

Lab #9

In lab #9 we are going to cover ways to analyze one sample continuous data. We will learn how to use SAS/R to carry out paired t -tests, calculate one-sample t confidence intervals, as well as find probabilities and quantiles in the T distribution.

1 Confidence intervals

For this section we will again be using your individual subsample from the NHANES data set. You can find this set on the course webpage under Data → Individual.

In SAS, the UNIVARIATE procedure can be used to calculate confidence intervals based on the t -distribution; all you need to do is specify the CIBASIC option. In R, the `t.test` function carries out t -tests and constructs confidence intervals, just as `binom.test` and `prop.test` did.

SAS:	R:
<pre>PROC UNIVARIATE DATA=MyNhanes CIBASIC; VAR trg; RUN;</pre>	<pre>attach(MyNhanes) t.test(TRG)</pre>

By default, SAS/R give you a 95% confidence interval (in SAS, this is located in the section of the output called “Basic Confidence Limits Assuming Normality”). This can be changed by also specifying ALPHA (in SAS) or `conf.level` (in R):

SAS:	R:
<pre>PROC UNIVARIATE DATA=MyNhanes CIBASIC ALPHA=.1; VAR trg; RUN;</pre>	<pre>t.test(TRG, conf.level=.9)</pre>

Does your confidence interval contain the true mean of entire NHANES sample (116.9)?

How about a 75% confidence interval? A 50% interval?

Note that both SAS and R by default carry out a hypothesis test of whether the population mean is equal to zero. Just because the software gives you this doesn't mean that this is a meaningful test to look at or report. You certainly don't need a statistical test to tell you that the average triglyceride levels of adult women are not equal to zero.

2 *t*-tests on one sample data

Suppose that we wanted to test our sample mean against a hypothesized value. The `t.test` procedure in SAS with the option `H0` allows us to do this. In R we can again use the `t.test` function with the argument `mu =`

Below is code testing triglyceride levels against a hypothesized value of 120. How would you interpret the results of this test?

SAS:

```
PROC TTEST DATA=MyNhanes H0 = 120;
  VAR trg;
RUN;
```

R:

```
t.test(TRG, mu = 120)
```

3 *t*-tests on paired data

For this section will be using the `cysticfibrosis.txt` dataset. Recall that this was a crossover study and that the data is paired. A “paired *t*-test” is exactly the same thing as a one-sample *t*-test done on the differences between the paired observations.

You have two ways of carrying out a paired *t*-test: (1) create a new variable which records the difference between patients’ performance on drug and placebo, then use `PROC UNIVARIATE/t.test` as above, or (2) instruct SAS/R to carry out a paired *t*-test. The second approach is easier, so we will focus on it:

SAS:

```
PROC TTEST DATA=cf;
  PAIRED Placebo*Drug;
RUN;
```

R:

```
detach(MyNhanes)
attach(cf)
t.test(Drug,Placebo,paired=TRUE)
t.test(Drug-Placebo) ## Same thing
```

Note that this gives you the same *p*-value and confidence intervals that we calculated in class. You could once again specify the `ALPHA/conf.level` option if you wanted confidence intervals other than the default 95% ones.

R note: Note the use of `detach`, which reverses `attach`. It is a good idea to detach a data set when you’re done using it (or open a new instance of R) if you’re going to move on to a different data set. It wouldn’t cause a problem in this case, but if you had a variable called `Drug` in both data sets, then this could cause problems, since R won’t know which `Drug` variable you want.

Finally, note the `paired=TRUE` option in R. If you don’t tell R to do a paired *t*-test, it will assume that the two sets of numbers you’re giving it come from two separate samples and do a two-sample *t*-test (which we will discuss in a few weeks). This represents a substantial loss of power because you are failing to take advantage of the paired design. Note that if you carry out this analysis, the *p*-value increases from 0.04 to 0.17.

4 Students curve: probabilities and quantiles

Calculating probabilities and quantiles from the t-distribution is very similar to that of the normal distribution. In R, the `pt` function calculates the area under Students curve to the left of any number. The only change is that you also have to specify the degrees of freedom. So, for example:

SAS:	R:
<code>CDF("T",1.96,5)</code>	<code>pt(1.96,5)</code>
<code>CDF("T",1.96,50)</code>	<code>pt(1.96,50)</code>
<code>CDF("T",1.96,500)</code>	<code>pt(1.96,500)</code>

This calculates the area to the left of 1.96 for Students curve with 5, 50, and 500 degrees of freedom, respectively. Notice that as df gets large, we get an answer very close to the 0.975 that we get from the normal distribution, but when $df = 5$, there is more area in the upper tail of Students curve (about 5.5%) than there is in the normal curve (2.5%).

Note that SAS is inconvenient to use for calculating probabilities and quantiles. Recall that we would need to use the complete syntax shown below to get one of our desired probabilities:

SAS:	R:
<code>DATA temp;</code>	<code>PUT x;</code>
<code>x = CDF("T",1.96,5)</code>	<code>RUN;</code>

Quantiles work the same way:

SAS:	R:
<code>QUANTILE("T",.975,5)</code>	<code>qt(.975,5)</code>
<code>QUANTILE("T",.975,50)</code>	<code>qt(.975,50)</code>
<code>QUANTILE("T",.975,500)</code>	<code>qt(.975,500)</code>

Again, when df is large, we get something close to the normal value, 1.96. But when df is smaller, we need to go out much wider to capture 95% of the area.