

ST 502 Final

Katelyn Settlemyre, Julia Farrell

2022-12-04

Introduction

This report will explore the Chi-Square test for homogeneity in detail. We will derive the likelihood ratio test (LRT) statistic used to conduct this test and explain the Pearson Chi-Square statistic that can be used as an approximation. Once the theory is well established, we then conduct a simulation. The goal is to determine how well the asymptotic rejection region performs at controlling the alpha level of the Pearson Chi-Square test and to determine the power of the asymptotic test when comparing certain alternative hypotheses.

The Chi-Square test for homogeneity is used in a specific case: comparing J multinomials with I classes (with $I, J \in \mathbb{N}$), where the researcher is interested in determining if the probabilities of each cell are the same across every multinomial.

Data Example

Use hospital data given to conduct a χ^2 test for homogeneity.

```
#create and print matrix of hospital data
rows <- rbind(a=c(41, 27, 51), b=c(36,3,40), c=c(169, 106, 109))
hospDat <- matrix(data=rows, nrow = 3, ncol = 3,
                  dimnames = list(c("A", "B", "C"),
                                c("Surgical Site Infections",
                                  "Pneumonia Infections",
                                  "Bloodstream Infections")))
summary(hospDat)
```

##	Surgical Site Infections	Pneumonia Infections	Bloodstream Infections
##	Min. : 36.0	Min. : 3.00	Min. : 40.00
##	1st Qu.: 38.5	1st Qu.: 15.00	1st Qu.: 45.50
##	Median : 41.0	Median : 27.00	Median : 51.00
##	Mean : 82.0	Mean : 45.33	Mean : 66.67
##	3rd Qu.: 105.0	3rd Qu.: 66.50	3rd Qu.: 80.00
##	Max. : 169.0	Max. : 106.00	Max. : 109.00

Now we will conduct a Chi-Square test for homogeneity using this sample data.

H_0 : The distribution of infections is the same for each hospital

H_1 : The distribution of infections is **not** the same for each hospital.

```
x = chisq.test(hospDat)
x
```

```
##
## Pearson's Chi-squared test
##
## data: hospDat
## X-squared = 30.696, df = 4, p-value = 3.531e-06
```

```
x$expected
```

```
## Surgical Site Infections Pneumonia Infections Bloodstream Infections
## A          50.29897          27.80756          40.89347
## B          33.39175          18.46048          27.14777
## C          162.30928          89.73196          131.95876
```

The p-value of our chi-square test statistic is extremely small, indicating that we reject the null hypothesis. The data in this example provides support for the alternative hypothesis that the multinomials from these hospitals are not homogeneous.

Deriving the Likelihood Ratio Test

The goal is to derive the likelihood ratio test for a generalized case comparing J independent multinomial distributions, each with I categories.

Let there be J independent multinomial distributions, each with I categories, where $I, J \in \mathbb{N}$. We want to test the hypothesis $H_0 = \pi_{11} = \pi_{12} = \dots = \pi_{1J}, \pi_{21} = \pi_{22} = \dots = \pi_{2J}, \pi_{I1} = \pi_{I2} = \dots = \pi_{IJ}$ vs. H_a : at least one probability differs

To derive the likelihood ratio test, we will initially look at the likelihood function, which is just the product of the J multinomials.

Expected Counts Under H_0 : $L(\pi'_{ij}|s) = \prod_{j=1}^J \binom{0}{0} \cdot \pi_{ij}^{n_{ij}} \cdot \pi_{2j}^{n_{2j}} \cdot \dots \cdot \pi_{Ij}^{n_{Ij}} \propto \prod_{j=1}^J \prod_{i=1}^I \pi_{ij}^{n_{ij}}$ subject to constraints

Under the null hypothesis, $\pi_{11} = \pi_{12} = \dots = \pi_{1J}$, so we will replace this with the common value π_1 . Similarly, we will continue forward considering the common values π_1, \dots, π_I . There is one restriction on these probabilities to make them valid, which is $\pi_1 + \pi_2 + \dots + \pi_I = 1$

Degrees of freedom for reference distribution:

$$\dim(\Omega) = J \cdot (I - 1)$$

$$\dim(\omega) = (I - 1)$$

$$df = J(I - 1) - (I - 1) = (I - 1)(J - 1)$$

.

. More derivation steps

.

.

The likelihood ratio test for the homogeneity hypothesis being tested is therefore given by $LRT = 2 \sum_{j=1}^J \sum_{i=1}^I Obs_{ij} \cdot \ln\left(\frac{Obs_{ij}}{Exp_{ij}}\right)$

Simulation

Goal:

- Determine how well the asymptotic rejection region performs at controlling α .
- Determine the power of the asymptotic test when comparing certain alternative situations.

