Introduction to Statistics: T-Tests, F-tests, ANOVAs

Dr. Niccole M. Pamphilis

Introduction

This week we learned about the one-sample t-test. This test allowed us to look at the mean for one of our sample variables and talk about what the mean in the population, that our sample came from, might look like (null hypothesis).

Keep in mind that the one-sample t-test is appropriate for variables that are continuous in nature, where we can calculate a mean value. If you cannot, or should not, calculate the mean for the variable using this test is not appropriate.

In the lab today we will look at two different ways to run a one-sample t-test: non-direction (two-tailed test) and directional (one-tailed tests). Throughout we will be using our cleaned Scottish data.

```
data6<-readRDS(file = "cleanedScot_data.rds")
```

One-Sample T-test

The code for running a one-sample t-test uses the t.test() command. You will also need to know what variable you are going to use and what your null hypothesis is to run this test.

Below is a one-sample t-test looking at our age variable. This is a continuous variable so looking at the mean is appropriate. In 2011 the median age in the Scotland was 40 years old, this is my best guess at what the mean in the population today might be, but again this value is anything we are interested in testing against (I was just trying to locate a reasonable value).

```
#Notice the set-up. We start with the t.test() command
  #Next we select our variable of interest
  #Finally, we tell it what the null hypothesis is, where mu denotes the population value,
  #our null.
t.test(data6$age,
       mu=40)
```

```
##
##  One Sample t-test
##
## data:  data6$age
## t = 14.896, df = 99, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 40
## 95 percent confidence interval:
##  56.91979 62.12021
## sample estimates:
## mean of x
##     59.52
```

In the results above you will see it reminds you that you ran a one-sample t-test. It provides you with the empirical t value (which is what we practice calculating by hand). It gives you the degrees of freedom that we looked at when using the table. Here is provides the p-value for the test, which we will talk about in class next week.

R also provides you with the alternative hypothesis, in case you forget what you ran.

We will discuss what the confidence intervals are in two weeks, so we will skip over that for now.

Finally, the output provides you with the mean of the variable you were testing. Here we can see the mean of the age variable was 59.52.

Directional One-Sample T-test

If we want to run a directional one-sample t-test, where we believe that our sample is not only different than a given value, but we have expectations about how it is different.

The code to run a directional one-sample t-test starts out the same as we saw above for the non-directional test. However, we are now going to add an additional comment to tell R we are looking for a directional difference.

```
#Null below is mean is equal to or less than 40

t.test(data6$age,
       mu=40,
       alternative="greater")
```

```
##
##  One Sample t-test
##
## data:  data6$age
## t = 14.896, df = 99, p-value < 2.2e-16
## alternative hypothesis: true mean is greater than 40
## 95 percent confidence interval:
##  57.34415      Inf
## sample estimates:
## mean of x
##     59.52
```

```
#Notice, that the comment about the alternative hypothesis is add.
#The alternative specifies what we think the alternative hypothesis is.
#Here, we are say the alternative is that the mean of age is greater than 40.
```

Note, that the output for the one-sample t-tests for the directional test is set-up in a similar fashion to the non-directional test. And in fact the empirical t-value did not change either (because that is the standardized difference between our observed sample mean and the expected value of the population).

```
#Null below is mean is equal to or greater than 40

t.test(data6$age,
       alternative="less",
       mu=40)
```

```
##
##  One Sample t-test
##
## data:  data6$age
## t = 14.896, df = 99, p-value = 1
```

```
## alternative hypothesis: true mean is less than 40
## 95 percent confidence interval:
##      -Inf 61.69585
## sample estimates:
## mean of x
##     59.52
```

R versus by hand

Keep in mind that everything R has done with the t.test command is exactly what we have been calculating by hand. In fact, we can walk through this process to see exactly that.

The t-statistic is calculated as:

$$t = \frac{\overline{x} - \mu}{\overline{\sigma}/\sqrt{n}}$$

We already learned how to calculate the mean of a variable:

```
mean(data6$age)
```

```
## [1] 59.52
```

We know what μ is because we set it in our hypotheses: 40.

We can calculate the standard error which is: $\frac{s}{\sqrt{n}}$

So we can put it all together and just use R like a calculator:

```
(mean(data6$age)-40)/(sd(data6$age)/sqrt(100))
```

```
## [1] 14.89569
```

And, you will see that the t-statistic we calculated by "hand" here is the same as what R produced when it ran the test for us.

