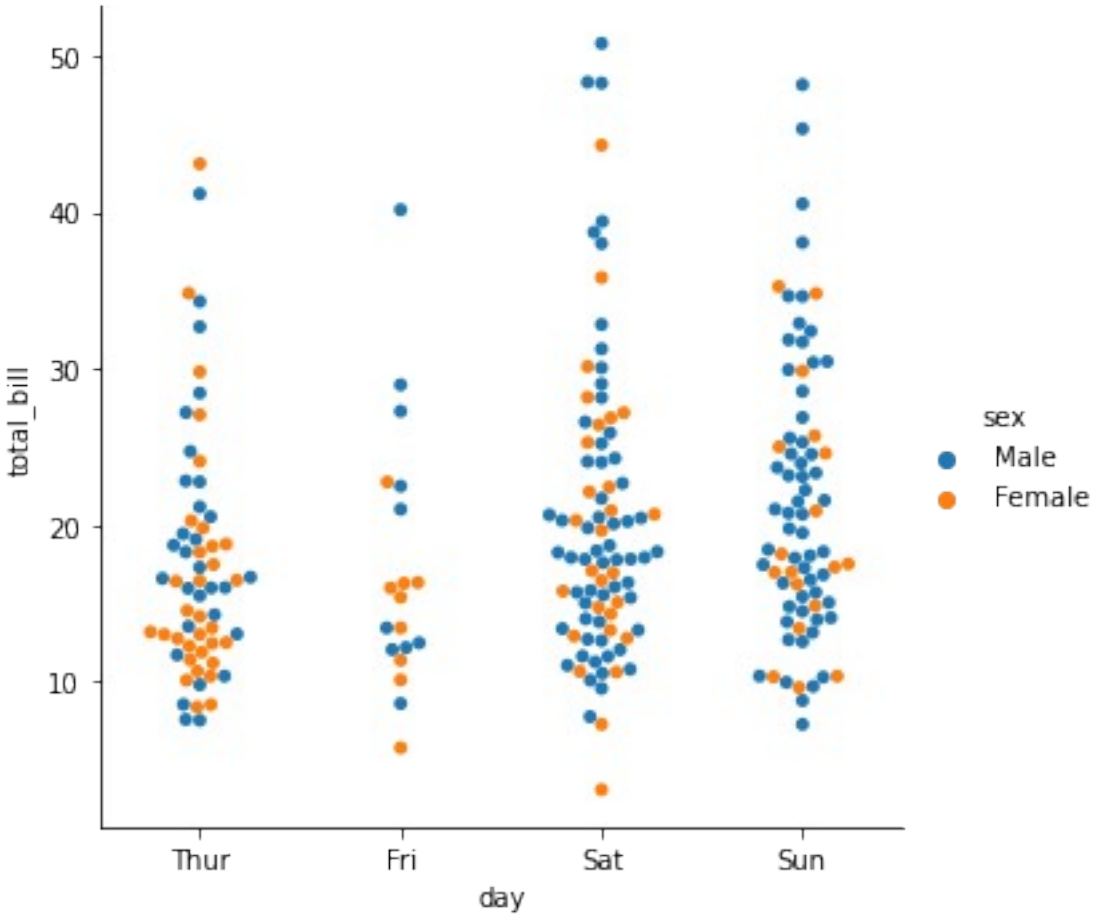

```
sns.catplot(data=tips,x='day',y='total_bill',kind='strip',jitter=0.2,h
ue='sex')
```

```
<seaborn.axisgrid.FacetGrid at 0x28be4681a00>
```



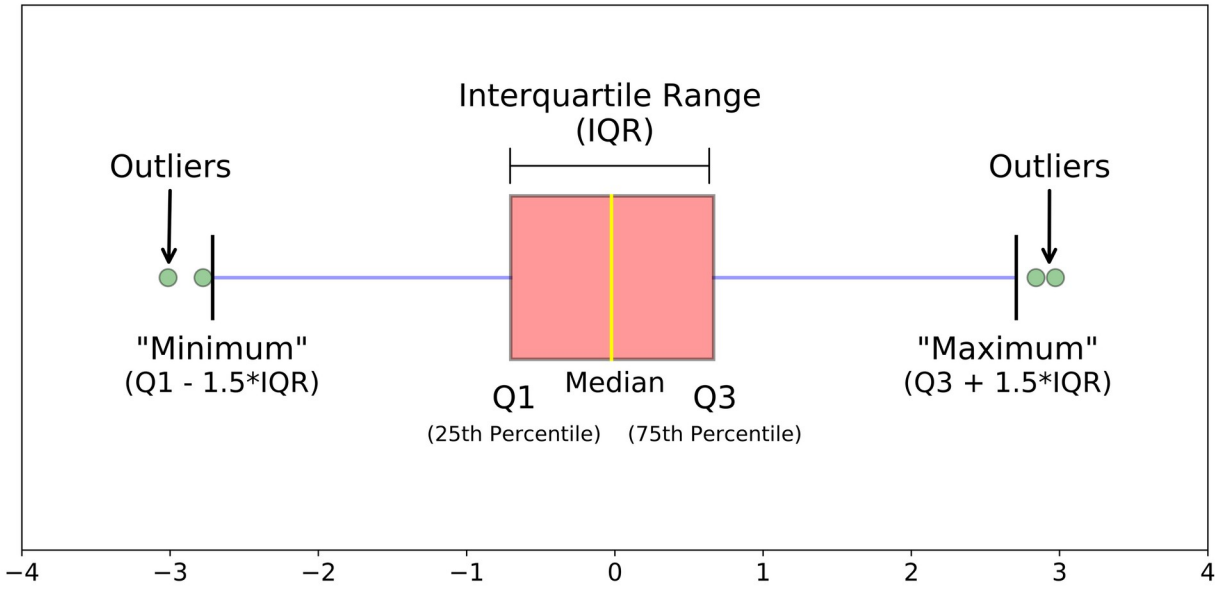

```
# swarmplot
sns.catplot(data=tips,x='day',y='total_bill',kind='swarm',hue='sex')
```

```
<seaborn.axisgrid.FacetGrid at 0x28be4be5700>
```



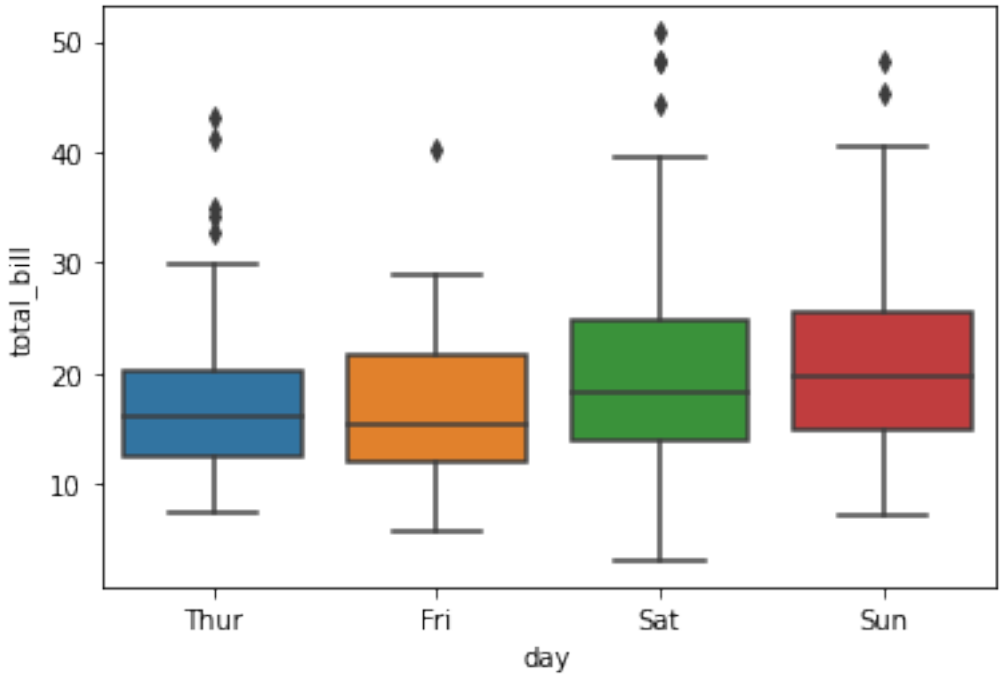

# Boxplot

A boxplot is a standardized way of displaying the distribution of data based on a five number summary (“minimum”, first quartile [Q1], median, third quartile [Q3] and “maximum”). It can tell you about your outliers and what their values are. Boxplots can also tell you if your data is symmetrical, how tightly your data is grouped and if and how your data is skewed.



