Module 8
Video Transcripts

Video 1: Probability and Statistics (6:46)

Welcome. In this video we will introduce you to some foundational concepts about probability and statistics. Let's begin by talking about what probability is. Probability is a branch of theoretical mathematics that focuses on predicting the outcome of events that have yet to happen. Probability is always measured between 0 and 1, or between 0% and 100%. An event with a probability of 0 will not happen. On the other hand, an event with a probability of 1 will happen for sure.

The probability of an event occurring, it's always going to be between 0 and 1, as displayed on this probability line. Keep in mind that because probability deals with events that have yet to happen, it's going to tell us just a prediction of what could happen. Let's now have a look at how to compute probability. A probability of a simple event happening can be computed by dividing the number of possible events which represents the number of times a certain event can happen by the number of possible outcomes, which is the total number of results that we can obtain in our experiment. Let's try to understand this formula with a couple of examples.

In our first example, we're rolling a fair die and we want to compute the probability of getting 7. In this case, number of possible events is 0, because we cannot obtain 7 where rolling a die. On the other hand, the number of possible outcomes is 6, because we could get any number between 1 and 6. Therefore, the probability of getting 7 is going to be 0 divided by 6, which is 0. For our second example, we're tossing a coin and we want to compute the probability of getting heads or tails. In this case, the number of possible events is 1. And the number of possible outcomes is 2 because we could get either heads or tails. Therefore, the probability of getting heads or tails is exactly 1 divided by 2, which is 0.5.

Let's now move on to statistics. Statistics is a branch of applied mathematics that focuses on studying the outcome of events that have already happened, by analyzing the frequency of past events. In statistics, we're concerned with the collection, organization, analysis, and presentation of data. And in statistics, we apply probability theory to draw conclusions from events that have already happened. Let's try to understand how statistics work with an example. Suppose that in a container there are blue and red marbles. And we want to determine whether the chances of drawing a blue or a red marble are the same. A statistician is asked to draw 100 marbles in total.

At the end of his experiment, he counts the marbles that he has extracted and counts that there are 49 blue marbles and 51 red ones. Therefore, at the end of his experiment he may conclude that the chances of extracting a blue or a red marble are about the same, although we did not get an exactly 50%.

Let's now try to understand the difference between probability and statistics with an example. In particular, we want to understand how likely are we to get 6 when rolling one die. If I were a probabilist, I would assume that the die is fair, and using the probability formulas I would determine that each face has a probability of exactly 1 over 6 of coming up. If I were a statistician, I could not make any assumptions about the die, but I would observe the results for a while and keep track of how often each number comes up. In the end, I would determine if the observations are consistent with the assumption of equal probability faces. In other words, probability theory draws conclusion a priori, or before, based on mathematical formulas, while statistics draws conclusions a priori, or after, based on real-life observations using the same rules as probability.

Let's try to understand the difference with another example. Here, we want to decide whether a coin is fair or not. First, assume that we tossed a coin 10 times. From a probability point of view, we know that there is exactly a 50% chance of getting 5 tails out of 10. From a statistic point of view, we may toss the coin 10 times and observe 4 tails. We may toss the coin again 10 times and observe 5 tails. And we may toss the coin a third time 10 times and observe 6 tails. Therefore, a statistician may conclude that the chances of getting 5 tails out of 10 is between 40% and 60%.

Assume now that we're asked to toss the coin 100 times. Again, a probabilist will know that there is exactly a 50% chance of getting 50 tails out of 100. A statistician may toss the coin 100 times and get 45 tails. He may toss the coin again 100 times and get 50 tails. And then, again, 100 times to get 55 tails. Therefore, we may conclude that there is a 45 to 55% chance of getting 50 tails out of 100. In short, as we increase the number of coin tosses, the statistician may observe that the chances of getting an equal number of heads or tails gets closer and closer and may conclude that the coin is fair.

