Homework 8 : Inference for the Difference Between Two Population Means

---

NAME: Andres Castano

NETID: ac986

DUE DATE: November 15, 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.

Problem 1

Are mean pulse rates different when students are taking a quiz versus when they are sitting in lecture? The *QuizPulse20* data contains the pulse rates for 20 randomly selected students from a large psychology class under two different scenerios: 1) when they were sitting in class taking a quiz and 2) when they were sitting in class during lecture.

μ_x = mean pulse rate for students taking a quiz

μ_y = mean pulse rate for students sitting in lecture

a) Read the `QuizPulse10` data into this homework document.

```
Pulse_data <- read.csv("QuizPulse20.csv")
dim(Pulse_data)
```

```
## [1] 20  4
```

b) Add a new column to this data frame containing the differences between the pulse rates for each student (Quiz - Lecture). Name this column "Differences."

```
Pulse_data$Differences = (Pulse_data$Quiz)-(Pulse_data$Lecture)
mean(Pulse_data$Differences)
```

```
## [1] 3
```

```
sd(Pulse_data$Differences)
```

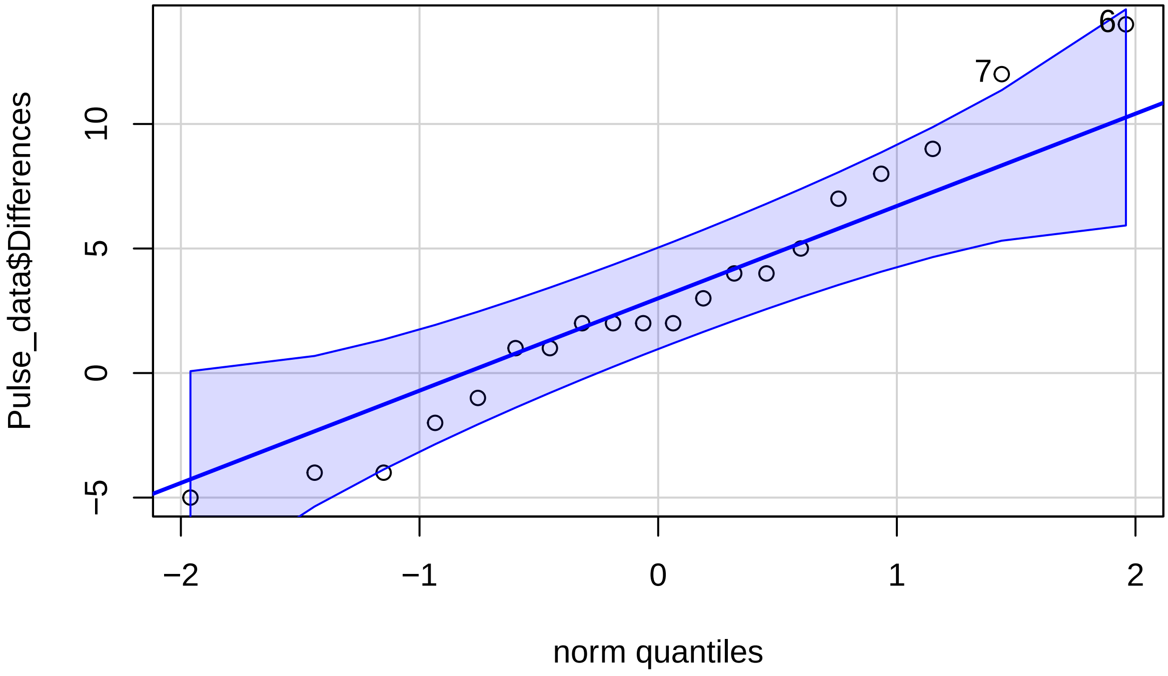
```
## [1] 5.129892
```

c) Since the sample of differences is small, it makes sense to check whether it seems reasonable that the differences are normally distributed. One way to check this is by looking at a Q-Q plot of the differences. The "car" package in R includes the function `qqPlot()`. To use this function, the "car" package will need to be installed. To install this package, from the R Studio menu choose *Tools > Install Packages....* A window will pop up in which you can specify "car" as the package to be installed. Then, just click on "Install". After this package is installed, run the following code to create the Q-Q plot. Knit your document to look at this plot. Do most of the points lie within the confidence bands? If so, it is reasonable to assume the differences are normally distributed.

```
library(car)
```

```
## Loading required package: carData
```

```
qqPlot(Pulse_data$Differences)
```



```
## [1] 6 7
```