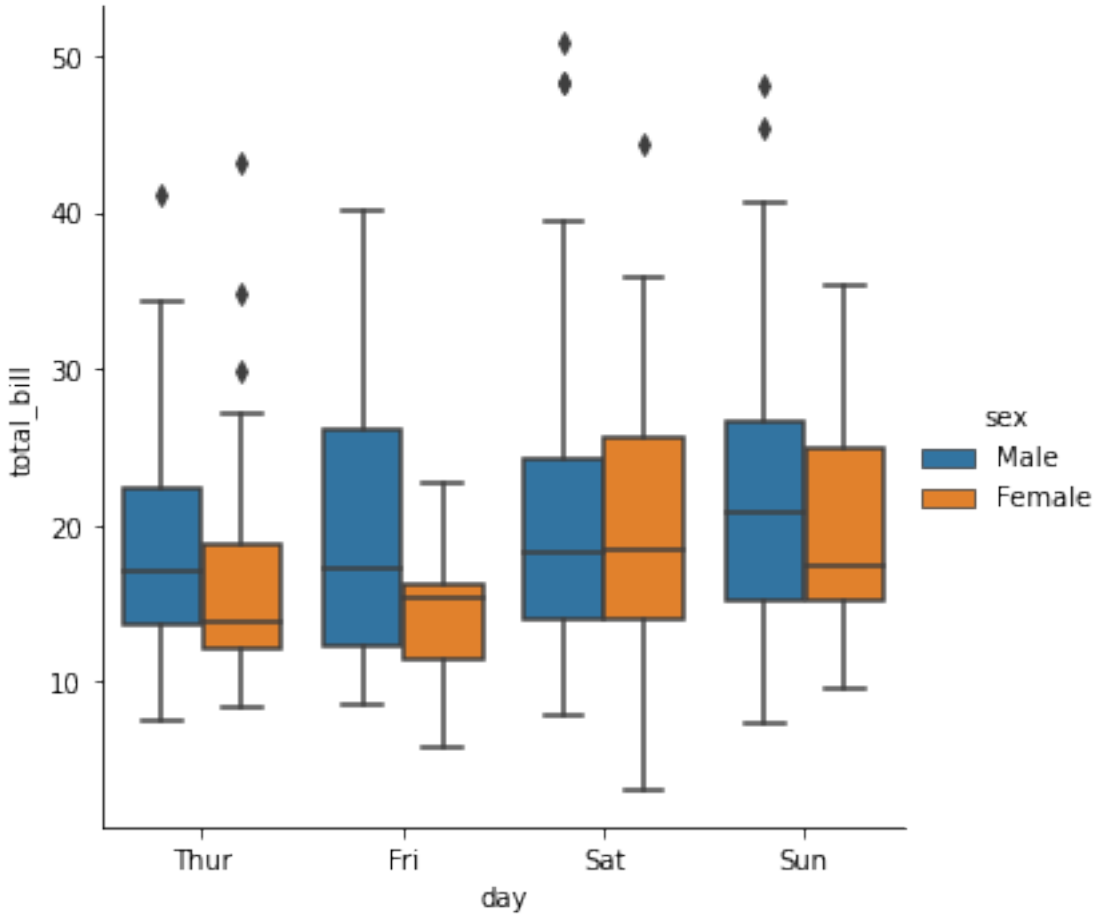

```
# Box plot
sns.boxplot(data=tips,x='day',y='total_bill')
```

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



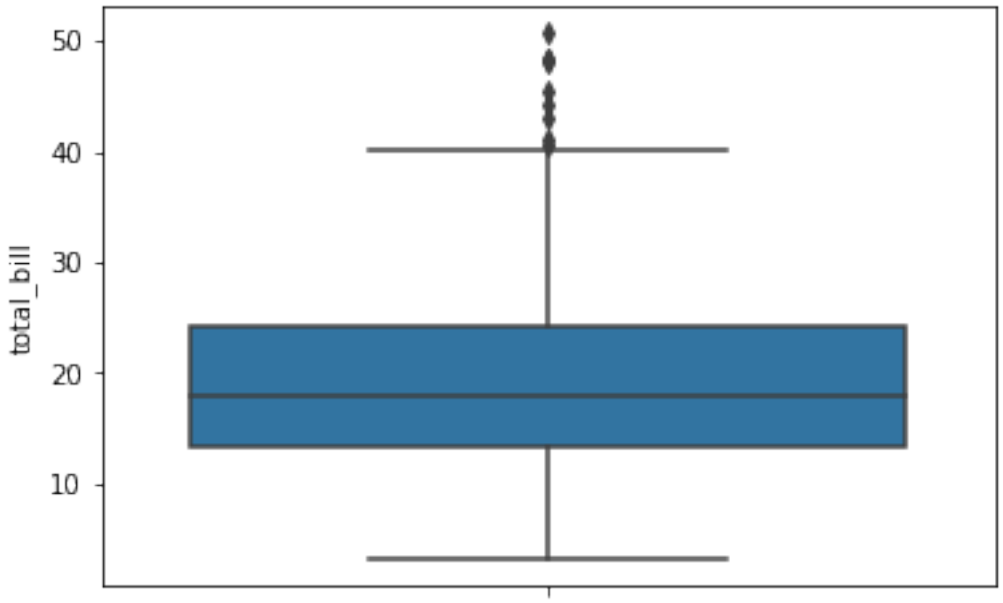

```
sns.catplot(data=tips,x='day',y='total_bill',kind='box',hue='sex')
```

```
<seaborn.axisgrid.FacetGrid at 0x28be4c60910>
```




```
# single boxplot --> numerical col
sns.boxplot(data=tips,y='total_bill')
```

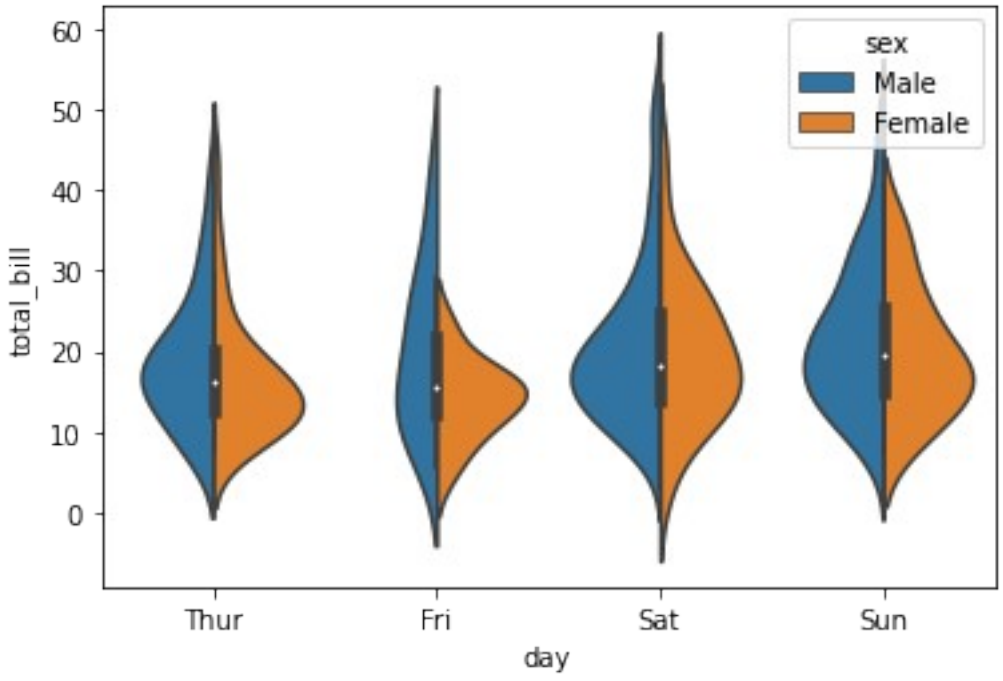
```
<AxesSubplot:ylabel='total_bill'>
```




Violinplot = (boxplot + kdeplot)

```
# violinplot
sns.violinplot(data=tips,x='day',y='total_bill',hue='sex',split=True)
```

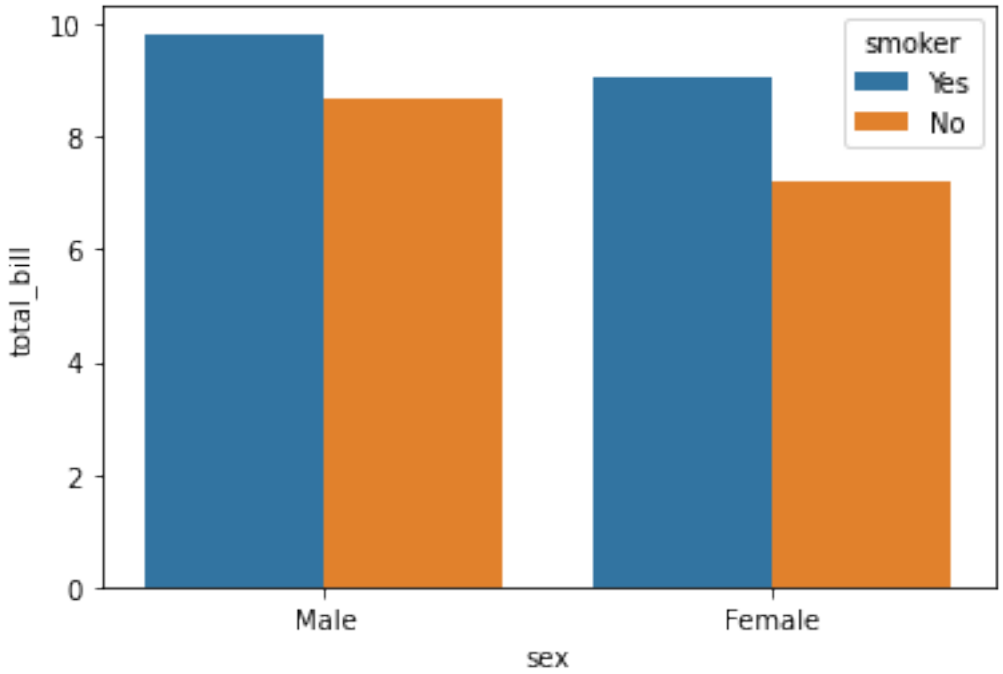
```
<AxesSubplot:xlabel='day', ylabel='total_bill'>
```




```
# bar plot
import numpy as np
```

