HYPOTHESIS TESTING

> by Dr Juan H Klopper

- Research Fellow
- School for Data Science and Computational Thinking
- Stellenbosch University

PACKAGES USED IN THIS NOTEBOOK

```
import numpy as np
import pandas as pd
from scipy import stats
```

```
import plotly.graph_objects as go
import plotly.express as px
import plotly.io as pio
pio.templates.default = 'plotly_white'
```

INTRODUCTION

In this notebook, we develop the intuition about the **scientific method**, which comprises the process of **hypotheses testing**, building on our knowledge gained in the previous notebook on randomess and sampling.

We have seen previously then, that we can repeatedly sample from a population and build a distribution of a specific statistic based on each sample. Now we consider the place of a specific sampling in relation to the sampling distribution.

In reality, we only do a study once. We base our results on a sample and want to know how this relates to the population. Having calculated results pertaining to our sample of subjects, we and others who have read our results can infer the results to the population. To do this, we have to develop the understanding of how our one study results fits in with the distribution built if we could repeat the study many, many times over.

Below, we work through some practical examples to build this understanding.

SAMPLE BASED ON PROPORTIONS

Consider a population with two mutually exclusive traits, these being A and B. It is known that trait A is present in 27% of the population and the remainder, 73%, have trait B. Our population size is 3000. We take a random sample of 100 subjects from the population and find that 13% have trait A. We ask the question: *Is this proportion representative of the known population proportions?*

We start by creating the population using the numpy `choice` function. This time we add weights to each sample space element. The weights refer to the 27% and the 73%, expressed as fractions (that sum to 1.0) and passed as a list to the `p` argument.

```
np.random.seed(42)
population = np.random.choice(['A', 'B'], size=3000, p=[0.27, 0.73])
```

This array of values can be stored in a dataframe object.

```
df = pd.DataFrame({'Trait':population})
df[:5] # Using indexing instead of df.head()
```

	Trait
0	B
1	B
2	B
3	B
4	A

The `unique` method shows the sample space elements and the `value_counts` return the frequency of each.

```
df.Trait.unique()
```

```
array(['B', 'A'], dtype=object)
```

The `value_counts` method is used to return the frequency and realtive frequency (proportions) of the two sample space elements.

```
df.Trait.value_counts()
```

```
B    2186
A     814
Name: Trait, dtype: int64
```

```
df.Trait.value_counts(normalize=True) # Proportions
```

```
B    0.728667
A    0.271333
Name: Trait, dtype: float64
```

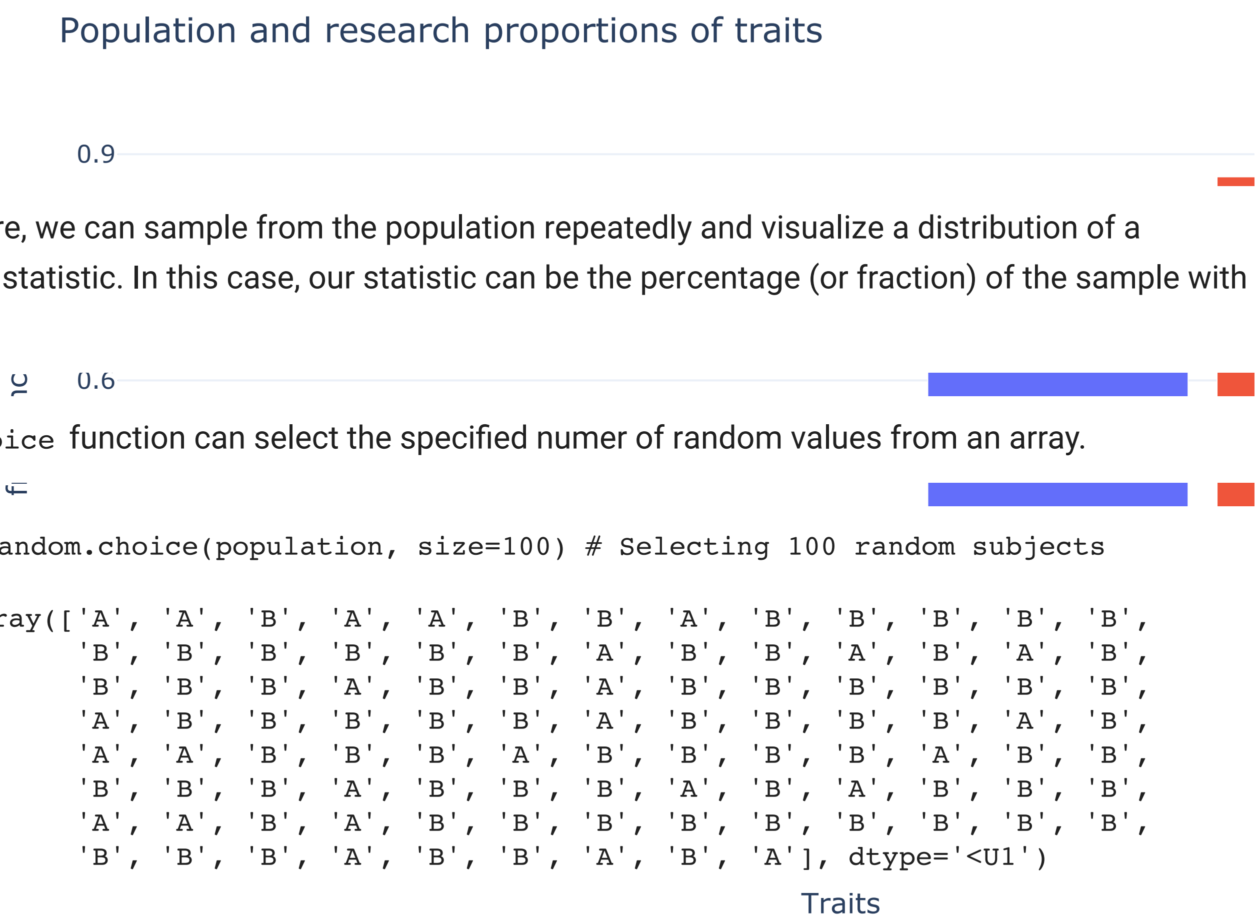
Remember that we use a bar chart to visualize the frequency of nominal categorical variables and below we view the proportions of the two sample space elements in this example.

```
px.bar(
    x=['A', 'B'],
    y=[0.27, 0.73],
    title='Relative frequency of traits in population',
    labels={
        'x':'Trait',
        'y':'Relative frequency'
    }
)
```

Hovering over the two bars shows the 0.27 and 0.73 proportions as expected.

