Vignettes for package mycor

Keon-Woong Moon cardiomoon@gmail.com

1-Oct-2014

Contents

Motivation ................................................................. 1
Solution ; Do not repeat yourself !! .................................... 1
Function “mycor” and Class “mycor” .................................... 2
Plot “mycor” object ....................................................... 3

Motivation

For correlation analysis : cor, cor.test and lm

When I began to study R software, I was really impressed with the function cor. As you knows, the ‘cor’ function returns all \( r \) values of every possible pairs of matrix or data.frame provided that it consists of numeric data only. For example, \( \text{cor} \text{mtcars[1:5]} \) acts just as expected.

\[
\begin{array}{cccccc}
\text{mpg} & \text{cyl} & \text{disp} & \text{hp} & \text{drat} \\
\text{mpg} & 1.0000 & -0.8522 & -0.8476 & -0.7762 & 0.6812 \\
\text{cyl} & -0.8522 & 1.0000 & 0.9020 & 0.8324 & -0.6999 \\
\text{disp} & -0.8476 & 0.9020 & 1.0000 & 0.7909 & -0.7102 \\
\text{hp} & -0.7762 & 0.8324 & 0.7909 & 1.0000 & -0.4488 \\
\text{drat} & 0.6812 & -0.6999 & -0.7102 & -0.4488 & 1.0000 \\
\end{array}
\]

But cor(iris) returns error because the data.frame consist of both numeric and factor variable.

\[
\begin{array}{cc}
\text{cor(iris)} \\
\end{array}
\]

Error: 'x' must be numeric

If you wanted to get \( p \) values as well as \( r \) values, you should use cor.test instead of cor. But cor.test can deal with only one pair of numeric vectors of the same length, neither a matrix nor a data.frame. Furthermore, if you wanted to get the slope and intercept of simple linear regression line of xyplot, you had to perform lm test for every pairs of numeric variables of the data.frame.

Solution ; Do not repeat yourself !!

My idea is that a single function deals with data.frame of mixed numeric, logical and factor variables, select numeric variables, perform cor.test and lm to get \( r,p\text{.slope} \) and intercept of every pairs of the variables for exploratory analysis. It can save my time and effort.
Function “mycor” and Class “mycor”

Use of mycor function is simple. Just call mycor with a data.frame. For example, just call cor(iris).
Unlike cor, it does not result in an error.

```r
require(lattice)
require(mycor)
mycor(iris)
```

$r$ value by Pearson's product-moment correlation

```
   Sepal.Length Sepal.Width Petal.Length Petal.Width
Sepal.Length     1.000   -0.118     0.872     0.818
Sepal.Width    -0.118     1.000    -0.428    -0.366
Petal.Length    0.872    -0.428     1.000     0.963
Petal.Width     0.818    -0.366     0.963     1.000
```

$p$ value (two.sided)

```
   Sepal.Length Sepal.Width Petal.Length Petal.Width
Sepal.Length   0.000     0.152       0       0
Sepal.Width    0.152     0.000       0       0
Petal.Length   0.000     0.000       0       0
Petal.Width    0.000     0.000       0       0
```

The mycor function returns an object of class “mycor”. This can be saved for print, summarize and plot. A S3 method for class formula can be used to function mycor. Function print.mycor shows the $r$ values and the $p$ values similar to the function cor. A mycor class object can be summarized with summary function, summary().

```r
out = mycor(iris, alternative = "greater", method = "kendall", digits = 2)
out1 = mycor(~mpg + disp + hp + wt, data = mtcars)
summary(out1)
```

$r$ value by Pearson's product-moment correlation

```
          mpg  disp  hp  wt
mpg   1.000 -0.848 -0.776 -0.868
disp  -0.848  1.000  0.791  0.888
hp    -0.776  0.791  1.000  0.659
wt    -0.868  0.888  0.659  1.000
```

$p$ value (two.sided)

```
          mpg  disp  hp  wt
mpg      0    0    0    0
disp     0    0    0    0
hp       0    0    0    0
wt       0    0    0    0
```

$slope$
```
mpg  disp  hp  wt
mpg  1.00 -0.04 -0.07 -5.34
disp -17.43 1.00 1.43 112.48
hp  -8.83 0.44 1.00 46.16
wt  -0.14 0.01 0.01 1.00

$ intercept

mpg  disp  hp  wt
mpg  0.00 29.60 30.10 37.29
disp  580.88 0.00 20.99 -131.15
hp  324.08 45.73 0.00 -1.82
wt   6.05 1.60 1.84 0.00
```

The `mycor` function uses `cor.test` internally, so you can use all options of `cor.test` - namely `alternatives`, `method`, `conf.level`, ... .

**Plot “mycor” object**

Probably most valuable function is `plot`. It is not a new function. It uses internally one of two popular function: `graphics::pairs()` and `lattice::parallelplot()`. In fact, `plot.mycor` function have four types of plot: Three variants of pairs and parallelplot. Call function `plot` with no option makes `pairs()`.

```
plot(out)
```
But if you specify the groups, you can get more pretty plot. You can use extra arguments which can used in `pairs()` or `parallelplot()`.

```r
plot(out, groups=species, main="Test of mycor::plot")
```

![Test of mycor::plot](image-url)
With `type=2` option, you can get histogram at diagonal panel.

```r
plot(out,type=2,groups=spe)
```
With type=3 option, you can get correlation plot at upper panels.

```r
plot(out1, type=3)
```
With `type=4` option, you can get parallelplot.

```r
plot(out, type=4, groups=spe)
```
plot(out1,type=4,groups=cyl)