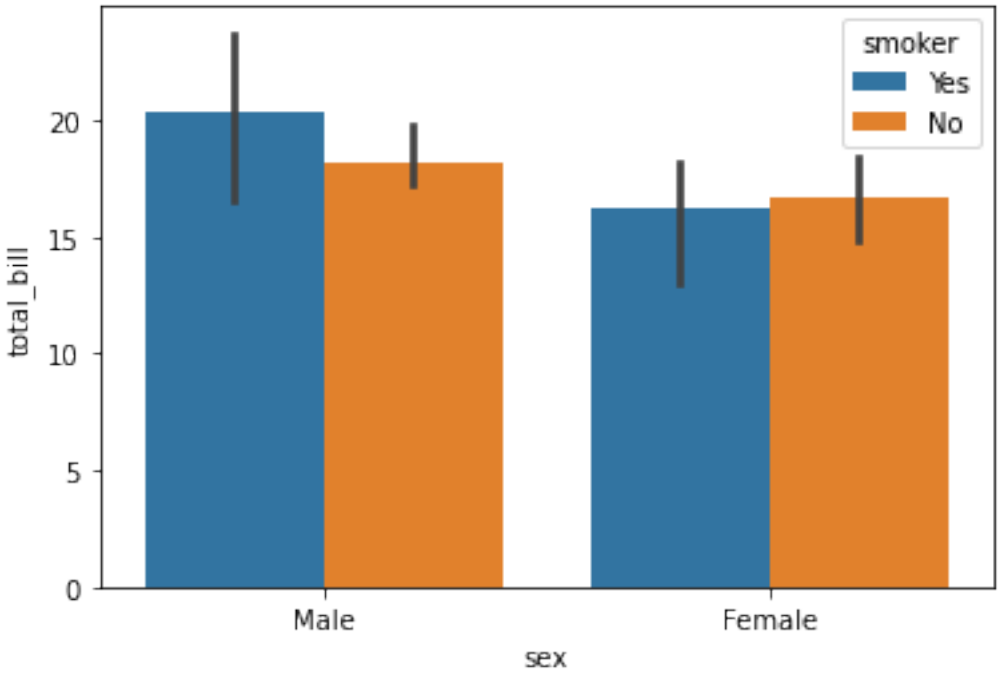
```
sns.barplot(data=tips,x='sex',y='total_bill',hue='smoker',estimator=np
.std,errorbar=None)
```

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




```
# bar plot
sns.barplot(data=tips,x='sex',y='total_bill',hue='smoker',estimator=np
.median)
```

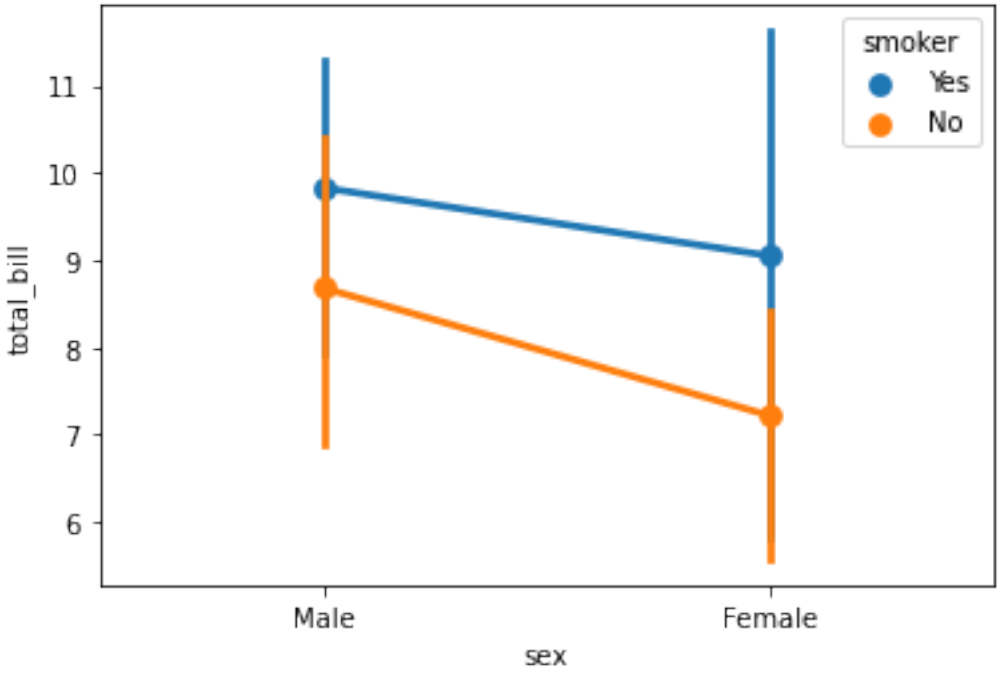
```
<AxesSubplot:xlabel='sex', ylabel='total_bill'>
```




When there are multiple observations in each category, it also uses bootstrapping to compute a confidence interval around the estimate, which is plotted using error bars

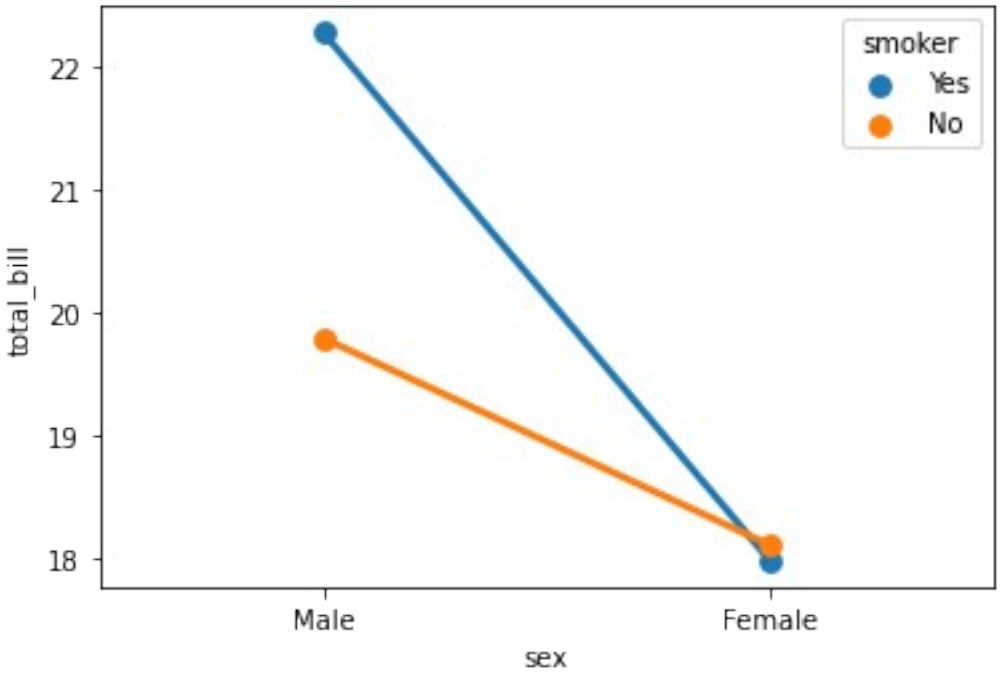
```
# point bar
sns.pointplot(data=tips,x='sex',y='total_bill',hue='smoker',estimator=
np.std)
```

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



```
sns.pointplot(data=tips, x='sex',
y='total_bill',hue='smoker',errorbar=None)
```

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

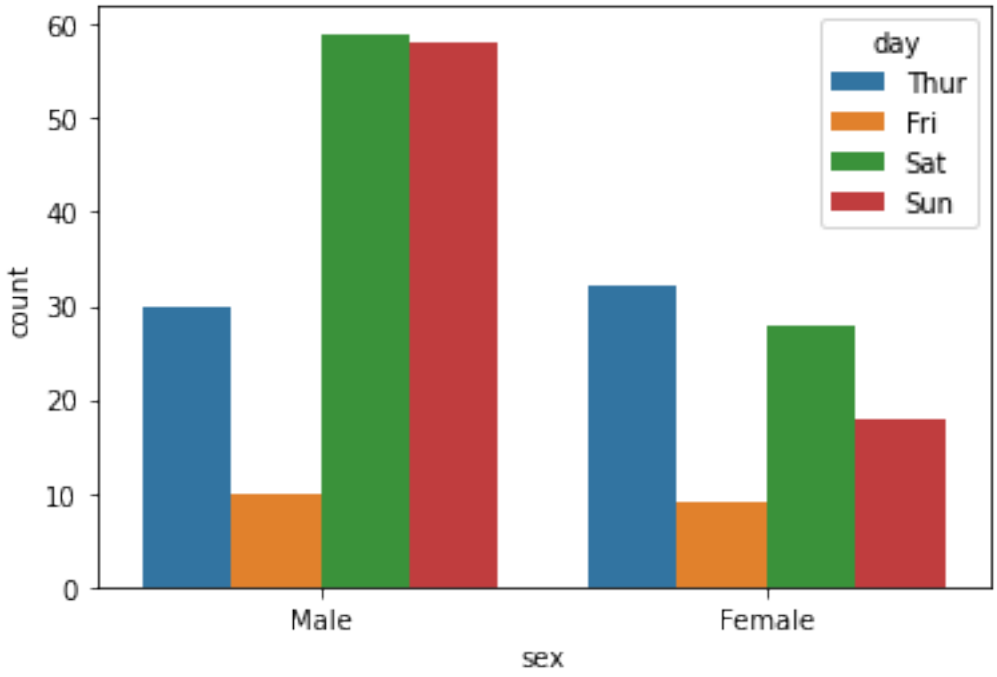



## count plot

A special case for the bar plot is when you want to show the number of observations in each category rather than computing a statistic for a second variable. This is similar to a histogram over a categorical, rather than quantitative, variable