Our imagined sample showed a relative frequency for the two traits as 0.13 and 0.87. A bar chart can visualize the research question proportions (*is the* $0.13 : 0.87$ *proportion representative*) and the population proportions ($0.13 : 0.73$).

```
go.Figure(
    data=go.Bar(
        x=['A', 'B'],
    y=[0.27, 0.73],
    name='Population proportions'
    )
).add_trace(
    go.Bar(
        x=['A', 'B'],
        y=[0.13, 0.87],
        name='Research proportions'
    )
).update_layout(title='Population and research proportions of traits',
                xaxis={'title':'Traits'},
                yaxis={'title':'Relative frequency'},
                bargap=0.2, # gap between bars of adjacent location coordinates
                bargroupgap=0.1) # gap between bars of the same location coordi
```



As before, we can sample from the population repeatedly and visualize a distribution of a specific statistic. In this case, our statistic can be the percentage (or fraction) of the sample with trait A.

The `choice` function can select the specified numer of random values from an array.

```
np.random.choice(population, size=100) # Selecting 100 random subjects
```

```
array(['A', 'A', 'B', 'A', 'A', 'B', 'B', 'A', 'B', 'B', 'B', 'B', 'B',
       'B', 'B', 'B', 'B', 'B', 'B', 'A', 'B', 'B', 'A', 'B', 'A', 'B',
       'B', 'B', 'B', 'A', 'B', 'B', 'A', 'B', 'B', 'B', 'B', 'B', 'B',
       'A', 'B', 'B', 'B', 'B', 'B', 'A', 'B', 'B', 'B', 'B', 'A', 'B',
       'A', 'A', 'B', 'B', 'B', 'A', 'B', 'B', 'B', 'B', 'A', 'B', 'B',
       'B', 'B', 'B', 'A', 'B', 'B', 'B', 'A', 'B', 'A', 'B', 'B', 'B',
       'A', 'A', 'B', 'A', 'B', 'B', 'B', 'B', 'B', 'B', 'B', 'B', 'B',
       'B', 'B', 'B', 'A', 'B', 'B', 'A', 'B', 'A'], dtype='<U1')
```

The numpy `unique` function return the sample space elements and with the `return_counts` argument set to `True`, it returns a 2-tuple. The first element is an array of the sample space elements and the second is an array of the frequencies of each of the sample space elements.

```
np.unique(np.random.choice(population, size=100), return_counts=True)
```

```
(array(['A', 'B'], dtype='<U1'), array([30, 70]))
```

We need the first element from the second array. We do this using indexing.

```
np.unique(np.random.choice(population, size=100), return_counts=True)[1][0]
```

```
26
```

Since our statistic is the proprotion of this first element above, we can divide it by the sample size.

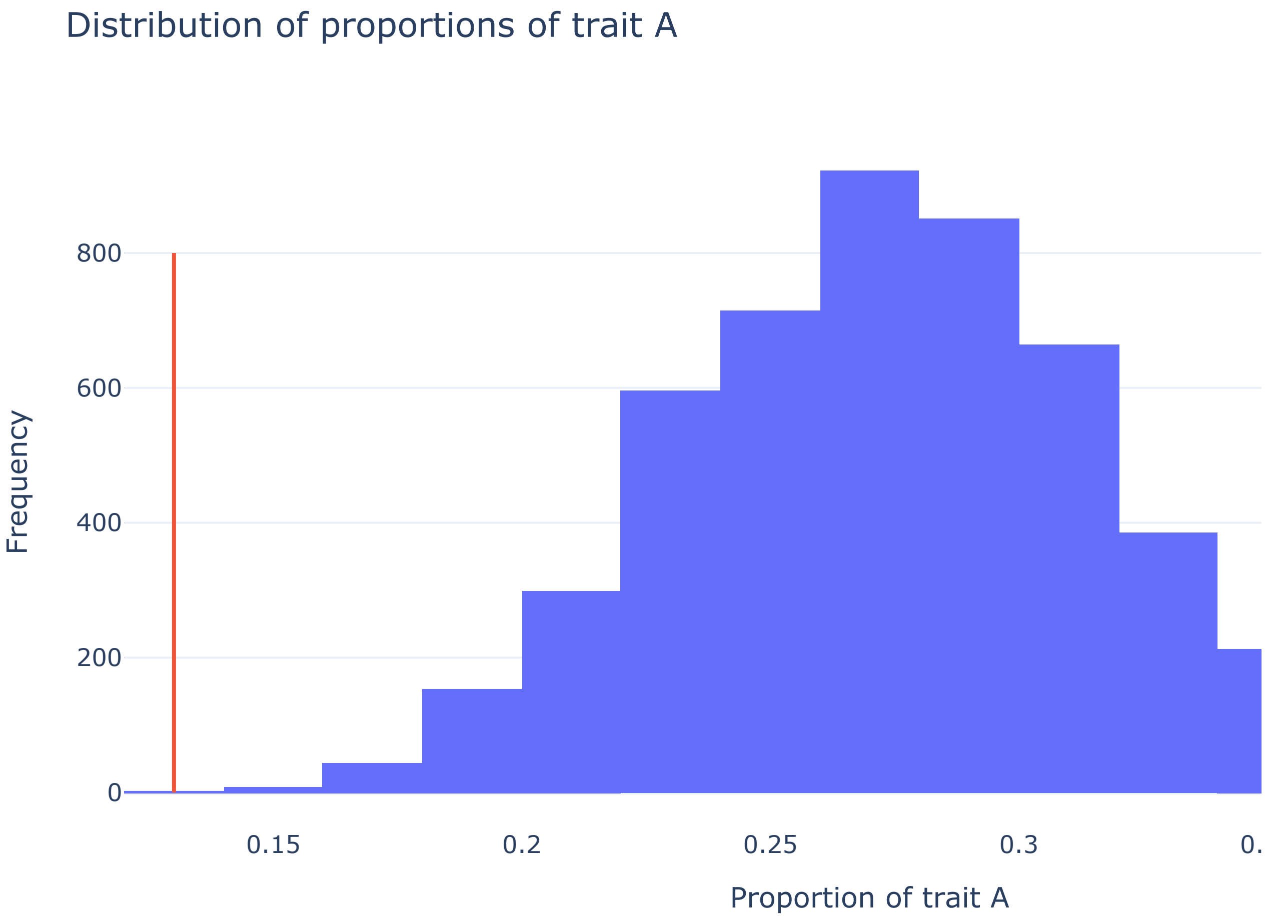
Below, we sample from the population 5000 times and record the proportion of subjects with trait A.

```
count = [] # Empty list to hold all the trait A proportions
n = 100 # Samples size

```

```
for i in range(5000):
  count.append(np.unique(np.random.choice(population, size=n), return_counts=Tru
```