In summary, probability focuses on predicting the outcome of future events and it's always measured on a scale from 0 to 1. Probability also lays the foundation for statistics. Statistics focuses on the outcomes of past events to draw a conclusion and uses the same rules as probability but applies them to past outcomes.

****

Video 2: Probability vs. Statistics in Python (9:54)

Welcome. In this video we will consolidate your knowledge about the difference between probability and statistic. In particular, after reiterating the difference between the two disciplines, we'll show you how to use different problems using statistic and probability using Python. Let's begin. As you know probability deals with predicting the likelihood of future event. Probability its primarily a theoretical branch of mathematics, which studies the consequences of mathematical definition. On the other hand, statistic deals with analyzing the frequency of past event.

Statistic is primarily an applied branch of mathematics, which tries to make sense of observations in the real world. Let's try to consolidate this knowledge with some example. In our first example, we consider the tossing of a coin. In a coin toss, the only events that can happen are flipping heads or flipping tail. These two events are the set of all possible events and represent the sample space.

As you know to calculate the probability of an event occurring, we sum how many times an even of interest can occur and we divide it by the size of the sample space. It is pretty easy to understand that, from a probabilistic or theoretical point of view, a fair coin will have exactly a 50% chance of being heads or tails. However, as we will see in the example below, from a statistical point of view, the observed probabilities of tossing a coin will not be exactly 50%. Observe the code in this code cell.

Here we'll begin by importing the library random, which will help us simulate the tossing of the coins. In this experiment we want to simulate the toss of n coins, each one flipped a 100 times. For this reason we have to find two different functions. Coin trial, which simulates tossing a coin a 100 times and simulate, which simulate the toss of each one of the coins. Let's have a closer look at each one of these functions. After defining the header for the coin trial function, we initialize a counter heads that will keep track of how many times we obtain heads when flipping a coin 100 times.

Next, we use a for loop to simulate flipping a coin 100 times. If we get a number that is lesser or equal than 0.5, it means that we have obtained a head and therefore we increment the value of the counter. At the end, the function returns the total number of heads. The function simulate, simulates the tossing of n coins. This function takes one argument n, which represents the total number of coins that we have. After defining the function header, we set a random seed for reproducibility of the results. Next, we create an empty list trials to simulate the outcomes of the trials. And we use a for loop to simulate the toss of n coins, by calling the function coin trial and by appending the result inside our list trials.

Finally, we compute the probability by summing the result of all the trials and dividing it by the number of coins we have. In the second code cell we simulate the result of getting heads when tossing a 10, a 100, a 1,000, or a 10,000 coins, each one a 100 times. You see that on average we always get a probability that it's close, but not equal to 50%. Let's now have a look at how to use probability in Python. Recall that probability is a number that reflects the chances or likelihood that a particular event will occur. Probabilities can be expressed as proportions that range from 0 to 1, or as percentages ranging from 0% to 100%. A probability of 0 indicates that there is no chance that a particular event will occur. Whereas a probability of 1 indicates that an event is certain to occur.

Let's consider an example. Here we want to consider the probability of drawing a certain cards from a standard deck with 52 cards. In particular, in this first code cell, we want to calculate the probability of drawing an ace. First of all, we need to define the size of the sample space. Because we have 52 cards in the deck, the size of the sample space cards will be equal to 52. Next, because we want to calculate the chances of getting an ace, we need to set the size of the possible outcomes equals to 4, because we only have 4 aces in a deck. Therefore, we set the variable aces equal to 4.

Next, we use the standard probability formula to compute the probability of drawing an ace by dividing aces by cards. And we use a print statement to print the probability rounded to 2 decimal places. When I run this code cell, you see that the probability of drawing an ace from a standard deck is 0.08. We can also decide to convert this probability to a percentage simply by multiplying it by a 100. In this code cell, we have multiplied ace probability by 100 and use the print statement to print the probability in a percentage form.

