Module 5 Lab

In this lab you will learn to use the R functions `t.test()`, `prop.test()`, and `binom.test()`, and understand their output. In the final section we will review non-technical summaries a bit more.

Student's t-Test

The function `t.test()` performs a one or two sample t-test. Here we will use it for a one sample test, and learn how to get other useful information from the function. Remember, for more information type `?t.test`.

First we will generate some data to work with. To make sure we have the same data, the first line of code below will set the "seed" for our random number generator. This is so that we have the same "random" numbers, and thus get the same results (use `?set.seed()` for details). I chose 1908 as our seed because that was the year William Gosset invented the t-test!

```
set.seed(1908) # William Gosset invents t-test
x <- rnorm(40, 5, 5) # Generate 40 N(5,5) random numbers
x # show those numbers
```

```
##  [1] 11.1976498  7.7199763  2.5668829 -0.4704134  7.3408905 -1.4265065
##  [7] 13.6187501  2.8265528 10.2994741  2.1248107  6.6034109 12.2413295
## [13]  0.5571060 14.8226579 15.5570144  4.3400826  8.2460864  4.1885037
## [19]  4.6949157 11.2558256  0.9389690  6.8483868  7.9967965  2.5325993
## [25]  1.1412952  5.0185188  6.2181406 -6.1710370  5.2649135  0.3246880
## [31]  5.0343295 11.5819554  9.1260516  4.8707156  0.2161102 -0.3875790
## [37]  4.1853589  2.6693378 13.1452207  3.9052319
```

The first line of code sets the seed for random number generation, and the second line generates 40 Normal random numbers with a mean of five, and standard deviation of five. Did you get the same "random" sample?

(You may think it strange we are doing a t-test on simulated data. It is! But, it will set us up well to explore the performance of the test in situations where we know the truth for future modules.)

Before we perform a one sample t-test, we have to establish what we are testing. How about we see whether `t.test()` can tell us about the evidence in the sample that the population mean is not equal to three (we know it isn't because we generated the data, but in a real data example you woulnd't). Then our null and alternative hypotheses, in statistical notation, are:

$$H_0 : \mu = 3$$

$$H_A : \mu \neq 3$$

To perform the test on our dataset, run the following line of code.

```r
t.test(x, mu = 3, conf.level = 0.95)
```

The first argument, `x`, is our data. The second argument, `mu = 3`, declares the null hypothesized value for μ; in this case: $H_0 : \mu = 3$. The function `t.test()` also returns a t-based confidence interval for the mean. Therefore, we use the third argument, `conf.level = 0.95`, to specify that we want a 95% confidence interval for the mean.

```
##
##  One Sample t-test
##
## data:  x
## t = 3.295, df = 39, p-value = 0.002101
## alternative hypothesis: true mean is not equal to 3
## 95 percent confidence interval:
##  3.992041 7.146209
## sample estimates:
## mean of x
##  5.569125
```

The first line of output reminds you of the data used, in case you have forgotten! The second line gives the value of the test statistic, the degrees of freedom, and the p-value. How would the p-value change if the test were one-sided? The next line states the alternative hypothesis in plain english. The next two lines give the two sided 95% confidence interval. Finally, `t.test` returns a point estimate for μ. We might conclude:

> There is convincing evidence the population mean (insert response variable here) is not equal to 3. We estimate the mean (insert response variable here) is 5.57, and with 95% confidence, the mean is between 3.99 and 7.14.

(If this was real data, you'd know what `x` measured and insert the approriate variable, e.g. "... population mean *income of US adults* is ...")

