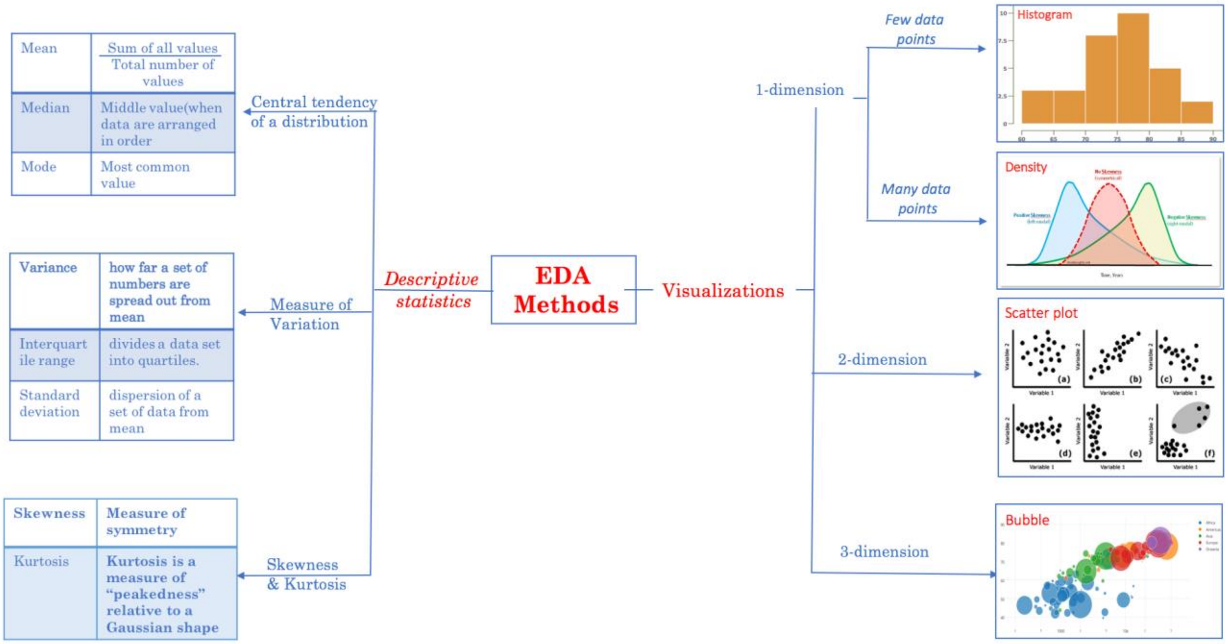
Analyze the data through data visualization using Seaborn

Using different plots to visualize data and will learn when to use these plots.

Sanket Doshi
Follow
Feb 3, 2019 · 6 min read

Data visualization is an important part of any data analysis. It helps us to recognize relations between variables and also to find which variables are significant or which variable can affect the predicted variable.



Mean	Sum of all values / Total number of values
Median	Middle value(when data are arranged in order
Mode	Most common value

Variance	how far a set of numbers are spread out from mean
Interquartile range	divides a data set into quartiles.
Standard deviation	dispersion of a set of data from mean

Skewness	Measure of symmetry
Kurtosis	Kurtosis is a measure of "peakedness" relative to a Gaussian shape

Explorative data analysis

In machine learning model while training any model you need to first find which features are important or on which features the result is dependent. This can be done using data analysis and data visualization.

We'll learn how to visualize different types of data, and what we can infer from that plot, and when to use them.

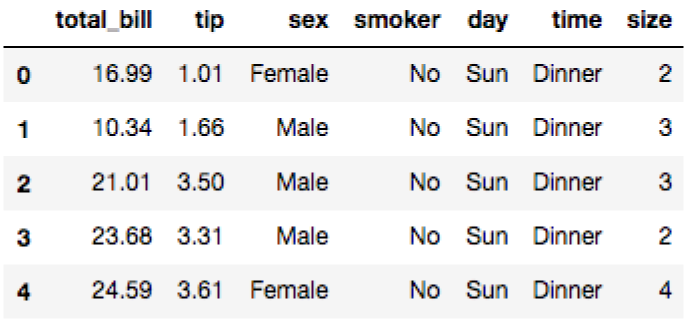
Seaborn is a library built on matplotlib. It's easy to use and can work easily with Numpy and pandas data structures.

