Lab 9: Inference for Paired Data and Errors in Inference

Overview

In this lab, we use t-tests to perform inference for related samples (repeated measures). We also consider the data set provided to be the **entire population of interest**. Because we are considering the data set to be the entire population of interest, we know the true population distribution of the provided variables. We select random samples from this data, and for each sample we perform estimation (with a confidence interval) and testing (with a hypothesis test). Given that we know the true parameter values, we can assess the overall performance of the confidence intervals and hypothesis tests.

The Data: Course evaluations

End of the semester course evaluations are often criticized as indicators of the quality of the course and instructor because they can reflect biases such as the level of difficulty of the course and physical appearance of the instructor. This data set contains information on course evaluations on 94 randomly selected professors teaching in total 463 classes at the University of Texas at Austin. The unit of analysis for this data set is the 463 courses, so there are 463 observations. Because one professor can teach more than course, the observations for these professors are not truly independent. More complex statistical methods beyond this course would be more appropriate to analyze the data. For the sake of simplicity in QTM 100, please treat the observations as independent and proceed with the analytical tools you know.

The data set `CourseEvals.csv` contains 18 variables:

Variable	Description
`prof_id`	Professor ID
`class_id`	Class ID
`course_eval`	Average course evaluation: (1) very unsatisfactory-(5) excellent
`prof_eval`	Average professor evaluation: (1) very unsatisfactory - (5) excellent
`rank`	Rank of professor: teaching, tenure track, tenured
`ethnicity`	Ethnicity of professor: not minority, minority
`gender`	Gender of professor: 1=male, 2=female
`language`	Language of school where professor received education: English or non-English
`age`	Age of professor
`cls_perc_eval`	percent of students in class who completed evaluation
`cls_did_eval`	Number of students in class who completed evaluation
`cls_students`	Total number of students in the class
`cls_profs`	Number of sections professors teach in a course: single, multiple
`cls_credits`	Number of credits in a class: one credit, multi-credit
`bty_avg`	Average beauty score of professor among 6 raters: (1) lowest - (10) highest
`pic_outfit`	Outfit of professor in picture: not formal, formal
`pic_color`	Color of professor's picture: color, black and white

```
# Set working directory
setwd("~/Your Working Directory/")
```

```r
# Import the dataset
evals<-read.csv("CourseEvals.csv",header=TRUE)
```

Paired T-Test

In last week's manual, we investigated whether the university's claim that 80% of student completed the course evaluations was valid or not. What if, instead, we were interested whether professor evaluations are significantly different from course evaluations? In this case, we are looking at two scores that are about the same individual and are likely related. The two averages are not independent. In order to answer this question, we need to use a paired t-test. Other examples of this type of question would include looking at pre- and post- treatment scores (does a person's depression level significantly change post treatment, for example).

First, we need to calculate the difference between these two scores for each professor, and look at the average.

```r
# Calculate difference
evals$diff <- evals$course_eval-evals$prof_eval

# Look at summary statistics
## Mean
mean(evals$diff)

## Standard deviation
sd(evals$diff)

## Histogram of difference
hist(evals$diff)
```

The average difference is -0.18. The negative value indicates that the course evaluations tend to be lower than the professor evaluations. The histogram shows that the difference values are normally distributed. Next, we can use a t-test to evaluate whether this difference is statistically significant from 0. We can do this one of two ways:

```r
# Conduct paired t test using the difference variable
t.test(evals$diff)

# Conduct paired t test using the two evaluation variables
t.test(evals$course_eval,evals$prof_eval, paired=T)
```

Note the default null hypothesis of the one sample t-test is $H_0 : \mu_{diff} = 0$, which is the typical null hypothesis for the paired t-test. In the second method, the two quantitative variables are separated by a comma.

Both tests produce equivalent results - the second method internally creates a difference variable by subtracting the second variable from the first. The benefit of calculating the difference variable is that we can use it to see the distribution of the differences and get other summary statistics, like the standard deviation of the differences, that are not provided in the t-test output.

The test statistic is $t = -19.16$ with 462 degrees of freedom. At the $\alpha = 0.05$ level of significance, we reject the null hypothesis that the average difference in evaluations is 0 ($p < 0.001$); we conclude that the average difference is significantly different than zero. We are 95% confident that the true average difference is in the interval -0.19 to -0.16. Because this average difference is negative, this means that the average course evaluation is significantly lower than the average professor score ($\mu_{course} - \mu_{prof} < 0 = \mu_{course} < \mu_{prof}$). Therefore, there is statistically significant evidence that professors get higher personal evaluations than course evaluations.

Errors in Inference

The `evals` data set has 463 observations and 18 variables. For this lab, we consider these 463 individuals to be the entire population of interest (rather than a sample from the population).

This lab also requires utilizing pre-written `R` functions. These functions are available for download on Canvas, and look very similar to the `R` scripts that you utilize to complete labs for this class. There are two ways to import the functions you will need. Either way, you should download `TestingFunctions.R` from `Canvas` and save it somewhere you can remember.

