BAS 471 Spring 2023 Homework on Unit 5 Parts 1-3 - Probability Models for Counts

Ryan Curling

2/27/2023

Note: these are your homework problems

Reminder about collaboration policy. You can develop a common set of R code with you and your friends. However, anything that is written interpretation, i.e., anything that follows a **Response:** needs to be written up in your own words. Homeworks that look to be near copy/pastes of each other will receive substantially reduced credit.

Question 1 - Probability distribution with a finite number of possible values

An Instagram influencer has an online store to sell their "merch". The store sells a total of 60 different items. You are asked to model the number of different items purchased by a subset of the site's "best" customers (those who have bought at least 5 different items).

Let X be the number of different items purchased by these customers. The possible values for X are between 5 and 60. You would like to use the following shape as the basis of your probability model:

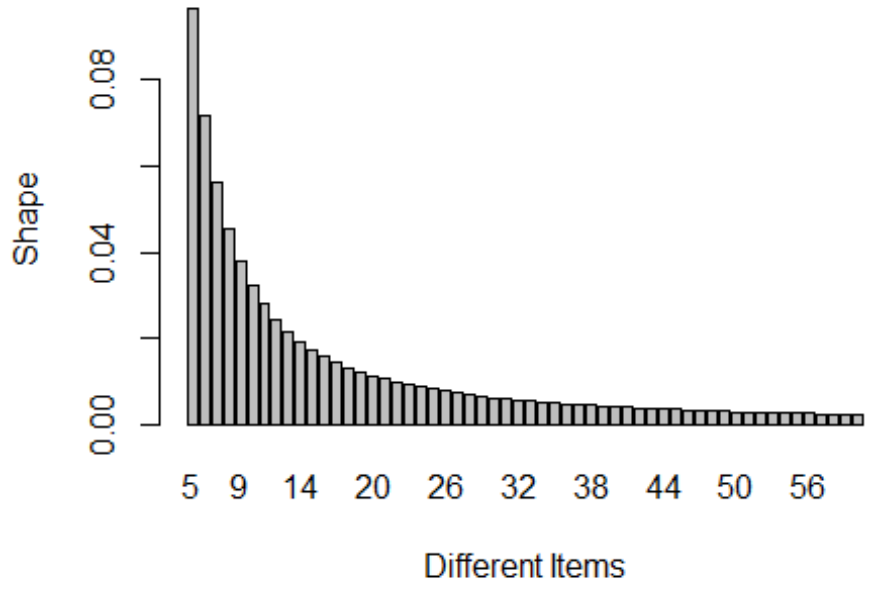
$$Shape = \frac{2+x^2}{\sqrt{1+x^7}} \qquad x = 5, 6, \ldots, 59, 60$$

a. Define a vector x that contains the possible values of X. Define a vector `shape` that gives the "shape values" (relative frequencies) of the values in x. Then, make a barplot to visualize the shape (have the horizontal axis be named "Different Items" and the vertical axis be named "Shape").

```
# Define the vector x with possible values of X
x <- 5:60

# Define the vector shape with shape values
shape <- (2 + x^2) / sqrt(1 + x^7)

# Create a barplot to visualize the shape
barplot(shape, names.arg = x, xlab = "Different Items", ylab = "Shape")
```



b. Explain why the values in `shape` (which are all between 0-1) cannot be used as the probabilities in your model.

Response: The values in 'shape' represent the relative frequencies, not the probabilities. The values in 'shape' need to be normalized in order to add up to 1 , which will represent probability.

```
# Normalize the shape vector
p <- shape / sum(shape)

# Verify that the sum of the probabilities is equal to 1
sum(p)
## [1] 1
```

c. Since there are a finite number of possible values of `x`, we can convert the `shape` vector into a vector of probabilities using the standard conversion method! Do so, defining a vector `p` whose elements give the probabilities of each value of `x`. Show that the sum of the elements in `p` equals 1, and print to the screen the value of `p[16]` (it should be 0.01599431).

```


# Generate a random sample of size 1000 from the probability distribution p
sample_data <- sample(x, size = 1000, replace = TRUE, prob = p)

# Print the summary statistics of the sample
summary(sample_data)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    5.00    7.00   11.00   16.02   22.00   60.00
sum(p)
## [1] 1
p[16]
## [1] 0.01599431
```

d. Using `which`, `sum`, and various logical conditions, determine the following probabilities.

