wbs
WARWICK BUSINESS SCHOOL
THE UNIVERSITY OF WARWICK

For the Change Makers

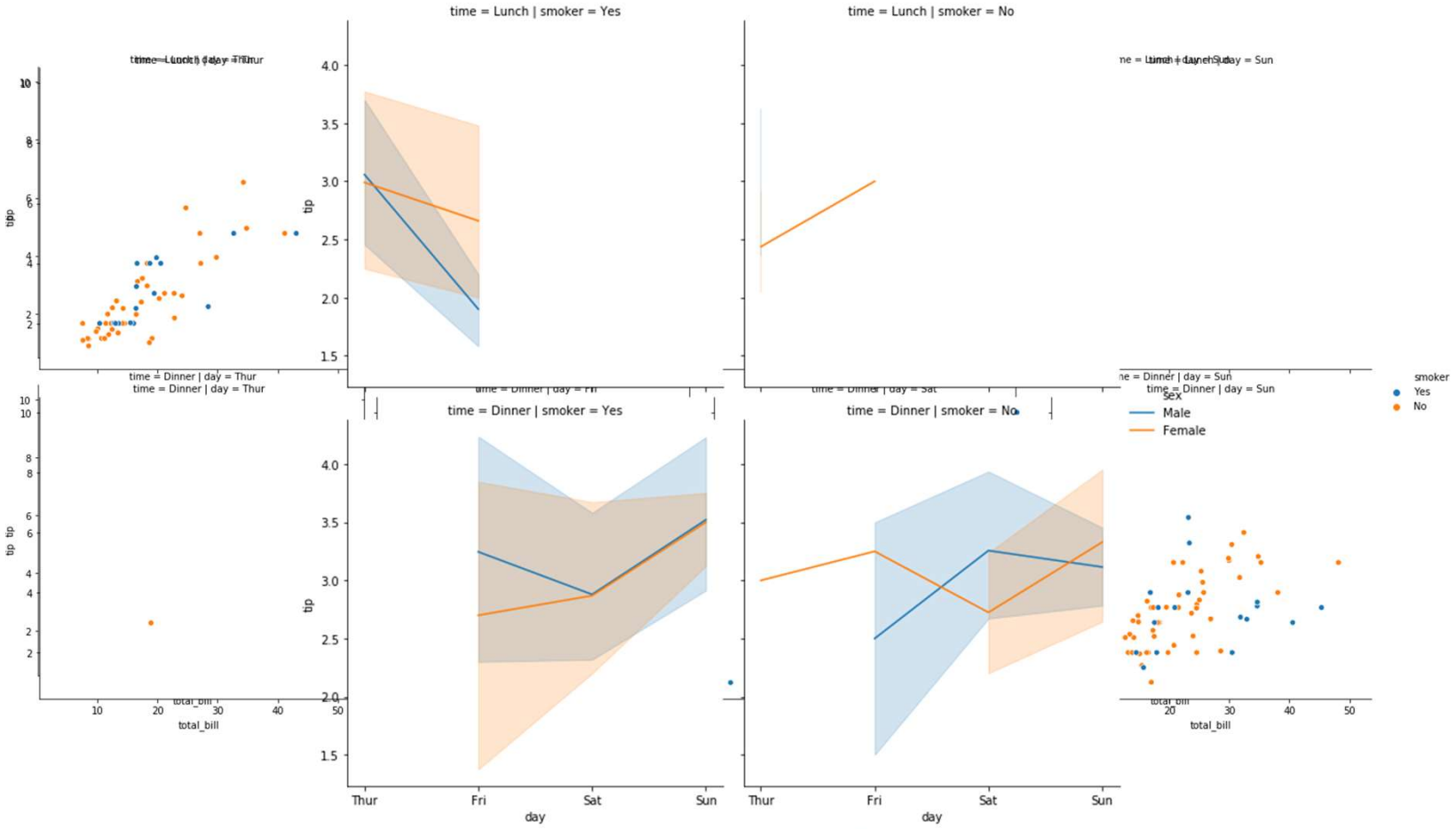
Advanced Programming for Data Science

Week 4: Data Visualization
Information Systems and Management
Warwick Business School

