

Binary response data

A “Bernoulli trial” is a random variable that has two points in its sample space, e.g. “success/failure,” “heads/tails,” “yes/no,” “0/1.”

The probability distribution of a Bernoulli trial is completely determined by the “success probability” p .

Suppose we have a sequence of independent Bernoulli trials

$$X_1, \dots, X_n$$

with

$$P(X_i = 1) = p_i.$$

The X_i are identically distributed if the p_i are all the same.

The probability of observing a given value for X_i is

$$P(X_i) = p_i^{X_i}(1 - p_i)^{1-X_i}.$$

The probability of observing the sequence X_1, \dots, X_n is

$$p_1^{X_1}(1 - p_1)^{1-X_1} \dots p_n^{X_n}(1 - p_n)^{1-X_n} = \prod_i p_i^{X_i}(1 - p_i)^{1-X_i}.$$

The binomial distribution for the total number of successes

The total number of successes is

$$T = X_1 + \cdots + X_n = \sum_i X_i.$$

For example, if $n = 4$, the sequences 1100 and 1001 would both give $T = 2$.

If the X_i are iid Bernoulli trials, the probability distribution of T is the “binomial distribution,” where

$$P(T = k) = \binom{n}{k} p^k (1 - p)^{n-k}.$$

The “cumulative probabilities” of the binomial distribution are

$$P(T \leq k) = P(T = 0) + P(T = 1) + \dots + P(T = k).$$

These can be calculated in R using

```
pbinom(k, n, p)
```

which returns the value $P(X \leq k)$ for n iid Bernoulli trials with success probability p .

Calculating binomial probabilities

Although `pbinom` is available in R, it is worth considering how it is calculated.

In the binomial coefficient

$$\binom{n}{k} = \frac{n!}{(n-k)!k!},$$

overflow is possible. But on the log scale this becomes

$$\log \binom{n}{k} = \sum_{j=n-k+1}^n \log j - \sum_{j=1}^k \log j.$$

Calculating binomial probabilities

The following R code calculates the binomial probability $P(X = k)$:

```
P = 0
if ((k>0) & (k<n)) { P = sum(log(seq(n-k+1, n))) - sum(log(seq(1, k))) }
P = P + k*log(p) + (n-k)*log(1-p)
P = exp(P)
```

The normal and Poisson approximations to the binomial distribution

The cumulative probabilities of the binomial distribution

$$P(T \leq k) = \sum_{j \leq k} P(T = j)$$

cannot be expressed in a simple formula. Thus it is common to use an approximation.

To derive the normal approximation, recall that the expected value and variance of each X_i are

$$EX_i = p \qquad \text{var } X_i = p(1 - p).$$

The Normal approximation (continued)

Therefore

$$ET = np \qquad \text{var } T = np(1 - p).$$

Thus the standardization of T is

$$\frac{T - np}{\sqrt{np(1 - p)}},$$

which by the central limit theorem will approximately follow a standard normal distribution when n is not too small.

The Normal approximation (continued)

Suppose we want to find the tail probability $P(T > k)$. Standardizing yields

$$P\left(\frac{T - np}{\sqrt{np(1-p)}} > \frac{k - np}{\sqrt{np(1-p)}}\right) = 1 - F\left(\frac{k - np}{\sqrt{np(1-p)}}\right),$$

where F is the standard normal CDF (pnorm in R).

The Poisson approximation

The Poisson approximation, is more accurate than the normal approximation when p is small. A Poisson random variable G has sample space $0, 1, 2, \dots$, with probabilities

$$P(G = k) = e^{-\lambda} \lambda^k / k!,$$

where $\lambda > 0$ is a parameter.

Using $\lambda = np$ provides a good approximation to the binomial distribution with sample size n and success probability p , if p is small.