It seems reasonable to suppose that the differences are normal distributed.

d) Create a 95% confidence interval for the difference between student mean pulse rate during a quiz and student mean pulse rate during lecture using the `t.test()` function in R. Do this by passing in the differences.

```
t.test(Pulse_data$Differences, conf.level=0.95)
```

```
##
##  One Sample t-test
##
## data:  Pulse_data$Differences
## t = 2.6153, df = 19, p-value = 0.01702
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
```

```
##  0.5991368 5.4008632
## sample estimates:
## mean of x
##         3
```

e) Interpret the confidence interval created in (d) in terms of the study.

Our 95% confidence interval is (0.5991368, 5.4008632). Which means that we are 95% confident that the population mean for the difference in pulse rates is between 0.5991368 and 5.4008632.

f) Complete the following steps to perform the test relevant to this study.

```
i) State the null and alternative hypotheses for this study

$H_{0}:  \mu_{quiz}-\mu_{lecture}=0$

$H_{A}:  \mu_{quiz}-\mu_{lecture} \neq 0$


ii) The output from (d) can also be used to perform a paired t-test.  What is the decision based on the
```

At significance level of 0.05, we have statistical evidence to reject the null hypothesis: p-value=0.01702 is lower that 0.05. Which means that there is a difference in pulse rates when students are taking a quiz versus when they are sitting in lecture.

Problem 2

Is there an association between increased pulse rate and test performance? The variable, `Score`, in the `QuizPulse20` data indicates each student's score on the quiz as a percent. Here we will investigate the relationship between an increase in pulse rate during a quiz and quiz performance. Due to measurement error, it was decided that a student's pulse will only be denoted as having increased during the quiz if it is at least 3 beats per minute higher than his/her pulse rate during lecture.

$\mu_x =$ mean score for students whose pulse increased during the quiz

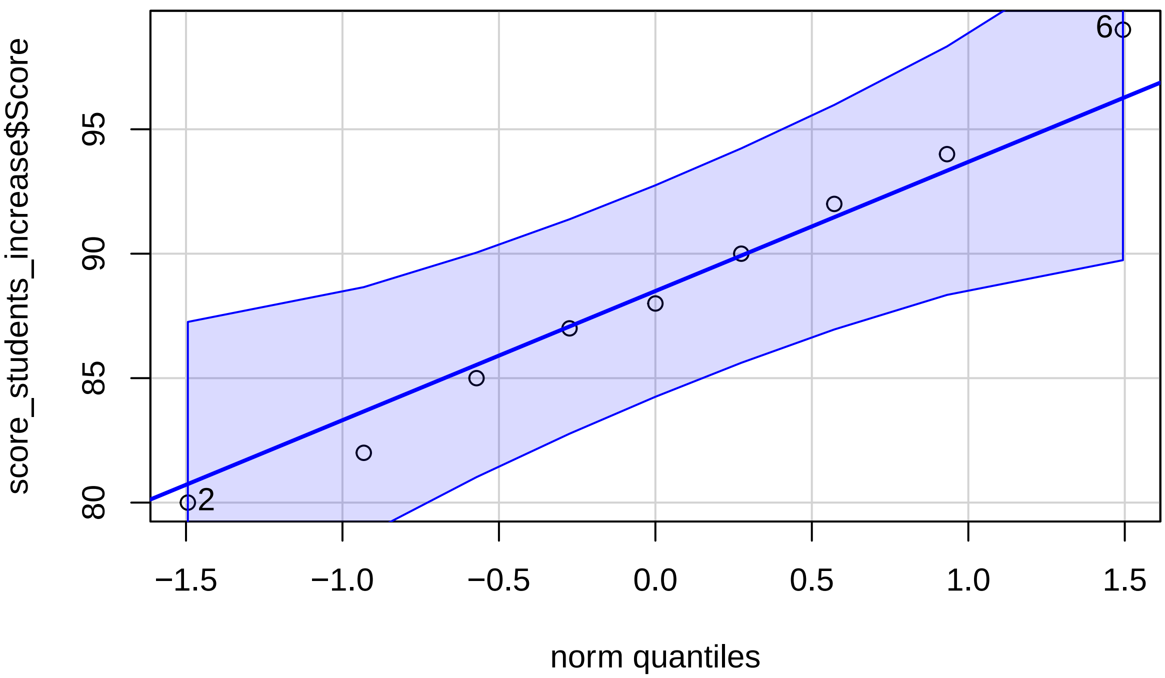
$\mu_y =$ mean score for students whose pulse did not increase during the quiz