When I run this code cell, you see that the probability of drawing an ace from a standard deck of cards is indeed 8%. We can also compute the probability of drawing a heart or a face card from a 52 card deck. In this case, it might be convenient to create a user defined function to calculate the probability of particular events. In this code cell, we have defined a function event probability that takes 2 arguments. Event outcomes, which represents the possible outcomes of an experiment. A sample space, which represent the size of the sample space of our experiment. This function computes the probability in a percentage form of a certain event and returns the result rounded to 1 decimal place. Let's try to compute the probability of drawing a heart.

Again, the sample space will have size 52 because we have 52 cards. Next, we set the variable hearts equals to 13. Which will also represent the outcomes of our experiment. Finally, I compute the probability of drawing a heart, heart underscore probability by calling the function event probability with the argument heart and cards. I can also try to determine the probability of drawing a face card. In this case, the number of possible outcomes will be 12. Because I, in a standard deck of cards I have 4 kings, 4 queens and 4 jacks.

Therefore, I can compute the probability of drawing a face card, face card probability, by calling the function event probability with the argument face cards and cards. When I run this code cell, I obtain the probabilities for my experiments. In particular, the probability of drawing a heart, it's equal to 25%. And the probability of drawing a face card, it's equal to 23.1%. One important thing to notice, is that in this example about probability we do not set a random seed. This is because from a probabilistic point of view, the results are purely theoretical and for the reason they won't change. Of course, if we were to simulate drawing a certain card from a statistical point of view, the results would be similar to the ones we obtained in the example above.

In summary, in this video, we have seen that probability and statistics are loosely connected. However, the first one deals with events in the future and the second one with events in the past. We also saw how to solve the coin toss problem from a statistic point of view, and we solved, and we computed the probability of drawing a certain card using probability.

****

Video 3: Sampling Using Python (9:55)

Welcome. In this video we're going to talk about sampling. In particular, we will focus on a particular sampling technique called simple random sampling. Finally, we will show you how to perform sampling on a data frame. Let's begin. Sampling is a technique that is used when we can collect data from an entire population to collect a representative sample.

There are two main categories of sampling. It can be probabilistic or non-probabilistic. Probability sampling is defined as a sampling technique in which the researcher chooses sample at random. Non-probability sampling is defined as a sampling technique in which the researcher selects the sample based on a selective judgment or some criteria.

Let's try to understand better the difference between probability and non-probability sampling. In probability sampling, the samples are randomly selected, which means that everybody has the same chance of getting selected. Of course, with this technique, bias is reduced and it helps us in creating accurate samples. The problem with this technique is that sometimes the correct audience might be difficult to find. With non-probability sampling, the samples are selected based on subjective judgments or the researcher's criteria. For this reason, not everyone has an equal chance to participate.

Of course, with this technique, bias is not a concern for the researcher and this method does not help in representing the population accurately. However, because of the criteria chosen, finding an audience might be very simple using this technique. As an example of non-probability sampling, we can consider selecting some -- a pop -- a subset of a population. Perhaps we could decide to only select a subject of the population only if it -- only if they had an engineering degree, for example. In this way, however, all the other people will never be selected with this sampling.

Let's have a look at a very popular sampling technique, simple random sampling. Simple random sampling is the simplest probability technique and it is an entirely random method of selecting a sample. This method is as easy as assigning numbers to the population and then choosing them at random. Let's try to understand how simple random sampling works with some examples. First, suppose you want to send a survey to the entire population of the United States. This would be nearly impossible to accomplish, because the United States has a population of 330 million. Therefore, you might decide to identify a sample of 1 million by using probability sampling to collect your data. In this way, you can make sure that 1 million people represent the entire population because they may come from different states, they might have a different ethnicity, they might be of different ages, and they might come from different backgrounds. In other words, sending a survey to 1 million people identified as our sample will help us question a set of citizens that are representative of the broader population and will limit the potential for bias.

