**R Advanced Data Visualization**

* Heat Map
* Mosaic Map
* Map Visualization
* 3D Graphs
* Correlogram

**Heat Map**

A **heat map** (or **heatmap**) is a graphical representation of data where the individual values contained in a matrix are represented as colors. It is a bit like looking a data table from above. It is really useful to display a general view of numerical data, not to extract specific data point. It is quite straight forward to make a heat map, as shown on the examples below. However be careful to understand the underlying mechanisms.

# The mtcars dataset:

data=as.matrix(mtcars)

head(data)

# Default Heatmap (left)

heatmap(data)

# Use 'scale' to normalize (right)

heatmap(data, scale="column")

# No dendrogram nor reordering for neither column or row

heatmap(data, Colv = NA, Rowv = NA, scale="column")

# 1: native palette from R

heatmap(data, scale="column", col = cm.colors(256))

heatmap(data, scale="column", col = terrain.colors(256))

# 2: Rcolorbrewer palette

library(RColorBrewer)

coul = colorRampPalette(brewer.pal(8, "PiYG"))(25)

heatmap(data, scale="column", col = coul)

**Mosaic Map**

In **mosaic**: Project **MOSAIC** Statistics and Mathematics Teaching Utilities ... A function of the transformation to be performed to the key.**map** column ... Unit Data · RBesT: **R** Bayesian Evidence Synthesis Tools · rvcheck: **R**/Package Version Check ... rdrr.io home **R language** documentation Run **R**code online Create free **R**

# Mosaic Plot Example

library(vcd)

mosaic(HairEyeColor, shade=TRUE, legend=TRUE)

Association Plots

To produce an extended association plot use **assoc(x, row\_vars, col\_vars)**where **x**is a contingency table, **row\_vars** is a vector of integers giving the indices of the variables to be used for the rows, and **col\_vars**is a vector of integers giving the indices of the variables to be used for the columns of the association plot.

# Association Plot Example

library(vcd)

assoc(HairEyeColor, shade=TRUE)

**Map Visualization**

## Getting started with ggmap

install.packages("ggmap")

library(ggmap)

The fastest way to get going is with the qmap class, which stands for “quick map plot”. Play around with the different types of parameter calls to render various plot types.

Some examples to start:

qmap(location = "new delhi")

qmap(location = "boston university")

qmap(location = "boston university", zoom = 14)

qmap(location = "boston university", zoom = 14, source = "osm")

**R 3d plots**

**Impressive package for 3D and 4D graph - R software and data visualization**

* Install plot3D package.
* Load plot3D package.
* Prepare the data.
* Scatter **plots**. Functions for scatter **plots** and texts in 2D and **3D**. Basic scatter **plot**. ...
* Line **plots**. Add confidence interval. ...
* text3D: **plot** 3-dimensionnal texts.
* text3D and scatter3D.
* **3D** Histogram.

# Install plot3D package

install.packages("plot3D")

# Load plot3D package

**library**("plot3D")

# Prepare the data

We’ll use the iris data set in the following examples :

data(iris)

**head**(iris)

Sepal.Length Sepal.Width Petal.Length Petal.Width Species

1 5.1 3.5 1.4 0.2 setosa

2 4.9 3.0 1.4 0.2 setosa

3 4.7 3.2 1.3 0.2 setosa

4 4.6 3.1 1.5 0.2 setosa

5 5.0 3.6 1.4 0.2 setosa

6 5.4 3.9 1.7 0.4 setosa

# x, y and z coordinates

x <- sep.l <- iris$Sepal.Length

y <- pet.l <- iris$Petal.Length

z <- sep.w <- iris$Sepal.Width

# Scatter plots

## Functions for scatter plots and texts in 2D and 3D

The function below will be used:

scatter3D(x, y, z, **...**, colvar = z, col = NULL, add = FALSE)

text3D(x, y, z, labels, colvar = NULL, add = FALSE)

points3D(x, y, z, **...**)

lines3D(x, y, z, **...**)