No we look at a histogram of all the trait A proportions. We also add a red vertical line at our original 13%.

```
go.Figure(
    data=go.Histogram(
        x=count,
        nbinsx=20,
        name='Proprotions'
    )
).add_trace(go.Scatter(
    x=[0.13, 0.13],
    y=[0, 800],
    mode='lines',
    name='Original proportion'
)).update_layout(title='Distribution of proportions of trait A',
                 xaxis={'title':'Proportion of trait A'},
                 yaxis={'title':'Frequency'})
```



Our imagined proportion of 0.13 occured with a very low frequency according to the histogram. It was unlikely to have such a proportion. We can actual give a proportion of times that we had proportions of 0.13 and smaller in our simulation.

```
np.sum(np.array(count) < 0.13) / 5000
```

```
0.0004
```

A statistical test to see if a proportion in a sample is different from known proportions is the χ^2 test for proportions. For this we use the `chisquare` function in the stats module in scipy. We pass two arguments, `f_obs` and `f_exp`. The values in our example will be two list, each with two elements. We multiply the proportions by the sample size in both cases.

```
stats.chisquare(
    f_obs=[0.13 * 100, 0.87 * 100],
    f_exp=[0.27 * 100, 0.73 * 100]
)
```

```
Power_divergenceResult(statistic=9.944190766108575, pvalue=0.0016135778718655
```

The 13% from the research question was an unlikely finding based on the histogram. Expressed as a *p* value using the proportion test, we see a very small value, which is a reflection of the histogram and the proportion of 0.0004 that we caluclated.

EXAMPLE BASED ON A DIFFERENCE IN MEANS

In this example we know the value of a continous numerical variable in each subject in a population. The sample space elements are on the interval $[0, 100)$. The distribution of the elements takes on a uniform distribution in the popluation.

```
# The random function returns a value between 0 and 1
population = np.random.random(3000) * 100
```

Imagine then that the population is spread over two neighbouring towns. A researcher suspects that there is a difference in the value of this variable between the two towns (not having access to all the known values as we do). A random sample of 100 individuals from each town results in a mean value of 45.3 for town A and 52.8 for town B. How can the researcher asses this differrence?

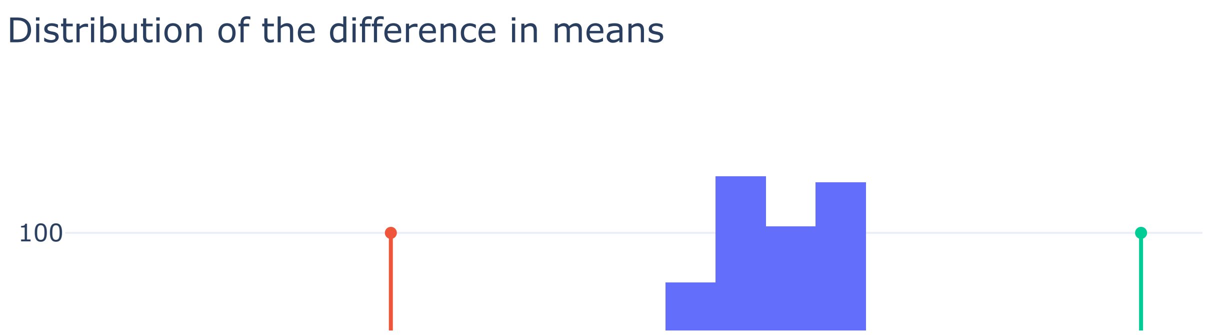
Once again, we resample repeatedly from the two towns and represent this simulation below. The test statistic is *difference in means*, with the researcher's difference being

$52.8 - 45.3 = 7.5$. Since there is no natural order between these two towns, we might also

We code our repeated sampling and visualize the distribution of our test statistic which is *difference in means*. We also visualize the researcher's difference in means.

```
difference = [] # Empty list to be populated by differences in for loop

for i in range(1000): # Loop 1000 times
  sample_A_ave = np.mean(np.random.choice(population, size=100)) # Mean of 100 s
  sample_B_ave = np.mean(np.random.choice(population, size=100)) # Mean of 100 s
  difference.append(sample_A_ave - sample_B_ave) # Append difference to list on
```

```
go.Figure(
    data=go.Histogram(
        x=difference,
        name='Difference distribution'
    )
).add_trace(
    go.Scatter(
        x=[-7.5, -7.5],
        y=[0, 100],
        name='A-B'
    )
).add_trace(
    go.Scatter(
        x=[7.5, 7.5],
        y=[0, 100],
        name='B-A'
    )
).update_layout(
    title='Distribution of the difference in means',
    xaxis={'title':'Difference in means'},
    yaxis={'title':'Frequency'}
)
```



It is worthwhile to note that the distribution of means seem to take on a bell-shaped curve, despite the fact that the variable was distributed uniformly in the population. We also note that the difference found by the researcher seems to have been *uncommon*. As with the proportion test above, there are statistical tests that can enumerate just how *uncommon* this finding was.

Assessing the research's finding follows the processes of the scientific method. These processes are termed **hypothesis testing** and is the corner stone of the scientific method.

HYPOTHESES TESTING

In hypothesis testing we have two views of our research question. We can see this as two views about how the data was generated. The two views are termed **hypotheses**. There are two hypotheses, the null and the alternative hypothesis.

The **null hypothesis** takes on a conservative approach. It states that the data was generated from clearly defined parameters and assumptions about randomness. Any deviation from the data generated from the null hypothesis is taken to be purely by chance. In our first eaxmple above, there was the assumption that the proportions of the traits in the population were $0.27 : 0.73$. In the second example it was that the data was on an interval from a uniform distribution. We simulated the data under these assumptions about randomness.

The **alternative hypothesis** states that something other than chance lead to a difference in the data from the prediction of the model under the null hypothesis.

Until we collect and analyze any data from a selected sample, we stand by the null hypothesis (about the distribution of the data in the population from which the sampe was taken). To *choose between* the two hypotheses, we require a statistic, termed the **test statistic**. In our example above, it was the proportion of the first trait and in the second, the difference in means.

The null hypothesis in the first example could be stated as: *The proportion of the A trait in the sample is* 0.27. The alternative hypothesis would then be: *The proportion of trait A in the sample* *is not* 0.27.

The null hypothesis in the second example could be stated as :*There is no difference in the means of the variable between the two towns*. The alternative hypothesis would then be: *There is a difference in the mean between the two towns*. Note that we do not subscribe which town has an average more or less than the other. This is referred to as a **two-tailed alternative hypothesis**. Depending on the order of subtraction, we would get a positive or a negative difference (unless they are equal, which is usually unlikely).

Simulation of possible test statistics using repeated sampling gave us a good idea of the distribution of the statistic and we could visualize *how likely* the research statisic was (the single instance of sampling that the researcher performed).

The question remains: *How unlikely (away from the most often found test statistics found during repeated sampling) must the research test statistic be before we reject the null hypothesis and accept the alternative hypothesis*? Note that if the test statistic is among the most often found test statistics, we fail to reject the null hypothesis. We can not accept or prove the null hypothesis. It is simply the finding given the assumptions.

