Assignment 8

Your name and student ID

Today's date

```
BEGIN ASSIGNMENT
requirements: requirements.R
generate: true
```

- Solutions released: Wednesday, April 7 by 10:00pm.*

Helpful hints:

- Every function you need to use was taught during lecture! So you may need to revisit the lecture code to help you along by opening the relevant files on Datahub. Alternatively, you may wish to view the code in the condensed PDFs posted on the course website. Good luck!
- Knit your file early and often to minimize knitting errors! If you copy and paste code for the slides, you are bound to get an error that is hard to diagnose. Typing out the code is the way to smooth knitting! We recommend knitting your file each time after you write a few sentences/add a new code chunk, so you can detect the source of knitting errors more easily. This will save you and the GSIs from frustration! **You must knit correctly before submitting.**
- If your code runs off the page of the knitted PDF then you will LOSE POINTS! To avoid this, have a look at your knitted PDF and ensure all the code fits in the file (you can easily view it on Gradescope via the provided link after submitting). If it doesn't look right, go back to your .Rmd file and add spaces (new lines) using the return or enter key so that the code runs onto the next line.

Section 1: Hemoglobin levels

In two wards for elderly patients in a local hospital the following levels of hemoglobin (grams per liter) were found for a simple random sample of patients from each ward.:

Ward A:

```
ward_a <- c(12.2, 11.1, 14.0, 11.3, 10.8, 12.5, 12.2, 11.9, 13.6, 12.7, 13.4, 13.7)
```

Ward B:

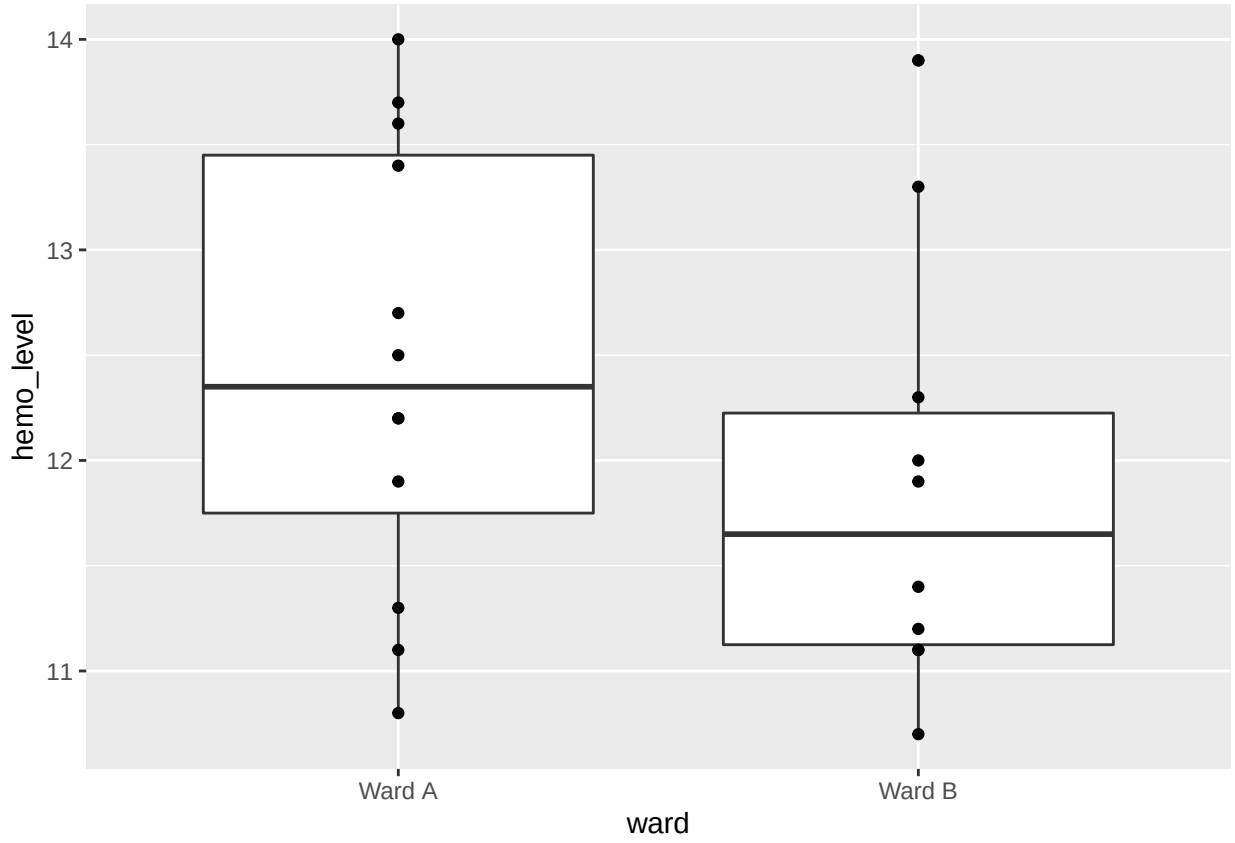
```
ward_b <- c(11.9, 10.7, 12.3, 13.9, 11.1, 11.2, 13.3, 11.4, 12.0, 11.1)
```

