index

January 14, 2022

1 Conducting T-Tests

1.1 Introduction

Previously you learned how to construct confidence intervals using the z-distribution (standard normal distribution), then learned how to construct them using the t-distribution instead. Using the t-distribution rather than the z-distribution for constructing confidence intervals allows you to handle instances where you don't know the population standard deviation or you have a small sample size. The same is true with hypothesis testing — a t-test (using the t-distribution) can be used instead of a z-test (using the z-distribution) under these conditions.

1.2 Objectives

You will be able to:

- Compare when you would use one sample vs. two sample t-tests
- Perform a one sample t-test and make conclusions about an experiment based on the results

1.3 Hypothesis Testing Using the t-Distribution

Recall that in frequentist hypothesis testing, you construct a test statistic from the measured data and use the value of that statistic to decide whether to reject the null hypothesis.

The test statistic is a lower-dimensional summary of the data but still maintains the discriminatory power necessary to determine whether a result is statistically significant for a given significance level.

1.4 t-test

t-tests (also called Student's t-test) are very practical hypothesis tests that can be employed to compare two averages (means) to assess if they are different from each other. You should run a t-test when you either: * Don't know the population standard deviation * You have a small sample size

Like a z-test, the t-test also tells you how significant the differences are i.e. it lets you know if those differences could have happened by chance. In this lesson, you will get an introduction to t-tests, in particular, the 1-sample t-test. There are additional kinds of t-tests including the 2-sample t-test and paired t-test. This lesson will show you the mathematical calculations behind a 1-sample t-test as well as how to perform a t-test in Python using NumPy and SciPy.

Detailed descriptions of hypothesis testing with t-tests can be found here and here

1.4.1 One Sample t-test

The 1-sample t-test is a statistical procedure used to determine **whether a sample of observations could have been generated by a process with a specific mean**. The one sample t-test compares the mean of your sample data to a known value. For example, you might want to know how your sample mean compares to a known population mean. Here is a quick example of a scenario where a 1-sample t-test could be applied.

Suppose you are interested in determining whether a bakery production line produces cakes with a weight of exactly 2 pounds. To test this hypothesis, you could collect a sample of cakes from the production line, measure their weights, and compare the sample with a value of 2 using a one-sample t-test.

1.4.2 Two Sample t-tests

The two-sample t-test is used to determine if two population means are equal. The main types of two-sampled t-tests are paired and independent tests. Paired tests are useful for determining how different a sample is affected by a certain treatment. In other words, the individual items/people in the sample will **remain the same** and researchers are comparing how they change after treatment. Here is an example of a scenario where a two-sample paired t-test could be applied:

The US Olympic weightlifting team is trying out a new workout technique to in an attempt to improve everyone's powerlifting abilities. Did the program have an effect at a 95% significance level?

Because we are looking at how specific individuals were affected by a treatment, we would use the paired t-test.

Independent two-sample t-tests are for when we are comparing two different, unrelated samples to one another. Unlike paired t-tests, we are not taking paired differences because there is no way to pair two unrelated samples! Here is an example of a scenario where a two-sample independent t-test could be applied:

Agricultural scientists are trying to compare the difference in soybean yields in two different counties in Mississippi.

You will learn more about the specifics of two sample t-tests in future lessons, but this lesson will focus on executing a one sample t-test.

1.4.3 Samples vs. Tails

Note that we now have two different labels where "one" and "two" appear repeatedly. Let's make sure the difference is clear!

Previously we learned about *one-tail* and *two-tail* tests. A one-tail test means that the alternative hypothesis contains something like $>$ or $<$, which means that we are only looking at the area under the curve on one side. Whereas a two-tail test means that the alternative hypothesis contains something like $\neq$ (which could mean greater than *or* less than), which means that we are looking at the area under the curve in two places, one on each side.

One-sample tests can be one-tail or two-tail tests, as can two-sample tests. Some quick examples:

- One-sample one-tail: $H_a : \mu < 3$

- One-sample two-tail: $H_a : \mu \neq 3$
- Two-sample one-tail: $H_a : \mu_1 < \mu_2$
- Two-sample two-tail: $H_a : \mu_1 \neq \mu_2$

1.4.4 Assumptions for Performing t-tests

When performing various kinds of t-tests, you assume that the sample observations have numeric and continuous values. You also assume that the sample observations are independent from each other (that is, that you have a simple random sample) and that the samples have been drawn from normal distributions. You can visually inspect the distribution of your sample using a histogram, for example.

In the case of unpaired two-sample t-tests, you also assume that the populations the samples have been drawn from have the same variance. For paired two-sample t-tests, you assume that the *difference* between the two sets of samples are normally distributed.

Regardless of the type of t-test you are performing, there are 5 main steps to executing them:

1) Set up null and alternative hypotheses

2) Choose a significance level

3) Calculate the test statistic (t-value)

4) Determine the critical t-value (find the rejection region)

5) Compare t-value with critical t-value to determine if we can reject the null hypothesis.

Now, you're going to go through these 5 steps in more detail to complete a t-test.