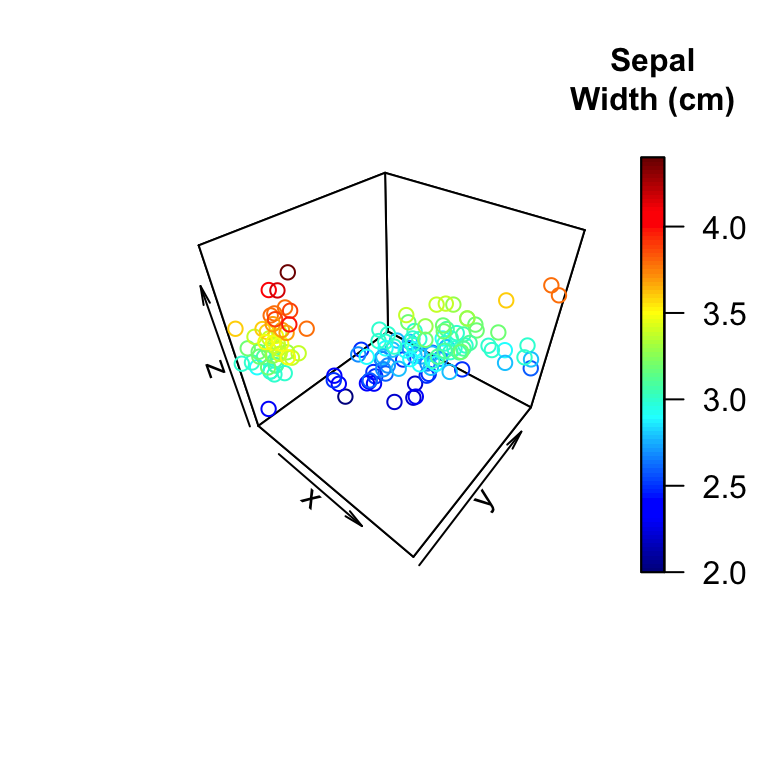
scatter2D(x, y, colvar = NULL, col = NULL, add = FALSE)

text2D(x, y, labels, colvar = NULL, col = NULL, add = FALSE)

* **x, y, z**: vectors of point coordinates
* **colvar**: a variable used for coloring
* **col**: color palette used for coloring the colvar variable
* **labels**: the text to be written
* **add**: logical. If TRUE, then the points will be added to the current plot. If FALSE a new plot is started
* **…**: additional persp arguments including xlim, ylim, zlim, xlab, ylab, zlab, main, sub, r, d, scale, expand, box, axes, nticks, tictype.

## Basic scatter plot

scatter3D(x, y, z, clab = c("Sepal", "Width (cm)"))



**Correlogram**

# Install R corrplot package

**corrplot** package is required to execute the R code in this article.

install.packages("corrplot")

# Data for correlation analysis

The **mtcars** data is used to compute **correlation matrix**.

**head**(mtcars)

mpg cyl disp hp drat wt qsec vs am gear carb

Mazda RX4 21.0 6 160 110 3.90 2.620 16.46 0 1 4 4

Mazda RX4 Wag 21.0 6 160 110 3.90 2.875 17.02 0 1 4 4

Datsun 710 22.8 4 108 93 3.85 2.320 18.61 1 1 4 1

Hornet 4 Drive 21.4 6 258 110 3.08 3.215 19.44 1 0 3 1

Hornet Sportabout 18.7 8 360 175 3.15 3.440 17.02 0 0 3 2

Valiant 18.1 6 225 105 2.76 3.460 20.22 1 0 3 1

# Computing correlation matrix

M<-cor(mtcars)

**head**(round(M,2))

mpg cyl disp hp drat wt qsec vs am gear carb

mpg 1.00 -0.85 -0.85 -0.78 0.68 -0.87 0.42 0.66 0.60 0.48 -0.55

cyl -0.85 1.00 0.90 0.83 -0.70 0.78 -0.59 -0.81 -0.52 -0.49 0.53

disp -0.85 0.90 1.00 0.79 -0.71 0.89 -0.43 -0.71 -0.59 -0.56 0.39

hp -0.78 0.83 0.79 1.00 -0.45 0.66 -0.71 -0.72 -0.24 -0.13 0.75

drat 0.68 -0.70 -0.71 -0.45 1.00 -0.71 0.09 0.44 0.71 0.70 -0.09

wt -0.87 0.78 0.89 0.66 -0.71 1.00 -0.17 -0.55 -0.69 -0.58 0.43

# Correlogram : Visualizing the correlation matrix

R **corrplot** function is used to plot the **graph of the correlation matrix**.

