Fundamentals of Data science and Machine Learning

Concepts, Techniques and Tools to Build Intelligent Systems

Module 4

Basic Mathematical Concepts Behind ML Algorithms

Ali Samanipour

May. 2023

1. Why we Should Know Mathematics
2. Statistics
3. Types Of Data
4. Distributions

Why we should know Mathematics?

A Machine Learning practitioner **needs to know the Mathematical concepts** behind the working of any algorithm as it will enable her/him **to tune the model and later explain the working of the model**

Objective

Focus on in-depth analysis of different probabilistic distributions and the main points we can infer from data.

1. Why we Should Know Mathematics
2. Statistics
3. Types Of Data
4. Distributions

Statistics

In short, **is a study of data**. It is a field of science that helps to **conclude**, **extract facts** and figures **after analyzing the data**.

Statistical Mean: Population Mean(μ)

Population Mean(μ) is the mean or the average calculated for the entire set of data(N)

$$\mu = \frac{\sum_{i=0}^{N} x_i}{N}$$

Statistical Mean: Sample Mean

Sample Mean is the mean or the average calculated on a set of random variables(n)selected from the entire population(N).

$$\bar{x} = \frac{\sum_{i=0}^{N} x_i}{n}; where\ n \subset N$$

Median
(If the length of the list is odd)

Median is nothing but the **middle value**, which is separating the **sorted list** into two equal halves, upper half and a lower half

$$Median = \left(\frac{n+1}{2}\right)^{th} term$$

Median
(If the length of the list is even)

Median is nothing but the **middle value**, which is separating the **sorted list** into two equal halves, upper half and a lower half

$$Median = \frac{\left(\frac{n}{2}\right)^{th} term + \left(\frac{n}{2}+1\right)^{th} term}{2}$$

Population Variance

Variance is a measure between the variables that how are they different from one another and how much are they different from one another.

$$\sigma^2 = \frac{\sum_{i-0}^{N}\left(x_i - \mu\right)^2}{N}$$

Sample Variance

It shows how the dataset or the values **differ from the mean** of the dataset.

$$s^2 = \frac{\sum_{i-0}^{N}\left(x_i - \overline{x}\right)^2}{N-1}$$

Population Standard Deviation

is the root of the variance of the dataset

$$\sigma^2 = \sqrt{\frac{1}{N}\sum_{i-0}^{N}\left(x_i - \mu\right)^2}$$

Sample Standard Deviation

Is the root of the variance of the dataset

$$s = \sqrt{\frac{1}{N-1}\sum_{i-0}^{N}\left(x_i - \bar{x}\right)^2}$$

Probability theory

Probability is a branch of mathematics that can be defined as **the chance or likelihood that an event will occur**.

1. Why we Should Know Mathematics
2. Statistics
3. Types Of Data
4. Distributions

Different Types of Data

Data on the Nominal Scale means if we change the data for a record of this type, then it wouldn’t alter the nature of the collection.

Let’s consider a house. If we paint the house with different colors, we can confirm that it will still be a house.

Different Types of Data(Continue)

Data on the Ordinal Scale means the scale is ranked. Those numerical values (Ranks) only make sense when they are ordered that makes them ordinal scales

We have a rating system from 1 to 5, one being the worst/dissatisfied, and five being the best/satisfied. **These values have additional information** apart from the numerical value; it can also be considered as five different categories.

Measurement of Data

Discrete variables are also known as meristic variables, which are generally **counted**. It only takes discrete values which are represented by natural numbers.

For example human population numbers.

Measurement of Data

Continuous variables are floating-point numbers whose precision is limited by the tools we are using

To measure the thickness of a hair, say we have three different instruments regular centimeter ruler, a caliper, and a micrometer. All these instruments have different precision, the lowest precision being regular centimeter ruler, and the highest precision is the micrometer

Measurement of Data

If we imagine a normal scale where both ends are joined, that will give us **a circular scale.**

Some of the features will be an hour, day, month, annual dates, etc. Information extracted from these data like differences, ratios are not sensible derivatives, but for these features, we have other methods to analyze them.

Measurement of Data

The ratio scale is the ratio between two variables that will give a piece of information.

For baking a cake, say the ratio of flour, water, and butter is 5:2:1, i.e., if we use 5 KG of flour, then we need 2 KG of water and 1 KG of butter.

Let's Start Our Project (World Happiness Report)

World Happiness Report is an initiative taken by the United Nations, where 156 countries are surveyed on how happy their citizens perceive themselves to be.

```
hr.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 158 entries, 0 to 157
Data columns (total 12 columns):
Country                          158 non-null object
Region                           158 non-null object
Happiness Rank                   158 non-null int64
Happiness Score                  158 non-null float64
Standard Error                   158 non-null float64
Economy (GDP per Capita)         158 non-null float64
Family                           158 non-null float64
Health (Life Expectancy)         158 non-null float64
Freedom                          158 non-null float64
Trust (Government Corruption)    158 non-null float64
Generosity                       158 non-null float64
Dystopia Residual                158 non-null float64
dtypes: float64(9), int64(1), object(2)
memory usage: 14.9+ KB
```

Let’s Start Our Project (World Happiness Report)

The main aim of this kind of data is to observe how happiness has evolved over the past years considering technology, conflicts, and government policies, social norms

```
hr.head(1).T
```

	0
Country	Switzerland
Region	Western Europe
Happiness Rank	1
Happiness Score	7.587
Standard Error	0.03411
Economy (GDP per Capita)	1.39651
Family	1.34951
Health (Life Expectancy)	0.94143
Freedom	0.66557
Trust (Government Corruption)	0.41978
Generosity	0.29678
Dystopia Residual	2.51738

1. Why we Should Know Mathematics
2. Statistics
3. Types Of Data
4. Distributions

Distributions

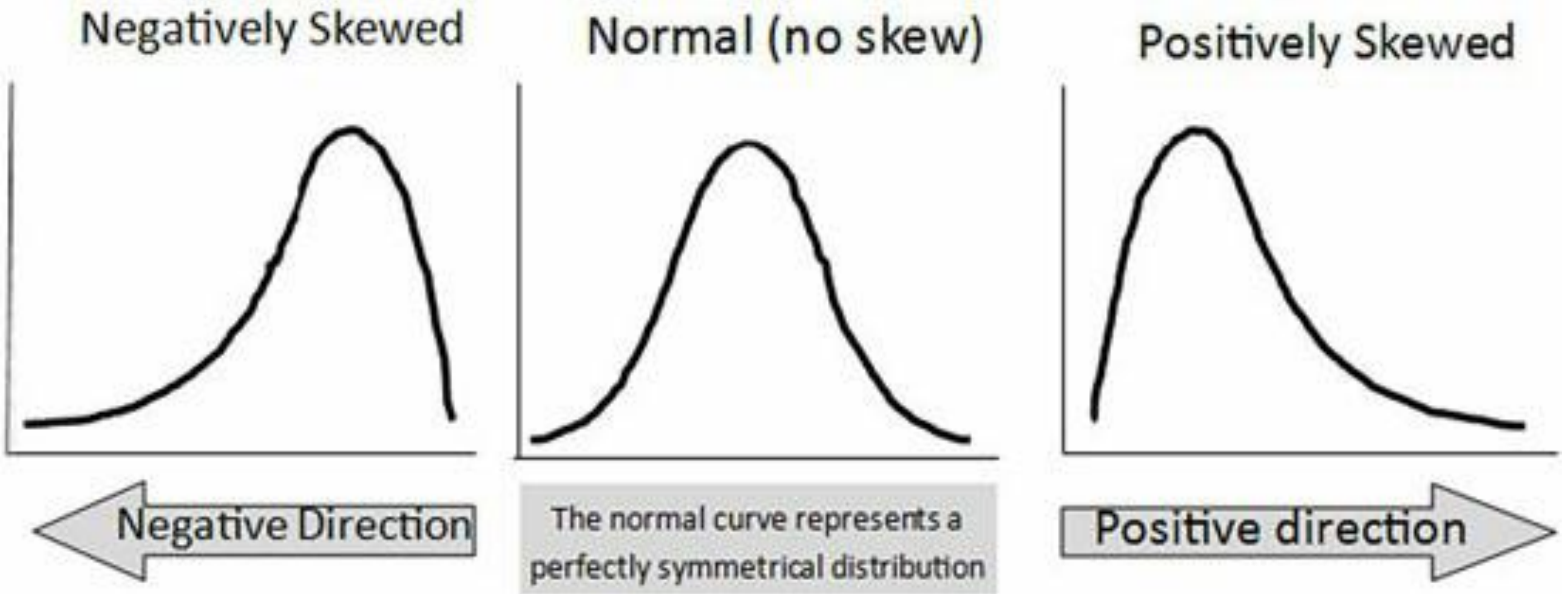
The distribution of a statistical dataset is the plot that shows us the **frequency of occurrence** in the dataset.

```
hr.describe().T
```

	count	mean	std	min	25%	50%	75%	max
Happiness Rank	158.0	79.493671	45.754363	1.00000	40.250000	79.500000	118.750000	158.00000
Happiness Score	158.0	5.375734	1.145010	2.83900	4.526000	5.232500	6.243750	7.58700
Standard Error	158.0	0.047885	0.017146	0.01848	0.037268	0.043940	0.052300	0.13693
Economy (GDP per Capita)	158.0	0.846137	0.403121	0.00000	0.545808	0.910245	1.158448	1.69042
Family	158.0	0.991046	0.272369	0.00000	0.856823	1.029510	1.214405	1.40223
Health (Life Expectancy)	158.0	0.630259	0.247078	0.00000	0.439185	0.696705	0.811013	1.02525
Freedom	158.0	0.428615	0.150693	0.00000	0.328330	0.435515	0.549092	0.66973
Trust (Government Corruption)	158.0	0.143422	0.120034	0.00000	0.061675	0.107220	0.180255	0.55191
Generosity	158.0	0.237296	0.126685	0.00000	0.150553	0.216130	0.309883	0.79588
Dystopia Residual	158.0	2.098977	0.553550	0.32858	1.759410	2.095415	2.462415	3.60214

