

Stat 5421 Lecture Notes: Simple Chi-Square Tests for Contingency Tables

Charles J. Geyer

October 02, 2023

Contents

1	License	1
2	One-Way Contingency Table	1
2.1	Goodness of Fit Test, Completely Specified Null Hypothesis	1
2.2	Hypothesis Tests with Composite Null Hypotheses	4
3	Using R Function GLM	10
4	Using R Function D for Derivatives	11

1 License

This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License (<http://creativecommons.org/licenses/by-sa/4.0/>).

2 One-Way Contingency Table

The data set read in by the R function `read.table` below simulates 6000 rolls of a fair die (singular of dice).

```
foo <- read.table(  
  url("http://www.stat.umn.edu/geyer/5421/mydata/multi-simple-1.txt"),  
  header = TRUE)  
print(foo)
```

```
##      y num  
## 1 1038  1  
## 2  964  2  
## 3  975  3  
## 4  983  4  
## 5 1035  5  
## 6 1005  6
```

2.1 Goodness of Fit Test, Completely Specified Null Hypothesis

We test the hypothesis that all six cells of the contingency table have the same probability (null hypothesis) versus that they are not that (alternative hypothesis).

Null Hypothesis (zero parameters)

Number	Probability
1	1/6
2	1/6
3	1/6
4	1/6
5	1/6
6	1/6

Alternative Hypothesis (5 parameters)

Number	Probability
1	π_1
2	π_2
3	π_3
4	π_4
5	π_5
6	$1 - \pi_1 - \pi_2 - \pi_3 - \pi_4 - \pi_5$

2.1.1 Pearson Chi-Squared Test Statistic

```
out <- with(foo, chisq.test(y))
print(out)
```

```
##
## Chi-squared test for given probabilities
##
## data: y
## X-squared = 4.904, df = 5, p-value = 0.4277
```

The default test statistic for the R function `chisq.test` is the Pearson chi-squared test statistic

$$X^2 = \sum_{\text{all cells of table}} \frac{(\text{observed} - \text{expected})^2}{\text{expected}}. \quad (1)$$

We write this in more mathematical notation as follows. Let I denote the index set for the contingency table, which is one-dimensional in this example, but may have any dimension in general, so making our subscript i range over an abstract set rather than a range of numbers means we do not have to have a different formula for every dimension of table. Let x_i denote the observed count for cell i , $\tilde{\pi}_i$ the cell probability for the i -cell under the null hypothesis, and n the sample size (so the random vector having components x_i is multinomial with sample size n and parameter vector having components $\tilde{\pi}_i$). Then the Pearson test statistic is

$$X^2 = \sum_{i \in I} \frac{(x_i - n\tilde{\pi}_i)^2}{n\tilde{\pi}_i}. \quad (2)$$

The Pearson chi-square test statistic is a special case of the Rao test statistic (also called score test and Lagrange multiplier test, see the likelihood handout for its definition).

Consequently, its asymptotic distribution is chi-square with degrees of freedom which is the difference of dimensions of the models being compared. Here the null model is completely specified (no adjustable parameter) so has dimension zero, and the multinomial has $k - 1$ degrees of freedom if the table has k cells because probabilities must sum to one, that is $\sum_{i \in I} \pi_i = 1$ so specifying $k - 1$ parameters determines the last one. Hence the degrees of freedom of the asymptotic chi-square distribution is $k - 1$. Here $k = 6$ so the degrees of freedom is 5, which is what the printout of the `chisq.test` function says.