Showing multiple relationships

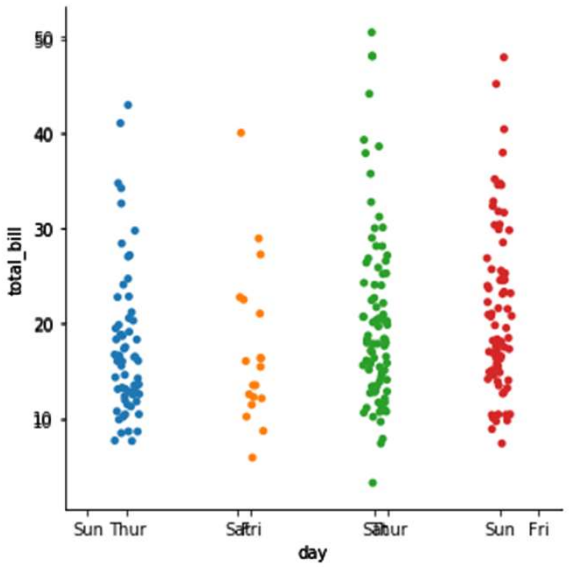
- Very often, when you explore your dataset, you would like to compare multiple relationships at once.
- You can create a grid of graphs and specify facets (grouping dimensions) by passing arguments:
  1. col: the variable used for splitting in horizontal direction.
  2. row: the variable used for splitting in vertical direction.

```
sns.relplot(x="total_bill", y="tip", col='day',
row='time', data=tips)
```

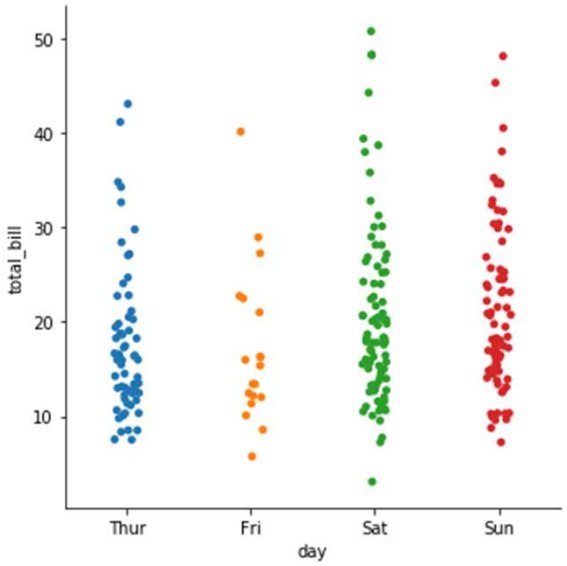

time = Lunch | smoker = Yes
time = Lunch | smoker = No
time = Dinner | day = Thur
time = Dinner | smoker = Yes
time = Dinner | smoker = No
time = Dinner | day = Sun
tip
total_bill
day
Thur
Fri
Sat
Sun
sex
Male
Female
smoker
Yes
No

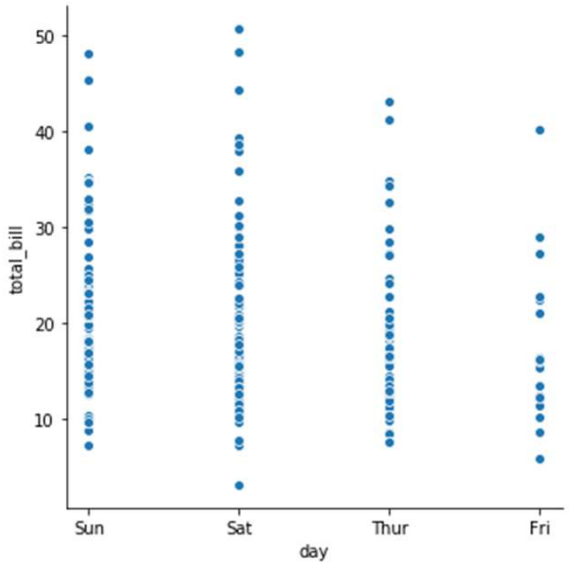
Plotting categorical variables

- `relplot()`, while is best to plot relationship between numeric variables, is able to handle categorical variables as well.
- `catplot()` is a function dedicated for categorical plotting.
- `sns.catplot(x="day", y="total_bill", data=tips)`



