Problem Set 5: Normal and Binomial Distribution

Your name and student ID

September 26, 2022

```
##
## Attaching package: 'dplyr'

## The following object is masked from 'package:testthat':
##
##     matches

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

Instructions

- Solutions will be released on Friday, September 30th.
- 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.

Part 1: Pregnancy Length Probabilities

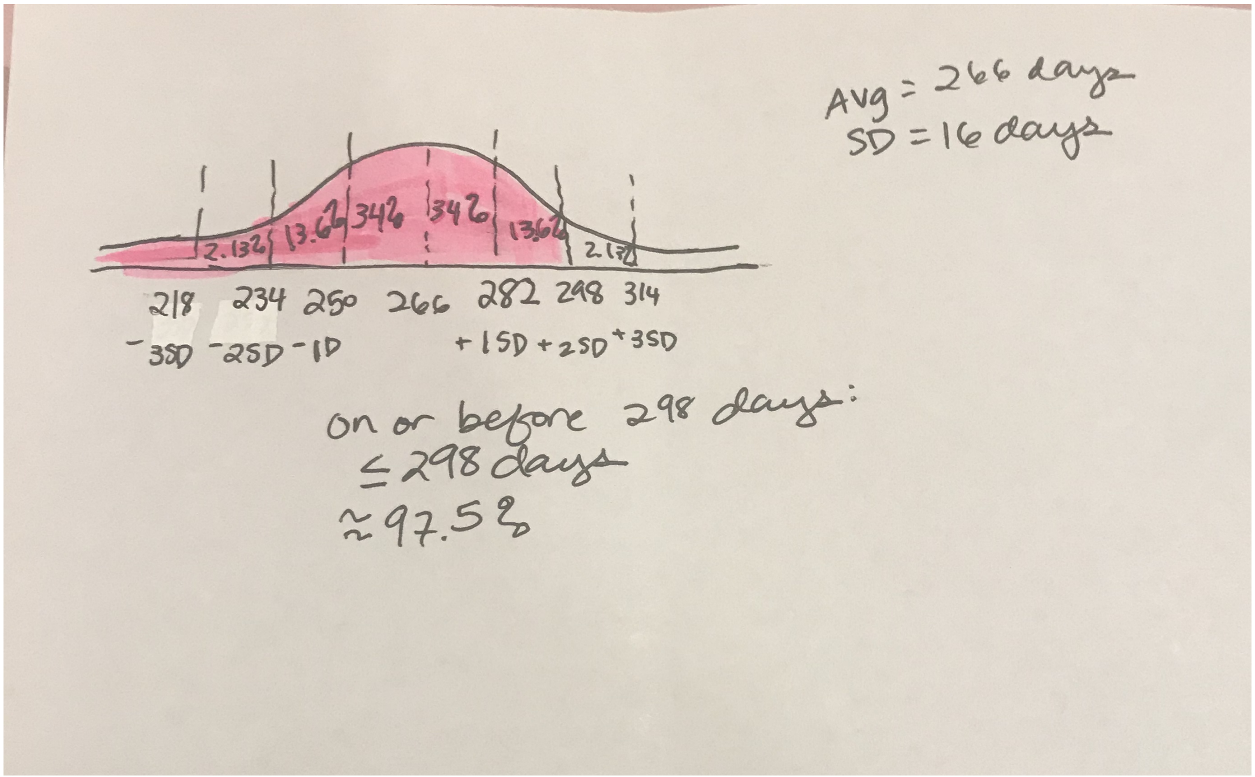
An average pregnancy for humans lasts 266 days, with a standard deviation of 16 days. Assume that human pregnancies are Normally distributed.

1. [1 point] Approximately what proportion of births are expected to occur on or before 298 days? To aid your answer, hand-draw (or use any software to sketch) a Normal curve and add dashed lines at the mean +/- 1SD, 2SD and 3SD. Calculate the proportion of births occurring on or before 298 days by shading this region under the curve. You shouldn't need to use R to perform any calculations for this question. Round the proportion to one decimal place.

(Use the code chunk below to include an image file of your drawing. To do this you need to delete the hashtag, upload the image to Datahub into the `src` directory and replace the file name with your file name. JPG or PNG will both work).

