Module 9
Video Transcripts

Video 1: Uniform Distribution (6:40)

Hello, and welcome to module nine. In this module, we'll cover different statistical concepts and tools that will help you understand better your data. In the last module, you learned about topics like distributions, random variables, probabilities, and all of this, and now we're going to learn how we can apply all of that to data science. The thing we're going to do is in this first network we're going to cover different statistical distributions. We're going to understand the most important ones that are the uniform, the Bernoulli, binomial, normal, exponential, Poisson, and the t distribution. So, let's start by importing all our libraries for plotting and creating data that are numpy and seaborn. So, the first distribution we're going to understand is the uniform distribution. This distribution is also known as the rectangular distribution and is the one that has a constant probability. The probability distribution function of continuous uniform distribution is this one here. So, it's a piecewise function, as you can see, that it's 0 if x is less than a, our value a, 1 over b minus a if x is between a and b, or 0 if x is bigger than b. We will be using the scipy for the rest of this notebook, and the idea here is that you can create these functions by yourself. I mean all of them have, they all have an equation, so you can easily with Python program that equation, but it's much better and easier to do it with good libraries like scipy.

One other advantage of using scipy like numpy is that it has some optimization underneath. So even though you can create code that is similar to this, it's very unlikely that you will reach the performance of these libraries. So, to begin, let's import the uniform distribution from scipy. We're actually using a scipy module called stats. So, in able to do all of this, the first thing we need to do is to create random numbers that will come from a uniform distribution. The idea of this in this module we're going to learn how to create the distributions knowing that we know, I mean, with the knowledge that we are aware that we have all the distributions. The idea is that in real life no one is going to tell you this data comes from this unless you have a very specific process. But what you will do is that you will understand your data, and you will plot it and perform operations, and you have to discover if this data is coming from a normal distribution, binomial, and all of that. So, with that in mind, let's start by creating 10,000 random numbers from the uniform distribution, and what this function takes is the start and it takes the scale. To do that, we're going to use the function rvs. I'm going to create this, and then I'm going to show you what's rvs. Rvs means random variates of given type. This is the way we can create an array of different values for a given distribution. So, it's just creating random numbers that comes from a distribution. These arguments here are not the same for each one of the distributions we're going to see in this module. As you will see, this right here needs a size, a loc, and a scale, but not all of them are the same. So, now we created our data uniform. Let's first try to see it. As I told you, there are numbers between 1 and 20, and there are 10,000 of them. They can be, I mean in the uniform distribution, you don't need to have only integers. You can also have decimal values. The important thing here is that you need to follow a uniform distribution. The way we're going to check this is a distribution plot. In seaborn it's very simple to plot a distribution. What we'll have to do is to use the function distplot. Then, we're going to pass this distribution the data uniform that contains the data from the uniform distribution. We're going to say bins 10, 20, sorry, and kde false.

This distribution plot, if you say kde is equal to true, it will also print kernel function here that will follow along the path of the distribution, but it's not in the bars what is actually a line, a smooth line. If you want to see it, you can just say kde equals to true, and here you'll see it. You can take the same amount of information from this line than from the bars. That's very important. And it's interesting here, the distribution function should look like a rectangle, something like this, but it's not quite a rectangle.

The problem here is that we're not using enough data to be able to account for this type of distribution. So, 10,000 points may seem like a lot, but you need more, much more points, to be able to get into the actual shape. A perfect distribution will have an infinite amount of numbers, that's not possible, but we can grow, and we will see that it's going to get much more closer. Just for the sake of trying, let me go here and just test for 100,000 points. Now, I'm going to plot it here, and as you can see, it's much more close to the actual data, to the actual form or shape we expected from the uniform distribution. If I go to a million points, 10 million points, 100 million points, a bill points, it's going to get closer and closer and closer to the shape of the rectangle.

****

Video 2: Bernoulli and Binomial Distributions (6:25)

Let's go now to the Bernoulli distribution. By the way, the people always hear the name Bernoulli. It's not always the same guy, that's an interesting fact. You can check, they were a family of mathematicians, physicians, and doctors, and you can see the history of the Bernoulli family. They're very interesting people that added a lot of value to the world of science. So, coming back to the world of statistics. The Bernoulli distribution has only two possible outcomes, the success and the failure. Normally, we say the success is 1 or the 0 is the failure. Like, if we say we toss a coin, we can say that heads is success, so 1, and that tails is 0, so a failure. The function we have for the Bernoulli distribution is this one right here. So, it depends on a value called k, and also, and this k is 0, 1, here, the success and failure, and this p is the probability of success. So, it depends only on two parameters. We have k and we have p, where k is 0, 1, and is the probability of success.