a) In a code chunk create two vectors. The first vector should contain the scores for students whose pulse increased during the quiz. The second vector should contain scores for students whose pulse did not increase during the quiz (according to the criterion stated in the description of the study).

```r
score_students_increase <- subset(Pulse_data, Differences>=3, select = Score)
score_students_notincrease <- subset(Pulse_data, Differences<3, select = Score)
```

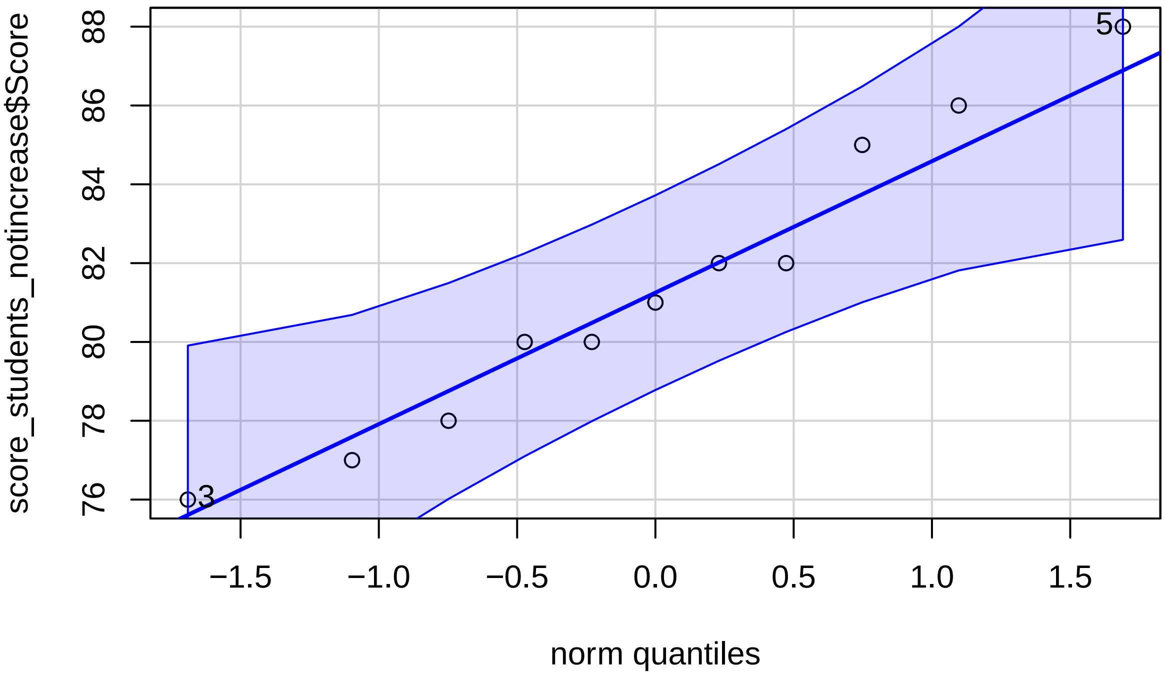
b) Since the sample size is small for both sets of students, it would be a good idea to look at the Q-Q plots of `Score` for the students whose pulse rate increased during the quiz and also for the sample of students whose pulse did not increase during the quiz. Use the `qqPlot()` function to create the Q-Q plots. Then, give an assessment of whether it seems reasonable to assume the populations these samples are drawn from are normally distributed.

```
qqPlot(score_students_increase$Score)
```



```
## [1] 6 2
```

```
qqPlot(score_students_notincrease$Score)
```



```
## [1] 5 3
```

