Grouped Count Data (Chi-Squared Tests)

ANTH 3720-001 Archaeological and Forensic Science Lab Methods: Data Analysis with R

Elic Weitzel

Jan. 29-31, 2021

1 Grouped Count Data

In some particular cases, you may end up with data in the form of counts for several groups For example, the number of instances of skeletal fractures at several sites, or between multiple time periods. Or the number of lithic debitage types recovered from a series of grid squares, or from different strata within a site. All of these are instances of count data divided up into groups.

2 Chi-Squared Tests

The most common way to analyze such data is using a chi-squared test. A chi-squared test evaluates whether the observed counts in each group are significantly different than expected. It does this by doing some math based on your data and calculating what the expected value for each group should be. It then assesses the difference between that expectation and the observed count.

Chi-squared tests can only be run on count data! Your data must be integers, not decimals and not proportions.

It is also important that every value in your analysis is greater than or equal to 5, but if your data don't quite meet this assumption there is a workaround.

2.1 Load Data

To run a chi-squared test, let's load the "ESASites" data from the `archdata` package and inspect it.

```
library(archdata)
```

```
## Warning: package 'archdata' was built under R version 4.1.2
```

```
data("ESASites")

head(ESASites)
```

```
##   TA BA TOA AA M FK BK NK CFS BS DS Bu Ax Ch SAx Pf
## 1  2  0   0  0 1  0  0  0  12  0  0  4  0  2   0  0
## 2  0  0   0  0 0  1  0  0   2  0  8  0  0  0   0  0
## 3  1  0   0  0 0  0  0  0   3  0  0  0  0  0   0  0
## 4  0  0   0  0 0  0  0  0   3  0  0  1  0  1   0  0
## 5  0  0   0  0 0  0  0  0   2  0  0  3  0  0   0  0
## 6  0  0   2  1 0  0  1  0   0  0  1  1  0  0   0  0
```

```
ncol(ESASites) #16 columns
```

```
## [1] 16
```

```
nrow(ESASites) #43 rows
```

```
## [1] 43
```

We can see that the ESASites data frame contains counts of 16 different artifact types from 43 different Early Stone Age assemblages.

Let's say that we're interested in comparing the counts of Tanged Arrows and Blade Arrows from these Early Stone Age assemblages and some other assemblages from the Late Stone Age.

So first, let's calculate the total count for each artifact type across all 43 assemblages. There are a few different ways you could do this, but a simple one is to use the `apply()` function. This function is very useful, and is an important one to know in R programming. Essentially, it takes another function and applies it across all the rows or columns of your data frame. Here, let's use it to apply the `sum()` function to all columns of the `ESASites` data frame. Doing so will give us the total count for each artifact type.

```
esasums <- apply(ESASites, 2, sum)
```

If you run the `apply` function above, you'll see that the resulting `esasums` object we created is a labeled vector of 16 values - one for each of the 16 artifact types. This `apply` function applied the `sum` function to the columns (denoted here as `2`, whereas `1` would refer to the rows of the data frame) of `ESASites`. Using the `apply` function, and related functions, can save you a lot of time in manipulating your data!

Now let's inspect the data we're interested in: Tanged Arrows and Blade Arrows. These were the first two columns of the `ESASites` data frame, but now that we've collapsed that into a single vector using the `apply` function, we're no longer dealing with a data frame. So we can index this object using a single value in brackets, whereas with a data frame, we would need two values - one for the rows and one for the columns.

```
esasums[1] #tanged arrows
```

```
##  TA
## 103
```

```
esasums[2] #blade arrows
```

```
## BA
## 15
```

```
esasums[1:2] #both tanged and blade arrows
```

```
##  TA  BA
## 103  15
```

Running this code shows us the specific values within the `esasums` object that correspond to the data we want - the first two values, corresponding to tanged and blade arrows.

Now let's get some data to which we can compare ours. We're interested in comparing the frequency of tanged and blade arrows from Early Stone Age sites to Late Stone Age sites, so we need some Late Stone Age data.

We search the literature and find counts of these 16 artifact types from Late Stone Age assemblages. Now let's input them here and make a new object for them called `lsasums`.

```
lsasums <- c(79, 23, 89, 12, 42, 167, 57, 3, 190, 36, 52, 127, 17, 12, 1, 9)
```

However, this vector isn't nicely labeled like our `esasums` object. Since we input the artifact counts in the same order as the `esasums`, we can simply steal the names from this `esasums` vector and apply them to our new `lsasums` vector.