By convention, we choose a *cut off* value to make this decision. This value is termed an α value and for various disciplines this is set at 0.05, or $0,01$, or even much smaller (particle physics comes to mind).

The mathematics that underlies the statistics for these tests consider a probability density function (PDF) and a cumulative distribution function (CDF) (in the case of constinuous numerical functions). The total area under the curve of the PDF is 1.0. In the case of the second example above, the area to the left of the red line added to the area to the right of the green line would represent the *p* value (to some approximation relevant to this discussion as a histogram is not a PDF). If this is less than the chosen α value, we reject the null hypothesis and accept the alternatibe hypothesis. Otherwise, we fail to reject the null hypothesis. Visually, the latter represents a *likely* statistic and the former an *unlikely statistic*. This is how some disciplines express statistical significance or finding a ststistically significant result.

Note that the α value is ARBITRARY.

As proper researchers, we have two hyothesis. With respect to continuous numerical variables for instance, the **null hypothesis** is our default and we state that there is no difference between the means, unless we collected evidence and it proves otherwise. Our **alternative hypothesis** is just that. There is a difference in the means. When the evidence (calculations) is not sufficient, we fail to reject the null hypothesis. If the evidence is there, we reject the null hypothesis and accept the alternative hypothesis. To do all this, we need an α value. We review how does this all fit together.

What we have learned above and remember form the previous notebook, is that our difference is but one of many, that will fall somewhere on a sampling distribution. Some test statistics occur commonly and some not so commonly. With a specific parameters (peratining to the test we use), we can construct a probability density function (PDF) and plot it. We find out where on the plot to draw our two horizontal lines that will show an area under the curve (using an α value of 0.05) to the left of the left-sided symmetrical line of 0.025 (2.5%), and another 2.5% to the right of the righ-sided symmetric vertical line. We calculated these symmetrical values using the `ppf` function (something we will later call critical values). The 2.5% reflects half of our $\alpha = 0.05$ decision.

Finally, we convert our test statistic appropriately and reflect it along the other side of the curve through symmetry. Our hypothesis is a two-tailed hypothesis (there is a difference), which depends which mean we subtract from which (resulting in a positive or a negative value).

Finally, we look towards negative and positive infinity from our *t* stistic lines and calculate the area under the curve or *p* value. If the *t* statistics are outside of two 5% lines (each at 2.5%), we will have a small *p* value (area). Given all the possible outcomes (differences in means), this would indicate that we discovered one of the lesser probable ones and reject our null hypothesis. We state that the difference is significant and (if the new drug had more of a reduction), we declare it it different from the old drug. If not, we fail to reject the null ypothesis and state that the two drugs are equally effective (using all these terms loosely).

To be sure, we also get one-tailed hypothesis. That is where we can make a strong argument that one mean will be more than the other. We then do not reflect the statistic one either side.

STATING A HYPOTHESIS BASED ON A RESEARCH QUESTION

Now that we know about hypothesis testing, let's put it to the test. More examples always help. We imagine a study where we are investigating a new intervention. We create two groups. In one, the participants receive a placebo intervention and in the other, a new intervention. In each group we meassure a certain variable for each individual. Our research question is: *Is there a difference in the variable between the placebo and intervention groups?*

It is an absolute must that we are able to state our research questions in a way that we can use hypothesis testing. In our research question above, we have a single variable and two groups. One group will recieve the new intervention and the other, a placebo (an empty) intervention. We will collect data point values for a variable following the interventions (real intervention and placebo) and measure the difference in the data between the two groups.

Our null hypothesis for this research question is: *There is no difference in the data for the variable between the two groups*. The null hypothesis is sometimes written as H_0.

How will we do this comparison, though? Well, that depends on the data type of the variable. Let's assume that it is a continuous numerical variable. If the assumptions for the use of parametric tests are met (which we will investigate in the next notebook), it means that we will compare the means of the variable between the two groups, i.e. the mean is our test statistic. If the placebo group has a mean for the variable of $\bar{X}_1$ and the new intervention group has a mean of $\bar{X}_2$, then we would state our nul hypothesis as: $H_0 : \bar{X}_1 = \bar{X}_2$. The means are equal.

Our alternative hypothesis would then be that the means are not the same. This is written as: $H_{\alpha} : \bar{X}_1 \neq \bar{X}_2$. What we have here is a two-tailed hypothesis. We merely state that there is a difference and we are not concerned with which group will have a mean of more or less than the other.

The aim is now to collect data and see if there is enough evidence to reject the null hypothesis and therefor accept the alternative hypothesis or, in the case that there is not enough evidence, to fail to reject the null hypothesis. These are important concepts. We never prove the null hypothesis. In fact, the sampling distributions on which we will base our statistical tests are created in view of the fact that no difference exists. Our study merely finds an unlikely difference or it does not.

To make the distinction between enough evidence or not, we set an α value. This is usually 0.05. If the area under the curve (in actual fact, the cummulative distribution function value) is less than the α value, i.e. a *p* value of less than the α value, we reject the null hypothesis and accept the alternative hypothesis. If not, then we fail to reject the null hypothesis.

GENERATING DATA

For the sake of some pratice, let's generate our own simulated data for our research question. We create two computer variables, one for the placebo group and one for the intervention group. Both sets of data point values for our imaginary variable will comes from a normal distribution.

For the intervention group, we choose a mean of 50 and a standard deviation of 5 and for the placebo group, a mean of 48 and a standard deviation of 7. We use the `norm.rvs()` function to generate the data.

```
intervention = stats.norm.rvs(loc=50,
                              scale=5,
                              size=100,
                              random_state=3)  # For reproducible results

placebo = stats.norm.rvs(loc=48,
                         scale=7,
                         size=100,
                         random_state=3)
```

Just as a sneak peek at how easy it is to calculate a *p* value, take a look the the line of code below. It returns a *t* statistic and a *p* value. Don't stare at it for too long, though. We will take the long route so that we understand how this is calculated.

```
stats.ttest_ind(intervention, placebo)
```

```
Ttest_indResult(statistic=2.4103616553305387, pvalue=0.016850966727719883)
```

Let's summarise and visualise our data. First, we look at the mean and then the standard deviation of the variable for each group.

```
print('Mean for intervention group: ', '\t', intervention.mean(), '\n',
      'Mean for placebo group: ', '\t', placebo.mean())
```