It seems reasonable to assume that the scores for the two groups are nearly normally distributed.

c) Create a 95% confidence interval for the difference in mean scores for the two groups of students using the `t.test()` function. Pass in the two groups' data as the first two arguments to `t.test`.

```
t.test(x=score_students_increase$Score, y=score_students_notincrease$Score, conf.level=0.95)
```

```
##
##  Welch Two Sample t-test
##
## data:  score_students_increase$Score and score_students_notincrease$Score
## t = 3.1408, df = 12.994, p-value = 0.007813
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##   2.244783 12.139056
## sample estimates:
## mean of x mean of y
##  88.55556  81.36364
```

d) Interpret the confidence interval from (c) in the context of the study.

Our 95% confidence interval is (2.244783, 12.139056). Which means that we are 95% confident that the population mean for the difference scores between the groups is between 2.244783 pp. and 12.139056 pp.

e) At the 0.05 significance level, is there evidence that the mean score for students whose pulses increase is different from the mean score for students whose pulses do not increase? Base your answer on the confidence interval created in (c).

The 95% confidence interval does not containt zero, which means that there is a difference in the scores among groups. The p-value confirms that result: p-value=0.007813 is lower that the significance level 0.05.

f) Increased pulse rate has been shown to be associated with higher stress levels. Based on the result of this study, would it make sense for the professor of this psychology class to take measures to reduce the stress level of students on days they are taking a quiz in order to increase their scores?

First we need to find evidence that the scores of the students with an increased in pulse rate are greater compared to students with no increase in pulse rate, so we can make a one tail hypothesis testing as follows:

$\mu_x =$ mean score for students whose pulse increased during the quiz

$\mu_y =$ mean score for students whose pulse did not increase during the quiz

$H_0 : \mu_x - \mu_y = 0$

$H_0 : \mu_x - \mu_y > 0$

```
t.test(x=score_students_increase$Score, y=score_students_notincrease$Score, alternative ="greater", con
```

```
##
##  Welch Two Sample t-test
##
## data:  score_students_increase$Score and score_students_notincrease$Score
## t = 3.1408, df = 12.994, p-value = 0.003906
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
##  3.136615      Inf
## sample estimates:
## mean of x mean of y
##  88.55556  81.36364
```

Given the result above, I think the idea to take measures to reduce the stress level of students on days when they are taking a quiz in order to increase their scores does not make sense. The students seem to perform better when they have an increase (greater or equal than 3) in pulse rates during a quiz compare to a lecture. Given that stress levels are positive related with pulse rates and and increase in pulse rates seem to boost the score of the students, if the professor try to reduce the stress level, maybe the students are not going to perform better (under the same quiz difficulty). This outcome might support the idea that the students perform better under pressure.

Power

The power of a test is the probability you will reject the null hypothesis when it is false. In problem 3 we will investigate how the power of a two independent sample t-test is influenced by the following factors:

1) Sample size
2) Effect Size - the absolute difference between the means of the two populations

Problem 3

a) In a code chunk here, define the following variables associated with two independent samples, $X_1, \ldots, X_{16}$ and $Y_1, \ldots, Y_{14}$ where the respective samples are drawn from $X_i \sim N(6, 4)$ and $Y_i \sim N(8, 8)$.

```
1. `nx` = $n_x$

2. `ny` = $n_y$

3. `mux` = $\mu_x$

4. `muy` = $\mu_y$

5. `sigx` = $\sigma_x$

6. `sigy` = $\sigma_y$
```

```
nx=16
ny=14
mux=6
muy=8
sigx=4
sigy=8
```