The `names()` function is what we'll use for this. If we run `names(esasums)`, we can see the 16 artifact type labels listed. So let's write a bit of code that will take the names of `esasums` and assign them to the names of `lsasums`.

```
names(esasums) #returns the 16 artifact type names
```

```
##  [1] "TA"  "BA"  "TOA" "AA"  "M"   "FK"  "BK"  "NK"  "CFS" "BS"  "DS"  "Bu"
## [13] "Ax"  "Ch"  "SAx" "Pf"
```

```
names(lsasums) #returns NULL because this vector is unlabeled
```

```
## NULL
```

```
names(lsasums) <- names(esasums) #assigns the names of esasums to lsasums
```

Now if we inspect the `lsasums` object, we'll see that it's a named vector just like `esasums`! This isn't necessary, but it helps us keep track of our data better once we start pulling out specific values.

Now, we said we're interested in comparing the frequencies of tanged and blade arrows between the Early and Late Stone Ages. So let's first create an object that contains both of these values for both time periods.

We can easily do this using indexing and the `rbind()` function. If we index the `esasums` and `lsasums` objects like above, we can pull out the first two values - tanged arrows and blade arrows. Then, we can wrap both of these indexed objects in the `rbind()` function, which binds these values together by rows (hence the *r* in `rbind`). This will create a new object that's basically a 2x2 table of our data.

```
taba <- rbind(esasums[1:2], lsasums[1:2])
```

Now our new `taba` object contains two rows and two columns. The first column is tanged arrows and the second is blade arrows. The first row is from `esasums`, based on the order we specified in the `rbind()` function, while the second row is from `lsasums`. If we wanted to, we could label the rows of this object too using the `rownames()` function.

```
rownames(taba) <- c("ESA", "LSA")
```

Now we have a beautifully labeled 2x2 table of our data!

Let's run a chi-squared test on it!

2.2 Chi-Squared Test of Independence

Because we're comparing multiple groups of counts to other groups of counts, we want a specific type of chi-squared test called a *chi-squared test of independence*. This is the most common variety of chi-squared test in most fields, archaeology and anthropology included.

As I said above, a chi-squared test will calculate expected counts of each artifact type. It does this based on the counts in each cell of our table, as well as the total counts for each row/column and the overall table. You don't need to worry about the details of this math, but it's simple enough and you can look it up if you're curious. The test will then compare our observed counts to these calculated expected counts and tell us if we have significant differences or not.

So let's use the `chisq.test()` function to run a chi-squared test of independence.

```
chisq.test(taba)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  taba
## X-squared = 3.0486, df = 1, p-value = 0.08081
```

Here we get a small output that tells us that the difference between the ESA and LSA counts for these two artifact types is marginally significant (p = 0.081). This p-value is not technically significant at the 0.05 level, but 0.08 is pretty close to 0.05. When you think about the definition of a p-value as a probability, is there really a big difference between 0.05 and 0.08? Not really. This is a good example of why treating the 0.05 alpha value as a hard cutoff point can be a bit silly... Even though our p-value is above 0.05, the difference between the ESA and LSA counts that we're seeing here is clearly unlikely if our null hypothesis of no difference were true.

In this case, I would report that there is a marginally significant difference between ESA and LSA arrow types (X2 = 3.049, df = 1, p = 0.08).

2.2.1 Chi-Squared Test Details

However, we often want a bit more information from this chi-squared test. If we assign the output of our `chisq.test` function to a new object, we can get some more detail.

```
taba.test <- chisq.test(taba)

taba.test
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  taba
## X-squared = 3.0486, df = 1, p-value = 0.08081
```

Here, we can see the same result that we got above if we run the object that contains our chi-squared test result.

We can also now easily extract the observed counts that we plugged in with our `taba` object, and also the expected counts for each cell in the table that our chi-squared test calculated.

```
taba.test$observed
```

```
##      TA BA
## ESA 103 15
## LSA  79 23
```

```
taba.test$expected
```

```
##           TA       BA
## ESA 97.61818 20.38182
## LSA 84.38182 17.61818
```

You can see that, based on the overall counts in our data, our test expected only 98 tanged arrows from the Early Stone Age. Our actual data contain 103 tanged arrows, but the test also determined that this isn't quite a big enough difference to be truly significant at an alpha of 0.05. It's very close though, so we can still most likely say this is a meaningful difference. It very well could be *practically* significant even if it's not *statistically* significant.