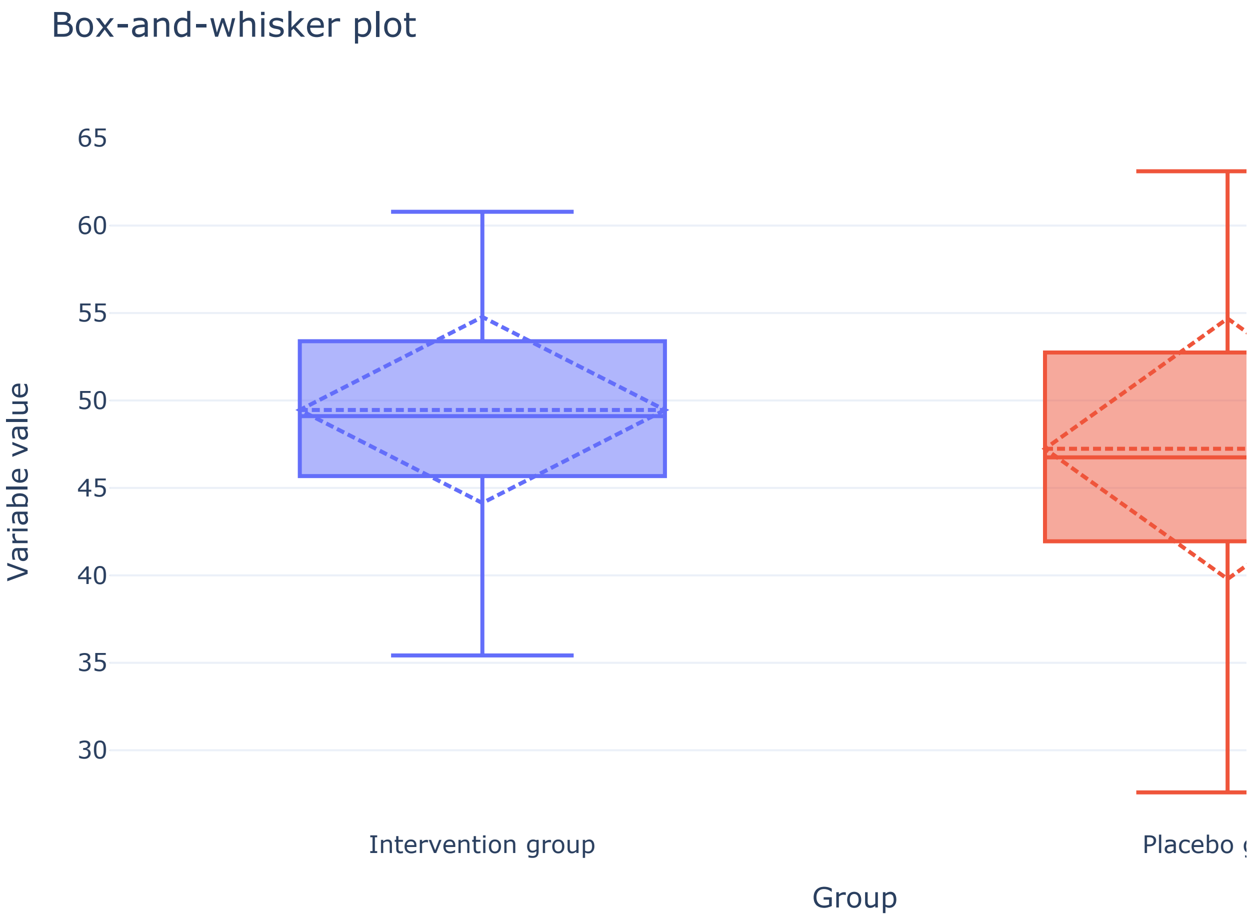
```
Mean for intervention group:     49.45681462796969
 Mean for placebo group:         47.23954047915756
```

```
print('Standard deviation for intervention group: ', '\t', intervention.std(),
      'Standard deviation for placebo group: ', '\t', placebo.std())
```

```
Standard deviation for intervention group:       5.319966842666896
 Standard deviation for placebo group:           7.447953579733653
```

A box-and-whisker plot will be more intuitive.

```
box_fig = go.Figure()

box_fig.add_trace(go.Box(y=intervention,
                         name='Intervention group',
                         boxmean='sd',
                         boxpoints='suspectedoutliers'))

box_fig.add_trace(go.Box(y=placebo,
                         name='Placebo group',
                         boxmean='sd',
                         boxpoints='suspectedoutliers'))

box_fig.update_layout(title='Box-and-whisker plot',
box_fig.update_layout(title='Box-and-whisker plot',
                      xaxis=dict(title='Group'),
                      yaxis=dict(title='Variable value'))

box_fig.show()
```



Take a guess. Do you think there is a statistically significant difference between the means?

IS THERE A DIFFERENCE?

The question now is whether there is a difference between the calculated means of 49.5 and 47.2. Well the difference in means are shown below. We can subtract one mean from the other in either order.

```
intervention.mean() - placebo.mean()
```

```
2.2172741488121304
```

```
placebo.mean() - intervention.mean()
```

```
-2.2172741488121304
```

We do remember from the previous notebook that this difference in means is but one of many possible means. Since we don't know the standard deviation for our variable in the population (we did not simulate a whole population and sample from it), we will make use of the *t* distribution. It is a theoretical sampling distribution based only on the sample size (known as the degrees of freedom). We have 200 participants in our study divided into two groups. To set up the *t* distribution, we need to known the degrees of freedom. This would simply be $200 - 2 = 198$. The sample size minus the number of groups.

Below, we create *t* distribution for 198 degrees of freedom.

```
t_vals = np.linspace(-3, 3, 200)  # Generating some values for the x-axis
t_pdf_vals = stats.t.pdf(t_vals, 198)  # Calculating the PDF value for each of


t_dist_fig = go.Figure()

t_dist_fig.add_trace(go.Scatter(x=t_vals,
                                y=t_pdf_vals,
                                mode='lines',
                                name='t distribution'))

t_dist_fig.update_layout(title='t distribution',
                         xaxis=dict(title='t values'),
                         yaxis=dict(title='PDF'))

t_dist_fig.show()
```

t distribution

Now we have to express our difference in means as a *t* statistic. We can use equation (1) below, where $\Delta \bar{X}$ is the difference in means.

$$t = \frac{\Delta \bar{X}}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}} \tag{1}$$