We'll be using inbuilt dataset provided by seaborn name `tips`.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

Loading the dataset

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



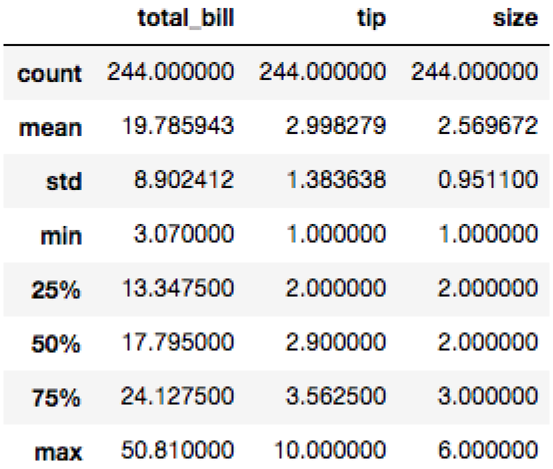
	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4

Tips dataset head

We can see that sex, smoker, day and time are categorical data. And total_bill, tip, and size are numerical data.

We will retrieve some common information such as min, max, unique and count for given numerical data.

```
tips.describe()
```



	total_bill	tip	size
count	244.000000	244.000000	244.000000
mean	19.785943	2.998279	2.569672
std	8.902412	1.383638	0.951100
min	3.070000	1.000000	1.000000
25%	13.347500	2.000000	2.000000
50%	17.795000	2.900000	2.000000
75%	24.127500	3.562500	3.000000
max	50.810000	10.000000	6.000000

Descriptive statistics of numerical data.

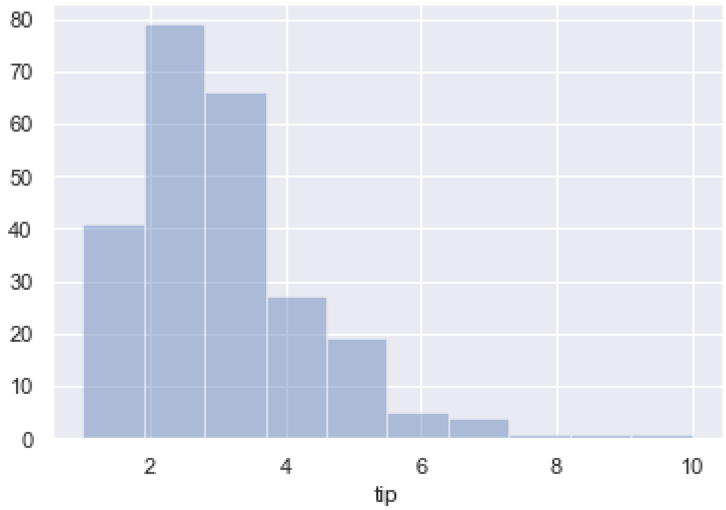
Now we'll infer relationships between different variables using plots.

Univariate plots

These plots are based on a single variable and show the frequency of uniques values of a given variable.

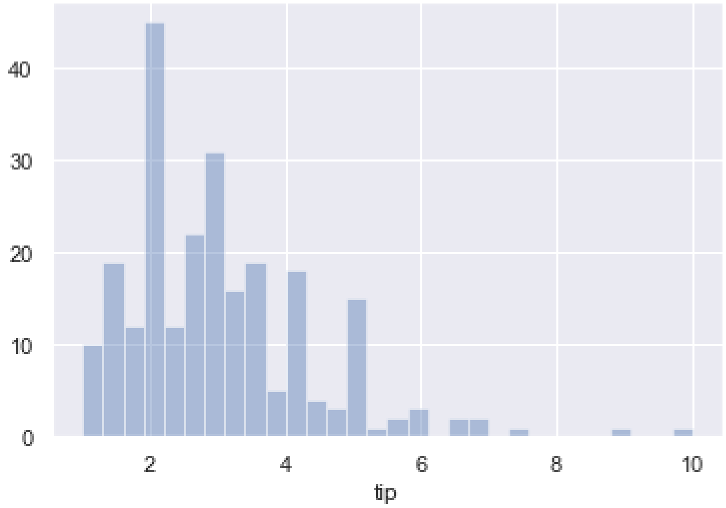
Histogram

```
sns.distplot(tips['tip'], kde=False, bins=10);
```



Histogram

Larger the bins value more accurate the result.

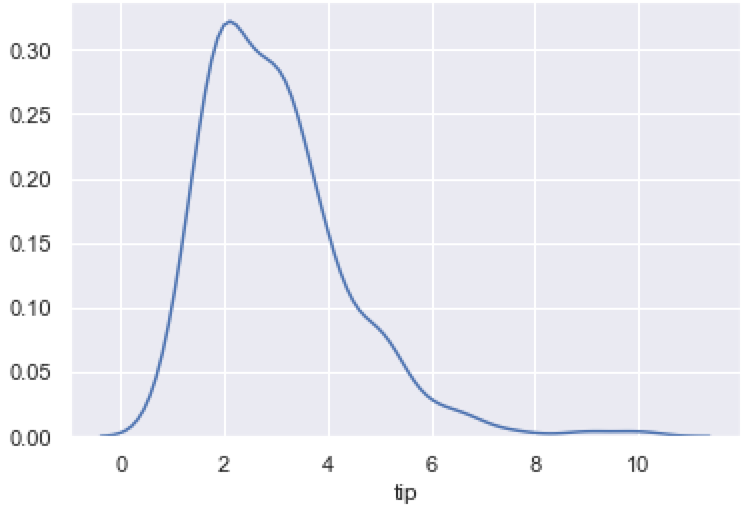


Histogram for bins=30

We can see that the count of different tip value present in the dataset and infer that most of the tips are between 2 and 4.

