Chi-square goodness-of-fit test

Put your name here

Put the date here

Introduction

In this assignment we will learn how to run the chi-square goodness-of-fit test. A chi-square goodness-of-fit test is similar to a test for a single proportion, except instead of two categories (success/failure), we now try to understand the distribution among three or more categories.

It is interesting to note that when there are only two categories, one can still run a chi-square test and the results are more or less equivalent to the one-proportion test.

Instructions

Presumably, you have already created a new project and downloaded this file into it. Please knit the document and work back and forth between this R Markdown file and the PDF output as you work through this module.

When you are finished with the assignment, knit to PDF one last time, proofread the PDF file **carefully**, export the PDF file to your computer, and then submit your assignment.

Sometimes you will be asked to add your own R code. That will appear in this document as a code chunk with a request for you to add your own code, like so:

```
## Add code here to [do some task]...
```

Be sure to remove the line `## Add code here to [do some task]...` when you have added your own code.

Sometimes you will be asked to type up your thoughts. That will appear in the document as follows:

Please write up your answer here.

Again, please be sure to remove the line "Please write up your answer here" when you have written up your answer. In these areas of the assignment, please use contextually meaningful full sentences/paragraphs (unless otherwise indicated) and proper spelling, grammar, punctuation, etc. This is not R code, but rather a free response section where you talk about your analysis and conclusions. If you need to use some R code as well, you can use inline R code inside the block between `\begin{answer}` and `\end{answer}`, or if you need an R code chunk, please go outside the `answer` block and start a new code chunk.

Load Packages

We load the standard `mosaic` package and the `openintro` package for the `hsb2` data.

```r
library(openintro)
library(mosaic)
```

We'll be doing some simulation, so let's set the seed.

```r
set.seed(9090)
```

Research question

We use a classic data set `mtcars` from a 1974 Motor Trend magazine to examine the distribution of the number of engine cylinders (with values 4, 6, or 8). Assuming that this data set is representative of all cars from 1974, were there an equal number of cars with 4, 6, and 8 cylinders?

Chi-squared

When we have three or more categories in a categorical variable, it is natural to ask how the observed counts in each category compare to the counts that we expected to see under the assumption of some null hypothesis. In other words, we're assuming that there is some "default" distribution to which we are going to compare our data. Sometimes, this default null comes from substantive expert knowledge. (For example, we might compare the 1974 distribution to a known distribution from another year.) Sometimes we're interested to see if our data deviates from a null distribution that predicts an equal number of observations in each category, which is the research question in this module.

First of all, what is the actual distribution of cylinders in our data? Let's convert `cyl` to a factor variable first, and then we'll count. (Since the levels are already called "4", "6", and "8", we do not need to specify `levels` or `labels`.)

```r
cyl <- factor(mtcars$cyl)
table(cyl)
```

```
## cyl
##  4  6  8
## 11  7 14
```

The numbers in these three cells are the "observed" values (usually denoted by O). What are the expected counts? Well, since there are 32 cars and there are 3 categories, we would expect there to be 10.6666667 cars at each level. This is the "expected" count (usually denoted by E).

Why isn't the expected count a whole number? In any given data set, of course, we will see a whole number of cars with 4, 6, or 8 cylinders. However, since this is just the expected count, it's the average over lots of possible sets of 32 cars under the assumption of the null. We don't need for this average to be a whole number.

How should the deviation between the data and the null distribution be measured? We could simply look at the difference between the observed counts and the expected counts $O - E$. However, there will be some positive values (cells where we have more than 10.6666667 cars) and some negative values (cells where we have fewer than 10.6666667 cars). These will all cancel out.

If this sounds vaguely familiar, it is because we encountered the same problem with the formula for the standard deviation. The differences $y - \bar{y}$ had the same issue. Do you recall the solution in that case? It was to square these values, making them all positive.

