```yaml
  Course: Statistical_Inference
  Lesson: T_Confidence_Intervals

- Class: text
  Output: "T_Confidence_Intervals. (Slides for this and other Data Science courses may be found at github https://github.com/DataScienceSpecialization/courses/. If you care to use them, they must be downloaded as a zip file and viewed locally. This lesson corresponds to 06_Statistical_Inference/08_tCIs.)"

- Class: text
  Output: In this lesson, we'll discuss some statistical methods for dealing with small datasets, specifically the Student's or Gosset's t distribution and t confidence intervals.

- Class: mult_question
  Output: In the Asymptotics lesson we discussed confidence intervals using the Central Limit Theorem (CLT) and normal distributions. These needed large sample sizes, and the formula for computing the confidence interval was Est +/- qnorm *std error(Est), where Est was some estimated value (such as a sample mean) with a standard error. Here qnorm represented what?
  AnswerChoices: the population mean; the population variance; the standard error; a specified quantile from a normal distribution
  CorrectAnswer: a specified quantile from a normal distribution
  AnswerTests: omnitest(correctVal='a specified quantile from a normal distribution')
  Hint: Which choice has part of the word 'qnorm' in it?

- Class: mult_question
  Output: In the Asymptotics lesson we also mentioned the Z statistic Z=(X'-mu)/(sigma/sqrt(n)) which follows a standard normal distribution. This normalized statistic Z is especially nice because we know its mean and variance. They are what, respectively?
  AnswerChoices: 0 and 1; 1 and 0; 0 and 0; 1 and 1
  CorrectAnswer:  0 and 1
  AnswerTests: omnitest(correctVal='0 and 1')
  Hint: Recall the definition of standard normal. It's centered around 0 and it has a standard deviation of 1 so its mean and variance are what?.

- Class: text
  Output: So the mean and variance of the standardized normal are fixed and known. Now we'll define the t statistic which looks a lot like the Z. It's defined as t=(X'-mu)/(s/sqrt(n)). Like the Z statistic, the t is centered around 0. The only difference between the two is that the population std deviation, sigma, in Z is replaced by the sample standard deviation in the t. So the distribution of the t statistic is independent of the population mean and variance. Instead it depends on the sample size n.

- Class: text
  Output: As a result, for t distributions, the formula for computing a confidence interval is similar to what we did in the last lesson. However, instead of a quantile for a normal distribution we use a quantile for a t distribution. So the formula is Est +/- t-quantile *std error(Est). The other distinction, which we mentioned before, is that we'll use the sample standard deviation when we estimate the standard error of Est.

- Class: mult_question
  Output: In the formula for the t statistic t=(X'-mu)/(s/sqrt(n)) what expression represents the sample standard deviation?
  AnswerChoices: X'; mu; s; n
  CorrectAnswer: s
  AnswerTests: omnitest(correctVal='s')
  Hint: X' and mu represent means, and n usually represents an integer like sample size.

- Class: text
  Output: These t confidence intervals are very handy, and if you have a choice between these and normal, pick these. We'll see that as datasets get larger, t-intervals look normal. We'll cover the one- and two-group versions which depend on the data you have.

- Class: text
  Output: The t distribution, invented by William Gosset in 1908, has thicker tails
```

```yaml
  than the normal. Also, instead of having two parameters, mean and variance, as the normal does, the t distribution has only one - the number of degrees of freedom (df).

- Class: text
  Output: As df increases, the t distribution gets more like a standard normal, so it's centered around 0. Also, the t assumes that the underlying data are iid Gaussian so the statistic (X' - mu)/(s/sqrt(n)) has n-1 degrees of freedom.

- Class: mult_question
  Output: Quick check. In the formula t=(X' - mu)/(s/sqrt(n)), if we replaced s by sigma the statistic t would be what asymptotically?.
  AnswerChoices: the standard normal; the standard abnormal; the population variance; Huh?
  CorrectAnswer: the standard normal
  AnswerTests: omnitest(correctVal='the standard normal')
  Hint: With the replacement the formula should look familiar, like a standardized normal perhaps?

