Test for Association of Categorical Variables

T.Scofield

You may click here to access the .qmd file.

The Big Picture

This test uses the chi-square statistic

$$\chi^2 = \sum_i \frac{(O_{i,j} - E_{i,j})^2}{E_{i,j}},$$

and quite closely resembles the goodness-of-fit test. The differences include

- Goodness-of-fit uses univariate data; the observed counts come from a frequency table of that variable. The test for association involves bivariate data; the observed counts come from a two-way table.
- The expected counts for the test for association are found using the totals in the margins of the two-way table. Specifically, the expected count in row i, column j is

$$E_{i,j} \;=\; \frac{(\text{total of row } i) \times (\text{total of column } j)}{\text{grand total}}.$$

- If the rule of thumb (all $E_{i,j} \geq 5$) is met, so that you may obtain a P-value using a theoretical chi-square distribution, you select the one with

$$df \;=\; [\#(\text{rows}) - 1] \times [\#(\text{columns}) - 1].$$

  for choosing the degrees of freedom.

We start with a simple example.

Example: Is there an association between the dominant hand of a child and that of the child's father?

Step 1: State hypotheses

$\mathbf{H}_0$: The dominant hand of a child and that of the child's father are independent.
$\mathbf{H}_a$: There is an association between the dominant hand of a child and that of the child's father.

Step 2: Compute a test statistic

We have data on these variables in the survey results found in the file

```r
ssurv <- read.csv("https://scofield.site/teaching/data/csv/ssurv.csv")
```

```r
names(ssurv)   # output has been suppressed
```

```r
tally(selfhandedness ~ dadhandedness, data=ssurv)   # summary two-way table
```

```
              dadhandedness
selfhandedness       L   R
                 1   0   0
               L 0   5  26
               R 2  23 223
```

Some respondents offer missing data, and the next line uses `filter()` to clean the data up a bit, giving us a better two-way table. I've used a pipe to `addmargins()` to give us marginal totals.

```r
domHandTable <- tally(selfhandedness ~ dadhandedness,
                      data=filter(ssurv, selfhandedness!="" & dadhandedness!=""))
domHandTable |> addmargins()
```

```
              dadhandedness
selfhandedness   L   R Sum
           L     5  26  31
           R    23 223 246
           Sum  28 249 277
```

The two-way table gives us four observed counts:

$$O_{1,1} = 5, \quad O_{1,2} = 26, \quad O_{2,1} = 23, \quad O_{2,2} = 223.$$

We get the corresponding four expected counts using the formula above:

$$E_{1,1} = \frac{(28)(31)}{277} = 3.13, \quad E_{1,2} = \frac{(249)(31)}{277} = 27.87, \quad E_{2,1} = \frac{(28)(246)}{277} = 24.87, \quad E_{2,2} = \frac{(249)(246)}{277} = 221.13.$$

Note that one is less than 5.

We can now compute the test statistic:

$$\chi^2 = \frac{(5-3.13)^2}{3.13} + \frac{(26-27.87)^2}{27.87} + \frac{(23-24.87)^2}{24.87} + \frac{(223-221.13)^2}{221.13} \doteq 1.399.$$

If we build lists in R from these numbers, we can use R to do this calculation:

```r
obs = c(5, 26, 23, 223)
expected = c(3.13, 27.87, 24.87, 221.13)
sum((obs-expected)^2/expected)
```

```
[1] 1.399113
```

Step 3: Compute a P-value

Because one expected count is under 5, we should use simulation to obtain a P-value. In class, I recommended another app, found at https://www.lock5stat.com/StatKey/advanced_association/advanced_association.html. Using that app, the approximate P-value is 0.328. Had we blundered forward and used a theoretical χ^2 distribution with `df`=1, we would have obtained a noticeably different P-value:

```r
1 - pchisq(1.393, df=1)
```

```
[1] 0.2378991
```

Step 4: Draw a conclusion.

At any of the usual significance levels, $alpha = 0.1$, $\alpha = 0.05$, or $\alpha = 0.01$, we would fail to reject the null hypothesis. This data is consistent with the two variables having independence.