Two-Sample T-test

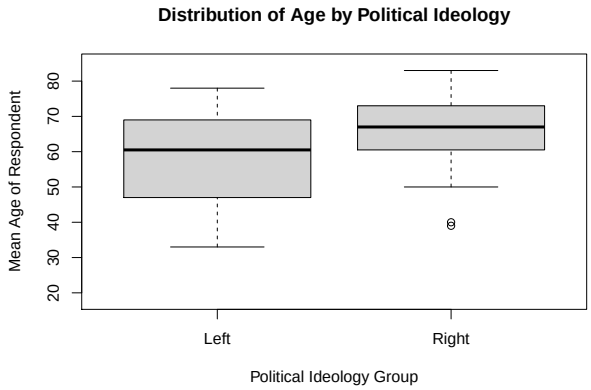
To run a two-sample t-test we need two groups to compare to each other,however, in our dataset we currently do not have any variables that are dichotomous in nature, so we are going to create a "dummy" variable, which is a variable with only two categories. For our example here I have chosen to create a binary left/right ideology variable, which is coded below:

```
#Here the code tells R to create a new variable called ideol_dum and attach it to my
#dataset

#The "ifelse" command tells are if an observation has a given trait code it one way (here,
#right and if it has any other value code it in another group (left)

data6$ideol_dum<-ifelse(data6$ideology>5, c("right"), c("left"))
```

Now, let's take a look at the distribution of age across our two groups. To do this, we can generate a boxplot.

```
boxplot(data6$age~data6$ideol_dum,
        ylim=c(18, 85),
        ylab="Mean Age of Respondent",
        xlab="Political Ideology Group",
```

```
        names=c("Left", "Right"),
        main="Distribution of Age by Political Ideology",
        aspect=1)
```



In the box plot above we can see that the median value for the right appears a bit higher than the median age for the left. Also, we can see that the right has at least two outliers at the lower end of the age distribution.

Keep in mind that the boxplot if for the median value and we are talking about comparing the means, so we also want to just look at the mean value for the two groups. We can do this by telling R to calculate the mean of age of a subset of the data (either the right or the left).

```
#Mean for observations on the Right

mean(data6$age[data6$ideol_dum =="right"], na.rm=T)
```

```
## [1] 65
```

```
#Mean for observation on the Left

mean(data6$age[data6$ideol_dum =="left"], na.rm=T)
```

```
## [1] 58.09259
```

```
#Notice the commands above use the na.rm=T to tell R to ignore missing values.
#What happens if you omit the na.rm command?
```

While the visual diagnosis is useful, we want to use a statistical test which can account for sample size and variation in the observation. So we can execute a two-sample t-test.

Similar to the one-sample t-test, the two-sample t-tests uses the same basic command t.test().

Now, in addition to specifying what variable we want to look at the mean of, and what our null hypothesis is (do you recall what the null hypothesis is for a two sample t-test?), we now need to add in a variable that tells R what the groups are.

```
#The ~is used to indicate that the next variable listed is the grouping variable. Here our
#ideology dummy variable.

t.test(data6$age~data6$ideol_dum, mu=0)
```

```
##
##  Welch Two Sample t-test
##
## data:  data6$age by data6$ideol_dum
```

```
## t = -2.124, df = 40.017, p-value = 0.0399
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -13.4799573  -0.3348575
## sample estimates:
##  mean in group left mean in group right
##            58.09259            65.00000
```

Now, in class you may recall that we talked about the equal variance an unequal variance assumption that we need to make when we run a two-sample t-test. By default R assumes unequal variance.

If you want R to assume equal variance you will need to specify that option using the var.equal=T statement.

```
#Two-Sample T-test Equal Variance Assumption

t.test(data6$age~data6$ideol_dum, mu=0, var.equal=T)
```

```
##
##  Two Sample t-test
##
## data:  data6$age by data6$ideol_dum
## t = -1.9639, df = 72, p-value = 0.0534
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -13.9186222   0.1038074
## sample estimates:
##  mean in group left mean in group right
##            58.09259            65.00000
```

```
#What happens to the results of the test when we assume equal variance? Did the result change?
```

Next we will explore how to test for equal and unequal variance between the groups.

Directional Test

The commands above have shown you how to run a two-sample t-test assuming equal and unequal variance. We also can further specify a directional hypothesis, depending on our confidence in the data.

