Homework 7: The one-sample t-test

NAME: Andres Castano

NETID: ac986

DUE DATE: November 1, 2016 by 1:00pm

For this homework, it will be helpful to have a copy of the knitted version of this document to answer the questions as much of it is written using mathematical notation that may be difficult to read when the document is not knitted.

Instructions

For this homework:

1. All calculations must be done within your document in code chunks. Provide all intermediate steps.
2. Incude any mathematical formulas you are using for a calculation. Surrounding mathematical expresses by dollar signs makes the math look nicer and lets you use a special syntax (called latex) that allows for Greek letters, fractions, etc. Note that this is not R code and therefore should not be put in a code chunk. You can put these immediately before the code chunk where you actually do the calculation.
3. To help you solve these problems, you will find it helpful to draw a picture (as was done in Lab 8). Please do so by hand on paper, but do not worry about submitting this picture. (You will indirectly get credit for having drawn the picture by being more likely to have the right answer!)
4. Be sure you have read Lab 8 carefully before starting this homework.

Hypothesis Testing for μ Using a t-distribution

Review: When can we treat $\bar{X}_n$ as normal?

In Lab 8, we saw that hypothesis testing for μ can be based on the distribution of $\bar{X}_n$. In the lab it was assumed that $\bar{X}_n \sim N(\mu, \sigma/\sqrt{n})$. Generally, this assumption can be made at least approximately when the X_i's are indepedent and one of the following conditions holds:

1) The sample size, n, is greater than 30 and the distribution isn't strongly skewed. In that case, the Central Limit Theorem applies. (For the case of sampling from a Bernoulli distribution, we have a more specific rule, namely that $n\hat{p}$ and $n(1-\hat{p})$ should both be greater than or equal to 10.)
2) The sample, $X_i, i = 1, \ldots, n$, is iid from $N(\mu, \sigma)$.

Doing inference when $\bar{X}_n \sim N(\mu, \sigma/\sqrt{n})$

If the population standard deviation σ is known, we can do tests (and form confidence intervals) based on the fact that, at least approximately,

$$\frac{\bar{X}_n - \mu}{\sigma/\sqrt{n}} \sim N(0,1).$$

This worked fine in Lab 8 because we were told σ. But typically we don't know σ. This is the *identical* situation as with confidence intervals, so this is a good time for you to review Lecture 13. The idea is to replace σ by S_n, the standard deviation of the sample.

If n is sufficiently large, we expect S_n to be so close to σ that we can just assume $\frac{\bar{X}_n - \mu}{S_n/\sqrt{n}}$ is approximately $N(0,1)$, and we can simply operate as in Lab 8.

If n is small, however, we have to account for the variability introduced by S_n, and we will find that

$$\frac{\bar{X}_n - \mu}{S_n/\sqrt{n}} \text{ is not } N(0,1).$$

Do we know what the distribution of $\frac{\bar{X}_n - \mu}{S_n/\sqrt{n}}$ is when n is small? The answer, unfortunately, is "no" in general **except for the special case that $X_i, i = 1, \ldots, n$, iid from $N(\mu, \sigma)$.**

If $X_i, i = 1, \ldots, n$, iid from $N(\mu, \sigma)$, then for **any** $n > 1$,

$$\frac{\bar{X}_n - \mu}{S_n/\sqrt{n}} \sim t_{n-1}.$$

That is, this random variable follows a t-distribution with $n - 1$ degrees of freedom.

We will still consider the three alternative hypotheses presented in Lab 8 and always take H_0 to be $\mu = \mu_0$.

Problem 1

A study was conducted of a simple random sample of 15 adult male patients following a new treatment for congestive heart failure. One of the variables measured on the patients was the increase in exercise capacity (in minutes) over a 4-week treatment period. Assume this variable is normally distributed. The previous treatment regime had produced an average of $\mu = 2$ minutes. The researchers wanted to evaluate whether the new treatment had a different impact on the increase in exercise capacity than the old treatment. The data can be found in the *Heart.csv* file.

a) What are the null and alternative hypotheses of this study?

The null hypotheses is $H_0 : \mu = 2$ and the alternative hypotheses $H_a : \mu \neq 2$

b) What is the distribution of the standardized sample mean, $\frac{\bar{X}_n - \mu_0}{S_n/\sqrt{n}}$, under the null hypothesis for this study?

Given that we do not know the population standard deviation σ we need to account for the aditional variability when we deciding to use S_n, then the standarized sample mean is distributed according to t with 14 degrees of freedon:

$$\frac{\bar{X}_n - \mu}{S_n/\sqrt{n}} \sim t_{n-1}$$

Rejection Regions and p-Values Using the t-distribution

Look back at Steps 1 – 4 at the bottom of page 1 of Lab 8. For Step 2, we used that $\bar{X}_n \sim N(\mu_0, \sigma/\sqrt{n})$ under H_0, which is equivalent to

$$\frac{\bar{X}_n - \mu_0}{\sigma/\sqrt{n}} \sim N(0,1).$$

When σ is unknown and when it is valid to use a t-distribution (see above), we replace the above with the fact that

$$\frac{\bar{X}_n - \mu_0}{S_n/\sqrt{n}} \sim t_{n-1}$$

under H_0. Other than that, all the same logic used in Lab 8 applies. Rather than using the quantiles of $N(0,1)$ you use the corresponding quantiles of t_{n-1}.