Using the `chisq.test()` command

The command is meant as a short-cut to the various calculations involved in testing for association between categorical variables. It requires you to supply it with the two-way table, which means you have to build that table (without marginal totals). So, in order to use it on the dominant-hand data above:

```r
domHandTable <- tally(selfhandedness ~ dadhandedness,
                      data=filter(ssurv, selfhandedness!="" & dadhandedness!=""))
domHandTable     # this two-way table was displayed above, too
```

```
              dadhandedness
selfhandedness   L   R
             L   5  26
             R  23 223
```

```r
chisq.test(domHandTable)
```

```
Warning in chisq.test(domHandTable): Chi-squared approximation may be incorrect

	Pearson's Chi-squared test with Yates' continuity correction

data:  domHandTable
X-squared = 0.74638, df = 1, p-value = 0.3876
```

There are several things to take note of, here.

1. The most glaring thing is the warning: "Chi-squared approximation may be incorrect." This indicates that it obtained its P-value from a chi-square distribution, but it feels guilty about doing so, as not all expected counts were high enough.
2. The χ^2 statistic reported here is 0.74638. But we calculate it above to be 1.393. This discrepancy is due to "Yates' continuity correction." If we run, instead, the command

```r
chisq.test(domHandTable, correct=FALSE)   # I've suppressed the output
```

```
then the test statistic (as well as the corresponding $P$-value), will
match calculations we have performed above.
```

3. We can use the dollar-sign notation to make filter down results of the command to things we may want particularly.

```r
chisq.test(domHandTable)$expected      # gives only the expected counts
```

```
Warning in chisq.test(domHandTable): Chi-squared approximation may be incorrect

              dadhandedness
selfhandedness         L         R
             L  3.133574  27.86643
             R 24.866426 221.13357
```

```r
chisq.test(domHandTable)$statistic     # produces the chi-square statistic
```

```
Warning in chisq.test(domHandTable): Chi-squared approximation may be incorrect

X-squared
0.7463755
```

```r
chisq.test(domHandTable)$p.value       # produces the P-value
```

```
Warning in chisq.test(domHandTable): Chi-squared approximation may be incorrect
```

```
[1] 0.3876262
```

4. Finally, one can get `chisq.test()` to forego using a theoretical chi-square distribution and, in lieu of that, compute the *P*-value directly from a simulation. One simply adds the `simulate.p.value` switch:

```
chisq.test(domHandTable, simulate.p.value=TRUE)
```

```

	Pearson's Chi-squared test with simulated p-value (based on 2000
	replicates)

data:  domHandTable
X-squared = 1.3925, df = NA, p-value = 0.3538
```

That comprises the essentials for chi-square tests for an association. There are still some scenarios where extra tips may be helpful. Below, I have included a few.

Extra R Tips

Building a two-way table when you only have frequency data

The website https://www.datacamp.com/tutorial/contingency-analysis-r gives a two-way table, already built, offering sport-choices by gender. When you have a ready table, but not yet available in R, you can build it directly. I build the one from the website above using commands like these:

```
sexAndSportTable <- rbind( c(35, 15, 50), c(10, 30, 60) )
rownames(sexAndSportTable) <- c('Female', 'Male')
colnames(sexAndSportTable) <- c('Archery', 'Boxing', 'Cycling')
sexAndSportTable
```

```
       Archery Boxing Cycling
Female      35     15      50
Male        10     30      60
```

Such a table can be handed directly to the `chisq.test()` command for its various computations (the results of which I suppress here):

```
chisq.test(sexAndSportTable)$expected
chisq.test(sexAndSportTable)$statistic
chisq.test(sexAndSportTable)$p.value
```

```r
chisq.test(sexAndSportTable)
```

A more round-about method that has its uses is to build a raw data frame containing this data, and then using tally:

```r
sexAndSportRawData <- rbind(
  do(35) * c(sex = "Female", sport="Archery"),
  do(15) * c(sex = "Female", sport="Boxing"),
  do(50) * c(sex = "Female", sport="Cycling"),
  do(10) * c(sex = "Male", sport="Archery"),
  do(30) * c(sex = "Male", sport="Boxing"),
  do(60) * c(sex = "Male", sport="Cycling")
)
head(sexAndSportRawData)
```

```
     sex   sport
1 Female Archery
2 Female Archery
3 Female Archery
4 Female Archery
5 Female Archery
6 Female Archery
```

```r
tally(sex ~ sport, data=sexAndSportRawData)
```

```
        sport
sex      Archery Boxing Cycling
  Female      35     15      50
  Male        10     30      60
```