Homework 4

Your name and student ID

Today's date

- Solutions will be released on Tuesday, September 29.
- This semester, homework assignments 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! **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.

[12 points] Part 1: Simulating birth defect data and sampling from an infinitely large population

The Center for Disease Control and Prevention (CDC) estimates that 1 in every 33 infants is born with a birth defect in the United States each year.

1) [3 points] Define a random variable for "birth defect". Write down the probability model for the random variable. Round each percentage to two decimal places (e.g., 0.43224 would be rounded to 43.22%). Is the sample space discrete or continuous?

[TODO: Replace the text with your list of proportions and your explanation.]

You might want to use the table template below to write out your probability model. If not, then delete it. *Knit now* to see how this table is rendered in your PDF.

Tables	Are	Cool
yadi	type stuff	X
yadi	more stuff	Y
yada	etc	Z

BEGIN SOLUTION

[1 point] Let BD represent the event "birth defect".

[1 point] Then the probability model is:

Birth defect	BD	BD'
Probability	1/33 = 3.03%	32/33 = 97.0%

[1 point] The sample space is discrete.

END SOLUTION

2) [2 points] Simulate data that equals 0 if there is no birth defect and equals 1 if there is a birth defect. Simulate this data for 200 births at a local hospital. Be sure to use the risk of birth defect from part a). Assign your simulated output the name `sim_01`. Print your simulated births to the screen.

Before you run your simulation, we will "set the seed". We all will set the seed to 100. This means that everyone's simulation will yield the exact same dataset.

```
set.seed(100)
# execute this line before you write your simulation code.
# only execute the set.seed() function one time.
```

```
sim_01 <- "YOUR ANSWER HERE"
sim_01
```

```
## [1] "YOUR ANSWER HERE"
```

```
# BEGIN SOLUTION
sim_01 <- rbinom(200, 1, prob = 1/33)
sim_01
```

```
##   [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
##  [38] 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
##  [75] 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0
## [112] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [149] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [186] 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0
```

```
# END SOLUTION
check_problem2()
```

```
## [1] "Checkpoint 1 Passed: You made an integer vector"
## [1] "Checkpoint 2 Passed: Correct number of elements!"
##
## Problem 2
## Checkpoints Passed: 2
## Checkpoints Errored: 0
## 100% passed
## --------
## Test: PASSED
```

Notice that `sim_01` is not a data frame, rather it is a vector of numbers. The following code stores `sim_01` as a data frame and changes its variable name. Run this code and view `sim_01` in the Viewer pane.

```
library(dplyr)
sim_01 <- as.data.frame(sim_01) # watch what happens to sim_01 in your environment
names(sim_01) # prints the variable names in the sim_01 data frame
```

```
## [1] "sim_01"
```

```r
sim_01 <- sim_01 %>% rename(birth_defect = sim_01)
```

3) [2 points] Write code to determine the number of birth defects that occurred in your simulation, and the corresponding proportion with birth defects. Assign your output the name `output_01`. Print `output_01` to the screen. Hint: Use `dplyr` functions to do this.

```
output_01 <- "YOUR ANSWER HERE"
output_01
```

```
## [1] "YOUR ANSWER HERE"
```

```
# BEGIN SOLUTION
output_01 <- sim_01 %>% summarize(number = sum(birth_defect),
                                  prop = mean(birth_defect))
output_01
```

```
##   number  prop
## 1      5 0.025
```

```
# END SOLUTION
check_problem3()
```

```
## [1] "Checkpoint 1 Passed: Correct! Output_01 should be a dataframe"
## [1] "Checkpoint 2 Passed: Correct number of columns (two columns)"
## [1] "Checkpoint 3 Passed: Correct number of rows (one row)"
##
## Problem 3
## Checkpoints Passed: 3
## Checkpoints Errored: 0
## 100% passed
## --------
## Test: PASSED
```