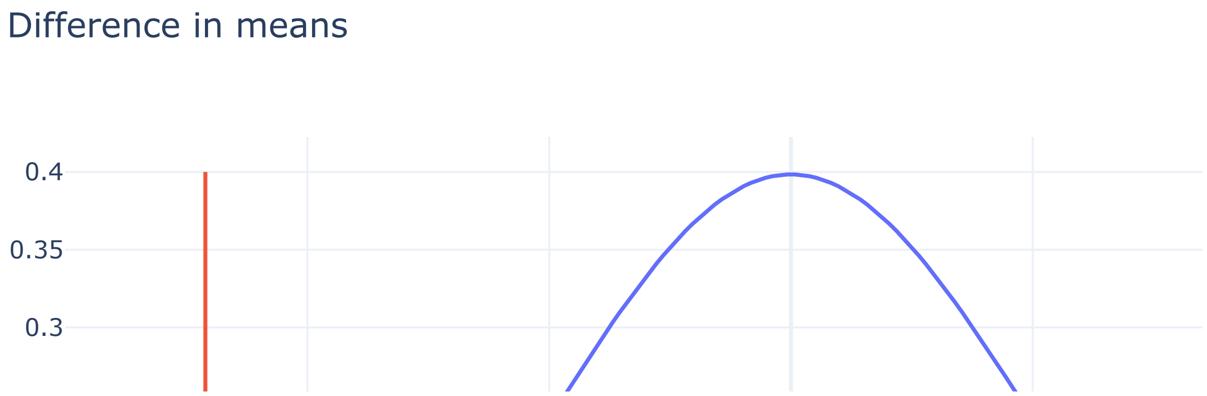
Let's go for a difference of -2.217, placebo group mean minus intervention group mean.

```
t_stat = (placebo.mean() - intervention.mean()) / (np.sqrt((placebo.std()**2 / 1
t_stat
```

```
-2.4225046120579825
```

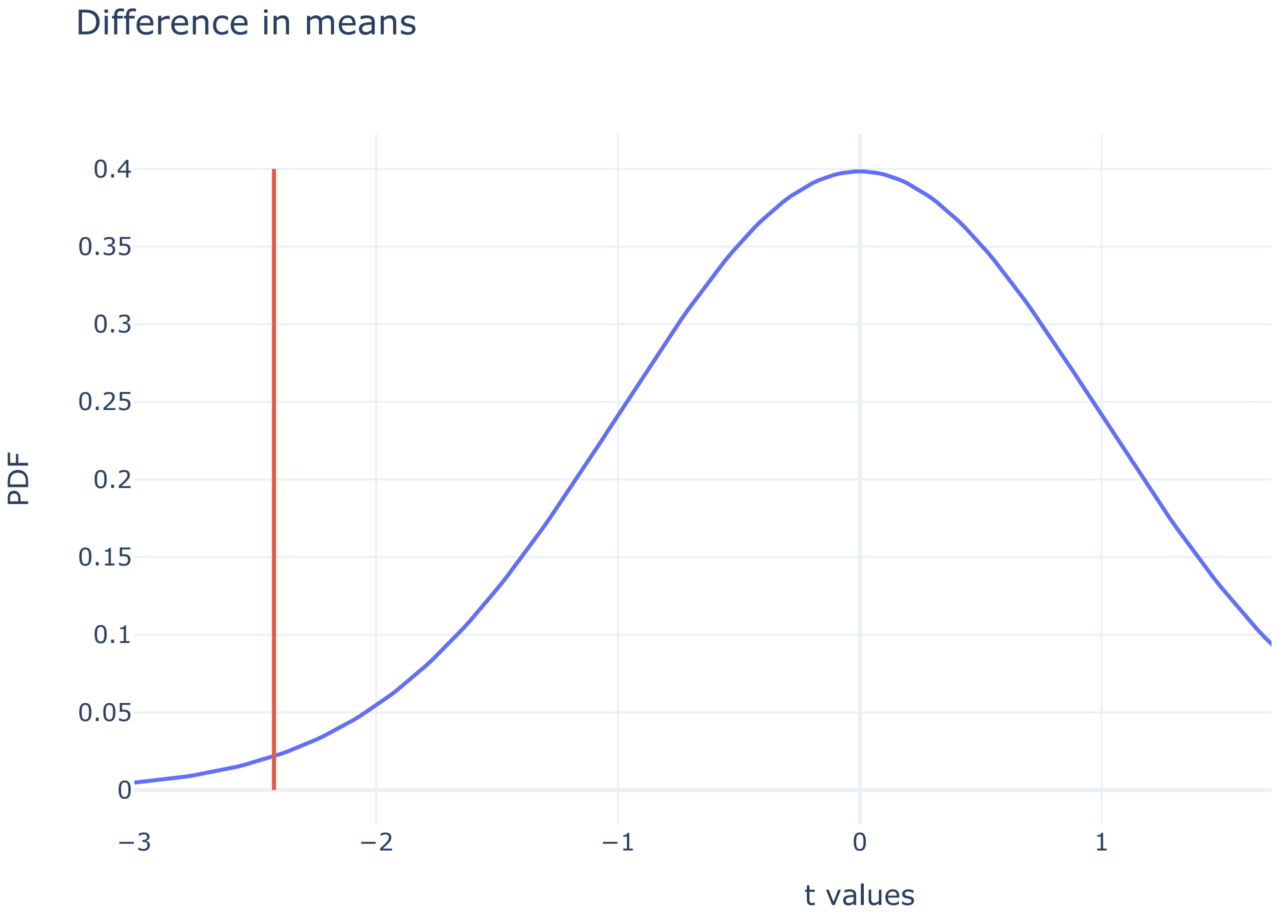
Now we have the *t* statistic value for the placebo group mean minus the intervention group mean. We can plot this as a horizontal line (in red below).

```
t_dist_fig.add_trace(go.Scatter(
    x=[t_stat, t_stat],
    y=[0,0.4],
    name='Placebo - Intervention',
    mode='lines'
))

t_dist_fig.update_layout(title='Difference in means')

t_dist_fig.show()
```



We have to refelect this on the other side as well for a two-tailed hypothesis. Our alternative hypothesis was that there was a difference, only.

```
t_dist_fig.add_trace(go.Scatter(
    x=[-t_stat, -t_stat],
    y=[0,0.4],
    name='Intervention - Placebo',
    mode='lines'
))

t_dist_fig.show()
```



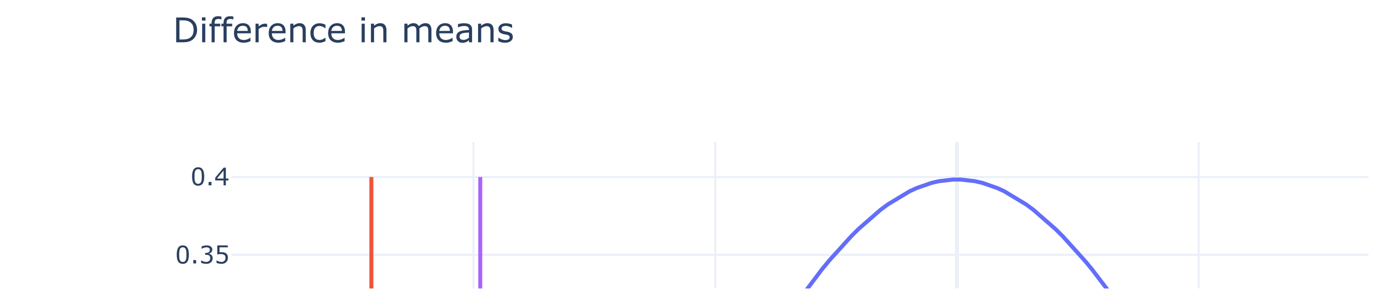
If we look at the area under the curve from negative infinity to the red line and from the green line to positive infinity, we are looking at the *p* value. To calculate this, we will simply calculate the value of the (red line) *t* statistic using the cummulative distribution function, `t.cdf`, and multiply it by 2.

```
stats.t.cdf(t_stat, 198) * 2
```

```
    0.01631418341067354
```