Likewise at the bottom of page 5, Steps 1 – 4 still apply, just Step 2 has a different null distribution.

Problem 2

a) Using a code chunk, import the data set *Heart.csv* into the workspace of this document (look back at Lab 6 if you need help). Make sure it is read in correctly.

```r
heart_data <- read.csv("Heart.csv")
dim(heart_data)
```

```
## [1] 15  1
```

b) Assuming the significance level of the test is 0.05, what is the rejection region of the test described in Problem 1?

The rejection region is defined by $\frac{\bar{x}_n-\mu_0}{s/\sqrt{n}} < t_{14,0.025}$ or if $\frac{\bar{x}_n-\mu_0}{s/\sqrt{n}} > t_{14,0.975}$. We construct this rejection using the quantiles of the t distribution as follows:

```r
n=15
df=n-1
lower = qt(0.025,df)
upper = qt(0.975,df)
lower
```

```
## [1] -2.144787
```

```r
upper
```

```
## [1] 2.144787
```

Under the null the rejection region is defined by $\frac{\bar{x}_n-\mu_0}{s/\sqrt{n}} < -2.144787$ or if $\frac{\bar{x}_n-\mu_0}{s/\sqrt{n}} > 2.144787$. A easier way to put this is: we are going to reject H_0 if $|\frac{\bar{x}_n-\mu_0}{s/\sqrt{n}}| > 2.144787$.

c) Compute the statistic, $\frac{\bar{x}_n-\mu_0}{s/\sqrt{n}}$, needed to perform this test using the variable `Increase` from the `Heart` data.

```r
mean_sample=mean(heart_data$Increase)
mean_sample
```

```
## [1] 2.466667
```

```r
sd_sample=sd(heart_data$Increase)
sd_sample
```

```
## [1] 0.8138679
```

```r
mu0=2
n=15
observed_statistic=(mean_sample-mu0)/(sd_sample/sqrt(n))
observed_statistic
```

```
## [1] 2.220744
```

d) Based on the rejection region, should we reject H_0?

As we can observed the statistic $|\frac{\bar{x}_n-\mu_0}{s/\sqrt{n}}| = 2.220744$ is greater than 2.144787, then we have statistical evidence to reject H_0. In particular the mean of the increase in exercise capacity in minutes with the new treatment is different compared with the old treatment.

Problem 3

a) What is the p-value of the test described in Problem 1?

The p-value is the probability under the null hypothesis of seeing something as extreme or more extreme than what was actually observed. In our case we observed the statistic $\frac{\bar{x}_n-\mu_0}{s/\sqrt{n}} = 2.2207$, the p-value is:

```r
n=15
df=n-1
pvalue = 2*(1-pt(observed_statistic,df)) # given that is a two tail case.
pvalue
```

```
## [1] 0.04337819
```

b) Should we reject H_0 at the 0.05 level based on this p-value?

At a significance level $(\alpha) = 0.05$=5% we should reject H_0 given that p-value is less than α (0.043378<0.05). However, if we decide to define the testing at $\alpha = 0.01$ we do not have statistical evidence to reject H_0.

Power calculation for a t-test

When the study described above was being planned, the researchers wanted to make sure that if the new treatment were to lead to a (population-level) $\mu = 3$ minute increase in exercise, that the study would be able to reject the null that $\mu = 2$ in favor of the alternative $\mu \neq 2$. In particular, they wanted to choose a sample size n large enough so that there would be a 90% probability that the study would reject the null hypothesis when $\mu = 3$.

They did a Monte Carlo study to decide what sample size would be required. In particular, they simulated data like the kind they expected to collect: $X_1, \ldots, X_n \sim N(\mu, \sigma)$. They took $\mu = 3$ since they wanted to know about a probability in the case that $\mu = 3$. From the previous literature, they guessed that $\sigma = 1$ minute would be a reasonable value. This was a big guess, but they figured it'd be better to have a rough sense of n required than to just pick a sample size completely arbitrarily.

Problem 4

a) Write a code chunk that does the following two things:

- draws a realization of a sample of size n from the distribution described above (for now, your first line of code can set `n = 10`).
- does exactly what was done in Problem 2 for parts (b)-(d) but on the sample of simulated data instead of on the actual data (don't forget that the quantile of the t-distribution will change since n is different). For part (d), the necessary comparison should be done using logic operations in R – the output of the code should be `TRUE` if the test rejects H_0 in favor of H_a and `FALSE` if the test fails to reject H_0.

```
n=10
df=n-1
simulated = rt(1, df)
simulated
```

```
## [1] -1.789646
```

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

```
## [1] -2.262157
```

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

```
## [1] 2.262157
```

```
(simulated <= lowercutoff) | (simulated >= uppercutoff)
```

```
## [1] FALSE
```

As in 2b, We reject H_0 when $\frac{\bar{x}_n-\mu_0}{s/\sqrt{n}}$ is either larger than or equal to the 97.5th percentile of $t_{9,0.975}$ or smaller than or equal to the 2.5th percentile of $t_{9,0.025}$. This ensures that the probability that $\frac{\bar{x}_n-\mu_0}{s/\sqrt{n}}$ is in the rejection region when H_0 is true is at most 0.05.

b) To perform a Monte Carlo simulation, we want to repeat (a) a very large number of times (perhaps `nsim = 100000`) and calculate the proportion of times the test rejects. Write a code chunk with a `for` loop that does this. Hint: It may help to look back at past examples of Monte Carlo studies. (Note: It is possible, but more advanced, to write this without a `for` loop. If you're up for the challenge, you should go for it. But this is not required.).