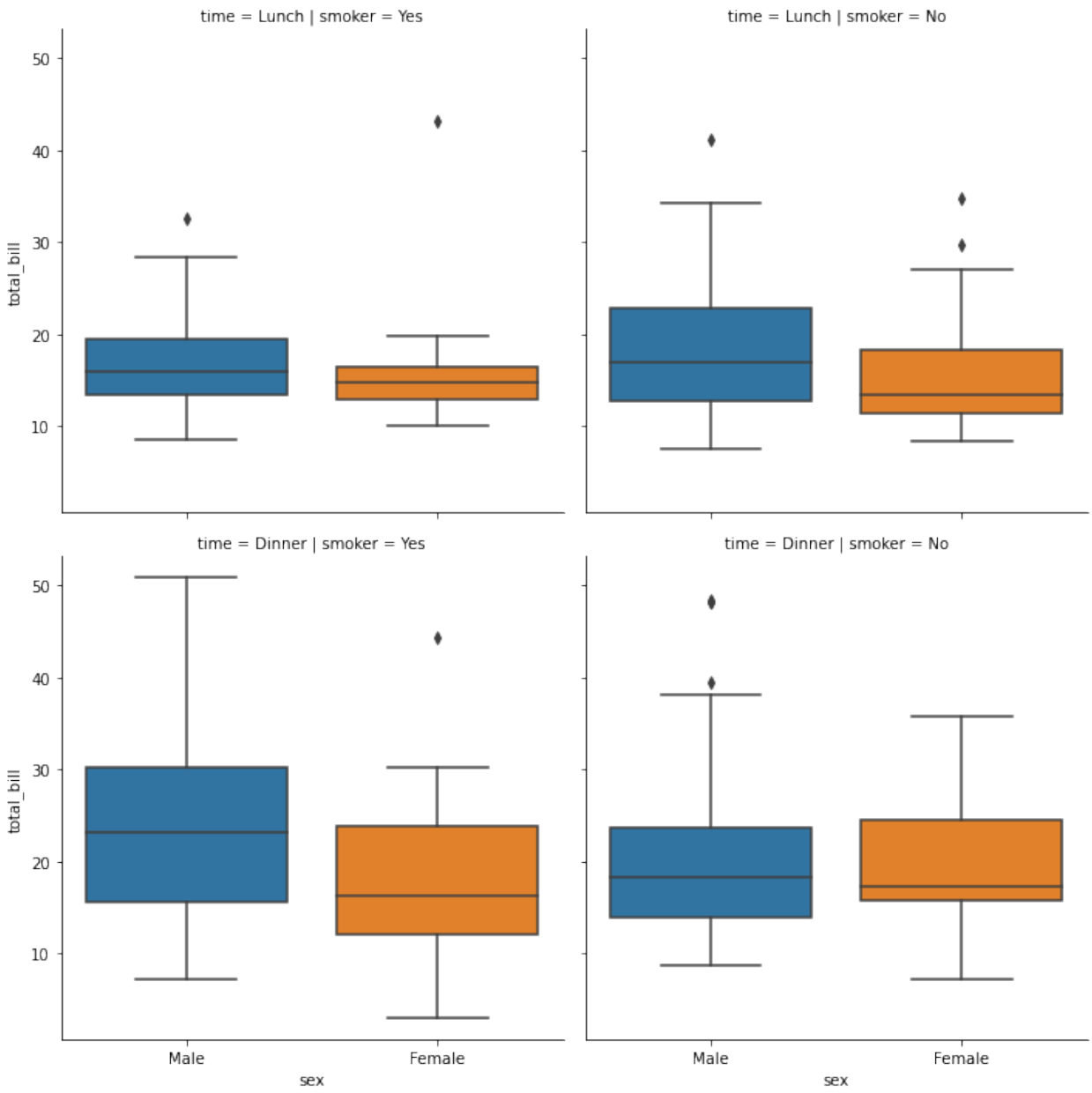
```
# count plot
sns.countplot(data=tips,x='sex',hue='day')
```

```
<AxesSubplot:xlabel='sex', ylabel='count'>
```




```
# faceting using catplot
sns.catplot(data=tips,
x='sex',y='total_bill',col='smoker',kind='box',row='time')
```

```
<seaborn.axisgrid.FacetGrid at 0x28be6277730>
```



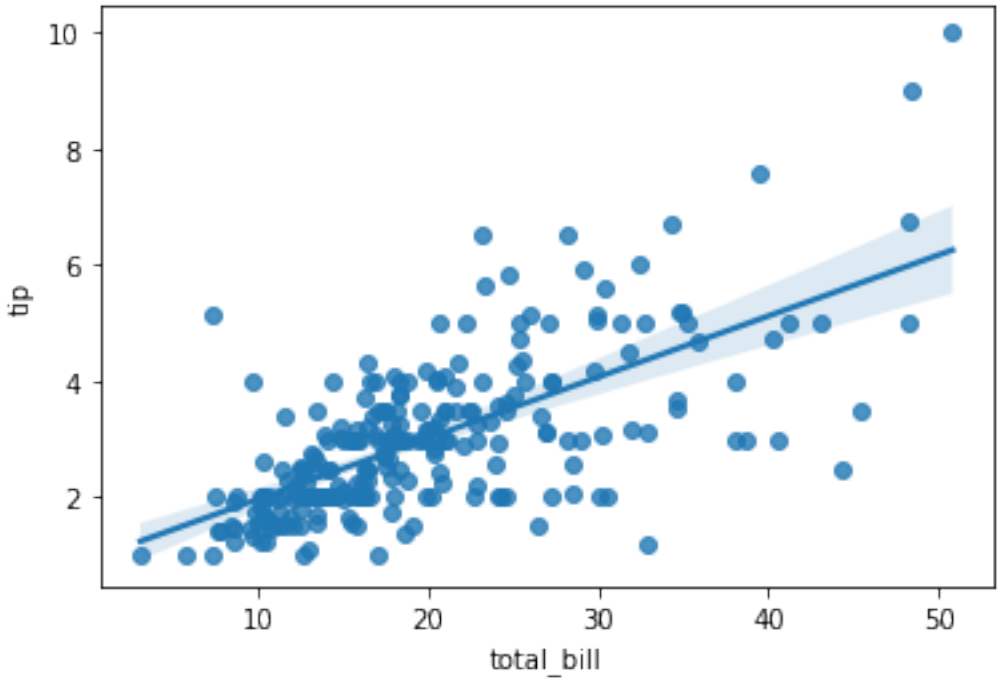

## Regression Plots

- regplot
- lmplot

In the simplest invocation, both functions draw a scatterplot of two variables, x and y, and then fit the regression model y ~ x and plot the resulting regression line and a 95% confidence interval for that regression.

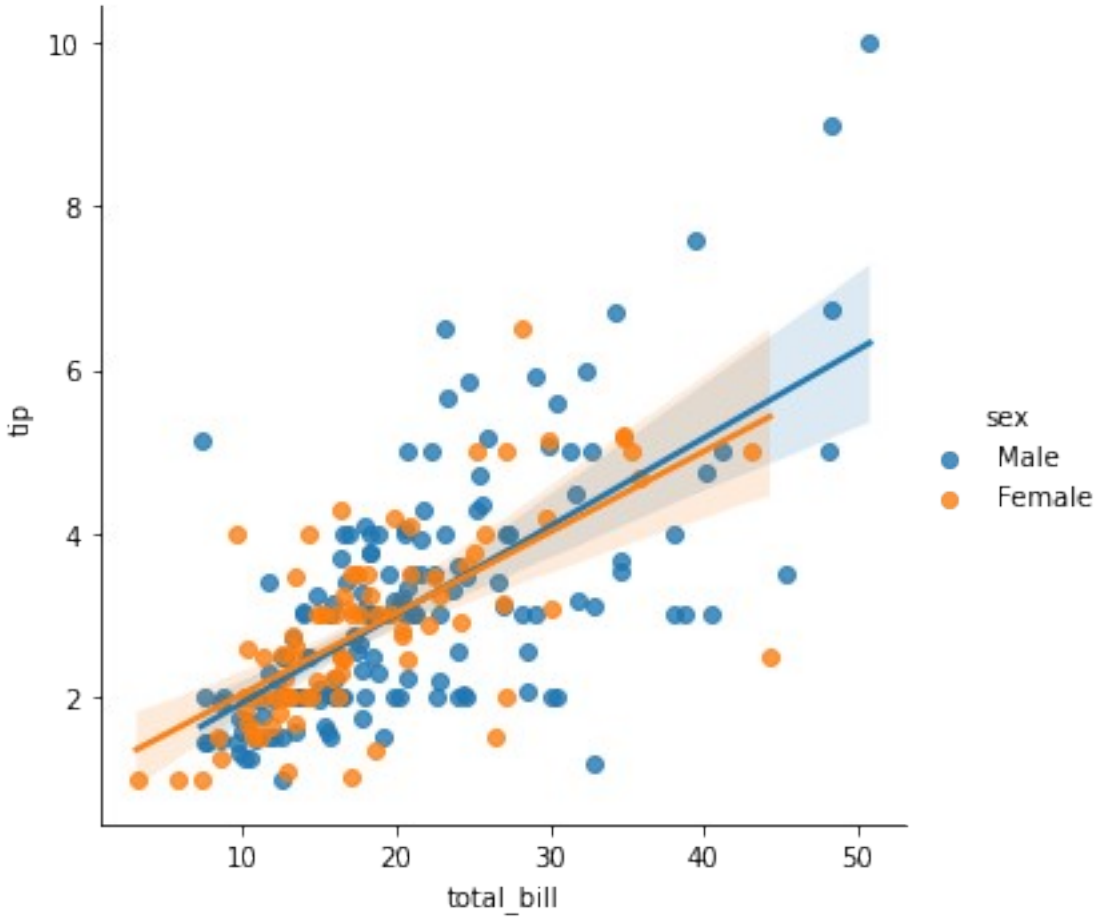
```
# axes level
# hue parameter is not available
sns.regplot(data=tips,x='total_bill',y='tip')
```