Let's begin with a sample experiment:

1.5 Sample Question:

Acme Ltd. wants to improve sales performance. Past sales data indicate that the average sale was 100 dollars per transaction. After training the sales force, recent sales data (from a random sample of 25 salesmen) is shown below.

```
[122.09, 100.64, 125.77, 120.32, 118.25,
  96.47, 111.4 ,  80.66, 110.77, 111.14,
  102.9, 114.54,  88.09,  98.59,  87.07,
 110.43, 101.9 , 123.89,  97.03, 116.23,
  108.3, 112.82, 119.57, 131.38, 128.39]
```

Did the training work?

1.5.1 Descriptive Statistics

Before completing the hypothesis test, let's calculate some summary statistics to see if the mean of the sample differed a substantial amount from the population. After, you can check to ensure that the data is relatively normal.

- **The population mean (μ):** Given as 100 (from past data).

- **The sample mean ($\bar{x}$):** Calculate from the sample data
- **The sample standard deviation (s):** Calculate from sample data
- **Number of observations (n):** 25 as given in the question. This can also be calculated from the sample data.
- **Degrees of Freedom (df):** Calculate from the sample as df = total no. of observations - 1

[17]:
```
## Import the packages
import numpy as np
from scipy import stats
import math

# For visualizing distributions - optional
import seaborn as sns
import matplotlib.pyplot as plt

# Sample data listed above
sample = np.array([122.09, 100.64, 125.77, 120.32, 118.25,  96.47, 111.4 , 80.
 ↪66,
       110.77, 111.14, 102.9 , 114.54,  88.09,  98.59,  87.07, 110.43,
       101.9 , 123.89,  97.03, 116.23, 108.3 , 112.82, 119.57, 131.38,
       128.39])
```

[18]:
```
# Population mean ( )
# (known from past data)
mu = 100
mu
```

[18]:
```
100
```

[19]:
```
# Sample mean (x̄) using NumPy array method mean()
x_bar = sample.mean()
x_bar
```

[19]:
```
109.5456
```

[25]:
```
# Sample standard deviation (sigma) using NumPy function std()
sigma = np.std(sample, ddof=1)
print("sigma: ", sigma)

### My Note
sigma_mine = stats.tstd(sample)
print("sigma_mine: ", sigma_mine)
```

```
sigma:  13.338774643871902
sigma_mine:  13.338774643871902
```

```
[26]: # Sample size (n)
      n = len(sample)
      n
```

```
[26]: 25
```

```
[27]: # Degrees of Freedom
      df = n-1
      df
```

```
[27]: 24
```

```
[28]: # Difference between sample mean and population mean
      diff = x_bar - mu
      diff
```

```
[28]: 9.545599999999993
```

Now, before we actually conduct a hypothesis test, we can summarize what we know about the data. This is a step that accidentally gets skipped sometimes, but it's actually really important!

It's useful to know what values you have found *before* trying to demonstrate that the values are statistically significant.

```
[29]: # Print the findings
      print(f"""
      The sample contains {n} observations, having a mean of {x_bar}
      and a standard deviation (sigma) of {round(sigma,3)}, with {df} degrees of␣
        ↪freedom.

      The difference between sample and population means is {round(diff, 2)}.
      """)

      # The sample contains 25 observations, having a mean of 109.5456
      # and a standard deviation (sigma) of 13.339, with 24 degrees of freedom.

      # The difference between sample and population means is 9.55.
```

```
The sample contains 25 observations, having a mean of 109.5456
and a standard deviation (sigma) of 13.339, with 24 degrees of freedom.

The difference between sample and population means is 9.55.
```

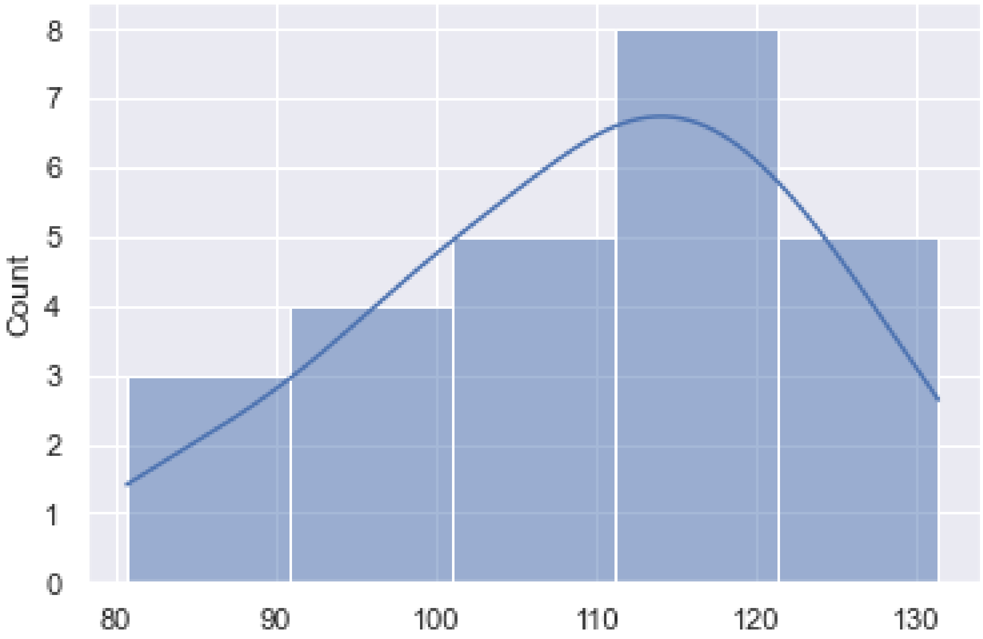
Ok, so we have found that the sample mean is about `$9.55` higher than the population mean. This indicates that, at least superficially, the training program appears to have increased sales by about `$9.55` per sale on average.