Comparisons of the normal and Poisson approximations

The following program uses R to calculate the exact Binomial right tail probability $P(T > k)$ and the approximations to this value using the normal and Poisson distributions.

```
n = 20
k = 2

M = NULL

## Consider different success probabilities.
for (p in c(0.01, 0.1, 0.2, 0.3, 0.4, 0.5))
{
  ## The probability from the normal approximation.
  N = 1 - pnorm((k-n*p)/sqrt(n*p*(1-p)))

  ## The probability from the Poisson approximation.
  P = 1 - ppois(k, n*p)

  ## The exact value from the binomial distribution.
```

```
B = 1 - pbinom(k, n, p)

M = rbind(M, c(p, B, N, P, (N-B)/B, (P-B)/B))
}

print(round(M, 3))
```

Based on the output to this program you will see that the normal approximation is closer to the exact value than the Poisson approximation when $p \approx 0.3$ or greater, but for smaller values of p , the Poisson approximation is more accurate.

Bivariate binary data – contingency tables

Suppose each individual in a random sample is measured in terms of two different binary variables X and Y . For example, individuals in a drug trial may have their gender (F/M) recorded, as well as their response to the drug (yes/no).

We can represent the data as a contingency table:

	Response	
	Y	N
F	n_{11}	n_{12}
M	n_{21}	n_{22}

The contingency table describes a sample of size $n = n_{11} + n_{12} + n_{21} + n_{22}$ from the probability distribution

	$Y = 1$	$Y = 2$
$X = 1$	p_{11}	p_{12}
$X = 2$	p_{21}	p_{22}

The expected value of n_{ij} is np_{ij} , and n_{ij} follows a binomial distribution with parameters p_{ij} and n . However n_{11} , n_{12} , n_{21} , and n_{22} are not independent.

Suppose we wish to simulate a sample of size n from the contingency table specified above. Put the four cells of the contingency table in the arbitrary order 11, 12, 21, 22, so the cumulative probabilities are

$$\begin{aligned}c_1 &= p_{11} \\c_2 &= p_{11} + p_{12} \\c_3 &= p_{11} + p_{12} + p_{21} \\c_4 &= 1.\end{aligned}$$

If U is uniformly distributed on $(0, 1)$,

$$\begin{aligned}P(U < c_1) &= c_1 = p_{11} \\P(c_1 \leq U < c_2) &= c_2 - c_1 = p_{12} \\P(c_2 \leq U < c_3) &= c_3 - c_2 = p_{21} \\P(c_3 \leq U) &= c_4 - c_3 = p_{22}.\end{aligned}$$

We can simulate the n_{11} , n_{12} , n_{21} , and n_{22} by simulating uniform random values U_1, \dots, U_n . For each U_i we add 1 to one of the cells, as follows.

$$\begin{array}{ll} U < c_1 & \text{add 1 to } n_{11} \\ c_1 \leq U < c_2 & \text{add 1 to } n_{12} \\ c_2 \leq U < c_3 & \text{add 1 to } n_{21} \\ c_3 \leq U < c_4 & \text{add 1 to } n_{22}. \end{array}$$

Here is a simple simulation study to show that the frequencies of the four cells agree with their probabilities.

```
## Set the cell probabilities here.
p11 = 0.2
p12 = 0.3
p21 = 0.1
p22 = 0.4

## Cumulative probabilities.
c1 = p11
c2 = c1+p12
c3 = c2+p21
c4 = c3+p22

## Simulate a contingency table.
U = runif(1e4)
N = array(0, c(2,2))
N[1,1] = sum(U <= c1)
N[1,2] = sum( (U > c1) & (U <= c2) )
N[2,1] = sum( (U > c2) & (U <= c3) )
```

```
N[2,2] = sum(U > c3)
```

The odds ratio and log odds ratio

For a univariate Bernoulli trial with success probability p , the “odds” is the ratio of the success probability to the failure probability:

$$p/(1 - p).$$

In a contingency table, if we know that $X = 1$, the probability that $Y = 1$ is $p_{11}/(p_{11} + p_{12})$. If we know that $X = 2$, the probability that $Y = 1$ is $p_{21}/(p_{21} + p_{22})$

Therefore if $X = 1$, the odds of $Y = 1$ versus $Y = 2$ are p_{11}/p_{12} . Similarly, if $X = 2$, the odds of $Y = 1$ versus $Y = 2$ are p_{21}/p_{22} .