A *p* value of 0.02 (rounded). Smaller than our chosen α value of 0.05, for sure. This is because these *t* statistic values fall outsde of the critical *t* values. These are the values that would represent 2.5% of the area under the curve on either side. We add them below.

```
t_crit = stats.t.ppf(0.025, 198)

t_dist_fig.add_trace(go.Scatter(
    x=[t_crit, t_crit],
    y=[0,0.4],
    name='Critical t statistic',
    mode='lines'
))

t_dist_fig.add_trace(go.Scatter(
    x=[-t_crit, -t_crit],
    y=[0,0.4],
    name='Critical t statistic',
    mode='lines'
))

t_dist_fig.show()
```



To the left of the purple line and to the right of the orange line, we find our areas of rejection. Each of these areas are 2.5% of the area under the curve. See how our *t* statistic(s) are within the areas of rejection.

Finally, we have enough evidence to reject our null hypothesis and accept our alternative hypothesis. There is a statistically significant difference in our variable compared between the two groups.

ONE-TAILED HYPOTHESIS

It might very well be that our alternative hypothesis is one-tailed. This can be a dangerous decision. We have to be able to make a reasonable argument to convince our peers that we expected that one mean would be higher or lower than the other. For a problem such as ours (above) that would mean that the *p* value is divided by 2. It can be dangerous and tempting to change our minds after the analysis and go for a one-tailed alternative hypothesis, especially if the *p* value was close to 0.05. A hypothesis must be set during the study design and we cannot change that after the fact.

Just for argument's sake let's look at the one-tailed hypotheses. First, a reminder of the two means.

```
print('Mean for intervention group: ', '\t', intervention.mean(), '\n',
      'Mean for placebo group: ', '\t', placebo.mean())
```

```
Mean for intervention group:      49.45681462796969
 Mean for placebo group:          47.23954047915756
```

Let's make group 1 the placebo group and group 2 the intervention group. For our first scenario, we state that the mean of the placebo group is greater than or equal to the mean of the intervention group. The alternative hypothesis is then that the mean of the placebo group is less than that of the intervention group. We state this in eqution (2) below, where $\bar{X}_1$ is the mean of the placebo group and $\bar{X}_2$ is the mean of the intervention group. To be clear, the alternative hypothesis is the one we are *hoping* to show.

$$\begin{aligned} H_0 &: \bar{X}_1 \geq \bar{X}_2 \\ H_\alpha &: \bar{X}_1 < \bar{X}_2 \end{aligned} \tag{2}$$

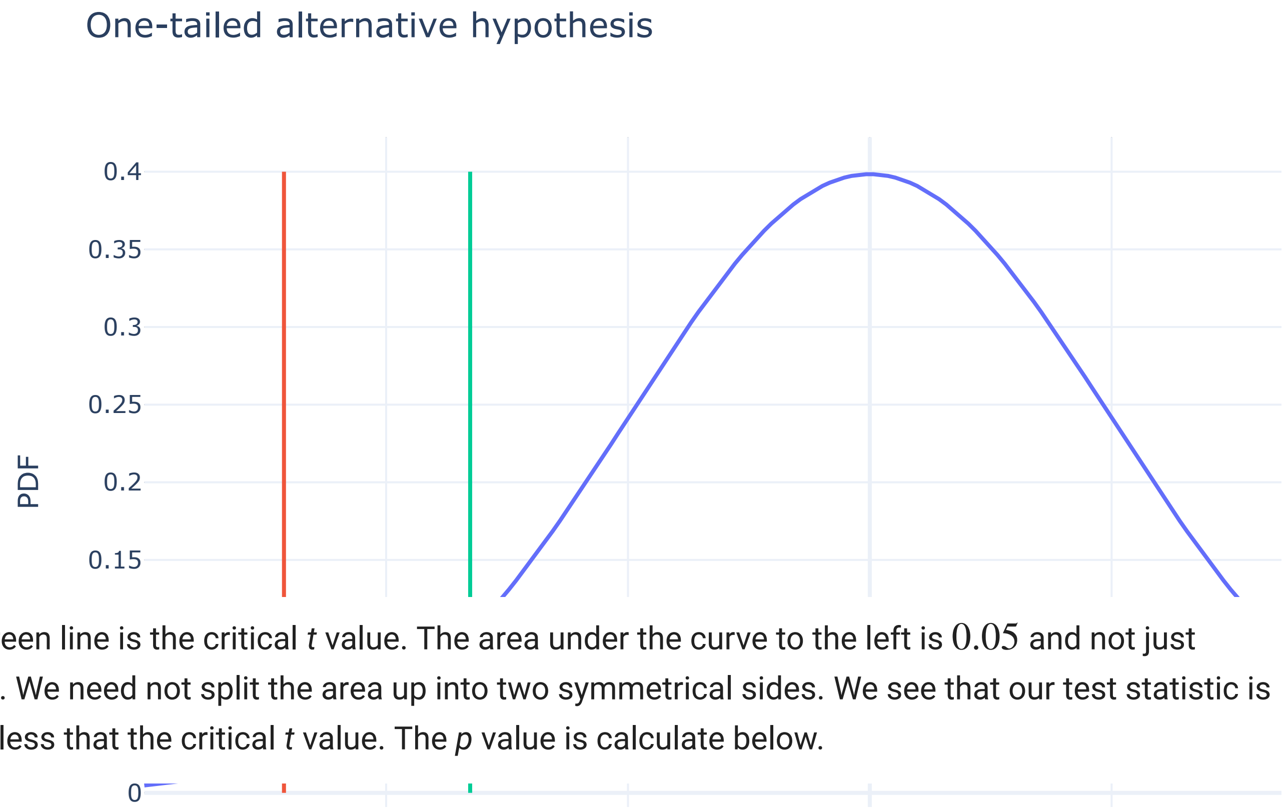
We need a critical *t* value which represents an area under the probability density curve which represents 0.05 of the total area (to the left). We can calculate this using the `ppf()` function, which we use below for 198 degrees of freedom.

```
t_crit = stats.t.ppf(0.05, 198)
t_crit
```

```
-1.6525857836172082
```