But, is that increase statistically significant?

If we want to investigate that question using a t-test, first we want to check that the sample is roughly normally distributed.

[32]:

```
sns.set(color_codes=True)
sns.histplot(sample, kde=True, bins=5);
```



Let's say that looks close enough. Now we'll actually set up and run the hypothesis test.

1.5.2 Step 1: Establish Null and Alternative Hypotheses

As you are trying to monitor a change in the sales performance after the training, the null hypothesis represents what should be your default assumption — that the training did not have the hypothesized impact.

H_0**:** *The null hypothesis is that the training did not increase sales, so:*

$$H_0 : \mu \leq 100$$

The alternative hypothesis is the one that you are testing. Our alternative hypothesis should address the expected change in the sales performance i.e. the sales performance has increased and the mean of sales post-training is greater than 100.

H_a**:** *The alternative hypothesis is that there is a positive change (i.e. the mean sales increased), so:*

$$H_a : \mu > 100$$

Why are these hypotheses written with μ rather than $\bar{x}$?

Hypotheses are always written in terms of population parameters like μ rather than sample statistics like $\bar{x}$, because our hypothesis is about a population, not a sample.

What we are really trying to understand here is whether these new sample data come from a *different population* than the one that produced the original μ of 100.

So, another way we could write the alternative hypothesis would be that $\mu > \mu_0$, where μ_0 is the original population mean (100) and μ is the population mean of those who had the new training.

We are using $\bar{x}$ in our calculations determining whether we can reject the null hypothesis, but $\bar{x}$ is not part of the hypotheses.

1.5.3 Step 2: Choose a Significance Level (Alpha)

Here we'll detour slightly to think some more about what it means to choose a given alpha value. The graphs below are from a different example (not the Acme Ltd. sales data) and are used for illustration purposes.

The significance level, also denoted as alpha or , is the probability of rejecting the null hypothesis when it is true. For example, a significance level of 0.05 indicates a 5% risk of concluding that a difference exists when there is no actual difference. Look at the following graphs for a better understanding:

In the graph above, the two shaded areas are equidistant from the null hypothesis value and each area has a probability of 0.025, for a total of 0.05. In statistics, you call these shaded areas the critical regions for a two-tailed test. If the population mean is 260, you'd expect to obtain a sample mean that falls in the critical region 5% of the time. The critical region defines how far away our sample statistic must be from the null hypothesis value before you can say it is unusual enough to reject the null hypothesis.

In the example shown here, the sample mean (330.6) falls within the critical region, which indicates it is statistically significant at the 0.05 level.

What if we had chosen a level of 0.01 instead?

The two shaded areas each have a probability of 0.005, the two of which add up to a total probability of 0.01. This time the sample mean does not fall within the critical region, and you fail to reject the null hypothesis. **This comparison shows why you need to choose your significance level before you begin your study.** It protects you from choosing a significance level because it conveniently gives you significant results!

In theory, a data scientist could determine that their results are statistically significant at the 0.05 level just by visually inspecting a graph like this. However it is more common to use statistical software to compute a p-value, then compare it to the stated significance level, rather than reading a graph.

For Acme's experiment, you can assume an α of 0.05.

1.5.4 Step 3: Calculate the t-statistic

Assuming that we are fulfilling the three requirements for a t-test mentioned above (i.e. normality, independence, and randomness), we are ready to calculate our t statistic using the formula for one-sample t-test given as:

2

$$t = \frac{\bar{x} - \mu}{\frac{s}{\sqrt{n}}}$$

(The *t-statistic* is also known as the *t-value*.)

Using the formula given above, calculate the t-statistic in Python:

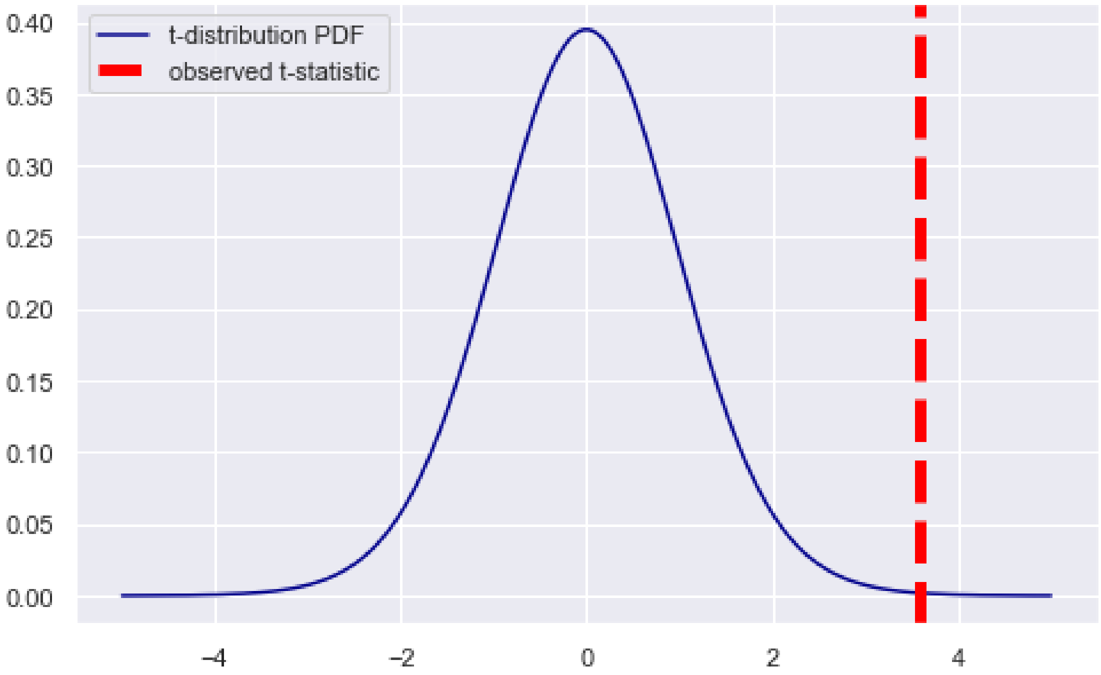
```
[33]: t = (x_bar -  mu)/(sigma/np.sqrt(n))
      t
      # 3.578139767278185
```

```
[33]: 3.578139767278185
```

