Topic 8: Distributions

What is Data Distribution?

Data Distribution is a list of all possible values, and how often each value occurs.

Such lists are important when working with statistics and data science.

The random module offers methods that returns randomly generated data distributions.

Random Distribution

A random distribution is a set of random numbers that follow a certain *probability density function*.

Probability Density Function: A function that describes a continuous probability. i.e. probability of all values in an array.

We can generate random numbers based on defined probabilities using the `choice()` method of the `random` module.

The `choice()` method allows us to specify the probability for each value.

The probability is set by a number between 0 and 1, where 0 means that the value will never occur and 1 means that the value will always occur.

Example

Generate a 1-D array containing 100 values, where each value has to be 3, 5, 7 or 9.

The probability for the value to be 3 is set to be 0.1

The probability for the value to be 5 is set to be 0.3

The probability for the value to be 7 is set to be 0.6

The probability for the value to be 9 is set to be 0

```
from numpy import random

x = random.choice([3, 5, 7, 9], p=[0.1, 0.3, 0.6, 0.0], size=(100))

print(x)
```

The sum of all probability numbers should be 1.

Even if you run the example above 100 times, the value 9 will never occur.

You can return arrays of any shape and size by specifying the shape in the `size` parameter.

Example

Same example as above, but return a 2-D array with 3 rows, each containing 5 values.

```
from numpy import random

x = random.choice([3, 5, 7, 9], p=[0.1, 0.3, 0.6, 0.0], size=(3, 5))

print(x)
```

Normal Distribution

The Normal Distribution is one of the most important distributions.

It is also called the Gaussian Distribution after the German mathematician Carl Friedrich Gauss.

It fits the probability distribution of many events, eg. IQ Scores, Heartbeat etc.

Use the `random.normal()` method to get a Normal Data Distribution.

It has three parameters:

`loc` - (Mean) where the peak of the bell exists.

`scale` - (Standard Deviation) how flat the graph distribution should be.

`size` - The shape of the returned array.

Example

Generate a random normal distribution of size 2x3:

```
from numpy import random

x = random.normal(size=(2, 3))

print(x)
```

Example

Generate a random normal distribution of size 2x3 with mean at 1 and standard deviation of 2:

```
from numpy import random

x = random.normal(loc=1, scale=2, size=(2, 3))

print(x)
```

Visualization of Normal Distribution

Example

```
from numpy import random
import matplotlib.pyplot as plt
import seaborn as sns

sns.distplot(random.normal(size=1000), hist=False)

plt.show()
```

Uniform Distribution

Used to describe probability where every event has equal chances of occuring.

E.g. Generation of random numbers.

It has three parameters:

`a` - lower bound - default 0 .0.

`b` - upper bound - default 1.0.

`size` - The shape of the returned array.

Example

Create a 2x3 uniform distribution sample:

```
from numpy import random

x = random.uniform(size=(2, 3))

print(x)
```

Visualization of Uniform Distribution

Example

```
from numpy import random
import matplotlib.pyplot as plt
import seaborn as sns

sns.distplot(random.uniform(size=1000), hist=False)

plt.show()
```

Chi Square Distribution

Chi Square distribution is used as a basis to verify the hypothesis.

It has two parameters:

`df` - (degree of freedom).

`size` - The shape of the returned array.

Example

Draw out a sample for chi squared distribution with degree of freedom 2 with size 2x3:

```
from numpy import random

x = random.chisquare(df=2, size=(2, 3))

print(x)
```

Visualization of Chi Square Distribution

Example

```
from numpy import random
import matplotlib.pyplot as plt
import seaborn as sns

sns.distplot(random.chisquare(df=1, size=1000), hist=False)

plt.show()
```

Binomial Distribution

Binomial Distribution is a *Discrete Distribution*.

It describes the outcome of binary scenarios, e.g. toss of a coin, it will either be head or tails.

It has three parameters:

`n` - number of trials.

`p` - probability of occurence of each trial (e.g. for toss of a coin 0.5 each).

`size` - The shape of the returned array.

Discrete Distribution: The distribution is defined at separate set of events, e.g. a coin toss's result is discrete as it can be only head or tails whereas height of people is continuous as it can be 170, 170.1, 170.11 and so on.

Example

Given 10 trials for coin toss generate 10 data points:

```
from numpy import random

x = random.binomial(n=10, p=0.5, size=10)

print(x)
```

Visualization of Binomial Distribution

Example

```
from numpy import random
import matplotlib.pyplot as plt
import seaborn as sns

sns.distplot(random.binomial(n=10, p=0.5, size=1000), hist=True, kde=False)

plt.show()
```

Hypothesis Testing a Mean

The following steps are used for a hypothesis test:

1. Check the conditions
2. Define the claims
3. Decide the significance level
4. Calculate the test statistic
5. Conclusion

For example:

- **Population**: Nobel Prize winners
- **Category**: Age when they received the prize.

