# students_t_test

July 23, 2019

[1]:
```
# src https://machinelearningmastery.com/
↪how-to-code-the-students-t-test-from-scratch-in-python/

# The Students t-Test is a statistical hypothesis test for testing whether two␣
↪samples are expected to have been drawn from the same population.

# The test works by checking the means from two samples to see if they are␣
↪significantly different from each other.
# It does this by calculating the standard error in the difference between␣
↪means, which can be interpreted to see how likely the difference is, if the␣
↪two samples have the same mean (the null hypothesis).

# The t statistic calculated by the test can be interpreted by comparing it to␣
↪critical values from the t-distribution. The critical value can be␣
↪calculated using the degrees of freedom and a significance level with the␣
↪percent point function (PPF).

# We can interpret the statistic value in a two-tailed test, meaning that if we␣
↪reject the null hypothesis, it could be because the first mean is smaller or␣
↪greater than the second mean
# To do this, we can calculate the absolute value of the test statistic and␣
↪compare it to the positive (right tailed) critical value, as follows:
    # If abs(t-statistic) <= critical value: Accept null hypothesis that the␣
↪means are equal.
    # If abs(t-statistic) > critical value: Reject the null hypothesis that the␣
↪means are equal.

# We can also retrieve the cumulative probability of observing the absolute␣
↪value of the t-statistic using the cumulative distribution function (CDF) of␣
↪the t-distribution in order to calculate a p-value. The p-value can then be␣
↪compared to a chosen significance level (alpha) such as 0.05 to determine if␣
↪the null hypothesis can be rejected:
    # If p > alpha: Accept null hypothesis that the means are equal.
    # If p <= alpha: Reject null hypothesis that the means are equal.
```

```
# In working with the means of the samples, the test assumes that both samples
↪were drawn from a Gaussian distribution. The test also assumes that the
↪samples have the same variance, and the same size, although there are
↪corrections to the test if these assumptions do not hold (Welch's t-test).

# There are two main versions of Students t-test:
    # Independent Samples. The case where the two samples are unrelated.
    # Dependent Samples. The case where the samples are related, such as
↪repeated measures on the same population. Also called a paired test.

# ttest_ind() - independent samples
# ttest_rel() - dependent samples

# 2 independent samples
# t = observed difference between sample means / standard error of the
↪difference between the means
# t = (mean(X1) - mean(X2)) / sed
    # where x1 and x2 are the 1st and 2nd data samples
    # where sed is the standard error of the difference between the means
        # sed = sqrt(se1^2 + se2^2)
            # wher se1 and se2 are standard errors for the first and second
↪datasets
                # se = std / sqrt(n)
                    # std - standard deviation
                    # n - number of observations in the sample
# These calculations make the following assumptions:
    # The samples are drawn from a Gaussian distribution.
    # The size of each sample is approximately equal.
    # The samples have the same variance.

# 2 dependent samples
# t = (mean(X1) - mean(X2)) / sed
    # sed = sd / sqrt(se1^2 + se2^2)
        # sd = sqrt((d1 - (d2**2 / n)) / (n - 1))
            # d1 = sum (X1[i] - X2[i])^2 for i in n
            # d2 = sum (X1[i] - X2[i]) for i in n
```

[2]:
```
# src: https://towardsdatascience.com/
↪inferential-statistics-series-t-test-using-numpy-2718f8f9bf2f

# The t test (also called Students T Test) compares two averages (means) and
# tells you if they are different from each other.
# The t test also tells you how significant the differences are;
# In other words it lets you know if those differences could have happened by
↪chance.

# A very simple example:
```

```
# Lets say you have a cold and you try a naturalistic remedy.
# Your cold lasts a couple of days.
# The next time you have a cold, you buy an over-the-counter pharmaceutical
# and the cold lasts a week.
# You survey your friends and they all tell you that their colds were of a
# shorter duration (an average of 3 days) when they took the
# homeopathic remedy.
# What you really want to know is, are these results repeatable?
# A t test can tell you by comparing the means of the two groups and
# letting you know the probability of those results happening by chance.

# Another example:
# Students T-tests can be used in real life to compare means.
# For example, a drug company may want to test a new cancer drug to find
# out if it improves life expectancy.
# In an experiment, theres always a control group (a group who are given
# a placebo, or sugar pill).
# The control group may show an average life expectancy of +5 years,
# while the group taking the new drug might have a life expectancy of +6 years.
# It would seem that the drug might work.
# But it could be due to a fluke.
# To test this, researchers would use a Students t-test to find out if
# the results are repeatable for an entire population.

# What is t-score?
# The t score is a ratio between the difference between two groups and
# the difference within the groups.
# The larger the t score, the more difference there is between groups.
# The smaller the t score, the more similarity there is between groups.

# A t score of 3 means that the groups are three times as different from each␣
↪other
# as they are within each other.

# When you run a t test, the bigger the t-value, the more likely it is that
# the results are repeatable.

# A large t-score tells you that the groups are different.
# A small t-score tells you that the groups are similar.

# Every t-value has a p-value to go with it.
# A p-value is the probability that the results from your sample data
# occurred by chance.
```

[3]:
```
# Load libraries

import numpy as np
```

```python
from scipy.stats import ttest_ind, ttest_rel, sem, t
```

[4]:
```python
# Generate Gaussian random samples datasets

np.random.seed(1)

data1 = 5 * np.random.randn(100) + 50
data2 = 5 * np.random.randn(100) + 51
```

[5]:
```python
stat, p = ttest_ind(data1, data2)

display(
    "Expected values for the test on these data",
    "t = {0}".format(stat),
    "p = {0}".format(p)
)
```

```
'Expected values for the test on these data'
```

```
't = -2.2620139704259556'
```

```
'p = 0.024782819014639627'
```

[6]:
```python
def independent_ttest(data1, data2, alpha):
    """Function to calculate t-test for 2 independent samples"""
    mean1, mean2 = np.mean(data1), np.mean(data2)
    se1, se2 = sem(data1), sem(data2)
    sed = np.sqrt(se1**2.0 + se2**2.0)
    t_stat = (mean1 - mean2) / sed
    ddof = len(data1) + len(data2) - 2
    critical_value = t.ppf(1.0 - alpha, ddof)
    p = (1.0 - t.cdf(abs(t_stat), ddof)) * 2.0
    return t_stat, ddof, critical_value, p
```

[7]:
```python
alpha = 0.05
t_stat, df, cv, p = independent_ttest(data1, data2, alpha)

display(
    "t = {0}".format(t_stat),
    "p = {0}".format(p)
)

display(df, cv)
```

```
't = -2.2620139704259556'
```

'p = 0.024782819014639745'

198

1.6525857836172075

[8]:
```python
# Results of t-test

# interpret via critical value
if abs(t_stat) <= cv:
    print('Accept null hypothesis that the means are equal.')
else:
    print('Reject the null hypothesis that the means are equal.')

# interpret via p-value
if p > alpha:
    print('Accept null hypothesis that the means are equal.')
else:
    print('Reject the null hypothesis that the means are equal.')
```

```
Reject the null hypothesis that the means are equal.
Reject the null hypothesis that the means are equal.
```

[9]:
```python
stat, p = ttest_rel(data1, data2)

display(
    "Expected values for the test on these data",
    "t = {0}".format(stat),
    "p = {0}".format(p)
)
```

'Expected values for the test on these data'

't = -2.3719009567078646'

'p = 0.019630798337126193'