b) Using the variables defined above, simulate two independent samples from the respective normal distributions and perform a t-test for the hypothesis, $H_0 : \mu_x = \mu_y$.

```
data_x=rnorm(n=16, mean=6, sd=4)
data_y=rnorm(n=14, mean=8, sd=8)
t.test(x=data_x, y=data_y, conf.level=0.95)
```

```
##
##  Welch Two Sample t-test
##
## data:  data_x and data_y
## t = -1.7898, df = 26.004, p-value = 0.08514
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -7.1222679  0.4921252
## sample estimates:
## mean of x mean of y
##  6.903338 10.218409
```

c) In this part you will perform a simulation to check whether the significance level of the test is in fact 0.05. Start by copying and pasting the code from (a) and (b) into one code chunk here. Steps (i)-(vi) will walk you through writing the simulation.

I have performed all the steps describe below in one unique chunk, please see below.

```
i) Remember that the significance level of a test corresponds to the probability the null hypothesis is

ii) Name the output of  `t.test` something like `t.sampxy`.  Save the p-value of your t-test by includin

iii) On the first line in your code chunk create a vector called `rej` that includes 5,000 zeros.

iv) Using a `for` loop, repeat the following process 5,000 times.


1. Simulate two independent samples, one from $N(\mu_x, \sigma_x)$ and the other from $N(\mu_y, \sigma_

2. Perform a t-test for these two samples (using the code in (ii) so that you also save the p-value).

3. Set `rej[i] = pvalue <= 0.05` to assign `rej[i]` to 0 in iteration i if the null hypothesis is not re

v) After the `for` loop created in the last step, estimate the probability that the null hypothesis is r
```

Here I have performed all the steps describe previously in one chunk.

```
set.seed=2
simulations=5000
rej=rep(0,simulations)
nx=16
ny=14
mux=6
muy=6
sigx=4
sigy=8
for (i in 1:simulations) {
data_x=rnorm(n=nx, mean=mux, sd=sigx)
data_y=rnorm(n=ny, mean=muy, sd=sigy)
t.sampxy=t.test(x=data_x, y=data_y, conf.level=0.95)
pvalue=t.sampxy$p.value
 if (pvalue  < 0.05) {
    rej[i]= 1
  }
}
proba_reject_h0=sum(rej)/simulations
proba_reject_h0
```

```
## [1] 0.0542
```

vi) Does the estimate for the significance level of the test indicate that it is 0.05? Why or why not.

Given the variation from sample to sample, we do not expect that α would be exactly 0.05, but as we can see above, the value is pretty near from 0.05.

d) Now let's estimate the power of the test performed in (b). Remember that the power of a test is the probability that we will reject H_0 when it is false. Repeat the simulation from part (c) where now `mux` and `muy` are set to their original values from part (a). What is the approximate power of this test?

```
set.seed=2
simulations=5000
rej=rep(0,simulations)
nx=16
ny=14
```

```
mux=6
muy=8
sigx=4
sigy=8
for (i in 1:simulations) {
data_x=rnorm(n=nx, mean=mux, sd=sigx)
data_y=rnorm(n=ny, mean=muy, sd=sigy)
t.sampxy=t.test(x=data_x, y=data_y, conf.level=0.95)
pvalue=t.sampxy$p.value
 if (pvalue  < 0.05) {
    rej[i]= 1
  }
}
power=sum(rej)/simulations
power
```

```
## [1] 0.1234
```

e) Using the code from part (d), include a code chunk for parts (i), (ii), and (iii), below, that changes the power calculation as specified. Then, for all three parts, make a general statement of how changing the parameter as indicated seems to impact the power calculation.

   i) Repeat the power calculation twice. First, increase `nx` to 25, keeping everything else fixed. For the next power calculation change `nx` to 50, keeping everything else fixed.