Kerner Density Estimate (KDE)

```
sns.distplot(tips['tip'],hist=False, bins=10);
```



Kernel density estimate of tip

KDE is a way to estimate the probability density function of a continuous random variable. It is used when you need to know the distribution of the variable.

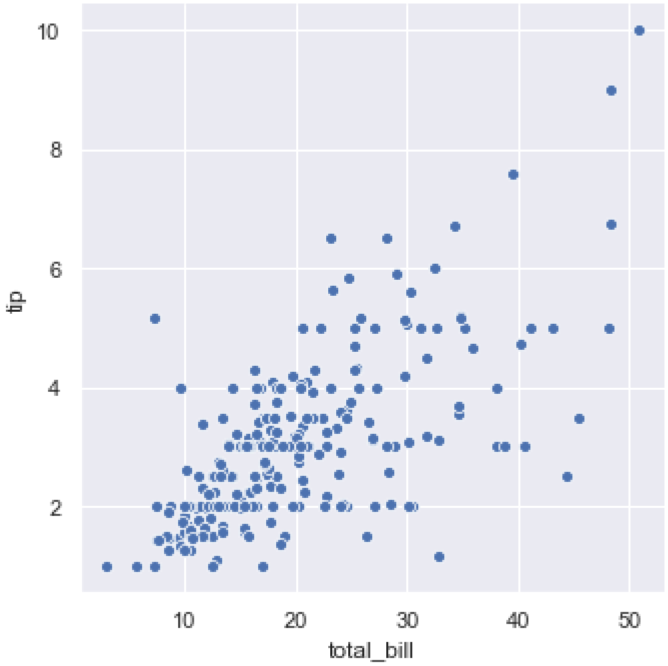
Bivariate Plots

This type of plots is used when you need to find a relation between two variables and to find how the value of one variable changes the value of another variable. Different types of plots are used based on the data type of the variable.

Statistical data types

Scatterplot

```
sns.relplot(x="total_bill", y="tip", data=tips);
```



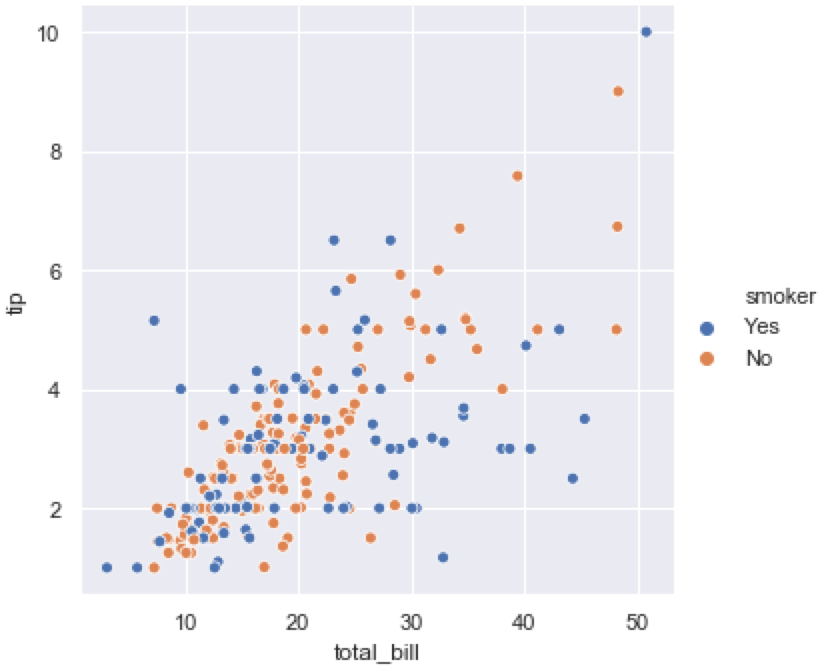
Scatterplot

Default plot type of `relplot` is scatterplot. It shows the relationship between two variables. So, if you need to find the correlation between two variables scatterplot can be used.

In the given example we can see that if `total_bill` is between 10–20 than the tip will be mostly above 2.