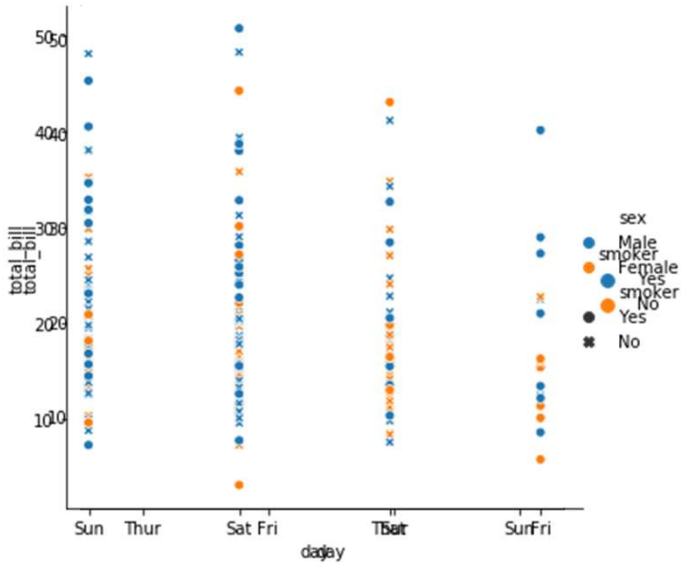
Spot three differences?





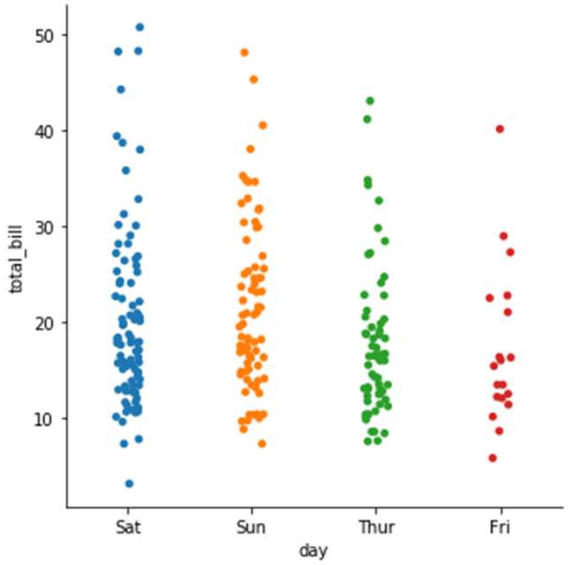
Adding dimension

- *catplot()* only support hue to add another dimension.
- If you need add more dimensions with different colours, styles or sizes, you can use *relplot()* instead.