The simplified format of the function is :

corrplot(corr, method="circle")

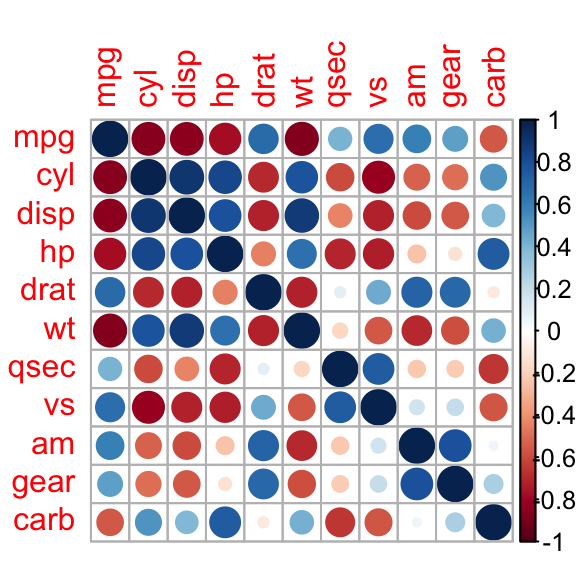
| Arguments | Description |
| --- | --- |
| corr | The correlation matrix to visualize. To visualize a general matrix, please use is.corr=FALSE. |
| method | The visualization method : “circle”, “color”, “number”, etc. |

## Visualization methods

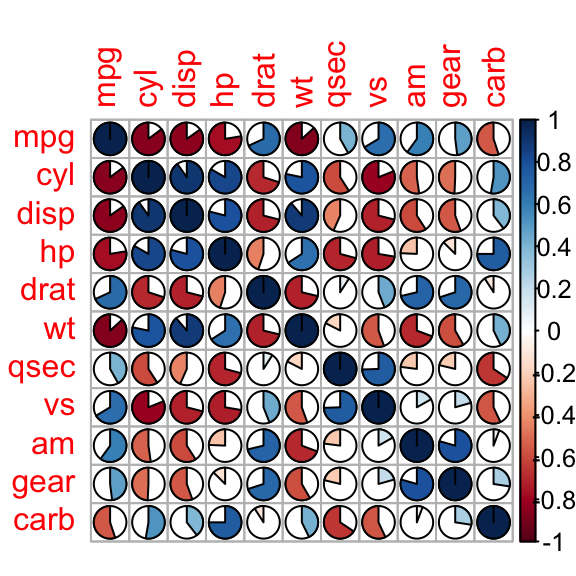
Seven different **visualization** methods can be used : “circle”, “square”, “ellipse”, “number”, “shade”, “color”, “pie”.

**library**(corrplot)

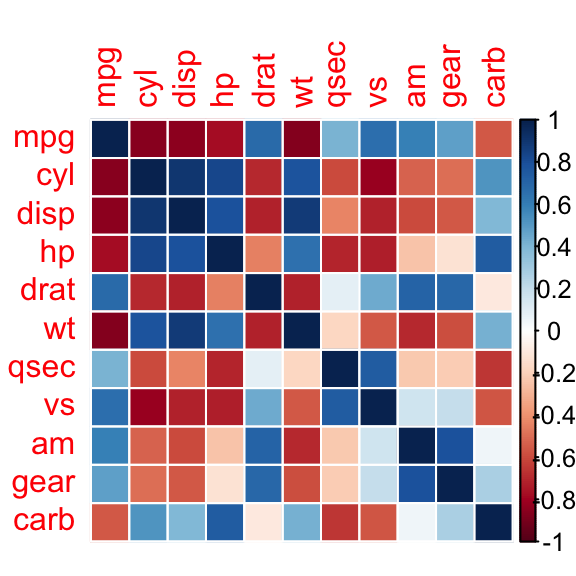
corrplot(M, method="circle")



corrplot(M, method="pie")