```
#knitr::include_graphics("src/Your-file-name.JPG")
```

```
# BEGIN SOLUTION NO PROMPT
knitr::include_graphics("src/A5_Normal-a.JPG")
```



Avg = 266 days
SD = 16 days

on or before 298 days:
≤ 298 days
≈ 97.5%

```
# END SOLUTION
```

Students should draw the Normal density curve with the pregnancy days corresponding to the mean and the mean +/- 1, 2, and 3 SD. They should notice that mean + 2SD = 298. They know that 95% of the data is between the mean +/- 2 SD, which implies that 2.5% of the data is above the mean + 2SD, or approximately 97.5% of the data is below 298 days.

2. [1 point] Check your answer from part a) using R code. Create a vector called p2 that stores 2 values: your answer from part a and the absolute difference between your answer from a and the exact probability that you calculated with code.

```
. = " # BEGIN PROMPT
p2 <- NULL # YOUR CODE HERE
p2
" # END PROMPT


# BEGIN SOLUTION

p2 <- c(pnorm(q = 298, mean = 266, sd = 16),
       abs(0.975 - pnorm(q = 298, mean = 266, sd = 16)))
# END SOLUTION
```

```
test_that("p2a", {
  expect_true(all.equal(p2[1], pnorm(q = 298, mean = 266, sd = 16), tol = 0.001))
  print("Checking: first value of p2 is correct")
})
```

```
## [1] "Checking: first value of p2 is correct"
## Test passed
```

```
test_that("p2b", {
  expect_true(all.equal(p2[2], abs(0.975 - pnorm(q = 298, mean = 266, sd = 16)), tol = 0.001))
  print("Checking: second value of p2 is correct")
})
```