The first way to import the functions is to open the file `TestingFunctions.R` in your `RStudio`. Highlight and run all of the code in this file to submit to the console. After it runs, you should see two functions in your `RStudio Environment` panel:

- `inference.means`—randomly selects samples from a given numerical variable and performs inference on that numerical variable.
- `plot.ci`—plots confidence intervals from an object created by `inference.means`.

The second way is using R code. The `source` command opens and runs the provided `R` Script for you, saving you the trouble of doing it yourself.

```
#Set working directory to the location of "Testing Functions" (if necessary)
setwd("~/Your Working Directory/")

#Run the Testing Functions file
source("TestingFunctions.R")
```

Identify the population distribution

Let's consider the variable `cls_perc_eval`. What is the true population distribution of this variable? To answer this, we should determine the shape of this distribution, the mean of this distribution, and the standard deviation of this distribution.

```
#Print histogram of course evaluation
hist(evals$cls_perc_eval)

#Find the mean and standard deviation
mean(evals$cls_perc_eval)
sd(evals$cls_perc_eval)
```

Among 463 courses, the percent completion is left skewed with an average percent of 74.4 and a standard deviation of 16.8.

Perform inference on multiple random samples from the population

Now let's take multiple random samples from this numerical variable, and perform inference on each sample. That is, for each sample, we will estimate the true population mean with a confidence interval. We can determine if the confidence interval actually captures the true mean value, which we know to be 74.4. If the confidence interval does not capture the true value, this is an error in estimation.

We will also perform a hypothesis test about μ. By default, the inference.means function tests the null hypothesis that the mean is equal to the true population value, versus the alternative that it is not. Hence, we are testing:

$H_0 : \mu = 74.4$ vs $H_a : \mu \neq 74.4$

The underlying assumption is that the null hypothesis is always true in reality, and thus we run the risk of committing a Type I error (rejecting H_0 when H_0 is true) when using the inference.means function. For each sample, we can determine if a Type I error was committed. The `inference.means` function has four arguments:

- `variable`—numerical variable of interest
- `sample.size`—the sample size n
- `alpha`—the level of significance (used both for confidence intervals and testing)
- `num.reps`—the number of random samples to generate

Let's take 100 samples of size $n = 50$, and perform inference at the $\alpha = 0.05$ level of significance. First, we store the inferential results in `sim1`, which represents results from "simulation 1". Then we type `sim1` to view the results.

```
#Store the results of the inference in a new data frame
sim1 <- inference.means(variable = evals$cls_perc_eval,
                        sample.size = 50,
                        alpha = 0.05,
                        num.reps = 100)

#View results
View(sim1)
```

The simulation results produces a data frame with 7 columns and 100 rows (one row for each of the 100 samples drawn and tested).

1. `samp.est`—the point estimate from the sample of size $n = 50$ (this is the sample mean).
2. `test.stat`—the t-statistic calculated for $H_0 : \mu = 74.4$ vs $H_a : \mu \neq 74.4$.
3. `p.val`—the p-value calculated for $H_0 : \mu = 74.4$ vs $H_a : \mu \neq 74.4$.
4. `decision`—the decision made ($p \leq \alpha$ reject H_0; otherwise, fail to reject H_0).
5. `lcl`—the lower bound of the confidence interval estimating μ (lower confidence limit).
6. `ucl`—the upper bound of the confidence interval estimating μ (upper confidence limit).
7. `capture`—indicates if the confidence interval captured the true parameter value $\mu = 74.4$.

Assess assumptions for inference

When performing inference about a mean, we have three assumptions to assess.

1. The data represent a random sample from the population. The function `inference.means` randomly selects observations from the population of 463 observations, so this assumption is satisfied.
2. All observations are independent. Because the function `inference.means` randomly selects observations from the population of 463 observations, this assumption is also satisfied.
3. The sampling distribution of the sample mean is approximately normally distributed. This is the only assumption you need to formally assess. Although the underlying population of percent completion is left skewed, the sampling distribution of the sample mean of percent completion would be normally distributed because we have a large enough sample size.

In this case, conditions are satisfied for valid inference. This means that we expect the inferential methods to perform according to the specified level of significance. That is, approximately 95% of confidence intervals

should capture the true mean $\alpha = 74.4$ and approximately 5% of tests should commit a Type I error (reject H_0 even though H_0 is true).

When conditions are not satisfied for valid inference our inferential methods may not perform according to the specified level of significance. That is, we may have more or less than 95% of confidence intervals that capture the true mean and more or less than 5% of tests could commit a Type I error (reject H_0 even though H_0 is true).

When $\alpha = 0.05$, the hypothesis test has a targeted Type I error rate of 5% and the confidence interval has a targeted capture rate of 95%. When assumptions are violated, we may see deviations from these targeted rates. This is what it means to have invalid inference—our hypothesis test or confidence interval is not performing as expected based on the targeted rate.