So instead of $O - E$, we will consider $(O - E)^2$. Finally, to make sure that cells with large expected values don't dominate, we divide by E:

$$\frac{(O - E)^2}{E}.$$

This put each cell on equal footing. Now that we have a reasonable measure of the deviation between observed and expected counts for each cell, we define χ^2 ("chi-squared", pronounced "kye-squared"—rhymes with "die-scared", or if that's too dark, how about "pie-shared"[1]) as the sum of all these fractions, one for each cell:

$$\chi^2 = \sum \frac{(O - E)^2}{E}.$$

A χ^2 value of zero would indicate perfect agreement between observed and expected values. As the χ^2 value gets larger and larger, this indicates more and more deviation between observed and expected values.

As an example, for our data, we calculate chi-squared as follows:

$$\chi^2 = \frac{(11 - 10.67)^2}{10.67} + \frac{(7 - 10.67)^2}{10.67} + \frac{(14 - 10.67)^2}{10.67} = 2.3.$$

In general, the expected counts do not have to be the same as they are here. This is a function of the null hypothesis which, for us, is that all cylinders are represented equally in the population.

Or we could just do it in R with the `chisq` command. We will store this value for later.

```
obs_chisq <- chisq(cyl)
obs_chisq
```

```
## X.squared
##    2.3125
```

The chi-square distribution

We know that even if the true distribution were 10.6666667 in each cell, we would not see that exactly if we collect a sample of 32 cars. (In fact, the "true" distribution is physically impossible because 10.6666667 is not a whole number!) So what kinds of numbers could we get?

Let's do a quick simulation to find out. We will use the `sample` command from the `mosaic` package.

First of all, recall the actual distribution of cylinders:

```
table(cyl)
```

```
## cyl
##  4  6  8
## 11  7 14
```

Under the assumption of the null, there should be an equal chance of seeing 4, 6, or 8 cylinders. The `sample` command below takes the `levels` of `cyl` ("4", "6", or "8") and grabs them at random until there are 32 values. We also have to specify `replace = TRUE` to let `sample` know that there can (and should!) be more than one car in any given level.

```
sample(levels(cyl), 32, replace = TRUE)
```

```
##  [1] "8" "8" "8" "6" "6" "8" "8" "6" "4" "8" "6" "4" "8" "4" "4" "6" "6"
## [18] "8" "8" "4" "4" "4" "8" "8" "4" "8" "4" "6" "8" "6" "4" "8"
```

[1] Rhyming is fun!

Let's do this a couple of times, this time organizing the output into a table:

```
table(sample(levels(cyl), 32, replace = TRUE))
```

```
##
##  4  6  8
## 11 14  7
```

```
table(sample(levels(cyl), 32, replace = TRUE))
```

```
##
##  4  6  8
## 11 10 11
```

```
table(sample(levels(cyl), 32, replace = TRUE))
```

```
##
##  4  6  8
## 12 10 10
```

Each table represents 32 randomly sampled cars from a population in which we're assuming that there are an equal number of 4, 6, and 8 cylinder cars.

Next, we need to calculate the χ^2 value for each sample. This is a simple matter of applying the `chisq` function to each random sample.

```
chisq(table(sample(levels(cyl), 32, replace = TRUE)))
```

```
## X.squared
##    0.4375
```

```
chisq(table(sample(levels(cyl), 32, replace = TRUE)))
```

```
## X.squared
##    0.8125
```

```
chisq(table(sample(levels(cyl), 32, replace = TRUE)))
```

```
## X.squared
##    0.0625
```