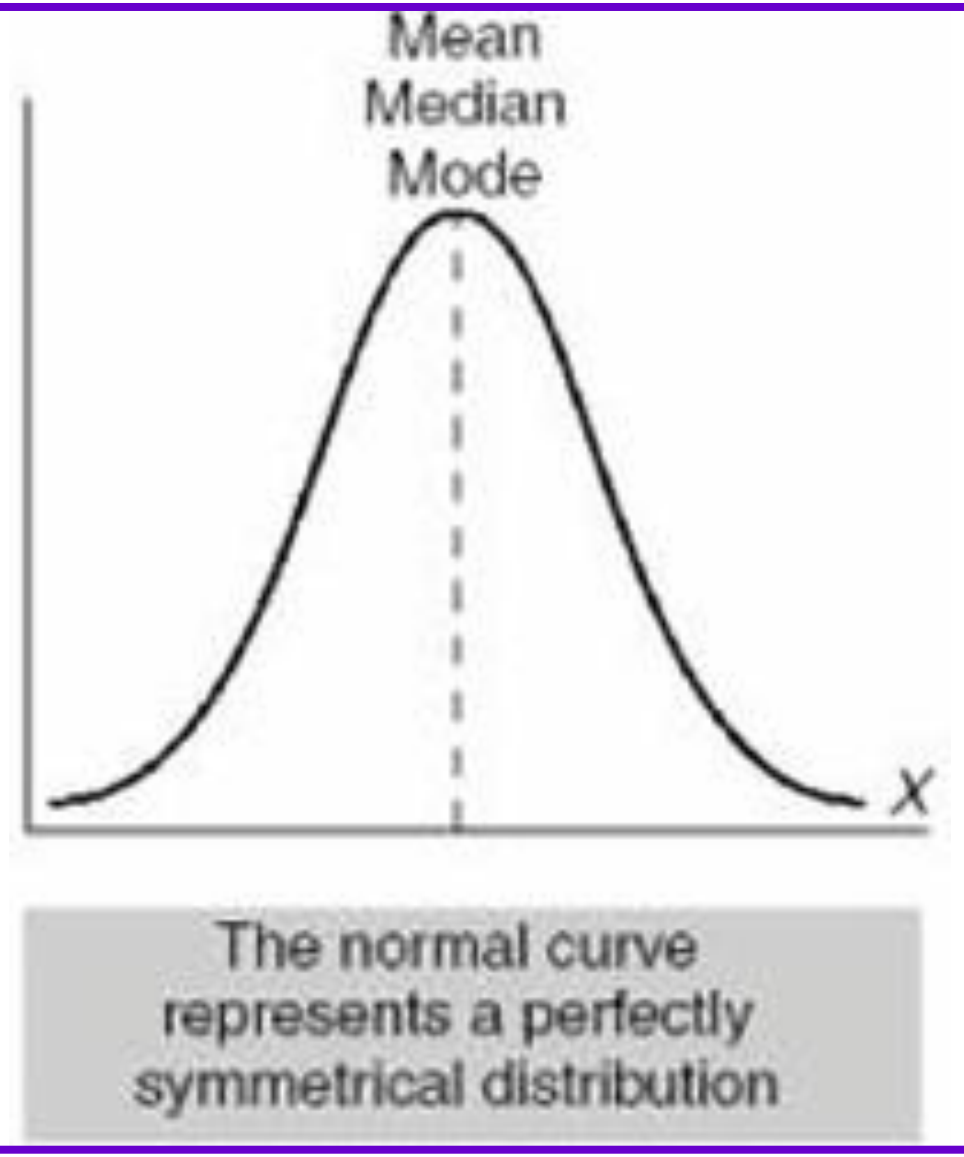
Distribution patterns

For continuous variables, we generally get some below curves or some curves which will resemble any one of the below curves.



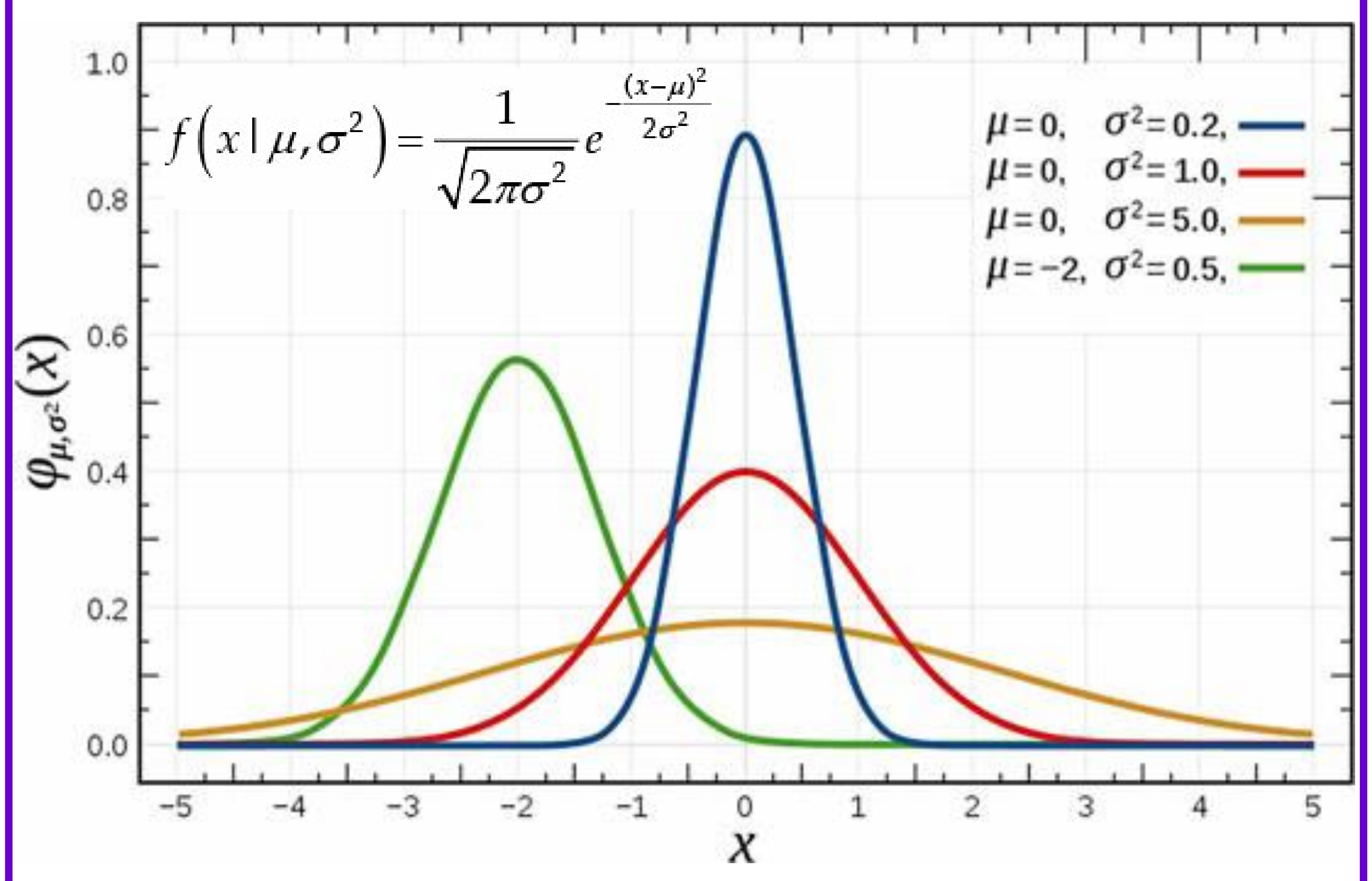
Normal Distribution (Gaussian Distribution or Bell Curve)

The **normal distribution** is a type of probability distribution that is **symmetric** about the mean of the data, i.e., the **density of the data near the mean value is higher than the tails**



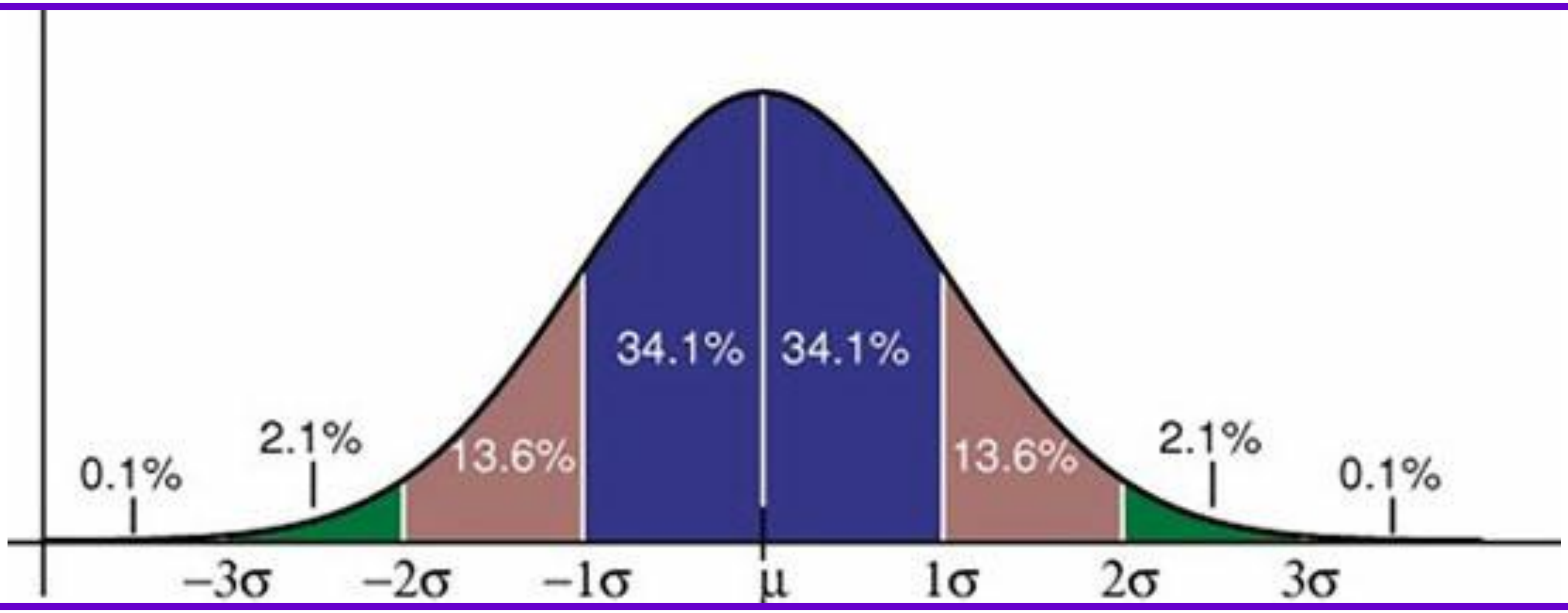
Gaussian Distribution

We can see different behaviors of the Gaussian distribution, with different values of μ and σ^2.



Empirical Rule

The Empirical Rule states that for a Gaussian/Normal distribution, around 68.2% will fall between the first standard deviation(all the numbers are an approximation)

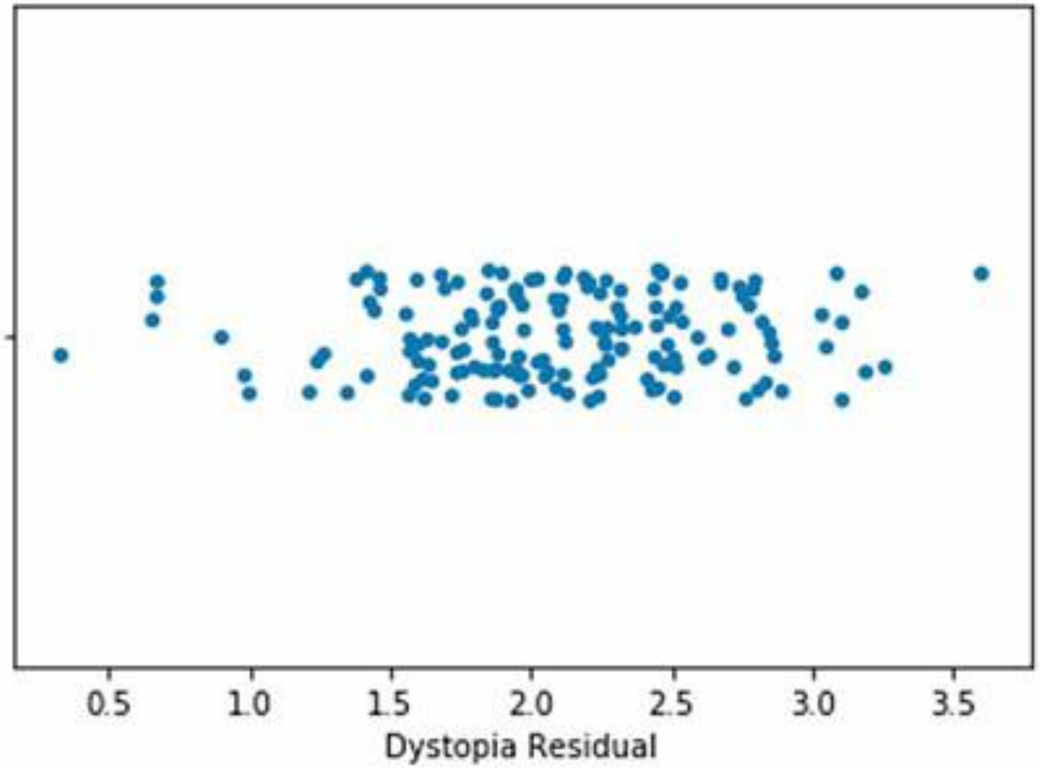


Distribution Plot

To visualize how the feature “Dystopia Residual” will look in terms of distribution, we need to plot a frequency distribution.

```
sns.stripplot(hr["Dystopia Residual"])
```