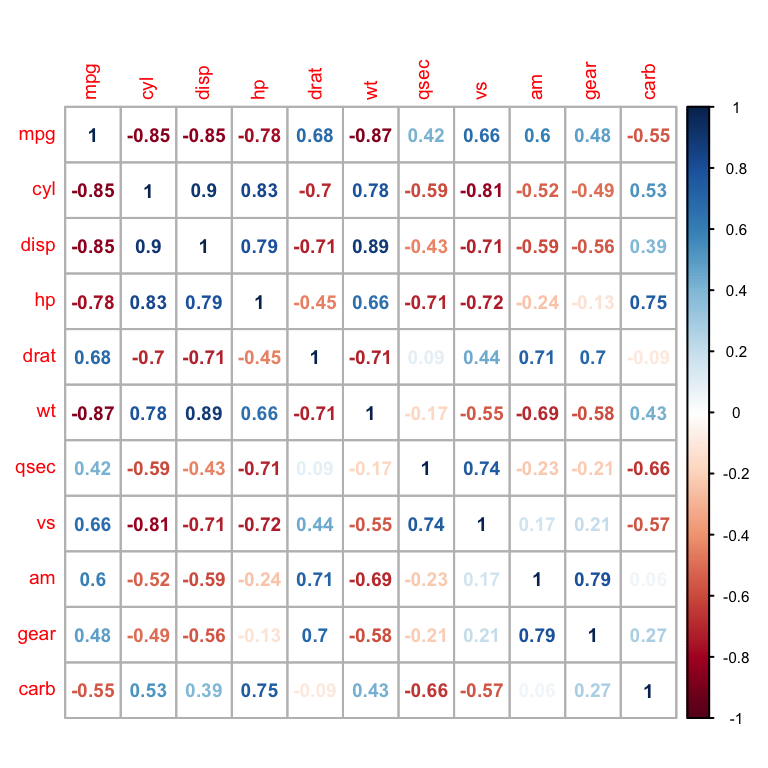


corrplot(M, method="color")



Display the **correlation coefficient** :

corrplot(M, method="number")

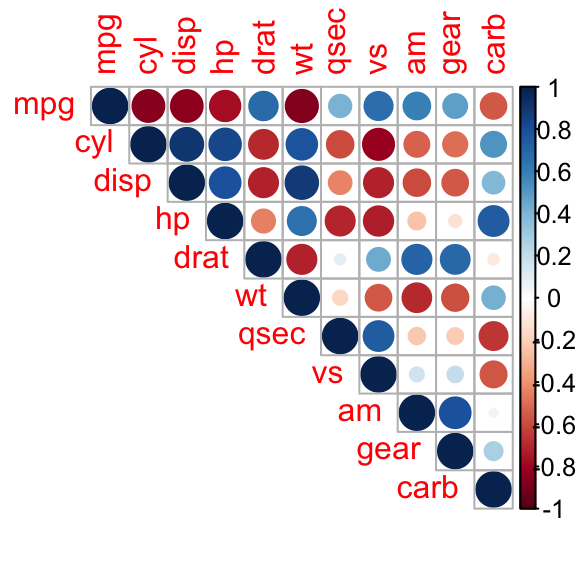


## Types of correlogram layout

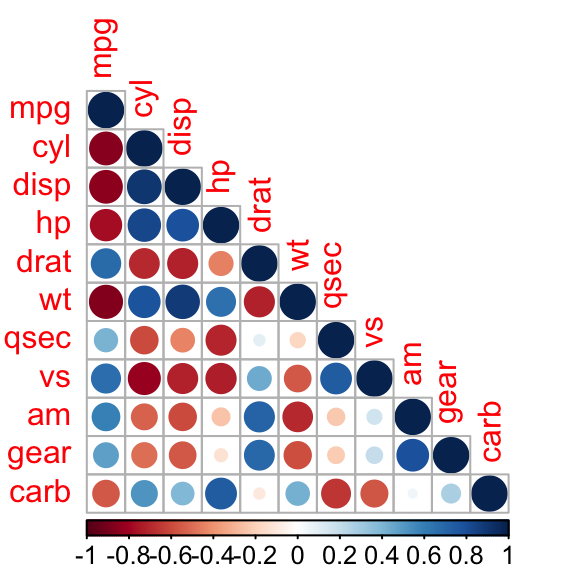
There are three types of layout :