2.1.2 Likelihood Ratio Test Statistic

We can also do a Wilks test (also called likelihood ratio test).

```
Gsq <- 2 * sum(out$observed * log(out$observed / out$expected))
print(Gsq)
```

```
## [1] 4.894725
```

```
Gsq.pval <- pchisq(Gsq, out$parameter, lower.tail = FALSE)
print(Gsq.pval)
```

```
## [1] 0.4288628
```

The probability mass function (PMF) for the multinomial distribution is

$$f_{\pi}(\mathbf{x}) = \binom{n}{\mathbf{x}} \prod_{i \in I} \pi_i^{x_i},$$

where the thingy with the round brackets is a multinomial coefficient. Since that does not contain the parameter, we can drop it in the likelihood

$$L(\pi) = \prod_{i \in I} \pi_i^{x_i},$$

so the log likelihood is

$$l(\pi) = \sum_{i \in I} x_i \log(\pi_i)$$

(in this handout we are using the convention that boldface denotes a vector and normal font its components, so π is the vector having components π_i). If $\hat{\pi}$ and $\tilde{\pi}$ are the maximum likelihood estimates (MLE) in the alternative and null hypotheses, respectively, then the Wilks statistic is

$$G^2 = 2[l(\hat{\pi}) - l(\tilde{\pi})] = 2 \sum_{i \in I} x_i \log \left(\frac{\hat{\pi}_i}{\tilde{\pi}_i} \right) \quad (3)$$

We already know the MLE in the null hypothesis (completely specified, all cell probabilities the same). To determine the MLE in the alternative hypothesis, we need to solve the likelihood equations for the multinomial distribution, and this is a bit tricky because of the degeneracy of the multinomial. We have to write one parameter as a function of the others to get down to $k - 1$ freely adjustable variables.

Fix $i^* \in I$ and eliminate that variable writing $I^* = I \setminus \{i^*\}$ so

$$l(\pi) = \sum_{i \in I^*} x_i \log(\pi_i) + x_{i^*} \log \left(\sum_{i \in I^*} \pi_i \right)$$

so for $i \in I^*$

$$\frac{\partial l(\pi)}{\partial \pi_i} = \frac{x_i}{\pi_i} - \frac{x_{i^*}}{1 - \sum_{i \in I^*} \pi_i} = \frac{x_i}{\pi_i} - \frac{x_{i^*}}{\pi_{i^*}}$$

Setting derivatives to zero and solving for the parameters gives

$$\pi_i = x_i \cdot \frac{\pi_{i^*}}{x_{i^*}}, \quad i \in I^*. \quad (4)$$

In order to make progress we have to figure out what π_{i^*} is. The only facts we have at our disposal to do that is that probabilities sum to one and cell counts sum to n . So we first note that (4) trivially holds when $i = i^*$, so summing both sides of (4) over all i gives

$$\sum_{i \in I} \pi_i = \frac{\pi_{i^*}}{x_{i^*}} \sum_{i \in I} x_i$$

or

$$1 = \frac{\pi_i^*}{x_{i^*}} \cdot n$$

or

$$\pi_i^* = \frac{x_{i^*}}{n}$$

and plugging this into (4) and writing $\hat{\pi}_i$ rather than π_i because our solution is the MLE gives

$$\hat{\pi}_i = \frac{x_i}{n}, \quad i \in I. \tag{5}$$

So we have found that the unrestricted MLE for the multinomial distribution is just the observed cell counts divided by the sample size. This is, of course, the obvious estimator, but we didn't know it was the MLE until we did the math.

We now also rewrite (3) gratuitously inserting n 's in the numerator and denominator so it uses the quantities used in the Pearson statistic

$$\begin{aligned} G^2 &= \sum_{i \in I} x_i \log \left(\frac{n\hat{\pi}_i}{n\tilde{\pi}_i} \right) \\ &= \sum_{\text{all cells of table}} \text{observed} \cdot \log \left(\frac{\text{observed}}{\text{expected}} \right) \end{aligned} \tag{6}$$

And that explains the calculation done in R.

Note that (6) is only for the case where we have the general alternative (cell probabilities can be anything). The general formula for comparing any two models is (3).

The tests are asymptotically equivalent so it is no surprise that the test statistics are very similar (4.904 and 4.8947), as are the P-values (0.4277 and (0.4289). Since the P-values are not small, the null hypothesis is accepted. The die seems fair.

