

# Problem Set 8

Your name and student ID

Today's date

**Run this chunk of code to load the autograder package!**

## Instructions

- Solutions will be released by Wednesday, March 22nd.
- This semester, problem sets are for practice only and will not be turned in for marks.

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!
- To avoid code running off the page, have a look at your knitted PDF and ensure all the code fits in the file. 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 a local hospital, there are two wards for elderly patients. Researchers took a simple random sample of hemoglobin levels (grams per liter) from patients in 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)
```

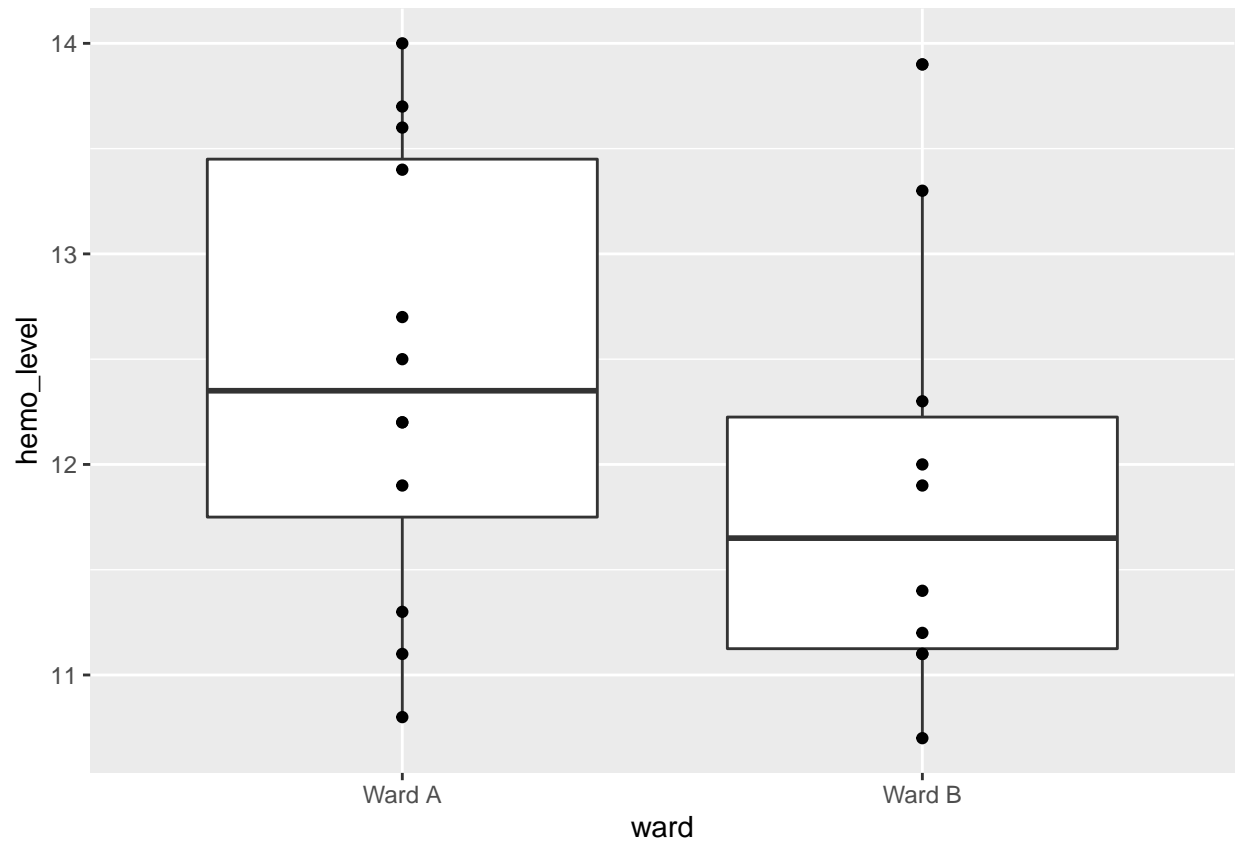
1. In a single ggplot, create two box plots to compare the hemoglobin values for Ward A and Ward B (the two boxplots should be next to one another on one plot). Also plot the raw data as points, overlaid on top of the box plots.

```
# This line of code creates a dataframe of hemoglobin levels from both wards
hemoglobin <- data.frame(hemo_level = c(ward_a, ward_b),
                        ward = c(rep("Ward A", 12), rep("Ward B", 10)))
```

```
. = " # BEGIN PROMPT
p1 <- NULL # YOUR CODE HERE
p1
" # END PROMPT

# BEGIN SOLUTION NO PROMPT
p1 <- ggplot(hemoglobin, aes(y = hemo_level, x = ward)) +
  geom_boxplot() +
  geom_point()

p1
```



```
# END SOLUTION
```

```
test_that("p1a", {
  expect_true("ggplot" %in% class(p1))
  print("Checking: p1 is a ggplot")
})
```

```
## [1] "Checking: p1 is a ggplot"
## Test passed
```

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

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

```
test_that("p1c", {
  expect_true(rlang::quo_get_expr(p1$mapping$x) == "ward")
  print("Checking: Ward is on the x-axis")
})
```