Let's see now how simple random sampling can be performed using Python. In Python, simple random sampling can be done by using the function sample from the random library. This function takes two arguments, population, which is a sequence such as a list, a set, a range, a dictionary, and so on, and k, which is the size of the return sample that we want. Let's see how to use the function with an example. Suppose you're a data scientist at a travel company and that we have access to many databases that collect data from different airlines. Assume further that all these databases are very big, but they all contain the same information about the different airlines. It might be a smart idea to use simple random sampling to only select a portion of your original database, perhaps by choosing only four databases, and use them for your analysis.

Let's see how to do this in Python. In this code cell we have imported the random library first. Next, we have defined the list airline data that contains the names of the airlines we have a database of. For reproducibility, as usual, we set a random seed. And finally, we use the sample function from the random library with the population argument equals to airlines data and k equals to 4, because we only want to choose four random databases from the original ones. When we run this code cell, you see that Python suggests us to use Air France, Poland, KLM, and Emirates for our analysis. Notice that the sample method returns a list with a random selection of a specified number from a sequence. Of course, the return sample is meant to be unbiased and it's meant to represent the total population.

Next, let's have a look at how to perform sampling on a data frame. Why is this important? Well, because sampling is a technique that is widely used in data science and machine learning. To perform sampling on a data frame, we can simply use the function sample from the pandas library to generate a random sample from our data frame.

Let's see how this works with an example. Here, we consider a dataset that contains data of flights between Australia cities. To read this dataset, we import the library pandas as pd. We use the read underscore csv function to read the dataset that we have downloaded. And we use the function head to visualize the first five rows of our data frame. You see that this data frame contains the column city, which represents the city where the flight originated from. The column city two represents the city where the flight has landed. Next, we have the passenger that were carried on the flight, the distance covered by that flight, the number seats, the year of the aircraft, and so on.

As first example, suppose that we wanted to extract only 10 random rows from our data frame. Doing so is very easy with pandas. We can simply apply the function sample to our original data frame by setting the argument n equals 10, which is going to tell Python that we only want to extract a random sample of our data frame that only has 10 rows. We name this new data frame flight 10 and we display it. As second example, we can also decide to generate a sample containing only a portion of the original -- of data that was in the original data frame. Here, for example, we want to extract a sample that is a quarter of the original data frame.

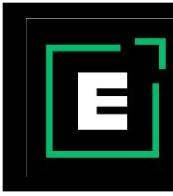
Again, we can do so by using the function sample. We apply the function sample to our original data frame, but this time we set the argument frac equal to 0.25 or 1/4. We redefine this new data frame to be flight quarter and we display it. We can also perform a sanity check to make sure that these last data frame that we have obtained is exactly a quarter of the size of the original one. We can do so by using an if statement where we check whether 1/4 times the number of the rows of the original data frame is equal to the number of the rows of flight quarter. If this is true, our print statement will print the string great. Perfect.

In summary, in this video, we have discussed that sampling is a technique used to select a portion of the initial data. We have also focused on a specific sampling technique called simple random sample, which is the most simple technique and chooses an unbiased set of the original data. And finally, we have seen how to perform sampling on a data frame using pandas.

****

Video 4: Random Variables and Probability Distribution Functions (10:37)

Welcome. In this video we will talk about random variables and probability distribution functions. Let's begin by talking about what is a random variable? A random variable is the set of possible values for a random experiment. A random variable is usually denoted by capital X. In the equation that you see here, x is the random variable, 2 is a possible value that this random variable can take, and the 1/2 is the associated probability with that value. Of course, because a random variable is the set of the possible value for an experiment, it's also the same as the sample space.



Let's try to understand what a random variable is better with a couple of examples. In our first example, we're considering throwing a six-faced die. In this case, we have 6 possible different outcomes, 1, 2, 3, 4, 5, 6. Therefore, we can define a random variable x associated with this experiment that can take the 6 different outcomes.

Notice that the size of the random variable is equal to the size of the sample space for this experiment, which is 6. As a second example, consider tossing a coin. When tossing a coin, we have two different possible outcomes, head or tails. If we map heads to 0 and tails to 1, then we define a random variable x associated with this experiment, but can only take the values 0 and 1. Notice that 2 that is the size of the random variable is also equal to the size of the sample space for this experiment. We can also use random variables to compute a probability. We can do so by using the following mathematical notation. We can include the name of our random variable and its corresponding value. And then assigning to it the probability of that value.