```
## [1] "Checking: second value of p2 is correct"
## Test passed
```

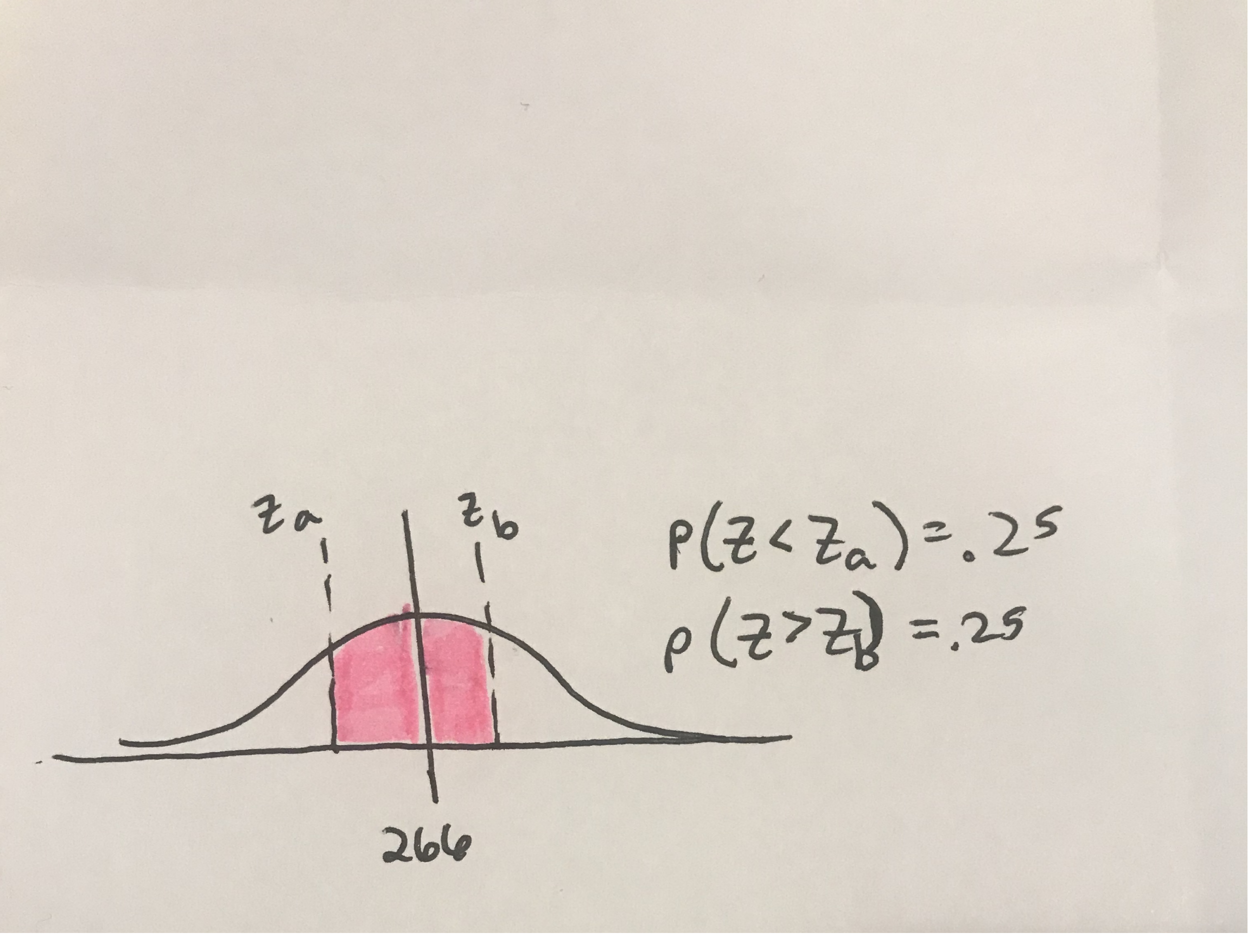
3. [1 point] What is the range, in days, that the middle 50% of pregnancies last? To aid your answer, hand-draw (or use any software to sketch) a Normal curve and shade in the area that the middle range represents. Then use R to calculate this middle range. Round the lower and upper bound of the range each to two decimal places.

(Use the code chunk below to include an image file of your drawing. To do so you need to delete the hashtag, upload the image to Datahub into the `src` directory and replace the file name with your file name. JPG or PNG will both work.)

```
#knitr::include_graphics("src/Your-file-name.JPG")

# your code here
```

```
# BEGIN SOLUTION NO PROMPT
knitr::include_graphics("src/A3_Normal.JPG")
```



```
# END SOLUTION
```

```
# BEGIN SOLUTION NO PROMPT
# want the quantile (aka percentile) such that 25% of the data is below it
qnorm(p = 0.25, mean = 266, sd = 16)
```

```
## [1] 255.2082
```

```
# the upper bound is the quantile (aka percentile) such that 75% of the data is
# below it
qnorm(p = 0.75, mean = 266, sd = 16)
```