A Test of Proportions - Normal Approximation to Binomial

Now we will perform a test about a proportion. Suppose a gambler purchases a pair of dice made for cheating. Each die is designed to roll a six one out of every four rolls (instead of one out of six). He rolls one die 100 times to test it, and gets 20 sixes. We want to know, is the die working as it is designed to? We test the null hypothesis that the true proportion of rolls that result in six is 0.25, and the alternative that the rate is different from 0.25.

$$H_0 : p = 0.25$$

$$H_A : p \neq 0.25$$

Since we have a sample size of 100 rolls, the Central Limit Theorem assures us that a test relying on a Normal approximation to the Binomial will perform well. The function we will use to perform the test is `prop.test()`.

```r
prop.test(x = 20, n = 100, p = 0.25, conf.level = 0.95, correct = FALSE)
```

```
##
##  1-sample proportions test without continuity correction
##
## data:  20 out of 100, null probability 0.25
## X-squared = 1.3333, df = 1, p-value = 0.2482
## alternative hypothesis: true p is not equal to 0.25
## 95 percent confidence interval:
##  0.1333669 0.2888292
## sample estimates:
##   p
## 0.2
```

The line of code and the output follow a similar format to `t.test()` in the first section. The `x = 20` and `n = 100` specify the number of "successes" and "trials", respectively. The `p = 0.25` argument declares the null hypothesized proportion. `conf.level` has the same meaning as in `t.test()` and `correct = FALSE` specifies not to do a continuity correction.

The first line of output gives the test conducted, and whether a continuity correction was made. The second line gives the data supplied to the function: "successes," "trials," and the null hypothesized proportion. The third line gives the value of the test statistic ($\chi^2 = 1.08$), the degrees of freedom, and the p-value. Note that the test statistic, $\chi^2 = 1.08$, is the square of the z-statistic. The next line reiterates the alternative hypothesis specified. Following that is a 95% confidence interval, and finally a point estimate for the unknown proportion.

See Module 5, Lecture 2, or run `?prop.test`, for more information about the function.

A Test of Proportions - Exact Binomial

The gambler is a little suspicious, so decides to test the other die. He is running out of patience though, so decides to roll the second die only 20 times, and rolls 3 sixes. Can we reject the null hypothesis that the true proportion of sixes for this die is 0.25, in favor of the alternative that the proportion not 0.25?

$$H_0 : p = 0.25$$
$$H_A : p \neq 0.25$$

This time we have may fewer rolls (Binomial trials) to base our decision on, so an exact Binomial test is better than the Normal approximation. This is because the Central Limit Theorem does not guarantee the sample mean will be approximately Normal with such a small sample size. Notice this time I stored the output in the variable "b"; storing test results in this way can sometimes be useful.

```r
b <- binom.test(3, 20, p = 0.25, conf.level = 0.95)
b
```

```
##
##  Exact binomial test
##
## data:  3 and 20
## number of successes = 3, number of trials = 20, p-value = 0.4394
## alternative hypothesis: true probability of success is not equal to 0.25
## 95 percent confidence interval:
##  0.03207094 0.37892683
## sample estimates:
## probability of success
##                   0.15
```

The output tells a story. The gambler rolled three sixes out of 20 rolls, but this is not unlikely if the cheat die is performing as it should. In fact, the probability a player will have a result this extreme, or more extreme, in 20 rolls is 0.4394. We estimate the true proportion of sixes is 0.15, but we are 95% confident that the proportion of sixes is between 0.032 and 0.344, which contains the die's target proportion. Based on the p-value and the confidence interval, we would fail to reject the null hypothesis that the die is performing as it is designed to.

Summary Examples

Here is an example non-technical summary of the exact Binomial test of proportion.

- The die came up six 3 times in 20 games. There is no evidence the true proportion of sixes is not 0.25 (exact Binomial test, p-value = 0.4394). We estimate the true proportion of sixes is 0.15, and with 95% confidence the true proportion is between 0.032 and 0.379.

As mentioned in Module 4 lab, there are many ways to incorrectly write a non-technical summary. Below are a few examples of **incorrect** or **incomplete** (or too technical) summary statements.

- We estimate the true proportion is 0.15, and thus reject the null hypothesis that the true proportion of sixes is 0.25.
- We fail to reject the null hypothesis.
- Based on a p-value of 0.4394, we conclude the true proportion of rolls that is six is 0.25.
- Based on the p-value, the probability is 0.4394 that the true proportion of rolls that result in six is 0.25.