- Class: figure
  Output: To see what we mean, we've taken code from the slides, the function myplot, which takes the integer df as its input and plots the t distribution with df degrees of freedom. It also plots a standard normal distribution so you can see how they relate to one another.
  Figure: tPlot.R
  FigureType: new

- Class: cmd_question
  Output: Try myplot now with an input of 2.
  CorrectAnswer: myplot(2)
  AnswerTests: omnitest(correctExpr='myplot(2)')
  Hint: Type myplot(2) at the command prompt.

- Class: cmd_question
  Output: You can see that the hump of t distribution (in blue) is not as high as the normal's. Consequently, the two tails of the t distribution absorb the extra mass, so they're thicker than the normal's. Note that with 2 degrees of freedom, you only have 3 data points. Ha! Talk about small sample sizes. Now try myplot  with an input of 20.
  CorrectAnswer: myplot(20)
  AnswerTests: omnitest(correctExpr='myplot(20)')
  Hint: Type myplot(20) at the command prompt.

- Class: text
  Output: The two distributions are almost right on top of each other using this higher degree of freedom.

- Class: figure
  Output:  Another way to look at these distributions is to plot their quantiles. From the slides, we've provided a second function for you, myplot2, which does this. It plots a lightblue reference line representing normal quantiles and a black line for the t quantiles. Both plot the quantiles starting at the 50th percentile which is 0 (since the distributions are symmetric about 0) and go to the 99th.
  Figure: tQuant.R
  FigureType: new

- Class: cmd_question
  Output: Try myplot2 now with an argument of 2.
  CorrectAnswer: myplot2(2)
  AnswerTests: omnitest(correctExpr='myplot2(2)')
  Hint: Type myplot2(2) at the command prompt.

- Class: text
  Output: The distance between the two thick lines represents the difference in sizes between the quantiles and hence the two sets of intervals. Note the thin horizontal and vertical lines. These represent the .975 quantiles for the t and normal distributions respectively. Anyway, you probably recognized the placement of the vertical at 1.96 from the Asymptotics lesson.

- Class: cmd_question
  Output: Check the placement of the horizontal now using the R function qt with the
```

```yaml
  arguments .975 for the quantile and 2 for the degrees of freedom (df).
  CorrectAnswer: qt(.975,2)
  AnswerTests: omnitest(correctExpr='qt(.975,2)')
  Hint: Type qt(.975,2) at the command prompt.

- Class: cmd_question
  Output: See? It matches the horizontal line of the plot. Now run myplot2 with an argument of 20.
  CorrectAnswer: myplot2(20)
  AnswerTests: omnitest(correctExpr='myplot2(20)')
  Hint: Type myplot2(20) at the command prompt.

- Class: text
  Output: The quantiles are much closer together with the higher degrees of freedom. At the 97.5 percentile, though, the t quantile is still greater than the normal. Student's Rules!

- Class: text
  Output: This means the the t interval is always wider than the normal. This is because estimating the standard deviation introduces more uncertainty so a wider interval results.

- Class: text
  Output: So the t-interval is defined as X' +/- t_(n-1)*s/sqrt(n) where t_(n-1) is the relevant quantile. The t interval  assumes that the data are iid normal, though it is robust to this assumption and  works well whenever the distribution of the data is roughly symmetric and mound shaped.

- Class: mult_question
  Output: Our plots showed us that for large degrees of freedom, t quantiles become close to what?
  AnswerChoices: standard normal quantiles; standard abnormal quantiles; very large numbers; very small numbers
  CorrectAnswer: standard normal quantiles
  AnswerTests: omnitest(correctVal='standard normal quantiles')
  Hint: Recall that the larger the degrees of freedom, the more the t distribution looked normal. Smaller degrees of freedom made it look abnormal.

- Class: text
  Output: Although it's pretty great, the t interval isn't always applicable. For skewed distributions, the spirit of the t interval assumptions (being centered around 0) are violated. There are ways of working around this problem (such as taking logs or using a different summary like the median).

- Class: text
  Output: For highly discrete data, like binary, intervals other than the t are available.

