Lab 7 – Hypergeometric, Geometric, and Poisson Distributions

This lab contains some numbered questions. You are required to submit:

1. One Word document (Lab7.pdf) that contains:
   - your written answers to the question and any charts produced
   - the commands you used to answer the question, when asked

2. One R script (Lab7.R) that contains all of your code.
   The script must run without errors.

Submit your files to the Lab 7 folder by 11:59pm <u>two</u> school days from now.

Packages required: none

Lab Objectives

In this lab, we will use hypergeometric, geometric, and Poisson probability distributions to model random variables arising in experiments related to traditional gambling games (cards, etc.) and some industrial contexts (manufacturing).

The functions we use will follow the same naming convention as binomial variables:

dbinom - density function for the distribution

pbinom - cumulative density function for the distribution

qbinom - inverse cumulative function (percentiles) for the distribution

rbinom - generate random values that follow the distribution

Hypergeometric Probabilities

Sampling *with replacement* can be modelled by a hypergeometric distribution. R provides a family of functions involving the hypergeometric distribution, similar in syntax and usage to the family of functions we saw for binomial distributions.

The help file for these functions says:

Usage

```
dhyper(x, m, n, k, log = FALSE)

phyper(q, m, n, k, lower.tail = TRUE, log.p = FALSE)

qhyper(p, m, n, k, lower.tail = TRUE, log.p = FALSE)

rhyper(nn, m, n, k)
```

Arguments

x, q vector of quantiles representing the number of white balls drawn without replacement from an urn which contains both black and white balls.

M the number of white balls in the urn.

N the number of black balls in the urn.

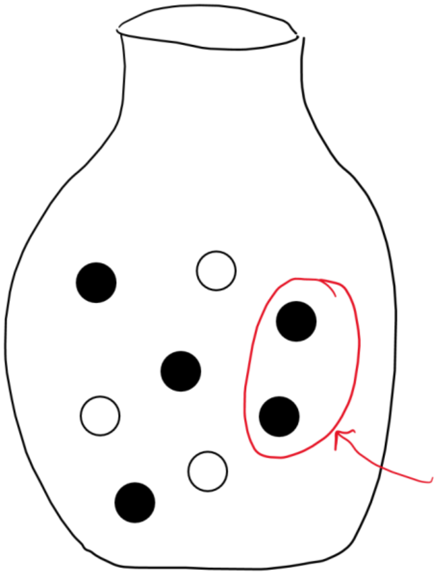
K the number of balls drawn from the urn.

P probability, it must be between 0 and 1.

Nn number of observations. If `length(nn) > 1`, the length is taken to be the number required.

log, log.p logical; if TRUE, probabilities p are given as log(p).

lower.tail logical; if TRUE (default), probabilities are *P[X ≤ x]*, otherwise, *P[X > x]*.



R syntax:

$m = 3$ (White, “success”)
$n = 5$ (Black, “failure”)
$k = 2$ (sample size)
$x = 0$ (successes in sample)

sample

The functions **dhyper** and **phyper** are analogous to **dbinom** and **pbinom**. For instance, we can find the probability of getting exactly two jacks when drawing ten cards from a standard 52-card deck:

```
> dhyper(2, 4, 48, 10)    #4 Jacks ("success") and 48 non-Jacks

[1] 0.1431157
```

We can also find the probability of drawing two or fewer jacks when drawing ten cards without replacement from a standard 52-card deck:

```
> phyper(2, 4, 48, 10)

[1] 0.9806076
```

1. Suppose we are drawing $n = 8$ cards from a standard 52-card deck without replacement. Using **dhyper**, create a table and a bar plot that gives that exact probability distribution for $X =$ the number of Aces obtained. Give clear and descriptive labels for your barplot.