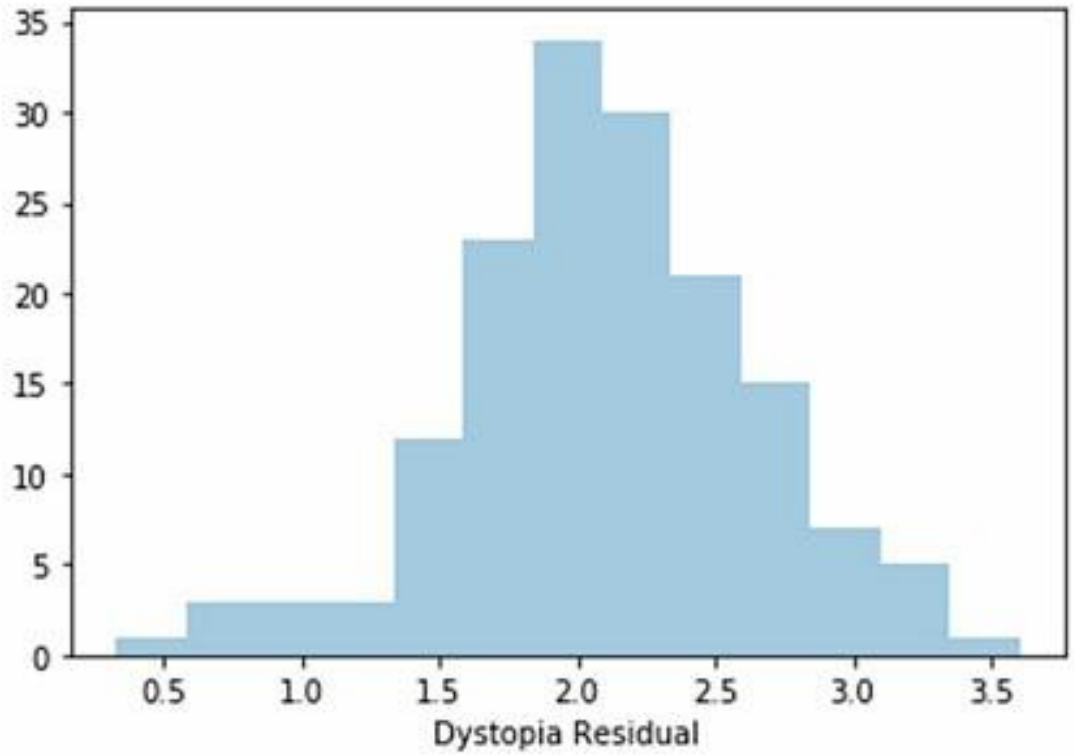
```
<matplotlib.axes._subplots.AxesSubplot at 0x1a18f60e48>
```



Normal Distribution Plot

```
sns.distplot(hr["Dystopia Residual"],kde=False)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x1a2461cf98>
```



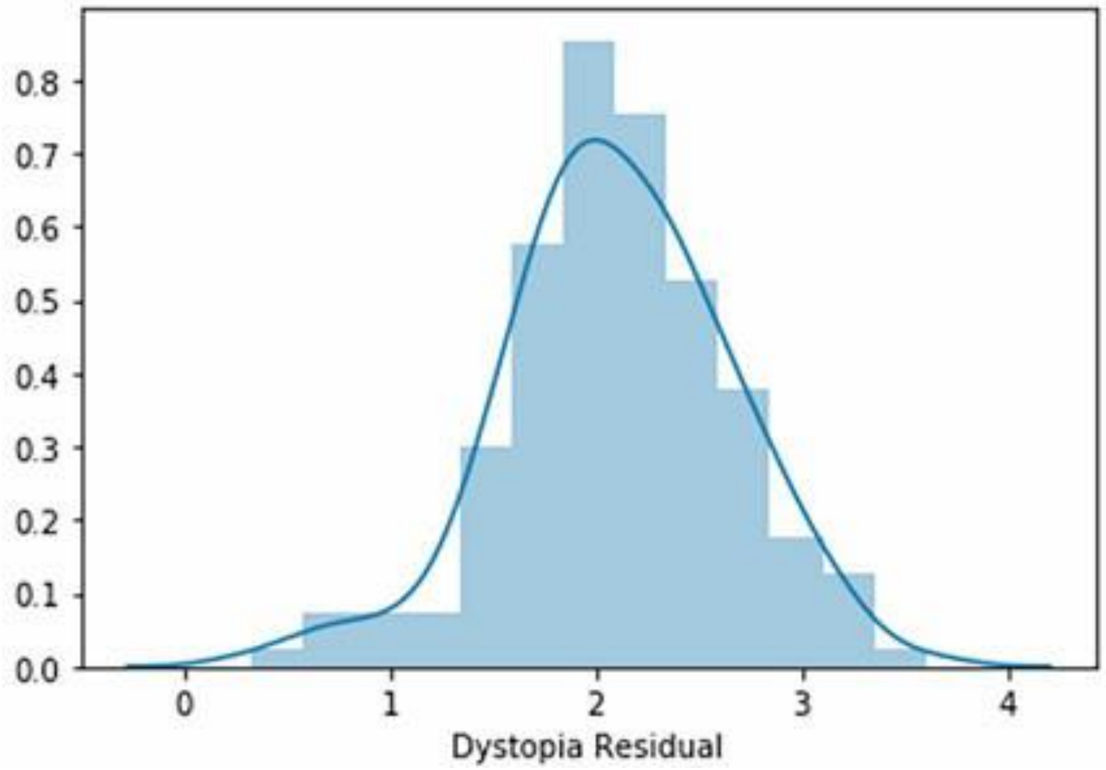
Seeing the above diagram, it is tough to predict the distribution, so we need to smooth the histogram to get the required curve.

KDE with Histogram for Dystopia Residual

Kernel Density Estimation is commonly also known as KDE, which is a way to create a smooth curve given a dataset using a density function.

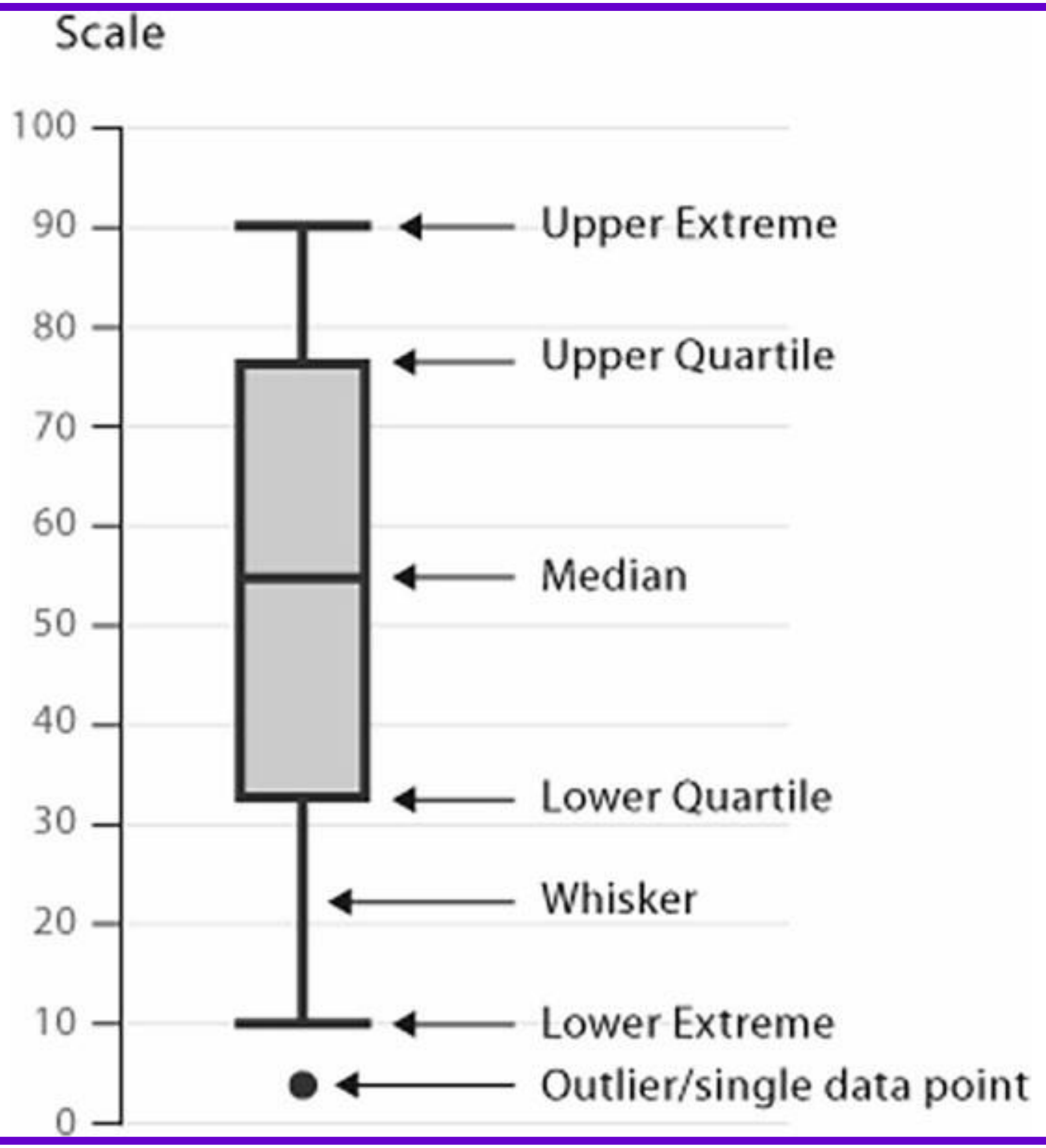
```python
sns.distplot(hr["Dystopia Residual"],kde=True)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x1a194471d0>
```