This probability of success shouldn't always be 0.5. That means that we're used to understanding like the, when you toss a coin, you will get a 50/50 percent of getting a head or getting a tails, but you can also have a loaded coin, and with a loaded coin, you'll find that sometimes the probability of heads is bigger than tails or backwards. For the sake of the example here, we're going to use a Bernoulli distribution with p equals to 0.5. In here, k is assumed to be 0 or 1, and we're going to create 10,000 cases again. So, again, probability is 0.5. You can see we're also using the stats module in scipy, but now we're importing the Bernoulli distribution. The rvs here, so the random variables we're creating is not the same as before. If you remember, in the uniform distribution, it asks us for the width, for the scale, something we called the loc, and all of that. In here, we're only asked to give the size and the p. So, when you do this, let me now run this, we have an array of 10,000 points that are always 1 or 0. If we plot it, as you will expect, this is what we will see. We will see a graph that will have half of the points in 0 and half of the points in 1. This is because the probability we gave was 0.5. If we say 0.6, as you can imagine, we'll have more data towards the 1. If we say 0.4, we'll have more data towards the 0. So, you can play with this and see different cases for the Bernoulli distribution.

Another distribution that is similar to the Bernoulli, but in here we have different trials, is called the binomial distribution. Binomial distribution is a very interesting one, and as you can see here, the term here on the right is almost the same as the Bernoulli distribution or that we have another factor and another term that is called the binomial here. And this binomial here, it's dependent on n and k. So, k is, again, 0 or 1 in this case, but n here is the number of trials we give. And one important thing is that each trial is independent of each other. But the definition of this symbol here, nk, is this, is a factorial of n divided by k factorial and that's multiplied by n minus k. So, again, you can create a function and the same with the Bernoulli distribution, but we're going to trust again scipy in Python. In this case, the way we use it is that we import binom here, the binomial distribution from scipy stats, and the only difference we have when creating the variate is that we have to give the number of trials. Here, we have 10 different trials of let's say toss in a loaded coin 10,000 times where the probability of heads is 0.6. You can think of it as like that.

Let me run this, and in here you will have, now you won't have 0 or 1. What you'll have here is something like this. It says here that, so 10 in this case is the number of trials you did. So, if you have a 6 here, this means that when you in the first trial you launched the, let's say we have a coin, you tossed a coin 10 times, and you got six heads. The next one you got 10, then 9, then 3, then 8, and then you have this distribution. If we plot it here, it's going to be very similar to a different one that is called the normal distribution, and this is the way we can plot it. So, if you remember some exercises and videos about like tossing coins and the central limit theorem, some of these things are very related to the concept of the binomial distribution, so this is a way we can start understanding all of that. This distribution is a distribution that is not a continuous function, okay. In this case, we don't have a continuous function. We have a summation of functions. That is not the same as having like a smooth function in this case.

****

Video 3: Normal Distribution (2:55)

The next distribution we're going to talk about is the normal distribution. The normal distribution may be the most important one in all of statistics. If you ask a statistician what you really need to understand to learn and understand statistics, they're going to tell you that you need to fully understand the normal distribution. You will see much more about it in the future videos, but for now you can just think of it as an important distribution that is followed by a lot of processes in the world of data science, which you learn in that. A lot of people call them, call this distribution the normal distribution. Some people, like physicists, they call it the Gaussian distribution. And there are some people that also call it the bell curve, because of the shape, we're going to see it very soon. The function behind the normal distribution is a distribution function density curve with mean mu centered deviation sigma. So, these are the parameters of our function. So, we have our x, our variable, and we have two parameters. We have sigma squared, that we also call the variance. If we take this 2 here, we have the standard deviation, and we have mu, that is also called the mean. And it's an exponential function, as you can see here, but it's a very specific type of exponential function. The way we create it here with Python is that we import it from Python, as usual, but in this case, it will ask us for three things. The size is how many points we want to create. So, in here I'm going to create 10,000 points. Loc is going to be mu, so we're going to create a mu 0, and scale is going to be 1.