Order the category

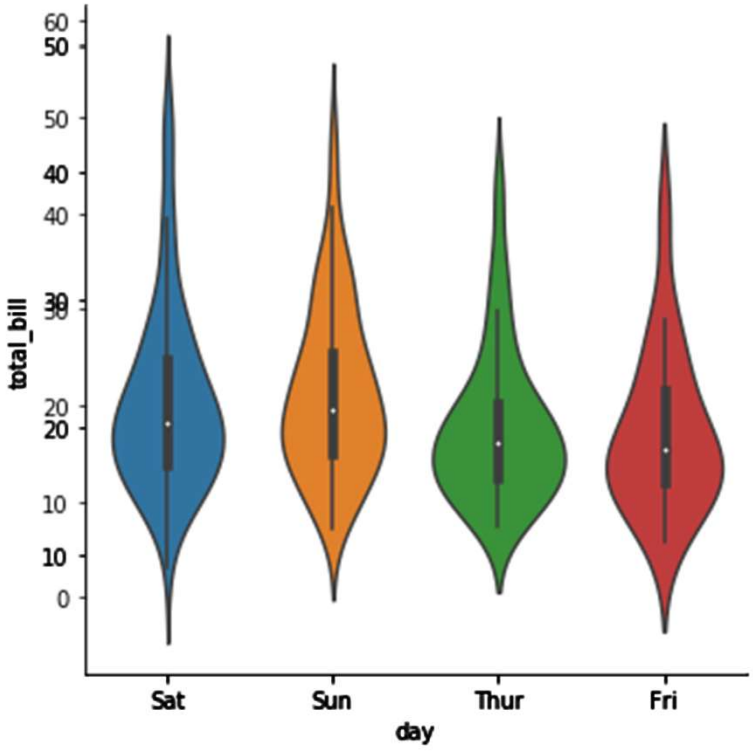
- By default, seaborn will try to infer the order of categories form your data, such as numerical or temporal order.
- However, it does not always work well. So we can manually set the order by passing a list of values to argument: order
- ```
  sns.catplot(x="day", y="total_bill", order=['Sat', 'Sun', 'Thur', 'Fri'], data=tips)
  ```


  ```

Visualizing the distribution

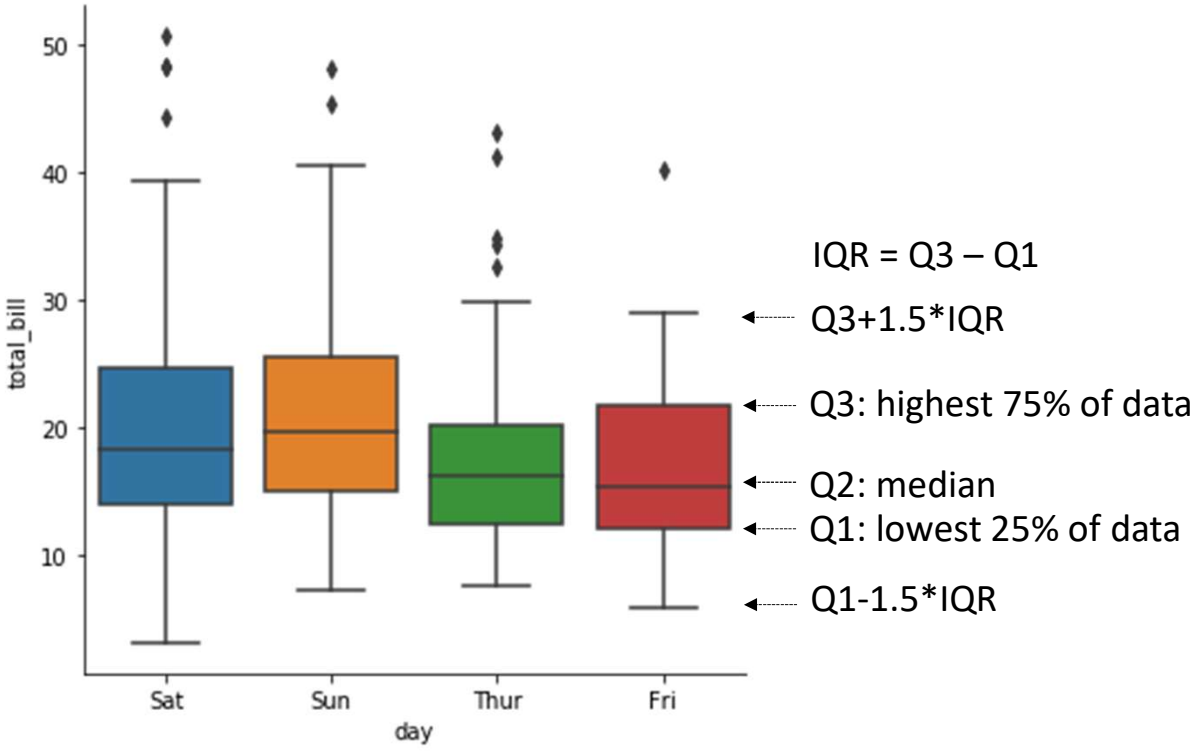
- With the default scatterplots, it gets harder to see the distribution of your data when the size gets bigger.
- We can use several other plotting approaches showing the distribution information, by passing argument kind with:

1. box and boxen
2. violin



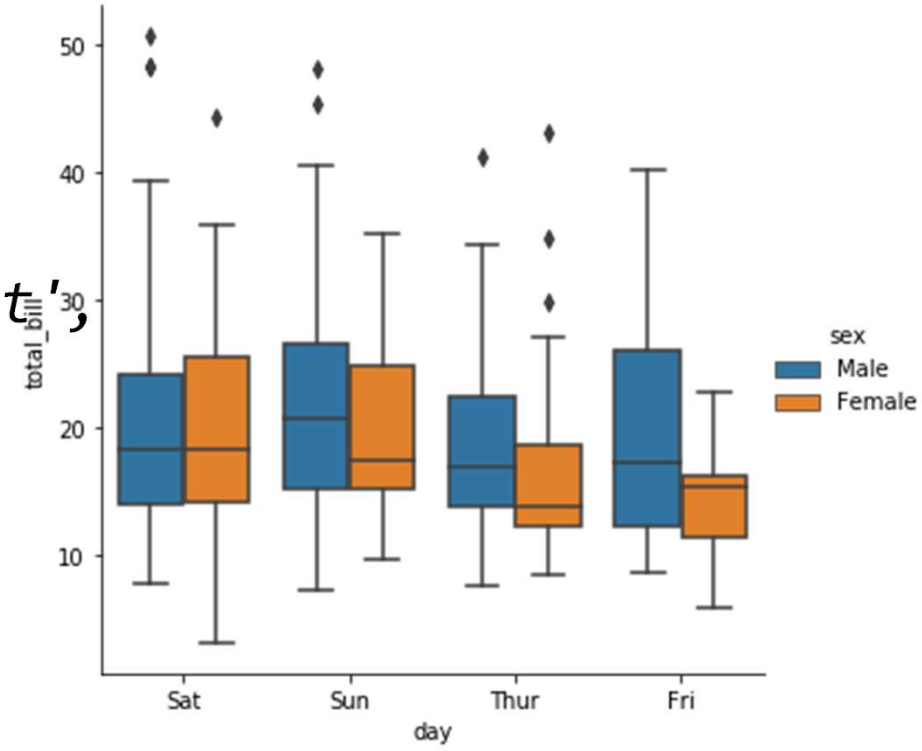
Box-plot explained