```
## [1] "Checking: Ward is on the x-axis"
## Test passed
```

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

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

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

```
## [1] "Checking: Made a boxplot"
## Test passed
```

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

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

**2. Comment on the similarities/differences portrayed by the plots, keeping in mind that the sample size is relatively small for these two wards.**

There is some overlap in the middle 50% of the data from these two wards. There do not appear 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. 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?**

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

Next, you will perform a two-sided t-test for the difference between the two samples' mean hemoglobin levels, where the null hypothesis is that the underlying means are the same. Assume that the degrees of freedom is 19.515 (you don't need to calculate the degrees of freedom, you can use this value directly).

**4. Write the null and alternative hypotheses for this test.**

$H_0$ : There is no difference in the mean hemoglobin levels between Ward A and Ward B.  $H_A$ : There is a difference in the mean hemoglobin levels between Ward A and Ward B.

**5. Calculate the mean difference between the two samples' mean hemoglobin levels and interpret this value.**

```
. = " # BEGIN PROMPT
mean_diff <- NULL # YOUR CODE HERE
mean_diff
" # END PROMPT

# BEGIN SOLUTION
hemoglobin %>% group_by(ward) %>% summarize(sample_mean = mean(hemo_level))
```

```
## # A tibble: 2 x 2
##   ward   sample_mean
##   <chr>         <dbl>
## 1 Ward A         12.4
## 2 Ward B         11.9
```

```
mean_diff <- 12.45 - 11.89
# END SOLUTION
```

```
test_that("p5a", {
  expect_true(all.equal(mean_diff, 0.56, tol = 0.01))
  print("Checking: Mean difference is correct")
})
```

```
## [1] "Checking: Mean difference is correct"
## Test passed
```

The sample mean difference in hemoglobin levels between Ward A and Ward B is 0.56 g/L.

**6. Calculate the 2-sided t-test statistic for the difference between the two samples' mean hemoglobin levels (by hand). Then use this test statistic and the degrees of freedom to calculate the p-value using a function in R. Round both the test statistic and p-value to 3 decimal places and assign them to a vector called `t_stat_p_val`. Confirm your test statistic and p-value are correct using R's built in `t.test` function. Does this p-value indicate that there is evidence to reject the null?**

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 the values for Ward A and Y includes the values for Ward B. `dplyr`'s `filter()` and `pull()` functions will be your friends.

```
. = " # BEGIN PROMPT
t_stat_p_val <- NULL # YOUR CODE HERE
t_stat_p_val
" # END PROMPT
```

```
# BEGIN SOLUTION NO PROMPT
# calculate the sample mean, sample variance, and sample size for the t test statistic
hemoglobin %>% group_by(ward) %>% summarize(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
```

```
# calculate the t test statistic
t_stat <- round((12.45-11.89)/(sqrt((1.140909/12) + (1.065444/10))),3)
```

```
# calculate the p-value
p_val <- round(pt(1.247157, df = 19.515, lower.tail = F) * 2,3)
```

```
# make a vector of test stat and p val
t_stat_p_val <- c(t_stat, p_val)
```

```
# Check t_stat and obtain p-value using `t.test`
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 == "Ward B") %>% pull(hemo_level)
## 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
```

```
# END SOLUTION
```

```
test_that("p6a", {
  expect_true(all.equal(t_stat_p_val[1], 1.247, tol = 0.01))
  print("Checking: t test-statistic is correct")
})
```

```
## [1] "Checking: t test-statistic is correct"
## Test passed
```

```
test_that("p6b", {
  expect_true(all.equal(t_stat_p_val[2], .227, tol = 0.01))
  print("Checking: p-value is correct")
})
```



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

p-value: 0.227. Thus there is a 22.7% chance of seeing a test statistic as extreme or more extreme under the null hypothesis that there is no difference in the mean hemoglobin levels between the wards. This is quite probable, so we conclude that there is no evidence against the null hypothesis.

7. Calculate the 95% confidence interval (by hand) for the mean difference in hemoglobin levels between Ward A and Ward B that you calculated above. You can do this using `dplyr` to calculate the individual components of the 95% CI equation. Continue to assume the degrees of freedom is 19.515. Round your lower and upper bounds to 3 decimal places and assign these to a vector called `CI`.

```
. = " # BEGIN PROMPT
CI <- NULL # YOUR CODE HERE
CI
" # END PROMPT

# BEGIN SOLUTION
# find the mean hemoglobin level, variance of hemoglobin level, and sample size of each ward
hemoglobin %>% group_by(ward) %>% summarize(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 would 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 )

# calculate the mean difference in hemo_level, the se_difference in hemo_level, and the t_star value
mean_diff <- 12.45 - 11.89
se_diff <- sqrt(1.140909/12 + 1.065444/10) # remember variance is SE^2
t_star <- qt(p = 0.025, df = deg_free, lower.tail = FALSE)

# calculate the CI using components calculated above and round to 3 decimal places
CI_lowerbound <- round(mean_diff - t_star * (se_diff),3)
CI_upperbound <- round(mean_diff + t_star * (se_diff),3)

