Lab 6: Probability Distributions and Random Variables

Question 1

We are going to revisit an example in the history of probability: one of the problems posed by French gambler Antoine Gombaud, better known by his *nome de plume* "Chevalier De Méré". As you may have read about in your notes, he was not a nobleman, despite his name, but an amateur mathematician and an avid gambler. One of De Méré's favorite games of chance was the following game of dice:

- You roll a fair, six-sided die four times;
- and you win if you get at least one six.

While it is possible to find—in an analytical way—the probability of winning in this type of game, in this lab, we are going to use simulations to approximate such a probability!

part a

Create a numeric vector `die` with values 1,2,3,4,5,6.

part b

Using `die`, write code to simulate one round of De Méré's game (four die rolls) and save the results into the object `four_rolls`.

part c

Using the `four_rolls` object, write a line of code to obtain the number of sixes rolled.

part d

Write code to repeat the process of *parts b-c* 10,000 times. That is, write code that will create a vector of 10,000 entries, where entry 1 contains the number of sixes rolled in round 1, entry 2 contains the number of sixes rolled in round 2, and so on. *Hint: is there a function we've introduced this week that can help make the code efficient?* Save the resulting vector into the object `count_sixes`.

part e

Using `count_sixes`, *approximate* the probability of getting at least one six in four rolls.

part f

Again, De Méré's game involves rolling a fair, six-sided die four times. Consider the number of times that a six is rolled across the four times. What probability distribution does this follow? State the name and any parameter(s).

part g

Use the probability distribution you identified in **part f** to *calculate* the probability of getting at least one six in four rolls. Your answer in *part e* should be close to what you get here. *Hint: is there an R function(s) in the Random Variables tutorial that can help you with this?*

Question 2

In the following question, we explore sampling from a Hypergeometric distribution in R using the `rhyper()` function. Specifically, I've tried to use the Hypergeometric distribution to simulate drawing cards from a standard 52-card deck and counting the number of spades (♠), similarly to how is done in the problem set. Feel free to use the help file for `rhyper()` while completing this question.

part a

My first attempt to write code is going strangely, as you can see by the output below. Explain what is going wrong, and how to fix it! Answer in two to three sentences.

```
rhyper(m = 13, n = 39, k = 60, nn = 1)
```

```
[1] NA
```

part b

This time, I'd like to draw 5 cards from the 52-card deck. My result is a bit more promising than previously, based on the output; however, is it correct? Explain your answer in two to three sentences.

```
rhyper(m = 13, n = 52, k = 5, nn = 1)
```

```
[1] 1
```

Question 3

We'll now revisit the multiple choice test question from your *Probability Distributions* worksheet.

- For this problem, the test will only have 10 multiple choice questions.
- Recall that you haven't studied at all for this test, and decide to answer each of the questions by picking one of the four answer choices at random, independently of your answers to the other questions.

part a

Let X be the number of correct guesses you make across the ten questions. What probability distribution does X follow? State the name of the distribution, along with its parameters.

part b

Create a data frame called `q2` with two columns:

- The first column contains, from least to greatest, the possible values of X.
- The second column contains $P(X = x)$ for each possible value of X. *Hint: might there be an R function to help you compute these?*

part c

Using `q2` and `ggplot()` , create a probability histogram for X.

- Tip: notice that the x-axis is treated continuously even though X is a discrete random variable. To fix this, you may use the `scale_x_continous()` layer with its `breaks` argument after your `geom` layer. For example, if your random variable has a range of 0,1,2,3,4,5: `scale_x_continuous(breaks = seq(from = 0, by = 1, to = 5)).` will do the trick.

part d

Tinker with your code to regenerate the plot for the following success probabilities: 0.1, 0.2, 0.3, ..., all the way up to 0.9. *You do not need to copy your code down.* How does the shape of the probability histogram change with the success probability, p? Use what you've seen to answer this question in two to three sentences.

Question 4

In this question, you'll use what you worked on previously to make *empirical histograms* for a few different *Poisson distributions*. We'll make empirical histograms rather than probability ones because the range of a Poisson random variable is infinite, as you may have seen in the notes!

part a

What does the Poisson distribution's parameter λ, represent? Answer in one to two sentences.

part b

Let Y be a Poisson random variable with parameter $\lambda = 2$. Generate a vector of 10,000 draws from Y, save it into the object `y_draws` , and create a one-column data frame called `q4` as the lone column.

part c

Use the `dplyr` function `summarise()` to create a two-column data structure:

- The first column should contain each unique value that appears among the 10,000 draws.
- The second column should contain the number of times each unique value appears.

Save this data structure back into the object `q4`.

part d

Add a column to the new `q4` data structure called `y_props` which contains the *proportion* of times that each unique value appears among the 10,000 draws.

part e

Using `q4` and `ggplot()` , create an empirical histogram for Y.

part f

Tinker with your code to regenerate the plot for the following values of λ: 4, 6, 8, 10. You do not need to copy your code down again.

How does the shape of the probability histogram change with the rate parameter λ? Use what you've seen to answer this question in two to three sentences.

x ## Last Question

Will you ensure that your submission to Gradescope…

1. is of a pdf generated from a qmd file,
2. has all of your code visible to readers,
3. and assigns each of the questions to all pages that show your work for that question?

(This one is easy! Just answer "yes" or "no")