**Section 3.2 – I×J contingency tables and inference procedures**

We now examine the extension of the 2×2 contingency table to an I×J contingency table. We begin by focusing on two separate ways that one can think of how the counts arise in a contingency table structure through using a multinomial probability distribution. Future chapters will examine contingency tables again by examining them from Poisson and hypergeometric probability distribution prospectives.

One multinomial distribution

Set-up:

* X denotes the row variable with levels i = 1, …, I
* Y denotes the column variable with levels j = 1, …, J
* P(X = i, Y = j) = πij
* 
* nij denotes the cell count for row i and column j
* 

Contingency tables summarizing this information are shown below:

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Y |  |
|  |  | 1 | 2  |   | J |  |
| X | 1 | 11 | 12 |  | 1J | 1+ |
| 2  | 21 | 22 |  | 2J | 2+ |
|   |  |  |  |  |  |
| I | I1 | I2 |  | IJ | I+ |
|  |  | +1 | +2 |  | +J | 1 |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Y |  |
|  |  | 1 | 2  |   | J |  |
| X | 1 | n11 | n12 |  | n1J | n1+ |
| 2  | n21 | n22 |  | n2J | n2+ |
|   |  |  |  |  |  |
| I | nI1 | nI2 |  | nIJ | nI+ |
|  |  | n+1 | n+2 |  | n+J | n |

The set-up given for these contingency tables fits right into the multinomial setting of the previous section. We now just categorize the responses with respect to X and Y. The probability mass function for observing particular values of n11, …, nIJ is



The MLE of πij is the estimated proportion  = nij/n.

We can also discuss marginal distributions for X and for Y as well:

* X has a multinomial distribution with counts ni+ for i = 1, …, I and corresponding probabilities πi+. The maximum likelihood estimate of πi+ is  = ni+/n.
* Y has a multinomial distribution with counts n+j for j = 1, …, J and corresponding probabilities π+j. The MLE of π+j is  = n+j/n

Example: Multinomial simulated sample (Multinomial.R)

As a quick way to see what a sample looks like in a 2×3 contingency table setting, consider the situation with n = 1,000 observations, π11 = 0.2, π21 = 0.3, π12 = 0.2, π22 = 0.1, π13 = 0.1, and π23 = 0.1. Below is how we can simulate a sample.

> pi.ij <- c(0.2, 0.3, 0.2, 0.1, 0.1, 0.1)

> pi.table <- array(data = pi.ij, dim = c(2,3), dimnames = list(X = 1:2, Y = 1:3))

> pi.table

 Y

X 1 2 3

 1 0.2 0.2 0.1

 2 0.3 0.1 0.1

> set.seed(9812)

> save <- rmultinom(n = 1, size = 1000, prob = pi.ij)

> c.table1 <- array(data = save, dim = c(2,3), dimnames = list(X = 1:2, Y = 1:3))

> c.table1

 Y

X 1 2 3

1 191 206 94

2 311 95 103

> c.table1/sum(c.table1)

 Y

X 1 2 3

 1 0.191 0.206 0.094

 2 0.311 0.095 0.103

I multinomial distributions

Instead of using one multinomial distribution, one can think of the data arising through separate multinomial distributions for each row. Thus, there are I multinomial distributions. This can be thought of as a direct extension to what we had in Section 1.2 with two binomial distributions (one for each row). Taking a sample in this type of format is often referred to as independent multinomial sampling.

Set-up:

* ni+ as fixed row counts
* P(Y = j | X = i) = πj|i represents the conditional probability of observing response category j given an item is in group i
* ni1, …, niJ are the counts with corresponding probabilities π1|i, …, πJ|i.
*  for i = 1, …, I

We can view the contingency table in terms of these conditional probabilities:

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Y |  |
|  |  | 1 | 2  |   | J |  |
| X | 1 | 1|1 | 2|1 |  | J|1 | 1 |
| 2  | 1|2 | 2|2 |  | J|2 | 1 |
|   |  |  |  |  |  |
| I | 1|I | 2|I |  | J|I | 1 |

The probability mass function for each row is



The likelihood function is the product of the I multinomial distributions:



The MLE of πj|i is . Notice how these estimates can be found from the previous MLEs in the one multinomial setting: .

Example: Multinomial simulated sample (Multinomial.R)

Consider again a 2×3 contingency table setting. Suppose π1|1 = 0.4, π2|1 = 0.4, π3|1 = 0.2, π1|2 = 0.6, π2|2 = 0.2, and π3|2 = 0.2. These conditional probabilities result from using πij in the previous example:

> pi.cond <- pi.table/rowSums(pi.table)

> pi.cond # pi\_j|i

 Y

X 1 2 3

 1 0.4 0.4 0.2

 2 0.6 0.2 0.2

The row totals were random variables in the previous example. Here, the row totals are fixed. Let n1+ = 400 and n2+ = 600. Below is how I simulate a sample:

> set.seed(8111)

> save1 <- rmultinom(n = 1, size = 400, prob = pi.cond[1,])

> save2 <- rmultinom(n = 1, size = 600, prob = pi.cond[2,])

> c.table2 <- array(data = c(save1[1], save2[1], save1[2], save2[2], save1[3], save2[3]), dim = c(2,3), dimnames = list(X = 1:2, Y = 1:3))

> c.table2

 Y

X 1 2 3

 1 162 159 79

 2 351 126 123

> rowSums(c.table2)

 1 2

400 600

> c.table2/rowSums(c.table2)

 Y

X 1 2 3

 1 0.405 0.3975 0.1975

 2 0.585 0.2100 0.2050

> round(c.table1/rowSums(c.table1),4)

 Y

X 1 2 3

 1 0.389 0.4196 0.1914

 2 0.611 0.1866 0.2024