2.2 Hypothesis Tests with Composite Null Hypotheses

The data set read in by the R function `read.table` below simulates 6000 rolls of an unfair die of the type known as six-ace flats. The one and six faces, which are on opposite sides of the die, are shaved slightly so the other faces have smaller area and smaller probability.

```
foo <- read.table(  
  url("http://www.stat.umn.edu/geyer/5421/mydata/multi-simple-2.txt"),  
  header = TRUE)  
foo
```

```
##      y num  
## 1 1047  1  
## 2 1017  2  
## 3  951  3  
## 4 1004  4  
## 5  952  5  
## 6 1029  6
```

2.2.1 Naive Goodness of Fit Test

(This test does not actually belong in this section. It has a simple null hypothesis.)

We start by testing the hypothesis that all six cells of the contingency table have the same probability (just like in the preceding section).

```

y <- foo$y
### Pearson chi-square test statistic
out <- chisq.test(y)
print(out)

##
## Chi-squared test for given probabilities
##
## data: y
## X-squared = 8.06, df = 5, p-value = 0.153
### likelihood ratio test statistic
Gsq <- 2 * sum(out$observed * log(out$observed / out$expected))
print(Gsq)

## [1] 8.094506
pchisq(Gsq, out$parameter, lower.tail = FALSE)

## [1] 0.1511036

```

The P -values are lower than before, so perhaps slightly suggestive that the null hypothesis is false, but still much larger than 0.05 (not that your humble author is suggesting that 0.05 is the dividing line between statistical significance and statistical non-significance, but these are not very low P -values). Hence we could again conclude the die seems fair. But now that would be wrong because we haven't even fit the six-ace hypothesis yet!

2.2.2 Goodness of Fit Test of Six-Ace Flats Hypothesis

2.2.2.1 Hypotheses

We test the six-ace-flats hypothesis versus the general multinomial distribution (alternative hypothesis).

Null Hypothesis (1 parameter)

Number	Probability
1	π_1
2	$(1 - 2\pi_1)/4$
3	$(1 - 2\pi_1)/4$
4	$(1 - 2\pi_1)/4$
5	$(1 - 2\pi_1)/4$
6	π_1

Alternative Hypothesis (5 parameters)

Number	Probability
1	π_1
2	π_2
3	π_3
4	π_4
5	π_5
6	$1 - \pi_1 - \pi_2 - \pi_3 - \pi_4 - \pi_5$

2.2.2.2 Maximum Likelihood Estimate

A calculation similar to the one in the preceding section (which we will mercifully not do, it will eventually be justified by the observed-equals-expected principle) says that the MLE for the six-ace hypothesis estimates the cell probabilities for the six and ace cells to be the average of the observed cell frequencies for these two cells

$$\hat{\pi}_1 = \hat{\pi}_6 = \frac{x_1 + x_6}{2n}$$

(because they are assumed to have the same area) and similarly for the other cells

$$\hat{\pi}_2 = \hat{\pi}_3 = \hat{\pi}_4 = \hat{\pi}_5 = \frac{x_2 + x_3 + x_4 + x_5}{4n}$$

(because they are assumed to have the same area).

This is the MLE for the six-ace flats model. The dimension of this model is one because if we know $\hat{\pi}_1$, then we can also figure out all of the other cell probabilities from the constraints that probabilities sum to one and the equality constraints above. Hence this model has dimension one.

```
nrolls <- sum(y)
pi.hat <- rep(NA, 6)
pi.hat[c(1, 6)] <- sum(y[c(1, 6)]) / 2 / nrolls
pi.hat[- c(1, 6)] <- sum(y[- c(1, 6)]) / 4 / nrolls
print(pi.hat)
```

```
## [1] 0.1730 0.1635 0.1635 0.1635 0.1635 0.1730
```

2.2.2.3 Pearson Chi-Squared Test

One might think that we could just use R function `chisq.test` as before.

```
out <- chisq.test(y, p = pi.hat)
print(out)
```

```
##
## Chi-squared test for given probabilities
##
## data: y
## X-squared = 3.7911, df = 5, p-value = 0.5799
```

But this is incorrect, because R function `chisq.test` does not know we estimated one parameter in the null hypothesis and this changes the degrees of freedom. And there is no way to tell it about this.