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



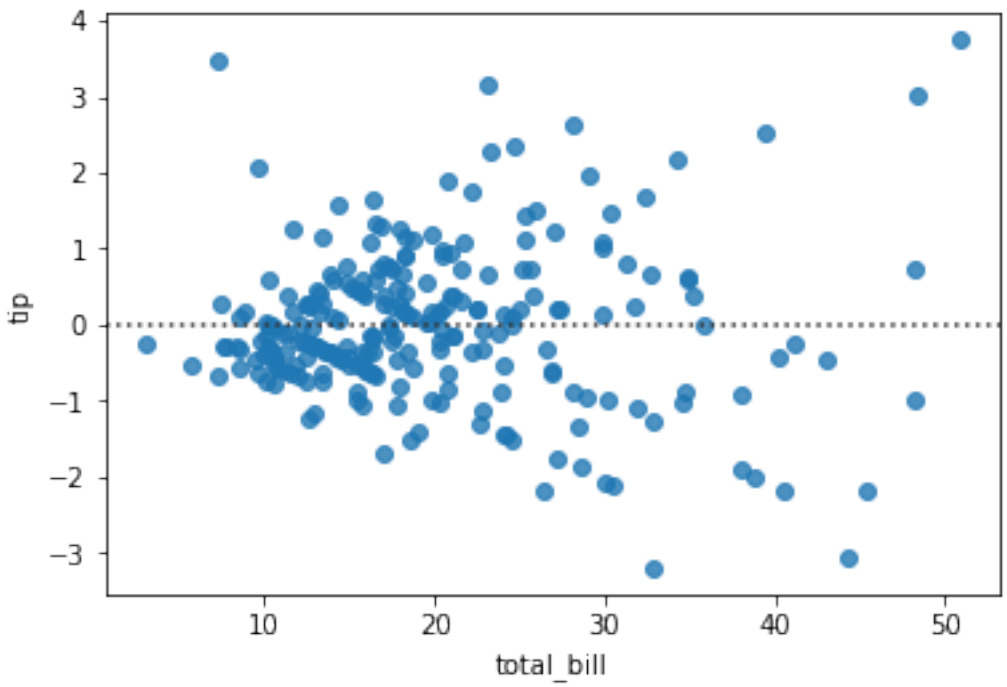

```
# figure level
sns.lmplot(data=tips,x='total_bill',y='tip',hue='sex')
```

```
<seaborn.axisgrid.FacetGrid at 0x28be693eb50>
```




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

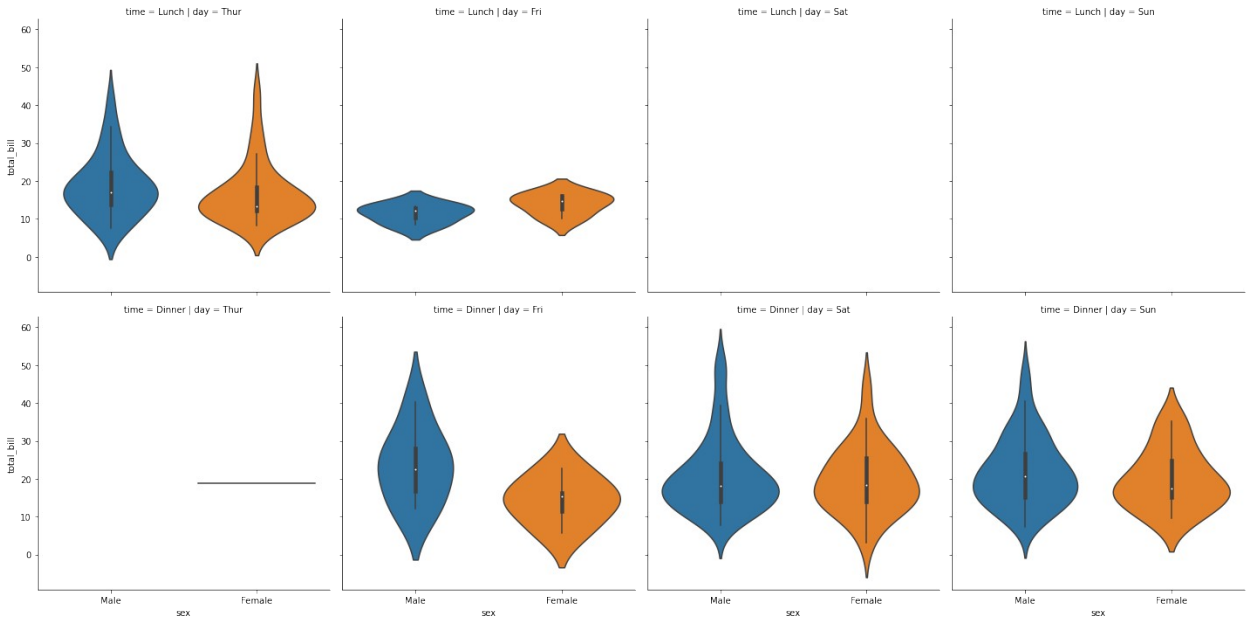
```
<AxesSubplot:xlabel='total_bill', ylabel='tip'>
```




A second way to plot Facet plots -> FacetGrid

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

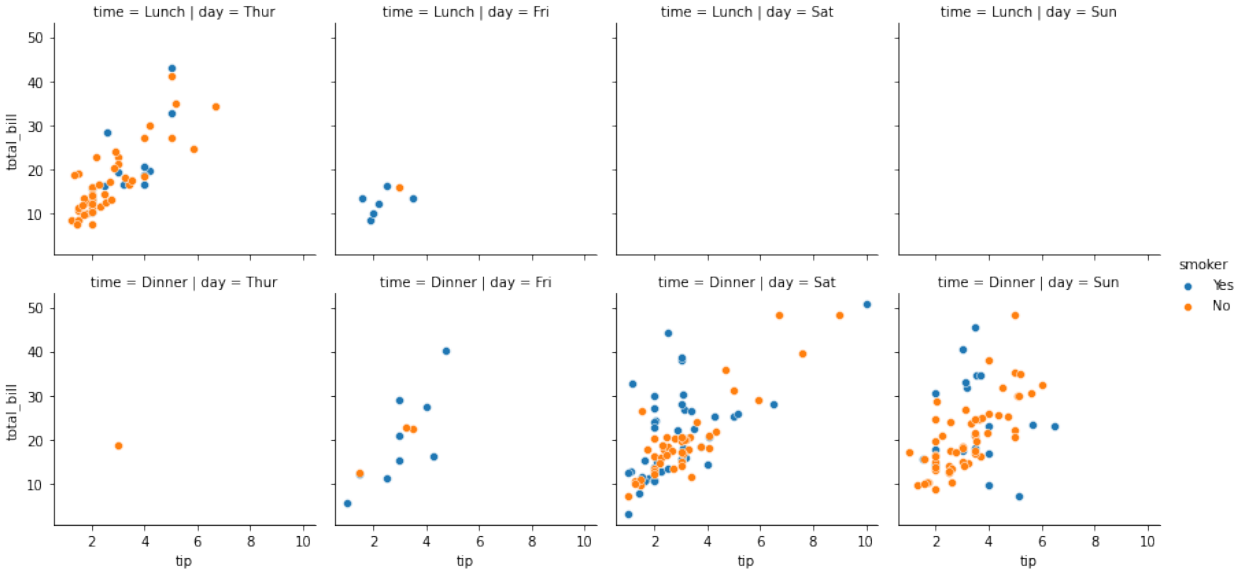
```
<seaborn.axisgrid.FacetGrid at 0x28be694e520>
```




```
## now work on facetgrid
g=sns.FacetGrid(data=tips,col='day',row='time',hue='smoker')
```

```
g.map(sns.scatterplot,'tip','total_bill')
g.add_legend()
```

