

Solution 2: Statistical inference (I)

Benjamin Smith

June 22, 2026

Part 1: Probability distributions

Question a: A contestant on a game show needs to answer 10 questions correctly to win the jackpot. However, if they get 4 incorrect answers, they are kicked off the show. Suppose one contestant consistently has a 80% chance of correctly responding to any question.

1. What is the probability distribution?
2. What is the probability that she will correctly answer 10 questions before 4 incorrect responses?
3. Write out the R code to calculate (b).

Solution

1. The experiment is modeled by the **Negative Binomial Distribution**, as it counts the number of failures (k) before a specified number of successes (r) occurs.
 - Success: Correct answer, $p = 0.8$
 - Failure: Incorrect answer, $q = 1 - p = 0.2$
 - Required successes: $r = 10$
 - Losing condition: $k = 4$ failures
2. The contestant wins if she achieves $r = 10$ successes before getting 4 failures. This means the total number of failures k must be in the set $\{0, 1, 2, 3\}$. The probability mass function is $P(X = k) = \binom{k+r-1}{r-1} p^r q^k$. The total probability of winning is the sum over the allowed values of k :

$$P(\text{Win}) = \sum_{k=0}^3 \binom{k+10-1}{10-1} (0.8)^{10} (0.2)^k$$

$$P(\text{Win}) = \binom{9}{9} (0.8)^{10} (0.2)^0 + \binom{10}{9} (0.8)^{10} (0.2)^1 + \binom{11}{9} (0.8)^{10} (0.2)^2 + \binom{12}{9} (0.8)^{10} (0.2)^3$$

$$P(\text{Win}) \approx 0.1074 + 0.2147 + 0.2362 + 0.1890 \approx 0.7473$$

3. The calculation can be performed concisely using R's cumulative negative binomial function, `pnbinom`.

```
# Calculates the probability of 0, 1, 2, or 3 failures (q)
# before 10 successes (size) are achieved.
pnbinom(q = 3, size = 10, prob = 0.8)

# Output: [1] 0.747323
```

Question b: A small town's police department issues 5 speeding tickets per month on average.

1. Using a Poisson random variable, what is the likelihood that the police department issues 3 or fewer tickets in one month?
2. What is the probability that 10 days or fewer elapse between two tickets being issued?
3. Write out the R code to calculate (a), (b).

Solution

1. First, we note that here $P(Y \leq 3) = P(Y = 0) + P(Y = 1) + \dots + P(Y = 3)$. Applying the probability mass function for a Poisson distribution with $\lambda = 5$, we find that

$$\begin{aligned} P(Y \leq 3) &= P(Y = 0) + P(Y = 1) + P(Y = 2) + P(Y = 3) \\ &= \frac{e^{-5}5^0}{0!} + \frac{e^{-5}5^1}{1!} + \frac{e^{-5}5^2}{2!} + \frac{e^{-5}5^3}{3!} \\ &= 0.27. \end{aligned}$$

```
# q is the number of events (3 or fewer).  
# lambda is the average rate of events per interval.  
ppois(q = 3, lambda = 5)
```

```
## [1] 0.2650259
```

2. We know the town's police issue 5 tickets per month. For simplicity's sake, assume each month has 30 days. Then, the town issues $\frac{1}{6}$ tickets per day. That is $\lambda = \frac{1}{6}$, and the average wait time between tickets is $\frac{1}{1/6} = 6$ days. Therefore,

$$P(Y < 10) = \int_0^{10} \frac{1}{6} e^{-\frac{1}{6}y} dy = 0.81$$

```
pexp(10, rate = 1/6)
```

```
## [1] 0.8111244
```

Part 2: Statistical inference

1. (AoS 6.6.2) Let $X_1, \dots, X_n \sim \text{Uniform}(0, \theta)$ and let $\hat{\theta} = \max\{X_1, \dots, X_n\}$. Find the bias, se and MSE of this estimator.
2. (AoS 6.6.3) Let $X_1, \dots, X_n \sim \text{Uniform}(0, \theta)$ and let $\hat{\theta} = 2\bar{X}_n$. Find the bias, se and MSE of this estimator.
3. Let $X_1, \dots, X_n \sim \text{Uniform}(0, 1)$. Let $Y_n = \bar{X}_n^2$. Find the limiting distribution of Y_n . (Hint: CLT)

Solution

1. The CDF G of $\hat{\theta}$ is

$$\begin{aligned}
G(\hat{\theta}) &= \mathbb{P}(\hat{\Theta} \leq \hat{\theta}) \\
&= \mathbb{P}\left(\max\{X_1, \dots, X_n\} \leq \hat{\theta}\right) \\
&= \prod_{i=1}^n \mathbb{P}(X_i \leq \hat{\theta}) \\
&= F_{\theta}(\hat{\theta})^n \\
&= \left(\frac{\hat{\theta}}{\theta}\right)^n
\end{aligned}$$