And we want to check the claim:

"The average age of Nobel Prize winners when they received the prize is **more** than 55"

By taking a sample of 30 randomly selected Nobel Prize winners we could find that:

The mean age in the sample ($\bar{x}$) is 62.1

The standard deviation of age in the sample (s) is 13.46

From this sample data we check the claim with the steps below.

1. Checking the Conditions

The conditions for calculating a confidence interval for a proportion are:

- The sample is randomly selected
- And either:
    - The population data is normally distributed
    - Sample size is large enough

A moderately large sample size, like 30, is typically large enough.

In the example, the sample size was 30 and it was randomly selected, so the conditions are fulfilled.

Note: Checking if the data is normally distributed can be done with specialized statistical tests.

2. Defining the Claims

We need to define a **null hypothesis** (H0) and an **alternative hypothesis** (H1) based on the claim we are checking.

The claim was:

"The average age of Nobel Prize winners when they received the prize is **more** than 55"

In this case, the **parameter** is the mean age of Nobel Prize winners when they received the prize (μ).

The null and alternative hypothesis are then:

Null hypothesis: The average age was 55.

Alternative hypothesis: The average age was **more** than 55.

Which can be expressed with symbols as:

H0: μ=55

H1: μ>55

This is a '**right** tailed' test, because the alternative hypothesis claims that the proportion is **more** than in the null hypothesis.

If the data supports the alternative hypothesis, we **reject** the null hypothesis and **accept** the alternative hypothesis.

3. Deciding the Significance Level

The significance level (α) is the **uncertainty** we accept when rejecting the null hypothesis in a hypothesis test.

The significance level is a percentage probability of accidentally making the wrong conclusion.

Typical significance levels are:

- α=0.1 (10%)
- α=0.05 (5%)
- α=0.01 (1%)

A lower significance level means that the evidence in the data needs to be stronger to reject the null hypothesis.

There is no "correct" significance level - it only states the uncertainty of the conclusion.

Note: A 5% significance level means that when we reject a null hypothesis:

We expect to reject a **true** null hypothesis 5 out of 100 times.

4. Calculating the Test Statistic

The test statistic is used to decide the outcome of the hypothesis test.

The test statistic is a standardized value calculated from the sample.

The formula for the test statistic (TS) of a population mean is:

$$\bar{x} - \mu s \cdot n$$

$\bar{x} - \mu$ is the **difference** between the **sample** mean ($\bar{x}$) and the claimed **population** mean (μ).

s is the sample standard deviation.

n is the sample size.

In our example:

```
The claimed (H0) population mean (μ) was 55

The sample mean (x¯) was 62.1

The sample standard deviation (s) was 13.46

The sample size (n) was 30
```

So the test statistic (TS) is then:

$$62.1 - 5513.46 \cdot 30 = 7.113.46 \cdot 30 \approx 0.528 \cdot 5.477 = 2.889 —$$

You can also calculate the test statistic using programming language functions:

Example

With Python use the scipy and math libraries to calculate the test statistic.

```
import scipy.stats as stats
import math

# Specify the sample mean (x_bar), the sample standard deviation (s), the mean claimed in the null-hypothesis
(mu_null), and the sample size (n)
x_bar = 62.1
s = 13.46
mu_null = 55
n = 30

# Calculate and print the test statistic
print((x_bar - mu_null)/(s/math.sqrt(n)))
```

Example

With R use built-in math and statistics functions to calculate the test statistic.

```
# Specify the sample mean (x_bar), the sample standard deviation (s), the mean claimed in the null-hypothesis
(mu_null), and the sample size (n)
x_bar <- 62.1
s <- 13.46
mu_null <- 55
n <- 30

# Output the test statistic
(x_bar - mu_null)/(s/sqrt(n))
```

5. Concluding

There are two main approaches for making the conclusion of a hypothesis test:

- The **critical value** approach compares the test statistic with the critical value of the significance level.
- The **P-value** approach compares the P-value of the test statistic and with the significance level.

Note: The two approaches are only different in how they present the conclusion.