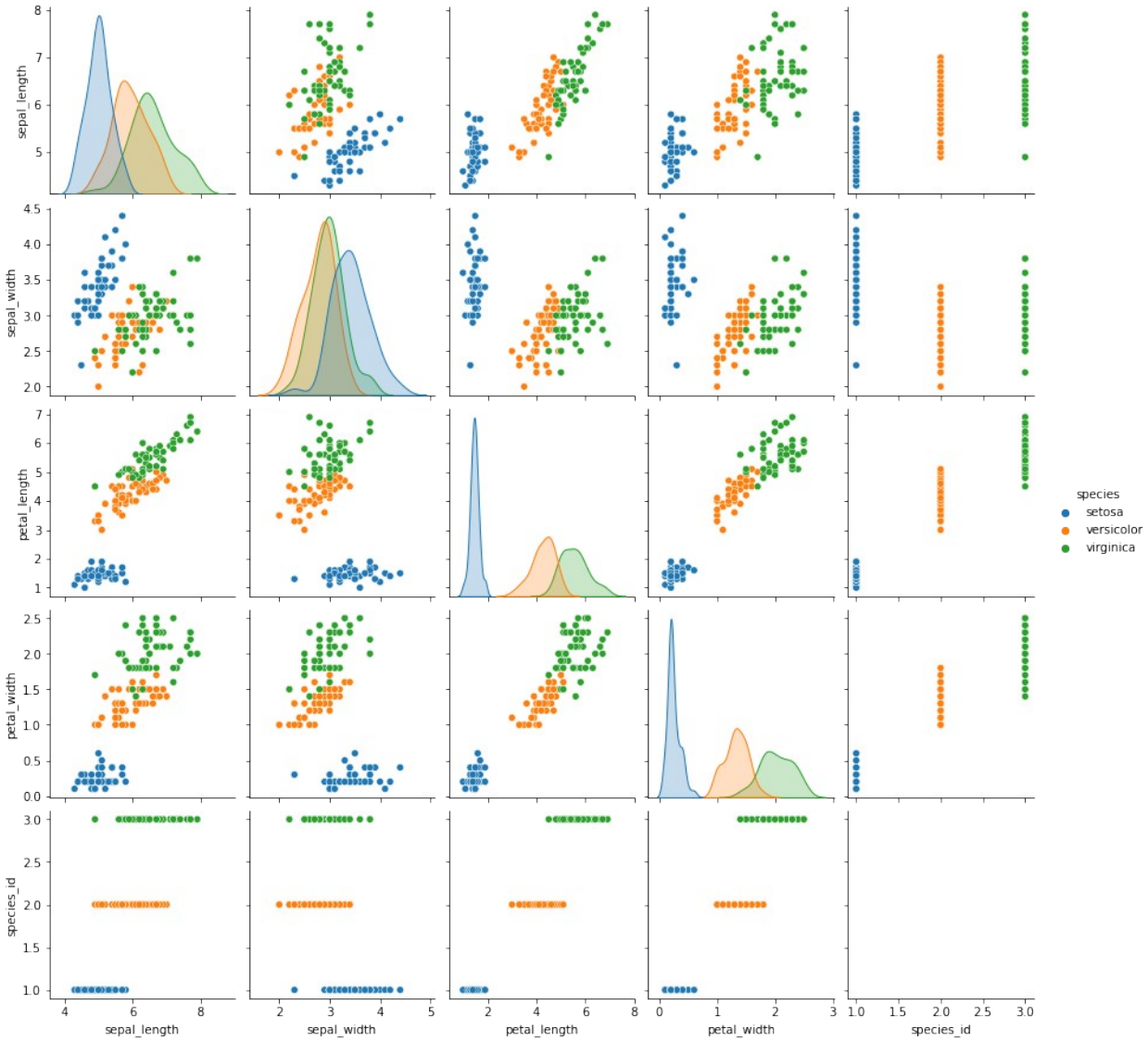
```
<seaborn.axisgrid.FacetGrid at 0x28be6a29b80>
```




## Plotting Pairwise Relationship (PairGrid Vs Pairplot)

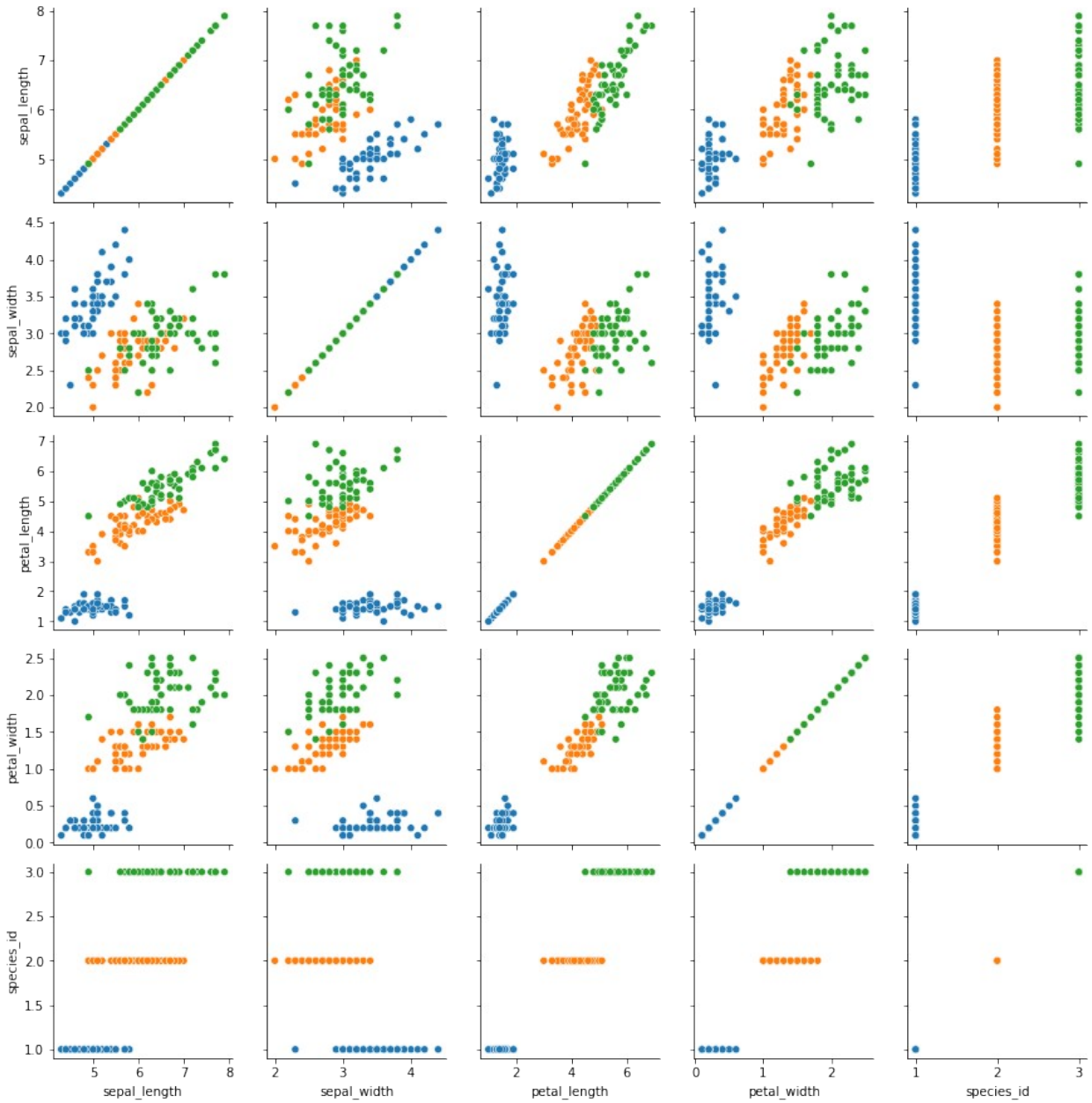
```
sns.pairplot(iris,hue='species')
```

```
<seaborn.axisgrid.PairGrid at 0x28be7c1d400>
```



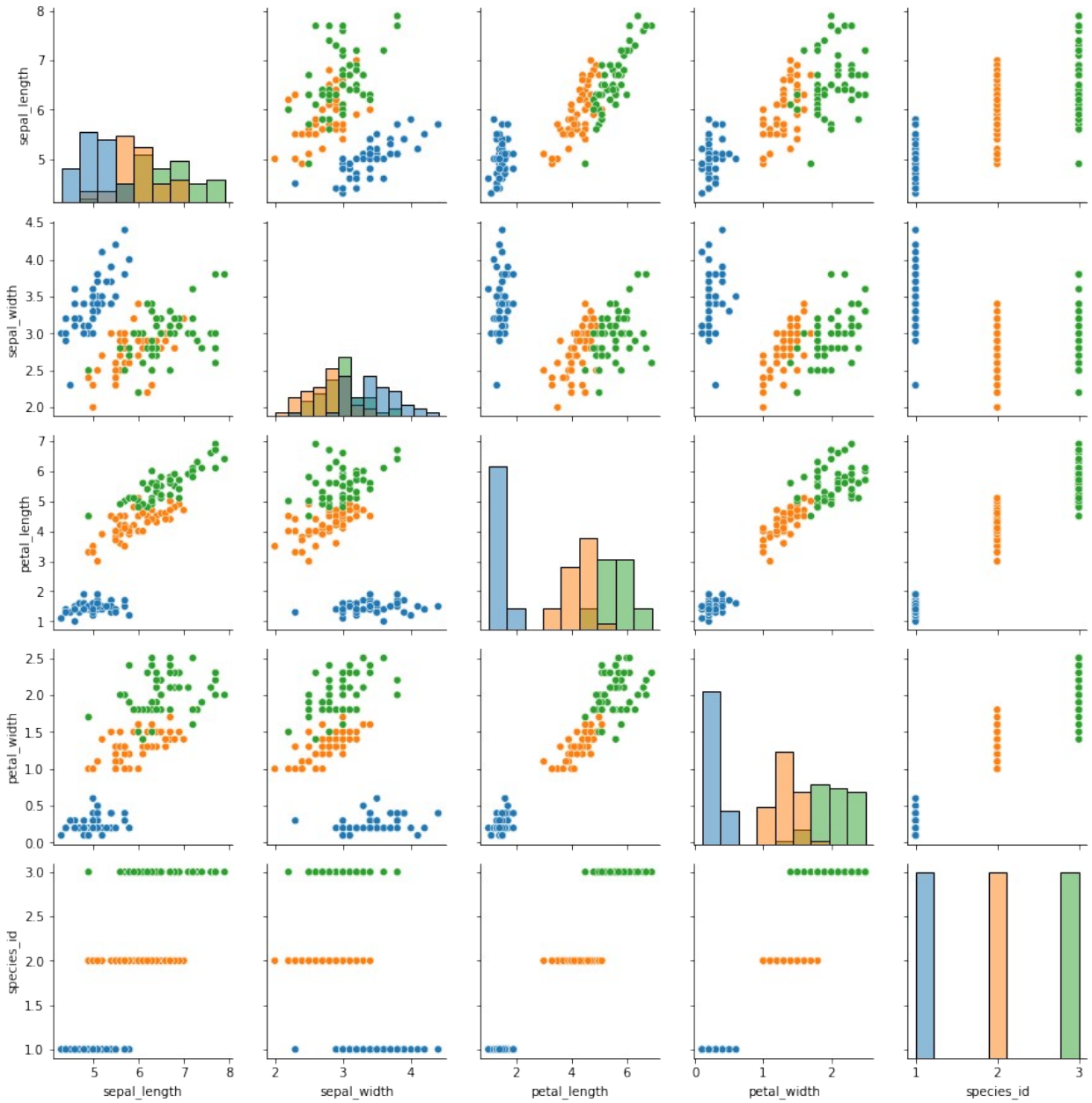

```
# PariGrid
g=sns.PairGrid(data=iris,hue='species')
g.map(sns.scatterplot)
```

```
<seaborn.axisgrid.PairGrid at 0x20a4b708460>
```




```
# map_dia --> map_offdia
g=sns.PairGrid(data=iris,hue='species')
g.map_diag(sns.histplot)
g.map_offdiag(sns.scatterplot)
```