We can now plot this together with our actual *t* statistic.

```
t_dist_fig = go.Figure()

t_dist_fig.add_trace(go.Scatter(x=t_vals,
                                y=t_pdf_vals,
                                mode='lines',
                                name='t distribution'))

t_dist_fig.add_trace(go.Scatter(
    x=[t_stat, t_stat],
    y=[0,0.4],
    name='Placebo - Intervention',
    mode='lines'))

t_dist_fig.add_trace(go.Scatter(
    x=[t_crit, t_crit],
    y=[0,0.4],
    name='Critical t value',
    mode='lines'))

t_dist_fig.update_layout(title='One-tailed alternative hypothesis',
                         xaxis=dict(title='t values'),
                         yaxis=dict(title='PDF'))

t_dist_fig.show()
```



The green line is the critical *t* value. The area under the curve to the left is 0.05 and not just 0.025. We need not split the area up into two symmetrical sides. We see that our test statistic is much less that the critical *t* value. The *p* value is calculate below.

```
stats.t.cdf(t_stat, 198)
```

```
0.00815709170533677
```

This is half of our original, two-tailed, *p* value. Our null hypothesis was that the mean of the placebo group was equal to or larger than the intervention group, but we found it to be less. We reject the null hypothesis and accept the alternative hypothesis.

We can also state the *opposite* one-tailed alternative hypothesis, (3).

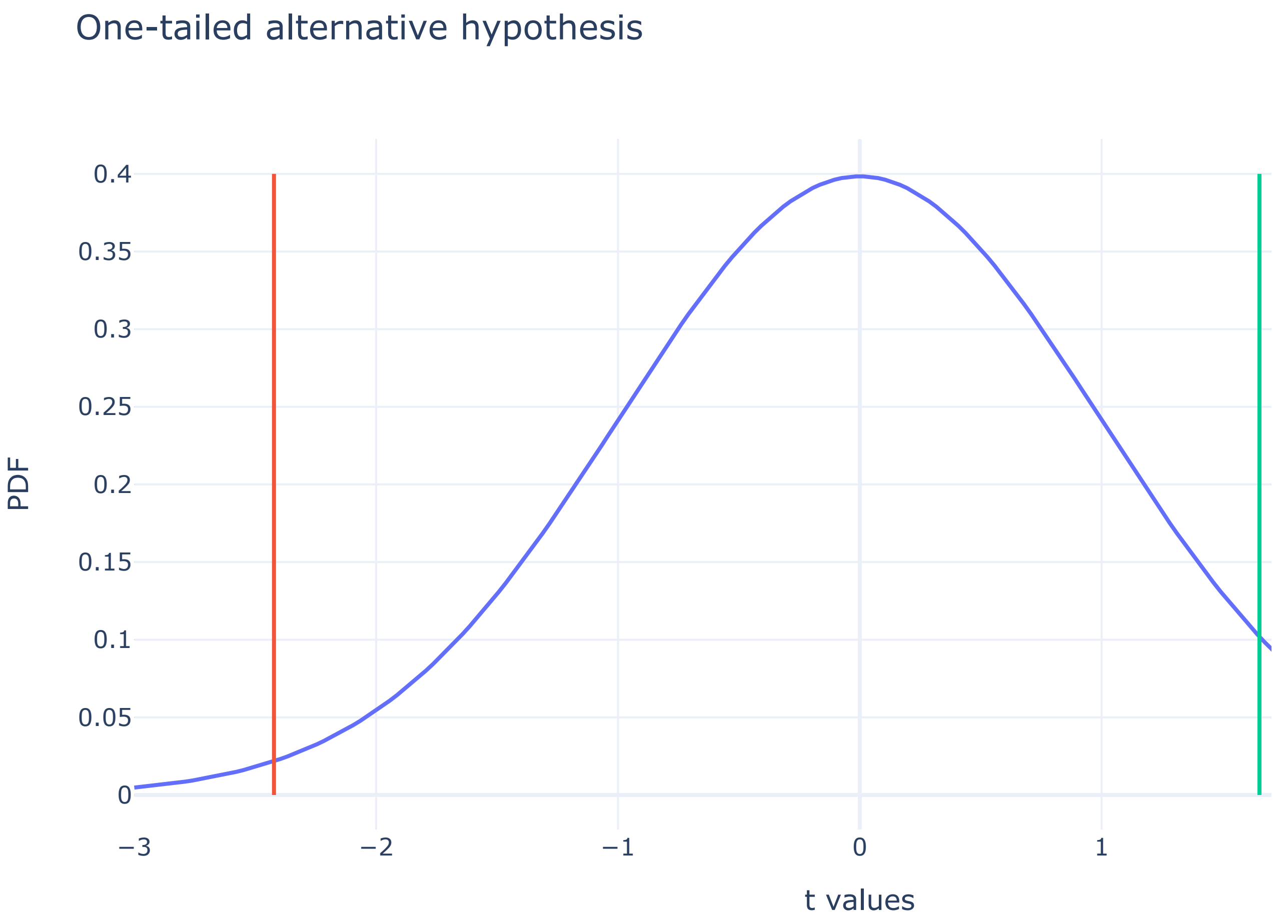
$$\begin{aligned} H_0 &: \bar{X}_1 \le \bar{X}_2 \\ H_\alpha &: \bar{X}_1 > \bar{X}_2 \end{aligned} \tag{3}$$

The critical critical *t* value is now calculated below, where we look at 0.05 of the area under the curve on the positive side.

```
t_crit = stats.t.ppf(0.95, 198)
t_crit
```

```
1.6525857836172075
```

```
t_dist_fig = go.Figure()

t_dist_fig.add_trace(go.Scatter(x=t_vals,
                                y=t_pdf_vals,
                                mode='lines',
                                mode='lines',
                                name='t distribution'))

t_dist_fig.add_trace(go.Scatter(
    x=[t_stat, t_stat],
    y=[0,0.4],
    name='Placebo - Intervention',
    mode='lines'))

t_dist_fig.add_trace(go.Scatter(
    x=[t_crit, t_crit],
    y=[0,0.4],
    name='Critical t value',
    mode='lines'))

t_dist_fig.update_layout(title='One-tailed alternative hypothesis',
                         xaxis=dict(title='t values'),
                         yaxis=dict(title='PDF'))

t_dist_fig.show()
```



Our rejection region is now to the right of the green line, but our difference is still the red line, very much outside the rejection area and we fail to reject the null hypothesis. Our *p* value is caluclated below, where we subtract the value (to the left of the red line) from the total area under the curve.

```
1 - stats.t.cdf(t_stat, 198)
```

```
0.9918429082946633
```

CONCLUSION

In this notebook we were introduced to hypothesis testing and some specific statistical tests to build an intuition of how hypothesis testing works. In understanding Data Science, we want to learn more about uncertainty, though. In the next notebook we review what we have learnt here, but start to introduce more concepts.

✓ 0s completed at 14:00 ● ×