Process:

```

#set seed for reproduction
set.seed(17)

#sample sizes
n1 <- c(20,30,50,100)
n2 <- n1
nCombos <- expand.grid(n1=n1, n2=n2)

#probabilities - three cases being tested
p1 <- c(1/3, 1/3, 1/3) #equal
p2 <- c(0.1, 0.3, 0.6) #mixed 1
p3 <- c(0.1, 0.1, 0.8) #mixed 2

#Use N=50000 random tables (start lower, end up here)
N <- 50000

# number of classes & multinomials being compared
I = 3 #classes
J = 2 #multinomials

# function to create two multinomials, using sample size and probabilities input
### sampleSize: (int)
### probs: (list of doubles) length of p split determines number of classes. All = 3 for this simulation
### returns the approximate alpha values from simulated multinomials

#create function to generate tables like in hospital example
multGenerate <- function(size1, size2, prob) {
  #generate multinomials
  mult1 <- rmultinom(1, size1, prob)
  mult2 <- rmultinom(1, size2, prob)
  #combine multinomials into table
  multTab <- cbind(mult1, mult2)
  #transpose table to have multinomials as the rows
  multTrans = t(multTab)
  #return table
  return(multTrans)
}

#testing in wrapper function
wrapAlpha <- function(size1, size2, prob){
  #generate table
  a <- multGenerate(size1, size2, prob)
  #compute chisq and compare to theoretical cutoff
  #here theoretical alpha = 0.05, but can easily update function to pick alpha
  #df = (I-1)(J-1) = (3-1)(2-1) = 2
  b <- chisq.test(a)
  c <- isTRUE(b$statistic > qchisq(0.95, 2))
  #return T/F value
  return(c)
}

##test wrapper; replicate many many times and store as vector

```

```

#alph <- replicate(5000, wrapAlpha(n1[2], n2[2],p1))
##take the mean of the vector to approximate alpha level
#mean(alph)

#set up vectors to store approximated alpha levels
p1Alphas <- c()
p2Alphas <- c()
p3Alphas <- c()

#for each combination of sample sizes, approximate alpha with each set of probabilities
for (i in 1:4) {
  for(j in 1:4){
    p1Alphas <- append(p1Alphas, mean(replicate(N,wrapAlpha(n1[i], n2[j], p1))))
  }
}

for (i in 1:4) {
  for(j in 1:4){
    p2Alphas <- append(p2Alphas, mean(replicate(N,wrapAlpha(n1[i], n2[j], p1))))
  }
}

for (i in 1:4) {
  for(j in 1:4){
    p3Alphas <- append(p3Alphas, mean(replicate(N,wrapAlpha(n1[i], n2[j], p1))))
  }
}

#combine all into data frame
n_1 <- c(20,20,20,20,30,30,30,30,50,50,50,50,100,100,100,100)
n_2 <- c(20,30,50,100,20,30,50,100,20,30,50,100,20,30,50,100)
alP1 <- replicate(16, "equal")
alP2 <- replicate(16, "mixed1")
alP3 <- replicate(16, "mixed2")
d1<-data.frame(alpha=p1Alphas, n1=n_1, n2=n_2, p=alP1)
d2<-data.frame(alpha=p2Alphas, n1=n_1, n2=n_2, p=alP2)
d3<-data.frame(alpha=p3Alphas, n1=n_1, n2=n_2, p=alP3)
alphaDat <- rbind(d1,d2,d3)

#plot

#Power inspection
#compare 1-2, 1-3, 2-3

#Similar to multGenerate, but allows for 2 different probabilities
pwrCompare <- function(size1, size2, prob1, prob2) {
  #generate multinomials
  mult1 <- rmultinom(1, size1, prob1)
  mult2 <- rmultinom(1, size2, prob2)

```

```

#combine multinomials into matrix
multTab <- cbind(mult1, mult2)
#transpose matrix
multTrans = t(multTab)
return(multTrans)
}

wrapPWR <- function(size1, size2, prob1, prob2){
  #generate table
  a <- pwrCompare(size1, size2, prob1, prob2)
  #compute chisq and compare to theoretical cutoff
  #here theoretical alpha = 0.05, but can easily update function to pick alpha
  #df = (I-1)(J-1) = (3-1)(2-1) = 2
  b <- chisq.test(a)
  c <- isTRUE(b$statistic > qchisq(0.95, 2))
  #return T/F value
  return(c)
}

##test wrapper; replicate many many times and store as vector
#pwr <- replicate(5000, wrapPWR(n1[3], n2[3],p1,p2))
##take the mean of the vector to approximate alpha level
#mean(pwr)

#set up vectors to store approximated power
pwrp1p2 <- c()
pwrp1p3 <- c()
pwrp2p3 <- c()

#for each combination of sample sizes, approximate power with each pair of probabilities
for (i in 1:4) {
  for(j in 1:4){
    pwrp1p2 <- append(pwrp1p2, mean(replicate(N,wrapPWR(n1[i], n2[j], p1,p2))))
  }
}

for (i in 1:4) {
  for(j in 1:4){
    pwrp1p3 <- append(pwrp1p3, mean(replicate(N,wrapPWR(n1[i], n2[j], p1,p3))))
  }
}

for (i in 1:4) {
  for(j in 1:4){
    pwrp2p3 <- append(pwrp2p3, mean(replicate(N,wrapPWR(n1[i], n2[j], p2,p3))))
  }
}

#combine all into data frame
be12 <- replicate(16, "1v2")
be13 <- replicate(16, "1v3")
be23 <- replicate(16, "2v3")
b1<-data.frame(power=pwrp1p2, n1=n_1, n2=n_2, p=be12)

```

```
b2<-data.frame(power=pwrp1p3, n1=n_1, n2=n_2, p=be13)
b3<-data.frame(power=pwrp2p3, n1=n_1, n2=n_2, p=be23)
powerDat <- rbind(b1,b2,b3)
```

```
#create graphs
```