The Critical Value Approach

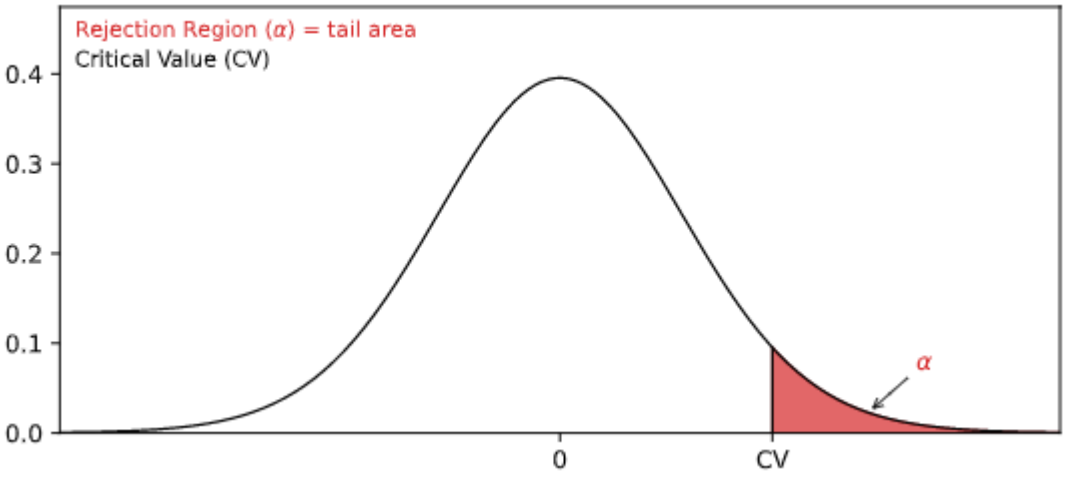
For the critical value approach, we need to find the **critical value** (CV) of the significance level (α).

For a population mean test, the critical value (CV) is a **T-value** from a student's t-distribution.

This critical T-value (CV) defines the **rejection region** for the test.

The rejection region is an area of probability in the tails of the standard normal distribution.

Because the claim is that the population mean is **more** than 55, the rejection region is in the right tail:



The size of the rejection region is decided by the significance level (α).

The student's t-distribution is adjusted for the uncertainty from smaller samples.

This adjustment is called degrees of freedom (df), which is the sample size (n)−1

In this case the degrees of freedom (df) is: 30−1=29—

Choosing a significance level (α) of 0.01, or 1%, we can find the critical T-value from a T-table, or with a programming language function:

Example

With Python use the Scipy Stats library `t.ppf()` function find the T-Value for an α = 0.01 at 29 degrees of freedom (df).

```
import scipy.stats as stats
print(stats.t.ppf(1-0.01, 29))
```

Example

With R use the built-in `qt()` function to find the t-value for an α = 0.01 at 29 degrees of freedom (df).

```
qt(1-0.01, 29)
```

Using either method we can find that the critical T-Value is ≈2.462—

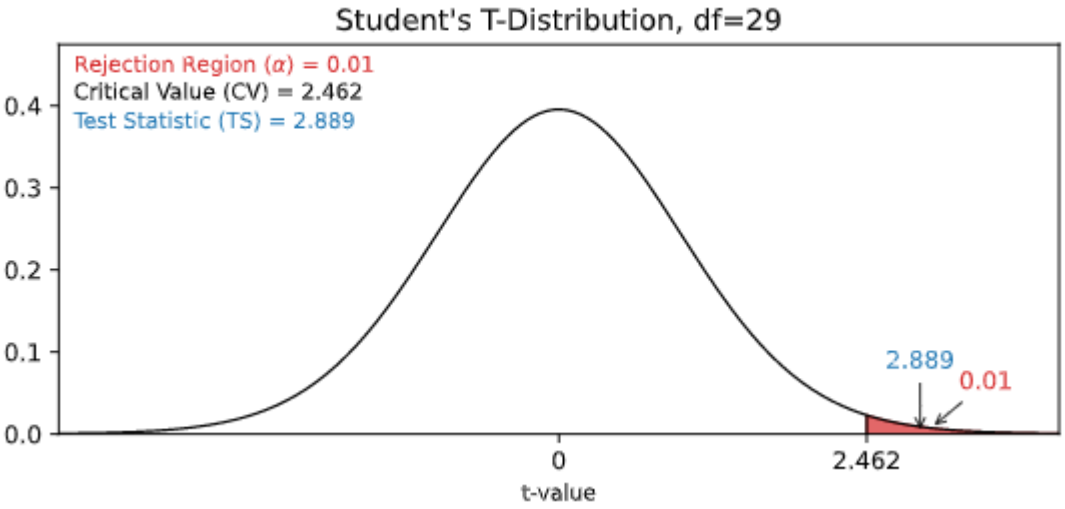
For a **right** tailed test we need to check if the test statistic (TS) is **bigger** than the critical value (CV).

If the test statistic is bigger than the critical value, the test statistic is in the **rejection region**.

When the test statistic is in the rejection region, we **reject** the null hypothesis (H0).

Here, the test statistic (TS) was ≈2.889— and the critical value was ≈2.462—

Here is an illustration of this test in a graph:



Since the test statistic was **bigger** than the critical value we **reject** the null hypothesis.