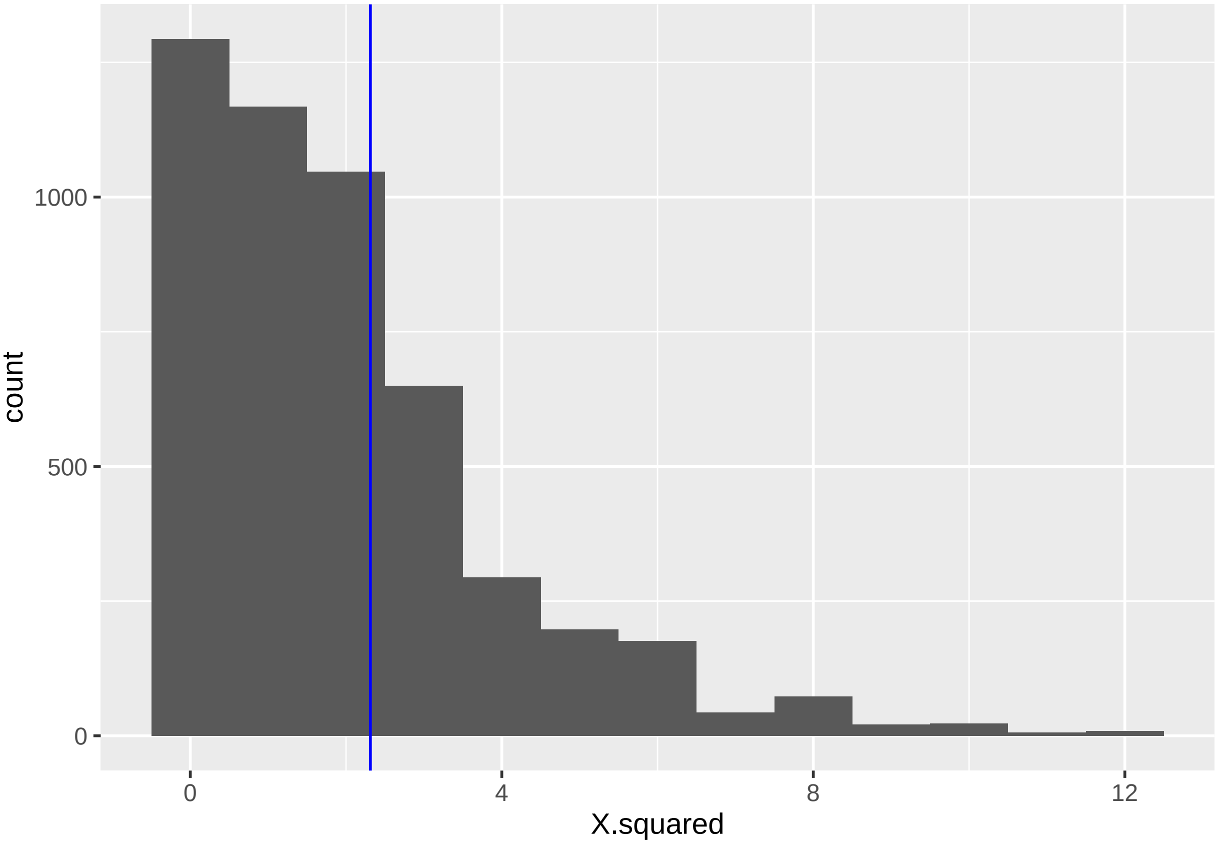
As before, we can use the `do` command to do this a bunch of times and save the results in a data frame.

```
sims <- do(5000) * chisq(table(sample(levels(cyl), 32, replace = TRUE)))
head(sims)
```

```
##   X.squared
## 1    2.3125
## 2    0.4375
## 3    8.3125
## 4    1.9375
## 5    4.0000
## 6   10.9375
```

Let's graph the resulting random values of χ^2 and include the chi-squared value for our actual data.

```
ggplot(sims, aes(x = X.squared)) +
    geom_histogram(binwidth = 1) +
    geom_vline(xintercept = obs_chisq, color = "blue")
```



A few things are apparent:

1. The values are all positive. This makes sense when you remember that each piece of the χ^2 calculation was positive. This is different from our earlier simulations that looked like normal models. (Z-scores can be positive or negative, but not χ^2.)
2. This is a severely right-skewed graph. Although most values are near zero, the occasional freak sample can have a large value of χ^2.
3. You can see that our sample (the blue line) does not seem that unusual. It's not way out into the right tail. Another way of saying this is that our data does not seem out of the ordinary for data sampled under the assumption of the null hypothesis.