We can add the third variable also in scatterplot using different colors or shape of dots.

```
sns.relplot(x="total_bill", y="tip", hue="smoker", data=tips);
```

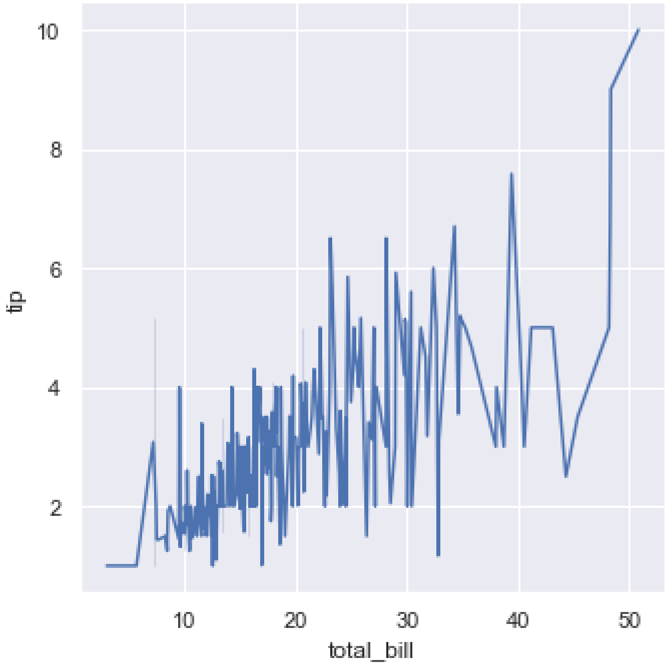


scatterplot for 3 variables

Lineplot

This plot is similar to the scatterplot but instead of dots, it displays the line joining all the dots by arranging the variable value represented on the x-axis.

```
sns.relplot(x="total_bill", y="tip", kind="line", data=tips)
```

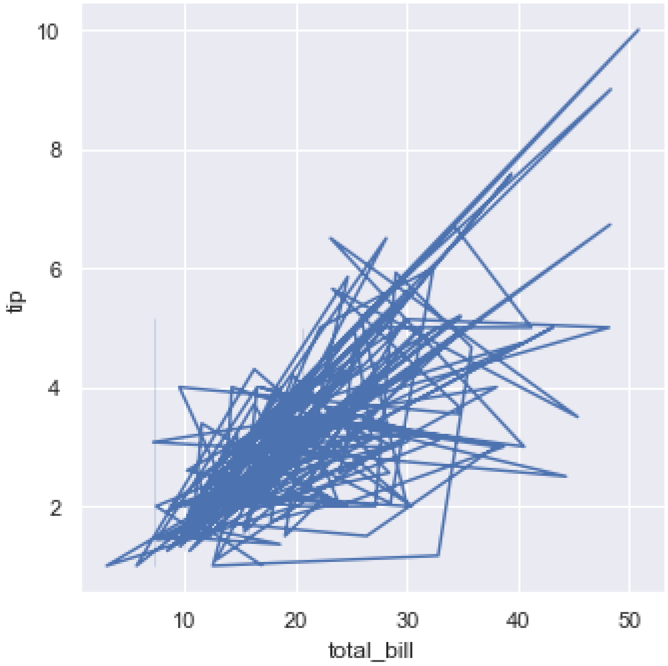


Lineplot

As we can see that distribution is too random of tips with respect to total_bills. We can infer that tip is not much dependent on the value of total_bill.

As line plot arranges the rows as per total_bill and then joins the points we can disable the sorting.

```
sns.relplot(x="total_bill", y="tip", sort=False, kind="line",
data=tips)
```



lineplot sort false

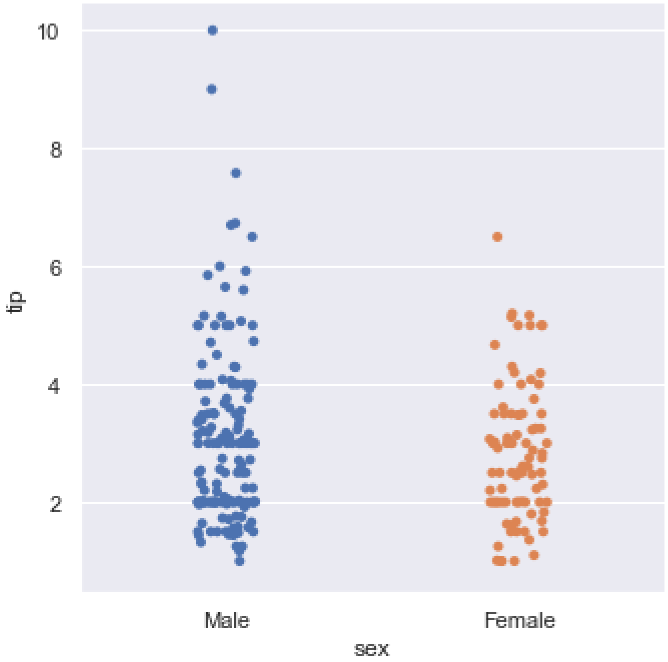
The plot is too messy and we cannot infer much accurately in this graph.