This example illustrates how many archaeologists and forensic anthropologist commonly use chi-squared tests: on a 2x2 table. But we can actually run such a test on any size table we want. So let's do that.

We have data on all 16 artifact types for both the Early and Late Stone Ages, so let's compare them all using a chi-squared test of independence.

```
sa.test <- chisq.test(rbind(esasums, lsasums))
```

```
## Warning in chisq.test(rbind(esasums, lsasums)): Chi-squared approximation may be
## incorrect
```

```
sa.test
```

```
##
##  Pearson's Chi-squared test
##
## data:  rbind(esasums, lsasums)
## X-squared = 38.688, df = 15, p-value = 0.0007133
```

We can see that our result is very significant: $p = 0.0007$. We would report this as a strongly significant difference between ESA and LSA artifact counts ($X2 = 38.688$, $df = 15$, $p < 0.0001$).

But note that we got a warning message when we ran this function, telling us that our "Chi-squared approximation may be incorrect."

There are a few reasons this can happen, but it's most often because we have small values in some of our cells. Remember from above that one key rule of chi-squared tests is that you can't have a count of less than 5 in any cell. If we look at our `esasums` and `lsasums` objects, we can see that indeed we do have some small counts that are less than 5, specifically for the NK and SAx artifact types.

The solution to this is to make use of a particular argument in the `chisq.test` function called `simulate.p.value`.

```
sa.test <- chisq.test(rbind(esasums, lsasums), simulate.p.value = TRUE)

sa.test
```

```
##
##  Pearson's Chi-squared test with simulated p-value (based on 2000
##  replicates)
##
## data:  rbind(esasums, lsasums)
## X-squared = 38.688, df = NA, p-value = 0.0004998
```

If we set `simulate.p.value` to `TRUE`, we can compensate for the small counts in some of those cells by using a simulation approach. In this case, R will use what's called a Monte Carlo simulation method with 2000 replicates to calculate a p-value. For our purposes, don't worry about what this really means mathematically, but suffice it to say that this is a way to get around the issue of counts being less than 5. However, due to the nature of this approach, degrees of freedom can no longer be calculated so you would simply state that you used a simulation approach in the chisq.test function to calculate your p-value and report the X2 statistic and p-value.

Now, let's pull out the observed and expected values to see the differences.

```
sa.test$observed
```

```
##          TA BA TOA AA  M  FK BK NK CFS BS DS  Bu Ax Ch SAx Pf
## esasums 103 15  67 15 47 101 43  2 246 29 53 136 19 11   2 18
## lsasums  79 23  89 12 42 167 57  3 190 36 52 127 17 12   1  9
```

```
sa.test$expected
```

```
##               TA      BA      TOA       AA        M       FK       BK       NK
## esasums 90.55074 18.9062 77.61492 13.43335 44.28031 133.3385 49.75315 2.487658
## lsasums 91.44926 19.0938 78.38508 13.56665 44.71969 134.6615 50.24685 2.512342
##              CFS       BS       DS       Bu       Ax       Ch      SAx       Pf
## esasums 216.9238 32.33955 52.24081 130.8508 17.91114 11.44323 1.492595 13.43335
## lsasums 219.0762 32.66045 52.75919 132.1492 18.08886 11.55677 1.507405 13.56665
```

We can see here that some of the differences between the observed and expected counts are quite large while some are quite small. It is therefore likely that not every artifact frequency is significantly different between these two time periods - only some.

