Post02

Ming Chen

December 2, 2017

Make correlation plot by using `corrplot` and `ggplot2`

Introduction and Motivation

We have learned the `cor` function in lecture to calculate the correlation coefficient between variables in data set. However, we will get a messy correlation matrix when the data set contains lots of variables. It is hard to read every numbers to analyze the correlations. Instead, it is helpful if we can visualize the correlation matrix by making a correlation plot.

There are several ways to make correlation plot:

- using `ggplot2` and `reshape2`
- using `corrplot`

I would like to introduce a useful package `corrplot` to help with making correlation plot in this post. Since we learned `ggplot2` in lecture, I will also discuss about how to make correlation plot by using package `ggplot2` and `reshape2`.

1. Data preparation

I choose the `mtcars` dataset which R already provided as an example. You can also try any other dataset by yourself.

```
#inspect the data
dim(mtcars)
```

```
## [1] 32 11
```

```
#get the correlations of the first 6 variables in columns of the dataset
data <- cor(mtcars[1:6])
data
```

```
##             mpg        cyl       disp         hp       drat         wt
## mpg   1.0000000 -0.8521620 -0.8475514 -0.7761684  0.6811719 -0.8676594
## cyl  -0.8521620  1.0000000  0.9020329  0.8324475 -0.6999381  0.7824958
## disp -0.8475514  0.9020329  1.0000000  0.7909486 -0.7102139  0.8879799
## hp   -0.7761684  0.8324475  0.7909486  1.0000000 -0.4487591  0.6587479
## drat  0.6811719 -0.6999381 -0.7102139 -0.4487591  1.0000000 -0.7124406
## wt   -0.8676594  0.7824958  0.8879799  0.6587479 -0.7124406  1.0000000
```

2. Using `corrplot` package

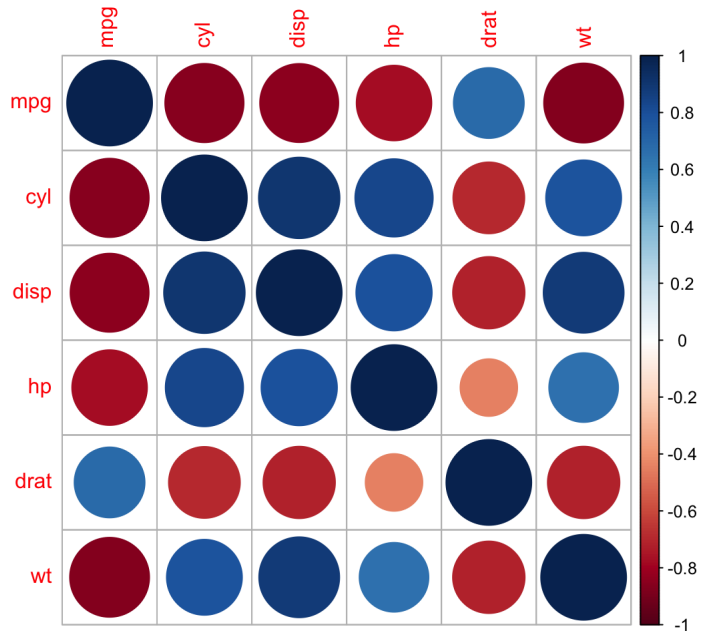
At first, install the package and load it. Then, let's try `corrplot` function to make a simple correlation plot.

```
#install.packages('corrplot')
library(corrplot)
```

```
## Warning: package 'corrplot' was built under R version 3.4.2
```

```
## corrplot 0.84 loaded
```