Consider this example. Here we're rolling a die. We know that when rolling a die we actually have 6 possible outcomes, and that if the die is fair, every face will come up with a probability of exactly 1 over 6. Therefore, we can assign 1 over 6 as the probability to each one of the values that we can expect. Notice that the sum of the probability equals 1, as it should be. So far, we have seen random variables that can only take a finite set of values, such as in the examples when we were rolling a die or we were tossing a coin. These type of random variables are called discrete random variables. There are cases; however, where random variables can take an infinite number of possible value. These type of random variables are called continuous random variables.

Let's try to understand what a continuous random variable is with some example. A continuous random variable is a variable that can take any possible value, such as depth, height, weight, and so on. Of course, we need also to assume that any of those measurements are not rounded, because we need to allow any possible value for our random variable. As an example, suppose that we want to find the range of depth measurements of a one-mile radius around the center of Lake Michigan. In this case, we can define a continuous random variable x that will take any possible value that we've measured. The range for this random variable x will go from the minimum depth measured to the maximum depth measured.

Now that we are familiar with random variables, let's look at probability distribution functions. A probability distribution is a list of all the possible outcomes of a random variable along with their corresponding probability values. As with random variables, we can have discrete and continuous probability distribution functions. These two images display an example of a discrete probability distribution function and of a continuous probability distribution function.

As an example of a discrete probability distribution function, consider the experiment where we're flipping a coin two times. It is easy to understand that for this experiment we have four different possible outcomes, two tails, tails and heads, heads and tail, or two heads. We can also define a random variable x that represents the number of heads that we obtain when flipping the coin. It's easy to understand that this random variable can take the values zero in case we get two tails, one in case we get heads and tail or tail and heads, or two in case we get two heads.

The probability associated with these values are the following. The value zero will have a probability of 0.25, because the outcome tail tail, it's only one out of the four possible that we can get with this experiment. The value one will have a probability of 0.5, because we can get one head with two outcomes out of the four possible ones. The value two will have a probability of 0.25, because we can get two heads as only one of the four possible outcomes. The table that you see here represents the probability distribution function for this experiment. The probability distribution function for this experiment can also be represented by a histogram.

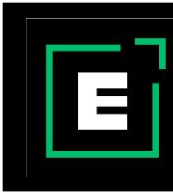
Notice that the value zero has a height of 0.25. The value one has a value of .5. And the value two has a value of 0.25 again. Notice further that the sum of the area of the single bins in this histogram equals one as expected. This distribution follows a trend which is also characteristic of a very popular and well-defined discrete distribution function, the binomial distribution. Let's now talk about continuous probability distribution functions.

Continuous probability distribution functions are closely related to continuous random variables. Because they can take infinitely many values, the probability that a continuous random variable will assume a particular value is equal to zero. For this reason, when we are considering continuous probability distribution function, we compute a probability in terms of intervals. Of course, because we're dealing with infinitely many values, we cannot express a continuous probability distribution function in tabular form, but instead we use an equation or a formula.

Let's try to understand this better with an example. Suppose you are collecting the weight of European women. After collecting your data and plotting it, you obtain this bell-shaped blue curve. The first thing to notice about this curve is that it follows a trend, which is also characteristic of a very popular probability distribution function, the normal, or Gaussian, distribution. The area under the blue curve, it's equal to one because it represents all the possible outcomes of your collection of your date. Because continuous probability distribution function express probability in terms of interval, if I wanted to know the probability of a weight being greater than, for example, 152 pounds, such as -- which is represented by the area in red on this graph, I would need to compute the area only after 152, which in this case is roughly equal to 0.38.