To better inspect which artifact types are driving this significant result, we can use what are called standardized residuals for each cell. Our `chisq.test()` function automatically calculates these for us, and we can pull them out by appending `$stdres` to our chi-squared test object.

```
sa.test$stdres
```

```
##               TA        BA       TOA         AA          M        FK        BK
## esasums  1.94528 -1.280773 -1.777523  0.6075263  0.5911925 -4.277749 -1.389289
## lsasums -1.94528  1.280773  1.777523 -0.6075263 -0.5911925  4.277749  1.389289
##                NK       CFS        BS         DS         Bu        Ax
## esasums -0.436779  3.192901 -0.843627  0.1526412  0.6864811  0.366597
## lsasums  0.436779 -3.192901  0.843627 -0.1526412 -0.6864811 -0.366597
```

```
##                  Ch        SAx        Pf
## esasums -0.1860172  0.5863911  1.770888
## lsasums  0.1860172 -0.5863911 -1.770888
```

Now we see a series of numbers that are not counts or expected counts, but expressions of how different our data are from 0. If a number is close to 0, that corresponds to a higher p-value: a greater probability that the difference is just random chance, and not "real." The further a standardized residual is away from 0, the lower the p-value. These standardized residuals are on the scale of z-scores on a distribution. A z-score of 0 is the mean of the distribution, and the score goes up the further you move from the mean in a positive direction and goes down the further you move in a negative direction.

Most importantly for us is the z-score that corresponds to the 95% confidence interval: the threshold for a significant result at alpha = 0.05.

That z-score is 1.96 or -1.96. So any standardized residual here that is greater than 1.96 or less than -1.96 (further from 0 than either value) is significant at the 0.05 level.

Inpecting our standardized residuals, we see that not all of the values are significantly different from each other. Only the values for FK and CFS are significant. The rest have standardized residuals that are closer to 0 than 1.96/-1.96.

The sign (+/-) in front of each standardized residual also tells us the direction of the change. For flake knives (FK), the Early Stone Age count has a standardized residual of -4.278 while the Late Stone Age is 4.278. This means that the count of flake knives is significantly lower than expected in the Early Stone Age and significantly higher than expected in the Late Stone Age. The standardized residual for tanged arrows (TA) is 1.945 in the Early and -1.945 in the Late Stone Age. This means that there are more blade arrows than expected in the Early Stone Age and fewer than expected in the Late Stone Age, but these differences are not quite significant (but they're very close) since the standardized residual is not greater than 1.96.

2.2.2 Reporting Results

When reporting the results of a chi-squared test, I would report not only the test statistic, degrees of freedom (when not using the simulated p-value argument), and p-value ($X2 = 38.688$, $p < 0.01$), but also a table of the observed counts, expected counts, and standardized residuals. You can make such a table as follows using the `data.frame()` function and then exporting this table using the `write.csv()` function. I also make use of the `t()` function which is a handy little function that *transposes* your data (hence "t"). This means that if you have an object that has 2 rows and 10 columns, it will transpose it so that there are 10 rows and 2 columns.

```
chisq.results.table <- data.frame(t(sa.test$observed), t(sa.test$expected), t(sa.test$stdres))
colnames(chisq.results.table) <- c("ESA Obs", "LSA Obs", "ESA Exp", "LSA Exp",
                                   "ESA Std. Res.", "LSA Std. Res.")

write.csv(chisq.results.table, "ChiSq_Test_Results_Table.csv")
```

Now there should be a .csv file containing this information in your working directory that you can further manipulate.

2.3 Goodness of Fit/One-Sample Chi-Squared Tests

Now let's say we want to know whether our tanged and blade arrow counts from Early Stone Age Norway are the same as those for Early Stone Age sites in Sweden. But the archaeologists in Sweden only reported proportions for their assemblages, not actual count data! What to do?

No worries, because there's a specific type of chi-squared test that will still work when comparing counts to proportions: a *goodness of fit or one-sample chi-squared test*. This second type of chi-squared test is also good to know about in case you should ever need to use it. The Goodness of Fit or One-Sample Chi-Squared Test applies when you are comparing your groups of counts to proportions/probabilities instead of other groups of counts.

So let's first input the proportions of tanged arrows and blade arrows that the Swedish archaeologists reported for their Early Stone Age sites.

```
esa.sweden <- c("TA" = 0.57, "BA" = 0.43)
```

Now let's run a chi-squared test on our count data for these two artifact types using these proportions as the comparison. We can do this using the `p =` argument in the `chisq.test()` function, which stands for probability.

```
chisq.test(esasums[1:2],
           p = esa.sweden)
```

```
##
##  Chi-squared test for given probabilities
##
## data:  esasums[1:2]
## X-squared = 44.166, df = 1, p-value = 3.017e-11
```

The result of this chi-squared test is very significant ($X2 = 44.166$, $df = 1$, $p < 0.0001$).