This plot takes time for large dataset as sorting is required first.

Categorical data types

Scatterplot

```
sns.catplot(x="sex", y="tip", data=tips);
```



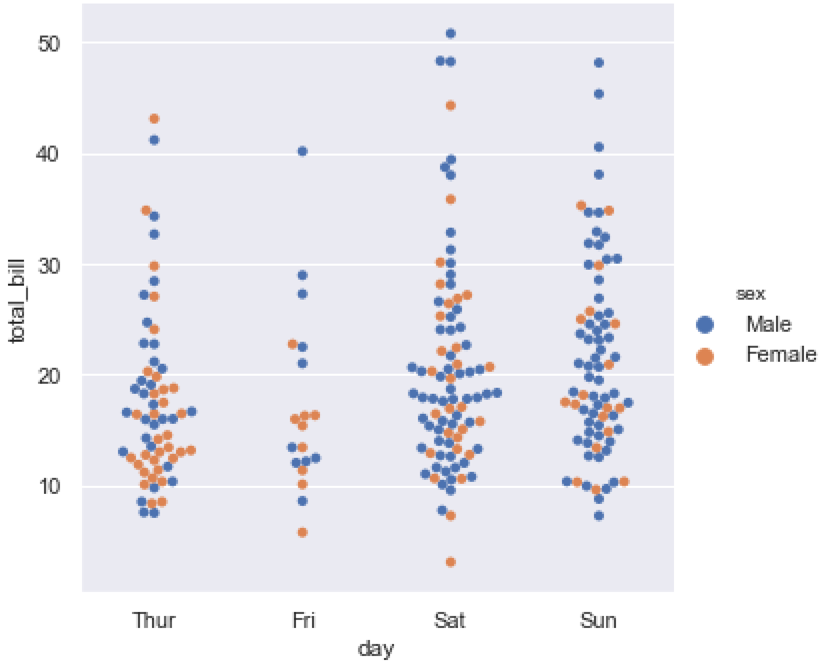
categorical scatterplot

We can see that most of the tips are concentrated between 2 and 4 irrespective of the `gender`.

Different types of scatterplots can be made using attribute `kind` in seaborn.

The third variable can be used using attribute `hue` in seaborn.

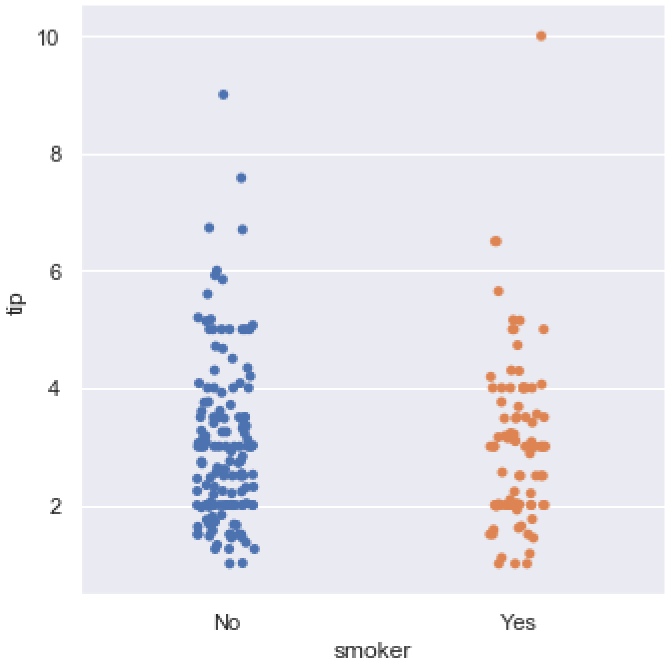
```
sns.catplot(x="day", y="total_bill", hue="sex", kind="swarm",
data=tips);
```



scatterplot using hue and kind

The categories to be represented on the x-axis are sorted as per the pandas categories. If you want the order of your demand you can use `order` attribute in seaborn.

