Chi-Squared Goodness of Fit Tests

Stat 145

Simple Goodness of Fit Tests

Simple goodness of fit tests are useful when we have a categorical variable with 2 or more possible values (Cat3+) and our null hypothesis tells us the (population) probabilities for each category.

Example Situations

Golf balls in the Yard Golf balls typically have a brand name and a number printed on them. Most of these numbers are 1, 2, 3, or 4. But are those 4 numbers equally likely?

Null hypothesis: $p_1 = 0.25$, $p_2 = 0.25$, $p_3 = 0.25$, $p_4 = 0.25$. (Each number is equally likely.)

Genetics Many genetic models predict specific proportions for genotypes or phenotypes. These are usually expressed as ratios like 3:1 or 1:2:1. These can be converted to proportions by dividing each number by the sum. For example, 1:2:1 turns into the probabilities 1/4, 2/4, and 1/4, and the null hypothesis is

Null hypothesis: $p_1 = 0.25$, $p_2 = 0.50$, $p_3 = 0.25$.

Alternative Hypothesis

The alternative hypothesis for these tests is always the same: not all of the proportions are what the null hypothesis says they are. Said another way: at least one[1] of the proportions is different from what the null hypothesis says. This means that there can be many ways for the null hypothesis to be false.

Summarizing the data

The data we collect for a goodness of fit test consists of one categorical variable, typically with 3 or more categories. These data are often summarized in a table with the counts the number of occurrences for each category.

Golf balls example **Data summary:** The numbers on 486 golf balls collected in a yard are tallied below

number on golf ball	1	2	3	4
observed count	137	138	107	104

Each observation in our data must be counted in **exactly one** cell of the table. In this case, each golf ball is counted in exactly one of the four cells of our table because each golf ball in the sample has exactly one of these four numbers.[2]

[1] Note: since the probabilities have to add up to 1, if one probability is different, at least two must be different.

[2] Truth in advertizing: The sample actually had 500 golf balls, but 14 of them either didn't have a number at all (usually a company logo or something instead) or had a number other than 1–4. We are only interested in the 486 that have a number 1–4.

Testing these hypotheses

To test these hypotheses we need

1. A test statistic

   This should be a number that looks one way when the null hypothesis is true and a different way when the null hypothesis is false.

2. The null distribution of the test statistic, so we can convert the test statistic into a p-value.

   a. **Randomization:** We could use simulations to create a randomization distribution. All we need is a way to simulate the situation when the null hypothesis is true. Then we compute our test statistic on each randomization sample to get our randomization distribution.

   b. **Theory:** If there is statistical theory that tells us the (approximate) shape of the null distribution, we could use that instead. (Spoiler alert: this one won't be normal.)

The Chi-Squared Statistic

While you might come up with a variety of possible test statistics to use for a simple goodness of fit test, the most commonly used test statistic is the chi-squared statistic:

$$X^2 = \sum \frac{(\text{observed count} - \text{expected count})^2}{\text{expected count}}$$

- **observed count:** the count observed in one cell of the table
- **expected count:** the count we would expect in a cell if the null hypothesis were true

Golf balls example Let's illustrate with our golf balls data set.

Expected Counts: Our null hypothesis says that we should have 25% with each number. 25% of 486 is 121.5, so our expected counts are 121.5 for each cell. (Note: expected counts do not have to be integers.)

number on golf ball	1	2	3	4
observed count	137	138	107	104
expected count	121.5	121.5	121.5	121.5

Test Statistic: We can use the observed and expected counts to compute the Chi-squared statistic.

number on golf ball	1	2	3	4
observed count (o)	137	138	107	104
expected count (e)	121.5	121.5	121.5	121.5
difference ($o - e$)	15.5	16.5	-14.5	-17.5
$(o-e)^2$	240.25	272.25	210.25	306.25
$(o-e)^2/e$	1.98	2.24	1.73	2.52

$X^2 = 1.98 + 2.24 + 1.73 + 2.52 = 8.47$

Here are two ways to do the computation in R:

```
(137 - 121.5)^2 / 121.5 + (138 - 121.5)^2 / 121.5 +
(107 - 121.5)^2 / 121.5 + (104 - 121.5)^2 / 121.5
```

```
## [1] 8.469136
```

```
# "fancy" way
o <- c(137, 138, 107, 104)
e <- c(121.5, 121.5, 121.5, 121.5)
sum( (o - e)^2 / e)
```

```
## [1] 8.469136
```

Randomization for Goodness of Fit

To create a randomization distribution for our test statistic, we need a way to simulate data when the null hypothesis is true. We can do this with `resample()` by taking advantage of some features of this function we haven't seen before. If we set the `size`, we can resample as many times as we like. Here are 20 simulated golf balls (twice):

```
resample( 1:4, size = 20)
```

```
##  [1] 2 3 4 2 4 4 2 1 3 4 4 2 2 3 2 3 2 2 1 4
```

```
resample( 1:4, size = 20)
```

```
##  [1] 4 3 2 2 1 4 4 4 2 4 3 1 2 1 4 4 2 3 4 4
```