And scale, in this case, is the sigma, okay? Not sigma squared, but sigma. In the case, sigma squared, and sigma is the same because there is 1, but it's that's not the case all the time. Something important here, when you have a normal distribution, that mu is 0 and sigma is 1, we call this this the standard normal distribution. So, let me create this and then, as you can see here, we have different points. And so, when we do this and we plot it, this is what we're going to see. You may have seen this before. It is the one they call the bell curve. It's a bell shape where almost all of the data is centered towards the mean. So, most of the data is close to 0 here, and it's getting, and we have less and less data if you go to the extremes of the graphic.

****

Video 4: Exponential, Poisson, and T Distributions (7:23)

For our final video in this notebook, we're going to talk about three more distributions. The first one is the exponential distribution. This describes events in a Poisson process. When you hear this, don't worry, it's not this weird thing. It's just a process in which events occur continuously and independently at a constant average rate. That's the definition of a Poisson point. By the way, it's not poison, it's Poisson, it's French. So, this was also a guy like Bernoulli. The general formula for the density function of the exponential distribution is this one here, and it looks like 1 over beta where beta we call it is scale parameter, and it has mu, that is also the mean. And so, that's basically it. But normally you'll see it online as this equation here. And when we do this, we call lambda the division between 1 over beta. So, normally we use this equation to express an exponential distribution. So, this is very simple. We just have to important it from scipy, and it's going to ask us for the scale, and the loc is going to be here, the mu is going to be the mean, and the scale here is going to be this parameter, lambda here and size. So, that's going to do this, this is what it's going to look like. And as you might imagine, it looks like an exponential function. But it's also, I mean you can also think of this as it's not that different from a normal distribution, meaning that you have this kind of a bell stuff here, but it's centered towards a different place. And it's not true in this case that all of the cases are very close to the mean, because the mean is 0 here, but we have a lot of cases also in the right side. So, this is the way we can represent the exponential distribution.

Let's go now to a Poisson distribution, and as you may imagine, this is very similar to the exponential distribution, but in here, we are modelling the number of events occurring in a given time interval. The math function is very similar to the last one, let me remind you here, we have lambda exponential minus lambda x, but in here, we have something like may remind you to the binomial because we have factorial and we have some divisions. So, this is the way we can put it here. By the way, with a little bit of math fun, you can prove that the normal distribution is a limiting case of the Poisson distribution with the parameter lambda going to infinity. I'm going to leave that too as an exercise. So, to use the distribution, we have to import it from scipy stats. It's going to ask us only for the mu, and it's going to ask us only for the size. This is the data Poisson here, and the plot is something like this. So, what's the difference here? The upper one is a continuous function. This is a discrete function. So, this is why we don't have like these lines here, stuff like that. This is also, this is actually not that kind of function. So, this is the way we define the Poisson distribution. The final distribution that is also a very important one is called the t distribution or the student's t distribution.

And this is a widely used distribution in hypothesis testing that we're going to see in the next video and plays a central role in the very popular t-test. A t distribution describes samples drawn from a full population that follows a normal distribution. So, this is very close also the normal distribution. The larger the sample of the t distribution the more t distribution resembles a normal distribution. And with this distribution, we have a parameter that we didn't use in the other ones, and it's called the degrees of freedom or df. And it can be defined as a number of values in the calculation that are free to vary without violating the result of the calculation. That's the definition of a degree of freedom. You'll see that a lot in the world of statistics. The formula for the probability, for the distribution is something like this. So, it looks kind of weird. I'm not going to lie to you, but if you check about this beta here, this is not a B, this is a beta, a capital beta, this is the beta function. And this mu is a positive integer that is called the shape parameter.

The formula for the beta function is this. So, it's an integral, if you don't know what is that, you can search for that in calculus, integral calculus, and you'll find what is an integral here, but it's the definition of the beta function, and this is called a special function, widely used in physics. So, to import it, we use import t and that's it. It's going to ask us for the degrees of freedom, again the mu, and the sigma. We do it here, and this is our data, but something is weird here. What's happening? This is not what we expected. Before, when we were creating this kind of stuff, with used the rvs, maybe that's the case. Because if you see here, I didn't put here dot rvs. So, let me do it again here with rvs. And no, we only have one point, because we're freezing in just one point. We don't want just one point. We want a whole distribution. So, to do that, we need to do something different for this distribution, and we have to create a linear space. A linear space, you can think of it as like a continuous base of numbers that can define a line maybe, and in here, we're creating one from 0.0001 to 0.9999, and we're creating 10,000 points of this. We do this, so again, the process is we create random variate from t.