4) [2 marks] Re-run your simulation four more times and assign the output to a unique name each time. Print to the screen the number and proportion after each run. (Basically, "recycle" the code above four times)

```
sim_02 <- "YOUR ANSWER HERE"
output_02 <- "YOUR ANSWER HERE"
output_02
```

```
## [1] "YOUR ANSWER HERE"
```

```
sim_03 <- "YOUR ANSWER HERE"
output_03 <- "YOUR ANSWER HERE"
output_03
```

```
## [1] "YOUR ANSWER HERE"
```

```
sim_04 <- "YOUR ANSWER HERE"
output_04 <- "YOUR ANSWER HERE"
output_04
```

```
## [1] "YOUR ANSWER HERE"
```

```
sim_05 <- "YOUR ANSWER HERE"
output_05 <- "YOUR ANSWER HERE"
output_05
```

```
## [1] "YOUR ANSWER HERE"
```

```
# BEGIN SOLUTION
#second run
sim_02 <- rbinom(200, 1, prob = 1/33)
sim_02 <- as.data.frame(sim_02) %>% rename(birth_defect = sim_02)
output_02 <- sim_02 %>% summarize(number = sum(birth_defect),
                                  prop = mean(birth_defect))
output_02
```

```
##   number  prop
## 1      5 0.025
```

```
#third run
sim_03 <- rbinom(200, 1, prob = 1/33)
sim_03 <- as.data.frame(sim_03) %>% rename(birth_defect = sim_03)
output_03 <- sim_03 %>% summarize(number = sum(birth_defect),
                                  prop = mean(birth_defect))
output_03
```

```
##   number prop
## 1     10 0.05
```

```r
#fourth run
sim_04 <- rbinom(200, 1, prob = 1/33)
sim_04 <- as.data.frame(sim_04) %>% rename(birth_defect = sim_04)
output_04 <- sim_04 %>% summarize(number = sum(birth_defect),
                                  prop = mean(birth_defect))
output_04
```

```
##   number prop
## 1      6 0.03
```

```r
#fifth run
sim_05 <- rbinom(200, 1, prob = 1/33)
sim_05 <- as.data.frame(sim_05) %>% rename(birth_defect = sim_05)
output_05 <- sim_05 %>% summarize(number = sum(birth_defect),
                                  prop = mean(birth_defect))
output_05
```

```
##   number prop
## 1      8 0.04
```

```r
# alternative solution (for loop or...)
# students with programming experience may choose to use a for loop or other
# iteration function to solve this question.
# END SOLUTION
check_problem4()
```

```
## [1] "Checkpoint 1 Passed: Correct! output_02 should be a dataframe"
## [1] "Checkpoint 2 Passed: Correct number of columns (two columns)"
## [1] "Checkpoint 3 Passed: Correct number of rows (one row)"
## [1] "Checkpoint 4 Passed: Correct! output_03 should be a dataframe"
## [1] "Checkpoint 5 Passed: Correct number of columns (two columns)"
## [1] "Checkpoint 6 Passed: Correct number of rows (one row)"
## [1] "Checkpoint 7 Passed: Correct! output_04 should be a dataframe"
## [1] "Checkpoint 8 Passed: Correct number of columns (two columns)"
## [1] "Checkpoint 9 Passed: Correct number of rows (one row)"
## [1] "Checkpoint 10 Passed: Correct! output_05 should be a dataframe"
## [1] "Checkpoint 11 Passed: Correct number of columns (two columns)"
## [1] "Checkpoint 12 Passed: Correct number of rows (one row)"
##
## Problem 4
## Checkpoints Passed: 12
## Checkpoints Errored: 0
## 100% passed
## --------
## Test: PASSED
```

5) [1 mark] Assign the vector p5 to the simulated proportions from each of your five simulation in *increasing* order.

```
p5 <- c("YOUR ANSWER HERE")
p5
```

```
## [1] "YOUR ANSWER HERE"
```

```
# BEGIN SOLUTION
p5 <- c(0.025, 0.025, 0.03, 0.04, 0.05)
# END SOLUTION
check_problem5()
```

```
## [1] "Checkpoint 1 Passed: You made a numeric vector"
## [1] "Checkpoint 2 Passed: You inputted 5 values"
## [1] "Checkpoint 3 Passed: Correct input for p5"
##
## Problem 5
## Checkpoints Passed: 3
## Checkpoints Errored: 0
## 100% passed
## --------
## Test: PASSED
```

6) [1 mark] Did you get close to the true value? Explain why there is variation in the proportions across the simulations.