- Box plot is used for descriptive statistics. It is created based on quartiles, and also called as box-and-whisker plot.
- Box is bounded by 1st and 3rd quartiles.
- Whiskers extend to 1.5 IQRs of 1st and 3rd quartiles.



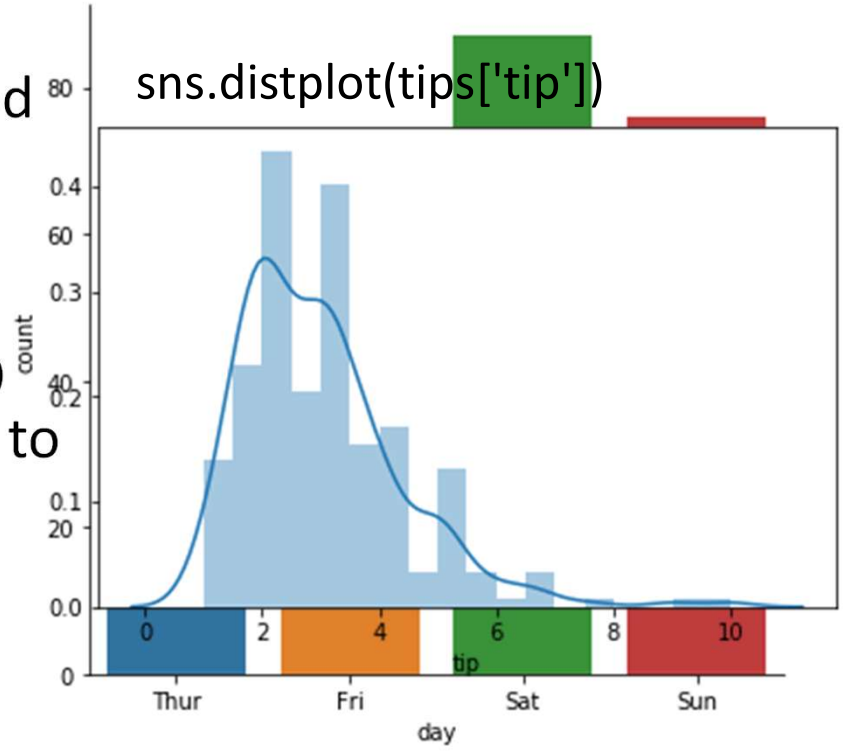
Adding dimensions to box plot

- Box plot also supports adding third dimensions with argument hue.
- `sns.catplot(x="day", y="total_bill", order=['Sat', 'Sun', 'Thur', 'Fri'], kind='box', hue= 'sex', data=tips)`



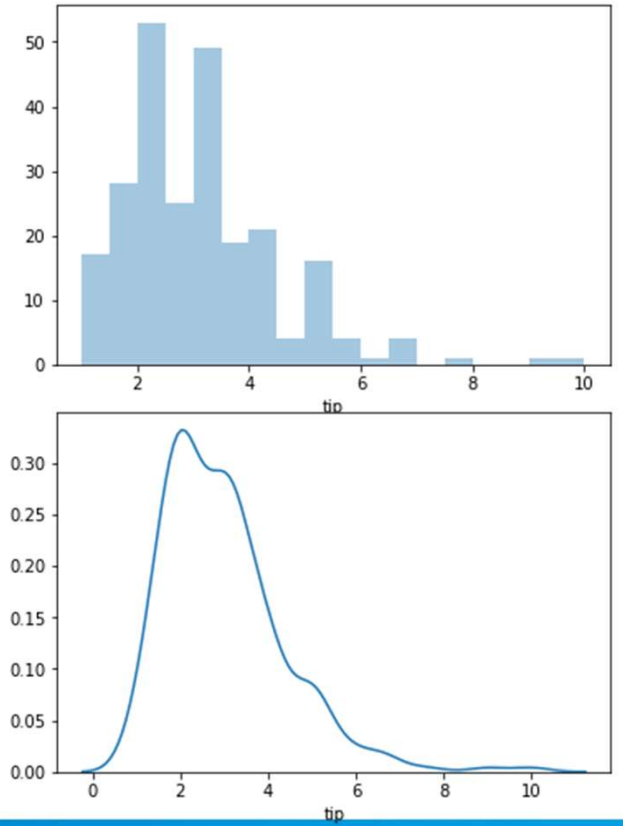
Descriptive statistics for individual variables

- So far, we focus on the relationships between two variables. We often need to have a better idea about individual variables.
- We can use histogram to help us.
- It can be "achieved" with *catplot()* for categorical variables with kind set to "count";