The sample generated a t-statistic of around 3.58. Where in the t-distribution is this located? Let's try visualizing the calculated t-statistic on top of a PDF.

(Note that this is the probability density function for the t-distribution, not the distribution of sales data.)

```
[36]: # Set up figure and axes
      fig, ax = plt.subplots(figsize=(8,5))

      # Generate data for plotting the PDF
      # (1) generate points on the x axis between -5 and 5:
      x = np.linspace(-5, 5, 200)
      # (2) use stats.t.pdf to get values on the probability density
      # function for the t-distribution
      y = stats.t.pdf(x, df, 0, 1)

      # Plot the PDF as a line graph
      ax.plot(x, y, color='darkblue', label="t-distribution PDF")

      # Plot a vertical line for our measured difference in sales t-statistic
      ax.axvline(t, color='red', linestyle='--', lw=5,label='observed t-statistic')

      ax.legend();
```



2.0.1 Step 4: Calculate Critical Value (Find Rejection Region)

Note that a positive t-value indicates that the sample mean is greater than the population mean. This means that the sample's average sales performance post-training is greater than average population sales performance. (We already knew this from our descriptive analysis when we found that $\bar{x} - \mu$ was a positive value, and the t-statistic is just that divided by $\frac{s}{\sqrt{n}}$.)

This sounds like good news, **BUT** is the increase high enough to reject the null hypothesis, which says that there is no significant increase?

We'll answer this question by calculating a critical t-value. It's possible to calculate a critical t-value with a t-table or by using Python `scipy.stats` module.

The critical value approach involves determining "likely" or "unlikely", by determining whether or not the observed test statistic is more extreme than would be expected if the null hypothesis were true. This involves comparing the observed test statistic to some cutoff value, called the **"critical value"**. >If the test statistic is more extreme than the critical value, then the null hypothesis is rejected in favor of the alternative hypothesis. If the test statistic is not as extreme as the critical value, then the null hypothesis is not rejected.

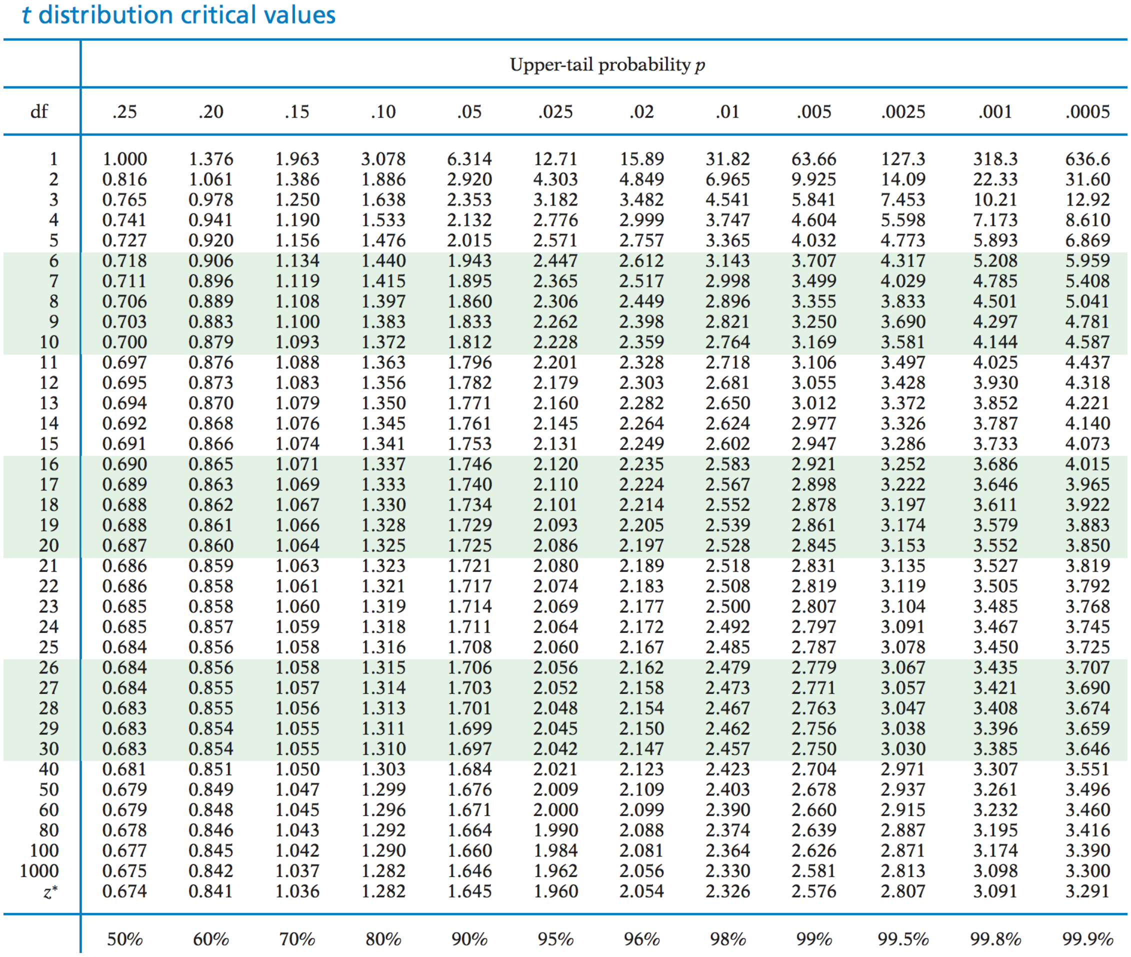
You need two values to find this:

The **alpha level**: given as 5% in the question.

You use this alpha level to determine what p-value to look up in this table. In the current example, we are executing a one-tailed t-test because we are only concerned with whether sales *increased*, not whether they were *different* in any way (which would be a two-tailed test). Therefore our p-value is just 0.05, from our 5% alpha.

Degrees of freedom, which is the number of items in the sample (n) minus 1: 25 – 1 = 24.

In the table below, look up the df of 24 and p of 0.05:



t distribution critical values

	Upper-tail probability *p*											
df	.25	.20	.15	.10	.05	.025	.02	.01	.005	.0025	.001	.0005
1	1.000	1.376	1.963	3.078	6.314	12.71	15.89	31.82	63.66	127.3	318.3	636.6
2	0.816	1.061	1.386	1.886	2.920	4.303	4.849	6.965	9.925	14.09	22.33	31.60
3	0.765	0.978	1.250	1.638	2.353	3.182	3.482	4.541	5.841	7.453	10.21	12.92
4	0.741	0.941	1.190	1.533	2.132	2.776	2.999	3.747	4.604	5.598	7.173	8.610
5	0.727	0.920	1.156	1.476	2.015	2.571	2.757	3.365	4.032	4.773	5.893	6.869
6	0.718	0.906	1.134	1.440	1.943	2.447	2.612	3.143	3.707	4.317	5.208	5.959
7	0.711	0.896	1.119	1.415	1.895	2.365	2.517	2.998	3.499	4.029	4.785	5.408
8	0.706	0.889	1.108	1.397	1.860	2.306	2.449	2.896	3.355	3.833	4.501	5.041
9	0.703	0.883	1.100	1.383	1.833	2.262	2.398	2.821	3.250	3.690	4.297	4.781
10	0.700	0.879	1.093	1.372	1.812	2.228	2.359	2.764	3.169	3.581	4.144	4.587
11	0.697	0.876	1.088	1.363	1.796	2.201	2.328	2.718	3.106	3.497	4.025	4.437
12	0.695	0.873	1.083	1.356	1.782	2.179	2.303	2.681	3.055	3.428	3.930	4.318
13	0.694	0.870	1.079	1.350	1.771	2.160	2.282	2.650	3.012	3.372	3.852	4.221
14	0.692	0.868	1.076	1.345	1.761	2.145	2.264	2.624	2.977	3.326	3.787	4.140
15	0.691	0.866	1.074	1.341	1.753	2.131	2.249	2.602	2.947	3.286	3.733	4.073
16	0.690	0.865	1.071	1.337	1.746	2.120	2.235	2.583	2.921	3.252	3.686	4.015
17	0.689	0.863	1.069	1.333	1.740	2.110	2.224	2.567	2.898	3.222	3.646	3.965
18	0.688	0.862	1.067	1.330	1.734	2.101	2.214	2.552	2.878	3.197	3.611	3.922
19	0.688	0.861	1.066	1.328	1.729	2.093	2.205	2.539	2.861	3.174	3.579	3.883
20	0.687	0.860	1.064	1.325	1.725	2.086	2.197	2.528	2.845	3.153	3.552	3.850
21	0.686	0.859	1.063	1.323	1.721	2.080	2.189	2.518	2.831	3.135	3.527	3.819
22	0.686	0.858	1.061	1.321	1.717	2.074	2.183	2.508	2.819	3.119	3.505	3.792
23	0.685	0.858	1.060	1.319	1.714	2.069	2.177	2.500	2.807	3.104	3.485	3.768
24	0.685	0.857	1.059	1.318	1.711	2.064	2.172	2.492	2.797	3.091	3.467	3.745
25	0.684	0.856	1.058	1.316	1.708	2.060	2.167	2.485	2.787	3.078	3.450	3.725
26	0.684	0.856	1.058	1.315	1.706	2.056	2.162	2.479	2.779	3.067	3.435	3.707
27	0.684	0.855	1.057	1.314	1.703	2.052	2.158	2.473	2.771	3.057	3.421	3.690
28	0.683	0.855	1.056	1.313	1.701	2.048	2.154	2.467	2.763	3.047	3.408	3.674
29	0.683	0.854	1.055	1.311	1.699	2.045	2.150	2.462	2.756	3.038	3.396	3.659
30	0.683	0.854	1.055	1.310	1.697	2.042	2.147	2.457	2.750	3.030	3.385	3.646
40	0.681	0.851	1.050	1.303	1.684	2.021	2.123	2.423	2.704	2.971	3.307	3.551
50	0.679	0.849	1.047	1.299	1.676	2.009	2.109	2.403	2.678	2.937	3.261	3.496
60	0.679	0.848	1.045	1.296	1.671	2.000	2.099	2.390	2.660	2.915	3.232	3.460
80	0.678	0.846	1.043	1.292	1.664	1.990	2.088	2.374	2.639	2.887	3.195	3.416
100	0.677	0.845	1.042	1.290	1.660	1.984	2.081	2.364	2.626	2.871	3.174	3.390
1000	0.675	0.842	1.037	1.282	1.646	1.962	2.056	2.330	2.581	2.813	3.098	3.300
z^*	0.674	0.841	1.036	1.282	1.645	1.960	2.054	2.326	2.576	2.807	3.091	3.291
	50%	60%	70%	80%	90%	95%	96%	98%	99%	99.5%	99.8%	99.9%