The density is therefore

$$g(\hat{\theta}) = \binom{n}{\hat{\theta}} \left(\frac{\hat{\theta}}{\theta}\right)^{n-1}$$

Thus,

$$\mathbb{E}_{\theta}(\hat{\theta}) = \int_0^{\theta} \hat{\theta} g(\hat{\theta}) d\hat{\theta} = \frac{n\theta}{n+1}$$

and

$$\text{bias} = \frac{n\theta}{n+1} - \theta = -\frac{\theta}{n+1}.$$

Also,

$$\mathbb{E}_{\theta}(\hat{\theta}^2) = \int_0^{\theta} \hat{\theta}^2 g(\hat{\theta}) d\hat{\theta} = \frac{n\theta^2}{n+2}$$

and so

$$\mathbb{V}_{\theta}(\hat{\theta}) = \frac{n\theta^2}{n+2} - \left(\frac{n\theta}{n+1}\right)^2 = \frac{n\theta^2}{(n+2)(n+1)^2}$$

The mse is

$$\text{bias}^2 + \mathbb{V} = \left(\frac{\theta}{n+1}\right)^2 + \frac{n\theta^2}{(n+2)(n+1)^2} = \frac{2\theta^2}{(n+1)(n+2)}$$

2. Recall that $\mathbb{E}(X_i) = \theta/2$, $\mathbb{V}(X_i) = \theta^2/12$. So

$$\mathbb{E}_{\theta}(2\bar{X}) = 2\mathbb{E}_{\theta}(\bar{X}) = 2\frac{\theta}{2} = \theta$$

and hence bias = 0. Now

$$\mathbb{V}_{\theta}(2\bar{X}) = 4\mathbb{V}_{\theta}(\bar{X}) = \frac{4\sigma^2}{n} = \frac{4\theta^2}{12n} = \frac{\theta^2}{3n}.$$

Since this estimator is unbiased,

$$\text{mse} = \mathbb{V}_{\theta}(\hat{\theta}) = \frac{\theta^2}{3n}.$$

3. $\mu = \mathbb{E}(X_i) = 1/2$ and $\sigma^2 = \mathbb{V}(X_i) = 1/12$. By the CLT,

$$\frac{\sqrt{n}(\bar{X} - \mu)}{\sigma} = \sqrt{12n} \left(\bar{X} - \frac{1}{2}\right) \rightsquigarrow N(0, 1).$$

Now $Y = g(\bar{X})$ where $g(s) = s^2$. And $g'(s) = 2s$ and $g'(\mu) = g'(1/2) = 2(1/2) = 1$. From the delta method,

$$\frac{\sqrt{n}(Y - g(\mu))}{|g'(\mu)|\sigma} = \sqrt{12n} \left(\bar{X} - \frac{1}{4}\right) \rightsquigarrow N(0, 1)$$

Part 3: Newton-Raphson implementation

The ABO-gene or ABO-locus is on chromosome 9. It has 3 alleles (antigens) (A, B, O) and it determines 4 blood type (A, B, AB, O).

We have a large random sample obtained from Berlin (Bernstein 1925, Sham's book page 44):

- $n_A = 9123$ blood type A
- $n_B = 2987$ blood type B
- $n_{AB} = 1269$ blood type AB
- $n_O = 7725$ blood type O

For instance, $n_A = 9123 = n_{AA} + n_{AO}$: Among 9123 blood type A individuals, some have genotype AA and the others have genotype AO .

Our interest is to estimate the allele frequencies of alleles A , B , and O . i.e. $p = \text{freq}(\text{allele } A)$, $q = \text{freq}(\text{allele } B)$, $1 - p - q = \text{freq}(\text{allele } O)$.

1. Write out the log-likelihood $L(p, q)$.
2. Is there a closed-form solution of this log-likelihood function?
3. Formulate the problem as a missing data problem and use the Newton-Raphson algorithm to find the MLEs, $\hat{p}$ and $\hat{q}$, that maximize the log-likelihood, $\ln L(p, q)$.

Solution

1. Let $X = (n_A, n_B, n_{AB}, n_O)$, X follows the multinomial distribution,

$$L(p, q) = \binom{n}{n_A, n_B, n_{AB}, n_O} (p^2 + 2p(1-p-q))^{n_A} (q^2 + 2q(1-p-q))^{n_B} (2pq)^{n_{AB}} ((1-p-q)^2)^{n_O}$$

The general approach to estimate allele frequency is maximum likelihood estimation. To find MLE, take the derivatives of the log-likelihood function, and find the pair of (p, q) that set the derivatives to 0.