- and *distplot()* for numeric variables.



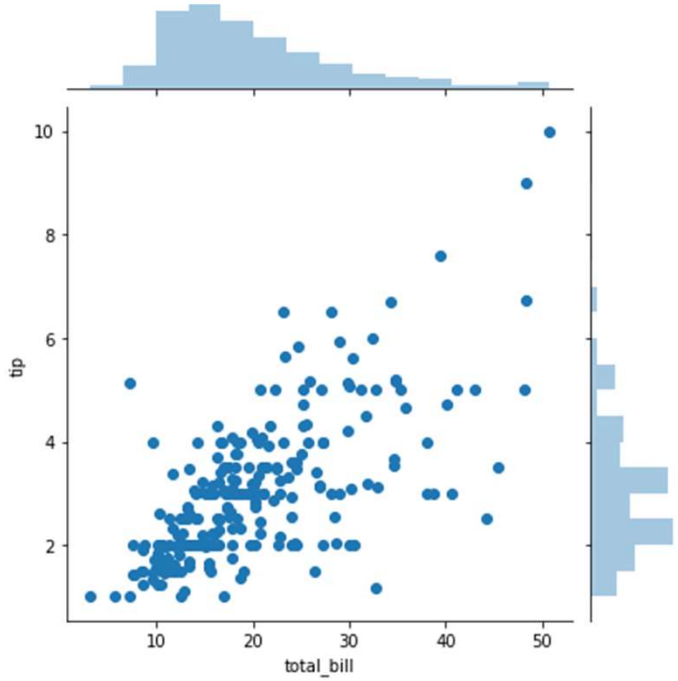
Histogram explained

- Histogram is used to visualize univariate distributions.
- By default, it fits a kernel density estimate (KDE), which is basically a non-parametric approach to create a smoothed line based on discrete data.
- You may turn on and off histogram and KDE by setting arguments:
  - kde: false or True
  - hist: False or True



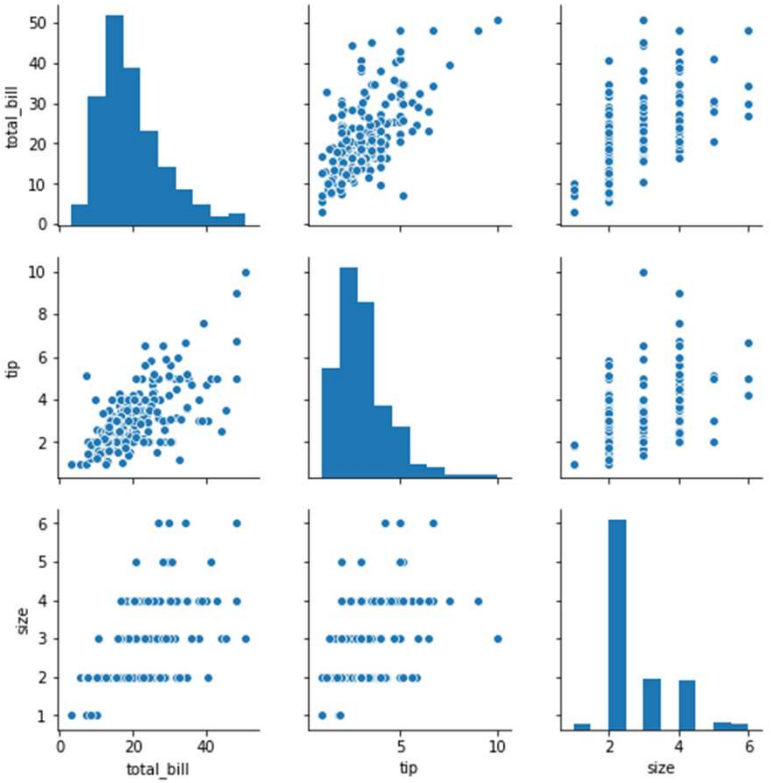
Joining histogram and scatter plots

- You can combine histogram (one variable) and scatter plots (two variables) together to give a comprehensive view.
- This can be done with *jointplot()*;
- *sns.jointplot(x="total_bill", y="tip", data=tips)*

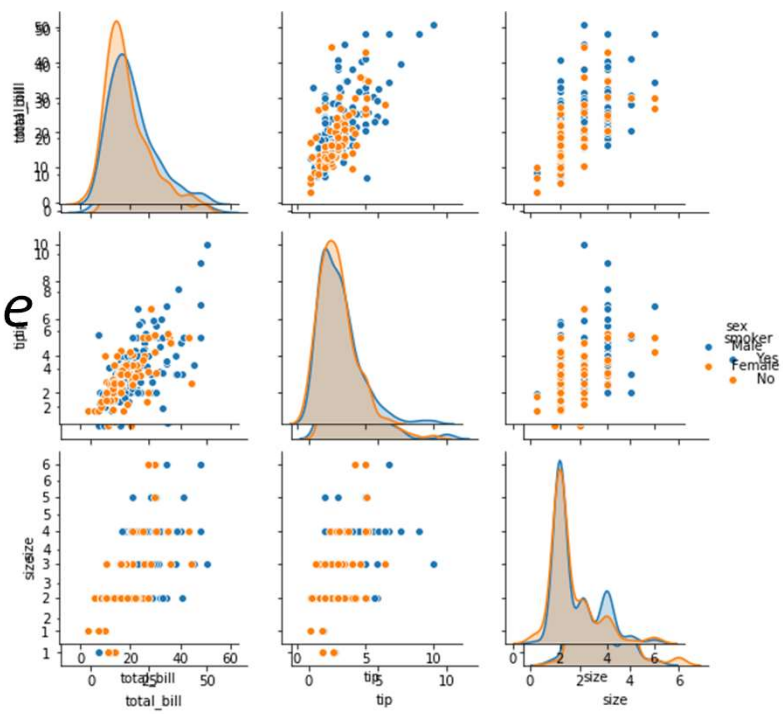


Visualizing pairwise correlations

- A very handy function *pairplot()* can be used to visualize the pairwise correlations for a quick overview.
- *sns.pairplot(tips)*
- When you have too many dimensions in you dataset, you may specify the dimensions in a list and pass to argument vars:



- *pairplot()* only plots continuous (numeric) variables. You may pass categorical variable with argument hue to group the data.
- *sns.pairplot(tips,hue='smoker')*



Exercise

- Iris flower data set is a famous data set originally used by biologist Ronald Fisher in 1936. It is about three different species of Iris flowers. It becomes a classic dataset for practicing data science techniques and is provided as built-in dataset in many Python libraries.
- Import the Iris dataset and explore factors related to the speciation.

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