- Class: text
  Output: However, paired observations are often analyzed using the t interval by taking differences between the observations. We'll show you what we mean now.

- Class: text
  Output: We hope you're not tired because we're going to look at some sleep data. This was the data originally analyzed in Gosset's Biometrika paper, which shows the increase in hours for 10 patients on two soporific drugs.

- Class: cmd_question
  Output: We've loaded the data for you. R treats it as two groups rather than paired. To see what we mean type sleep now. This will show you how the data is stored.
  CorrectAnswer: sleep
  AnswerTests: omnitest(correctExpr='sleep')
  Hint: Type sleep at the command prompt.

- Class: text
  Output: We see 20 entries, the first 10 show the results (extra) of the first drug (group 1) on each of the patients (ID), and the last 10 entries the results of the second drug (group 2) on each patient (ID).
```

```
- Class: figure
  Output:  Here we've plotted the data in a paired way, connecting each patient's two results with a line, group 1 results on the left and group 2 on the right. See that purple line with the steep slope? That's ID 9, with 0 result for group 1 and 4.6 for group 2.
  Figure: sleepPlot.R
  FigureType: new

- Class: text
  Output: If  we just looked at the 20 data points we'd be comparing group 1 variations with group 2 variations. Both groups have quite large ranges. However, when we look at the data paired for each patient, we see that the variations in results are usually much smaller and depend on the particular subject.

- Class: cmd_question
  Output: To clarify, we've defined some variables for you, namely g1 and g2. These are two 10-long vectors, respectively holding the results of the 10 patients for each of the two drugs. Look at the range of g1 using the R command range.
  CorrectAnswer: range(g1)
  AnswerTests: omnitest(correctExpr='range(g1)')
  Hint: Type range(g1) at the command prompt.

- Class: cmd_question
  Output: So g1 values go from -1.6 to 3.7. Now look at the range of g2. We see that the ranges of both groups are relatively large.
  CorrectAnswer: range(g2)
  AnswerTests: omnitest(correctExpr='range(g2)')
  Hint: Type  range(g2) at the command prompt.

- Class: cmd_question
  Output:  Now let's look at the pairwise difference. We can take advantage of R's componentwise subtraction of vectors and create the vector of difference by subtracting g1 from g2. Do this now and put the result in the variable difference.
  CorrectAnswer: difference <- g2-g1
  AnswerTests: expr_creates_var("difference"); omnitest(correctExpr='difference <- g2-g1')
  Hint: Type  difference <- g2-g1 at the command prompt.

- Class: cmd_question
  Output:  Now use the R function mean to find the average of difference.
  CorrectAnswer: mean(difference)
  AnswerTests: omnitest(correctExpr='mean(difference)')
  Hint: Type mean(difference) at the command prompt.

- Class: text
  Output: See how much smaller the mean difference in this paired data is compared to the group variations?

- Class: cmd_question
  Output:  Now use the R function sd to find the standard deviation of  difference and put the result in the variable s.
  CorrectAnswer: s <- sd(difference)
  AnswerTests: expr_creates_var("s"); omnitest(correctExpr='s <- sd(difference)')
  Hint: Type s <- sd(difference) at the command prompt.

- Class: cmd_question
  Output:  Now recall the formula for finding the t confidence interval, X' +/- t_(n-1)*s/sqrt(n). Make the appropriate substitutions to find the 95% confidence intervals for the average difference you just computed. We've stored that average difference in the variable mn for you to use here. Remember to use the R construct c(-1,1) for the +/- portion of the formula and the R function qt with .975 and n-1 degrees of freedom for the quantile portion. Our data size is 10.
  CorrectAnswer: mn + c(-1,1)*qt(.975,9)*s/sqrt(10)
  AnswerTests:  omnitest(correctExpr='mn + c(-1,1)*qt(.975,9)*s/sqrt(10)')
  Hint: Type mn + c(-1,1)*qt(.975,9)*s/sqrt(10) at the command prompt.

- Class: text
  Output: This says that with probability .95 the average difference of effects (between the two drugs) for an individual patient is between .7 and 2.46 additional
```

```
    hours of sleep.