We create a linear space, and then we create the pdf. So, you can search for all of these terms in the documentation of the library. It's very simple. When we do that, now we have an array. That's what we expected. And so, if we do a distplot, and remember we were doing distplots all the time before, if you do this, what you get is something like this. This is not actually a right plot for that. I mean this is not a wrong plot, but this is not the distribution plot you will expect from this type of distribution. And the thing here is that we don't want this distribution plot. We want to have a line plot. Because we created data in a different way here. We created a linear space, and so it's not the same as we did before. So, for us to be able to plot it correctly, we need to use a line plot. And as you can see here, this is very similar to just a normal distribution, and we will cover more about the t distribution in the next video. See you very soon.

****

Video 5: Confidence Intervals (8:20)

Welcome to the second notebook. Here, we're going to learn about confidence intervals and hypothesis testing the outline be we'll start with confidence intervals and z-scores. Then we'll see the difference between the z-score and the t-score, and finally, we'll see more about hypothesis testing and the t-test. Also, a little bit about p-values. Let's get started. When you do statistics, something very important that you need to know is that you are giving estimates.

Okay, you're not saying that you have the right or complete exact answer in your hand. You're saying that there's a big chance that the number you're reporting is the one you are reporting. But you need to find a way of estimating how good is your estimation, and that is what a confidence interval will do for you. A definition is, is that this is an interval that contains the unknown parameter, let's say the mean, with a certain degree of confidence. Well, let's say you have a distribution. This looks like a normal distribution that we saw in the last videos, and if you have a 95 percent confidence interval, you're saying that there's a 95 percent that a population mean will fall in this area. Again, this is thinking about the mean. Well, we can have different confidence intervals for different sources, like for the variance and stuff like that. What you're also saying is that all of the values that are not in the lower limit or upper limit are called outliers. We're not going to cover outliers here in this lecture, but as a simple definition, they are values that are outside of our interval of confidence. When we have a standard bell curve that is a normal distribution that we'll call standard bell curve Z, that means that we have mean 0, a standard deviation 1. We can define a z-score as the value zeta r such that the probability of zeta being minus zeta r and zeta r is equal to r. In other words, this is the probability that r is between the point zeta r and zeta mn, minus zeta r and plus zeta r. Well, this is the same here. We need to define a way of putting the mean in this confidence interval, what we're saying is that there are two values that will cut our data from here to here.

We would like say that mu is about x plus and minus some margin of error. Again, we can never be 100 percent sure if the unknown mu will really be within our margin of error. But with larger sample sizes, we have a higher probability that mu will be in our bounds. A way of defining this is defining something called a z-score. In a z-score, it's just dividing the random distribution variable minus the mean and dividing that by the standard deviation. If we assume normality, you can prove that this equation here can be transformed to something like this. So, let's say we want a 95 percent confidence interval, something like this. You can prove that the way the probability of the mean can fit into that interval is something this. So, we have this x here. We have 1.96 that we'll see soon about what is that. We have sigma, and we have the square root of our mean, sorry, of our population size. And we are saying that our mu, our mean is going to move into those intervals. If we apply the central limit theorem for a large size n, the above equation, so this one here, is approximately true. We can say that for a 95 percent confidence interval for mu when sigma is known, then you have this confidence interval here. This is the definition. So, the way you can plot, you can have this type of plot, is dependent on you defining this here. Or generally, if we don't want to say we have 95 percent or we want a general way of defining our confidence interval without being 95 percent, it can be 99 percent, 90 percent, 68 percent, we can define something called zeta alpha over 2, and this is the value that cuts off an area of alpha over 2 in the upper tail, in the standard normal distribution, and if we do this, we can define 1 minus alpha confidence interval for the population mean as something like this. So, this equation here is a very important equation in the world of statistics, and it's just saying that depending on how big or small we want our confidence interval, we have to calculate this zeta alpha over 2 here to get the lower and upper limit of our confidence interval. To calculate a 95 percent confidence interval, we can use the interval function or the ppf function that you saw in our videos. The way to do this is that we import it, we imported stats, and we can use norm interval. So, in here, I'm using the normal distribution, and inside of this you have this equation here. A 95 percent confidence interval is giving us the interval between minus 1.95 and 1.95.