2. Here you will approximate the probability distribution for $X =$ the number of aces obtained when drawing 8 cards from a standard 52-card deck using simulation.
   a. Write a function that simulates drawing **n.cards** cards **m.trials** times, using the **sample** function. Your function should output a table giving the relative frequencies, as well as a bar plot. Run your function for **n.card**=8 and **m.trials**= 10^6 and record the table and bar plot.

   b. Write a function that simulates drawing **n.cards** cards **m.trials** times, using the **rhyper** function. As before, your function should output a table giving the relative frequencies, as well as a bar plot. Run your function for **n.cards**=8 and **m.trials** =10^6 and provide a table and a bar plot.

   [The tables and bar plot from parts (a) and (b) should be very similar to the one obtained using **dhyper**.]

3. During one stage in the manufacture of integrated circuits, a coating must be applied. Suppose that in a batch of 999 chips, 333 did not receive a thick enough coating. Now 300 of the 999 chips are randomly selected for testing. Give the commands, along with your output, to compute the following probabilities (answers are in brackets):
    a. Exactly 100 do not receive a thick enough coating [0.05835]
    b. 100 or fewer do not receive a thick enough coating [0.5305]
    c. Fewer than 100 do not receive a thick enough coating [0.4721]
    d. At least 110 do not receive a thick enough coating [0.08254]
    e. Between 90 and 110 (inclusive) do not receive a thick enough coating [0.8759]

Geometric Probabilities

The geometric distribution gives the probability that the *first* success in a series of Bernoulli trials will occur on trial X. It is assumed that each trial has a probability p of success and probability q of failure. Therefore

$$P(X = x) = p^{x-1}q$$

The function **dgeom** calculates this probability for you. However, note that R defines things a bit differently, so

```
dgeom(x, p)
```

is the probability of getting x failures before the first success

$$\texttt{dgeom(x, p)} = p^x q$$

The Geometric Distribution Usage

```
dgeom(x, prob, log = FALSE)

pgeom(q, prob, lower.tail = TRUE, log.p = FALSE)

qgeom(p, prob, lower.tail = TRUE, log.p = FALSE)

rgeom(n, prob)
```

Arguments

x, q vector of quantiles representing the number of failures in a sequence of Bernoulli trials before success occurs.

p vector of probabilities.

n number of observations. If `length(n) > 1`, the length is taken to be the number required.

prob probability of success in each trial. `0 < prob <= 1`.

log, log.p logical; if TRUE, probabilities p are given as log(p).

lower.tail logical; if TRUE (default), probabilities are *P[X ≤ x]*, otherwise, *P[X > x]*.

One difference between the geometric distribution and the binomial and hypergeometric distributions is that if X is a geometric variable then there is no upper bound on X (while $X \leq n$ for a binomial or hypergeometric variable). For example, if you roll a die 10 times, the maximum number of 3's you can get is 10. If you draw 8 cards without replacement, you will get at most 4 aces. But if you purchase a lottery ticket every week until you get a winning one – an experiment modelled by the geometric distribution – you may theoretically be purchasing lottery tickets forever!

For practical purposes, we will choose a "cutoff" point (or artificial maximum value) for X so that we can produce a table and/or histogram of probabilities.

4. A student decides to purchase lottery tickets until she wins a prize. Suppose the probability of winning a prize is 1/3. Let $X =$ the number of tickets the student has bought when they first win a prize. Use **dgeom** to generate a table and a barplot showing the probability distribution for X. Exclude all probabilities less than 0.00005.

5. Approximate the probability distribution for $X =$ the number of lottery tickets the student has bought when they first win a prize. Do it in two ways:
    a. By writing a function that uses the function **sample** to simulate **m.reps** repetitions of the experiment in which a student buys a lottery ticket every day until she wins. For each repetition, record the value $X =$ the number of tickets (including the final winning ticket) until the first win. Record a table showing the relative frequencies of X and a bar plot showing the distribution of X. Run your function for **m.reps** = 10^6 and record the table and bar plot (with appropriate axis labels and title).

    b. By writing a function that simulates **m.trials** repetitions of the same experiment, but use **rgeom** to generate the simulated values of X. As before, your function should output a table giving the relative frequencies, as well as a bar plot. Run your function for **m.trials** $= 10^6$ and provide a table and a bar plot. NOTE: Ensure the first value of X is 1 (not 0).