- Class: cmd_question
  Output:  We could also just have used the R function t.test with the argument
    difference to get this result. (You can use the default values for all the other
    arguments.) As with the other R test functions, this returns a lot of information.
    Since all we're interested in at the moment is the confidence interval we can pick
    this off with the construct x$conf.int. Try this now.
  CorrectAnswer: t.test(difference)$conf.int
  AnswerTests:  omnitest(correctExpr='t.test(difference)$conf.int')
  Hint: Type t.test(difference)$conf.int at the command prompt.

#- Class: video
#  Output: As the slides showed,  R provides several ways of using t.test to find the
confidence interval of this data. Would you like to see the R code to see 4
alternatives (including the two we just went through) and how to display them nicely?
You'll need an internet connection to see it.
#  VideoLink: "http://wilcrofter.github.io/slidex/markDown/ttest.html"

- Class: figure
  Output: Here's code from the slides which shows four different ways of using t.test
    (including the two we just went through) to find the confidence interval of this
    data. The code also shows how to display the intervals nicely in a 4 x 2 array.
  Figure: plot4Ttests.R
  FigureType: new


- Class: text
  Output: We now present methods, using t confidence intervals, for comparing
    independent groups.

- Class: text
  Output: Suppose that we want to compare the mean blood pressure between two groups in
    a randomized trial. We'll compare those who received the treatment to those who
    received a placebo. Unlike the sleep study, we cannot use the paired t test because
    the groups are independent and may have different sample sizes.

- Class: text
  Output: So our goal is to find a 95% confidence interval of the difference between
    two population means. Let's represent this difference as mu_y - mu_x. How do we do
    this? Recall our formula X' +/- t_(n-1)*s/sqrt(n).

- Class: text
  Output: First we need a sample mean, but we have two, X' and Y', one from each group.
    It makes sense that we'd have to take their difference (Y'-X') as well, since we're
    looking for a confidence interval that contains the difference mu_y-mu_x. Now we need
    to specify a t quantile. Suppose the groups have different sizes n_x and n_y.

- Class: mult_question
  Output: For one group we used the  quantile t_(.975,n-1). What do you think we'll use
    for the quantile of this problem?
  AnswerChoices: t_(.975,n_x-1); t_(.975,n_y-n_x-2); t_(.975,n_x+n_y-1);
    t_(.975,n_x+n_y-2)
  CorrectAnswer: t_(.975,n_x+n_y-2)
  AnswerTests: omnitest(correctVal='t_(.975,n_x+n_y-2)')
  Hint: We lose one degree of freedom from each group because we've calculated the
    sample mean from each group, so we add the two sizes and subtract two.

- Class: text
  Output: The only term remaining is the standard error which for the single group is
    s/sqrt(n). Let's deal with the numerator first. Our interval will assume (for now) a
    common variance s^2 across the two groups. We'll actually pool variance information
    from the two groups using a weighted sum. (We'll deal with the more complicated
    situation later.)

- Class: text
  Output: We call the variance estimator we use the pooled variance. The formula for it
    requires two variance estimators (in the form of the standard deviation), S_x and
    S_y, one for each group. We multiply each by its respective degrees of freedom and
```

```
  divide the sum by the total number of degrees of freedom. This  weights the respective variances; those coming from bigger samples get more weight.

- Class: mult_question
  Output: Which of the following represents the numerator of this expression?
  AnswerChoices: (n_x-1)(S_x)^2+(n_y-1)(S_y)^2; (n_x)(S_x)^2+(n_y)(S_y)^2; (n_x)(S_x)+(n_y)(S_y)
  CorrectAnswer: (n_x-1)(S_x)^2+(n_y-1)(S_y)^2
  AnswerTests: omnitest(correctVal='(n_x-1)(S_x)^2+(n_y-1)(S_y)^2')
  Hint: We need variances so the choice without the squared S terms is incorrect. Recall that the degrees of freedom is one less than the sample size for each group so that eliminates another choice and only one choice remains.