Also, calculate this by hand, like not by hand but in more steps, the ppf and having 1 minus this here, and this is something close to this, and you have the lower limit and the upper limit. We can calculate the interval like this. So, let's say you have to define something called the interval here, and we're going to say that's called alpha, and that's going to be 95 percent, and the interval end is 1 minus 1 minus over alpha over 2, and this is the definition we have here. Now, we need to calculate the z multiplier. The z multiplier is this value here, and that depends on the ppf and the interval end, and then we define our standard deviation or x bar that is right here and our n. If we do that, we have, and then we can use the standard calculation, and that means by hand implementing this equation here. So, this x bar minus zeta multiple, zeta multiple will always be this zeta alpha over 2, sd is standard deviation, over the square root of n, so you have all of this equation right here. So, you can do this by hand, or you can use the interval function. So, if you do that, you get the same result but in a much easier way. You only have to pass the alpha, the loc that it will be like the mean, and the scale that is just standard deviation over the square root of the population size. So, this is a way we can define confidence intervals in Python. It's very simple to use. The concept is very important as well. Make sure you understand what it means to have a lower limit and upper limit and what it means to have a zeta score and an actual representation of our mean and/or sigma. Something very important here is that if you want to use all of the things I just mentioned, you need to know mu and sigma, because if you don't know them, you can calculate any of this. And we'll see in later videos what we can do if we don't know this sigma.

****

Video 6: Z-score vs. T-score (4:16)

As I mentioned in the last video, we use z-score when we have a normal distribution, and now I'm going to define that when we have a t-distribution we use the t-score. When you don't know the population variance or sigma, you use the t-score. So, if you know them, you will use the variance, and you will use the z-score. Almost all of the cases you don't know sigma, or you don't know the variance, so you will be always using the t-score for the confidence intervals. So, it's going to be much more common that you'll find yourself using the t-score instead of the z-score. The idea of the confidence interval is to mitigate the issue of population versus sample, so if you know all the parameters, it's not that common that you will be using at all a confidence interval. As we saw in the last video, the stats package has a library for the t distribution. And we also saw that it's very simple and similar to the normal distribution but with the difference that you have to define the degrees of freedom. And you can remember that the degrees of freedom in this case is n minus 1. So, if you have 21 observations, the degrees of freedom will be 20. Just an example here. Well let's calculate the t-score. The equation is not that simple, but it's not that complicated at all, and you can find that in every book or package you want to see in the world of statistics. So, let's say we have some observations here, and we want to find a confidence interval of 95 percent with a t-score. Well, this is how you do it. You have your observations in a list in Python, and the first thing you need to do is calculate n as the number of observations, and then you calculate x bar. That is also known as the sample mean. Then you calculate the standard deviation of your calculation of your variable here of your list. You define what will be the alpha. In this case, that means how big is the confidence interval, and now we're going to use just the stats package t interval calculated t multiplier. This t multiplier will be this t alpha over 2 df. And we're going to define that as n minus 1, and you're going to get the first 1 here.

So, if you do that, and you plug that into the equation of x bar minus or plus the t multiplier and you multiply it by the standard deviation and divide that by the square root of the sample size, this is what you'll get. You'll get a confidence interval of 119.81 plus/minus 3.52, but I mean this is the longer way of calculating all of this. You can do this much more simpler by using the stats interval, and that's it. You just need to define the alpha here, the len of the observations, the loc, and the scale will be the sem for the standard deviation. And you have the same results. So, this is just a way for you to use the t-score, to define t-score. So, when you use z-score, you know the standard variance, and you calculate a z-multiplier that's also called the theta alpha over 2, or if you don't know the variance, you calculate the t-score, and you need to define the degrees of freedom. Now, it's very related to the number of cases we have is nnn [phonetic] minus 1 the cases, and you define a t multiplier that is also known as the t alpha over 2 df where df means the degrees of freedom. Reasonable to do in Python, as you can see here, and now we have all this knowledge, we can go to the next step, and let's talk about hypothesis testing.

