Lab 7: Inference for a Single Mean and Errors in Inference

Part 1: Inference for a Single Mean

In this part of the lab we use the one sample t-test to perform inference for a single mean. We also use the *t* distribution to (1) calculate p-values based on t test statistics, and (2) calculate t-scores used in confidence intervals for specific confidence levels.

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[1]. This data set includes evaluations on the same professors, and therefore the observations 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:

`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 class
`cls_level`	class level: lower, upper
`cls_profs`	number of sections professors teach in a course: single, multiple
`cls_credits`	number of credits of 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

Exploring the data

Researchers are concerned about the validity of the results due to non-response because many students choose not to submit end of the semester evaluations. The university administration claims that there is an overall 80% response rate in course evaluations. Let's begin by exploring the `cls_perc_eval` variable.

```
evals<-read.csv("C:/Users/smcclin/Documents/Labs/CourseEvals.csv",header=TRUE)
library(mosaic)
favstats(evals$cls_perc_eval)
hist(evals$cls_perc_eval)
```

[1]Source: Hamermesh, Daniel and Parker, Amy. (2005). "Beauty in the classroom: instructors' pulchritude and putative pedagogical productivity." *Economics of Education Review, 24(4): 369-376.*

Among 463 courses, the percent completion is right skewed with an average percent of 74.4 and a standard deviation of 16.8. In order to investigate if the average percent completion is statistically significantly different from 80%, we need to perform a one sample t-test.

One-sample t-test

Our hypotheses are H_0: $\mu = 80$ vs H_a: $\mu \neq 80$, where μ is the true average percent completion. Although the data are right-skewed, the sample size is large ($n = 463$) enough for conditions for valid inference to be satisfied. The one sample t-test can be performed with the `t.test` command.

```
t.test(evals$cls_perc_eval,mu=80)
```

The first argument is the quantitative variable being tested, and the second argument provides the value tested in the null hypothesis. The `t.test` returns results for both the hypothesis test and the confidence interval. The default settings are to perform a two-sided alternative hypothesis and to calculate a 95% confidence interval.

The test statistic is $t = 5.7$ with 462 degrees of freedom. At the $\alpha = 0.05$ level of significance, we reject H_0 ($p < 0.001$) and conclude that the true mean percent completion is significantly lower than 80%. We are 95% confident that the true mean percent completion is in the the interval 72.9 to 76.0 percent. The university is achieving lower than the claimed average completion rate of 80%.

You can add additional arguments to the `t.test` function to change the form of the alternative hypothesis or the confidence level. For example, to calculate a 90% confidence interval use the `conf.level` argument.

```
t.test(evals$cls_perc_eval,mu=80,conf.level=0.90)
```

When specifying the confidence level, you must input a number between 0 and 1. To test H_0: $\mu = 80$ vs H_a: $\mu < 80$ (a one-sided less than alternative hypothesis) use the `alternative` argument.

```
t.test(evals$cls_perc_eval,mu=80,alternative="less")
```

Note that when testing a one-sided alternative, the confidence interval has a lower bound of `-Inf` for a less than alternative, and an upper bound `Inf` for a greater than alternative.

The t distribution

Just as we used `pnorm` and `qnorm` to calculate probabilities and quantiles from the normal distribution, we can use `pt` and `qt` to calculate probabilities and quantiles from the t distribution. The probabilities can be used to calculate p-values based on a test statistic, and the quantiles can be used to idenfity t-scores for confidence intervals of a certain confidence level. By default, the t functions utilize lower tail areas.

Suppose we performed a one-sample t-test with a two-sided H_a with 50 degrees of freedom and a test statistic of $t = -2$. The p-value for this test would be given by twice the *lower* tail area under the curve. The first argument is the test statistic, and the second argument is the degrees of freedom.

```
2*pt(-2,df=50)
```

This function calculates the area under the curve less than -2 for a t distribution with 50 degrees of freedom, and then multiplies that value by 2 to yield a p-value for a two-sided H_a of 0.0509.