* **“full”** (default) : display full **correlation matrix**
* **“upper”**: display upper triangular of the **correlation matrix**
* **“lower”**: display lower triangular of the **correlation matrix**

corrplot(M, type="upper")



corrplot(M, type="lower")

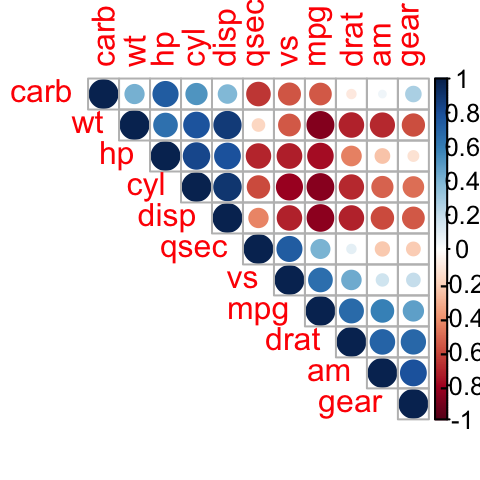


## Reordering the correlation matrix

The **correlation matrix** can be reordered according to the **correlation coefficient**. This is important to identify the hidden structure and pattern in the matrix. **“hclust”** for hierarchical clustering order is used in the following examples.

# correlogram with hclust reordering

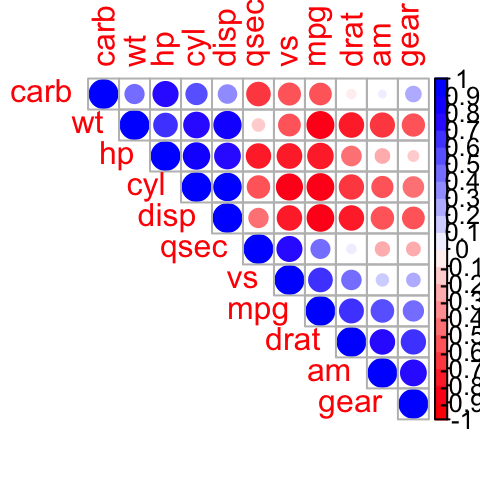
corrplot(M, type="upper", order="hclust")



# Using different color spectrum

col<- colorRampPalette(c("red", "white", "blue"))(20)

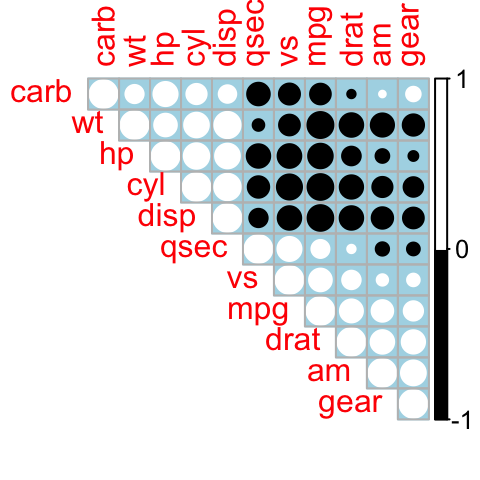
corrplot(M, type="upper", order="hclust", col=col)



# Change background color to lightblue

corrplot(M, type="upper", order="hclust", col=c("black", "white"),

bg="lightblue")



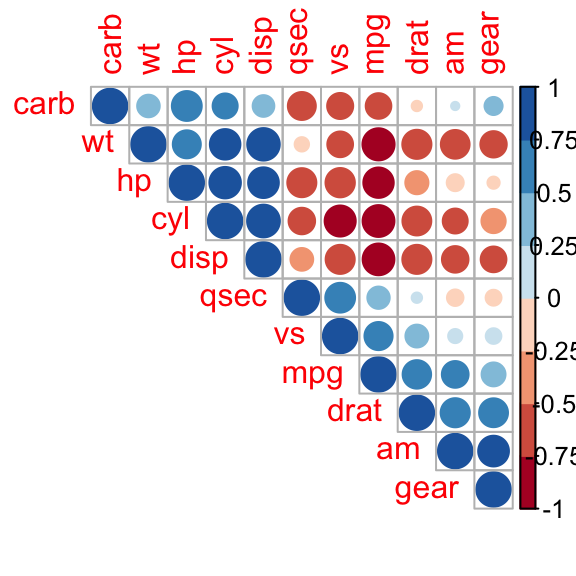
## Changing the color of the correlogram