[TODO: YOUR ANSWER HERE]

BEGIN SOLUTION

The true value is $1/33 = 0.03$ percent, which is very close to the median of the simulated values. There is variation across the simulations because the data was generated randomly, so some samples had more birth defects than others, even though they were all drawn from the same underlying distribution.

END SOLUTION

7) [1 mark] Suppose that rather than simulating 5 samples of size 200, we simulated 5 samples of size 1000. In 1-2 sentences, how would you expect the group of proportion estimates from part e) to be different? Comment both on the accuracy of these values at predicting the true value, and their variance. If you're not sure, you can re-run your simulation with a larger sample size and see how the results change to deduct the difference.

[TODO: YOUR ANSWER HERE]

BEGIN SOLUTION

The proportion estimates would be less variable (because sample size is larger) and closer to the true underlying value.

END SOLUTION

[8 points] Part 2: Probability of HIV and Hepatitis C

Approximately 1.1 million Americans have HIV and 3.5 million Americans have Hepatitis C (HCV). The number of individuals with coinfection (e.g., both HIV and HCV) is 300,000. Among individuals with HIV, approximately 25% have Hepatitis C. The total US population was approximately 321 million at the time of these statistics.

references for these stats:

- https://www.cdc.gov/hiv/basics/statistics.html
- https://www.cdc.gov/media/releases/2016/p0504-hepc-mortality.html
- https://www.cdc.gov/hepatitis/populations/hiv.htm

8) [2 points] Calculate the probability that a randomly chosen American will have HIV. Calculate the probability that a randomly chosen American will have HCV. Convert to percentages and round to two decimal places. Save these values as the vector p2a with the proportion for HIV first then HCV. Don't inclue the % in your answer.

```
p8 <- c("YOUR ANSWER HERE")
p8
```

```
## [1] "YOUR ANSWER HERE"
```

```
# BEGIN SOLUTION
p8 <- c(0.34, 1.09)
# END SOLUTION
check_problem8()
```

```
## [1] "Checkpoint 1 Passed: You made a numeric vector"
## [1] "Checkpoint 2 Passed: You inputted 2 values"
## [1] "Checkpoint 3 Passed: Correct input for p8"
##
## Problem 8
## Checkpoints Passed: 3
## Checkpoints Errored: 0
## 100% passed
## --------
## Test: PASSED
```

BEGIN SOLUTION

Let HIV represent HIV and HCV represent HCV. P(HIV) = 1.1M/321M = 0.003426791 = 0.34% P(HCV) = 3.5M/321M = 0.01090343 = 1.09%

END SOLUTION

9) [2 points] Without using the number of co-infections provided in the question, calculate the probability that someone will have both HIV and HCV. Convert to a percent rounded to two decimal places and save to object p2b. Don't include the percent in your answer.

```
p9 <- "YOUR ANSWER HERE"
p9
```

```
## [1] "YOUR ANSWER HERE"
```

```
# BEGIN SOLUTION
p9 <- 0.09
# END SOLUTION
check_problem9()
```

```
## [1] "Checkpoint 1 Passed: Correct! It is numeric"
## [1] "Checkpoint 2 Passed: Correct rounding"
## [1] "Checkpoint 3 Passed: Correct answer for p9"
##
## Problem 9
## Checkpoints Passed: 3
## Checkpoints Errored: 0
## 100% passed
## --------
## Test: PASSED
```

BEGIN SOLUTION

We know that P(HCV | HIV) = 0.25

We also know that P(HCV | HIV) = P(HCV & HIV)/P(HIV) This implies that P(HCV & HIV) = P(HCV | HIV) X P(HIV) = 0.25 X (1.1/321) = 0.0008566978 = 0.09%