If we performed a one-sample t-test with a two-sided H_a with 50 degrees of freedom and a test statistic of $t = 2$, the p-value for this test would be given by twice the *upper* tail area under the curve. To calculate the upper tail area, we need to take the complement of the lower tail area.

```
2*(1-pt(2,df=50))
```

Alternatively, you can set the argument `lower.tail` to `FALSE` to calculate an upper tail area and avoid having to take the complement.

```
2*pt(2,df=50,lower.tail=F)
```

Because the t distribution is symmetric, a test statistic of positve two or negative two yields the same p-value of 0.0509.

Use the `qt` function to identify a t-score for a specific confidence interval. To do this, we first need to identify the appropriate area under the curve that corresponds to the specific confidence level. For a 95% confidence interval, this would correspond to a lower tail area under the curve of 0.025. In general, for a specified α, use $\alpha/2$ as the lower tail area under the curve to calculate the t-score. Remember, the quantile function calculates a value that corresponds to a lower tail under the curve.

```
qt(0.025,df=50)
```

Given 50 degrees of freedom, the quantile that corresponds to the 2.5^{th} percentile is -2.01. We report the absolute value of this quantity - the t-score used to calculate a 95% confidence interval with 50 degrees of freedom is 2.01. Equivalently, you could also calculate the 97.5^{th} percentile to yield the positive t-score.

```
qt(0.975,df=50)
```

Part 2: Errors in Inference

In this part of the lab, **we will 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: Youth Risk Behavior Surveillance System

The Youth Risk Behavior Surveillance System (YRBSS) has been conducted every two years since 1991 by the Centers for Disease Control and Prevention (CDC) in order to obtain information from adolescents regarding trends in risky behavior, such as smoking, drinking, drug use, diet, and physical activity. In 2013, 47 states participated in this school-based survey, yielding 13,583 respondents and 213 variables. Full survey and data documentation can be accessed on the CDC website. A subset of this data set which has no missing data for 17 selected variables is provided in the file `yrbss2013.csv`[2].

`age`	*Q1: How old are you?*
`gender`	*Q2: What is your sex?*
`height_m`	calculated variable: height in meters
`weight_kg`	calculated variable: weight in kilograms
`bmi`	calculated variable: body mass index=`weight_kg`/`height_m`^{2}
`BMIPCT`	calculated variable: BMI percentile for age and sex
`seatbelt`	*Q9: How often do you wear a seat belt when riding in a car driven by someone else?*
`seatbelt2`	calculated variable: `seatbelt` never vs otherwise
`ride_drunkdriver`	*Q10: During the past 30 days, have you ridden in a car or other vehicle driven by someone who had been drinking alcohol?*
`drive_drunk`	*Q11: During the past 30 days, how many times did you drive a car or other vehicle when you had been drinking alcohol?*
`drive_text`	*Q12: During the past 30 days, on how many days did you text or e-mail while driving a car or other vehicle?*
`carried_weapon`	*Q13: During the past 30 days, did you carry a weapon such as a gun, knife, or club?*
`unsafe_school`	*Q16: During the past 30 days, did you not go to school because you felt you would be unsafe at school or on your way to or from school?*
`bullied`	*Q24: During the past 12 months, have you ever been bullied on school property?*
`sad`	*Q26: During the past 12 months, did you ever feel so sad or hopeless almost every day for two weeks or more in a row that you stopped doing some usual activities?*
`days_smoke`	*Q33: During the past 30 days, on how many days did you smoke cigarettes?*
`days_drink`	*Q43: During the past 30 days, on how many days did you have at least one drink of alcohol?*

Getting started

Import the `yrbss2013.csv` data set into R. Begin by examining the overall data set.

```
yrbss<-read.csv("C:/Users/smcclin/Documents/Labs/Lab8 Errors in inference/yrbss2013.csv",header=T)
str(yrbss)
summary(yrbss)
```

[2] The variables `days_smoke` and `days_drink` were originally coded in categories of '0 days', '1 or 2 days', '3 to 5 days', '6 to 9 days', '10 to 19 days', '20 to 29 days', and 'All 30 days'. The number of days provided in this data set was randomly generated according to the category specified.