It does correctly calculate the Pearson chi-squared test statistic.

```
tstat <- out$statistic
print(tstat)
```

```
## X-squared
## 3.791136
```

But the degrees of freedom is one less than what it thinks because we estimated one parameter in the null hypothesis. In general, if we estimated k parameters, then the degrees of freedom would be k less.

So the P -value for the Pearson chi-squared test is

```
df <- out$parameter - 1
print(df)
```

```
## df
## 4
```

```
pchisq(tstat, df = df, lower.tail = FALSE)
```

```
## X-squared  
## 0.4350099
```

2.2.2.4 Likelihood Ratio Test

We can also do a likelihood ratio test instead.

```
expected <- out$expected  
print(expected)
```

```
## [1] 1038 981 981 981 981 1038
```

```
observed <- y
```

```
tstat <- 2 * sum(observed * log(observed / expected))  
print(tstat)
```

```
## [1] 3.789174
```

```
pchisq(tstat, df = df, lower.tail = FALSE)
```

```
## [1] 0.4352892
```

We used (6) as the formula for the likelihood ratio test statistic.

2.2.3 Tests of Model Comparison

The preceding two sections were somewhat unsatisfactory. They seem to say that both null hypothesis (fair dice and six-ace flats (unfair dice)) fit the data. But they can't both be right. Of course the test statistics seem to say that the six-ace flats model fits even better than the fair dice model. But that is not how hypothesis tests work.

To really get a handle on these data, we need a better hypothesis test.

Goodness of fit tests always have anything at all as the alternative hypothesis.

General tests of model comparison can compare any two models, so long as they are nested (the null hypothesis is a submodel of the alternative hypothesis). Here the fair dice model is the special case of the six-ace flats model when $\alpha = 1/6$. So these models are nested. (For more on the concept of nested models, see the notes on likelihood.)

So now we want to compare these two models. One with MLE cell probabilities `pi.hat` and the other with MLE cell probabilities

```
pi.twiddle <- rep(1 / 6, 6)
```

The difference in number of (freely adjustable) parameters of the two models is one (one parameter in six-ace flats, zero parameters in fair dice). So the test statistics (Wald, Wilks, and Rao) are approximately chi-square on one degree of freedom.

Null Hypothesis (zero parameters)

Number	Probability
1	1/6
2	1/6
3	1/6
4	1/6
5	1/6

Number	Probability
6	1/6

Alternative Hypothesis (1 parameter)

Number	Probability
1	π_1
2	$(1 - 2\pi_1)/4$
3	$(1 - 2\pi_1)/4$
4	$(1 - 2\pi_1)/4$
5	$(1 - 2\pi_1)/4$
6	π_1

2.2.3.1 Likelihood Ratio Test

The likelihood ratio test is

```
### likelihood ratio test statistic
Gsq <- 2 * sum(y * log(pi.hat / pi.twiddle))
print(Gsq)
```

```
## [1] 4.305331
```

```
Gsq.pval <- pchisq(Gsq, 1, lower.tail = FALSE)
print(Gsq.pval)
```

```
## [1] 0.03799309
```

We used (3) as the formula for the likelihood ratio test statistic.

Now $P = 0.038$ is moderately strong evidence that the null hypothesis (fair die) is false and the alternative (six-ace flat) is true. The moral of the story: you cannot conclude anything about a hypothesis (model) until you have actually fit that hypothesis (model), and, if you want to compare two models, then use a test of model comparison comparing them.

