Lab 8: Sampling Distribution of the Mean and Inference for a Single Mean

Part 1 Overview: Sampling Distribution of the Mean

In Lab 5, we investigated the ways in which the statistics from a random sample of data can serve as point estimates for population parameters. We were interested in formulating a *sampling distribution* of our estimate in order to learn about the properties of the estimate, such as its distribution.

In Lab 5, we looked at a categorical variable in the data set (bullied). In this Lab, we will consider a numerical variable in our dataset `days_drink` which focuses on how many days the students had at least one drink of alcohol.

The data

Every two years the CDC conducts national surveys in schools to monitor and assess the six largest contributors to youth morbidity and mortality. These contributors include not only health risks such as high body mass index, but also risky behaviors such as tobacco and alcohol use, as well as drunk driving and failure to use seatbelts. 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 16 selected variables is provided in the file yrbss2013.csv.

Variable	description
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=height m/(weight kg)^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?

Let's load the data.

```
#Set working directory
setwd("location of working directory")

#Import the data set
yrbss <- read.csv("yrbss2013.csv", header = T)
```

Sampling Distribution of the Mean

Let's look at the distribution of days the students drank at least one drink of alcohol by calculating a few summary statistics and making a histogram.

```
days_drink <- yrbss$days_drink

#View summary statistics
summary(days_drink)
hist(days_drink)
```

(Do it yourself) Describe this population distribution.

Similar to the ideas carried out in Lab 5 with sampling distribution for proportions, we will not always have access to the entire population, so we can obtain estimates of parameters such as the mean based on random samples.

```
samp_dd1 <- sample(x = days_drink, size = 50)
```

(Do it yourself) Describe the distribution of this sample. How does it compare to the population distribution?

Let's estimate the average days students drank at least one drink of alcohol in the sample.

```
mean(samp_dd1)
```

Similar to what we observed for the sampling distribution for proportions, depending on which 50 students you selected, your estimate could be a bit above or a bit below the true population mean of 1.454 days.

(Do it yourself) Take a second sample, also of size 50, and call it samp_dd2. How does the mean of samp_dd2 compare with the mean of samp_dd1? Suppose we took two more samples, one of size 100 and one of size 1000. Which would you think would provide a more accurate estimate of the population mean?

Now, since we know that every time we take another random sample, we get a different sample mean, let us write a for-loop to take 50000 samples of size 50 from the population, calculate the mean of each sample, and store each result in a vector called sample_means50.

```
sample_means50 <- rep(NA, 50000)
#Creates an empty vector of 50000 lines

for(i in 1:50000){
   samp_d <- sample(days_drink, 50)
   #Creates a vector with 50 values from the "days_drink" vector

   sample_means50[i] <- mean(samp_d)
   #Adds the mean of samp to the sample_means vector
```

```
  }

hist(sample_means50)
```

(Do it yourself): How many elements are in sample_means50? Describe the sampling distribution and be sure to specifically note its center. Would you expect the distribution to change if we instead collected 500,000 sample means?

Let's consider this code line by line to figure out what it does.

1. `sample_means50 <- rep(NA, 50000)` *initialized a vector*. We created a vector of 50000 zeros called `sample_means50`. This vector will will store values generated within the `for` loop.
2. `for(i in 1:50000){` calls the `for` loop itself and "for every element `i` from 1 to 50000, run the following lines of code".
3. `samp_d <- sample(days_drink, 50)` uses the sample function to draw a sample of size 50 from the `days_drink` variable. Then it assigns this sample to a variable samp_d, which is then a vector with 50 values.
4. `sample_means50[i] <- mean(samp_d)` computes the mean of samp_d and saves the value as i-th element of the vector sample_means50.
5. `}` indicates the end of the code to be repeated 50,000 times.
6. `hist(sample_means50, breaks=25)` produces a histogram of sample_means50 using break size of 25.

(Do it yourself) To make sure you understand what you've done in this loop, try running a smaller version. Initialize a vector of 100 zeros called `sample_means_small`. Run a loop that takes a sample of size 50 from `days_drink` and stores the sample mean in `sample_means_small`, but only iterate from 1 to 100. Print the output to your screen. How many elements are in the object `sample_means_small`? What does each element represent?