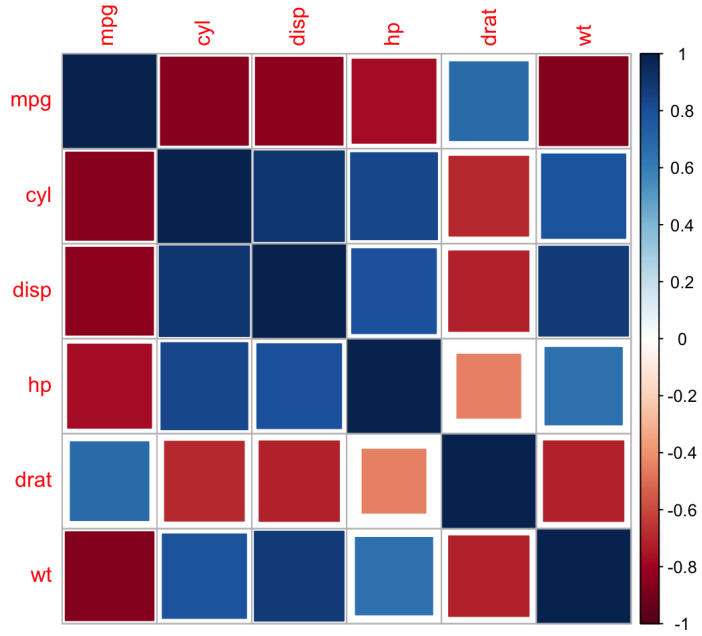
```
corrplot(corr = data)
```



It is easy to make a simple plot. Moreover, we have different options to make the plot more beautiful.

- change argument `method` to plot different shapes or just numbers.

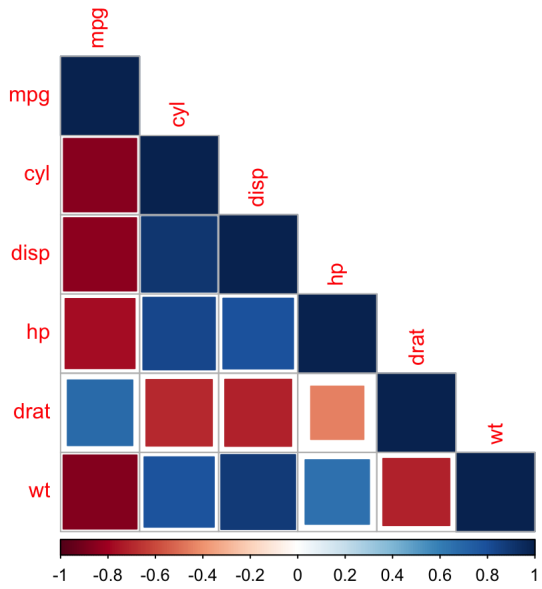
```
corrplot(corr = data, method = 'square')
```



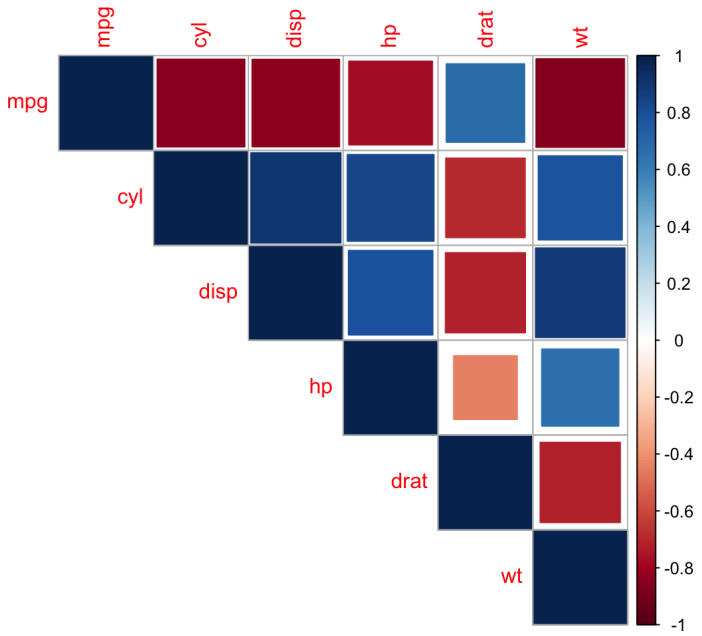

```
corrplot(corr = data, method = 'number')
```