If we assign this test output to an object, we can crack it open to pull out the expected values and standardized residuals, as above.

```
ns.test <- chisq.test(c(esasums[1], esasums[2]),
           p = esa.sweden)

ns.test$observed
```

```
##  TA  BA
## 103  15
```

```
ns.test$expected
```

```
##    TA    BA
## 67.26 50.74
```

```
ns.test$stdres
```

```
##        TA        BA
##  6.645718 -6.645718
```

Now we can see that our chi-squared test used the Swedish probabilities that we fed it to calculate expected artifact counts. For TA this count was 67.26 and for BA it was 50.74. Our test then compared our observed counts of 103 and 15 to these expected counts.

Furthermore, if our p-value for the test wasn't enough, our standardized residuals reveal that these observed counts are very different from expected. A standardized residual of 6.65 is far beyond the 1.96 threshold of the 95% confidence interval around 0. Standardized residuals matter a bit more when you're dealing with a more complex table in which any of a number of rows and columns could be driving a significant result.

We can also compare our full 16 type Norwegian assemblage to the corresponding probabilities from Sweden. Let's create an object that contains the probabilities for the artifact types from Early Stone Age Sweden, and then name these values as we did previously

```
esa.sweden.full <- c(0.19, 0.02, 0.07, 0.03, 0.06, 0.07, 0.05, 0.002, 0.18, 0.01,
                     0.03, 0.24, 0.02, 0.01, 0.01, 0.008)
names(esa.sweden.full) <- names(esasums)

esa.sweden.full
```

```
##    TA    BA   TOA    AA     M    FK    BK    NK   CFS    BS    DS    Bu    Ax
## 0.190 0.020 0.070 0.030 0.060 0.070 0.050 0.002 0.180 0.010 0.030 0.240 0.020
##    Ch   SAx    Pf
## 0.010 0.010 0.008
```

We can now run a one-sample chi-squared test using this vector of probabilities.

```
ns.test.full <- chisq.test(esasums, p = esa.sweden.full)
```

```
## Warning in chisq.test(esasums, p = esa.sweden.full): Chi-squared approximation
## may be incorrect
```

Note that we again got a warning message when we ran this function telling us that our "Chi-squared approximation may be incorrect." This is due to the same issue as above, and can be solved in the same way by setting the `simulate.p.value` argument to `TRUE`.

```
ns.test.full <- chisq.test(esasums, p = esa.sweden.full, simulate.p.value = T)

ns.test.full
```

```
##
##  Chi-squared test for given probabilities with simulated p-value (based
##  on 2000 replicates)
##
## data:  esasums
## X-squared = 220.11, df = NA, p-value = 0.0004998
```

```
ns.test.full$observed
```

```
##  TA  BA TOA  AA   M  FK  BK  NK CFS  BS  DS  Bu  Ax  Ch SAx  Pf
## 103  15  67  15  47 101  43   2 246  29  53 136  19  11   2  18
```

```
ns.test.full$expected
```

```
##      TA      BA     TOA      AA       M      FK      BK      NK     CFS      BS
## 172.330  18.140  63.490  27.210  54.420  63.490  45.350   1.814 163.260   9.070
##      DS      Bu      Ax      Ch     SAx      Pf
##  27.210 217.680  18.140   9.070   9.070   7.256
```

```
ns.test.full$stdres
```

```
##         TA         BA        TOA         AA          M         FK         BK
## -5.8681100 -0.7447285  0.4567862 -2.3766516 -1.0374346  4.8814959 -0.3580283
##         NK        CFS         BS         DS         Bu         Ax         Ch
##  0.1382385  7.1510316  6.6509864  5.0199710 -6.3503793  0.2039702  0.6440744
##        SAx         Pf
## -2.3593815  4.0046220
```

We can see that the artifact counts in Norway and Sweden are significantly different from expected ($X2 = 220.11$, $p < 0.001$). We can also see, based our standardized residuals, that it's the counts for TA, AA, FK, CFS, BS, DS, Bu, SAx, and Pf that are driving this result. Note that the standardized residuals are only provided for our data, not the Swedish data for which we only had probabilities, but that these values are the same, just with the opposite sign, since there are only two columns here.