- P(X=14); about 2.7%
- P(X < 21); about 73.7%
- P(X is at least 40 or between 25-30 or less than 10); about 58.5%; hint, you'll want three logical conditions separated by "or"s
- P(X is an even number); about 46.1%; hint, use a `seq` command
- P(X is between 6 and 10); 6 and 10 are included; about 34.7%

```
#P(X=14)
p.x14 <- p[which(x==14)]
p.x14
## [1] 0.02745122

#P(X < 21)
p.x21 <- sum(p[which(x<21)])
p.x21
## [1] 0.7367205


#P( X is at least 40 or between 25-30 or less than 10 )

p.x3 <- sum(p[which(x >= 40 | (x >= 25 & x <= 30) | x < 10)])
p.x3
## [1] 0.5854816

#P(X is an even number)

p.xeven <- sum(p[which(x %% 2 == 0)])
p.xeven
## [1] 0.4609054

#P(X is between 6 and 10 inclusive)
p.x610 <- sum(p[which(x >= 6 & x <= 10)])
p.x610
## [1] 0.3470419
```

Question 2 - Probability distribution with an infinite number of possible values

Grocery shoppers on Sunday can be split into two main groups: daily (picking up a few things they need for the next day or two) and weekly (picking up all items they need for the coming week). Let x be the number of items people have in their shopping cart at checkout. The possible values of x are 1, 2, 3, … (no real upper limit, so we'll take that to be "infinity").

After studying historical data, you find that the following shape does a decent job at describing the relative frequencies of x.

$$Shape = xe^{-x/2} + 0.01x^2e^{-x/8} \qquad x = 0, 1, 2, 3, \ldots$$

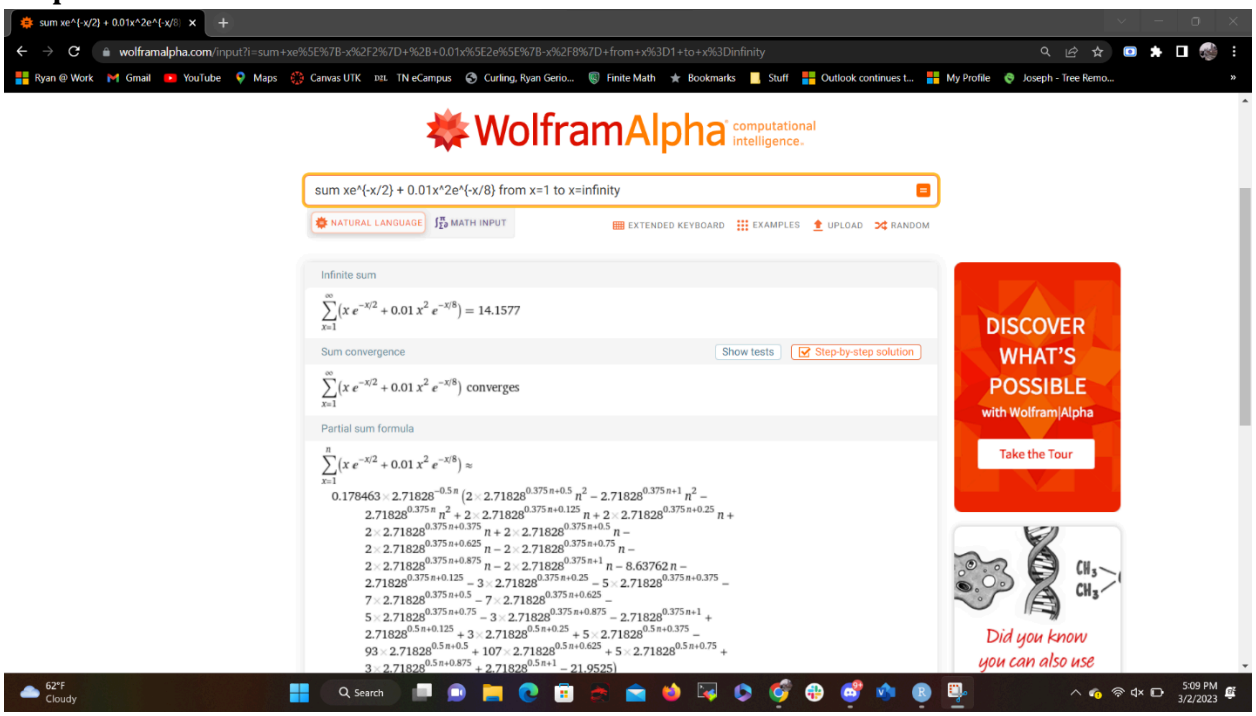
To convert the shape equation into the equation for the probabilities, it's not possible to do the standard shape conversion trick (`p <- shape/sum(shape)`) because there are an infinite number of possible values of x.