Hopefully you found the right value, `1.711`! This is our critical t-value.

You can double-check that this is the right answer using the CDF (cumulative distribution function) of the t-distribution from `scipy.stats`. The CDF gives the probability that the t-statistic will be less than or equal to a given value. In this case, we are plugging in 1.711, our critical t-value:

[37]:
```
stats.t.cdf(1.711, df=24)
```

[37]:
```
0.9500110459177917
```

That checks out! It says that there is a 95% probability that the t-statistic will be less than or equal to 1.711, which is 1 minus our chosen alpha.

More often as a data scientist you will use code to find the critical t-value, rather than looking it up in a table. Conveniently, there is another function from `scipy.stats` that is just the inverse of the CDF shown above, called a PPF. This stands for "percent point function". Given a probability, we use the PPF to compute the corresponding value — in this case, the critical t-value!

(Note that we need to pass in 1 minus alpha, not alpha, because we are finding the t-value where there is a 95% probability we are "right" and 5% probability we are "wrong", not a 5% probability we are "right".)

```
[38]: t_crit = stats.t.ppf(1 - 0.05, df=24)
      t_crit
```

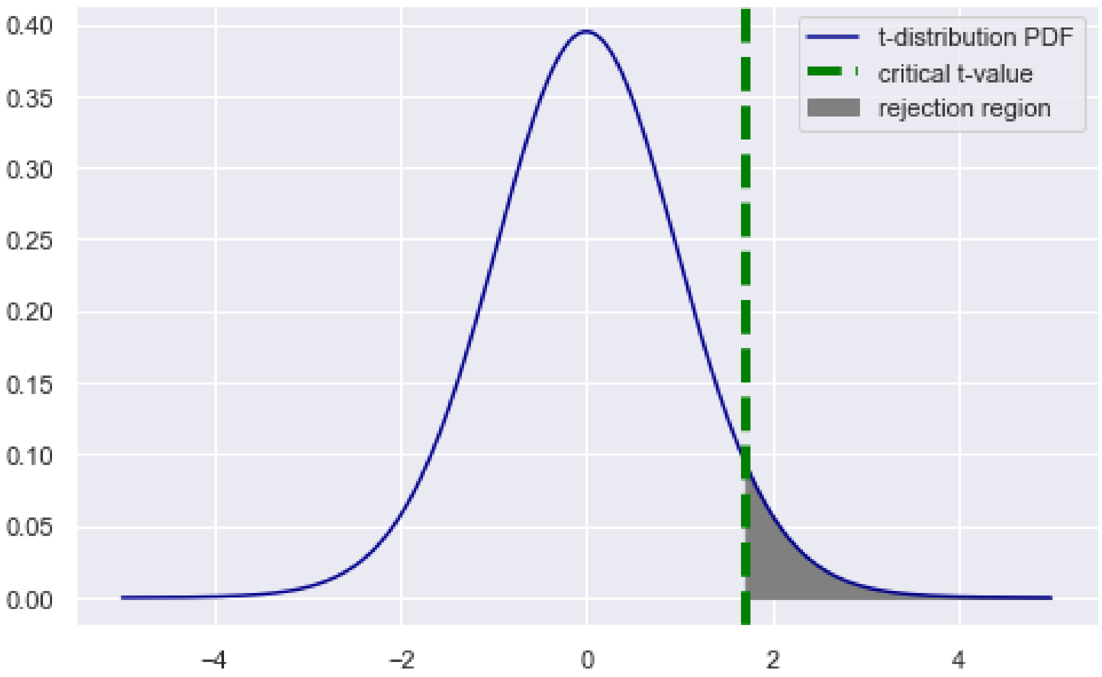
```
[38]: 1.7108820799094275
```

This is the same critical t-value we found in the table! With more digits, so it looks a little bit different.