It's easier to look at a table, especially because we need 486, not 20.

```
set.seed(12345)
tally( ~ resample( 1:4, size = 486))
```

```
## resample(1:4, size = 486)
##   1   2   3   4
## 112 123 134 117
```

```
tally( ~ resample( 1:4, size = 486))
```

```
## resample(1:4, size = 486)
##   1   2   3   4
## 133 100 127 126
```

Calculating chi-squared was tedious. Let's use `chisq()` to do that for us.

```
set.seed(1234)
chisq(tally( ~ resample( 1:4, size = 486)))
```

```
## X.squared
##  1.292181
```

```
chisq(tally( ~ resample( 1:4, size = 486)))
```

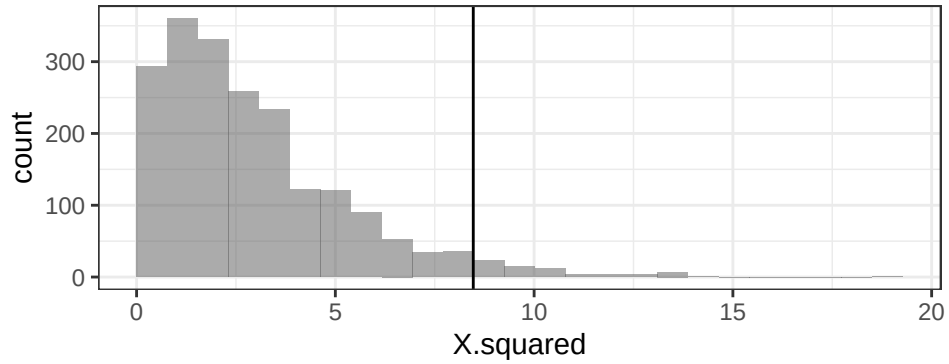
```
## X.squared
##  8.781893
```

That's all we need to generate a randomization distribution with `do()`:

```
Golf_rand <- do(2000) * chisq(tally( ~ resample( 1:4, size = 486)))
head(Golf_rand, 3)
```

```
##   X.squared
## 1 0.3539095
## 2 1.7201646
## 3 0.4691358
```

```
gf_histogram( ~ X.squared, data = Golf_rand, boundary = 0) %>%
  gf_vline(xintercept = ~ 8.469)
```



Wow! That certainly doesn't look like a normal distribution, but it is correct. This test statistic gives a skewed distribution and can't go below 0 (since all the terms in the sum are non-negative).

One-tailed p-value calculation

Although there are many ways for the null hypothesis to be false, they all tend to make our test statistic larger (because the observed counts will be farther from the expected counts, at least in some cells). For this reason, our p-value is computed by determining the proportion of our randomization distribution that is at least as large as our observed test statistic.

```
tally( ~(X.squared >= 8.469), data = Golf_rand)
```

```
## (X.squared >= 8.469)
##  TRUE FALSE
##    71  1929
```

```
prop( ~(X.squared >= 8.469), data = Golf_rand)
```

```
## prop_TRUE
##    0.0355
```

So our p-value is just a bit below 0.05. That provides some evidence that the numbers on golf balls are not uniformly distributed, but it isn't overwhelming evidence.

Goodness of Fit Practice

Skill Building

1. Alice wants to test whether a standard 6-sided die is fair.
   a. Express her null hypothesis using proportions.
   b. Compute the expected counts if she rolls the die 100 times.
2. A genetic model predicts a 12:3:1 ratio of 3 phenotypes. Suppose we want to test this hypothesis using a sample of size 200.
   a. Express the null hypothesis using proportions.
   b. Compute the expected counts for each phenotype.
3. A genetic model predicts a 9:3:3:1 ratio of 4 phenotypes. Suppose we want to test this hypothesis with a sample of size 125.
   a. Express the null hypothesis using proportions.
   b. Compute the expected counts for each phenotype.
   c. Use `resample()` to create (and save) one randomization sample of size 125 assuming this ratio holds. You will need to tell R the sample size *and the proportions* for resampling:

```
tally(resample(1:4, size = 125, p = c(9/16, 3/16, 3/16, 1/16)))
```

```
## X
##  1  2  3  4
## 70 24 23  8
```

   Here's another way to write this:

```
resample(1:4, size = 125, p = c(9/16, 3/16, 3/16, 1/16)) %>%
  tally()
```

```
## X
##  1  2  3  4
## 61 36 22  6
```

   Both of them say "Resample, then tally."
   d. Repeat part b for a second randomization sample. How do your two tables compare?
   e. Calculate the chi-squared test statisitic "by hand" for one of your two randomization samples. (You can use R to do the arithmetic.)
   f. Now use `chisq()` to check your work. Note: `chisq()` also needs to know what probabilities to use. You can do this one of two ways:

```
chisq(
  tally(
    resample(1:4, size = 125, p = c(9/16, 3/16, 3/16, 1/16)),
  ),
  p = c(9/16, 3/16, 3/16, 1/16)
)
```

```
## X.squared
##  17.17956
```

   You can make the code a little easier to read if you do it like this:

```
resample(1:4, size = 125, p = c(9/16, 3/16, 3/16, 1/16)) %>%
  tally() %>%
  chisq(p = c(9/16, 3/16, 3/16, 1/16))
```

```
## X.squared
##  5.773333
```

This way makes it clear that you first resample, then tally the results, then compute the chi-squared statistic.

g. Now create 2000 chi-squared statistics from 2000 randomization samples and create a histogram of these statistics.