```
## [1] 276.7918
```

```
# END SOLUTION
```

Thus, the range is from 255.21 days to 276.79 days.

Part 2: Assessing Normality and Interpreting QQ Plots

The number of trees for nine plots of land, each of 0.1 hectare, have been recorded. They are: 18, 4, 22, 15, 18, 19, 22, 12, 12. Are these data Normally distributed?

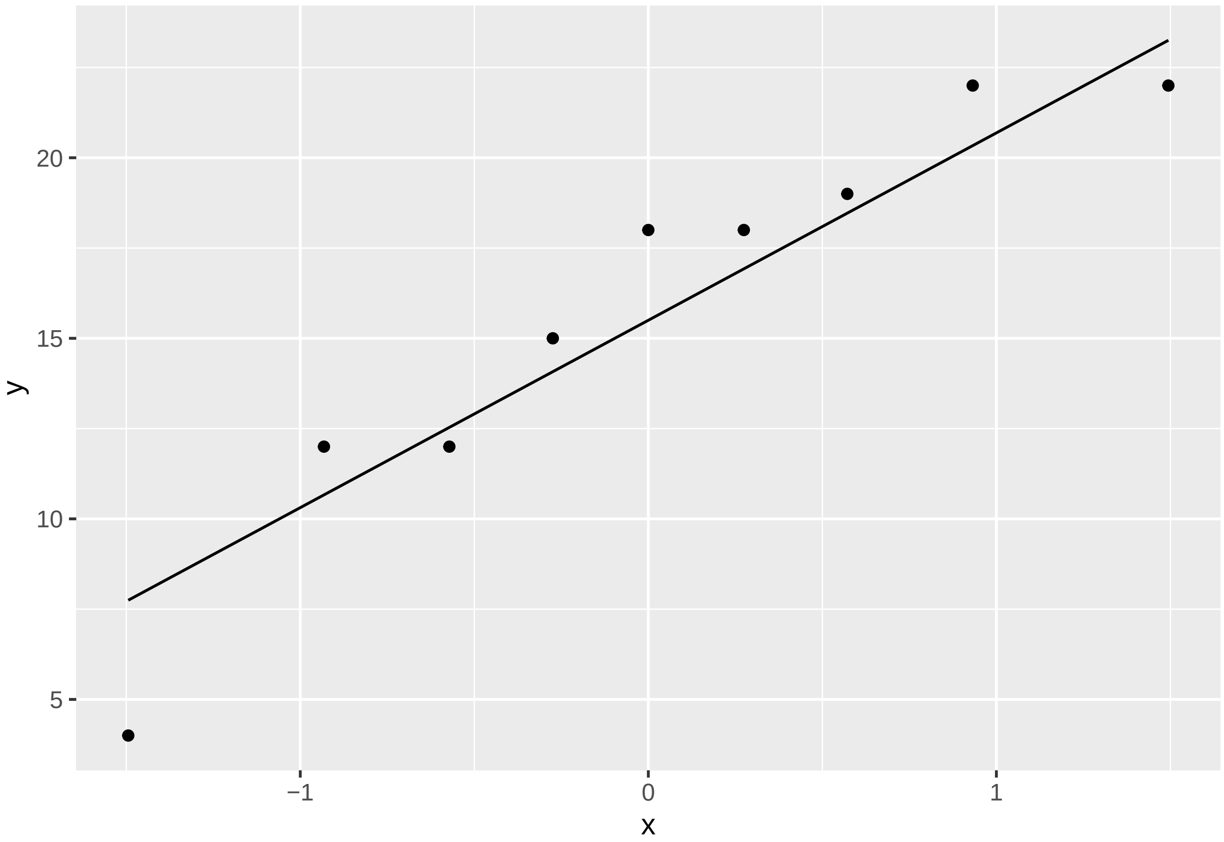
4. [1 point] Make a Normal quantile plot for these data using R. Remember, to make a ggplot of these data, you need to first input the data as a vector and then convert that vector to a dataframe. Example code has been provided to help get you started. After making the plot, assess whether the data appear to approximately follow a Normal distribution.