- change argument `type` to represent the plot in different ways: `full`, `lower`, `upper`

```
corrplot(corr = data, method = 'square', type = 'lower')
```

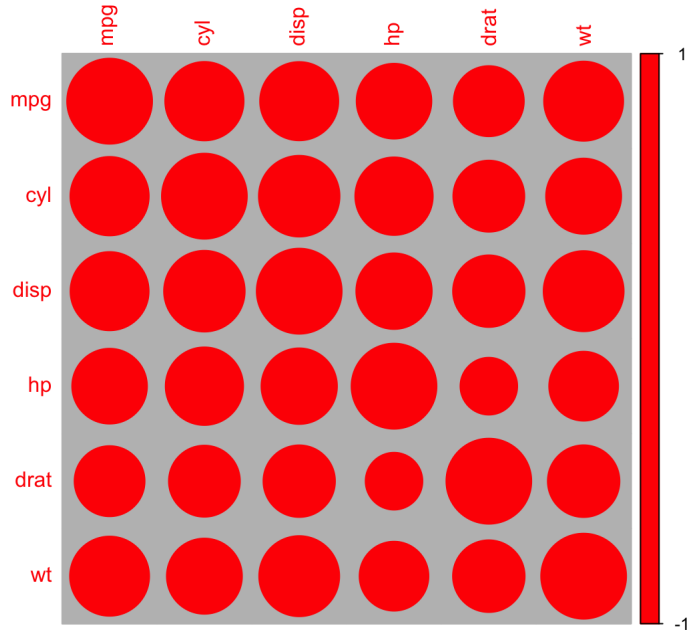



```
corrplot(corr = data, method = 'square', type = 'upper')
```



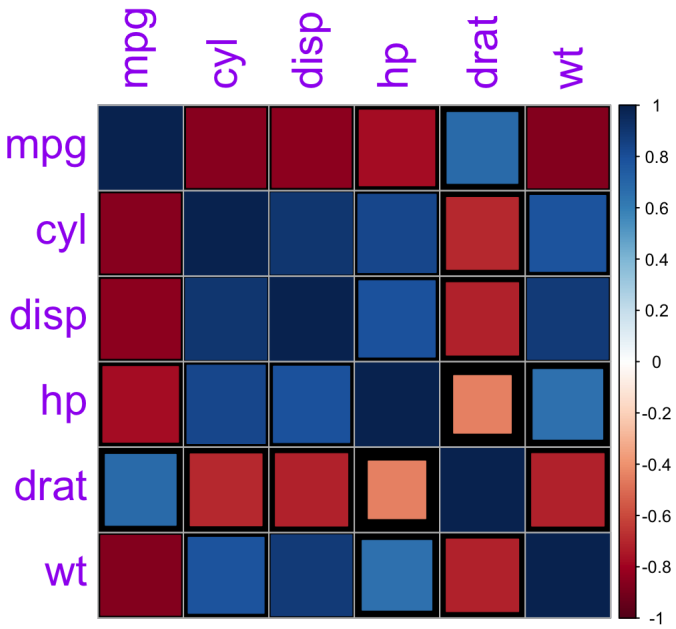
- change argument `col` to decide the color of shapes and using argument `bg` to decide the color of the background of the plot.

```
corrplot(corr = data, method = 'circle', type = 'full', col = 'red', bg = 'grey')
```



- using argument `tl.cex` to change the size of the text label
- using argument `tl.col` to change the color of the text label

```
corrplot(corr = data, method = 'square', type = 'full', bg = 'black',
         tl.cex = 2, tl.col = 'purple')
```



3. Using `ggplot2` package

At first, install the package and load it. Here, we need a another package `reshape2` to help with making a simple correlation plot. Therefore, I also install package `reshape2` and load it.

```
#install.packages('ggplot2')
#install.packages('reshape2')
library(ggplot2)
library(reshape2)
```