       The results of both of your simulations should be very similar to the results obtained with the **dgeom** function.

The function **pgeom** computes cumulative probabilities. For example, we can compute the probability of having to purchase 7 or fewer tickets in order to get a winner:

```
> pgeom(7-1, 1/3)

[1] 0.9414723
```

6. In an industrial "torture test", a light switch is turned on then off repeatedly until it fails. Each on-off attempt has a probability 0.001 of failure. Record the R command used to calculate each of the probabilities:
    a. That the switch will fail by the time it has been turned on and off 500 times [0.3936]
    b. That the switch will not fail until it has been turned on and off at least 1200 times [0.3013]
    c. That the switch will fail when it has been turned on and off between 1000 and 2000 times [0.2329]

Poisson Probabilities

A Poisson distribution can be used to model a variable $X =$ the number of times a certain event occurs during a fixed time interval, given the average number of occurrences, λ.

From the help file:

The Poisson Distribution - Usage

```
dpois(x, lambda, log = FALSE)

ppois(q, lambda, lower.tail = TRUE, log.p = FALSE)

qpois(p, lambda, lower.tail = TRUE, log.p = FALSE)

rpois(n, lambda)
```

Arguments

x vector of (non-negative integer) quantiles.

q vector of quantiles.

p vector of probabilities.

n number of random values to return.

lambda vector of (non-negative) means.

log, log.p logical; if TRUE, probabilities p are given as log(p).

lower.tail logical; if TRUE (default), probabilities are $P[X \le x]$, otherwise, $P[X > x]$.

Like the geometric distribution, there is no theoretical upper bound on the number of nonzero probabilities. So like before, we will restrict our tables to those where probabilities are at least 0.00005.

Unlike the other experiments we have modelled so far, we cannot simulate these experiments using the function **sample.** Instead, we will compare the exact probabilities to ones that are generated by **rpois**.

7. A factory produces fibre optic cables that have an average of 0.75 flaws per meter. Use **dpois** to give an exact probability distribution for the number of flaws in one metre of cable as both a table and a bar plot.

8. Use **rpois** to write a function that simulates selecting **m.reps** meter-long fibre optic cables and counting the number of flaws on each. For each cable, $X =$ the number of flaws. As before, your function should output a table giving the relative frequencies of X, as well as a bar plot. Run your function for **m.reps** = 10^6 and provide a table and a bar plot. Your results should be very similar to the ones you got in Question 7.

The function **ppois** function computes cumulative probabilities.

9. A city records an average of 34 transformer failures per year. Give the commands, along with your output, to compute the following probabilities (answers are in brackets):
   a. Exactly 34 transformers fail in a year [0.06825]
   b. 30 or fewer transformers fail in a year [0.2804]
   c. Fewer than 30 transformers fail in a year [0.2235]
   d. More than 38 transformers fail in a year [0.2166]
   e. At least 38 transformers fail in a year [0.2681]
   f. Between 30 and 40 transformers (inclusive) fail in a year [0.6429]
   g. 34 or fewer transformers fail in each of two consecutive years [0.2975]
   h. 68 or fewer transformers fail over a two-year period [0.5322]

10. It is possible to simulate a Poisson variable using **sample**. Suppose X is a Poisson random variable with $\lambda = 5$. Then $\mu_X = \lambda = 5$ and $\sigma_X = \sqrt{\lambda} = 2.236$.

    a. Calculate $P(X = 4)$ using the formula for Poisson (not R).

    A Poisson variable is like a binomial variable with an *infinite* number of trials. Let Y be a binomial random variable with $n = 10^6$ and $p = \frac{5}{10^6} = 0.000\ 005$. Then

    $$\mu_Y = np = 5 \text{ and}$$

    $$\sigma_Y = \sqrt{npq} = \sqrt{10^6 \cdot 0.000005 \cdot 0.999995} = 2.236$$

    b. Calculate $P(X = 4)$ using the Binomial formula (not R).