```r
library(tidyverse)

# example code
counts <- c(1, 2, 3)
tree_data <- data.frame(counts)

# your code here
```

```r
# BEGIN SOLUTION NO PROMPT
counts <- c(18, 4, 22, 15, 18, 19, 22, 12, 12)
tree_data <- data.frame(counts)
ggplot(tree_data, aes(sample = counts)) + geom_qq() + geom_qq_line()
```



```r
# END SOLUTION
```

The QQ Plot of the data quantiles against Normal quantiles is roughly linear, so we believe the data approximately follows a Normal distribution.

Part 3: Conducting a study about general anxiety disorder

Suppose that a new treatment for general anxiety disorder has undergone safety and efficacy trials and, based on these data, 30% of patients with general anxiety disorder are expected to benefit from the new treatment. You are conducting a follow-up study and have enrolled 8 participants with general anxiety disorder so far. These patients do not know each other and represent individuals who responded to a call for study participants that they saw on a flier on campus.

5. [1 point] Let X represent the number of enrolled patients who benefit from the treatment. Does X meet the assumptions of a Binomial distribution? Thoroughly explain why or why not.

Solution: Yes, because: Fixed number of obsevrations (8) All the observations appear to be independent (they don't know each other) Each is either a success (benefit) or failure (no benefit) The probability of success is the same for each person

6. [1 point] Using one of the distributions whose assumptions X meets, calculate (by hand) the probability that exactly 5 participants will benefit from the treatment. Show your work.

```
. = " # BEGIN PROMPT
p6 <- NULL # YOUR CODE HERE
p6
" # END PROMPT


# BEGIN SOLUTION
# ${n \choose k}p^k(1-p)^{n-k}$
# ${8 \choose 5}0.3^5(1-0.3)^{8-5}$ = 0.04667544
p6 <- 0.04667544
# END SOLUTION
```

```
## Test ##
test_that("p6a", {
  expect_true(between(p6, 0, 1))
  print("Checking: p6 is a value between 0 and 1")
})
```

```
## [1] "Checking: p6 is a value between 0 and 1"
## Test passed
```

```
## Test ##
test_that("p6b", {
  expect_true(all.equal(p6, 0.04667544, tol = 0.01))
  print("Checking: p6 is the correct probability")
})
```

```
## [1] "Checking: p6 is the correct probability"
## Test passed
```

7. [1 point] Confirm your previous calculation using an R function and store your answer to p7.

```
. = " # BEGIN PROMPT
p7 <- NULL # YOUR CODE HERE
p7
" # END PROMPT


# BEGIN SOLUTION
p7 <- dbinom(x = 5, size = 8, prob = 0.3)
# END SOLUTION
```

```
## Test ##
test_that("p7a", {
  expect_true(between(p7, 0, 1))
  print("Checking: p7 is a value between 0 and 1")
})
```

```
## [1] "Checking: p7 is a value between 0 and 1"
## Test passed
```

```
test_that("p7b", {
  expect_true(all.equal(p7, dbinom(x = 5, size = 8, prob = 0.3), tol = 0.1))
  print("Checking: p7 is the correct probability")
})
```

```
## [1] "Checking: p7 is the correct probability"
## Test passed
```

8. [1 point] Calculate (by hand) the probability that 6 or more participants will benefit from the treatment. Show your work.

```
. = " # BEGIN PROMPT
p8 <- NULL # YOUR CODE HERE
p8
" # END PROMPT


# BEGIN SOLUTION

#${8 \choose 6}0.3^6(1-0.3)^{8-6} + {8 \choose 7}0.3^7(1-0.3)^{8-7} + {8 \choose 8}0.3^8(1-0.3)^{8-8}$

p8 <- 0.01129221
# END SOLUTION
```

```
## Test ##
test_that("p8a", {
  expect_true(between(p8, 0, 1))
  print("Checking: p8 is a value between 0 and 1")
})
```

```
## [1] "Checking: p8 is a value between 0 and 1"
## Test passed
```

```
test_that("p8b", {
  expect_true(all.equal(p8, 0.01129221, tol = 0.1))
  print("Checking: p8 is the correct probability")
})
```

```
## [1] "Checking: p8 is the correct probability"
## Test passed
```

9. [1 point] Confirm your previous calculation using the function `pbinom()` and store your answer to `p9`.