In summary, in this video we have learned that a random variable is the set of possible values for a random experiment, and that we can use random variables to express the probability of a certain event happening. Furthermore, random variables can be discrete or continuous. Next, we have learned that random variables are closely connected to probability distributions, and that probability distribution map the possible outcomes of a random variable to their corresponding probability value. As for random variables, we can have discrete and continuous probability distribution functions. Discrete probability distribution functions can be represented as a table, whereas continuous probability distribution functions can be expressed as a formula or with an equation.

****



Video 5: Random Variables and Probability Distribution Functions in Python (9:19)

Welcome. In this video we will talk about random variables and probability distribution functions. After a quick recap, we will show you how to implement discrete and continuous random variables and probability distribution functions in Python. As you may recall, a random is the set of possible values from a random experiment. Of course, depending on the nature of the experiment, random variables can assume discrete or continuous values. For example, you know that we can define a random variable x to be the number which comes up when you roll a fair die.

In this case, x can take values 1, 2, 3,, 4, 5, or 6 and; therefore, it's considered a discrete random variable. We can also define continuous random variables. For example, x can be the height of students in a class if you decide not to round your measurements. Since the height can assume any values, x is considered a continuous random variable. We have also learned that probability distribution function is a list of all the possible outcomes of random variables along with their corresponding probability. Again, depending on the experiment and on the random variable associated with it, we can have a discrete or a continuous probability distribution function.

Let's now have a look at how to implement discrete random variables and probability distribution functions in Python. Let's consider the example of tossing an unfair coin. In this experiment, we have two possible outcomes, heads, which we can map to zero, and tails, which we can map to one. Then we know that the corresponding random variable for this experiment, x, can take values only zero and one. Next, because the coin is not fair, we can define the probability of getting heads as 2/3 and the probability of getting tails as 1/3. To implement this experiment in Python we can use the rv discrete function from the library scipy stats. This function takes one required argument. Values, which is a two-array, like xk and pk, where xk are the integers that your random variable can take, pk are the associated non-zero probabilities. And, of course, xk and pk must have the same shape.

Let's have a look at the code. We start by importing the necessary libraries as usually. So, we import the module stats from scipy and matplotlib. Next, we define our discrete random variable rv that can take only values zero and one. We defined the associated probability as 2/3 and 1/3. And next, we generate our probability distribution functions using the function rv discrete. Notice here that we have passed to this function our random variable and our probability vector. Next, we plot the probability distribution function using Matplotlib. Of course, the probability distribution function that we have obtained is discrete. There is something worth noticing about this distribution. You can see that it only takes values zero and one, and that their probability sum to one. Any probability distribution function with these characteristics is called Bernoulli distribution. In fact, the Bernoulli distribution is a discrete distribution having only two possible outcomes, zero and one, in which one occurs with probability p and zero occurs with probability q equals 1 minus p where p is, of course, between 0 and 1.

For our second experiment, let's consider tossing a fair coin. Again, we have only two possible outcomes, heads which we can map to zero and tails which we can map to one. Our random variable x for this experiment can only take values zero and one. And now, because the coin is fair, both heads and tails will have a probability of 1/2 of turning up. The code to implement this experiment is very similar to the one we've just seen. The only thing that we've changed is the vector of probabilities that now takes values 0.5 and 0.5.

Again, we generate the probability distribution function using rv discrete and we use matplotlib for plot. Again, we have obtained a discrete probability distribution function. The shape of the distribution that we have obtained follows, again, the one of a Bernoulli distribution, because it only takes values zero and one, but this time the probabilities are both equal to 0.5.

Let's now have a look at continuous probability distribution functions in Python. Let's have a look at this following example. In this code cell, we have imported a dataset that includes the weights and heights of some individuals. As usual, we begin by importing pandas as pd, and we use the read underscore csv function from pandas to read our dataset into the data frame df. Next, we use the function head to visualize the first five rows. You see that this data frame contains three columns; one describing the gender of the individuals, the second describes the height, and the third one the weight. For simplicity, we're only interested in the weights of female individuals.