Sample size and the sampling distribution

Mechanics aside, let's return to the reason we used a `for` loop: to compute a sampling distribution, specifically, to look at the approximate sampling distribution of the sample means:

```
hist(sample_means50)
```

The sampling distribution that we computed tells us much about estimating the average days students drank at least one drink of alcohol. Because the sample mean is an unbiased estimator, the sampling distribution is centered at the true average of `days drink` of the the population, and the spread of the distribution indicates how much variability is induced by sampling only 50 of the data points.

To get a sense of the effect that sample size has on our distribution, let's build up two more sampling distributions: one based on a sample size of 10 and another based on a sample size of 100.

```
#Initialize vectors of 50000, one for sample means of 10 and of 100
sample_means10 <- rep(NA, 50000)
sample_means100 <- rep(NA, 50000)

#Run for loop, drawing samples of 10 and 100
for(i in 1:50000){
```

```
  samp <- sample(days_drink, 10)
  sample_means10[i] <- mean(samp) #assign means of sample size 10 to the vector
  samp <- sample(days_drink, 100)
  sample_means100[i] <- mean(samp) #assign means of sample size 100
}
```

Here we're able to use a single `for` loop to build two distributions by adding additional lines inside the curly braces. Don't worry about the fact that `samp` is used for the name of two different objects. In the second command of the `for` loop, the mean of `samp` is saved to the relevant place in the vector `sample_means10`. With the mean saved, we're now free to overwrite the object `samp` with a new sample, this time of size 100. In general, anytime you create an object using a name that is already in use, the old object will get replaced with the new one.

To see the effect that different sample sizes have on the sampling distribution, plot the three distributions on top of one another.

```
par(mfrow = c(3, 1))

xlimits <- range(sample_means10)

hist(sample_means10, xlim = xlimits)
hist(sample_means50, xlim = xlimits)
hist(sample_means100, xlim = xlimits)
```

The first command specifies that you'd like to divide the plotting area into 3 rows and 1 column of plots (to return to the default setting of plotting one at a time, use `par(mfrow = c(1, 1))`). The `breaks` argument specifies the number of bins used in constructing the histogram. The `xlim` argument specifies the range of the x-axis of the histogram, and by setting it equal to `xlimits` for each histogram, we ensure that all three histograms will be plotted with the same limits on the x-axis.

(Do it yourself): When the sample size is larger, what happens to the center? What about the spread?

Part 2 Overview: Inference for a Single Mean

In Part 2 of this lab we use t-tests 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. 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:

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

One Sample T-Test

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. Is that true? Let's begin by exploring the `cls_perc_eval` variable, which shows the percent of students in the class who completed an evaluation.

```
#Set working directory
setwd("YOUR/WORKING/DIRECTORY/")
# Import the dataset
evals<-read.csv("CourseEvals.csv",header=TRUE)

#Create histogram of the cls_perc_eval variable
hist(evals$cls_perc_eval)

# View summary statistics
summary(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. 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 left-skewed, the sample size (n = 463) is large enough for conditions for valid inference to be satisfied. The one sample t-test can be performed with the t.test command.

```
#Perform one-sample t-test
t.test(evals$cls_perc_eval,mu=80)
```

The first argument is the numerical 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 test and to calculate a 95% confidence interval.

The test statistic is t=-7.2 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 evaluating if the true mean of evaluations is different from 80, at the 90% confidence level
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 $Ha : \mu < 80$ (a one-sided less than alternative hypothesis) use the `alternative` argument.

```
# One-sided t-test evaluating likelihood that evaluation percent is less than 80
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 identify 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 Ha of 0.0509. If we performed a one-sample t-test with a two-sided Ha 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 positive 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 2.5th 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 97.5th percentile to yield the positive t-score.

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

Ideas for this manual were obtained from the lab work written for OpenIntro by Andrew Bray and Mine Çetinkaya-Rundel.