The critical t-value marks the boundary of the **rejection region**. Because this is a one-tailed test, we can simply say that the rejection region is anything greater than the critical t-statistic. If our sample t-statistic falls into the rejection region, we can reject the null hypothesis!

We visualize this below:

```
[39]: # Set up figure and axes
      fig, ax = plt.subplots(figsize=(8,5))

      # Plot the PDF as a line graph
      # (x and y were created in previous plotting code)
      ax.plot(x, y, color='darkblue', label="t-distribution PDF")

      # Plot a vertical line for our critical t-value
      ax.axvline(t_crit,color='green',linestyle='--',lw=4,label='critical t-value')

      # Fill area under line graph where x value is creater than critical t
      ax.fill_betweenx(y,x,t_crit,where=x > t_crit,color="gray",label="rejection␣
        ↪region")

      ax.legend();
```

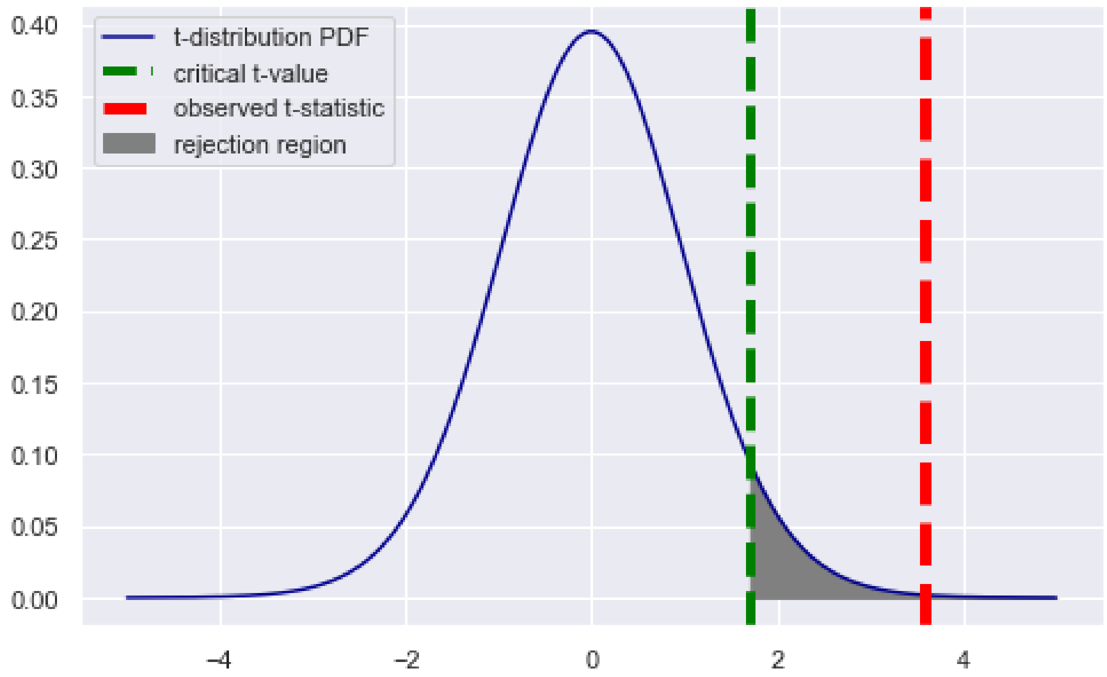


2.0.2 Step 5: Compare Sample t-value with Critical t-value: Can We Reject the Null Hypothesis?

We can begin by combining the previous two graphs, to show our observed t-value alongside the critical t-value and rejection region:

[41]:

```
# Set up figure and axes
fig, ax = plt.subplots(figsize=(8,5))

# Plot the PDF as a line graph
# (x and y were created in previous plotting code)
ax.plot(x, y, color='darkblue', label="t-distribution PDF")

# Plot a vertical line for our critical t-value
ax.axvline(t_crit,color='green',linestyle='--',lw=4,label='critical t-value')

# Fill area under line graph where x value is creater than critical t
ax.fill_betweenx(y,x,t_crit,where=x > t_crit,color="gray",label="rejection␣
 ↪region")

# Plot a vertical line for our measured difference in sales t-statistic
ax.axvline(t, color='red', linestyle='--', lw=5,label='observed t-statistic')

ax.legend();
```



Is our observed t-statistic in the rejection region?

Yes! It's hard to even see the rejection region shading at this extreme part of the distribution, but we are in it. That means we have completed the hypothesis test, and now we only need to interpret the result.

To break down the points:

- Our observed t-statistic is greater than the critical t-value,
- which means that it is in the rejection region,
- which means that we can reject the null hypothesis at a significance level of 0.05,
- which means that **there was a statistically significant increase in sales performance for those who received the training** (at our specified significance level)

Although it isn't strictly necessary because we already have our answer, we could also calculate a p-value for this test. Recall that the p-value is the probability that we would observe at least this extreme of a t-statistic given that the null hypothesis is actually correct.

One way we could do this would be finding the highest value in the df = 24 row of the lookup table above that is still below our t-statistic, then reporting that column. This tends to be fairly imprecise. It appears that our p-value is somewhere between 0.001 and 0.0005, since our t-statistic is between 3.467 and 3.745. If the details aren't important, you could just report this as $p < 0.001$.