Here I calculated the proportion of times tha the test rejects assuming that H_0 is true (there was not clear signal that in the problem that I have to assume the contrary, as we need it to do it in order to calculate the power).

```
n=10
df=n-1
simulations=100000
simulated = rt(simulations, df)
lowercutoff = qt(0.025, df)
uppercutoff = qt(0.975, df)
num_reject = sum((simulated <= lowercutoff) | (simulated >= uppercutoff))
probability = num_reject/simulations
probability
```

```
## [1] 0.05114
```

As expected this value should be near to the significance level or type I error (in this case $\alpha = 0.05$), e.g, the probability of reject H_0 when H_0 is true.

c) Use the code from part (b) to estimate the power for four values of n. Try a wide enough range of n values to see a wide range of powers. Also, include $n = 15$ to see what they would have estimated the power of their own study to have been.

Given the information provided at the beginning of question 4, the power is defined intuitevely as the hability to detect that the alternative hypotheses applies. In this case is our hability to detect that the true mean is $\mu = 3$, assumming a standard deviation $\sigma = 1$

To calculate the power in this question we need to compute P(reject H_0 | H_A is true):

First, the power for the study described at the beginning of 4 with a n=15 and $\sigma = 1$ is:

```
u0=2
ua=3
n=15
df=n-1
s=1
ncp =  (ua - u0) / (s/sqrt(n))
lowercutoff = qt(0.025, df)
uppercutoff = qt(0.975, df)
power = pt(lowercutoff, df, ncp) + (1- pt(uppercutoff, df, ncp))
power
```

```
## [1] 0.9490865
```

We can verifiied the calculation with the next R code:

```
power.t.test(n, delta=1, sd=s, sig.level = 0.05, type = "one.sample", alternative="two.sided", strict =
```

```
##
##      One-sample t test power calculation
##
##               n = 15
##           delta = 1
##              sd = 1
##       sig.level = 0.05
##           power = 0.9490865
##     alternative = two.sided
```

In this part we are going to present te power of the study for different values of n:

Power With n=10

```
u0=2
ua=3
n=10
df=n-1
s=1
ncp =  (ua - u0) / (s/sqrt(n))
nsim=100000
simulated = rt(nsim, df, ncp)
lowercutoff = qt(0.025, df)
uppercutoff = qt(0.975, df)
num_reject = sum((simulated <= lowercutoff) | (simulated >= uppercutoff))
power = num_reject / nsim
power
```

```
## [1] 0.80351
```

Power With n=20

```
u0=2
ua=3
n=20
df=n-1
s=1
ncp =  (ua - u0) / (s/sqrt(n))
nsim=100000
simulated = rt(nsim, df, ncp)
lowercutoff = qt(0.025, df)
uppercutoff = qt(0.975, df)
num_reject = sum((simulated <= lowercutoff) | (simulated >= uppercutoff))
power = num_reject / nsim
power
```

```
## [1] 0.98863
```

Power With n=30

```
u0=2
ua=3
n=30
df=n-1
s=1
ncp =  (ua - u0) / (s/sqrt(n))
nsim=100000
simulated = rt(nsim, df, ncp)
lowercutoff = qt(0.025, df)
uppercutoff = qt(0.975, df)
num_reject = sum((simulated <= lowercutoff) | (simulated >= uppercutoff))
power = num_reject / nsim
power
```

```
## [1] 0.99957
```

Power With n=50

```
u0=2
ua=3
n=50
df=n-1
s=1
ncp =  (ua - u0) / (s/sqrt(n))
nsim=100000
simulated = rt(nsim, df, ncp)
lowercutoff = qt(0.025, df)
uppercutoff = qt(0.975, df)
num_reject = sum((simulated <= lowercutoff) | (simulated >= uppercutoff))
power = num_reject / nsim
power
```

```
## [1] 1
```

d) Find a value of n (using trial-and-error) for which the power is close to the desired 90%.

By trial and error I find that n=13 gives a power near of 90%.

```
u0=2
ua=3
n=13
df=n-1
s=1
ncp =  (ua - u0) / (s/sqrt(n))
nsim=100000
simulated = rt(nsim, df, ncp)
lowercutoff = qt(0.025, df)
uppercutoff = qt(0.975, df)
num_reject = sum((simulated <= lowercutoff) | (simulated >= uppercutoff))
power = num_reject / nsim
power
```

```
## [1] 0.91087
```