Chi-square as a sampling distribution model

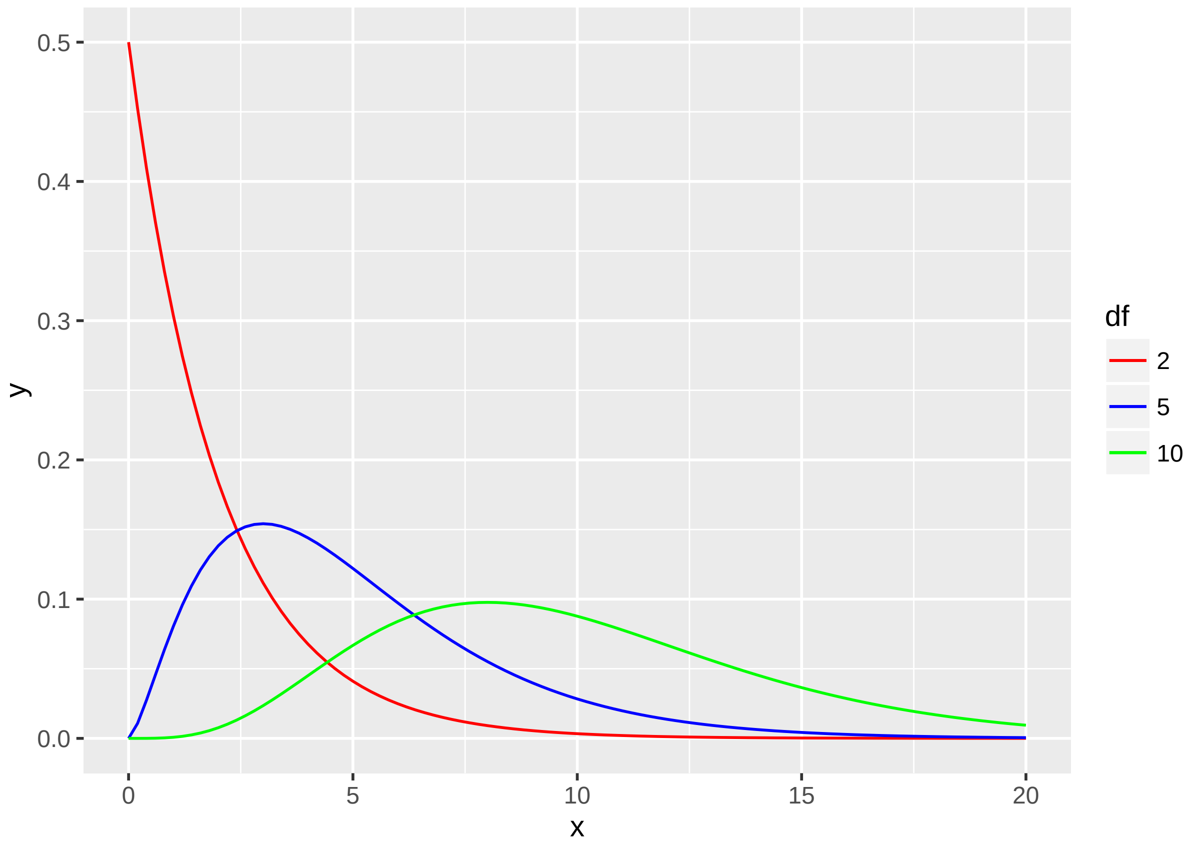
Just like there was a mathematical model for our simulated data before (the normal model back then), there is also a mathematical model for this type of simulated data. It's called (not surprisingly) the *chi-square distribution*.

There is one new idea, though. Although all normal models have the same bell shape, there are many different chi-square models. This is because the number of cells can change the sampling distribution. Our engine cylinder example had three cells (corresponding to the categories “4”, “6”, and “8”). But what if there were 10 categories? The shape of the chi-square model would be different.

The terminology used by statisticians to distinguish these models is *degrees of freedom*, abbreviated *df*. The reason for this name and the mathematics behind it are complicated and technical. Suffice it to say for now that if there are c cells, you use $c - 1$ degrees of freedom. For our car example, there are 3 cylinder categories, so $df = 2$.

Look at the graph below that shows the theoretical chi-square models for varying degrees of freedom. (Don’t worry about the code here. You won’t need to do this on your own.)

```r
ggplot(data.frame(x = c(0, 20)), aes(x)) +
    stat_function(fun = dchisq, args = list(df = 2), aes(color = "2")) +
    stat_function(fun = dchisq, args = list(df = 5), aes(color = "5" )) +
    stat_function(fun = dchisq, args = list(df = 10), aes(color = "10")) +
    scale_color_manual(name = "df",
                       values = c("2" = "red",
                                  "5" = "blue",
                                  "10" = "green"),
                       breaks =  c("2", "5", "10"))
```



The red curve (corresponding to $df = 2$) looks a lot like our simulation above. But as the degrees of freedom increase, the mode shifts further to the right.