END SOLUTION

10) [2 points] Are HIV and HCV infections independent? Show work to support your answer. Uncomment your selection.

```
#p10 <- "independent"
#p10 <- "not independent"
# BEGIN SOLUTION
p10 <- "not independent"
# END SOLUTION
check_problem10()
```

```
## [1] "Checkpoint 1 Passed: Correct selection"
## [1] "Checkpoint 2 Passed: Correct"
##
## Problem 10
## Checkpoints Passed: 2
## Checkpoints Errored: 0
## 100% passed
## --------
## Test: PASSED
```

BEGIN SOLUTION

If HIV and HCV were independent, then:

$P(HIV \& HCV) = P(HCV) \times P(HIV) = 0.01090343 \times 0.00342679 = 0.00003736378 = 0.003736378\%$

However, from part b) we know that P(HIV & HCV) = 0.09%, which is much higher. Thus these infections are not independent.

Alternative explanation:

We know that P(HCV) = 1.09% from part (a) and that P(HCV | HIV) = 25% from the question. If these infections were independent, these two quantities would be the same. Thus, they are not independent.

END SOLUTION

11) [2 points] In general, does P(A|B) equal P(B|A)? Calculate P(HIV | HCV) and report whether or not it is equal to P(HCV | HIV).

[TODO: YOUR ANSWER HERE]

BEGIN SOLUTION

Using information from part (b) and part (a):

[1 mark] P(HIV | HCV) = P(HIV & HCV)/P(HCV) = 0.0008566978/0.01090343 = 7.86%

[0.5 marks] The P(HIV | HCV) is 7.86% which is different from P(HCV | HIV) which is 25%.

END SOLUTION

[9 points] part 3: Screening for lung cancer

Background reading: Read pages 258-261 (and optionally 261-264) of Baldi & Moore, Edition 4. (For earlier editions, look for the section on diagnostic testing in medicine or on screening which covers sensitivity, specificity, negative predictive value, and positive predictive value).

Lung cancer is a leading cause of cancer-related deaths in the United States. Researchers examined the idea of testing all Medicare-enrolled heavy smokers for lung cancer with a computed tomography (CT) scan every year. In this population, the lifetime chance of developing lung cancer is high. In any given year, approximately 3% of heavy smokers develop lung cancer. The CT scan positively identifies lung cancer 89% of the time, and it gives a negative results for 93% of individuals who do not have lung cancer.

12) [3 points] Use probability notation to express the three proabilities cited. Make sure to define each event using a capital letter (or two). Provide the terminology for the 89% and 93% values based on your readings.

[TODO: YOUR ANSWER HERE]

BEGIN SOLUTION

Solution:

[0.5 marks] Establish RVs as letters: Let L represent the event lung cancer and CT represent a positive CT scan.

[1 mark] P(CT | L) = 0.89. This is the sensitivity of the test. [1 mark] P(CT' | L') = 0.93. This is the specificity of the test. [0.5 marks] P(L) = 0.03. This is the probability of lung cancer.

END SOLUTION

13) [3 points] What percent of CT scans in this target population would be positive? Answer this question by making either a probability tree or using absolute frequencies. Show your work.

[TODO: YOUR ANSWER HERE]

<Note: If you are writing your solutions in R markdown you may want to upload an image of a hand-drawn tree diagram (this is optional). If so use the following code, or delete if not using. Be sure to remove the option "eval = F" if using this code or it won't run when you knit the file!:>

BEGIN SOLUTION

Solution 1. Absolute frequencies. Suppose that there were 1000 people in the target population.

- 30 of them truly have lung cancer. Of these 30, 26.7 will test positive for lung cancer. The remaining 3.3 will test negatively.
- 970 of them will not have lung cancer. Of these, 902.1 will correctly test negative, will the remaining 67.9 incorrectly test positive
- The total number of positive tests is the sum of the true positives and the false positives. This is: 26.7 + 67.9 = 94.6. Thus 94.6/1000 = 9.46% of the population will test positive.

Solution 2. Probability tree.

END SOLUTION

14) [1 point] We will now solve for the probability that a Medicare-enrolled heavy smoker who gets a positive scan actually has lung cancer. Write out the probability statement for this amount.

[TODO: YOUR ANSWER HERE]

BEGIN SOLUTION

[1 mark] We are trying to solve for P(L | CT).