****

Video 7: Hypothesis Testing Part 1: T-test (5:38)

For the final video in this module, we'll talk about hypothesis testing and the t-test. We'll also define the p-values. We'll do this with an example, and this is a theoretical example, and you can read this with more care if you want later. Suppose we have a hypothesized or baseline value that we call p, and we want to obtain from, and then we obtain from our data a value called, let's call it p hat, that is smaller than p. If we're interested in thinking about whether p hat is significantly smaller than p, we have to quantify, and this will be to assume the true value where p and then we need to compute the probability of getting a value smaller or as small as the one we observed. And we can do the same thing for the case where if p hat was bigger or larger than p. If the probability is very low, we can think that the hypothesized value for p is incorrect. And this is the framework of hypothesis testing. This is the idea behind all of this knowledge. The way we do this is we begin with a null hypothesis, and we will call that h naught. This is the hypothesis that the true portion is in fact p. And we also need an alternative hypothesis that we'll call H1 or the hypothesis that the true mean is significantly smaller than p. Here we have the mean, but we can use whatever we want. When we have the two hypotheses, we'll use the data to test why hypothesis we should believe. Here, the word significance is defined in terms of a probability threshold called alpha. Such that we deem in a particular result significant if the probability of obtaining the result under the null distribution is less than alpha. A common value for alpha is 0.05, corresponding to 1 over 20 chance of error. So, that's a very small error. Once we obtain a particular value and evaluate its probability under the null hypothesis, that probability is known as a p value. This is the definition of a p value. So, in these three paragraphs, we have defined a null hypothesis. We defined the testing for the hypothesis testing, and we defined alpha and also the p value. We need more definitions that will help you in the future, and one is the concept of a one-tailed hypothesis test, and in that type of test, we choose one direction or alternate hypothesis. We either hypothesize that the test is significantly big, or the statistic is significantly small. The two-tailed hypothesis test, we have both directions. We have to hypothesize that the test statistic is simply different from the predicted value We normally have two types of error when we think about statistics, and they're called the type I error or false positive, and the type II error or false negative. A type I error is the one that happens when the null hypothesis is true although we reject it.

So, remember that the probability of a type I error is alpha. And the alpha is the one we define here above. A false negative or type II error happens when the null hypothesis is false, but we fail to reject it. So, this is a gaming with words because it's weird we talk like this, like we fail to reject something, we say something is false because we fail to reject that it was true. So, it's a weird way of saying things. This is the standard way of talking statistics. Finally, the statistical power of a test is the probability of rejecting the null hypothesis when it's false or equivocally 1 minus the probability of type error II. So, with all of these definition we can get started with the most famous statistical test, and that's called the t-test. A t-test is a type of inferential statistic used to determine if there is a significant difference between the mean of two groups, which may be related in certain features. You can also have a definition for one side, and we'll see that, but it's normally defined in terms of seeing if two means are different or not. The t-test looks for three things important, and that's the t-statistic or the t-score we define, the t-distribution values that we calculated also, and the degrees of freedom to determine the significance of your result. When you define a t-test for the difference in mean, the definition of the null hypothesis is the mean of the first distribution minus the second one is equal to 0, so they're the same, or the alternate hypothesis is that they're not the same, so their difference has to be different than 0. And remember here, mu is the means for each population.

****

Video 8: Hypothesis Testing Part 2: One-sided T-test (5:08)

So, normally we have two types of t-test. One is called the one-sided t-test and the two-sided t-test. When we have a one-sided t-test, we want to see whether the means of the sample are the same, is the same as the mean of the population. We'll do this with an example in Python. We'll use data for a body temperature. If you search online, you'll find that the standard temperature for a normal male is 98.6 degrees Fahrenheit. In this case, we'll test how likely it is that our sample mean is equal to that. So, we have the knowledge that globally a standard male, a normal standard male will have a temperature of an average of 98.6. But what we want to do is to see that if this sample we have comes from that type of distribution or this is a different type of data set. The data is stored in a csv file called body temp. We don't have pandas defined here, so I'm going to import pandas as pd. And now, let's see our data here.