Alternatively, there is functionality within `scipy.stats` to find a more precise p-value using the "survival function" SF:

[42]:
```
p_value = stats.t.sf(t, df=24)
p_value
```

```
[42]: 0.0007589472707557043
```

So, our p-value is about 0.00076, meaning that if the null hypothesis is correct (that there was no increase in sales performance with the training), we would expect it to produce results like we observed 0.076% of the time — very improbable!

Given our alpha of 0.05, this p-value is smaller than our alpha, therefore we can again say that we reject the null hypothesis at a significance level of 0.05.

2.1 A Simpler Process with `ttest_1samp`

Going through step by step and creating those visualizations is useful for teaching and learning the steps of conducting a t-test, but there are tools to make the process a lot faster once you understand how it works. Going forward, you'll usually want to use these tools rather than the process shown above.

Specifically, there are t-test functions built into `scipy.stats`!

To repeat the above test, we are again conducting a one-sample, one-sided t-test, so we will want to use `ttest_1samp`.

We are still using these hypotheses:

$ H_{0}:   100 $

$ H_{a}: > 100 $

and $\alpha = 0.05$.

First, we pass the sample data and the mean we are using for comparison into the function. Note that we are **not** passing in the alpha value, so there will be one additional step to determine whether we reject the null hypothesis after this step.

[43]:
```
results = stats.ttest_1samp(
    a=sample,    # the entire array-like sample
    popmean=100 # the mean you are testing the sample against
)

results
```

```
[43]: Ttest_1sampResult(statistic=3.578139767278185, pvalue=0.0015178945415114085)
```

This `results` variable contains both the t-statistic and the p-value. Below we compare the t-statistic from this calculation to the one we calculated earlier:

[45]:
```
print(t)
print(results.statistic)
print(results.pvalue)
```

```
3.578139767278185
3.578139767278185
0.0015178945415114085
```

We can either use the t-statistic and rejection region approach next, or we can just use the p-value!

2.1.1 Rejection Region Approach with `ttest_1samp`

One approach would be to compute the critical t-value, then compare the t-value from our `ttest_1samp` results to that critical t-value.

[46]:

```python
alpha = 0.05

t_crit = stats.t.ppf(1 - alpha, df=len(sample)-1)

if results.statistic > t_crit:
    print("Null hypothesis rejected.")
    print("The increase in sales performance is statistically significant at␣
 ↪the", alpha, "level.")
else:
    print("Failed to reject the null hypothesis.")
    print("The increase in sales performance is not statistically significant␣
 ↪at the", alpha, "level.")
```

```
Null hypothesis rejected.
The increase in sales performance is statistically significant at the 0.05
level.
```

2.1.2 p-value Approach with `ttest_1samp`

The code for this is simpler (because you don't have to calculate a critical t), although conceptually it is a bit more complicated.

The `ttest_1samp` function assumes that you are performing a ***two-sided*** one-sample t-test. This means that when you specify a mean of 100, it assumes your null hypothesis is $\mu = 100$ and your alternative hypothesis is $\mu \neq 100$. This doesn't affect the t-statistic, but it does affect the p-value.

Let's look at that more closely:

[47]:

```python
print(p_value)
print(results.pvalue)
```

```
0.0007589472707557043
0.0015178945415114085
```

The p-value from `results` looks like it's twice as large as the one we calculated! In fact, it's exactly twice as large:

[48]:

```python
print(p_value)
print(results.pvalue / 2)
```

```
0.0007589472707557043
0.0007589472707557043
```

In this case with very small p-values it doesn't affect our answer either way — both 0.0015 and 0.00076 are smaller than 0.05.

But in other cases this difference in p-values could mean the difference between rejecting and failing to reject the null hypothesis!

So, what should we do?

Simply **divide the p-value result from `ttest_1samp` by 2 if you are using it for a one-sided t-test**. If you are using it for a two-sided t-test, no further action is required.

In conclusion, we can use the p-value from `ttest_1samp` to answer our research question like this:

[49]:

```
if (results.pvalue / 2) < alpha:
    print("Null hypothesis rejected.")
    print("The increase in sales performance is statistically significant at␣
 ↪the",
          alpha, "level.")
else:
    print("Failed to reject the null hypothesis.")
    print("The increase in sales performance is not statistically significant␣
 ↪at the",
          alpha, "level.")
```

```
Null hypothesis rejected.
The increase in sales performance is statistically significant at the 0.05
level.
```

That was easy! This p-value approach meant that we never had to calculate the critical t, we just handed the work over to SciPy. It was conceptually a bit more challenging but the code was shorter.

Unless you are specifically instructed to use one approach or the other (rejection region or p-value), either one works well.

Also, note that you might also see a couple different techniques for extracting information from the t-test results object. It is designed so you can use dot notation or indexing notation to retrieve the same values:

[59]:

```
print("results.statistic:", results.statistic)
print("results[0]:        ", results[0])
print("results.pvalue:    ", results.pvalue)
print("results[1]:        ", results[1])
```

```
results.statistic: 3.578139767278185
results[0]:        3.578139767278185
results.pvalue:    0.0015178945415114085
results[1]:        0.0015178945415114085
```

2.2 Summary

In this lesson, you saw a quick introduction to hypothesis testing using frequentists methods with t-values and p-values. You saw how a one sample t-test can be applied to contexts where the population mean is unknown and you have a limited amount of sample data. You looked at all the stages required for such hypothesis testing with a description of steps and also, how to perform these steps in Python.