This means that the sample data supports the alternative hypothesis.

And we can summarize the conclusion stating:

```
The sample data supports the claim that "The average age of Nobel Prize winners when they received the prize is more
than 55" at a 1% significance level.
```

The P-Value Approach

For the P-value approach we need to find the **P-value** of the test statistic (TS).

If the P-value is smaller than the significance level (α), we **reject** the null hypothesis (H0).

The test statistic was found to be ≈2.889—

For a population proportion test, the test statistic is a T-Value from a student's t-distribution.

Because this is a **right** tailed test, we need to find the P-value of a t-value **bigger** than 2.889.

The student's t-distribution is adjusted according to degrees of freedom (df), which is the sample size (30)−1=29—

We can find the P-value using a T-table, or with a programming language function:

Example

With Python use the Scipy Stats library `t.cdf()` function find the P-value of a T-value bigger than 2.889 at 29 degrees of freedom (df):

```
import scipy.stats as stats
print(1-stats.t.cdf(2.889, 29))
```

Example

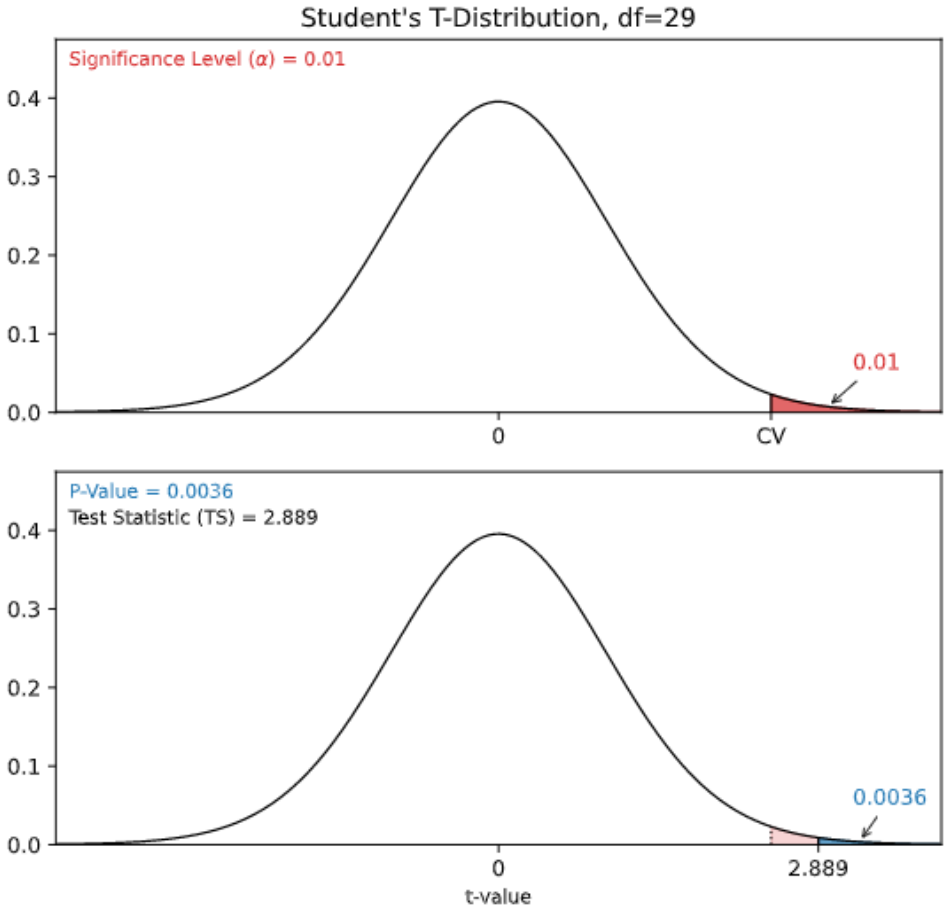
With R use the built-in `pt()` function find the P-value of a T-Value bigger than 2.889 at 29 degrees of freedom (df):

```
1-pt(2.889, 29)
```

Using either method we can find that the P-value is ≈0.0036—

This tells us that the significance level (α) would need to be bigger than 0.0036, or 0.36%, to **reject** the null hypothesis.

Here is an illustration of this test in a graph:



This P-value is **smaller** than any of the common significance levels (10%, 5%, 1%).

So the null hypothesis is **rejected** at all of these significance levels.

And we can summarize the conclusion stating:

The sample data **supports** the claim that "The average age of Nobel Prize winners when they received the prize is more than 55" at a **10%, 5%, or 1% significance level**.

Note: An outcome of an hypothesis test that rejects the null hypothesis with a p-value of 0.36% means:

For this p-value, we only expect to reject a true null hypothesis 36 out of 10000 times.