Zero is for male and 1 is for female. We have male and female data here. We also have degrees per minute. That's for the heartbeat but we're only going to use here the sex variable and the temp variable. The way we define a one-sided t-test in Python is that we need to define the true mean. What's the true mean in here is the average temperature of a normal male in the world? And we need to say that we want to see if our mean is equivalent to that. But the way we do this is we use scipy stats, and we use the function called t-test 1samp. So, this is a one-sided t-test in Python. We are only going to take the male. So, this is a filtering instruction that you saw in other videos, and we're going to take only the temperature here. And we're going to calculate if the mean of this here is going to be if it's the same to the true mean we have. If we do this, as you can see here, we have a very, very, very small p value. What does that mean? That there's only a 0.00003 chance that we will see this result from purely random data. What does that mean? It's that if I encounter some data in the street and I take it, and that's random data, of course, it's very unlikely that that population, that this what we have here in the sample comes from that random thing I found on the street and not from the, like when male temperatures are measured in the world. So, we're very sure here that this temperature is very similar to the average temperature here. Let's just do another example. I'm going to create some normal distribution data. We're going to have with mu 5 and scale 10, and I'm going to create two sets of data. So, it's going to be like an array of data, and I'm going to test if my distribution comes from the normal test with a normal distributed data with mean 5. This is what we have here, and you have two p values. These two p values are bigger than 0.05, so this means for you that if you remember what we said here, we failed to reject the null hypothesis, so this cannot be the case that these are coming from that type of distribution. Let's go to the two-sided t-test. This is a test where you have two data samples, and you have different means or maybe the same. And what you want to do is that the null hypothesis for you is that both groups have the same or equal mean.

We don't know, we don't need to know a population parameter for this. We only need to know the data. So, we only need to have the data here. And the function t-test ind that is the way we do this in Python will take care of calculating the mean of each of these. So, we run this here, and as you can see here, we have a p value of 0.02. That means that there's only a two percent chance that we will see this result from purely random data. So, even though they're not exactly the same, it's supposed, I mean they're very close to each other, and it's very unlikely that they're just coming from random, purely random data.

****

Video 9: Hypothesis Testing Part 3: Confidence Intervals for Difference in Means (3:46)

You can also calculate a 95% interval of a t-test, or like you're going to calculate a confidence interval, or a difference in the means. And you do this with a z-distribution. And that is, again, the normal distribution with c means here a standard normal distribution with mu 0 and sigma equal to 1. So, let's say I have these two lists of observations here and I want to calculate the 95% interval confidence that they have the same mean. So, I'm going to do this by hand. So, it's going to be, it's much more simpler to do it with a t-test, but now let's say I know the variance and I know all of this, and I'm just going to calculate the confidence interval by hand using the equation we saw above. The first thing is calculate the mean. This is very simple with num pi. Calculate standard deviation, calculate the sizes of the samples. We set our confidence level that it is .95, that means a 95% confidence interval. We calculate the observed difference of the means. The standard the standard error, it is not the same standard deviation as the standard error. It is just the square root with standard deviation squared divided by the size of the first population, plus the standard deviation of the second population divided by the second population. And finally calculate a z-multiplier we saw before in other video to be in order to have this equation for a confidence interval. We have all of that, we can calculate our lower t and our upper t, and that's just the difference between the mean, between the minus, the z multiplier multiplied by the standard error. That is minus for lower and upper with a plus. We do this and now our confidence interval is just this. Okay, so, that's our confidence interval and we calculated that. And so, this is the way we do this with a z-distribution, so calculated the confidence interval for difference in means with all the things we know. If you want to know the p value, you have to use the t statistic. And you calculate that by, the t stat is just the difference in means divided by standard error. And the p value comes from the cdf function in norm here, just like in other videos, and we just round it here to four decimals. And this is our p value. So, this is the way we use the constant of p values, hypothesis testing, t-test, one-sided, two-sided, and we also have the hypothesis.

And all of these definitions here are very important for you to understand. And so, you have all the code. The code is the easy part here. Coding all of this in Python is just two lines of code very simple. Here, the interesting part is understanding the concepts and see how can you apply it. But you see, the assignments in this module, you'll get the chance to practice all of this a lot. And you'll see how to calculate a t-test and the p values and all of this information. So, thank you for being here in this module with me again. I'm hoping that this was helpful for you. We covered a lot of different things. We covered distributions, a lot of different distributions. And then we covered confidence intervals, c score, t scores, and finally, p values, confidence intervals, and hypothesis testing. Hopefully, you learned a lot and see you in the modules.

****

---END---