CI <- c(CI_lowerbound, CI_upperbound)
# END SOLUTION

test_that("p7a", {
  expect_true(all.equal(CI[1], -0.378, tol = 0.01))
  print("Checking: Lowerbound is correct")
})
```

```
## [1] "Checking: Lowerbound is correct"
## Test passed
```

```
test_that("p7b", {
  expect_true(all.equal(CI[2], 1.498, tol = 0.01))
  print("Checking: Upperbound is correct")
})
```

```
## [1] "Checking: Upperbound is correct"
## Test passed
```

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

### 8. Interpret the 95% CI in the context of this problem. Is there evidence to reject the null?

The 95% CI for the sample mean difference in hemoglobin levels between Ward A and Ward B is (-0.378, 1.498). 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 in hemoglobin levels between the two wards. Because the confidence interval crosses 0 (our null value), there is no evidence against the null hypothesis (of no difference between hemoglobin levels) at the 5% level.

## Open Heart Surgery

The amount of 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. You wish to know if the mean operating time of open heart surgeries at this hospital exceeds four hours. Set up appropriate null and alternative hypotheses for investigating this issue.**

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

10. Test the hypotheses you formulated in question 9 and assign the associated p-value to the object called `p_value_10`. (Do not use the `t.test` function for this question).

```
. = " # BEGIN PROMPT
p_value_10 <- NULL # YOUR CODE HERE
p_value_10
" # END PROMPT

# BEGIN SOLUTION
# find the mean and standard error for operating times
surg_data %>% summarize(mean = mean(op_time),
                        se = sd(op_time)/sqrt(16))
```

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

```
# calculate the t test statistic
t_stat <- (273.9454 - 240)/7.305622

#calculate the p-value
p_value_10 <- pt(t_stat, df = 15, lower.tail = F)
# END SOLUTION
```

```
test_that("p10a", {
  expect_true(all.equal(p_value_10, 0.000158235, tol = 0.01))
  print("Checking: p-value is correct")
})
```

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

11. What are your conclusions in the context of this question?

The p-value is 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. Construct a 95% CI for the mean operating time (in *hours*). Round your lower and upper bounds to 3 decimal places and assign them to a vector called `CI_12`. Interpret the confidence interval in the context of this problem.

```
. = " # BEGIN PROMPT
CI_12 <- NULL # YOUR CODE HERE
CI_12
" # END PROMPT

# BEGIN SOLUTION
# find t critical value
t_star_12 <- qt(p = 0.975, df = 15) # df = n - 1

# lower bound
lower_bound_12 <- round((273.9454 - t_star_12*7.305622)/60, 3) # divide by 60 to get hours

# upper bound
upper_bound_12 <- round((273.9454 + t_star_12*7.305622)/60, 3)

CI_12 <- c(lower_bound_12, upper_bound_12)
# END SOLUTION
```

```
test_that("p12a", {
  expect_true(all.equal(CI_12[1], 4.306, tol = 0.01))
  print("Checking: Lowerbound is correct")
})
```

```
## [1] "Checking: Lowerbound is correct"
## Test passed
```

```
test_that("p12b", {
  expect_true(all.equal(CI_12[2], 4.825, tol = 0.01))
  print("Checking: Upperbound is correct")
})
```

```
## [1] "Checking: Upperbound is correct"
## Test passed
```

Thus, using a method that includes the null value 95 times out of 100, our 95% CI is 4.306 hours to 4.825 hours.

13. 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 ( $\mu_d$  is the mean difference). Which of the following could be a possible 95% confidence interval for  $\mu_d$ ? Assign the letter corresponding to your answer to p13. For example, if your answer is a, assign `p13 <- "a"`.

- a. -2.30 to -0.70
- b. -1.20 to 0.90
- c. 1.50 to 3.80
- d. 4.50 to 6.90

```
. = " # BEGIN PROMPT
p13 <- NULL # YOUR CODE HERE
p13
" # END PROMPT
```

```
# BEGIN SOLUTION
p13 <- "b"
# END SOLUTION
```

```
test_that("p13a", {
  expect_true(p13 == "b")
  print("Checking: p13 is the correct confidence interval")
})
```

```
## [1] "Checking: p13 is the correct confidence interval"
## Test passed
```

14. 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  $\mu_d$ . Based on the p-value which one of the following is true? Assign the letter corresponding to your answer to p14.

- a. Both a 95% CI and a 99% CI will contain 0.
- b. A 95% CI will contain 0, but a 99% CI will not.
- c. A 95% CI will not contain 0, but a 99% CI will.
- d. Neither a 95% CI nor a 99% CI interval will contain 0.

```
. = " # BEGIN PROMPT
p14 <- NULL # YOUR CODE HERE
p14
" # END PROMPT
```

```
# BEGIN SOLUTION
p14 <- "c"
# END SOLUTION
```

```
test_that("p14a", {
  expect_true(p14 == "c")
  print("Checking: p14 is the correct answer choice")
})
```

```
## [1] "Checking: p14 is the correct answer choice"
## Test passed
```