

χ^2 TESTS AND FISHER'S EXACT TESTS IN R

R is an open source statistical package with versions available for Linux, Mac OS X, and Windows. In addition to being available without charge, it is *very* powerful and very flexible. It's been my statistical package of choice for the past 4-5 years. Running a χ^2 test or Fisher's exact test, as you need to do for the first question in Problem #1, is very straightforward—once you get the data in. In the example that follows, we'll imagine that you've determined the number A_1 and A_2 alleles contributed by fathers to offspring of each maternal genotype as follows:

Maternal genotype	Paternal gamete	
	A_1	A_2
A_1A_1	12	17
A_1A_2	4	25
A_2A_2	15	4

Let me show you the code first. Then I'll explain it.

```
> alleles <- matrix(c(12, 4, 15, 17, 25, 4), nr=3,
+ dimnames=list(c("A1A1", "A1A2", "A2A2"), c("A1", "A2")))
> alleles
      A1 A2
A1A1 12 17
A1A2  4 25
A2A2 15  4
> chisq.test(alleles)
```

Pearson's Chi-squared test

```
data: alleles
X-squared = 20.2851, df = 2, p-value = 3.937e-05
```

```
> fisher.test(alleles)
```

Fisher's Exact Test for Count Data

```
data: alleles
p-value = 2.957e-05
alternative hypothesis: two.sided
```

The first thing to know about R is that it's command-line oriented. If you've ever used Linux, the DOS box in Windows, or the terminal in Mac OS X, you'll be familiar with the idea of a "prompt". That's the `>` that you see at the start of some lines. It indicates that R is waiting for you to type something. Sometimes you'll have a long command to type, as I do on the first line. IF you havent finished your command yet when you hit return, you'll get a different prompt, the `+` that you see on the second line. This reminds you that you're in the middle of typing a command. So what does that first line do?

- `alleles` — This is the name of an object that the first command creates in which to store the data. I do this so that I can easily refer to the data later. You can give the object pretty much any name you want, as long as it doesn't start with a number. You could call it `Fred`, if you wanted to, or `x`, if you don't want to type so much.
- `<-` — This is the "operator" that assigns the result of the command we put on the right side to the object, `alleles`, on the left side.
- `matrix` — This tells R that we're going to construct a matrix with our data. The stuff in between the pair of parentheses that start on this line and end on the next line tell R how to construct the matrix.
- `c(12, 4, 15, 17, 25, 4),` — As you can probably guess this is the data. The `c()` constructs a column of data (each element separated by a comma. Notice that we list the data by going down the first column to the bottom and beginning again at the top. The comma at the end of this parenthesis lets R know that the next "argument" is coming.
- `nr=3,` — This argument tells R that there are three rows in our data. This means that the column of data you just created has to have a multiple of three elements in it. In our case we have six, so we're fine. The comma tells us that the next argument is coming.¹ Notice that I hit the return key after the comma so we now go to the next line, which has a prompt of `+` because the command isn't finished. Instead of

¹You can have spaces around the `=` if you'd like. It's up to you.

the comma, you could put a parenthesis here,). When you hit return, you'd get a > instead of a +, indicating that the command was done.

- This line is optional, but I rather like it. It allows me to check to make sure I've entered the data the way that I thought I did. `dimnames` means that I'm going to give names to the rows and columns. There's a `list` with two elements, a column of maternal genotypes, `c("A1A1", "A1A2", "A2A2")`, and a column of paternal alleles, `c("A1", "A2")`. There are three parentheses at the end of the line to balance the three that preceded it.² When you hit return you get the primary prompt, >.
- Now I just type `alleles` at the prompt and I see the table of data displayed before me. If you'd left off the `dimnames`, i.e., if you'd entered only

```
> alleles <- matrix(c(12, 4, 15, 17, 25, 4), nr=3
```

You'd get a table that looks like this

```
> alleles
      [,1] [,2]
[1,]   12   17
[2,]    4   25
[3,]   15    4
```

As you can see, including `dimnames` makes it easier to see that the data you've entered matches what you actually wanted to enter.

- That's the hard part. To run a χ^2 test just type `chisq.test(alleles)` and hit return. You'll get the following result.

Pearson's Chi-squared test

```
data: alleles
X-squared = 20.2851, df = 2, p-value = 3.937e-05
```

You can run a Fisher's exact test by typing `fisher.test(alleles)`.

²If you don't see them, ask me.

Fisher's Exact Test for Count Data

```
data: alleles  
p-value = 2.957e-05  
alternative hypothesis: two.sided
```

In most cases the P -values won't be too different, but if they are different, the exact test is right. Since it's just as easy to run a Fisher's exact test in R as a χ^2 test, I prefer the Fisher's exact test. The χ^2 is based on an approximation and was useful before the advent of powerful desktop computers, but the only reason to use it now is if you don't have a computer handy and need to do the calculations by hand.

In this case the χ^2 test gives us a P value of 3.9×10^{-5} and Fisher's exact test gives us a P value of 3.0×10^{-5} . So regardless of which test we choose we have very strong evidence that there are significant differences in the proportion of A_1 sperm fertilizing eggs of the three different maternal genotypes.

Creative Commons License

These notes are licensed under the Creative Commons Attribution-NonCommercial-ShareAlike License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-sa/2.5/> or send a letter to Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.