- Class: mult_question
  Output: Which of the following represents the total number of degrees of freedom?
  AnswerChoices: (n_x-1)+(n_y-1); (n_x+n_y); (n_x+n_y-1); (n_x+n_y+2)
  CorrectAnswer: (n_x-1)+(n_y-1)
  AnswerTests: omnitest(correctVal='(n_x-1)+(n_y-1)')
  Hint: Recall that the degrees of freedom is one less than the sample size for each group. We asked this a few questions ago, though we've put this answer in a different, but equivalent form.

- Class: text
  Output: Now recall we're calculating the standard error term which for the single group case was s/sqrt(n). We've got the numerator done, by pooling the sample variances. How do we handle the 1/sqrt(n) portion? We can simply add 1/n_x and 1/n_y and take the square root of the sum. Then we MULTIPLY this by the sample variance to complete the estimate of the standard error.

- Class: text
  Output: Now we'll plug in some numbers from the slides based on an example from Rosner's book Fundamentals of Biostatistics, a very good, if heavy, reference book. We want to compare blood pressure from two independent groups.

- Class: cmd_question
  Output: The first is a group of 8 oral contraceptive users and the second is a group of 21 controls. The two means are X'_{oc}=132.86 and X'_{c}=127.44, and the two sample standard deviations are s_{oc}= 15.34 and s_{c}= 18.23. Let's first compute the numerator of the pooled sample variance by weighting the sum of the two by their respective sample sizes. Recall the formula (n_x-1)(S_x)^2+(n_y-1)(S_y)^2 and fill in the values to create a variable sp.
  CorrectAnswer: sp <- 7*15.34^2 + 20*18.23^2
  AnswerTests:  expr_creates_var('sp'); omnitest(correctExpr='sp <- 7*15.34^2 + 20*18.23^2',correctVal=8293.8672)
  Hint: Type sp <- 7*15.34^2 + 20*18.23^2 at the command prompt. Here 7 and 20 are each one less than the given sample sizes, and 15.34 and 18.23 are the respective standard deviations. We square these to convert them to variances.

- Class: cmd_question
  Output: Now how many degrees of freedom are there? Put your answer in the variable ns.
  CorrectAnswer: ns <- 8+21-2
  AnswerTests:  expr_creates_var('ns'); omnitest(correctExpr='ns <- 8+21-2',correctVal=27)
  Hint: Add the two sample sizes and subtract 2. Put the result in ns.

- Class: cmd_question
  Output: Now divide sp by ns, take the square root and put the result back in sp.
  CorrectAnswer: sp <- sqrt(sp/ns)
  AnswerTests:  expr_creates_var('sp'); omnitest(correctExpr='sp <- sqrt(sp/ns)')
  Hint: Type sp <- sqrt(sp/ns) at the command prompt.

- Class: cmd_question
  Output: Now to find the 95% confidence interval. Recall our basic formula X' +/- t_(n-1)*s/sqrt(n) and all the changes we need to make for working with two independent samples. We'll plug in the difference of the sample means for X' and our variable ns for the degrees of freedom when finding the t quantile. For the standard error, we multiply sp  by the square root of the sum 1/n_{oc} + 1/n_{c}. The values for this problem are X'_{oc}=132.86 and X'_{c}=127.44, n_{oc}=8 and n_{c}=21. Be sure
```

```
  to use the R construct c(-1,1) for the +/- portion and the R function qt with the correct percentile and degrees of freedom.
  CorrectAnswer: 132.86-127.44+c(-1,1)*qt(.975,ns)*sp*sqrt(1/8+1/21)
  AnswerTests: omnitest(correctExpr='132.86-127.44+c(-1,1)*qt(.975,ns)*sp*sqrt(1/8+1/21)')
  Hint: Type 132.86-127.44+c(-1,1)*qt(.975,ns)*sp*sqrt(1/8+1/21) at the command prompt.

- Class: text
  Output: Notice that 0 is contained in this 95% interval. That means that you can't rule out that the means of the two groups are equal since a difference of 0 is in the interval.