1. [1 point] In one ggplot, create two box plots to compare the hemoglobin values for Ward A and Ward B. Also plot the raw data as points, overlaid on top of the box plots.

```
BEGIN QUESTION
name: p1
manual: false
points: 1
```

```
hemoglobin <- data.frame(hemo_level = c(ward_a, ward_b),
                         ward = c(rep("Ward A", 12), rep("Ward B", 10)))
```

```
p1 <- ggplot(hemoglobin, aes(y = hemo_level, x = ward)) + geom_boxplot() + geom_point() # SOLUTION
p1
```



```
## Test ##
test_that("p1a", {
  expect_true("ggplot" %in% class(p1))
  print("Checking: ggplot defined")
})
```

```
## [1] "Checking: ggplot defined"
## Test passed
```

```
## Test ##
test_that("p1b", {
  expect_true(identical(p1$data, hemoglobin))
  print("Checking: hemoglobin data used")
})
```

```
## [1] "Checking: hemoglobin data used"
## Test passed
```

```
## Test ##
test_that("p1c", {
  expect_true(rlang::quo_get_expr(p1$mapping$y) == "hemo_level")
  print("Checking: hemo_level on y-axis")
})
```

```
## [1] "Checking: hemo_level on y-axis"
## Test passed
```

```
## Test ##
test_that("p1d", {
  expect_true(rlang::quo_get_expr(p1$mapping$x) == "ward")
  print("Checking: hemo_level on y-axis")
})
```

```
## [1] "Checking: hemo_level on y-axis"
## Test passed
```

```
## Test ##
test_that("p1e", {
  expect_true("GeomBoxplot" %in% class(p1$layers[[1]]$geom))
  print("Checking: boxplot created")
})
```

```
## [1] "Checking: boxplot created"
## Test passed
```

```
## Test ##
test_that("p1f", {
  expect_true("GeomPoint" %in% class(p1$layers[[2]]$geom))
  print("Checking: data points overland")
})
```

```
## [1] "Checking: data points overland"
## Test passed
```

2. [1 points] Comment on the similarities/differences portrayed by the plots, keeping in mind that the sample size is relatively smallfor these two wards.

```
BEGIN QUESTION
name: p2
manual: true
```

There is some overlap in the middle 50% of the data form these two wards. There do not appeard to be outliers in either distribution. Both samples appear to be roughly symmetric. The sample median is higher in Ward A than Ward B.

3. [2 points] What two assumptions do you need to make to use any of the t-procedures? Because each ward has a rather small sample size (n $<$ 12 for both), what two characteristics of the data would you need to check for to ensure that the t-procedures can be applied?

```
BEGIN QUESTION
name: p3
manual: true
```

- Two assumptions: SRS, normality of underlying dataset
- No outliers, data has similar shapes

4. [3 points] Using only `dplyr` and `*t` functions, create a 95% confidence interval for the mean difference between Ward A and Ward B. You can do this by using `dplyr` to calculate the inputs required to calculate the 95% CI, and then plugging these values in on a separate line of code (or using your calculator). Use a degrees of freedom of 19.515 (You don't need to calculate the degrees of freedom, you can use this value directly). Show your work and interpret the mean difference and its 95% CI. Round your solution to 3 decimal places.

```
BEGIN QUESTION
name: p4
manual: false
points: 3
```

```
. = " # BEGIN PROMPT
# YOUR CODE HERE

# THEN, ASSIGN YOUR FINAL ANSWERS BELOW:
CI_lowerbound <- 'YOUR ANSWER HERE'
CI_upperbound <- 'YOUR ANSWER HERE'
" # END PROMPT

# BEGIN SOLUTION NO PROMPT
hemoglobin %>% group_by(ward) %>% summarise(sample_mean = mean(hemo_level),
                                            sample_var = var(hemo_level),
                                            n = length(hemo_level))
```

```
## # A tibble: 2 x 4
##   ward   sample_mean sample_var     n
## * <chr>        <dbl>      <dbl> <int>
## 1 Ward A        12.4       1.14    12
## 2 Ward B        11.9       1.07    10
```

```
# here is how you calculate degrees of freedom
deg_free <- ( (1.140909/12) + (1.065444/10) )^2/( (1/11)*(1.140909/12)^2 + (1/9)*(1.065444/10)^2)
mean_diff <- 12.45 - 11.89
se_diff <- sqrt(1.140909/12 + 1.065444/10)
t_star <- qt(p = 0.025, df = deg_free)

CI_upperbound <- mean_diff - t_star * (se_diff)
CI_lowerbound <- mean_diff + t_star * (se_diff)
# END SOLUTION
```

```
## Test ##
test_that("p4a", {
  expect_true(all.equal(round(CI_upperbound, 3), 1.498, 0.001))
  print("Checking: value of upperbound")
})
```

```
## [1] "Checking: value of upperbound"
## Test passed
```

```r
## Test ##
test_that("p4b", {
  expect_true(all.equal(round(CI_lowerbound, 2), -0.38, 0.01))
  print("Checking: value of lowerbound")
})
```

```
## [1] "Checking: value of lowerbound"
## Test passed
```