```
. = " # BEGIN PROMPT
p9 <- NULL # YOUR CODE HERE
p9
" # END PROMPT


# BEGIN SOLUTION
p9 <- 1 - pbinom(q = 5, size = 8, prob = 0.3)
# END SOLUTION
```

```
## Test ##
test_that("p9a", {
  expect_true(between(p9, 0, 1))
  print("Checking: p9 is a value between 0 and 1")
})
```

```
## [1] "Checking: p9 is a value between 0 and 1"
## Test passed
```

```
test_that("p9b", {
  expect_true(all.equal(p9, 1 - pbinom(q = 5, size = 8, prob = 0.3), tol = 0.1))
  print("Checking: p9 is the correct probability")
})
```

```
## [1] "Checking: p9 is the correct probability"
## Test passed
```

10. [1 point] Re-confirm your previous calculation, this time using the function `dbinom()`, and store your answer to `p10`.

```
. = " # BEGIN PROMPT
p10 <- NULL # YOUR CODE HERE
p10
" # END PROMPT


# BEGIN SOLUTION
p10 <- dbinom(x = 6, size = 8, prob = 0.3) +
  dbinom(x = 7, size = 8, prob = 0.3) +
  dbinom(x = 8, size = 8, prob = 0.3)
# END SOLUTION
```

```
test_that("p10a", {
  expect_true(between(p10, 0, 1))
  print("Checking: p10 is a value between 0 and 1")
})
```

```
## [1] "Checking: p10 is a value between 0 and 1"
## Test passed
```

```
test_that("p10b", {
  expect_true(all.equal(p10, dbinom(x = 6, size = 8, prob = 0.3) +
  dbinom(x = 7, size = 8, prob = 0.3) +
  dbinom(x = 8, size = 8, prob = 0.3), tol = 0.1))
  print("Checking: p10 is the correct probability")
})
```

```
## [1] "Checking: p10 is the correct probability"
## Test passed
```

11. [1 point] Interpret the binomial coefficient, $\binom{8}{7}$, in the context of this study. Write out all the possible combinations to achieve $\binom{8}{7}$.

$\binom{8}{7}$ is the number of ways to have 7 individuals benefitting out of 8 study participants. There are eight possible ways to see 7 successes across 8 individuals:

11111110 11111101 11111011 11110111 11101111 11011111 10111111 01111111

12. [1 point] Calculate the number of patients you would expect to benefit from the treatment. Then calculate the standard deviation of this estimate. Write a sentence to interpret the mean. If the mean is not a whole number, what whole number is most probable?

$\mu = np = 8 * 0.3 = 2.4$

$\sigma = \sqrt{np \times (1 - p)} = 1.3$

We expect 2.4 patients to benefit out of the 8. An average of 2.4 implies that seeing two patients benefit is the most probable number (because 2.4 is closer to 2 than it is to 3).

13. [1 point] Should you apply a Normal approximation to these data using the μ and σ you calculated in the last question? Why or why not?

No, because $np = 2.4$ is much smaller than 10, which is the rule of thumb threshold we used to decide whether we should apply the Normal approximation.

Late Pre-Term Birth Weights (From Baldi and Moore, 3E question 11.32, 4E question 11.34)

How much of a difference do a couple of weeks make for a baby's birthweight? Late preterm babies are born with 35 to 37 weeks of completed gestation. The distribution of birth weight (in grams) or late preterm babies is approximately normally distributed with a mean of 2750 grams and a standard deviation of 560 grams, N(2750,560).

14. [1 point] What is the 25th percentile of the birthweights for late-preterm term babies?

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

# BEGIN SOLUTION
p14 <- qnorm(0.25, mean = 2750, sd = 560)
# END SOLUTION
```

```
test_that("p14a", {
  expect_true(between(p14, 2000, 2400))
  print("Checking: p14 is a birthweight in grams")
})
```

```
## [1] "Checking: p14 is a birthweight in grams"
## Test passed
```

```
test_that("p14b", {
  expect_true(all.equal(p14, qnorm(0.25, mean = 2750, sd = 560), tol = 0.01))
  print("Checking: p14 is the correct weight for the 25th percentile")
})
```

```
## [1] "Checking: p14 is the correct weight for the 25th percentile"
## Test passed
```