2.2.3.2 Rao Test

The Rao test depends on what we take to be the parameter of the six-ace hypothesis, say the probability α for the six-ace cells. If the cell probabilities are $(\alpha, \beta, \beta, \beta, \beta, \alpha)$, then

$$\beta = \frac{1 - 2\alpha}{4}$$

and

$$l(\alpha) = (x_1 + x_6) \log(\alpha) + (x_2 + x_3 + x_4 + x_5) \log(\beta)$$

so

$$\begin{aligned} l'(\alpha) &= \frac{x_1 + x_6}{\alpha} + \frac{x_2 + x_3 + x_4 + x_5}{\beta} \cdot \frac{d\beta}{d\alpha} \\ &= \frac{x_1 + x_6}{\alpha} - \frac{x_2 + x_3 + x_4 + x_5}{2\beta} \end{aligned} \quad (7)$$

and

$$\begin{aligned} l''(\alpha) &= -\frac{x_1 + x_6}{\alpha^2} - \frac{x_2 + x_3 + x_4 + x_5}{\beta^2} \cdot \left(\frac{d\beta}{d\alpha}\right)^2 \\ &= -\frac{x_1 + x_6}{\alpha^2} - \frac{x_2 + x_3 + x_4 + x_5}{4\beta^2} \end{aligned} \quad (8)$$

so observed Fisher information is

$$J(\alpha) = \frac{x_1 + x_6}{\alpha^2} + \frac{x_2 + x_3 + x_4 + x_5}{4\beta^2}$$

and expected Fisher information is

$$\begin{aligned} I(\alpha) &= \frac{n\alpha + n\alpha}{\alpha^2} + \frac{n\beta + n\beta + n\beta + n\beta}{4\beta^2} \\ &= \frac{2n}{\alpha} + \frac{n}{\beta} \end{aligned} \tag{9}$$

Hence the Rao test statistic is

$$R = l'(\tilde{\alpha})^2 / I(\tilde{\alpha})$$

where $\tilde{\alpha}$ is the value specified by the null hypothesis (which is $1/6$).

```
pi.zero <- pi.twiddle[1]
rao <- ((y[1] + y[6]) - sum(y[2:5]) / 2) / pi.zero^2 / (3 * sum(y) / pi.zero)
print(rao)
```

```
## [1] 4.332
```

```
rao.pval <- pchisq(rao, 1, lower.tail = FALSE)
print(rao.pval)
```

```
## [1] 0.03740227
```

2.2.3.3 Wald Test

The Wald test depends on what we take to be the constraint function, say

$$\alpha - 1/6$$

(using the notation of the preceding section). Then the Wald test statistic is

$$\hat{\alpha} - 1/6$$

divided by its estimated standard deviation, which we figured out in the preceding section. Fisher information for the six-ace-flats model is derived above (9). The only difference here is that for the Rao test we plug in the value $\tilde{\alpha}$ for the null hypothesis and for the Wald test we use the value $\hat{\alpha}$ for the alternative hypothesis.

```
alpha.hat <- pi.hat[1]
beta.hat <- (1 - 2 * alpha.hat) / 4
fish.hat <- 2 * nrolls / alpha.hat + nrolls / beta.hat
```

```
wald <- (alpha.hat - 1 / 6)^2 * fish.hat
print(wald)
```

```
## [1] 4.254241
```

```
wald.pval <- pchisq(wald, 1, lower.tail = FALSE)
print(wald.pval)
```

```
## [1] 0.03915244
```

2.2.4 Summary

```
fred <- matrix(c(Gsq, rao, wald, Gsq.pval, rao.pval, wald.pval), ncol = 2)
rownames(fred) <- c("Wilks", "Rao", "Wald")
colnames(fred) <- c("statistic", "p-value")
round(fred, 4)
```

```
##      statistic p-value
## Wilks    4.3053 0.0380
## Rao     4.3320 0.0374
## Wald    4.2542 0.0392
```

IMHO the likelihood ratio test is easier here. But your choice. They all say approximately the same thing.