Box Plot

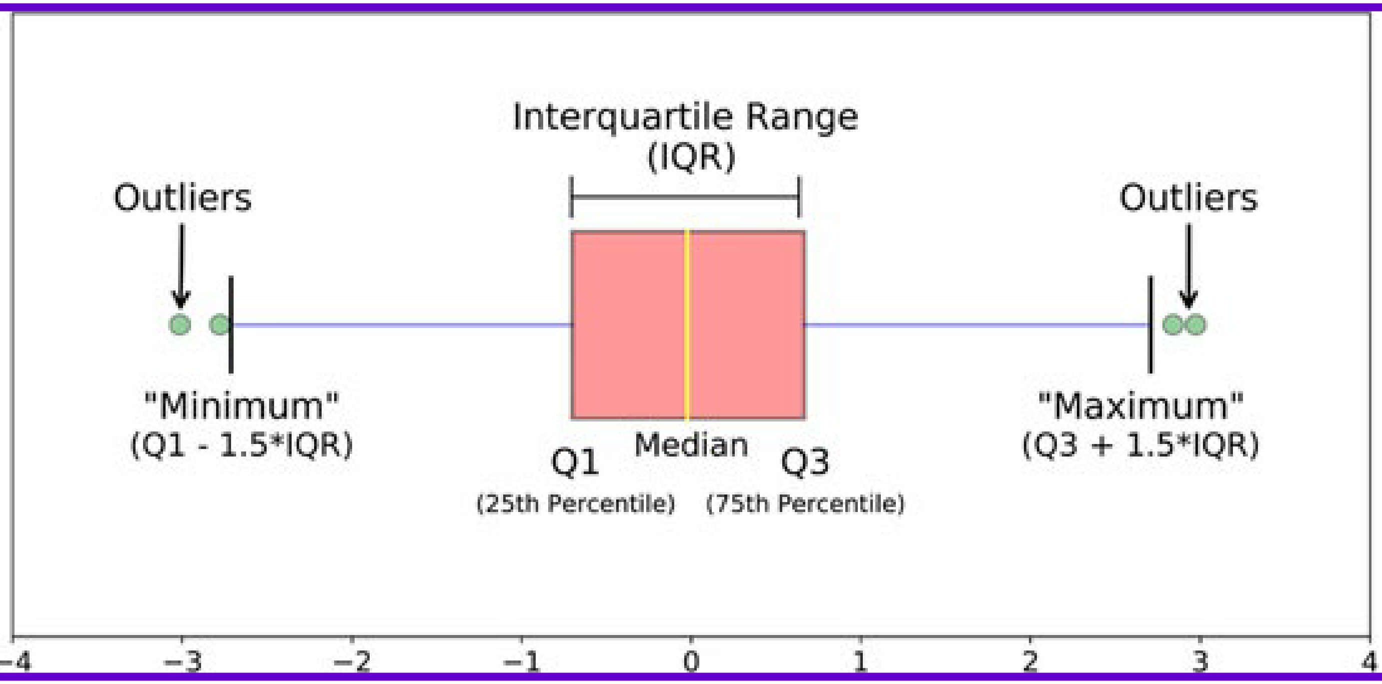
Box Plot gives us information like the outliers, **symmetry** of the data, **how are the data grouped**, and the **skewness** of the data.



Box Plot Calculations

Let us take a series of numbers:

{3, 7, 8, 5, 12, 14, 21, 13, 18}

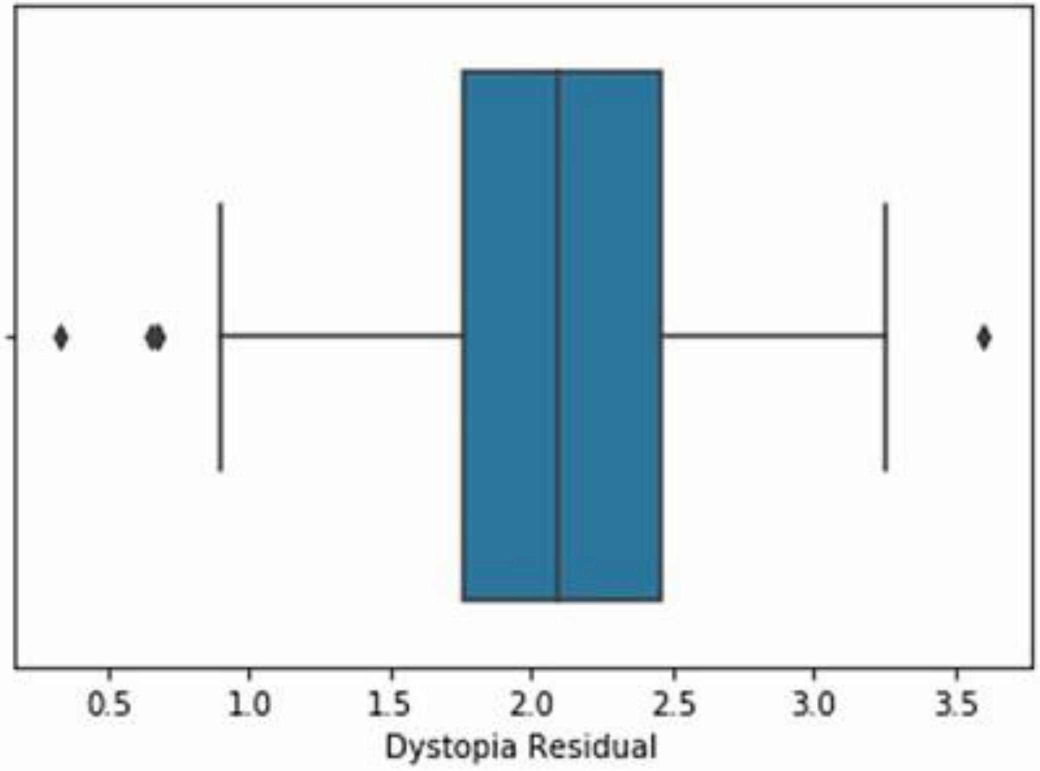


Box-plot for Dystopia Residual variable

We can say that there are some data points on both left and right sides, which are considered outliers.

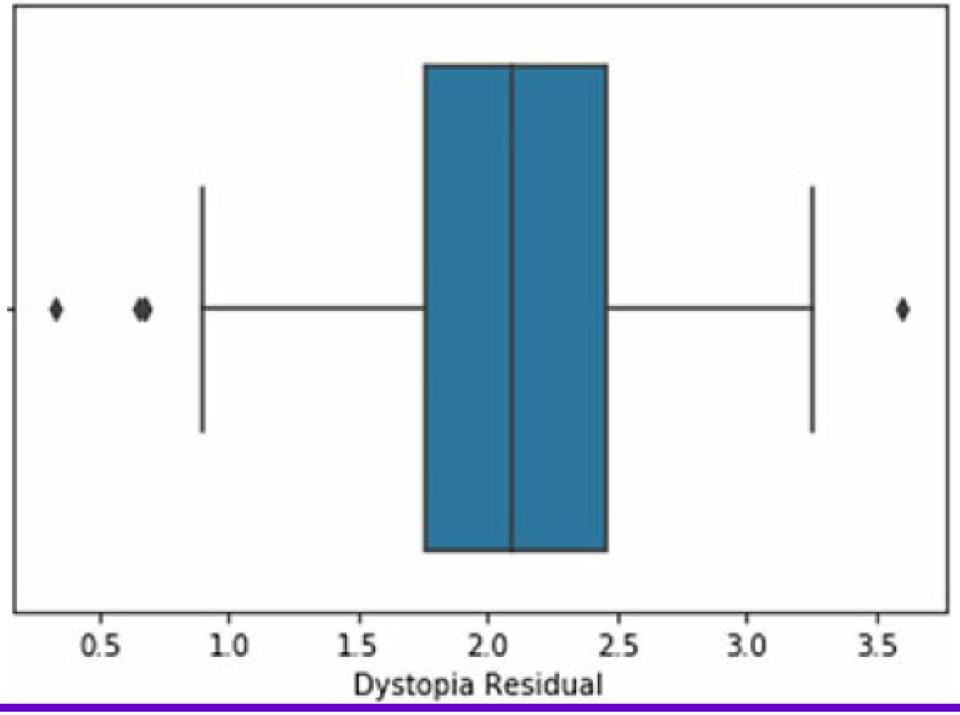
```
sns.boxplot(hr["Dystopia Residual"])
```

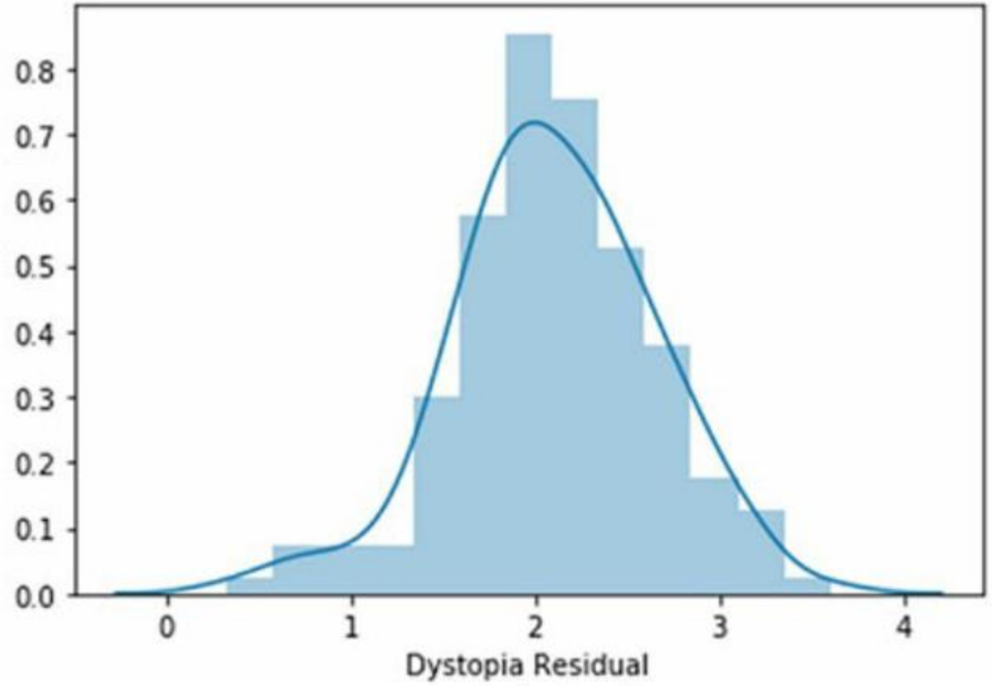
```
<matplotlib.axes._subplots.AxesSubplot at 0x1a2456d7b8>
```



Interpret the figures, Compare Box Plot with Distribution Pattern

If a random value is chosen then the **probability to get the value which will lie in the outliers will be rare and probability and** to **get to the value between the range ~1.8 to ~2.5 is very high**.