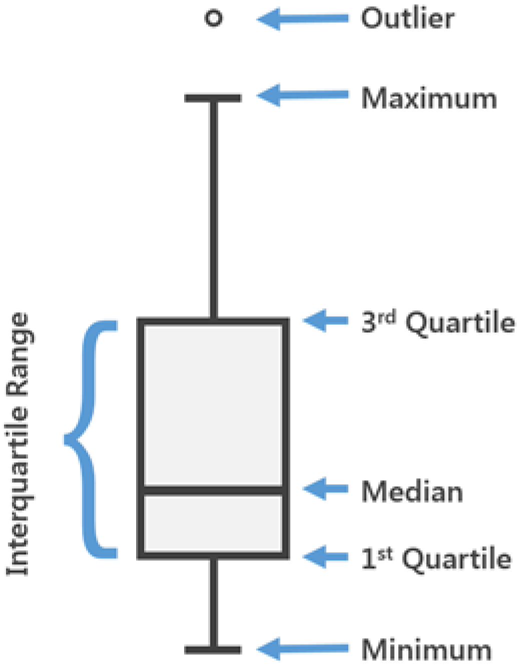
```
sns.catplot(x="smoker", y="tip", order=["No", "Yes"], data=tips);
```



scatterplot using order

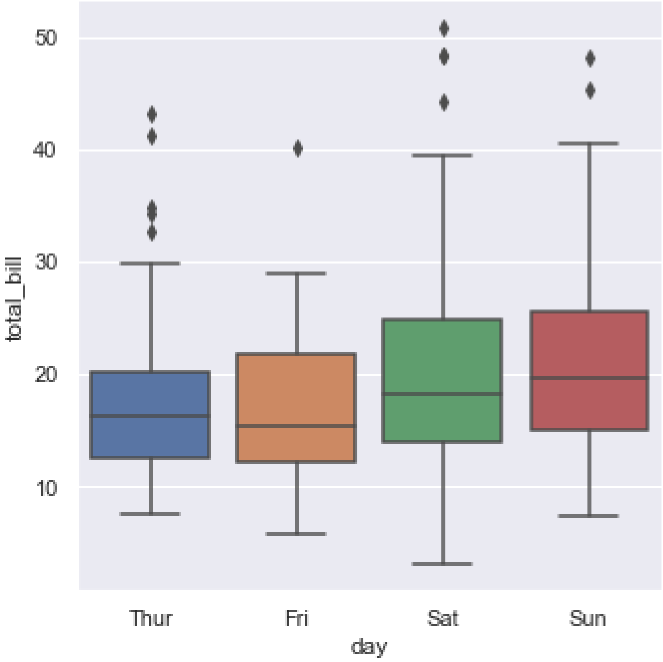
Boxplot

A box plot (or box-and-whisker plot) shows the distribution of quantitative data in a way that facilitates comparisons between variables or across levels of a categorical variable. It can give all the stats provided in dataset `.describe` in a single plot. If the dataset is too large and the range of value is too big then it shows some values as outliers based on an inter-quartile function.



boxplot description

```
sns.catplot(x="day", y="total_bill", kind="box", data=tips);
```

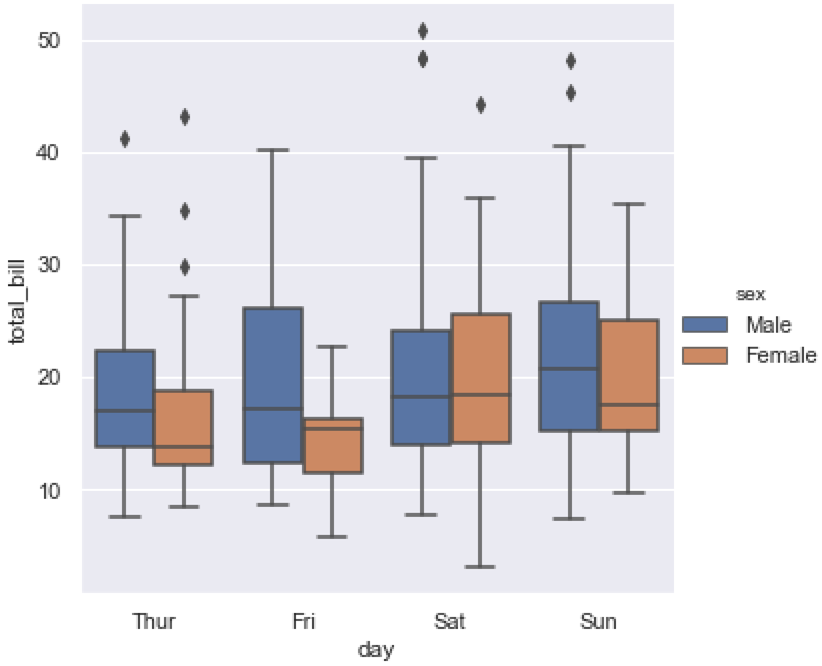


boxplot

The shaded box represents the values between 25-quartile and 75-quartile. The horizontal line in the shaded box shows the median. Two horizontal lines at the bottom and at the top represent the minimum, and the maximum value respectively. The dots represent the outliers calculated based on the inter-quartile function. Using these plots we can compare values for different categories in a single graph. We can infer from the given graph that the amount of total_bill is higher on weekends than weekdays.