- Class: text
  Output: Getting tired? Let's revisit the sleep problem and instead of looking at the data as paired over 10 subjects we'll look at it as two independent sets each of size 10. Recall the data is stored in the two vectors g1 and g2; we've also stored the difference between their means in the variable md.

- Class: cmd_question
  Output: Let's compute the sample pooled variance and store it in the variable sp. Recall that this is the sqrt(weighted sums of sample variances/deg of freedom). The weight of each is the sample size-1. Use the R function var to compute the variances of  g1 and g2. The degrees of freedom is 10+10-2 = 18.
  CorrectAnswer: sp <- sqrt((9*var(g1)+9*var(g2))/18)
  AnswerTests:  expr_creates_var('sp'); omnitest(correctExpr='sp <- sqrt((9*var(g1)+9*var(g2))/18)')
  Hint: Type sp <- sqrt((9*var(g1)+9*var(g2))/18) at the command prompt.

- Class: cmd_question
  Output: Now  the last term of the formula, the standard error of the mean difference, is simply sp times the square root of the sum 1/10 + 1/10. Find the 95% t confidence interval of the mean difference of the two groups g1 and g2. Substitute md and sp into the formula you used above.
  CorrectAnswer: md + c(-1,1)*qt(.975,18)*sp*sqrt(1/5)
  AnswerTests:  ANY_of_exprs('md + c(-1,1)*qt(.975,18)*sp*sqrt(1/5)','md + c(-1,1)*qt(.975,18)*sp*sqrt(1/10 + 1/10)')
  Hint: Type md + c(-1,1)*qt(.975,18)*sp*sqrt(1/5) at the command prompt.

- Class: cmd_question
  Output: We can check this manual calculation against the R function t.test. Since we subtracted g1 from g2, be sure to place g2 as your first argument and g1 as your second. Also make sure the argument paired is FALSE and var.equal is TRUE. We only need the confidence interval so use the construct x$conf.  Do this now.
  CorrectAnswer: t.test(g2,g1,paired=FALSE,var.equal=TRUE)$conf
  AnswerTests:  omnitest(correctExpr='t.test(g2,g1,paired=FALSE,var.equal=TRUE)$conf')
  Hint: Type t.test(g2,g1,paired=FALSE,var.equal=TRUE)$conf at the command prompt.

- Class: cmd_question
  Output: Pretty cool that it matches, right? Note that 0 is again in this 95% interval so you can't reject the claim that the two groups are the same. (Recall that this is the opposite of what we saw with paired data.) Let's run t.test again, this time with paired=TRUE and see how different the result is. Don't specify var.equal and look only at the confidence interval.
  CorrectAnswer: t.test(g2,g1,paired=TRUE)$conf
  AnswerTests:  omnitest(correctExpr='t.test(g2,g1,paired=TRUE)$conf')
  Hint: Type t.test(g2,g1,paired=TRUE)$conf at the command prompt.

- Class: text
  Output: Just as we saw when we ran t.test on our vector, difference! See how the interval excludes 0? This means the groups when paired have much different averages.

- Class: text
  Output: Now let's talk about calculating confidence intervals for two groups which have unequal variances. We won't be pooling them as we did before.

- Class: text
  Output: In this case the formula for the interval is similar to what we saw before, Y'-X' +/- t_df * SE, where as before Y'-X' represents the difference of the sample means. However, the standard error SE and the quantile t_df are calculated
```

```
    differently from previous methods. Here SE is the square root of the sum of the
    squared standard errors of the two means, (s_1)^2/n_1 + (s_2)^2/n_2 .

- Class: text
  Output: When the underlying X and Y data are iid normal and the variances are
    different, the normalized statistic we started this lesson with, (X'-mu)/(s/sqrt(n)),
    doesn't follow a t distribution. However, it can be approximated by a t distribution
    if we set the degrees of freedom appropriately.

- Class: text
  Output: The formula for the degrees of freedom is a complicated fraction that no one
    remembers.  The numerator is the SQUARE of the sum of the squared standard errors of
    the two sample means. Each has the form s^2/n. The denominator is the sum of two
    terms, one for each group. Each term has the same form. It is the standard error of
    the mean raised to the fourth power divided by the sample size-1. More precisely,
    each term looks like (s^4/n^2)/(n-1). We use this df to find the t quantile.