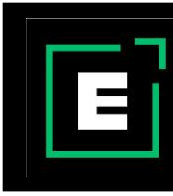
Therefore, we decided to remove all the rows where gender is equal to male. When we run this code cell and we visualize the first five rows again, you see that now the gender is only female. Next, we're interested in plotting the weight in our data frame in a histogram. The shape of the distribution above is really similar to the shape of a very common continuous distribution called the normal or Gaussian distribution. In fact, we will show you that we can fit a normal distribution on top of the one that we've just obtained.

In order to do so, we need to extract from the data in our data frame the mean and the certain deviation, as we will need these parameters in order to generate our fitting distribution. To do so, we use the function mean and standard deviation on the column weight to compute the mean and the standard deviation of a distribution. Now, we can use these parameters that we just computed to generate the normal distribution using the function norm from scipy stats. In this code cell, I'm importing the function norm to generate the distribution, and then I gen -- I generate the pdf by passing to the function norm the mean and the standard deviation that I just computed.

Next, we need to choose the range on which we'll plot our probability distribution function. Of course, it makes sense to choose as lower bound the minimum weight in our data frame and, as upper bound, the maximum weight in our data frame. Next, we generate a sequence of equally spaced values between the minimum weight and the maximum weight, and we generate the probabilities associated with each values.

Finally, we plot the sample distribution and the normal distribution that we have just generated on top of each other. You see from this graph that the orange line, which describes the normal distribution that I just obtained, fits our data very well. In summary, in this video, we have shown you what a random variable and a probability distribution function are, and how to implement discrete and continuous probability distribution function in Python.

****



Video 6: Probability Mass and Probability Density Functions (11:55)

Welcome. In this video, after a quick recap about probability distribution functions, we will explain the connection between discrete probability distribution function and probability mass function and between continuous probability distribution function and probability density function. As you may recall, a probability distribution is a way to represent the possible values and the respective probabilities of a random variable. We know by now that there are two types of probability distribution, discrete and continuous probability distributions.

Discreet distributions are used when we have a discrete random variable, whereas a continuous distribution is used when we have continuous random variables. These two classes are represented by a probability mass function in case we have a discrete probability distribution function or a probability density function in case we're dealing with a continuous probability distribution function.

Let's have a look at the connection between discrete probability distribution function and probability mass function with a couple of examples. For our first example, we're going to roll a die a very high number of times and look at the proportion of each of the faces of the die. As we know, we can use a random variable X to describe the possible outcomes of the die. We know then that the random variable X can take values 1, 2, 3, 4, 5 and 6. The aim of the probability mass function is to describe the probability of each possible value.

For example, in our case, it describes the probability to get a 1, a 2, a 3 and so on. We also know that if the die is fair, the probability of getting a 1, it's going to be equal to the probability of getting 2, 3, 4, 5 and 6. And because there are six possible outcomes, then each one of these probability will be equal to 1 over 6. We can verify if this is true in Python as follows. In this code cell, as usual, we import the necessary libraries at the beginning. We are going to be using numpy to simulate rolling the die, and matplotlib for plotting.

Next we define a variable num throws equals to 100,000, which is the number of times we decide to roll the die. We also initialize our outcomes to zero at the beginning. To do so, we use the numpy function zeros to define a zero array, a zero vector with length 100,000. Then we'll simulate our throws. To do this, we use a for loop and we use the function random choice to simulate the outcome of each roll. We append outcome to our array. At the end of our simulation, we extract the outcomes and the number of time each one turned up. We do so using the function unique from Python. This function extracts the possible outcomes from the function out of -- sorry -- from the variable outcomes and then we compute the probability of each single outcome.

Finally we plot our results. We have obtained this distribution. Of course, since the die is fair, we notice that each outcome has the same probability of about 1 over 6 of turning up. The shape of the distribution that we have obtained follows the shape of a famous distribution called the uniform distribution. In our second example, we want to calculate the probability that a patient will receive a prescription over a year. In particular, we want to visualize the probability distribution of 200 patients who visit a doctor office each year. We know that each patient goes to see the doctor three times a year and each of them can receive up to three prescriptions in total.