Chi-square goodness-of-fit test

The formal inferential procedure for examining whether data from a categorical variable fits a proposed distribution in the population is called a *chi-square goodness-of-fit test*.

We can use the chi-square model as the sampling distribution as long as the sample size is large enough. This is checked by calculating that the expected cell counts (not the observed cell counts!) are at least 5 in each cell.

We'll walk through the engine cylinder example from top to bottom using the rubric.

Exploratory data analysis

Use data documentaton (help files, code books, Google, etc.), the str command, and other summary functions to understand the data.

[Type `?mtcars` at the Console to read the help file.]

```
str(mtcars)
```

```
## 'data.frame':    32 obs. of  11 variables:
##  $ mpg : num  21 21 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 ...
##  $ cyl : num  6 6 4 6 8 6 8 4 4 6 ...
##  $ disp: num  160 160 108 258 360 ...
##  $ hp  : num  110 110 93 110 175 105 245 62 95 123 ...
##  $ drat: num  3.9 3.9 3.85 3.08 3.15 2.76 3.21 3.69 3.92 3.92 ...
##  $ wt  : num  2.62 2.88 2.32 3.21 3.44 ...
##  $ qsec: num  16.5 17 18.6 19.4 17 ...
##  $ vs  : num  0 0 1 1 0 1 0 1 1 1 ...
##  $ am  : num  1 1 1 0 0 0 0 0 0 0 ...
##  $ gear: num  4 4 4 3 3 3 3 4 4 4 ...
##  $ carb: num  4 4 1 1 2 1 4 2 2 4 ...
```

Prepare the data for analysis.

```
# Although we've already done this above,
# we include it here again for completeness.
cyl <- factor(mtcars$cyl)
```

Make tables or plots to explore the data visually.

```
table(cyl)
```

```
## cyl
##  4  6  8
## 11  7 14
```

Hypotheses

Identify the sample (or samples) and a reasonable population (or populations) of interest.

The sample is a set of 32 cars from a 1974 Motor Trends magazine. The population is all cars from 1974. (We would not want to assume that the number of cylinders in cars today would be similar to the distribution in 1974.)

Express the null and alternative hypotheses as contextually meaningful full sentences.

H_0: In 1974, there were the same number of cars with 4, 6, and 8 cylinders.

H_A: In 1974, there weren't the same number of cars with 4, 6, and 8 cylinders.

Express the null and alternative hypotheses in symbols.

$H_0 : p_4 = p_6 = p_8$

There is no easy way to express the alternate hypothesis in symbols because any deviation in any of the categories can lead to rejection of the null. You can't just say $p_4 \neq p_6 \neq p_8$ because two of these categories might be the same and the third different and that would still be consistent with the alternative hypothesis.

So the only requirement here is to express the null in symbols.

Model

Identify the sampling distribution model.

We use a χ^2 model with 2 degrees of freedom.

Check the relevant conditions to ensure that model assumptions are met.

- Random
  - We do not know how Motor Trends magazine sampled these 32 cars, so we're not sure if this list is random or representative of all cars from 1974. We should be cautious in our conclusions.
- 10%
  - As long as there are at least 320 different car models, we are okay. This sounds like a lot, so this condition might not quite be met. Again, we need to be careful. (Also note that the population is not all automobiles manufactured in 1974. It is all *types* of automobile manufactured in 1974. There's a big difference.)
- Expected cell counts
  - This condition says that under the null, we should see at least 5 cars in each category. We expect 10.6666667 in each cell, so this condition is met.

Mechanics

Compute the test statistic.