The log transformation is performed since finding a maximizer of $L(p, q)$ is equivalent to finding a maximizer of $\ln L(p, q)$. The log-likelihood is,

$$\ln L(p, q) \sim n_A \ln(p^2 + 2p(1-p-q)) + n_B \ln(q^2 + 2q(1-p-q)) + n_{AB} \ln(2pq) + n_O \ln((1-p-q)^2) \quad (1)$$

Take the partial derivatives of the log-likelihood and set them to 0,

$$\frac{\partial \ln L(p, q)}{\partial p} = \frac{2(1-p-q)}{p(2-p-2q)} n_A + \frac{2}{2p+q-2} n_B + \frac{1}{p} n_{AB} - \frac{2}{1-p-q} n_O = 0 \quad (2)$$

$$\frac{\partial \ln L(p, q)}{\partial q} = \frac{2}{p+2q-2} n_A + \frac{2(1-p-q)}{q(2-2p-q)} n_B + \frac{1}{q} n_{AB} - \frac{2}{1-p-q} n_O = 0 \quad (3)$$

2. It is hard to find the explicit form of the (p, q) from equation (2) and (3) **directly**. We consider solve the problem **iteratively**.
3. Newton-Raphson algorithm is another method that can be applied to estimate the ABO allele frequency. The algorithm is initially designed to find the roots (or zeros) of a real-valued function iteratively.

In the ABO blood type settings, maximizing the likelihood is equivalent to finding the roots of the derivative of log-likelihood function. Since it is hard to find the roots directly, one can approximate it iteratively. Here denote the parameters to be estimated as $\vec{\theta} = (p, q)$, denote the log-likelihood shown in equation (1) as $f(\vec{\theta}) = \ln L(\theta)$, then the partial derivatives of the log-likelihood (score function) is $f'(\vec{\theta}) = f'(p, q)$, the second derivatives (Hessian matrix, or observed information $-I(\theta)$) as $f''(\vec{\theta}) = f''(p, q)$. The explicit forms of the score function and observed information are shown in the appendix.

4. Expectation-Maximum (EM) algorithm is a method for obtaining the Maximum likelihood estimates (MLE) of parameters iteratively. It usually contains two parts: **E-step (expectation)** computes an expected value of the log-likelihood using current estimate for the parameters; **M-step (Maximization)** calculates MLE based on the log likelihood in the E-step, then updates the estimates of the parameters.

The EM algorithms are often used when the model contains unobserved data. In the ABO settings, one could observe the phenotype counts (n_A, n_B, n_{AB}, n_O) , while the genotype counts n_{AA} or n_{AO} in blood A group and n_{BB} or n_{BO} in blood B group are missing. This leads to a problem when applying direct counting (Sham, 1998) to estimate p, q , the allele frequency of A and B respectively. The following steps will show how this missing data problem could be solved by the EM algorithm.

In each iteration k , firstly, the **E-step** computes the expected value of the log-likelihood $h(p, q)$ using the observed data $n_{\text{obs}} = (n_A, n_B, n_{AB}, n_O)$ and current parameter value $p^{(k)}, q^{(k)}$:

$$Q_k(p, q) = E[h_k(p, q) | n_{\text{obs}}, p^{(k)}, q^{(k)}] \quad (4)$$

and $h_k(p, q) \sim 2n_{AA} \log p^{(k)} + n_{AO} \log(2p^{(k)}(1-p^{(k)}-q^{(k)})) + 2n_{BB} \log q^{(k)} + n_{BO} \log(2p^{(k)}(1-p^{(k)}-q^{(k)})) + n_{AB} \log(2p^{(k)}q^{(k)}) + 2n_O \log(1-p^{(k)}-q^{(k)})$

Since the log-likelihood is linear w.r.t. the missing data, therefore, when taking the expectation for each component, the missing data can be imputed in this case. For example, since $E[2n_{AA} \log p^{(k)} | n_{\text{obs}}, p^{(k)}, q^{(k)}] = 2 \log p^{(k)} E[n_{AA} | n_{\text{obs}}, p^{(k)}, q^{(k)}]$, and under the HWE assumption,

$$E[n_{AA} | n_{\text{obs}}, p^{(k)}, q^{(k)}] = \frac{\text{freq}(AA)}{\text{freq}(AA) + \text{freq}(AO)} n_A = \frac{p^{(k)} p^{(k)}}{p^{(k)} p^{(k)} + 2p^{(k)} (1 - p^{(k)} - q^{(k)})} n_A := n_{AA}^{(k)}$$

By the similar calculation, we can get the imputed data $n_{AA}^{(k)}, n_{AO}^{(k)}, n_{BB}^{(k)}, n_{BO}^{(k)}$ for each iteration.