Data reorganization:

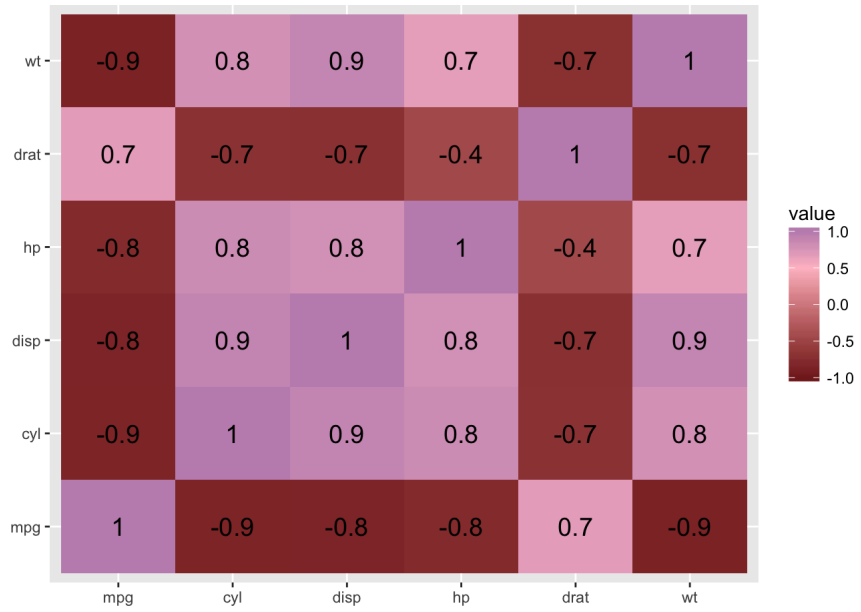
```
#use `melt` function to break up the correlation table
melted_data <- melt(data)

#inspect the melted data
head(melted_data, 10)
```

```
##    Var1 Var2      value
## 1   mpg  mpg  1.0000000
## 2   cyl  mpg -0.8521620
## 3  disp  mpg -0.8475514
## 4    hp  mpg -0.7761684
## 5  drat  mpg  0.6811719
## 6    wt  mpg -0.8676594
## 7   mpg  cyl -0.8521620
## 8   cyl  cyl  1.0000000
## 9  disp  cyl  0.9020329
## 10   hp  cyl  0.8324475
```

For ggplot2, we have to use `melt` function to break up the correlation table into a long format table. So we get Var1 as x-aexis, Var2 as y-aexis and value to plot.

Let's make a full correlation plot by using `ggplot` function:

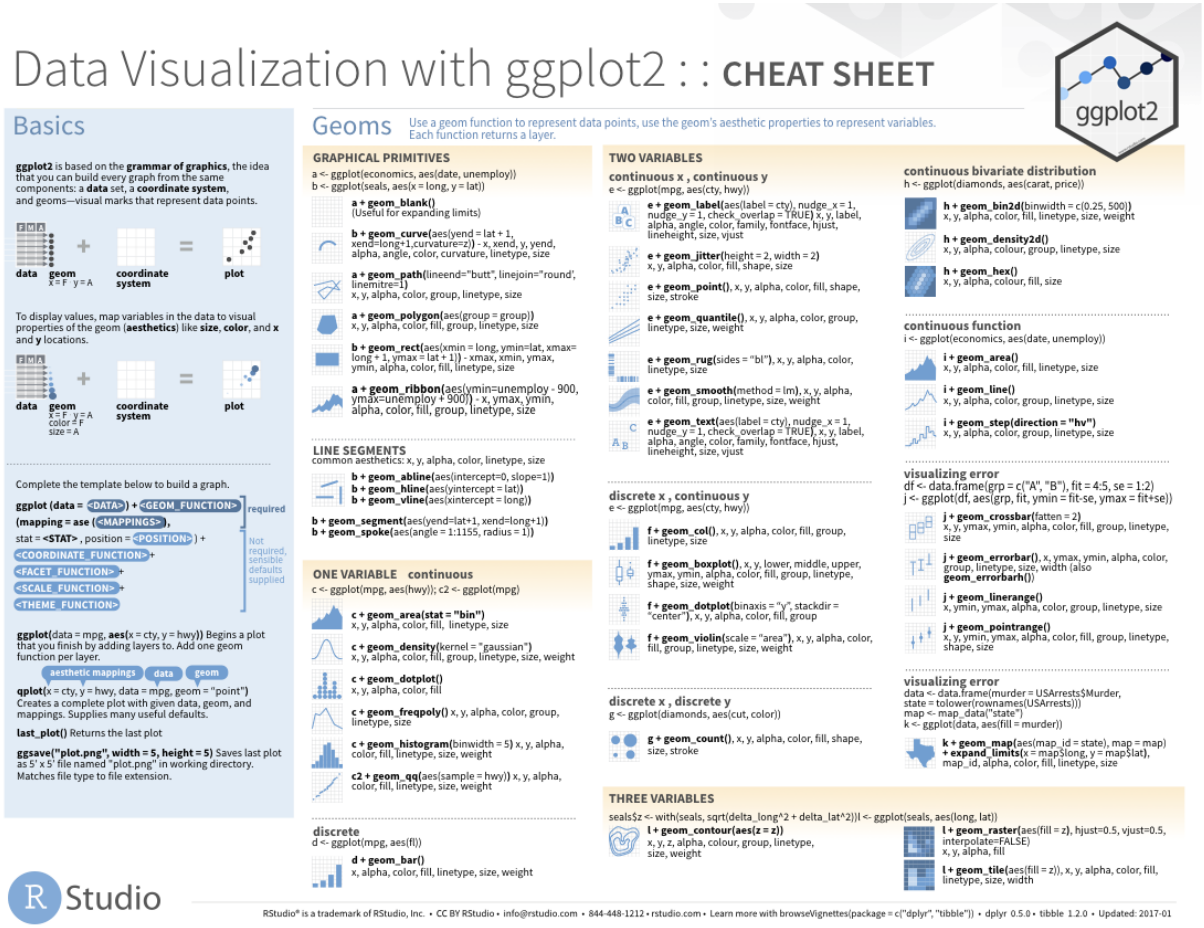
```
ggplot(data = melted_data, aes(x = Var1, y = Var2, fill = value)) +
      geom_tile() +
      scale_fill_gradient2(midpoint = 0.5, mid = 'pink', limits = c(-1, 1))+
      labs(x = '', y = '') +
      geom_text(aes(x = Var1, y = Var2, label = round(value, 1)), size = 5)
```



Some functions mentioned when I make correlation plot by using `ggplot2` :

- `scale_fill_gradient2` function is used with the argument limit = c(-1,1) as correlation coefficients range from -1 to 1.
- `geom_tile` function is used to tile Plane With Rectangles
- `geom_text` funxtion is used to adds text directly to the plot.
- `labs` function is used to change axis labels and legend titles.

You can find more functions here:



4. Conclusion

From the class, we learned to use ggplot2 to make histogram, scatterplot, barchart an so on. Visualization is an improtant tool to analyze data. Making correlation plot is also ver helpful when you deal with many correlation coefficients. It is quite easy to plot it by using `corrplot` function. You can also use `ggplot2` , which is a little bit more complicated. `ggplot2` is also fun and have lots of functions to explore.

Reference

An Introduction to corrplot Package

Plot variable Correlations

Rdocument

ggcorrplot

ggcorr

Plot correlation matrix with R in specific data range

Seven Easy Graphs to Visualize Correlation Matrices in R

Plot a correlation matrix with ggplot2

Correlations