END SOLUTION

15) [1 point] Calculate the probability value from question 14 based on your previous work from question 13. Store the answer as a percentage rounded to one decimal place in the object p15.

```
p15 <- "YOUR ANSWER HERE"
p15
```

```
## [1] "YOUR ANSWER HERE"
```

```
# BEGIN SOLUTION
p15 <- 28.2
# END SOLUTION
check_problem15()
```

```
## [1] "Checkpoint 1 Passed: Correct, it's numeric"
## [1] "Checkpoint 2 Passed: Correct, it is a percentage"
## [1] "Checkpoint 3 Passed: Correct rounding"
## [1] "Checkpoint 4 Passed: Correct"
##
## Problem 15
## Checkpoints Passed: 4
## Checkpoints Errored: 0
## 100% passed
## --------
## Test: PASSED
```

BEGIN SOLUTION

Based on absolute frequencies: Of the 94.6 people with a positive scan, 26.7 will actually have lung cancer. Thus, 26.7/94.6 = 28.2% of the positive scans will actually have lung cancer.

Using the probability tree: 0.0946 of the population will have a positive scan, of which 0.0267 will actually have lung cancer. Thus 0.0267/0.0946 = 28.2% of the positive scans will actually have lung cancer.

END SOLUTION

16) [1 point] What term from your reading does this value represent? Store the answer in object p16.

```
p16 <- "YOUR ANSWER HERE"
p16
```

```
## [1] "YOUR ANSWER HERE"
```

```
check_problem16
```

```
## function ()
## {
##     problem_num <- 16
##     max_scores[problem_num] <<- 1
##     num_tests <<- 2
##     problem_types[problem_num] <<- "autograded"
##     problem_names[problem_num] <<- sprintf("Problem %d", problem_num)
##     tests_failed <<- num_tests
##     checkpoint(checkpoint_number = 1, test = class(p16) == "character",
##         correct_message = "Correct, it is character", error_message = "Is your answer a character?")
##     checkpoint(checkpoint_number = 2, test = grepl("po.*pred.*value|ppv",
##         p16, ignore.case = TRUE))
##     if (tests_failed == 0 && problem_types[problem_num] != "free-response") {
##         scores[problem_num] <<- max_scores[problem_num]
##     }
##     else {
##         scores[problem_num] <<- 0
##     }
##     assert_that(tests_failed <= num_tests, tests_failed >= 0,
##         msg = sprintf("Did you set your num_test correctly for problem %d?",
##             problem_num))
##     return_score(problem_num, num_tests, tests_failed)
## }
## <bytecode: 0x000000001e1f4e88>
```

```
# BEGIN SOLUTION
p16 <- "Positive Predictive Value"
# END SOLUTION
check_problem16()
```

```
## [1] "Checkpoint 1 Passed: Correct, it is character"
## [1] "Checkpoint 2 Passed"
##
## Problem 16
## Checkpoints Passed: 2
## Checkpoints Errored: 0
## 100% passed
## --------
## Test: PASSED
```

Check your score

Click on the middle icon on the top right of this code chunk (with the downwards gray arrow and green bar) to run all your code in order. Then, run this chunk to check your score.

```
# Just run this chunk.
total_score()
```

```
##                      Test Points_Possible          Type
## Problem 1  NOT YET GRADED               3 free-response
## Problem 2           PASSED               2    autograded
## Problem 3           PASSED               2    autograded
## Problem 4           PASSED               2    autograded
## Problem 5           PASSED               1    autograded
## Problem 6  NOT YET GRADED               1 free-response
## Problem 7  NOT YET GRADED               1 free-response
## Problem 8           PASSED               2    autograded
## Problem 9           PASSED               2    autograded
## Problem 10          PASSED               2    autograded
## Problem 11 NOT YET GRADED               1 free-response
## Problem 12 NOT YET GRADED               3 free-response
## Problem 13 NOT YET GRADED               3 free-response
## Problem 14 NOT YET GRADED               1 free-response
## Problem 15          PASSED               1    autograded
## Problem 16          PASSED               1    autograded
```