Restaurant Tips dataset

The owner of a bistro called First Crush in Potsdam, New York, is interested in studying the tipping patterns of its patrons. He collected restaurant bills over a two-week period that he believes provide a good sample of his customers. The data from 157 bills are stored in RestaurantTips and include the amount of the bill, size of the tip, percentage tip, number of customers in the group, whether or not a credit card was used, day of the week, and a coded identity of the server.

In particular, PctTip shows the tip amount expressed as a percentage of the bill. Most people use a fairly regular percentage (which may vary from person to person) of the total bill when deciding how big a tip to leave. Some economists have theorized that people tend to reduce that percentage when the bill gets large, but it could also be the other way around, customers might be more generous when eating in larger groups, thus spending more money, due to peer pressure. We can use the RestaurantTip data to see if there is evidence to support either theory, or perhaps there is no consistent relationship between the size of the bill and percent tip.

- Choose variables Bill and PctTip to analyse their linear dependency through Pearson's correlation coefficient. Just looking at the scatterplot, it is hard to tell whether this coefficient is significantly different from zero (check this!). Conduct a permutation test to test the null hypothesis that the correlation coefficient is 0 vs the alternative that it is different from 0. Run $R = 10000$ simulations.

- Repeat the analysis deleting the values for three customers that left a tip greater than 30% of the bill. These generous customers seem to be outliers.

Permutation Test for the correlation coefficient

Permutation testing can be traced back to at least Fisher (1935). Instead of comparing the actual value of a test statistic to a standard statistical distribution, the reference distribution is generated from the data themselves, as described below. Permutation provides an efficient approach to testing when the data do not conform to the distributional assumptions of the statistical method one wants to use (e.g. normality).

Recall the null hypothesis is that correlation is equal to 0. This means that there is no linear relationship between the two variables. If that is true, then any of the Y observations is just as likely to appear with any of the X's. In other words, Y_i is just as likely to appear with X_i as it is to appear with $X_j,\ i \neq j$.

Suppose we have a set of n ordered pairs $(X_i, Y_i),$ for $i = 1, \ldots, n$. Steps:

1. Calculate the observed correlation between the variables. Call it r_{obs}.

2. Permute the Y's among the $X's$ (in $n!$ ways !), i.e., hold one variable (X) constant and permute the Y. We will actually do that for a random set of R permutations instead.

3. For each permutation, calculate r (i.e., r between the Y variable permuted and the X).

4. For an upper-tail test, the p-value is:

$$\frac{\text{number of r's} > r_{obs}}{R}$$

   For a two-tailed test:

$$\frac{\text{number of } |r's| > |r_{obs}|}{R}$$

Using this method you have computed the permutation distribution of r under the null hypothesis of no correlation between the variables. You can plot it using an histogram and represent the p-value on it.

Observations

When programming in R, try to avoid loops. In this case, take advantage of the vectorization in R. You can use functions *sample* and *replicate* although there are definitely more elegant ways to do that.

Unless you use the command *set.seed* before starting the simulations, everyone is going to obtain different results. The important thing here is that these results lead you to the same conclusion.