Lesson 05: Discrete Distributions

References

- Black, Chapter 5 Discrete Distributions (pp. 129-156)
- Davies, Chapter 16 Common Probability Distributions (pp. 332-342)
- Stowell, Chapter 7 Probability Distributions (pp. 87-97)
- Library Reserves: Teetor, Chapter 8 Probability (pp. 186-187)

Exercises:

1) Suppose a gambler goes to the race track to bet on four races. There are six horses in each race. He picks one at random out of each race and bets on each of the four selections. Assuming a binomial distribution, answer the following questions.

   i) The gambler wins all four races.

```
# R provides binomial probabilties directly
# dbinom(x, size, prob, log = FALSE)  # density function
# pbinom(q, size, prob, lower.tail = TRUE, log.p = FALSE)  # distribution function
# qbinom(p, size, prob, lower.tail = TRUE, log.p = FALSE)  # quantile function
# rbinom(n, size, prob)  # here n is the number of random variates to generate
# size is number of trials, often labeled as the binomial parameter n
# prob is the binomial parameter p, probability of success
# x is a binomial random variable in [0, n]
# we can compute binomial (n, p) probabilities to four significant digits by
# probability <- dbinom(x, size = n, prob = p)

prob_win_four <- dbinom(x = 4, size = 4, prob = 1/6)
sprintf("%.4f", prob_win_four)  # result printed with four places behind decimal
```

```
## [1] "0.0008"
```

   ii) The gambler loses all four races.

```
# This is the same as the probability that he/she wins no races: x = 0.

prob_loses_four <- dbinom(x = 0, size = 4, prob = 1/6)
sprintf("%.4f", prob_loses_four)
```

```
## [1] "0.4823"
```

   iii) The gambler wins exactly one race.

```
prob_wins_exactly_one <- dbinom(x = 1, size = 4, prob = 1/6)
sprintf("%.4f", prob_wins_exactly_one)
```

```
## [1] "0.3858"
```

   iv) The gambler wins at least one race.

```
# There are a couple of ways to get at this. One would be to take the union
# of the events that he/she wins 1, 2, 3, or 4 races, use dbinom(), and
# then use the addition rule to get the probability of the union.
# Another way, which we show here, is to take one minus the probability
# that he wins no races, a quantity we had computed in 1) a) ii).
```

```r
prob_wins_at_least_one <- 1 - dbinom(x = 0, size = 4, prob = 1/6)
sprintf("%.4f", prob_wins_at_least_one)
```

```
## [1] "0.5177"
```

2) A woman claims she can tell by taste if cream is added before a tea bag is placed in a tea cup containing hot water. An experiment is designed. A series of cups of tea will be prepared with n of them having the cream added prior to the tea bag and n of them with the cream added after the tea bag. This gives a total of $2n$ cups of tea. The sequence of tea cups is presented in random order. If the woman cannot discriminate it will be expected on average she would guess at random and be correct on half the tea cups. Answer the following questions assuming the number of successes follows a binomial distribution with probability equal to 0.5 and $2n$ total binomial trials.

   i) If the total number of binomial trials is $2n = 20$, what is the probability the woman is correct more than 15 out of 20 times?

```r
prob_15_of_20 <- pbinom(q = 15, size = 20, prob = 0.5, lower.tail = FALSE)
sprintf("%.4f", prob_15_of_20)
```

```
## [1] "0.0059"
```

   ii) To reduce the amount of labor, how small can the total number of binomial trials be while keeping the probability of $2n$ consecutive successes at or below 0.05? (We use $2n$ = the number of trials since half have the cream before and half after the tea bag.)

```r
# Here we can use a while-loop to find the desired value of n = 2m.
# We continue to have half of the cups with cream first (binomial p = 0.5),
# but now we are looking for a distribution function value in the upper tail
# at or below 0.05. This is the value we get from the pbinom() function.
target_p_value <- 0.05  # we will stop looking when we meet this target or go under
# later in the course we will call this a critical value in hypothesis testing
current_p_value <- 1.00  # initialze for search
n <- 0  # initialize number of cups... we will increase by 2 in each iteration
while (current_p_value > target_p_value) {
    n <- n + 2  # increase by one for this iteration
    # we are talking about consecutive hits greater than or equal to n - 1
    current_p_value <- pbinom(q = n-1, size = n, prob = 0.5, lower.tail = FALSE)
    # print out intermediate results for each iteration
    cat("\n Number of Consecutive Cups Correctly Identified:", n, "p_value: ",
    sprintf("%.4f", current_p_value))
    }
```

```
##
##  Number of Consecutive Cups Correctly Identified: 2 p_value:  0.2500
##  Number of Consecutive Cups Correctly Identified: 4 p_value:  0.0625
##  Number of Consecutive Cups Correctly Identified: 6 p_value:  0.0156
```

```r
# when we break out of the while-loop it means that the p_value target has been met
# at this point we know the value of n... the number of consecutive successes
cat("\n\nLady Tasting Tea Solution: ", n, "Consecutive Cups Correctly Labeled",
    "\n p-value: ",sprintf("%.4f", current_p_value),"<= 0.05 critical value")
```

```
##
##
## Lady Tasting Tea Solution:  6 Consecutive Cups Correctly Labeled
##  p-value:  0.0156 <= 0.05 critical value
```