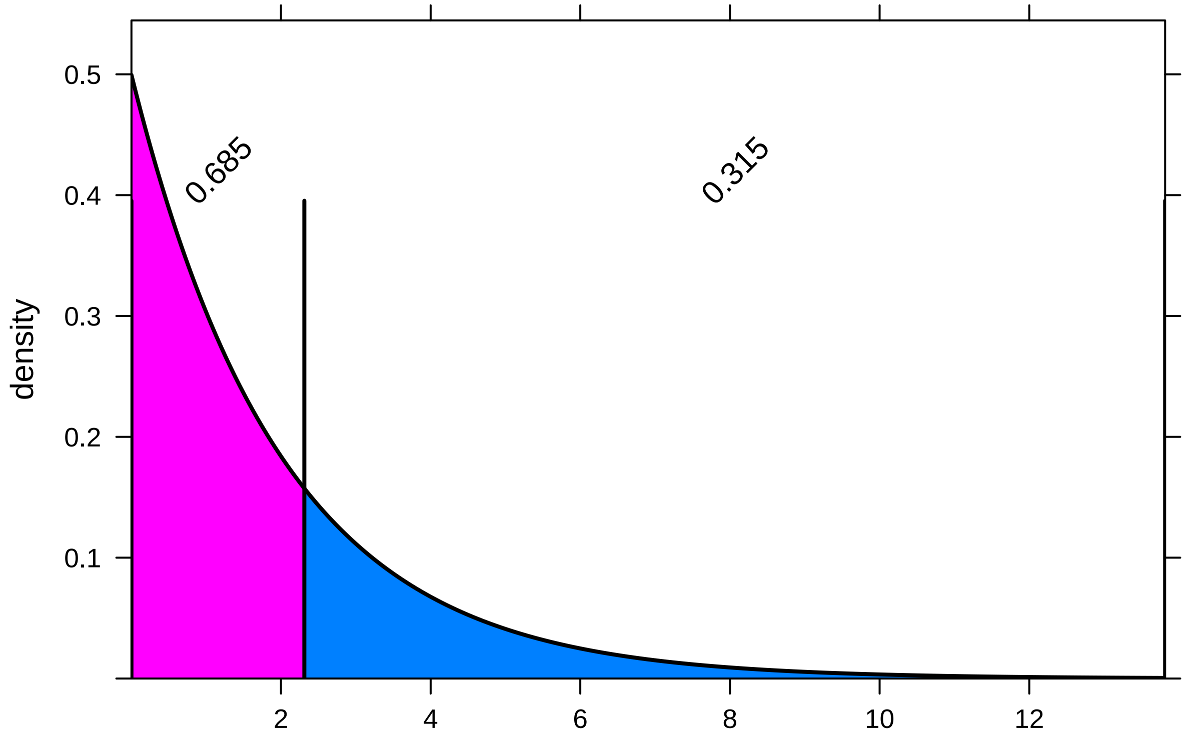
```
cyl_table <- table(cyl)
cyl_test <- chisq.test(cyl_table)
cyl_test$statistic
```

```
## X-squared
##    2.3125
```

Commentary: The `chisq.test` command is a little unusual in that you first have to make a table, and then run the test on the table (not the vector of cylinder values.) The χ^2 test statistic is, of course, the same value we computed manually by hand earlier.

Plot the null distribution.

```
pdist("chisq", df = cyl_test$parameter, q = cyl_test$statistic)
```



```
## X-squared
##  0.685336
```

Commentary: The `pdist` command is familiar, but since we are using a chi-square model and not a normal model, we use `"chisq"` and not `"norm"`. This also requires us to specify the number of degrees of freedom. (Without it, R does not know which chi-square distribution to use.) We could just type $df = 2$ here, but the degrees of freedom is stored in the output of the `chisq.test` command in the `parameter` variable.

Calculate the P-value.

```
1 - pdist("chisq", df = cyl_test$parameter, q = cyl_test$statistic,
         plot = FALSE)
```

```
## X-squared
##  0.314664
```

```
cyl_test$p.value
```

```
## [1] 0.314664
```

Commentary: Values that are as extreme or even more extreme than the test statistic are in the right tail. If we use `pdist`, remember that it always shades to the left by default, so we have to subtract the output from 1 to get the correct P-value. Also remember to add `plot = FALSE` as we don't really need to look at the same picture again. (Ignore the `X-squared` above the P-value. This is carried over from the name assigned to `cyl_test$statistic` and isn't removed by the `pdist` command.)