However, it *is* possible to convert the shape equation into the equation for the probabilities by multiplying the shape equation by the reciprocal of a "well-chosen" number!

$$P(X = x) = \frac{1}{???}\left(xe^{-x/2} + 0.01x^2e^{-x/8}\right) \qquad x = 1, 2, 3, \ldots$$

a) Use Wolfram-Alpha to find the value of ??? in the equation above. Provide a screenshot of your work in Wolfram (paste into Word after knitting). Sanity check: it's a little above 14.15.

Response:



b) Although the values of x technically go up to infinity, define x to be the integer sequence 1 to 80. Then, *using your equation* for $P(X = x)$ (after you've found the value of `???` in (a); no `p <- shape/sum(shape)` here!), define p to be the vector of corresponding probabilities. Remember that to take e and raise it to a power in R, you need to write it as `exp()`, with the power going inside the parentheses.

Show that `p[1]` = 0.04346438 and that `sum(p)` is "basically 1" (I was getting 0.998097), implying that x and p provide a "reasonable enough" PMF for the number of items in the cart since we've captured "most" of the probabilities (even though the possible values of x go to infinity).

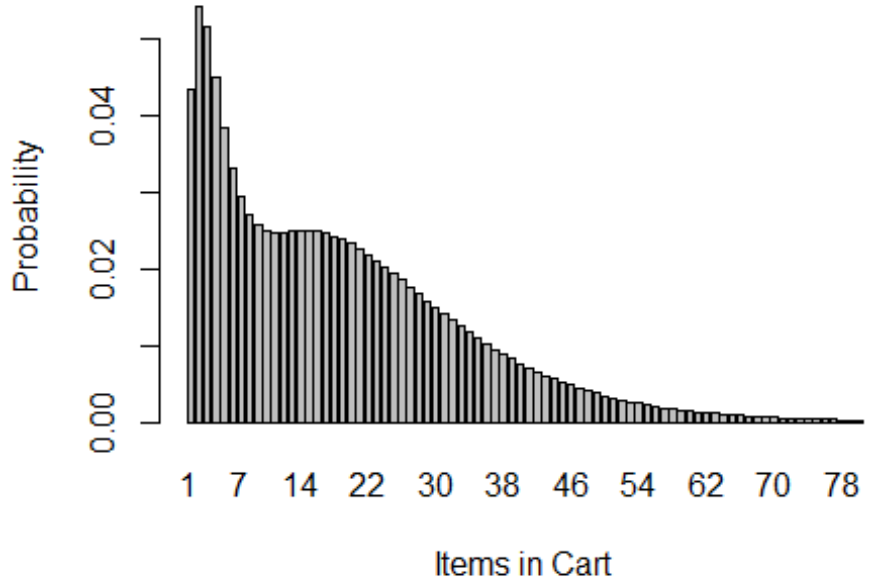
```
recip <- 14.1577
x <- 1:80
p <- (1/recip) * (x * exp(-x/2) + 0.01 * x^2 * exp(-x/8))

p[1]
## [1] 0.04346438

sum(p)
## [1] 0.998097
```

c) Provide a bar chart of the PMF. Label the x-axis "Items in Cart" and the y-axis "Probability". One hump represents the daily shoppers; the other represents the weekly shoppers.

```
barplot(p, names.arg = x, xlab = "Items in Cart", ylab = "Probability")
```



d) Use your PMF vectors x and p to find the probability that a shopper has between 4 and 20 items in their cart. Sanity check: a little under 47%.

```
sum(p[4:20])
## [1] 0.4688217
```

Question 3 - identifying when to use various notorious discrete probability distributions.

For each of the following random quantities, choose which probability model would be your "go to" in the absence of data. Respond with either the uniform, geometric, binomial, negative binomial, Poisson, or zero-Truncated Poisson, and *briefly* justify your answer. Do not choose negative binomial when the geometric distribution could be used (the geometric is a simpler, specific case of one flavor of the negative binomial).

a. The number of times you must flip 5 pennies (you toss them all at once) until all 5 come up tails.

Response: Geometric distribution. We are looking for how long it takes in terms of trials before our event of landing 5 tails occurs. With a fixed probability of our event occurring in each trial, we know there is a memorylessness property.

b. The number of tomatoes rejected by a hard-to-please shopper before 4 "keepers" are selected (each tomato has a probability `p` of being a keeper and a probability `1-p` of being rejected).