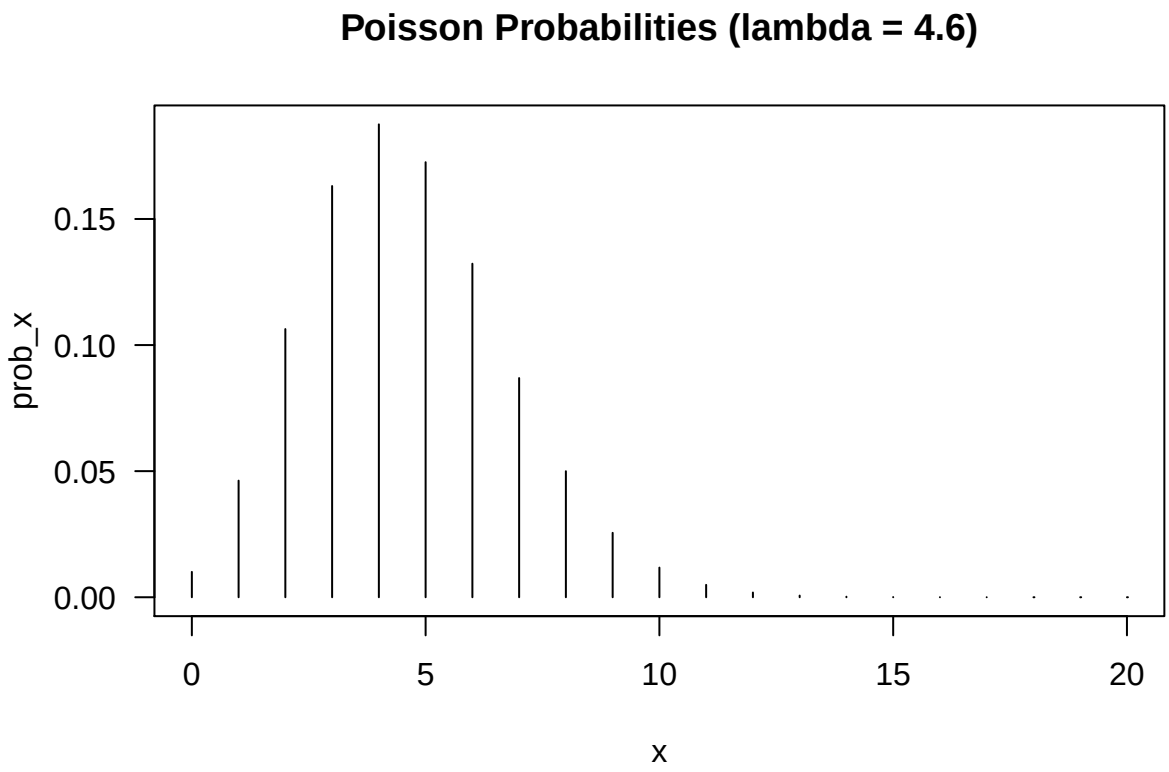
3) An emergency room has 4.6 serious accidents to handle on average each night. Using the Poisson

distribution, calculate the distribution of accidents per night. (In other words, what is the probability of 0, 1, 2, . accidents per night?) Plot the result.

```
# Again, we turn to built-in functions in R.
# dpois(x, lambda, log = FALSE)  # density function
# ppois(q, lambda, lower.tail = TRUE, log.p = FALSE)  # distribution function
# qpois(p, lambda, lower.tail = TRUE, log.p = FALSE)  # quantiles
# rpois(n, lambda)  # random variate generation
# To answer the question, we set lambda to 4.6 and look at the density function.
# Let's do it in a for-loop and print out intermediate results
for (x in 0:20)
    cat("\n x:", x, "prob:", sprintf("%.4f", dpois(x, lambda = 4.6)))
```

```
##
##  x: 0 prob: 0.0101
##  x: 1 prob: 0.0462
##  x: 2 prob: 0.1063
##  x: 3 prob: 0.1631
##  x: 4 prob: 0.1875
##  x: 5 prob: 0.1725
##  x: 6 prob: 0.1323
##  x: 7 prob: 0.0869
##  x: 8 prob: 0.0500
##  x: 9 prob: 0.0255
##  x: 10 prob: 0.0118
##  x: 11 prob: 0.0049
##  x: 12 prob: 0.0019
##  x: 13 prob: 0.0007
##  x: 14 prob: 0.0002
##  x: 15 prob: 0.0001
##  x: 16 prob: 0.0000
##  x: 17 prob: 0.0000
##  x: 18 prob: 0.0000
##  x: 19 prob: 0.0000
##  x: 20 prob: 0.0000
```

```
# let's plot this probability function
x <- 0:20
prob_x <- dpois(x, lambda = 4.6)
plot(x, prob_x, las = 1, type = "h")
title("Poisson Probabilities (lambda = 4.6)")
```



4) A production process occasionally produces at random a defective product at a rate of 0.001. If these products are packaged 100 at a time in a box and sold, answer the following questions and compare your answers. Plot the distributions for each type of variable over the range 0, 1, 2, 3, 4. What do you conclude?

i) Using the binomial distribution, calculate the probability a box has zero defectives.

```
# Here, we let n = 100 and p = 0.001, and compute the probability x = 0
sprintf("%.4f", dbinom(x = 0, size = 100, prob = 0.001))
```

```
## [1] "0.9048"
```

ii) Using the Poisson distribution, calculate the probability a box has zero defectives.

```
# Let lambda = n * p = 100 * 0.001 = .1
sprintf("%.4f", dpois(0, lambda = 0.1))
```

```
## [1] "0.9048"
```