Alternatively, you can see that the `chisq.test` command gives us the exact same value, storing the result in the `p.value` variable of the output.

Conclusion

State the statistical conclusion.

We fail to reject the null.

State (but do not overstate) a contextually meaningful conclusion.

There is insufficient evidence that in 1974, there weren't the same number of cars with 4, 6, and 8 cylinders.

Identify the possibility of either a Type I or Type II error and state what making such an error means in the context of the hypotheses.

If we made a Type II error, that would mean that there really was a difference in the number of 4, 6, or 8 cylinder cars, but our sample didn't give us enough evidence of a difference to prove it more conclusively.

Confidence interval

There is no confidence interval for a chi-square test. Since our test is not about measuring some parameter of interest (like p or $p_1 - p_2$), there is no interval to produce.

However we will perform a different kind of analysis…

Post hoc analysis

When we reject the null (which we did not do above), we are left with a very vague alternative hypothesis: there is some difference somewhere in one or more categories. Often, we want to follow up to figure out which categories are the ones deviating from the null expectation.

The best way to do this is to look at the *residuals*. A residual for a cell measures how far away the observed count is from the expected count. It does no good just to calculate $O - E$ however; cells with a large count may be far away from their expected values only because they are large numbers. What we want is some kind of relative distance.

We could use the chi-square component from each cell; in other words, we could look at

$$\frac{(O-E)^2}{E}.$$

It is more traditional in statistics to look at the square root of this quantity:

$$\frac{(O-E)}{\sqrt{E}}.$$

Additionally, the above quantity can be positive or negative, and that gives us more information about the direction in which there is a deviation.

Because we failed to reject the null, we didn't have any evidence of a difference anywhere and so there's not much point in examining the residuals. We'll do it here just for practice.

```
cyl_test$residuals
```

```
## cyl
##          4          6          8
##  0.1020621 -1.1226828  1.0206207
```

These numbers don't mean anything in absolute terms; they are only interpretable relative to each other. For example, the first residual is positive, but tiny compared to the others. This means that the observed number of 4-cylinder cars is very close to the expected value. On the other hand, the number of observed 6-cylinder cars is somewhat less than expected, whereas the number of observed 8-cylinder cars is a bit more than expected. If you go back to the table we made earlier, you can verify that.

What if the null is not a uniform distribution?

Suppose we didn't expect an equal number of 4, 6, and 8 cylinder models. How would we run the test under a different null?

Suppose that we expected 35% 4-cylinder cars, 40% 6-cylinder cars, and 25% 8-cylinder cars. We would run the test as follows:

```
cyl_test_2 <- chisq.test(cyl_table, p = c(0.35, 0.4, 0.25))
```

Once you've run the `chisq.test` command, you can see the expected counts in each category.

```
cyl_test_2$expected
```

```
##    4    6    8
## 11.2 12.8  8.0
```

You can check on your own that 11.2 is 35%, 12.8 is 40%, and 8.0 is 25% of 32.

The numbers defining the null have to add up to 1 (since they are percentages). This causes some trouble, for example, when you have a percentage that has an infinite number of decimal places. For example, what if we expected under the null a distribution of 6/13, 4/13, 3/13? You can't express any of these as a decimal without rounding. Well it turns out that `chisq.test` can handle any set of numbers as long as you set `rescale.p =  TRUE`:

```
cyl_test_3 <- chisq.test(cyl_table, p = c(6, 4, 3), rescale.p = TRUE)
```

Also note that these change the statements of the null distribution (in sentences and in symbols). For example, in the example above where the null is 35% 4-cylinder cars, 40% 6-cylinder cars, and 25% 8-cylinder cars, our null in symbols would be

$H_0 : p_4 = 0.35, p_6 = 0.4, p_8 = 0.25$

Your turn

Use the `hsb2` data and determine if the proportion of high school students who attend general programs, academic programs, and vocational programs is 15%, 60%, and 25% respectively. If you reject the null, run a post hoc analysis and comment on the cells that seem to be contributing the most to the discrepancy between observed and expected counts.