With nx=25:

```
set.seed=2
simulations=5000
rej=rep(0,simulations)
nx=25
ny=14
mux=6
muy=8
sigx=4
sigy=8
for (i in 1:simulations) {
data_x=rnorm(n=nx, mean=mux, sd=sigx)
data_y=rnorm(n=ny, mean=muy, sd=sigy)
t.sampxy=t.test(x=data_x, y=data_y, conf.level=0.95)
pvalue=t.sampxy$p.value
 if (pvalue  < 0.05) {
    rej[i]= 1
  }
}
power=sum(rej)/simulations
power
```

```
## [1] 0.1336
```

With nx=50

```
set.seed=2
simulations=5000
rej=rep(0,simulations)
nx=50
ny=14
```

```
mux=6
muy=8
sigx=4
sigy=8
for (i in 1:simulations) {
data_x=rnorm(n=nx, mean=mux, sd=sigx)
data_y=rnorm(n=ny, mean=muy, sd=sigy)
t.sampxy=t.test(x=data_x, y=data_y, conf.level=0.95)
pvalue=t.sampxy$p.value
 if (pvalue  < 0.05) {
    rej[i]= 1
  }
}
power=sum(rej)/simulations
power
```

```
## [1] 0.1322
```

The increase of nx improve a little bit the power of the test.

```
ii) Repeat the power calculation twice.  First, increase `nx` and `ny` to 25, keeping everything else f
```

With nx=25 and ny=25

```
set.seed=2
simulations=5000
rej=rep(0,simulations)
nx=25
ny=25
mux=6
muy=8
sigx=4
sigy=8
for (i in 1:simulations) {
data_x=rnorm(n=nx, mean=mux, sd=sigx)
data_y=rnorm(n=ny, mean=muy, sd=sigy)
t.sampxy=t.test(x=data_x, y=data_y, conf.level=0.95)
pvalue=t.sampxy$p.value
 if (pvalue  < 0.05) {
    rej[i]= 1
  }
}
power=sum(rej)/simulations
power
```

```
## [1] 0.2046
```

With nx=50 and ny=50:

```
set.seed=2
simulations=5000
rej=rep(0,simulations)
nx=50
ny=50
mux=6
muy=8
sigx=4
```

```
sigy=8
for (i in 1:simulations) {
data_x=rnorm(n=nx, mean=mux, sd=sigx)
data_y=rnorm(n=ny, mean=muy, sd=sigy)
t.sampxy=t.test(x=data_x, y=data_y, conf.level=0.95)
pvalue=t.sampxy$p.value
 if (pvalue  < 0.05) {
    rej[i]= 1
  }
}
power=sum(rej)/simulations
power
```

```
## [1] 0.336
```

When both nx and ny increase, the power of the test increases greater than only when nx changes.

iii) Repeat the power calculation twice. First, increase the effect size, |`mux` - `muy`| to 5, keeping

With effect size=5:

```
set.seed=2
simulations=5000
rej=rep(0,simulations)
nx=16
ny=14
mux=6
muy=11
sigx=4
sigy=8
for (i in 1:simulations) {
data_x=rnorm(n=nx, mean=mux, sd=sigx)
data_y=rnorm(n=ny, mean=muy, sd=sigy)
t.sampxy=t.test(x=data_x, y=data_y, conf.level=0.95)
pvalue=t.sampxy$p.value
 if (pvalue  < 0.05) {
    rej[i]= 1
  }
}
power=sum(rej)/simulations
power
```

```
## [1] 0.5162
```

With effect size=10:

```
set.seed=2
simulations=5000
rej=rep(0,simulations)
nx=16
ny=14
mux=6
muy=16
sigx=4
sigy=8
for (i in 1:simulations) {
data_x=rnorm(n=nx, mean=mux, sd=sigx)
data_y=rnorm(n=ny, mean=muy, sd=sigy)
t.sampxy=t.test(x=data_x, y=data_y, conf.level=0.95)
pvalue=t.sampxy$p.value
 if (pvalue  < 0.05) {
    rej[i]= 1
  }
}
power=sum(rej)/simulations
power
```

```
## [1] 0.9776
```

The power of the test increases when we are interested in identify a greater effect size (keeping the rest of parameters constant). For effect size = 5 the power is about 51%, whereas for an effect size =10 the power is almost 98%.

```
pchisq(6.12, 2, lower.tail = FALSE)
```

```
## [1] 0.0468877
```