The yrbss data set has 8,482 observations and 17 variables. For this lab, we consider these 8,482 individuals to be the entire population of interest (rather than a sample from the population).

This lab also requires utilizing pre-written R functions, like the cereal simulation in Lab 1. To get started with this lab, you should have downloaded TestingFunctions.R from Blackboard. Submit the functions contained within the script TestingFunctions.R to the RStudio console by going to

1. File → Open File → TestingFunctions.R.
2. Hightlight and run *all* code in this file to submit to the console. You should see the following files in your workspace in the upper right panel:
   - inference.means - randomly selects samples from a given quantitative variable and performs inference on that quantitative variable
   - plot.ci - plots confidence intervals from an object created by inference.means
3. Close TestingFunctions.R by clicking on the "x" next to the file name.

Identify the population distribution

Let's consider the variable height_m. 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.

```
hist(yrbss$height_m)
library(mosaic)
favstats(yrbss$height_m)
```

The population of height in meters appears to be approximately normally distributed with a mean of 1.687 meters and a standard deviation of 0.10 meters.

Perform inference on multiple random samples from the population

Now let's take multiple random samples from this quantitative variable, and perform inference on each sample. That is, for each sample we will estimate the true population mean μ with a confidence interval. For each sample, we can determine if the confidence interval actually captures the true mean value, which we know to be 1.687. 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: $\mu = 1.687$ vs H_a: $\mu \neq 1.687$

When using the inference.means function, 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). For each sample, we can determine if a Type I error was committed. The inference.means(variable, sample.size, alpha, num.reps) function has four arguments:

- variable - quantitative variable of interest
- sample.size - the sample size n
- alpha - the level of significance (used both for confidence intervals and testing)
- num.reps - the numbers 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.

```
sim1<-inference.means(variable=yrbss$height_m, sample.size=50, alpha=0.05, num.reps=100)
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 test statistic calculated for H_0: $\mu = 1.687$ vs H_a: $\mu \neq 1.687$
3. `p.val` - the p-value calculated for H_0: $\mu = 1.687$ vs H_a: $\mu \neq 1.687$
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 $\mu = 1.687$

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 8,482 observations, so this assumption is satisfied.

2. *All observations are independent.*

   Because the function `inference.means` randomly selects observations from the population of 8,482 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. Because the underlying population of height is approximately normally distributed, the sampling distribution of the sample mean height should also be normally distributed (regardless of 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 $\mu = 1.687$ 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.

```
hist(sim1$samp.est,main="Sample Means")
hist(sim1$test.stat,main="t test statistics")
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: $\mu = 1.687$. We can calculate this by looking at a frequency table showing the distribution of the decision made.

```
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 either `inference.means` or `inference.proportions`
- `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 parameter being tested is $\mu = 1.687$.

```
plot.ci(results=sim1,true.val=1.687)
```

Here, we can see four confidence intervals in red that do not actually capture the true parameter value of $\mu = 1.687$, 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 frequence table showing the distribution of whether or not the true parameter value was captured.

```
table(sim1$capture)
```

This reinforces what we observed - that 96 out of the 100 intervals captured the true parameter value of $\mu = 1.687$.

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 $\mu = 1.687$ and in which we failed to reject the null hypothesis (H_0: $\mu = 1.687$). These results agree with each other as both support that 1.687 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 $\mu = 1.96$. These results are also in agreement because they both support the incorrect inference that 1.687 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

Examing 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.

```
sim1long<-inference.means(variable=yrbss$height_m, sample.size=50, alpha=0.05,
                          num.reps=10000)
table(sim1long$capture)
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.