Box-plot calculation

As the box-and-whiskers plot only shows that there are outliers but not the values of the outliers.

```
import numpy as np
def box_plot_calculation(data):
    data = data.values
    q25, q75 = np.percentile(data, 25), np.percentile(data, 75)
    print('Quartile 25: {} | Quartile 75: {}'.format(q25, q75))
    IQR = q75 - q25
    print('IQR: {}'.format(IQR))

    cut_off = IQR * 1.5
    MIN, MAX = q25 - cut_off, q75 + cut_off
    print('Cut Off: {}'.format(cut_off))
    print('Minimum: {}'.format(MIN))
    print('Maximum: {}'.format(MAX))

    outliers = [x for x in data if x < MIN or x > MAX]
    outliers.sort()
    print('Feature Outliers: {}'.format(len(outliers)))
    print('Outliers:{}'.format(outliers))
```

```
box_plot_calculation(hr["Dystopia Residual"])
```

```
Quartile 25: 1.75941 | Quartile 75: 2.4624149999999996
IQR: 0.7030049999999997
Cut Off: 1.0545074999999995
Minimum: 0.7049025000000004
Maximum: 3.516922499999999
Feature Outliers: 5
Outliers:[0.32858000000000004, 0.6542899999999999, 0.67042, 0.67108, 3.6021400000000003]
```

Skewed Distribution

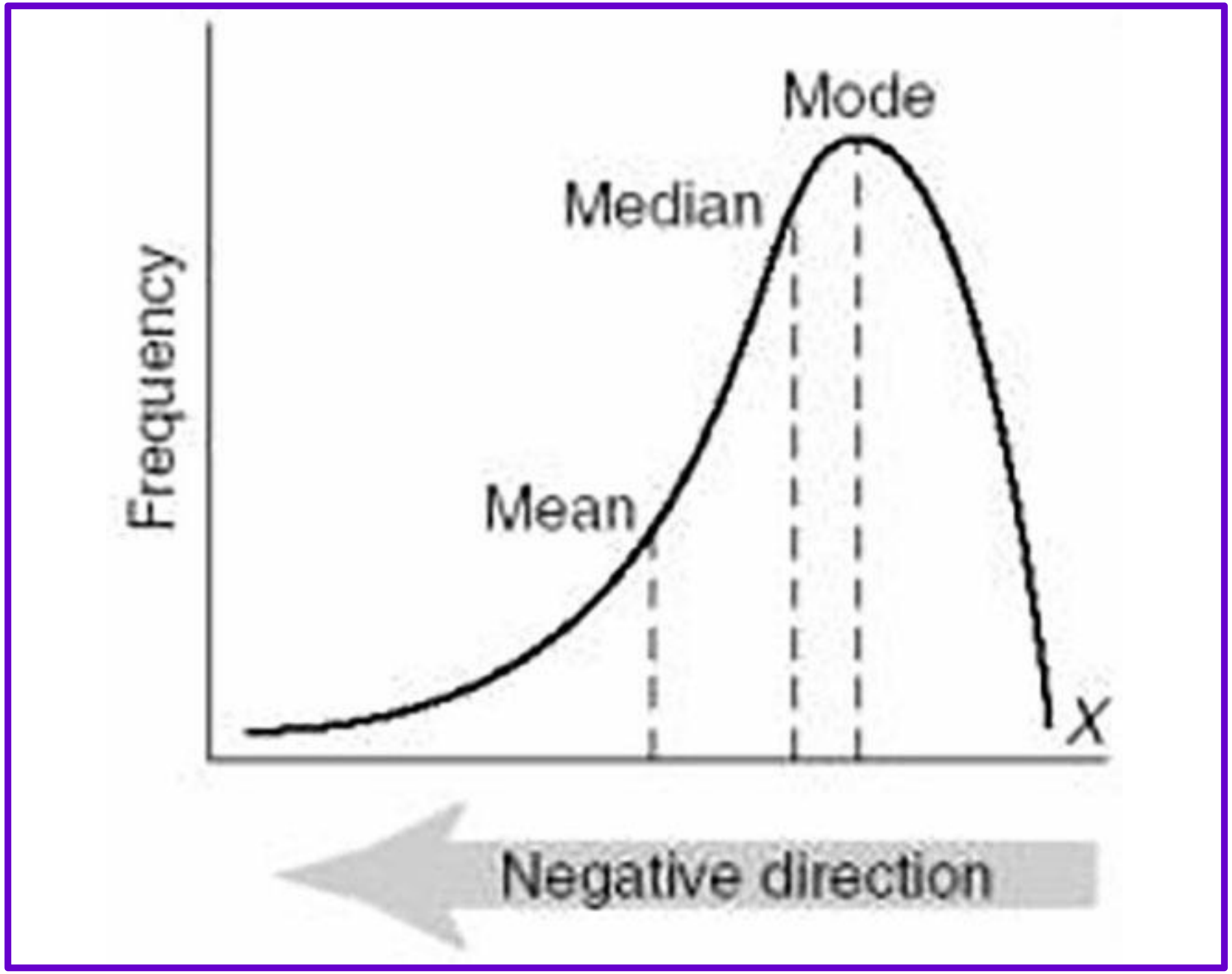
A distribution is said to be a **skewed** distribution when the data points are dense either towards the left or the right side of the curve.

```
hr.skew()
```

```
Happiness Rank                     0.000418
Happiness Score                    0.097769
Standard Error                     1.983439
Economy (GDP per Capita)          -0.317575
Family                            -1.006893
Health (Life Expectancy)          -0.705328
Freedom                           -0.413462
Trust (Government Corruption)      1.385463
Generosity                         1.001961
Dystopia Residual                 -0.238911
dtype: float64
```

Left Skewed
(Negative Skewness)

That indicated less frequency on the left, and the frequency gradually increases towards the right.

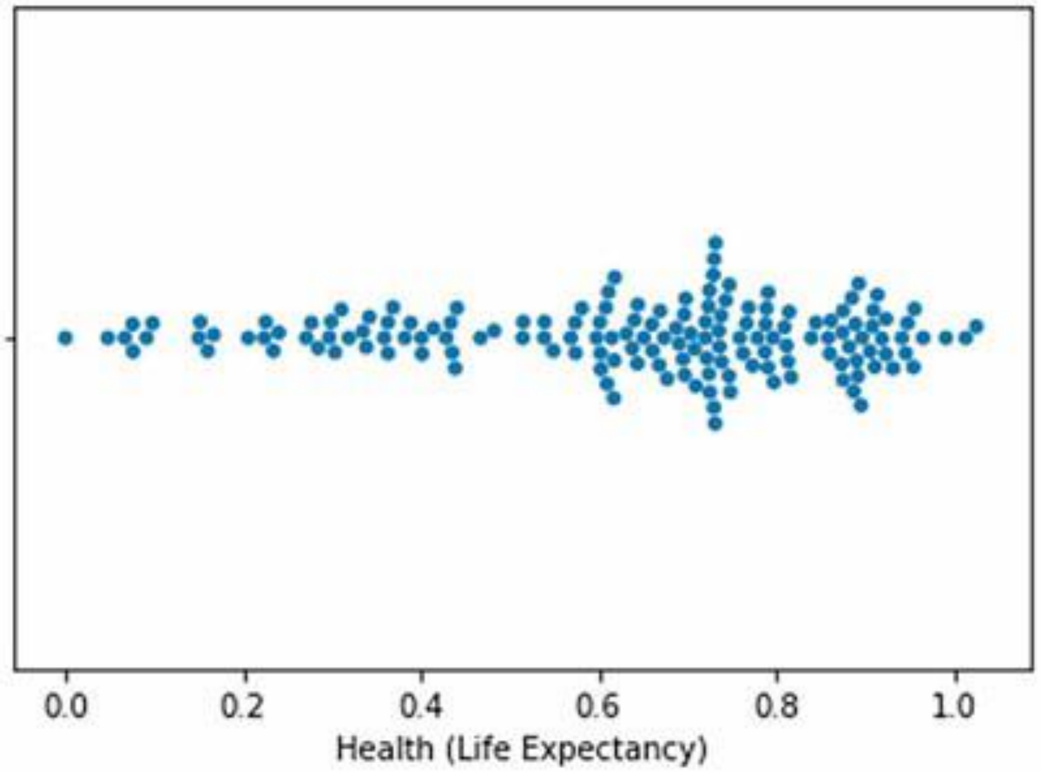


Swarm Plot for Health (Life Expectancy)

The Swarm plot gives us the same distribution, but the difference is that we can see the individual data points that give us an idea of the density for the data points.

```
sns.swarmplot(hr["Health (Life Expectancy)"])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x1a238676d8>
```

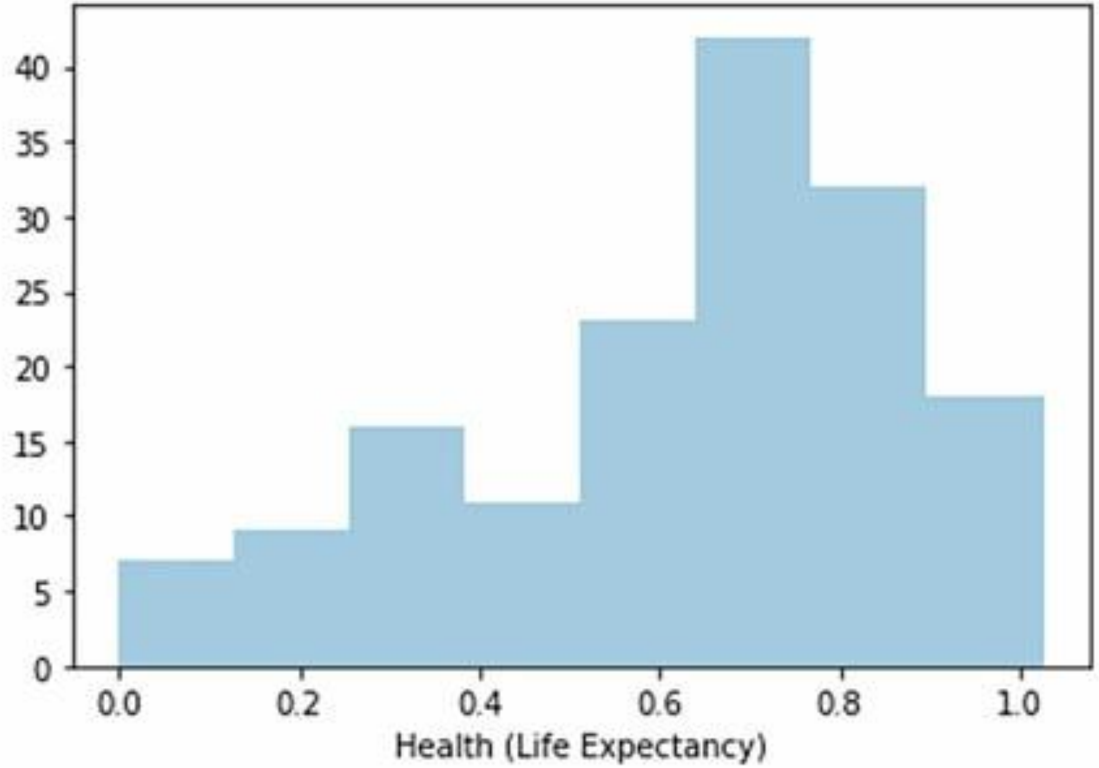


Frequency Distribution Plot

The frequency distribution is more flushed towards the right side of the chart. This is what is unique about the left-skewed distribution.

```
sns.distplot(hr["Health (Life Expectancy)"],kde=False)
```