5. [1 points] Interpret the mean difference and its 95% CI you just calculated.

```
BEGIN QUESTION
name: p5
manual: true
```

The sample mean difference is 0.56 and its 95% CI goes from -0.44 to 1.56. This means that if we were to repeat this procedure 100 times, we would expect that 95 of the CIs would contain the true difference. The range of the difference goes from negative to positive indicating that at the 5% level there is no evidence against the null hypothesis of no difference.

Perform a two-sided t-test for the difference between the two samples, where the null hypothesis is that the underlying means are the same. Start by writing down the null and alternate hypotheses, then calculate the test statistic (showing your work) and p-value. Continue to assume that the degrees of freedom is 19.515. Verify the p-value by running the t-test using R's built in function. Show the output from that test. Hint: to perform the t-test using R's built in function, you need to pass the function an x and y argument, where x includes that values for Ward A and Y includes the values for Ward B. `dplyr`'s `filter()`and `pull()` functions will be your friends.

6. [1 point] Calculate the t-test statistics

```
BEGIN QUESTION
name: p6
manual: false
points: 1
```

```
t_statistics <- round(1.247157,2) # SOLUTION
```

```
## Test ##
test_that("p6", {
  expect_true(all.equal(round(t_statistics, 2), 1.25, 0.01))
  print("Checking: value of test statistic")
})
```

```
## [1] "Checking: value of test statistic"
## Test passed
```

$t = \frac{(\bar{x}_1 - \bar{x}_2) - (\mu_1 - \mu_2)}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}}$ $t = \frac{(12.45 - 11.89) - 0}{\sqrt{\frac{1.140909}{12} + \frac{1.065444}{10}}}$ $t = 0.56/0.4490213 = 1.247157$

7. [1 point] We need to compare this t-statistic to a t distribution with 19.515 degrees of freedom:

```
BEGIN QUESTION
name: p7
manual: false
points: 1
```

```
p_value <- pt(1.247157, df = 19.515, lower.tail = F) * 2 # SOLUTION
p_value
```

```
## [1] 0.2271006
```

```
## Test ##
test_that("p7", {
  expect_true(all.equal(round(p_value, 2), 0.23, 0.01))
  print("Checking: p-value")
})
```

```
## [1] "Checking: p-value"
## Test passed
```

8.[2 points] Interpret the p value you got in the context of the this question. Are there evidence against null hypothesis?

```
BEGIN QUESTION
name: p8
manual: false
points: 2
```

Thus there is a 22.7% chance of seeing a difference of the size we saw or larger under the hypothesis of no difference. This is quite probable, so we conclude that there is no evidence against the null hypothesis.

Check this against the `t.test` output:

```
t.test(x = hemoglobin %>% filter(ward == "Ward A") %>% pull(hemo_level),
       y = hemoglobin %>% filter(ward == "Ward B") %>% pull(hemo_level),
       alternative = "two.sided")
```

```
##
##  Welch Two Sample t-test
##
## data:  hemoglobin %>% filter(ward == "Ward A") %>% pull(hemo_level) and hemoglobin %>% filter(ward ==
## t = 1.2472, df = 19.515, p-value = 0.2271
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.3781372  1.4981372
## sample estimates:
## mean of x mean of y
##     12.45     11.89
```

The time to perform open heart surgery is normally distributed. Sixteen patients (chosen as a simple random sample from a hospital) underwent open heart surgery that took the following lengths of time (in minutes).

```
op_time <- c(247.8648, 258.4343, 315.6787, 268.0563, 269.9372, 320.6821,
             280.5493, 225.3180, 243.8207, 251.5388, 304.9706, 277.3140,
             278.6247, 269.3418, 248.0131, 322.9812)
surg_data <- data.frame(op_time)
```

9. [1 point] You wish to know if the mean operating time of open heart surgeries at this hospital exceeds four hours. Set up appropriate hypotheses for investigating this issue.

```
BEGIN QUESTION
name: p9
manual: false
points: 1
```

$H_0 : \mu = 4$ hours (240 mins) $H_a : \mu > 4$ hours (240 mins)