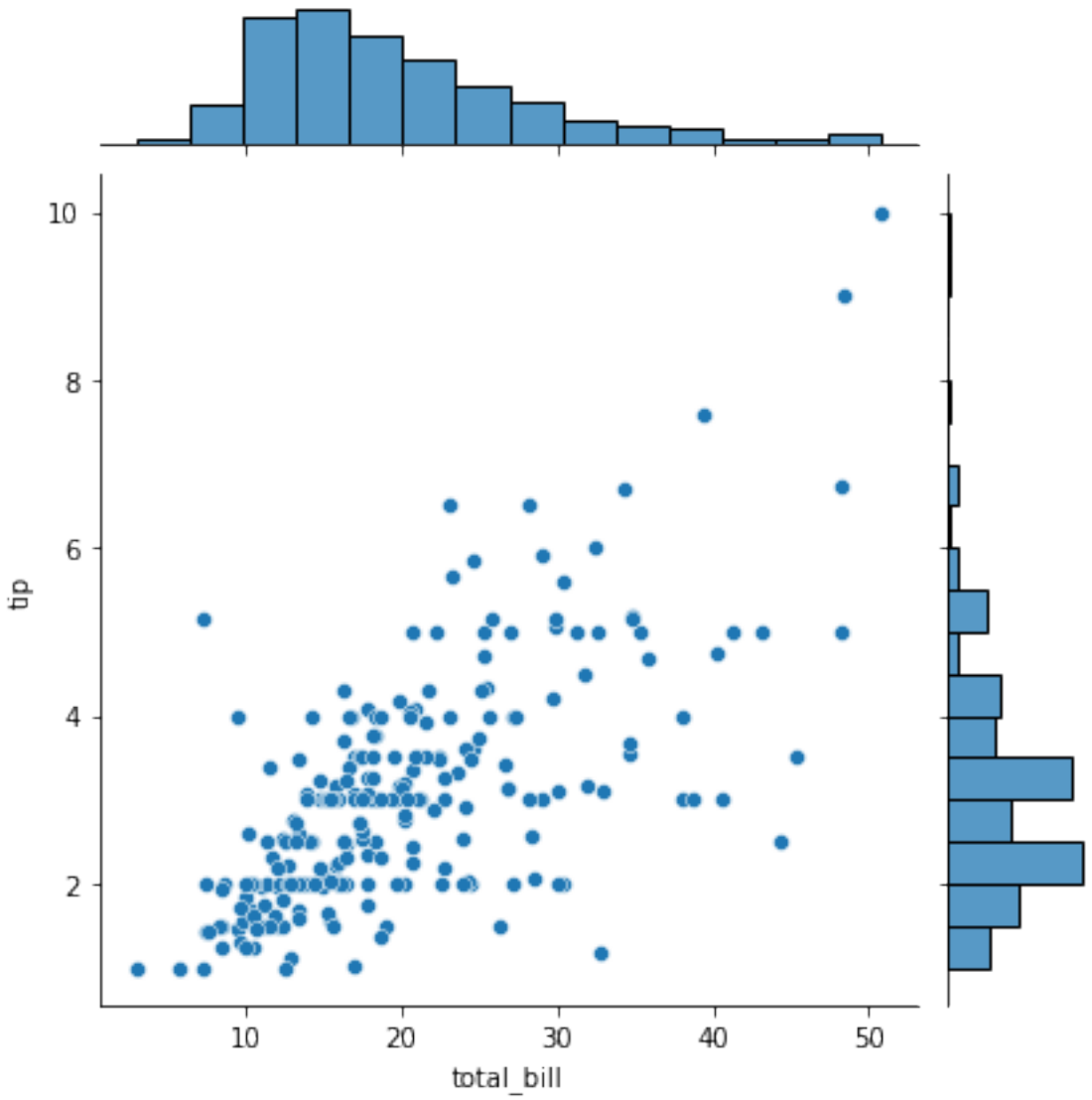
```
<seaborn.axisgrid.PairGrid at 0x20a4e6ad670>
```




## JointGrid Vs jointplot

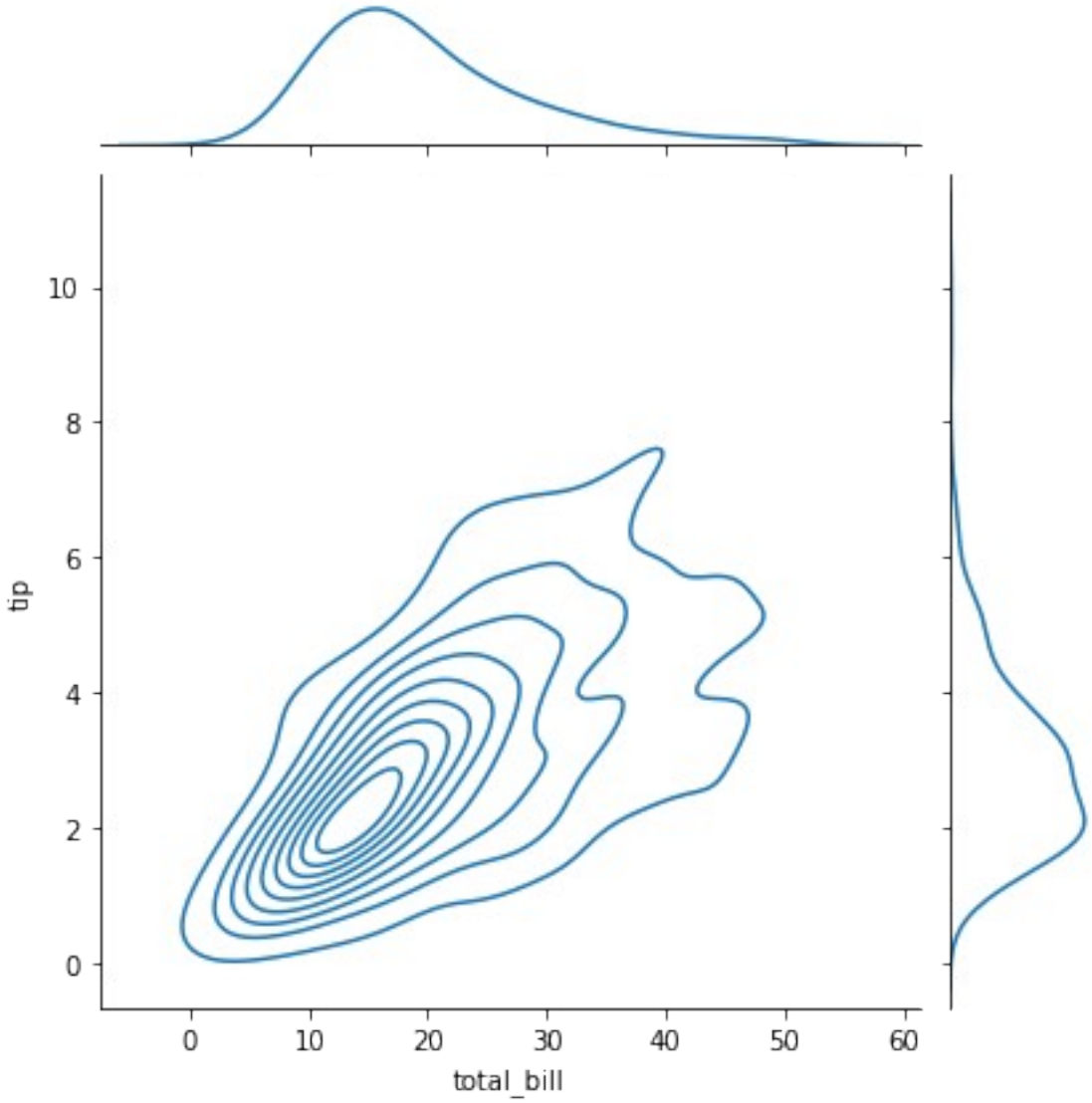
```
sns.jointplot(data=tips,x='total_bill',y='tip')
```

```
<seaborn.axisgrid.JointGrid at 0x28be9b9c730>
```



