R news and tutorials contributed by (573) R bloggers

- Home
- About
- RSS
- add your blog!
- Learn R
- R jobs
- Contact us

Welcome!

Follow @rbloggers 28K fo

Here you will find daily **news and tutorials about R**, contributed by over 573 bloggers. There are many ways to **follow us -**

By e-mail:

Your e-mail here
Subscribe
25169 readers
BY FEEDBURNER

On Facebook:

R blogg...
30K likes
Like Page
Be the first of your friends to like this

If you are an R blogger yourself you are invited to add your own R content feed to this site (**Non-English** R bloggers should add themselves- here)

Jobs for R-users

- Senior Statistician/Statistician (>$100/year) @ Shinjuku-ku, Tōkyō-to, Japan
- Junior Data Scientist @ Farnham, England
- Vehicle Valuations Manager @ Farnham, England
- Manager – Quantitative Analytics @ London, United Kingdom
- Pharmacometrics

14
6
6

Group Meetup: 24 February 2016 @ The Green Door
- Clusters of (French) Regions
- New Release of RStudio (v0.99.878)
- An Introduction to Time Series with JSON Data
- EARL London 2016 – Call for Abstracts
- Using SVG graphics in blog posts

Other sites

- Statistics of Israel
- SAS blogs
- Jobs for R-users

More on Exploring Correlations in R

August 28, 2012
By Stephen Turner

(This article was first published on **Getting Genetics Done**, and kindly contributed to R-bloggers)

About a year ago I wrote a post about producing scatterplot matrices in R. 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).

```
## Correlation matrix with p-values. See http://goo.gl/nahmV f
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("Mus
  if(!identical(rownames(m), colnames(m))) stop("Row and colum
  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])
}

```

```r
# 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)
```

explore-correlations.r hosted with ♥ by GitHub view raw

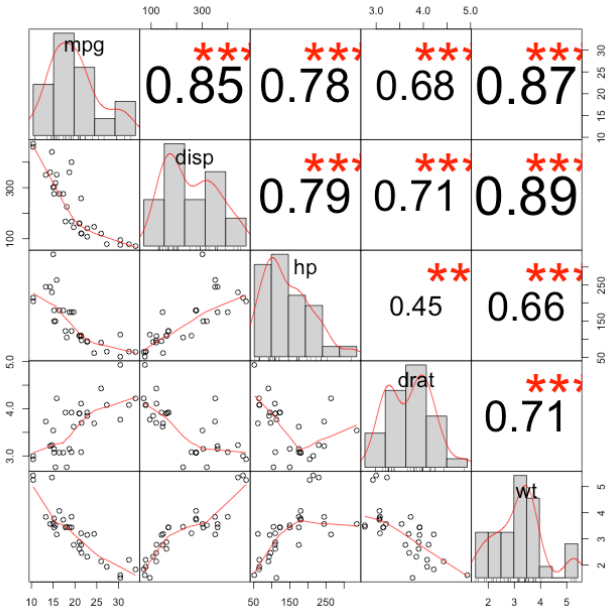
The `cor()` function will produce a basic correlation matrix. 12 years ago Bill Venables provided a function on the R help mailing list for replacing the upper triangle of the correlation matrix with the p-values for those correlations (based on the known relationship between *t* and *r*). The `cor.prob()` 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 for helping out with this one).

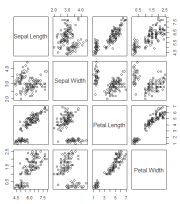
```
> cor(mydata)
            mpg       disp         hp       drat         wt
mpg   1.0000000 -0.8475514 -0.7761684  0.6811719 -0.8676594
disp -0.8475514  1.0000000  0.7909486 -0.7102139  0.8879799
hp   -0.7761684  0.7909486  1.0000000 -0.4487591  0.6587479
drat  0.6811719 -0.7102139 -0.4487591  1.0000000 -0.7124406
wt   -0.8676594  0.8879799  0.6587479 -0.7124406  1.0000000

> cor.prob(mydata)
            mpg          disp            hp          drat
mpg          NA  9.380327e-10  1.787835e-07  1.776240e-05 1.29
disp -0.8475514            NA  7.142679e-08  5.282022e-06 1.22
hp   -0.7761684  7.909486e-01            NA  9.988772e-03 4.14
drat  0.6811719 -7.102139e-01 -4.487591e-01            NA 4.78
wt   -0.8676594  8.879799e-01  6.587479e-01 -7.124406e-01

> 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
```

explore-correlations-output.txt hosted with ♥ by GitHub view raw

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):

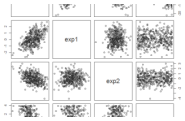


Related



Scatterplot matrices in R
In "R bloggers"

Why an inverse-Wishart prior may not be such a good idea
In "R bloggers"



Breaking the rules with spatial correlation
Similar post

Comments are closed.

Search & Hit Enter

Recent popular posts

- Finding the K in K-means by Parametric Bootstrap
- The Easiest Way to Learn ggplot2

- New Release of RStudio (v0.99.878)

Most visited articles of the week

1. Installing R packages
2. How to Learn R
3. Using apply, sapply, lapply in R
4. What has Kaggle learned from 2 million machine learning models?
5. In-depth introduction to machine learning in 15 hours of expert videos
6. How to Make a Histogram with Basic R
7. New Release of RStudio (v0.99.878)
8. Scatterplots
9. Computing and visualizing PCA in R

Sponsors

Jobs for R users

- Senior Statistician/Statistician (>$100/year) @ Shinjuku-ku, Tōkyō-to, Japan
- Junior Data Scientist @ Farnham, England
- Vehicle Valuations Manager @ Farnham, England
- Manager – Quantitative Analytics @ London, United Kingdom
- Pharmacometrics Bootcamp @ Wellesley, Massachusetts, US
- Data Scientist for TIBCO (>$100K/year)
- Senior Financial Analyst – Consumer Marketing Analytics @ Seattle, US