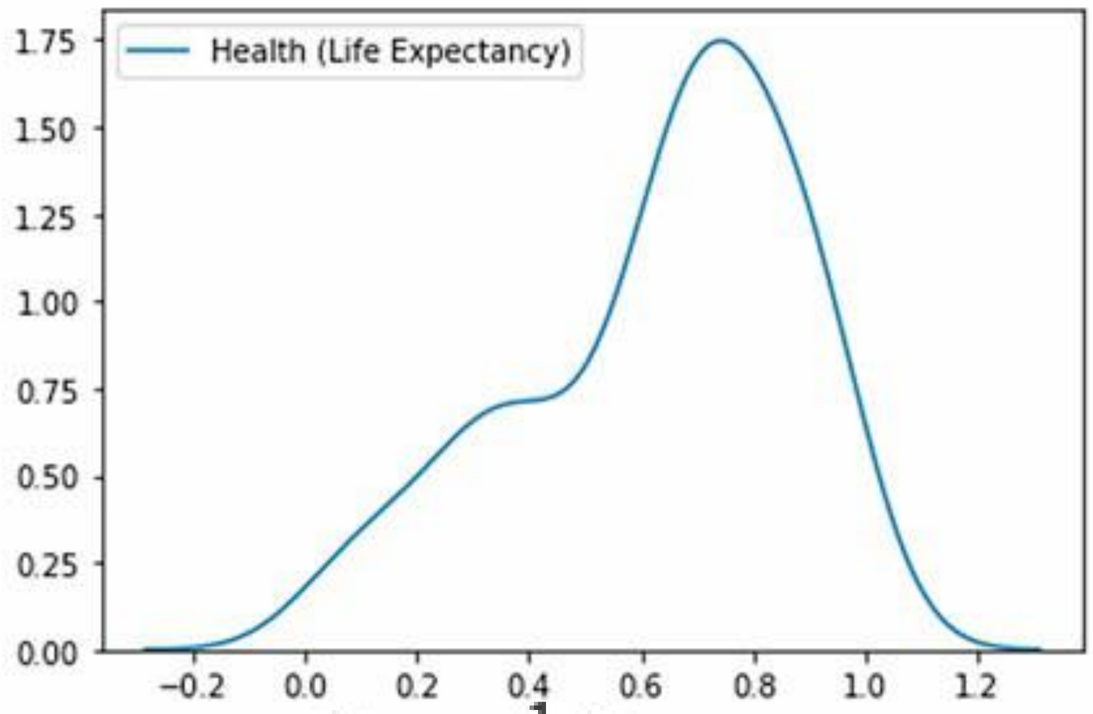
```
<matplotlib.axes._subplots.AxesSubplot at 0x1a2477bf28>
```



Kernel Density Estimation (Gaussian Kernel)

The Gaussian curve is smooth, and the intricate details for the density are missing.

```
sns.kdeplot(hr["Health (Life Expectancy)"], kernel="gau")

<matplotlib.axes._subplots.AxesSubplot at 0x1a25b44cc0>
```

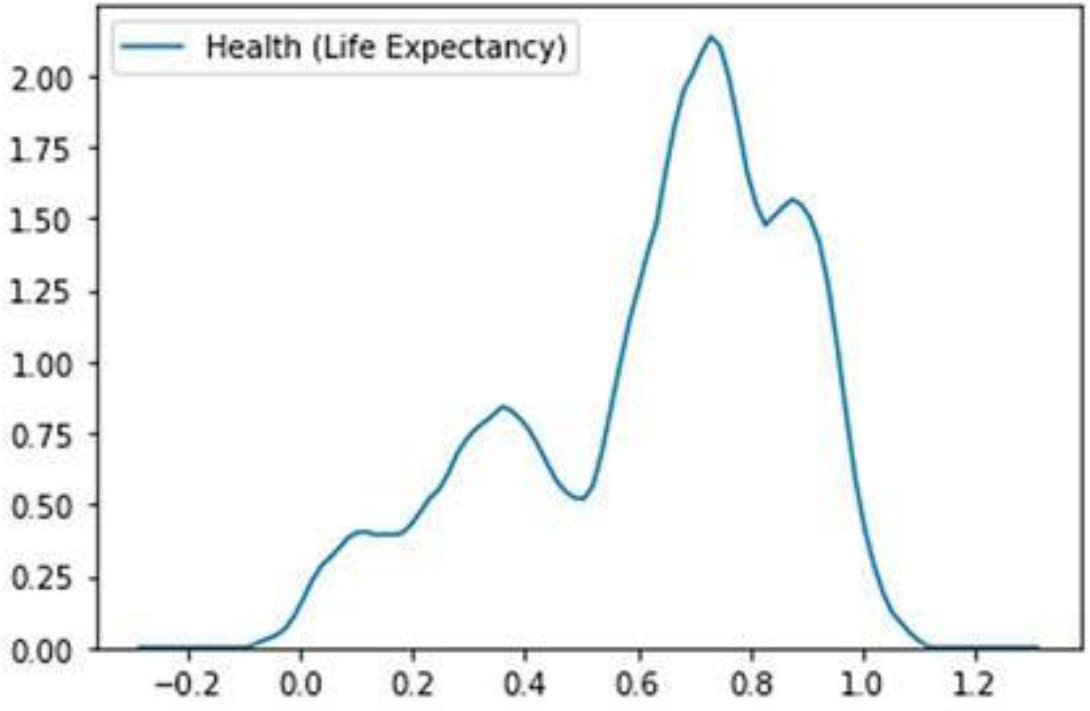


$$K(x)=\frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}x^2}\;;where\;\mu=0,\sigma^2=1$$

Kernel Density Estimation (Cosine Kernel)

The Cosine Kernel will give a more accurate density compared to the Gaussian plot.

```
sns.kdeplot(hr["Health (Life Expectancy)"], kernel="cos")
<matplotlib.axes._subplots.AxesSubplot at 0x1a2561ff60>
```



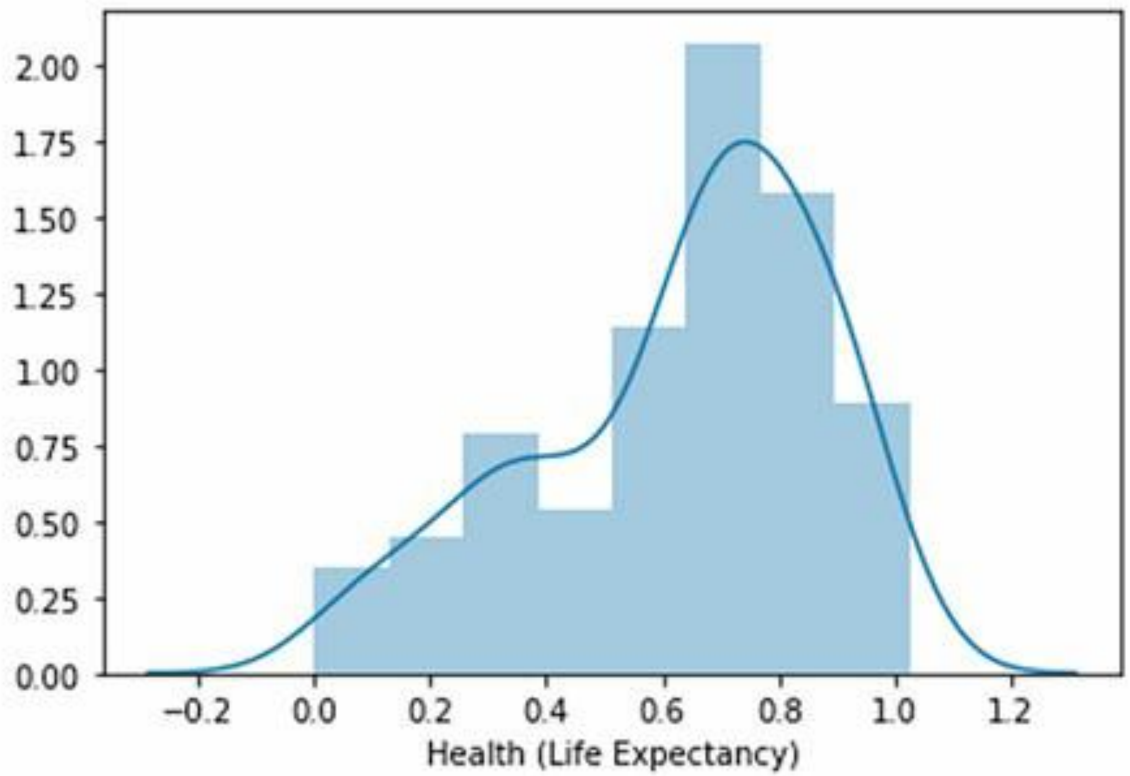
$$K(x) = \frac{\pi}{4}\cos\left(\frac{\pi}{2}x\right)$$

KDE with Histogram for Left Skewed Data

But we will mostly use Gaussian Kernel with the histogram

```
sns.distplot(hr["Health (Life Expectancy)"],kde=True)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x1a2364fa58>
```



Box-Plot Calculations for Health (Life Expectancy)

Here we can see that there are no outliers from the output of the function “**box_plot_claculatio**”

```
box_plot_calculation(hr["Health (Life Expectancy)"])
```

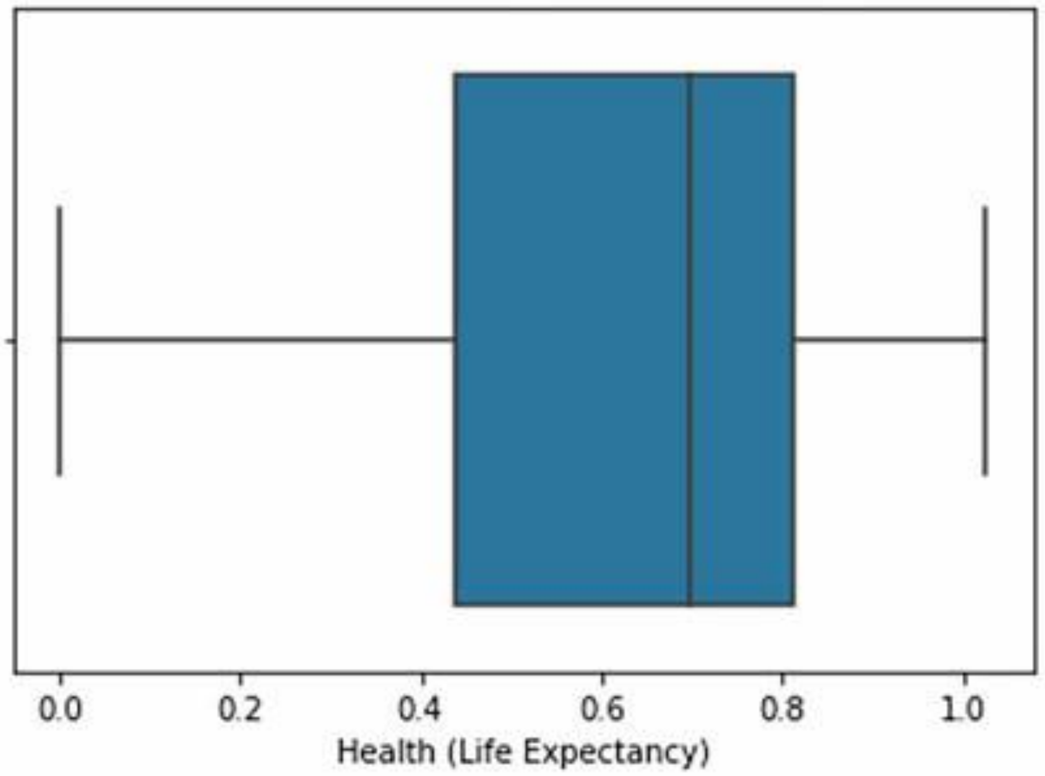
```
Quartile 25: 0.439185 | Quartile 75: 0.8110125
IQR: 0.37182750000000003
Cut Off: 0.5577412500000001
Minimum: -0.11855625000000009
Maximum: 1.3687537500000002
Feature Outliers: 0
Outliers:[]
```

Box-Plot for Health (Life Expectancy)

We can confirm that there are no outliers, and the shape of the plot is different from the last distribution.