15. [1 point] What is the 90th percentile of the birthweights for late-preterm babies?

```
. = " # BEGIN PROMPT
p15 <- NULL # YOUR CODE HERE
p15
" # END PROMPT

# BEGIN SOLUTION
p15 <- qnorm(0.9, mean = 2750, sd = 560)
# END SOLUTION
```

```
test_that("p15a", {
  expect_true(between(p15, 3000, 3500))
  print("Checking: p15 is a birthweight in grams")
})
```

```
## [1] "Checking: p15 is a birthweight in grams"
## Test passed
```

```
test_that("p15b", {
  expect_true(all.equal(p15,qnorm(0.9, mean = 2750, sd = 560), tol = 0.01))
  print("Checking: p15 is the correct weight for the 90th percentile")
})
```

```
## [1] "Checking: p15 is the correct weight for the 90th percentile"
## Test passed
```

16. [1 point] What is the range of the middle 50% of birthweights for late-preterm babies?

```
. = " # BEGIN PROMPT
p16 <- NULL # YOUR CODE HERE
p16
" # END PROMPT

# BEGIN SOLUTION
p16 <- c(qnorm(0.25, mean = 2750, sd = 560), qnorm(0.75, mean = 2750, sd = 560))
# END SOLUTION
```

```
test_that("p16a", {
  expect_true(all.equal(p16[1], qnorm(0.25, mean = 2750, sd = 560), tol = 0.01))
  print("Checking: first value of p16 is correct")
})
```

```
## [1] "Checking: first value of p16 is correct"
## Test passed
```

```
test_that("p16b", {
  expect_true(all.equal(p16[2], qnorm(0.75, mean = 2750, sd = 560), tol = 0.01))
  print("Checking: second value of p16 is correct")
})
```

```
## [1] "Checking: second value of p16 is correct"
## Test passed
```

17. Think back to lab05 when we studied the distribution of full-term birthweights N(3350,440). Compare the percentiles you calculated above between full term babies and late-preterm babies. What do you notice?

Note that the larger standard deviation for the late-preterm babies, 560 g vs 440 g, makes the ranges bigger and the percentiles farther from the mean. Since the 25th percentile for the late-preterm births is 2372.3 g, well below 2500 g, we know that more than 25% of the late-preterm babies are low birth weight.

Drosophila (From Baldi and Moore, 3E questions 11.20 and 11.22, 4E question 11.23)

The common fruit fly, Drosophila melanogaster, is the most studied organism in genetic research because it is small, easy to grow, and reproduces rapidly. The length of the thorax (where the wings and legs attach) in a population of male fruit flies is approximately Normal with mean 0.800 millimeters (mm) and standard deviation 0.078 mm.

18. [1 point] Choose a male fruit fly at random. Calculate the probability that the fly you choose has a thorax longer than 1 mm (convert to a percentage and round to two decimal places).

```r
. = " # BEGIN PROMPT
p18 <- NULL # YOUR CODE HERE
p18
" # END PROMPT

# BEGIN SOLUTION
p18 <- round(pnorm(1, mean = 0.8, sd = 0.078, lower.tail=FALSE)*100,2)
# END SOLUTION
```

```r
test_that("p18a", {
  expect_true(between(p18, 0, 100))
  print("Checking: p18 is a value between 0 and 100")
})
```

```
## [1] "Checking: p18 is a value between 0 and 100"
## Test passed
```

```r
test_that("p18b", {
  expect_true(all.equal(p18, .52, tol = 0.01))
  print("Checking: p18 is the correct percent")
})
```

```
## [1] "Checking: p18 is the correct percent"
## Test passed
```