10. [1 point] Test the hypotheses you formulated in part (a). Report the p-value. (Do not use the `t.test` function for this question)

```
BEGIN QUESTION
name: p10
manual: false
points: 1
```

```
. = " # BEGIN PROMPT
p_value_10 <- 'YOUR ANSWER HERE'
p_value_10
" # END PROMPT

# BEGIN SOLUTION
surg_data %>% summarise(mean = mean(op_time), se = sd(op_time)/sqrt(16))
```

```
##       mean       se
## 1 273.9454 7.305622
```

```
p_value_10 <- pt((273.9454-240)/7.305621, df = 15, lower.tail = F)
p_value_10
```

```
## [1] 0.0001582348
```

```
# END SOLUTION
```

```
## Test ##
test_that("p10", {
  expect_true(all.equal(round(p_value_10, 2), 0, 0.01))
  print("Checking: p-value")
})
```

```
## [1] "Checking: p-value"
## Test passed
```

11. [1 point] What are your conclusions in the context of the question?

```
BEGIN QUESTION
name: p11
manual: true
```

The p-value of 0.000158, which is very small. There is only a miniscule chance of seeing the sample mean we saw (or larger) if the null hypothesis is true. Thus we reject the null hypothesis in favor of the alternative, that the operating time exceeds 4 hours.

12. [3 points] Construct a 95% CI on the mean operating time (in hours).

```
BEGIN QUESTION
name: p12
manual: false
points: 3
```

```
. = " # BEGIN PROMPT
# YOUR CODE HERE

# THEN, ASSIGN YOUR ANSWERS BELOW:
CI_lowerbound_12 <- 'YOUR ANSWER HERE'
CI_upperbound_12<-'YOUR ANSWER HERE'
" # END PROMPT

# BEGIN SOLUTION NO PROMPT
CI_lowerbound_12 <- 4.31
CI_upperbound_12 <- 4.73

# END SOLUTION
```

```
## Test ##
test_that("p12a", {
  expect_true(all.equal(round(CI_upperbound_12, 2), 4.73, 0.01))
  print("Checking: value of upperbound")
})
```

```
## [1] "Checking: value of upperbound"
## Test passed
```

```
## Test ##
test_that("p12b", {
  expect_true(all.equal(round(CI_lowerbound_12, 2), 4.31, 0.01))
  print("Checking: value of lowerbound")
})
```

```
## [1] "Checking: value of lowerbound"
## Test passed
```

qt(p = 0.975, df = 15) $\bar{x} \pm t^* \frac{s}{\sqrt{n}}$ $273.9454 \pm 2.13145 \times 7.305621 = 258.3738$ to $289.517 = 4.31$ hours to 4.73 hours

Thus, using a method that includes the null value 95 times out of 100, our 95% CI is 4.31 hours to 4.73 hours.

13. [1 point] Suppose you were testing the hypotheses $H_0 : \mu_d = 0$ and $H_a : \mu_d \neq 0$ in a paired design and obtain a p-value of 0.21. Which one of the following could be a possible 95% confidence interval for μ_d?

```
BEGIN QUESTION
name: p13
manual: false
points: 1
```

```
. = " # BEGIN PROMPT
# Uncomment one of the following choices:
# p13 <- '-2.30 to -0.70'
# p13 <- '-1.20 to 0.90'
# p13 <- '1.50 to 3.80'
# p13 <- '4.50 to 6.90'
" # END PROMPT

# BEGIN SOLUTION NO PROMPT
p13 <- "-1.20 to 0.90"
# END SOLUTION
```

```
## Test ##
test_that("p13", {
  expect_true(p13 == "-1.20 to 0.90")
  print("Checking: selection")
})
```

```
## [1] "Checking: selection"
## Test passed
```

14. [1 point] Suppose you were testing the hypotheses $H_0 : \mu_d = 0$ and $H_a : \mu_d \neq 0$ in a paired design and obtain a p-value of 0.02. Also suppose you computed confidence intervals for μ_d. Based on the p-value which one of the following is true?

```
BEGIN QUESTION
name: p14
manual: false
points: 1
```

```
. = " # BEGIN PROMPT
# Uncomment one of the following choices:
# p14 <- 'Both a 95% CI and a 99% CI will contain 0.'
# p14 <- 'A 95% CI will contain 0, but a 99% CI will not.'
# p14 <- 'A 95% CI will not contain 0, but a 99% CI will.'
# p14 <- 'Neither a 95% CI nor a 99% CI interval will contain 0.'
" # END PROMPT

# BEGIN SOLUTION NO PROMPT
p14 <- "A 95% CI will not contain 0, but a 99% CI will."
# END SOLUTION
```

```
## Test ##
test_that("p14", {
  expect_true(p14 == "A 95% CI will not contain 0, but a 99% CI will.")
  print("Checking: selection")
})
```

```
## [1] "Checking: selection"
## Test passed
```