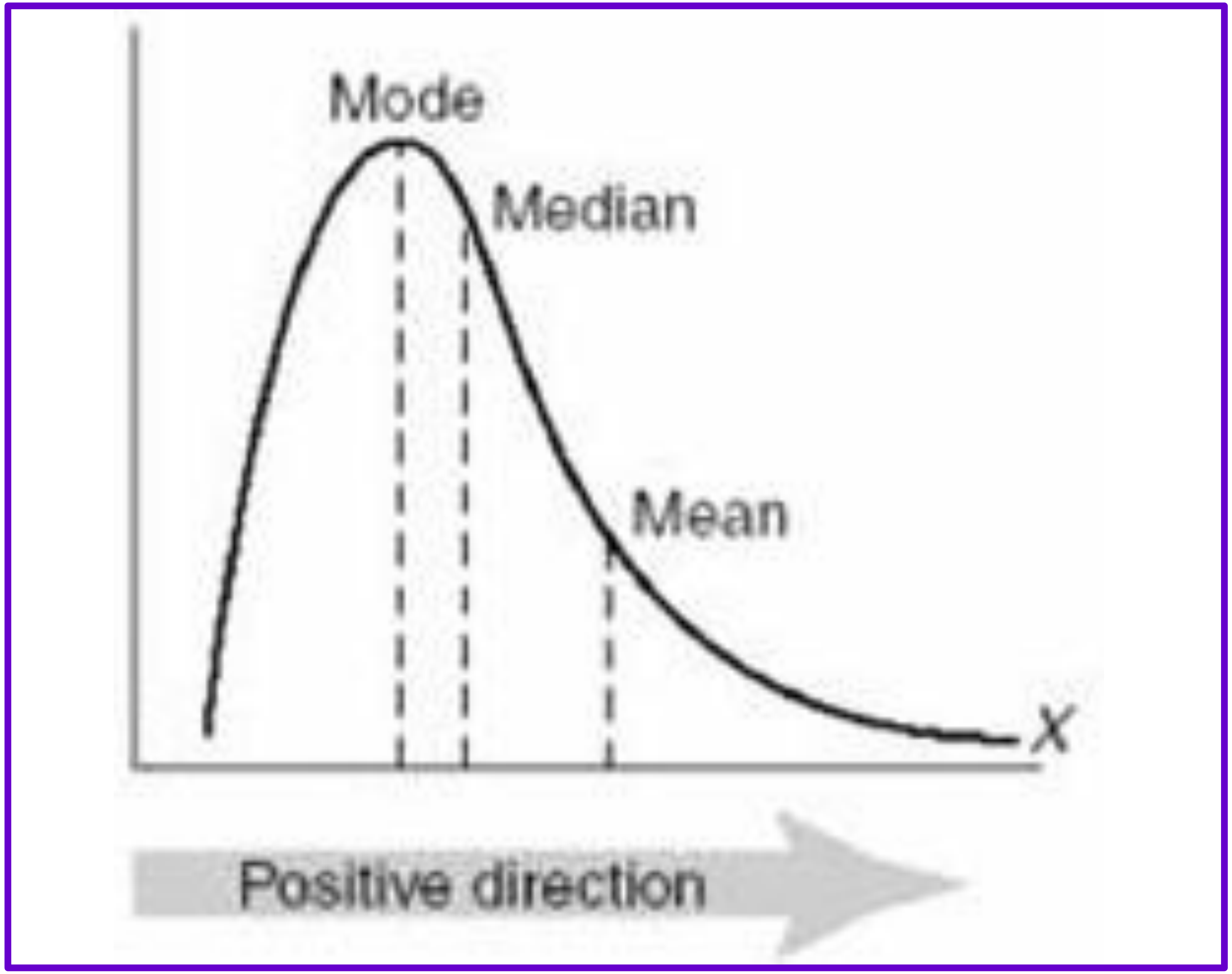
```
sns.boxplot(hr["Health (Life Expectancy)"])
```

<matplotlib.axes._subplots.AxesSubplot at 0x1a2370a9e8>



Right Skewed (Positive Skewness)

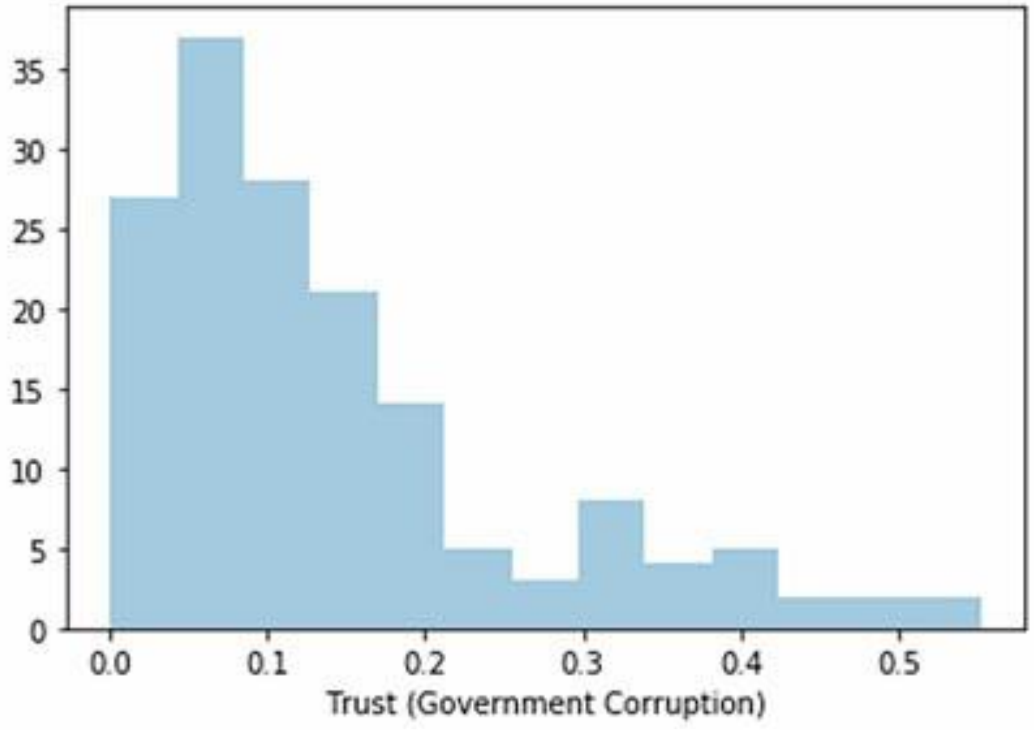
We can clearly say that the density of the values is more towards the left side, and there is a long tail at the right end.



Distribution Plot

We can see that the density of the data points is concentrated towards the left side of the distribution, and we have a tail on the right side

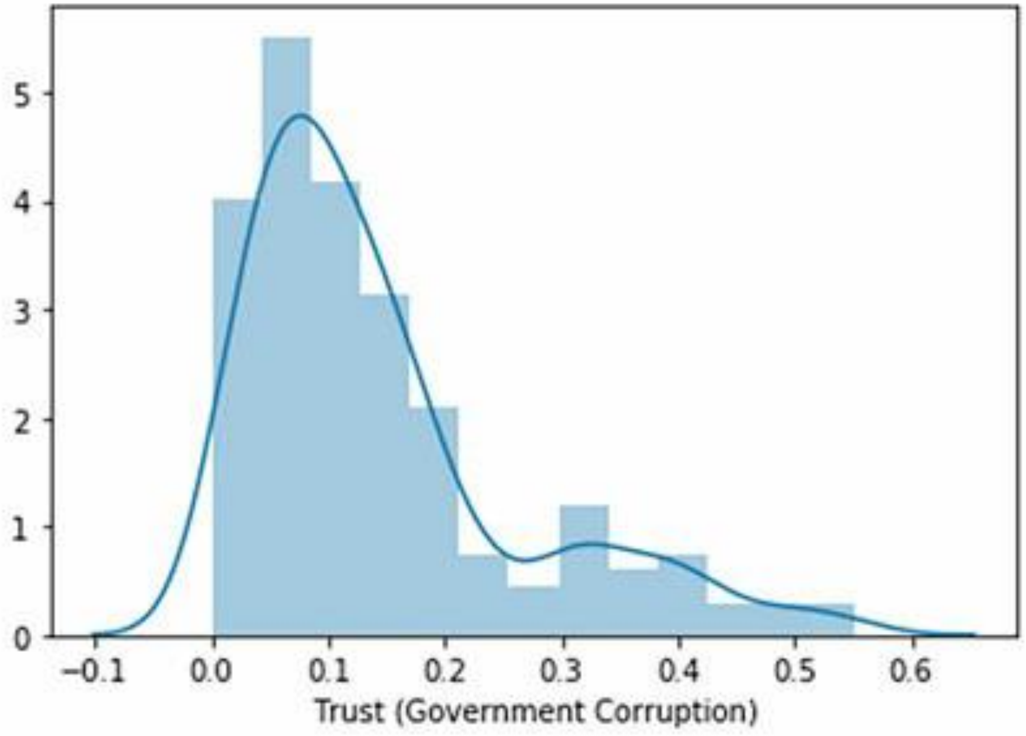
```
sns.distplot(hr["Trust (Government Corruption)"],kde=False)

<matplotlib.axes._subplots.AxesSubplot at 0x1a1986b0b8>
```



KDE with Default bandwidth estimator

If we overlay the optimal curve from KDE using the Scott estimator, this will look something like the below figure

```
sns.distplot(hr["Trust (Government Corruption)"],kde=True)
<matplotlib.axes._subplots.AxesSubplot at 0x1a19b2a080>
```



Box Plot Calculation for Right Skewed Data

We had written a code snippet before, and to find out the values related to the Box-Plot to create it, let us execute that and see all the values

```
box_plot_calculation(hr['Trust (Government Corruption)'])
```