3 Using R Function GLM

We can also assume Poisson sampling and use R function `glm`. To fit the six-ace-flats model we also need a “predictor” of the six and ace cells of the contingency table.

```
six_ace <- 1:6 %in% c(1, 6) |> as.numeric()
gout0 <- glm(y ~ 1, family = poisson)
gout1 <- glm(y ~ six_ace, family = poisson)
fitted(gout0)
```

```
##      1      2      3      4      5      6
## 1000 1000 1000 1000 1000 1000
```

```
fitted(gout1)
```

```
##      1      2      3      4      5      6
## 1038  981  981  981  981 1038
```

We see that we have fit the “same” models with the same expected values (but the first “same” is in scare quotes because here we are assuming Poisson sampling and before we were assuming multinomial sampling). And the `glm` method of R generic function `anova` does Rao and LRT tests.

```
anova(gout0, gout1, test = "LRT")
```

```
## Analysis of Deviance Table
##
## Model 1: y ~ 1
## Model 2: y ~ six_ace
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1         5     8.0945
## 2         4     3.7892  1  4.3053 0.03799 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(gout0, gout1, test = "Rao")
```

```
## Analysis of Deviance Table
##
## Model 1: y ~ 1
## Model 2: y ~ six_ace
##   Resid. Df Resid. Dev Df Deviance   Rao Pr(>Chi)
## 1         5     8.0945
## 2         4     3.7892  1  4.3053 4.332 0.0374 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

And these agree exactly with what we had under multinomial sampling (magic). And the `glm` method of R generic function `summary` does Wald tests.

```
summary(gout1)
```

```
##
## Call:
## glm(formula = y ~ six_ace, family = poisson)
##
## Coefficients:
##           Estimate Std. Error z value Pr(>|z|)
## (Intercept)  6.88857    0.01596 431.513  <2e-16 ***
## six_ace      0.05648    0.02714   2.081  0.0374 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
## Null deviance: 8.0945  on 5  degrees of freedom
## Residual deviance: 3.7892  on 4  degrees of freedom
## AIC: 60.26
##
## Number of Fisher Scoring iterations: 3
```

And the test for the coefficient for `six_ace` agrees with what we had under multinomial sampling. Except the test statistic here is normal and the test statistic there is chi-squared on one degree of freedom, so the square of the test statistic here is the square of the test statistic there. But the P -values are the same.

4 Using R Function `D` for Derivatives

We can avoid doing calculus and let R do it (for simple expressions)

```
foo <- quote((x1 + x6) * log(alpha) +
             (x2 + x3 + x4 + x5) * log((1 - 2 * alpha) / 4))
class(foo)
```

```
## [1] "call"
```

```
bar <- D(foo, "alpha")
bar
```

```
## (x1 + x6) * (1/alpha) - (x2 + x3 + x4 + x5) * (2/4/((1 - 2 *
## alpha)/4))
```

```
baz <- D(bar, "alpha")
baz
```

```
## -((x1 + x6) * (1/alpha^2) + (x2 + x3 + x4 + x5) * (2/4 * (2/4)/((1 -
## 2 * alpha)/4)^2))
```

Now `bar` is an R expression that expresses the first derivative of the log likelihood derived above (7). Now `baz` is an R expression that expresses the second derivative of the log likelihood derived above (8). log likelihood derived above (7).

And we can use these expressions as follows.

```
score.factory <- function(y) {
  for (i in 1:6) assign(paste0("x", i), y[i])
  function(alpha) eval(bar)
}
score <- score.factory(y)
score(alpha.hat)
```

```
## [1] 0
```

```
score(pi.zero)
```

```
## [1] 684
```

That `alpha.hat` is a zero of the score function proves it is the MLE for the alternative hypothesis (given the theory in the notes on exponential families). We need to standardize `score(pi.zero)` to get the Rao test statistic.

```
fisher.factory <- function(y) {  
  for (i in 1:6) assign(paste0("x", i), y[i])  
  function(alpha) - eval(baz)  
}  
fisher <- fisher.factory(y)  
score(pi.zero)^2 / fisher(pi.zero)
```

```
## [1] 4.251227
```

```
rao
```

```
## [1] 4.332
```

This differs from what we did above (`rao`) because before we used expected Fisher information and here we used observed Fisher information because the computer did not know how to go from observed to expected.