The “odds ratio” is the ratio of the odds of Y when $X = 1$ to the odds of Y when $X = 2$:

$$\frac{p_{11}/p_{12}}{p_{21}/p_{22}} = \frac{p_{11}p_{22}}{p_{12}p_{21}}.$$

If the odds ratio is greater than 1, concordant responses ($X = 1, Y = 1$ or $X = 2, Y = 2$) are more common than discordant responses ($X = 1, Y = 2$ or $X = 2, Y = 1$). If the odds ratio is less than 1, discordant responses are more common.

An important property of the odds ratio is that if the roles of X and Y are switched, the odds ratio is unchanged.

The odds of $X = 1$ versus $X = 2$ when $Y = 1$ are p_{11}/p_{21} . The odds of $X = 1$ versus $X = 2$ when $Y = 2$ are p_{12}/p_{22} . Viewed this way, the odds ratio is

$$\frac{p_{11}/p_{21}}{p_{12}/p_{22}} = \frac{p_{11}p_{22}}{p_{12}p_{21}},$$

which is the same as we had above.

Tests of independence

An important question about a contingency table is whether X and Y are independent. When this is the case, two numbers p and q can be found so that the population probability distribution can be written

$$\frac{pq}{(1-p)q} \mid \frac{p(1-q)}{(1-p)(1-q)}$$

The contingency table can be written this way if and only if the odds ratio is one.

The log odds ratio

It is common to work with the log-transformed odds ratio,

$$\log p_{11} + \log p_{22} - \log p_{12} - \log p_{21}.$$

The log odds ratio is zero when X and Y are independent. It is positive when concordant responses are more common than discordant responses, and is negative when discordant responses are more common than concordant responses.

Estimation and inference for the log odds ratio

The sample odds ratio

$$\frac{n_{11}n_{22}}{n_{12}n_{21}}$$

estimates the population odds ratio

$$\frac{p_{11}p_{22}}{p_{12}p_{21}}.$$

The sample log odds ratio

$$\log n_{11} + \log n_{22} - \log n_{12} - \log n_{21}$$

estimates the population log odds ratio

$$\log p_{11} + \log p_{22} - \log p_{12} - \log p_{21}.$$

The standard error of the log odds ratio is approximately

$$\sqrt{\frac{1/p_{11} + 1/p_{12} + 1/p_{21} + 1/p_{22}}{n}}.$$

Since we don't know the p_{ij} , in practice the plug-in estimate of the standard error is used

$$\sqrt{1/n_{11} + 1/n_{12} + 1/n_{21} + 1/n_{22}}.$$

The following simulation evaluates the coverage properties of the 95% confidence interval based on the plug-in standard error estimate for the log odds ratio. Note that if any of the cell counts n_{ij} are zero, the standard error is infinite. In this case the confidence interval always covers the true value.

```
## Simulate a 2x2 table with sample size n and cell probabilities
## given in P.
simtab = function(P, n)
{
  ## Convert to cumulative probabilities.
  CP = cumsum(P)

  ## Storage for the data being simulated.
  N = array(0, c(2,2))

  ## Simulate one contingency table.
  U = runif(n)
  N[1,1] = sum(U <= CP[1])
  N[1,2] = sum( (U > CP[1]) & (U <= CP[2]) )
  N[2,1] = sum( (U > CP[2]) & (U <= CP[3]) )
  N[2,2] = sum(U > CP[3])
}
```

```
    return(N)
}

## The sample size.
n = 50

## Array of coverage indicators.
C = array(0, 1000)

for (k in 1:1000)
{
  ## Generate the four cell probabilities (p11, p12, p21, p22).
  P = 0.1 + 0.8*runif(4)
  P = P / sum(P)

  N = simtab(P, n)

  ## The sample log-odds ratio and its standard error.
  LR = log(N[1,1]) + log(N[2,2]) - log(N[1,2]) - log(N[2,1])
  SE = sqrt(1/N[1,1] + 1/N[1,2] + 1/N[2,1] + 1/N[2,2])
}
```

```
## The population log-odds ratio.  
PLR = log(P[1]) + log(P[4]) - log(P[2]) - log(P[3])  
  
## Check for coverage.  
if (!is.finite(SE)) { C[k] = 1 }  
else { C[k] = (LR-2*SE < PLR) & (LR+2*SE > PLR) }  
}
```