To use the third variable in box plots

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

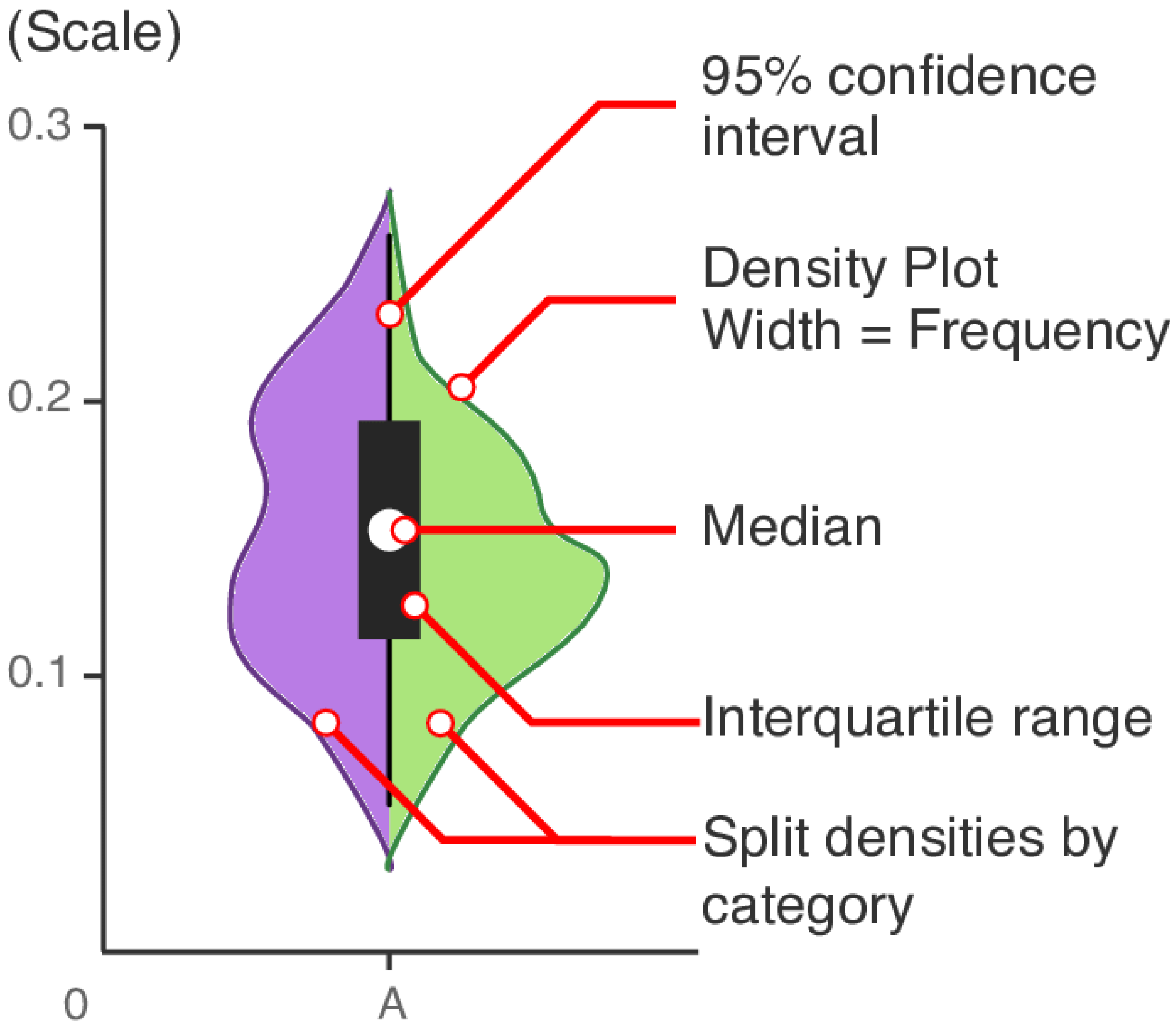


Boxplot using the third variable

We can get so much information from a single graph. In this graph, we can see that the average amount of total_bill for females is always less than the males. So, we can say that `total_bill` amount is dependent on `sex` .