To indicate that you have a directional hypothesis you will can add another option to your line of code. This time the option is alternative=“lesser” or alternative=“greater”, depending on what our alternative is.

```
#Here we are saying that mean for group 1 is less than the mean for group .

t.test(data6$age~data6$ideol_dum, mu=0, alternative = "less")
```

```
##
##  Welch Two Sample t-test
##
## data:  data6$age by data6$ideol_dum
## t = -2.124, df = 40.017, p-value = 0.01995
## alternative hypothesis: true difference in means is less than 0
## 95 percent confidence interval:
##       -Inf -1.431496
## sample estimates:
##  mean in group left mean in group right
##            58.09259            65.00000
```

```r
#It is important to know which group R is treating as group 1 and group 2, so you know which
#it is subtracting. In this situation R makes left group 1 and right group 2.
```

```r
#This time we are assuming that the mean for group 2 (right) is greater than the mean of
#group 1 (left)

t.test(data6$age~data6$ideol_dum, mu=0, alternative = "greater")
```

```
##
##  Welch Two Sample t-test
##
## data:  data6$age by data6$ideol_dum
## t = -2.124, df = 40.017, p-value = 0.9801
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
##  -12.38332       Inf
## sample estimates:
##  mean in group left mean in group right
##            58.09259            65.00000
```

T-test without a binary variable

There may be situations where you do not have a binary variable, but you would still like to compare the mean between two subgroups. We can do this using a similar subsetting command to what we saw earlier when we created the binary variable. This approach tells R to compare the mean for two groups specified on a variable with multiple groups, but does not require you to create a new variable.

```r
#Here the subset command is used twice, the first time I am selecting the highest values
#on the political attention variable to calculate the mean for. The second subset command
#tells R what the other group will be, here those with the lowest political attention.

t.test(subset(data6$age, data6$polattention==10),
       subset(data6$age, data6$polattention==0),
       mu=0)
```

```
##
##  Welch Two Sample t-test
##
## data:  subset(data6$age, data6$polattention == 10) and subset(data6$age, data6$polattention == 0)
## t = -0.10059, df = 5.4545, p-value = 0.9235
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -21.60576  19.93910
## sample estimates:
## mean of x mean of y
##  64.50000  65.33333
```

Subsetting

Above we needed to create a new dummy variable in order to have two groups to compare for our two-sample t-test. We used the subset command and created a new variable. When we created our variable we used information on a single variable to determine how an observation was coded. However, we can also use

information across multiple variables to subset our data, where information on a few traits is necessary to determine groupings, we can do this in a few ways.

```
#Here I am subsetting the age variable based on education and ideology.

#Also, note that I have not created a new variable. I have only temporarily told R to look
#at this set

mean(subset(data6$age, data6$education=="medium" & data6$ideol_dum=="left"))
```

```
## [1] 59.25
```

If you want create multiple paired subsets across two variables you can use the tapply function. We will introduce it here, but it is a common function you will see pop up again and again.

```
#Here the first variable indicates what will be subset.

#The list tells R what value pairings it should look at.

#The final option "mean", tells R what to do with the groups.
tapply(data6$age, list(data6$education, data6$ideol_dum), mean)
```

```
##            left    right
## low    64.00000 69.50000
## medium 59.25000 67.50000
## high   54.26087 57.83333
```

Confidence Intervals

In class we talked about confidence intervals. By default in R (any many other statistical programs) the confidence intervals are calculated based on a 95% default level. However, this is something we can change, if we want.

To change the confidence level in a test or calculation, you need to add the conf.level= option.

Notice that in the first example there is no difference in the confidence interval when I add the option at set it to .95 or if I leave it omitted (because it is the default assumption R makes).

```
t.test (data6$age)
```

```
##
##  One Sample t-test
##
## data:  data6$age
## t = 45.42, df = 99, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  56.91979 62.12021
## sample estimates:
## mean of x
##     59.52
```

```
t.test (data6$age,
        conf.level=0.95)
```

```
##
##  One Sample t-test
```

```
##
## data:  data6$age
## t = 45.42, df = 99, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  56.91979 62.12021
## sample estimates:
## mean of x
##     59.52
```

If I want to change the level to 90% or 99% I can do this by adjusting the conf.level option as shown below.

```
t.test (data6$age, conf.level=0.90)
```

```
##
##  One Sample t-test
##
## data:  data6$age
## t = 45.42, df = 99, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 0
## 90 percent confidence interval:
##  57.34415 61.69585
## sample estimates:
## mean of x
##     59.52
```

```
t.test (data6$age, conf.level=0.99)
```

```
##
##  One Sample t-test
##
## data:  data6$age
## t = 45.42, df = 99, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 0
## 99 percent confidence interval:
##  56.07824 62.96176
## sample estimates:
## mean of x
##     59.52
```

F-tests

F-tests allow us to compare the variances between two groups. We use this ratio of variance test when trying to figure which type of two-sample t-test to run. There are other applications of this test that are used which you will see as you progress in your quantitative studies.