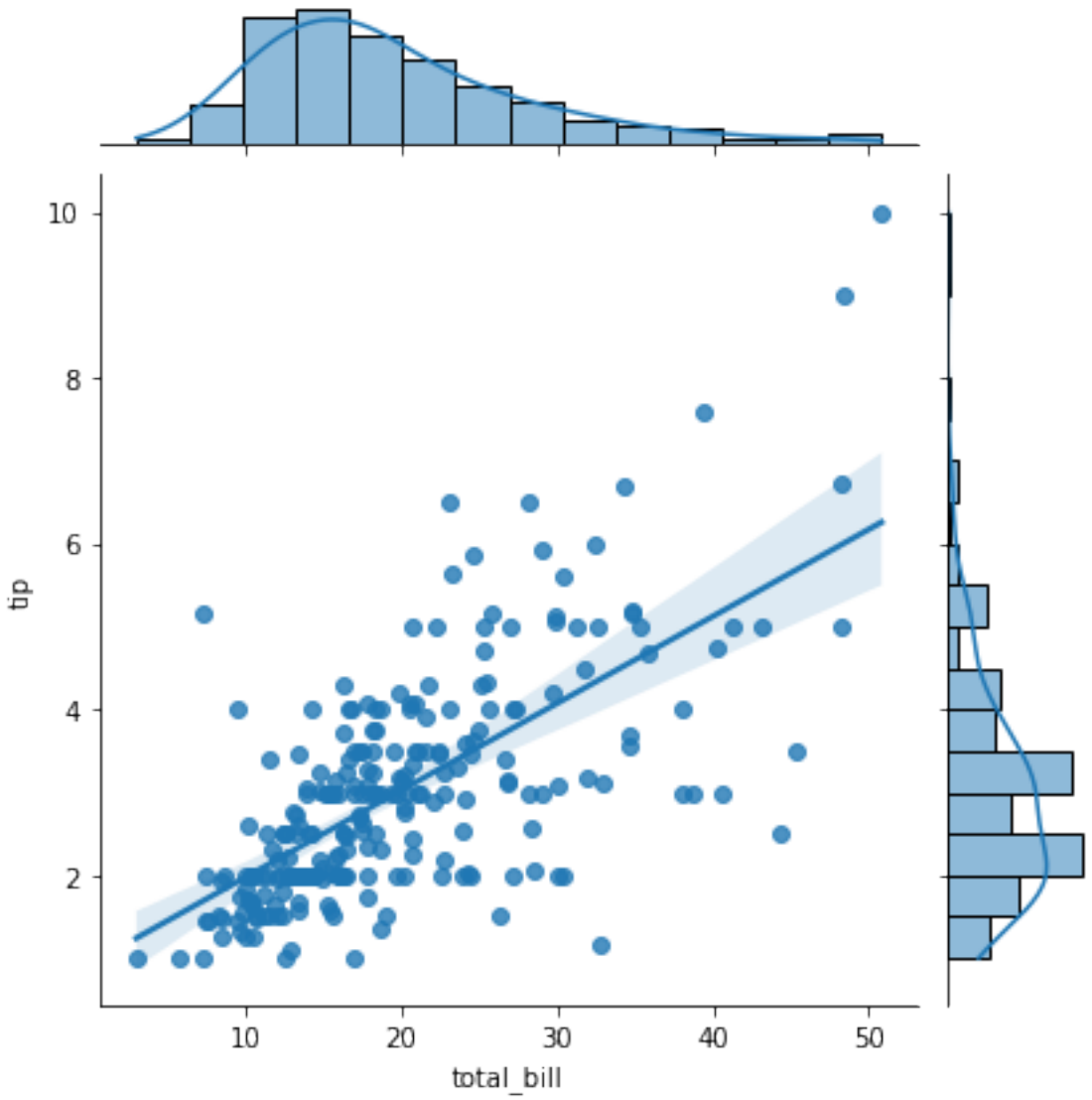

```
sns.jointplot(data=tips,x='total_bill',y='tip',kind='kde')
```

```
<seaborn.axisgrid.JointGrid at 0x28be9ba35e0>
```



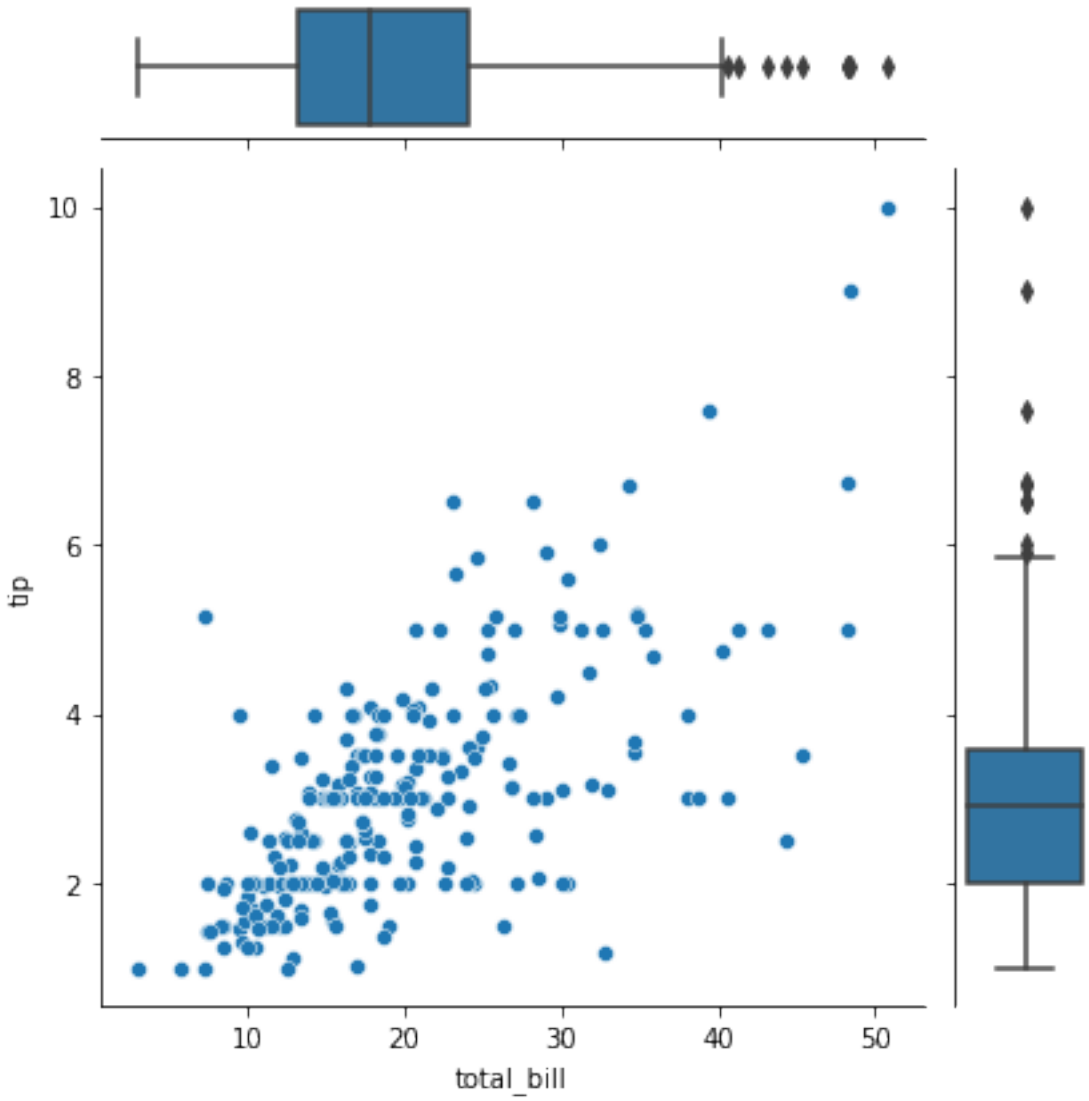

```
sns.jointplot(data=tips,x='total_bill',y='tip',kind='reg')
```

```
<seaborn.axisgrid.JointGrid at 0x28be9dc7d90>
```




```
g=sns.JointGrid(data=tips,x='total_bill',y='tip')
g.plot(sns.scatterplot,sns.boxplot)
```

```
<seaborn.axisgrid.JointGrid at 0x28be9df06a0>
```




## Utitlity functions

```
# get dataset names
sns.get_dataset_names()
```

```
['anagrams',
 'anscombe',
 'attention',
 'brain_networks',
 'car_crashes',
 'diamonds',
 'dots',
 'dowjones',
 'exercise',
 'flights',
 'fmri',
```

```
 'geyser',
 'glue',
 'healthexp',
 'iris',
 'mpg',
 'penguins',
 'planets',
 'seaice',
 'taxis',
 'tips',
 'titanic']
```

```
# load dataset
sns.load_dataset('exercise')
```

```
    Unnamed: 0  id     diet  pulse    time     kind
0            0   1  low fat     85   1 min     rest
1            1   1  low fat     85  15 min     rest
2            2   1  low fat     88  30 min     rest
3            3   2  low fat     90   1 min     rest
4            4   2  low fat     92  15 min     rest
..         ...  ..      ...    ...     ...      ...
85          85  29   no fat    135  15 min  running
86          86  29   no fat    130  30 min  running
87          87  30   no fat     99   1 min  running
88          88  30   no fat    111  15 min  running
89          89  30   no fat    150  30 min  running

[90 rows x 6 columns]
```

```
sns.load_dataset('planets')
```

```
               method  number  orbital_period   mass  distance  year
0     Radial Velocity       1       269.300000   7.10     77.40  2006
1     Radial Velocity       1       874.774000   2.21     56.95  2008
2     Radial Velocity       1       763.000000   2.60     19.84  2011
3     Radial Velocity       1       326.030000  19.40    110.62  2007
4     Radial Velocity       1       516.220000  10.50    119.47  2009
...               ...     ...              ...    ...       ...   ...
1030          Transit       1         3.941507    NaN    172.00  2006
1031          Transit       1         2.615864    NaN    148.00  2007
1032          Transit       1         3.191524    NaN    174.00  2007
1033          Transit       1         4.125083    NaN    293.00  2008
1034          Transit       1         4.187757    NaN    260.00  2008

[1035 rows x 6 columns]
```