Response: Negative binomial distribution. Measuring number of failures before we reach 4 successes.

c. The number of cars traveling westbound on Kingston Pike that turn left onto Cherokee Blvd between 2-3pm on Tuesdays.

Response: Poisson distribution. Measuring "generic" counts.

d. The number of cellphones ever owned by current owners of an iPhone 14 Pro Max.

Response: Zero-truncated Poisson. The current owners of an Iphone 14 Pro Max must have owned at least one phone.

e. The number of potato chips in a bag of 100 that have that weird green tint.

Response: Binomial distribution. We have a known and fixed number of trials because each bag has exactly 100 chips.

f. The number of times that Comcast customers call customer support in a given month.

Response: Poisson distribution. We are measuring generic counts in the given month.

g. The number of BAS 471 students that have ordered something from the food-delivery robots you see around campus (there are 94 students currently enrolled in the course).

Response: Binomial distribution. We have a fixed number of independent trials in 94, and counting the number of students who orded from the robot as a "success".

h. The number of licks that a cat gives your hand when greeting you (the cat might not lick you at all).

Response: Poisson distribution. This is a generic count without context.

i. The number on the red Powerball (picked at random between the numbers 1 and 26).

Response: Uniform distribution. Each ball number 1-26 has an equal probability of being chosen.

Question 4: notorious discrete probability distributions in R

a. To maximize their time sleeping, many commuter students leave at the last possible moment for early-morning classes. Most of the time, traffic and light timings work out so that they arrive on time. Occasionally, they are late. Imagine that the

probability of making it on time is 78%, and that the outcomes are independent from day to day. The number of days someone is on time, out of the 36 days they travel to their early-morning class, can be modeled with a binomial distribution. Find the probability that:

- they are on time 32 or more days; about 7.8%
- they are on time every day ; about 0.013%
- they are on time between 25 and 30 days (inclusive); about 75.5%

```
n <- 36
p <- 0.78



p.32ormore <- 1 - pbinom(31,n,p)
p.32ormore
## [1] 0.07751563



p.everyday <- dbinom(n,n,p)
p.everyday
## [1] 0.0001304385



p.25and30 <- pbinom(30,n,p) - pbinom(24,n,p)
p.25and30
## [1] 0.7554183
```

b. Let's model the number of times that a student purchases food in the student union (over the course of a semester) with a Poisson distribution with an average of 3.2. Determine the value of `lambda` for this Poisson (hint: this is an easy one), then find:

- the probability someone makes 5 or more purchases (about 21.9%)
- the probability someone makes exactly 8 purchases (about 1.1%)
- the probability someone makes the average number of purchases

```
lambda <- 3.2



p.5ormore <- 1 - ppois(4,lambda)
p.5ormore
## [1] 0.2193875
```

```
p.8 <- dpois(8,lambda)
p.8
## [1] 0.0111157
```

```
p.avg <- dpois(lambda , lambda )
## Warning in dpois(lambda, lambda): non-integer x = 3.200000
p.avg
## [1] 0
```

c. You are asked to give out flyers about the Master's in Business Analytics program to people around Haslam. Imagine that each person, independently, has a 4.2% chance of accepting the flyer. After a while, you have only a single flyer to give away.

The geometric distribution provides a model for the number of people you must ask *before* someone takes the last flyer. For the following questions, figure out what it's asking about the number of "failures" (refusals of the flyer) before the first "success" (accepting the flyer), then find the requested probabilities.

- the probability that you will give away the last flyer to the 10th person you ask (about 2.9%).

- the probability that you'll give away your last flier to the 11th, 12th, 13th, 14th, or 15th person you ask (about 12.6%)
- the probability that you will have to ask 25 or more people to give away your last flyer (about 35.7%)

```
p <- 0.042
```

```
p.last10 <- dgeom(9 , p)
p.last10
## [1] 0.02854558
```

```
p.11to15 <- sum(dgeom(10:14, p))
p.11to15
## [1] 0.1257201
```

```
p.25ormore <- 1 - pgeom(23, p)
p.25ormore
## [1] 0.3570855
```

d. Right after you give away your last flyer, you're handed 6 more. Find the probability that:

- you have to ask 120 or more people to give them all away (about 61.6%)
- you have to ask at least 130 but no more than 160 people to give them all away (about 20.9%)

Remember that the negative binomial counts up the number of *failures* before a certain number of successes. A failure is someone refusing a flyer. A success is someone accepting a flyer. Translate these questions into how many failures are required to describe the event, then use the `nbinom` suite of functions.

```
p.120ormore <- 1 - sum(dnbinom(0:113, 6, p))
p.120ormore
## [1] 0.6163461
```

```
p.130to160 <- sum(dnbinom(130:162, 6, p))
p.130to160
## [1] 0.2095405
```

Question 5 (bonus)

One way to evaluate a pitcher is to evaluate his "no runs when no swings" probability. This refers to the probability that a pitcher throws strikes consistently enough so that, if no batter ever swings at the ball, the game would end without a run being scored. In other words, each inning has 3 strikeouts occur before 4 walks occur.