Last week we ran a two-sample t-test comparing the mean of age for those on the left versus the right with our ideol_dum variable. But, we did not know which test to use (assuming equal or unequal variance) and each test produced a different result. Let's review:

```
#Unequal variance
t.test(data6$age~data6$ideol_dum, mu=0)
```

```
##
##  Welch Two Sample t-test
##
```

```
## data:  data6$age by data6$ideol_dum
## t = -2.124, df = 40.017, p-value = 0.0399
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -13.4799573  -0.3348575
## sample estimates:
##  mean in group left mean in group right
##            58.09259            65.00000
```

In the above example we reject the null at the 95% confidence level, means between groups are not equal in the population.

```
#Equal variance
t.test(data6$age~data6$ideol_dum, mu=0, var.equal=T)
```

```
##
##  Two Sample t-test
##
## data:  data6$age by data6$ideol_dum
## t = -1.9639, df = 72, p-value = 0.0534
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -13.9186222   0.1038074
## sample estimates:
##  mean in group left mean in group right
##            58.09259            65.00000
```

Assuming equal variance we fail to reject the null hypothesis at the 95% confidence level, cannot reject null that means between groups are equal.

So, let's run an F-test and see which test was the appropriate one to run, and which results are therefore the appropriate ones to interpret.

The F-test use the command var.test(), as this is a variance test. We also need to tell R what the variable it is calculating the mean of is and the grouping variable. The ratio command is our null hypothesized value (equal variance =1).

```
var.test(data6$age~data6$ideol_dum,
         ratio=1)
```

```
##
##  F test to compare two variances
##
## data:  data6$age by data6$ideol_dum
## F = 1.4039, num df = 53, denom df = 19, p-value = 0.4203
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.6139935 2.8017767
## sample estimates:
## ratio of variances
##           1.403935
```

Based on the results here we would fail to reject the null hypothesis of equal variance. So, we should have run the test assuming equal variance. If we do this, what is our finding from the correct two-sample t-test?

Anovas

The last test we covered this week was the Anova (analysis of variance). Here we looked at the means for groups relative to how much variation was present in the groups. This approach allowed us to make statements about whether or not the means of the groups indicated any true meaningful differences between how the groups behaved relative to one another.

The Anova tests the means for +3 groups and indicates whether or not at least one group is behaving differently than the rest.

To run an Anova in R you use the aov(). Here, we will look at the mean of age relative to education level. As with other commands we have run before the first variable is what we are testing, and the second variable following the ~ is our grouping variable.

```
aov(data6$age ~ data6$education)
```

```
## Call:
##    aov(formula = data6$age ~ data6$education)
##
## Terms:
##                 data6$education Residuals
## Sum of Squares          1478.73  15522.23
## Deg. of Freedom               2        97
##
## Residual standard error: 12.65002
## Estimated effects may be unbalanced
```

Notice, that in the output above it provides you with the table of results calculating the variation attributable to groups and the variation attributable to observations. But, there are no p-values to tell us if there are statistically meaningful differences. To do this, we have to add one more command: summary(). This command can be added in one of two ways.

```
#First, we can add the summary command directly in to the line of code running the Anova

summary(aov(data6$age ~ data6$education))
```

```
##                 Df Sum Sq Mean Sq F value Pr(>F)
## data6$education  2   1479   739.4    4.62 0.0121 *
## Residuals       97  15522   160.0
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Notice, that now we see the F-statistic and the accompanying p-value.

```
#Second, you could run the test and save the results as an object in R and then tell R to summarise the

#This is useful if you want to look at these results again later without have to rerun the
#test

aov.results<-aov(age ~ education, data=data6)
```

```
summary(aov.results)
```

```
##             Df Sum Sq Mean Sq F value Pr(>F)
## education    2   1479   739.4    4.62 0.0121 *
## Residuals   97  15522   160.0
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Notice above that I used the option data= instead of data$ to tell R where to find my variables.
This is a shorthand bit of code which can save you from typing data$ in from of every variable you need.