#- Class: video
#  Output: Would you like to see this formula nicely displayed? You'll need an internet
  connection to do this.
#  VideoLink: "http://wilcrofter.github.io/slidex/markDown/diffVar.html"

- Class: figure
  Output: Here's the formula. You might have to stretch the plot window to get it
    displayed more clearly.
  Figure: plotdiffVar.R
  FigureType: new

- Class: text
  Output: Let's plug in the numbers from the blood pressure study to see how this
    works. Recall we have two groups, the first with size 8 and X'_{oc}=132.86 and
    s_{oc}=15.34 and the second with size 21 and X'_{c}=127.44 and s_{c}=18.23.

- Class: cmd_question
  Output: Let's compute the degrees of freedom first. Start with the numerator. It's
    the square of the sum of two terms. Each term is of the form s^2/n. Do this now and
    put the result in num. Our numbers were 15.34 with size 8 and 18.23 with size 21.
  CorrectAnswer: num <- (15.34^2/8 + 18.23^2/21)^2
  AnswerTests:  expr_creates_var('num'); omnitest(correctExpr='num <- (15.34^2/8 +
    18.23^2/21)^2',correctVal=2046.6418737445)
  Hint: Type  num <- (15.34^2/8 + 18.23^2/21)^2 at the command prompt.

- Class: cmd_question
  Output: Now the denominator. This is the sum of two terms. Each term has the form
    s^4/n^2/(n-1). These look a little different than the form displayed but they're
    equivalent. Put the result in the variable den. Our numbers were 15.34 with size 8
    and 18.23 with size 21.
  CorrectAnswer: den <- 15.34^4/8^2/7 + 18.23^4/21^2/20
  AnswerTests:  expr_creates_var('den'); omnitest(correctExpr='den <- 15.34^4/8^2/7 +
    18.23^4/21^2/20',correctVal=136.123536407433)
  Hint: Type  den <- 15.34^4/8^2/7 + 18.23^4/21^2/20 at the command prompt.

- Class: cmd_question
  Output: Now divide num by den and put the result in mydf.
  CorrectAnswer: mydf <- num/den
  AnswerTests:  expr_creates_var('mydf'); omnitest(correctExpr='mydf <- num/den')
  Hint: Type  mydf <- num/den at the command prompt.

- Class: cmd_question
  Output: Now with the R function qt(.975,mydf) compute the 95% t interval. Recall the
    formula. X'_{oc}-X'_{c} +/- t_df * SE. Recall that SE is the square root of the sum
    of the squared standard errors of the two means, (s_1)^2/n_1 + (s_2)^2/n_2 . Again
    our numbers are the following. X'_{oc}=132.86  s_{oc}=15.34  and n_{oc}=8 .
    X'_{c}=127.44  s_{c}=18.23  and n_{c}=21.
  CorrectAnswer: 132.86-127.44 +c(-1,1)*qt(.975,mydf)*sqrt(15.34^2/8 + 18.23^2/21)
  AnswerTests:  omnitest(correctExpr='132.86-127.44
    +c(-1,1)*qt(.975,mydf)*sqrt(15.34^2/8 + 18.23^2/21)')
  Hint: Type  132.86-127.44 +c(-1,1)*qt(.975,mydf)*sqrt(15.34^2/8 + 18.23^2/21) at the
    command prompt.
```

```
- Class: text
  Output: Don't worry about these nasty calculations. R makes things a lot easier. If you call t.test with var.equal set to FALSE, then R calculates the  degrees of freedom for you. You don't have to memorize the formula.


- Class: text
  Output: Congrats! You've concluded this rather t-dious lesson on all things t related - statistics, distributions, intervals. Hope you're not too teed off!

- Class: mult_question
  Output: "Would you like to receive credit for completing this course on
    Coursera.org?"
  CorrectAnswer: NULL
  AnswerChoices: Yes;No
  AnswerTests: coursera_on_demand()
  Hint: ""
```