Imagine that each pitch independently has a 55% chance of being a "strike" and a 45% chance of being a "ball". A batter has a "strike out" if 3 strikes occur before 4 balls. A batter has a "walk" if 4 balls occur before 3 strikes. You've seen a similar problem back in an activity done in class for Unit 1, and on the Unit 2 Monte Carlo simulation worksheet, but this problem takes the analysis much further.

a) What's the name of the "notorious" distribution that counts up the number of failures that occur before a target number of successes?

Response: Negative Binomial distribution.

b) Let a ball be considered a "failure" and a strike a "success". A strikeout occurs when there are 3 strikes (successes) *before* 4 balls (failures). In other words, a strikeout occurs when the number of "failures" before the third success is at most 3. Use the appropriate `d` or `p` function in R to calculate the probability of a strikeout (remember in Unit 1 how tedious it was to enumerate all possible ways a strikeout could occur?). It turns out to be close to 74.5%.

```
```

```r
strike <- 0.55
ball <- 0.45

p_strikeout <- pnbinom(3, size = 3, prob = 0.55)
p_walk <- pnbinom(4, size = 4, prob = 0.45)

p_strikeout
## [1] 0.7447361
```

c) Imagine a similar game called "buskerball" where a batter has a "strike out" if 5 strikes occur before 7 balls. A batter has a "walk" if 7 balls occur before 5 strikes. If the probability of throwing a strike is 0.55, what's the probability of a strikeout? Sanity: close to 82.6%.

```r
strike <- 0.55
ball <- 0.45

prob_strikeout <- pnbinom(6, size = 5, prob = 0.55)

prob_strikeout
## [1] 0.8261996
```

d) Back to baseball. Although I haven't seen this as an official sports analytics interview question, I wouldn't be surprised if some team asks it. Imagine no batters ever swing at the ball. Thus, an "inning" will end with no runs scored as long as 3 strikeouts occur before 4 walks. Use one of the notorious distributions to find the probability that an inning has no runs scored (about 96%). Then, find the probability that the game ends without any runs being scored (9 innings in a row, each with no runs scored; about 69%).

```r
# we need to find the probability that 3 strikeouts occur before 4 walks

prob.noruns <- pnbinom(3, 4, p_strikeout)
prob.noruns
## [1] 0.924483

prob.noscoreingame <- prob.noruns^9
prob.noscoreingame
## [1] 0.4932766
```

e) Unfortunately, the pitcher who has the 55% chance of throwing a strike is injured and had to be replaced by a pitcher that only has a 45% chance of throwing a strike. No batters swing at his pitches.

e1) Find the PMF (x and p) of the number of points scored in an *inning*, i.e., - The probability an inning ends with 0 points scored (3 strikeouts occur before 4 walks) - The probability an inning ends with 1 point scored (the 3rd strikeout occurs after exactly 4 walks) - The probability an inning ends with 2 points scored (the 3rd strikeout occurs after

exactly 5 walks) - The probability an inning ends with 3 points scored (the 3rd strikeout occurs after exactly 6 walks) - etc. - Sanity check: the probability an inning ends with 3 points scored is 0.03613058 - Let `x` be `0:25` (so you go up to 25 points scored) and `p` be the corresponding probabilities. Print out the contents of your `p` vector once you've defined them, then make a barplot to show the PMF. Have the x-label be "Points Scored in Inning" and y-label be "Probability".

e2) Now, simulate the PMF of the total number of runs scored over the course of a *game* (9 innings) by sampling (with replacement) 9 values from the PMF for runs scored in an inning that you derived in part `e1`. - Run 100,000 trials, storing the total number of points scored in the game in a vector named `total.points` - Include a barplot of the results, i.e., `barplot(table(total.points))`. Have the x-label be "Points Scored in Game" and y-label be "Probability" - What is the probability that at most 10 points are scored during these 9 innings (about 91%)?