Then the **M-step** computes the MLE based on the likelihood in equation (4). To be specific,

$$\begin{aligned} \frac{\partial Q_k(p, q)}{\partial p} &= \frac{2n_{AA}^{(k)} + n_{AO}^{(k)} + n_{AB}}{p} - \frac{n_{AO}^{(k)} + n_{BO}^{(k)} + n_O}{1 - p - q} = 0 \\ \frac{\partial Q_k(p, q)}{\partial q} &= \frac{2n_{BB}^{(k)} + n_{BO}^{(k)} + n_{AB}}{q} - \frac{n_{AO}^{(k)} + n_{BO}^{(k)} + n_O}{1 - p - q} = 0 \end{aligned}$$

Thus, the updated values of parameters are

$$p^{(k+1)} = \frac{2n_{AA}^{(k)} + n_{AO}^{(k)} + n_{AB}}{2n} \quad q^{(k+1)} = \frac{2n_{BB}^{(k)} + n_{BO}^{(k)} + n_{AB}}{2n}$$

```
df <- function(p, q) {
  dfp <- 2*(1 - p - q)*nA / (p*(2 - p - 2*q)) + 2*nB / (2*p + q - 2) + nAB/p - 2*n0 / (1 - p - q)
  dfq <- 2*nA / (p + 2*q - 2) + 2*(1 - p - q)*nB / (q*(2 - 2*p - q)) + nAB/q - 2*n0 / (1 - p - q)
  c(dfp, dfq)
}

d2f <- function(p, q) {
  pp <- -4*(1 - p - q)^2*nA / (p^2*(2 - p - 2*q)^2) - 2*nA / (p*(2 - p - 2*q)) -
    4*nB / ((2 - 2*p - q)^2) - nAB / (p^2) - 2*n0 / ((1 - p - q)^2)
  pq <- 4*(1 - p - q)*nA / (p*(2 - p - 2*q)^2) - 2*nA / (p*(2 - p - 2*q)) -
    2*nB / ((2 - 2*p - q)^2) - 2*n0 / ((1 - p - q)^2)
  qp <- -2*nA / ((2 - p - 2*q)^2) + 4*(1 - p - q)*nB / (q*(2 - 2*p - q)^2) -
    2*nB / (q*(2 - 2*p - q)) - 2*n0 / ((1 - p - q)^2)
  qq <- -4*nA / ((2 - p - 2*q)^2) - 4*(1 - p - q)^2*nB / ((q^2*(2 - 2*p - q)^2)) -
    2*nB / (q*(2 - 2*p - q)) - nAB / (q^2) - 2*n0 / ((1 - p - q)^2)
  matrix(c(pp, pq, qp, qq), nrow = 2, ncol = 2, byrow = T)
}

max <- 10000
epsilon <- 1e-5
iter <- 0
p <- 0.3333333
q <- 0.3333333
diff1 <- 1
diff2 <- 1
nA <- 9123
nB <- 2987
nAB <- 1269
n0 <- 7725

theta <- c(p, q)
rp <- NULL
rq <- NULL
```

```

rlike <- NULL
rp[1] <- p
rq[1] <- q
rlike[1] <- log_like(nA, nB, nAB, n0, p, q)

while (diff1 > epsilon & diff2 > epsilon & iter < max) {
  p <- theta[1]
  q <- theta[2]

  theta.new <- theta - solve(d2f(p,q)) %*% df(p, q)

  diff1 <- abs(theta[1] - theta.new[1])
  diff2 <- abs(theta[2] - theta.new[2])

  theta <- theta.new

  log_lik <- log_like(nA, nB, nAB, n0, theta[1], theta[2])

  iter <- iter + 1

  rp[iter+1] <- theta[1]
  rq[iter+1] <- theta[2]
  rlike[iter+1] <- log_lik
}

knitr::kable(
  data.frame(c(0:(length(rp)-1)), rp, rq, rlike),
  col.names = c("iteration $k$", "$p^{\{k\}}$", "$q^{\{k\}}$", "log likelihood"), booktabs = TRUE,
  align = "cccr",
  caption = 'Results for Newton-Raphson algorithm'
)

```

Table 1: Results for Newton-Raphson algorithm

iteration k	$p^{(k)}$	$q^{(k)}$	log likelihood
0	0.3333333	0.3333333	-32186.43
1	0.4175144	0.0171487	-29913.64
2	0.2981320	0.0318471	-27106.11
3	0.3034715	0.0546993	-25646.39
4	0.2950282	0.0820775	-24968.81
5	0.2892635	0.1013210	-24828.66
6	0.2877549	0.1063252	-24822.77
7	0.2876857	0.1065546	-24822.76
8	0.2876856	0.1065550	-24822.76