Therefore, we can define a random variable X that takes the number of prescriptions each patient received. Therefore, X can take values zero, 1, 2, or 3. In this table, we have displayed the possible scenarios for a given patient who could receive prescription during their appointments. In one scenario, a patient can receive no prescription over the year. For three scenarios, patients can receive one prescription a year. There are three scenarios where a patient can receive two prescriptions a year and one scenario where they can receive three prescriptions a year. Therefore, the associated probabilities with this experiment would be one-eighth in case the patient has received zero prescription, three-eighths in case the patients have received one or two prescriptions over the year. And again, one-eighths in the case the patient has received three prescriptions over a year. We can simulate this experiment in Python.

Again, we import the necessary libraries as usual. We'll use numpy, scipy stats and matplotlib. For reproducibility of the results, we have also set a random seat at the beginning of our code. Next we set the number of patient that we have, so 200 in this case, and as we did for the previous experiment, we initialize the random variable that counts the number of prescription for each one of the patients.

Next we simulate the number of prescription written for all patient using a for loop and the numpy function rounded. Once we have simulated what happens to each patient, we use, again, the function unique to extract the values and to count how many times each outcome occurs. We compute the probability associated with each outcome and we use matplotlib to plot the results.

As expected, we obtain a histogram displaying the probabilities associated with this experiment. You see that the outcome -- the probabilities associated with the outcome zero and three add about a value that is equal to one-eighth, and the probabilities associated with one and two have a probability associated equal to about a 3 over 8. We are already a little familiar with this distribution. Its trending fact follows the shape of the binomial distribution. In this case, the probability mass function associated with this experiment takes values one-eighth, three-eighths, three-eighths and one-eighth.

Next, let's talk about continuous probability distribution functions and their connection to probability density function. As you know, a continuous distribution describes the probabilities of possible values of a continuous random variable. A probability density describes the relationship between the continuous random variable observations and their probability. The overall shape of the probability density is referred to as a probability distribution, and the calculation of probability for specific outcomes of a random variable is performed by a probability density function.

Let's try to understand this better with an example. Suppose we are measuring the height of the Rocky Mountains. We find that the shortest mountain is 17,034 meters tall, and that the tallest mountain is 43,056 meters tall. Because the Rocky Mountains is extensive, we want to find the probability that any random mountain is shorter than 3000 meters. Interestingly enough, during our measurement, we also find that the distribution of the mountain heights follows a normal distribution with mean equals to 3062 and standard deviation 500. We can plot this distribution as follows.

As usual, we import the necessary libraries and we define range between the shortest and the tallest mountain that we have, and we use the function norm dot PDF to display our distribution on top of it. Of course, what we have obtained, it's a distribution that follows a normal distribution. If we want to compute the probability that the mountain is shorter than 3000 meters, we can use the function CDF on our random variable. In this code cell, we have declared normal random variable H with mean 3062 and standard deviation 500. Next we compute the probability density that any mountain is shorter than 3000 meters by using the function CDF.

Next we plot our probability distribution function. Observe that the result that we have obtained is expressed as a percentage because we have multiplied our probability by 100. Observe also that for our convenience, we will plot a vertical line up to 3000 on top of our distribution. It appears that about 45.06% of the Rocky Mountains are shorter than 3000 meters. This probability is also represented by the area to the left of the vertical line in the graph above. Of course, if we wanted to compute the probability that a mountain is taller than 3000 meter, we would need to subtract our result from 1 like we did in this code cell. We see now that the probability that a mountain is taller than 3000 meter is equal to 54.93%. This probability is also represented by the area on the right of the vertical line.

In summary, in this video, after a quick recap of our probability distribution functions, we have shown you the connection between discrete probability distribution and probability mass function and between continuous probability distribution and probability density function.

****

----------**END**----------