Examine performance of hypothesis testing

Now examine the inferential results related to the hypothesis test by visualizing the distribution of the sample means, the test statistics, and the p-values.

```
#Histogram of sample means
hist(sim1$samp.est, main = "Sample Means")

#Histogram of test statistics
hist(sim1$test.stat, main = "t test statistics")

#Histogram of p-values
hist(sim1$p.val, main = "p-values")
```

The histogram of your sample means should be approximately normally distributed (this represents the sampling distribution of the sample mean) because we already determined that this assumption was satisfied. When the sampling distribution assumption is satisfied, the test statistic will also be approximately normally distributed, and the p-value will be approximately uniformly distributed.

In how many instances did we commit a Type I error? This occurs when we erroneously reject $H_0 : \mu = 74.4$. We can calculate this by looking at a frequency table showing the distribution of the decisions made.

```
#Frequency table of decisions
table(sim1$decision)
```

In this case, 4 of my 100 tests rejected H_0, indicating an observed Type I error rate of 4%. This is pretty close to the targeted level (5%). `Due to the random nature of the simulation, your results may appear different than mine`.

Examine performance of confidence interval estimation

Now examine the inference results related to confidence interval estimation by visualizing the confidence intervals with the `plot.ci` function. This function takes two arguments:

- `results`—the name of the object that contains the simulation results from `inference.means`;
- `true.val`—the true value of the parameter being tested.

In this case, our simulation results are stored in the object `sim1`; the true value of the population mean is $\mu = 74.4$.

```
#Generate plot of confidence intervals
plot.ci(results = sim1, true.val = 74.4)
```

Here, we can see four confidence intervals in red that do not actually capture the true parameter value of $\mu = 74.4$, which represent an error in estimation. These four intervals actually correspond to the same samples where we committed a Type I error. In these four instances, we happened to observe sample means which were further away from the population mean, leading us to commit an error. Because 96 out of 100 intervals did actually capture the true parameter value, we can say that our observed confidence level is 96%, which is pretty close to the targeted level of 95%.

You can also obtain these results numerically rather than visually by looking at a frequency table showing the distribution of whether or not the true parameter value was captured.

```
#Frequency table displaying number of times the true mean was captured
table(sim1$capture)
```

This reinforces what we observed—that 96 out of the 100 intervals captured the true parameter value of $\mu = 74.4$.

Agreement between the confidence interval and the hypothesis test

You can more directly explore the relationship between confidence interval estimation and hypothesis testing by looking at a contingency table showing the relationship between whether or not the confidence interval captured the true parameter and whether we rejected or failed to reject the null hypothesis.

```
table(sim1$capture, sim1$decision)
```

Although my results may appear different than yours, you should see something similar. My results show that there were 96 samples in which the confidence interval captured the true parameter $\mu = 74.4$ and in which we failed to reject the null hypothesis ($H_0 : \mu = 74.4$). These results agree with each other as both support that 74.4 is a plausible value for μ. We can also see that there are 4 instances in which an error was committed; that is, in four samples, we erroneously rejected the H_0 and the confidence interval did not capture $\mu = 74.4$. These results are also in agreement because they both support the incorrect inference that 74.4 is not a plausible value for μ.

Lastly, there are two combinations which have zero entries. When performing a hypothesis test about a mean it is not possible to have confidence interval results and hypothesis test results that do not agree, such as a hypothesis test that fails to reject H_0 and a confidence interval that does not capture μ, or a hypothesis test which rejects H_0 but where the corresponding confidence interval does capture μ. Both of these situations have instances in which an inferential error is made by one method and a correct conclusion is made by the other—these results are in disagreement and do not occur for inference about a mean.

Long run performance

Examining the inferential results from 100 random samples can give us a good idea if the test is behaving as it should. However, the results will not be definitive since we could reasonably expect some variation in the 100 samples. Increasing the number of random samples to generate and test can give us a better idea of the long run performance of the test. Here, we run the simulation 10000 times.

```r
# Run simulation 10000 times
sim1long <- inference.means(variable = evals$cls_perc_eval,
                            sample.size = 50, alpha = 0.05,
                            num.reps = 10000)

#Frequency table of captures
table(sim1long$capture)

#Frequency table of decisions
table(sim1long$decision)
```

My results show that 491 out of 10000 samples erroneously rejected the null hypothesis, which gives an observed type I error rate of 491/10000, or 4.91%. This is close to the targeted Type I error rate of 5%.

Similarly, 9509 out of 10000 samples produced confidence intervals that captured the true parameter value. This gives an observed confidence interval coverage of 9509/10000, or 95.09%. This is close to the targeted confidence level of 95%.

These results indicate that the hypothesis test is committing errors at the targeted level and the confidence interval is capturing the true parameter value at the targeted level. This indicates that both inferential methods are behaving as expected, and inferential results are valid. If we had observed deviations from this, even if they are small, like an observed Type I error rate of 7% and an observed confidence interval coverage of 93%, this would indicate that the test is performing worse than the targeted levels and that inferential results are not valid.