```
---
title: 'Unofficial Kahn Academy R Supplement: Two Way Tables for Categorical
Data'
output:
  pdf_document: default
  html_notebook: default
  html_document: default
---
```

In the last lesson we briefly touched on one-way frequency tables, or tables in which the frequency of different values of a single variable are displayed. Lesson two focuses on two-way frequency tables, or tables in which the frequencies of values of two variables are displayed. Two-way frequency tables (or "crosstabs" as they are also called) are often very useful when you want to look at the no-nonsense, very basic relationship between two categorical variables. There are other more advanced methods to test the relationship between categorical variables, but crosstabs are a often a good starting point to lay bare the structure of the data in a way that the most basic user or reader can understand.

You can also use it to look at the relationship between two continual variables, but here there are other methods, namely correlation, that are probably more appropriate. We'll address correlation in a later lesson.

The basic two-way frequency table is very simple. I know I just said that two-way tables are best for categorical variables, but the built-in datasets in R are continual, so we'll once again draw from the built-in **mtcars** dataset. In the last lesson, we showed you how to create an object from **mtcars**, but now that that has been demonstrated we'll just go ahead and refer to it without making it an object. If you want to look at a list of the variables in a dataset--built in or otherwise, just put in the name of the dataset between the parantheses in the *ls()* command, so in the case of **mtcars** it would be *ls(mtcars)*. To create a simple two-way frequency table just use the *table* command with the two variables inside the parantheses (seperated by a comma). Remember, that because R is capable of holding multiple datasets at a time, you need to explicitly say that the variable is coming from a certain dataset, so for the number of **gears** in **mtcars** it would be **mtcars$gear**. To include, say number of gears and number of cylinders in a cross-tab, we would just type:

```{r}
table(mtcars$gear, mtcars$cyl)
```

However, if you are going to be using the same dataset for a long time, you can also use the *attach* command to, well, "attach" the dataset so that R knows that you are drawing from that dataset. If you do this you don't have to refer to the parent dataset everytime you use a variable. You use this command by simply placing the name of the dataset inside the attach parantheses, so with mtcars it would be *attach(mtcars)*. Conversely, to tell R to stop using that dataset, you use the *detach* command.

The first variable mentioned represents the variable represented by the column to the far left (so there are 3, 4, and 5 gears), and the second variable

mentioned represents ths top row (so 4,6, and 8 and cylinders), with all the values in between representing the number of cars that have each combination of cylinders and gears. AYCS, the most common combination appears to be cars with 8 gears and 3 cyliners. This has 12 cases, or, in statistical jargon, N=12.

However, with crosstabs it's often easier and more efficient to look at percentages or proportions instead of raw numbers. To look at proportions, make the frequency table an object, then run the *prop.table* command on that object.

```{r}
raw_table<-table(mtcars$gear, mtcars$cyl)
prop.table(raw_table)
```

Although once again remember that instead of separately making the frequency table, assigning it to an object, then performing *prop.table* on that table, we can just do it in one line of code with multiple parantheses and it will give us the same result.

```{r}
prop.table(table(mtcars$gear, mtcars$cyl))
```

Once again, you can spruce this up so that you don't have to paste it into an excel worksheet and add the title and such. This website has some of the information on more advanced crosstab possibilities: http://www.statmethods.net/stats/frequencies.html.