**More on Exploring Correlations in R**

August 28, 2012

By [Stephen Turner](http://www.r-bloggers.com/author/stephen-turner/)

(This article was first published on [**Getting Genetics Done**](http://gettinggeneticsdone.blogspot.com/2012/08/more-on-exploring-correlations-in-r.html), and kindly contributed to [R-bloggers)](http://www.r-bloggers.com/)

About a year ago I wrote a post about producing [scatterplot matrices in R](http://gettinggeneticsdone.blogspot.com/2011/07/scatterplot-matrices-in-r.html). These are handy for quickly getting a sense of the correlations that exist in your data. Recently someone asked me to pull out some relevant statistics (correlation coefficient and p-value) into tabular format to publish beside a scatterplot matrix. The built-in cor() function will produce a correlation matrix, but what if you want p-values for those correlation coefficients? Also, instead of a matrix, how might you get these statistics in tabular format (variable *i*, variable *j*, r, and p, for each *i*-*j* combination)? Here's the code (you'll need the PerformanceAnalytics package to produce the plot).

|  |  |
| --- | --- |
| 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 | ## Correlation matrix with p-values. See http://goo.gl/nahmV for documentation of this function  cor.prob <- function (X, dfr = nrow(X) - 2) {  R <- cor(X, use="pairwise.complete.obs")  above <- row(R) < col(R)  r2 <- R[above]^2  Fstat <- r2 \* dfr/(1 - r2)  R[above] <- 1 - pf(Fstat, 1, dfr)  R[row(R) == col(R)] <- NA  R  }    ## Use this to dump the cor.prob output to a 4 column matrix  ## with row/column indices, correlation, and p-value.  ## See StackOverflow question: http://goo.gl/fCUcQ  flattenSquareMatrix <- function(m) {  if( (class(m) != "matrix") | (nrow(m) != ncol(m))) stop("Must be a square matrix.")  if(!identical(rownames(m), colnames(m))) stop("Row and column names must be equal.")  ut <- upper.tri(m)  data.frame(i = rownames(m)[row(m)[ut]],  j = rownames(m)[col(m)[ut]],  cor=t(m)[ut],  p=m[ut])  }    # get some data from the mtcars built-in dataset  mydata <- mtcars[, c(1,3,4,5,6)]    # correlation matrix  cor(mydata)    # correlation matrix with p-values  cor.prob(mydata)    # "flatten" that table  flattenSquareMatrix(cor.prob(mydata))    # plot the data  library(PerformanceAnalytics)  chart.Correlation(mydata) |

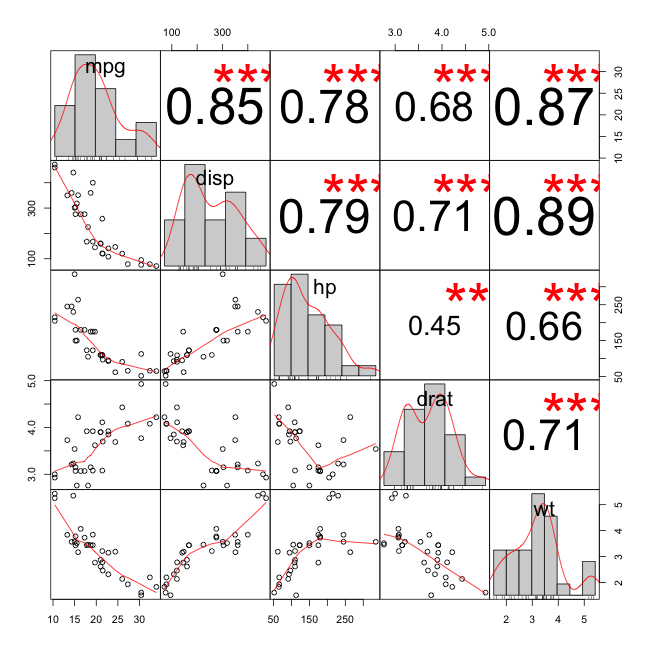
[view raw](https://gist.github.com/stephenturner/3492773/raw/explore-correlations.r) [explore-correlations.r](https://gist.github.com/stephenturner/3492773#file-explore-correlations-r) hosted with ❤ by [GitHub](https://github.com)

The cor() function will produce a basic correlation matrix.  12 years ago [Bill Venables provided a function on the R help mailing list](https://stat.ethz.ch/pipermail/r-help/2000-January/009758.html) for replacing the upper triangle of the correlation matrix with the p-values for those correlations (based on the known relationship between *t* and *r*). The cor.prob() function will produce this matrix.  
  
Finally, the flattenSquareMatrix() function will "flatten" this matrix to four columns: one column for variable *i*, one for variable *j*, one for their correlation, and another for their p-value (thanks to [Chris Wallace on StackOverflow](http://stackoverflow.com/questions/12116207/flatten-matrix-in-r-to-four-columns-indexes-and-upper-lower-triangles) for helping out with this one).

|  |  |
| --- | --- |
| 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 | > cor(mydata)  mpg disp hp drat wt  mpg 1.0000000 -0.8475514 -0.7761684 0.6811719 -0.8676594  disp -0.8475514 1.0000000 0.7909486 -0.7102139 0.8879799  hp -0.7761684 0.7909486 1.0000000 -0.4487591 0.6587479  drat 0.6811719 -0.7102139 -0.4487591 1.0000000 -0.7124406  wt -0.8676594 0.8879799 0.6587479 -0.7124406 1.0000000    > cor.prob(mydata)  mpg disp hp drat wt  mpg NA 9.380327e-10 1.787835e-07 1.776240e-05 1.293958e-10  disp -0.8475514 NA 7.142679e-08 5.282022e-06 1.222322e-11  hp -0.7761684 7.909486e-01 NA 9.988772e-03 4.145827e-05  drat 0.6811719 -7.102139e-01 -4.487591e-01 NA 4.784260e-06  wt -0.8676594 8.879799e-01 6.587479e-01 -7.124406e-01 NA    > flattenSquareMatrix(cor.prob(mydata))  i j cor p  1 mpg disp -0.8475514 9.380327e-10  2 mpg hp -0.7761684 1.787835e-07  3 disp hp 0.7909486 7.142679e-08  4 mpg drat 0.6811719 1.776240e-05  5 disp drat -0.7102139 5.282022e-06  6 hp drat -0.4487591 9.988772e-03  7 mpg wt -0.8676594 1.293958e-10  8 disp wt 0.8879799 1.222322e-11  9 hp wt 0.6587479 4.145827e-05  10 drat wt -0.7124406 4.784260e-06 |

[view raw](https://gist.github.com/stephenturner/3492876/raw/explore-correlations-output.txt) [explore-correlations-output.txt](https://gist.github.com/stephenturner/3492876#file-explore-correlations-output-txt) hosted with ❤ by [GitHub](https://github.com)

Finally, the chart.Correlation() function from the PerformanceAnalytics package produces a very nice scatterplot matrix, with histograms, kernel density overlays, absolute correlations, and significance asterisks (0.05, 0.01, 0.001):

[](http://i0.wp.com/3.bp.blogspot.com/-p9TU0FkR0MQ/UDvzQBEEOlI/AAAAAAABA8Q/ScF-742PvWE/s1600/explore-correlations.png)

Getting Genetics Done by Stephen Turner is licensed under a Creative Commons Attribution-ShareAlike 3.0 Unported License.