One sample T-tests

In this vignette, we work through a one-sample T-test. We'll use the same data we used in `vignette("one-sample-z-test")`, so you compare the result of the tests.

Problem setup

Let's suppose that a student is interesting in estimating how many memes their professors know and love. So they go to class, and every time a professor uses a new meme, they write it down. After a year of classes, the student has recorded the following meme counts, where each count corresponds to a single class they took:

$$3, 7, 11, 0, 7, 0, 4, 5, 6, 2$$

Note: For a Z-test, we need to know the population standard deviation σ. With T-tests, this is unnecessary, and we estimate the standard deviation from the data. This results in some additional uncertainty in our test statistic, which is reflected in the heavier tails of the T distribution compared to the normal distribution.

Assumption checking

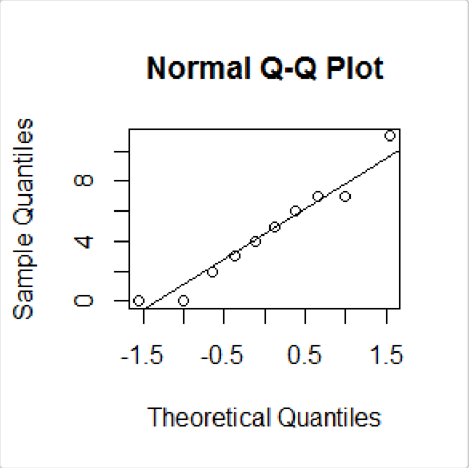
Before we can do a T-test, we need to make check if we can reasonably treat the mean of this sample as normally distributed. This happens is the case of either of following hold:

1. The data comes from a normal distribution.
2. We have lots of data. How much? Many textbooks use 30 data points as a rule of thumb.

Since we have a small sample, we let's check if the data comes from a normal distribution using a normal quantile-quantile plot.

```r
# read in the data
x <- c(3, 7, 11, 0, 7, 0, 4, 5, 6, 2)

# make the qqplot
qqnorm(x)
qqline(x)
```



Since the data lies close the line $y = x$, and has no notable systematic deviations from line, it's safe to treat the sample as coming from a normal distribution. We can proceed with our hypothesis test.

Null hypothesis and test statistic

Let's test the null hypothesis that, on average, professors know 3 memes. That is

$$H_0 : \mu = 3 \qquad H_A : \mu \neq 3$$

First we need to calculate our T-statistic. Let's use do this with R. Remember that the T-statistic is defined as

$$T = \frac{\bar{x} - \mu_0}{s / \sqrt{n}} \sim t_{n-1}$$

where $\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$ is the sample mean, μ_0 is our proposed value for the population mean, $s = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})^2}$ is the sample standard deviation, and n is the sample size. This test statistic then has a T distribution with $n - 1$ degrees of freedom.

Calculating p-values

In R this looks like:

```
n <- length(x)

# calculate the z-statistic
t_stat <- (mean(x) - 3) / (sd(x) / sqrt(n))
t_stat
#> [1] 1.378916
```

Since the sample size is 10, to calculate a two-sided p-value, we need to find

$$\begin{aligned} P(|t_9| \geq |1.38|) &= P(t_9 \geq 1.38) + P(t_9 \leq -1.38) \\ &= 1 - P(t_9 \leq 1.38) + P(t_9 \leq -1.38) \end{aligned}$$

To do this we need to c.d.f. of a t_9 distribution.

```
library(distributions3)

T_9 <- StudentsT(df = 9)  # make a T_9 distribution
1 - cdf(T_9, 1.38) + cdf(T_9, -1.38)
#> [1] 0.2008985
```

Note that we saved `t_stat` above so we could have also done

```
1 - cdf(T_9, abs(t_stat)) + cdf(T_9, -abs(t_stat))
#> [1] 0.2012211
```

which is slightly more accurate since there is no rounding error.

So our p-value is about 0.20. You should verify this with a T-table. Note that you should get the *same* value from `cdf(T_9, 1.38)` and looking up `1.38` on a T-table.

You may also have seen a different formula for the p-value of a two-sided T-test, which makes use of the fact that the T distribution is symmetric:

$$P(|t_9| \geq |1.38|) = 2 \cdot P(t_9 \leq -|1.38|)$$

Using this formula we get the same result:

```
2 * cdf(T_9, -1.38)
#> [1] 0.2008985
```

Finally, sometimes we are interest in one sided T-tests. For the test

$$H_0 : \mu \leq 3 \qquad H_A : \mu > 3$$

the p-value is given by

$$P(t_9 > 1.38)$$

which we calculate with

```
1 - cdf(T_9, 1.38)
#> [1] 0.1004493
```

For the test

$$H_0 : \mu \geq 3 \qquad H_A : \mu < 3$$

the p-value is given by

$$P(t_9 < 1.38)$$

which we calculate with

```
cdf(T_9, 1.38)
#> [1] 0.8995507
```

Using the t.test() function

If you want to verify that your calculation is correct, R has a function `t.test()` that performs T-tests and calculates T confidence intervals for means.

To get a T statistic, degrees of freedom of the sampling distribution, and the p-value we pass `t.test()` a vector of data. We tell `t.test()` our null hypothesis by passing the `mu` argument. In our case, we want to test `mu = 3`.

```
t.test(x, mu = 3)
#> 
#>  One Sample t-test
#> 
#> data:  x
#> t = 1.3789, df = 9, p-value = 0.2012
#> alternative hypothesis: true mean is not equal to 3
#> 95 percent confidence interval:
#>  2.0392 6.9608
#> sample estimates:
#> mean of x 
#>       4.5
```

If we don't specify `mu`, `t.test()` assumes that we want to test the hypothesis $H_0 : \mu = 0$. This looks like

```
t.test(x)
#> 
#>  One Sample t-test
#> 
#> data:  x
#> t = 4.1367, df = 9, p-value = 0.002534
#> alternative hypothesis: true mean is not equal to 0
#> 95 percent confidence interval:
#>  2.0392 6.9608
#> sample estimates:
#> mean of x 
#>       4.5
```

We can also get one-sided p-values from `t.test()` by specifying the alternative hypothesis. For the test

$$H_0 : \mu \leq 3 \qquad H_A : \mu > 3$$

we would use

```
t.test(x, mu = 3, alternative = "greater")
#> 
#>  One Sample t-test
#> 
#> data:  x
#> t = 1.3789, df = 9, p-value = 0.1006
#> alternative hypothesis: true mean is greater than 3
#> 95 percent confidence interval:
#>  2.505919      Inf
#> sample estimates:
#> mean of x 
#>       4.5
```

For the test

$$H_0 : \mu \geq 3 \qquad H_A : \mu < 3$$

we would use

```
t.test(x, mu = 3, alternative = "less")
#> 
#>  One Sample t-test
#> 
#> data:  x
#> t = 1.3789, df = 9, p-value = 0.8994
#> alternative hypothesis: true mean is less than 3
#> 95 percent confidence interval:
#>      -Inf 6.494081
#> sample estimates:
#> mean of x 
#>       4.5
```

Both of these results agree with our hand calculations from earlier.