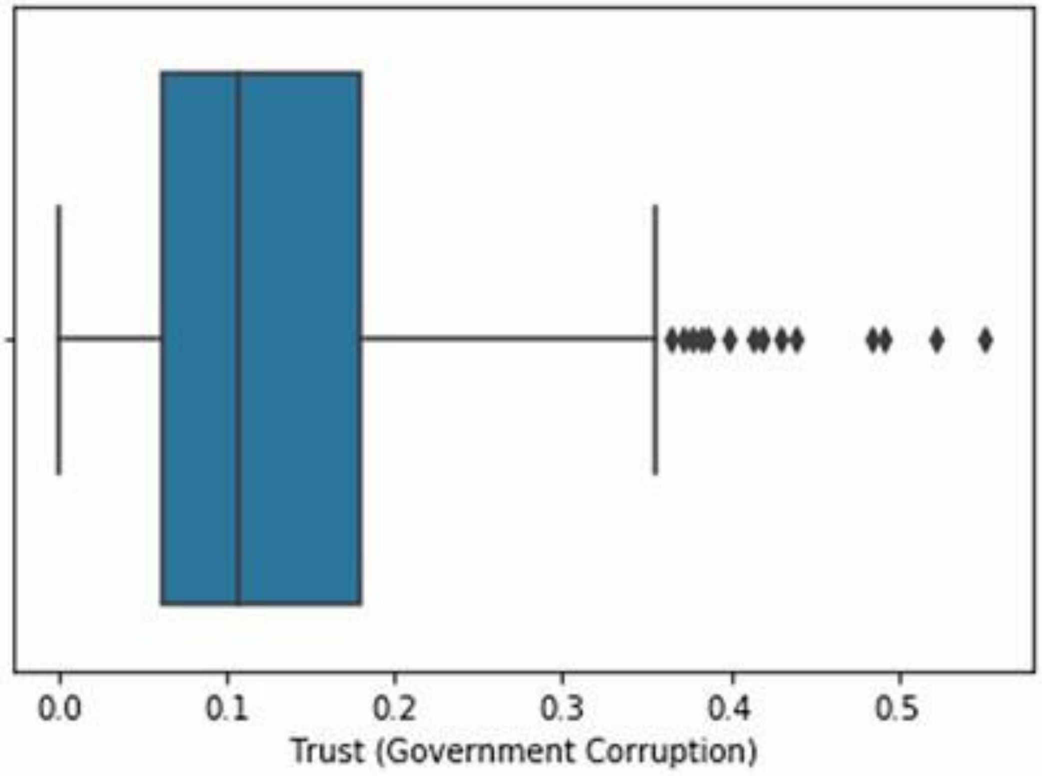
```
Quartile 25: 0.061675 | Quartile 75: 0.18025500000000003
IQR: 0.11858000000000002
Cut Off: 0.17787000000000003
Minimum: -0.11619500000000002
Maximum: 0.358125
Feature Outliers: 14
Outliers:[0.36503, 0.37124, 0.37798000000000004, 0.38331, 0.38583, 0.39928, 0.41372, 0.41978000000000004, 0.42922, 0.
43843999999999994, 0.48357, 0.4921, 0.52208, 0.55191]
```

Box Plot Trust (Govt. Corruption)

With all the derived details, we plotted the box plot, and we can see quite a lot of outliers, and all the outliers are above the “Maximum” so, we are expecting to see all the outliers on the righthand side of the plot.

```
sns.boxplot(hr["Trust (Government Corruption)"])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x1a19c1a198>
```

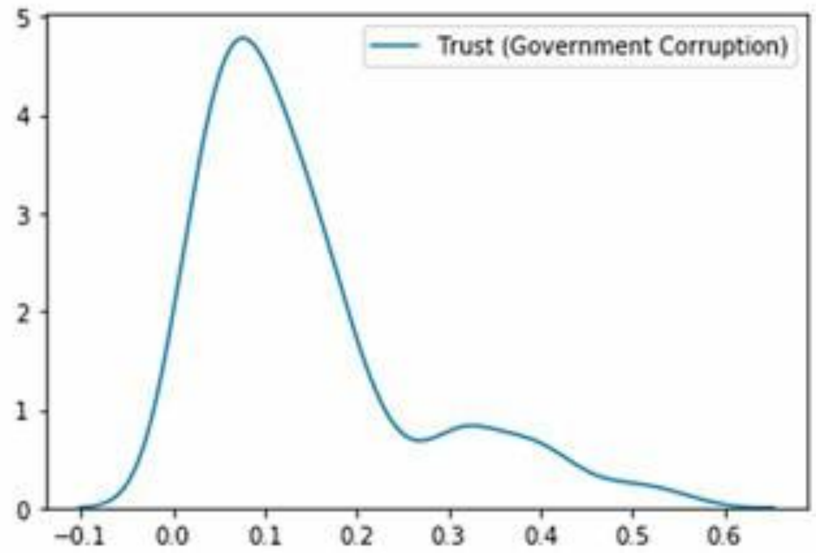


Gaussian Kernel with "scott"

By default, KDE uses "scott" to estimate the bandwidth (h), and it is one of the most popular methods

```
sns.kdeplot(hr["Trust (Government Corruption)"], kernel="gau", bw='scott')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x1a18fb7710>
```

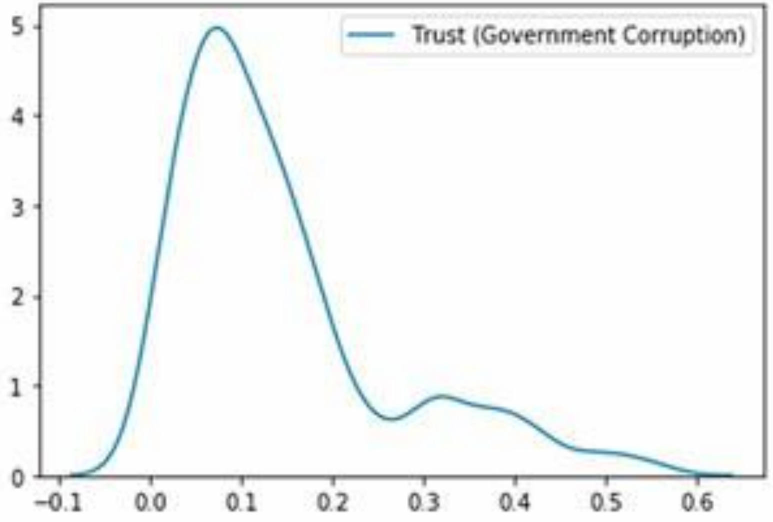


Gaussian Kernel with "silverman"

"scott" and "silverman" are similar, and results are similar as well. We can also take a look at "silverman" bandwidth estimator

```
sns.kdeplot(hr["Trust (Government Corruption)"], kernel="gau", bw='silverman')
```

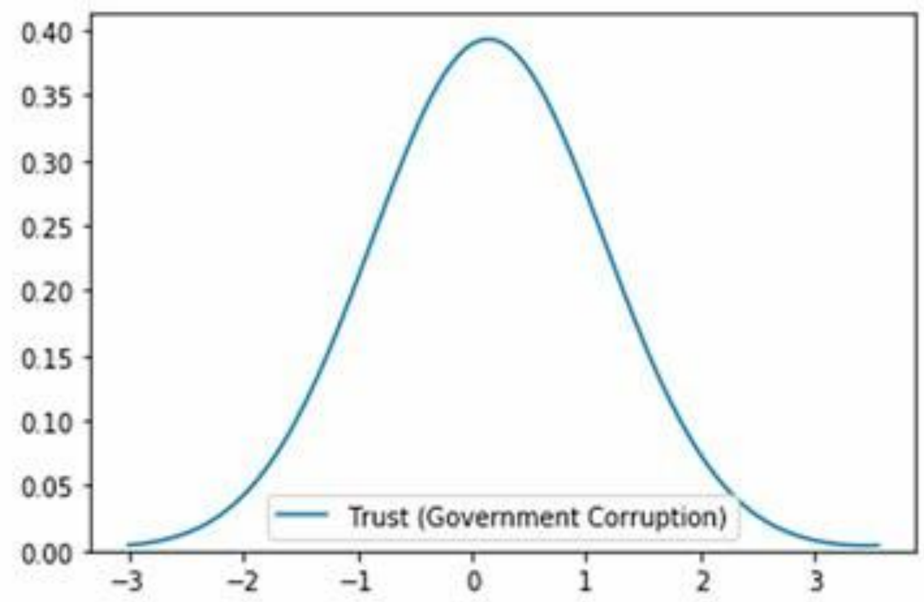
```
<matplotlib.axes._subplots.AxesSubplot at 0x1a1b6a4da0>
```



KDE with Bandwidth=1

If the value of the bandwidth is high like bandwidth = 1 or 2, then it over-smooths the curve, and we won’t get any information from it, this is like we have a binned frequency table with only one bin

```
sns.kdeplot(hr["Trust (Government Corruption)"], kernel="gau", bw=1)

<matplotlib.axes._subplots.AxesSubplot at 0x1a255157f0>
```

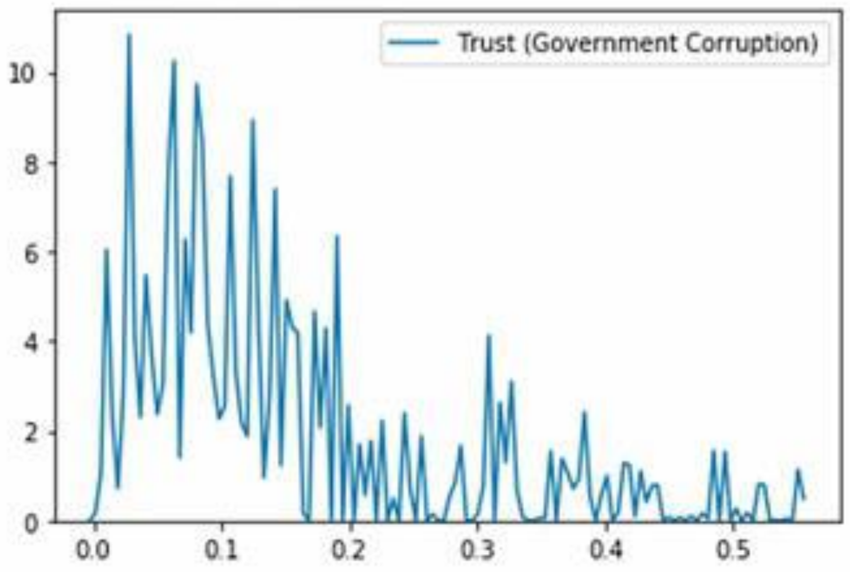


KDE with Bandwidth=0.001

It will be a similar problem if bandwidth = 0.001 then the problem of under-smoothing

```
sns.kdeplot(hr["Trust (Government Corruption)"], kernel='gau', bw=0.001)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x1a25a8b7f0>
```



Mean Integrated Squared Error (MISE) (Silverman's rule)

We need to strive for the optimal value of bandwidth, which will select the right number of bins to plot the histogram and then smooth the histogram

$$\Rightarrow E \| f_h - f \|_2^2 \qquad \ldots(i)$$

$$MISE(h) = E\left[\int \left(f_h - f\right)^2 dx\right] \qquad \ldots(ii)$$

Gaussian Basis Function

If Gaussian Basis Function is used to approximate and the underlying density is a Gaussian, then the optimal choice for h (i.e., the bandwidth that minimizes/reduces the MISE) is given by the general formula:

$$h = \left(\frac{4\hat{\sigma}^5}{3n} \right)^{\frac{1}{5}}$$

Gaussian Approximation or Silverman's rule

h of Gaussian Basic function is not a good fit for long tails and skewed distribution as it is optimized for Gaussian Distribution. From the general formula, some changes were made to make h more robust.

$$h = 0.9 * \min\left(\hat{\sigma}, \frac{IQR}{1.34}\right) * n^{-\frac{1}{5}}$$

Course References

[1] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*. Pearson, 2021.

[2] T. Ghosh and S. K. B. Math, *Practical Mathematics for AI and Deep Learning: A Concise yet In-Depth Guide on Fundamentals of Computer Vision, NLP, Complex Deep Neural Networks and Machine Learning (English Edition)*. BPB Publications, 2022.

[3] M. P. Deisenroth, A. A. Faisal, and C. S. Ong, *Mathematics for Machine Learning*. Cambridge University Press, 2020.

[4] T. V. Geetha and S. Sendhilkumar, *Machine Learning: Concepts, Techniques and Applications*. CRC Press LLC, 2023.

[5] A. Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly Media, 2023.

[6] O. Theobald, *Machine Learning for Absolute Beginners: A Plain English Introduction (Third Edition)*. Scatterplot Press, 2021.

Accessing Course Resource

linkedin.com/in/Samanipour

t.me/SamaniGroup

github.com/Samanipour