4. Create a randomization distribution for the situation in problems 1 and 2. Use histograms to compare the 3 randomization distributions. What do you notice?

Two Gotchas

There are two things that can trip you up when performing a randomization test this way.

1. If the sample size is small, you migth not get any observations in one of the cells.

   This effectively the size of the table, and then funny things start to happen. You can fix this by using `factor()`. `factor()` is R's primary way of handling categorical data. In addition to recording the values, it keeps track of the *possible* values, even if they don't occur.

   ```
   factor(1:4)
   ```

   ```
   ## [1] 1 2 3 4
   ## Levels: 1 2 3 4
   ```

   ```
   resample(factor(1:4), size = 3)
   ```

   ```
   ## [1] 3 4 1
   ## Levels: 1 2 3 4
   ```

   This allows our table to have 0's in it.

   ```
   # sure to be one missing if we only sample 3 times from 4 categories
   tally(resample(1:4, size = 3))
   ```

   ```
   ## X
   ## 1 3 4
   ## 1 1 1
   ```

   ```
   tally(resample(factor(1:4), size = 3))
   ```

   ```
   ## X
   ## 1 2 3 4
   ## 1 0 1 1
   ```

2. The `chisq()` function is expecting a table.

   When we use `tally()`, everything is fine. But if you are entering your own observed values, you need to make sure they are in a table.

   ```
   observed1 <- c(25, 22, 29, 18); observed1
   ```

   ```
   ## [1] 25 22 29 18
   ```

   ```
   observed2 <- c(25, 22, 29, 18) %>% as.table(); observed2
   ```

```
##  A  B  C  D
## 25 22 29 18
```

```
chisq(observed1) # INCORRECT! -- being treated like raw data.
```

```
## X.squared
##         0
```

```
chisq(observed2) # CORRECT -- calculating from a table of counts
```

```
## X.squared
##  2.765957
```

This actually means we can use a short-cut to compute our randomization distribution:

```
# with tally()
set.seed(1234)
resample(1:4, size = 125, p = c(9/16, 3/16, 3/16, 1/16)) %>%
  tally() %>%
  chisq(p = c(9/16, 3/16, 3/16, 1/16))
```

```
## X.squared
##  4.976889
```

```
# without tally()
set.seed(1234)
resample(1:4, size = 125, p = c(9/16, 3/16, 3/16, 1/16)) %>%
  chisq(p = c(9/16, 3/16, 3/16, 1/16))
```

```
## X.squared
##  4.976889
```

The good news is that soon we will learn how to get R to automate this whole process for us (whew!).

Additional Examples

In each of the examples below, create a randomization distribution for the chi-squared statistic and use it to compute a p-value.

5. Plant Genetics. According to a genetic model, the ratio of colors of seed coats should be 12:3:1. The observed colors are given in the table below. Is this data compatible with this particular genetic theory?

Seed coat color	black	brown	pale
Count	321	77	31

6. More Plant Genetics. Below are some data from one of Mendel's original studies of pea plants.

Phenotype	yellow, round	yellow, wrinkled	green, round	green, wrinkled
Count	315	101	108	32

According to his theory of independent assortment (Mendel's second law), the phenotype ratios should be 9:3:3:1. Is this data compatible with this particular genetic theory?

7. Birth Days. A sample of 700 random births revealed the following data regarding the day of the week on which the birth occurred. Is there evidence that some days are more common than others or might the differences in our table simply be due to sampling variability?

Day	Sun	Mon	Tue	Wed	Thr	Fri	Sat
Births	84	110	124	104	94	112	72

8. One proportion or Chi-squared? Fifty subjects were given two unmarked cups of coffee and asked which one they preferred. 31 preferred the fresh brewed coffee and 19 preferred the instant coffee.

Is this evidence that a majority of people prefer fresh brewed coffee? Conduct the hypothesis test two ways.

- Using a test for one proportion
- Using a chi-squared goodness of fit test

How do the results compare?

9. Frizzled Feathers The Frizzle fowl is a variety of chicken with curled feathers. In a 1930 experiment, Launder and Dunn crossed Frizzle fowls with with the Leghorn variety which has straight feathers. The first generation (F1) produced all slightly frizzled chicks. This made the researchers suspect a co-dominant genetic model. To test this, they interbred F1 to get F2 chicks and recorded the feather type for each chick.

frizzled	slightly frizzled	straight
23	50	20

The codominant model predicts a 1:2:1 ratio. Why?

Use these data to assess the genetic model.