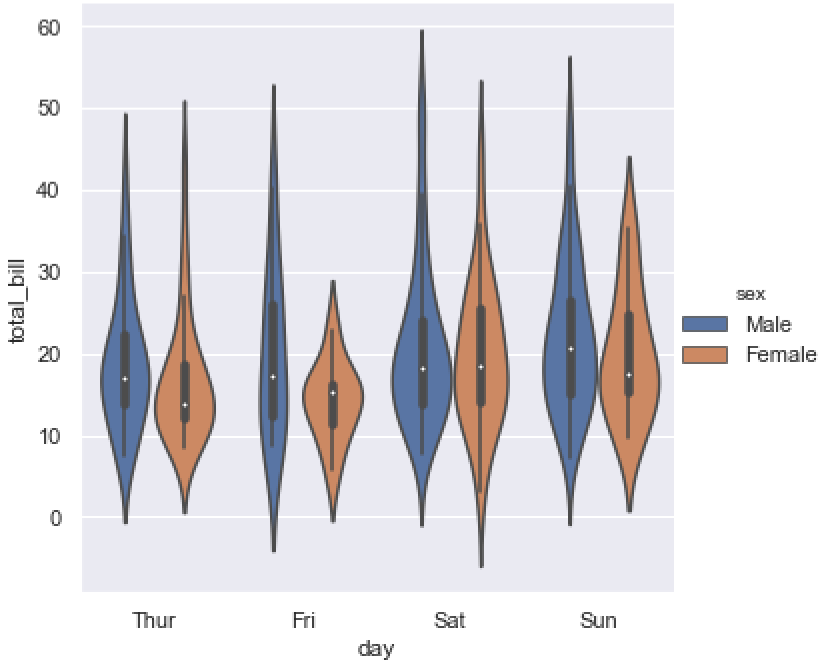
Violin Plot

This plot is used to visualize the distribution of the data and its probability density. This chart is a combination of a Box Plot and a Density Plot. So if you need to find the frequency distribution along with box plot than us violin plot.



Violin plot description

```
sns.catplot(x="day", y="total_bill", hue="sex",
            kind="violin", data=tips);
```



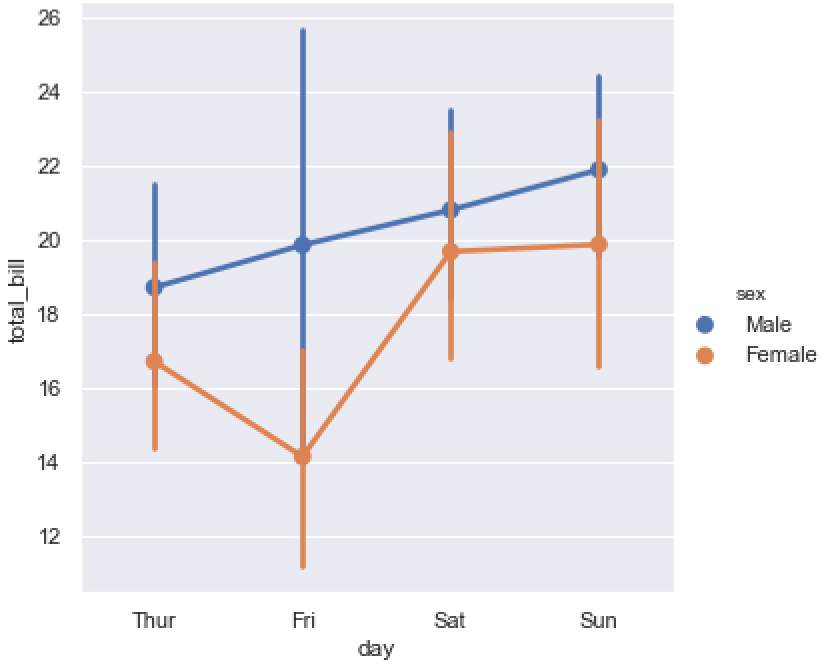
violin plot

On Friday we can see that female’s total_bill is much less than male’s total_bill.

Point plots

A point plot represents an estimate of central tendency for a numeric variable by the position of scatter plot points and provides some indication of the uncertainty around that estimate using error bars. Point plot shows only mean values and error rate surrounding those mean values. They are not very much informative but are easy to find the change in a variable based on different categories.

```
sns.catplot(x="day", y="total_bill", hue="sex", kind="point", 
data=tips);
```



Point plot

Using this plot it’s so simple to find changes in total_bill according to days. The total_bill is rising for male’s as the weekend arises while it decreases on Friday for females and jumps on Saturday and remains mostly constant on Sunday.

These plots can be used for various data analysis and we can infer information regarding relations between different variables and can help to extract more significant features from the dataset.

Towards Data Science

A Medium publication sharing concepts, ideas, and codes.

Follow

169

- Data Science

- Seaborn
- Python
- Machine Learning
- Matplotlib

169 claps