As shown in the above section, the color of the correlogram can be customized. **RcolorBrewer** palette of colors are used in the R script below :

**library**(RColorBrewer)

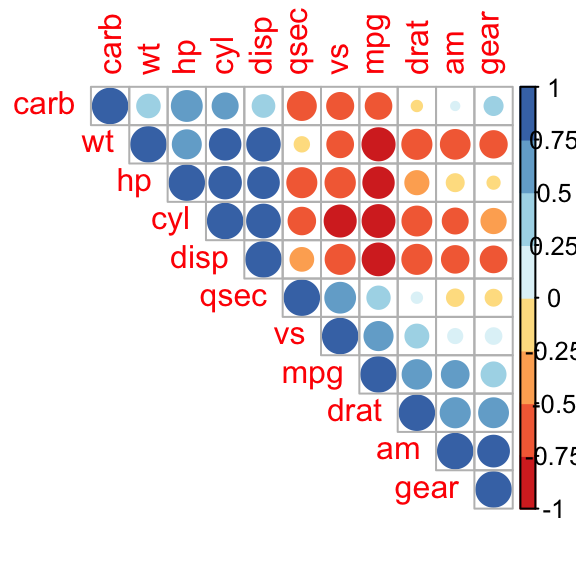
corrplot(M, type="upper", order="hclust",

col=brewer.pal(n=8, name="RdBu"))



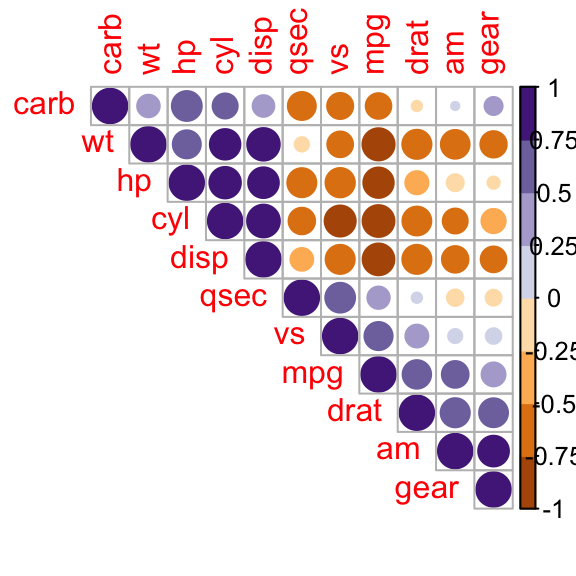
corrplot(M, type="upper", order="hclust",

col=brewer.pal(n=8, name="RdYlBu"))



corrplot(M, type="upper", order="hclust",

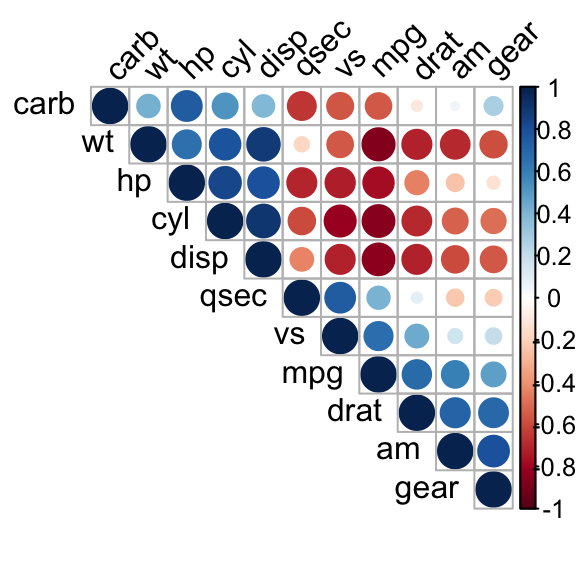
col=brewer.pal(n=8, name="PuOr"))



## Changing the color and the rotation of text labels

**tl.col** (for text label color) and **tl.srt** (for text label string rotation) are used to change text colors and rotations.

corrplot(M, type="upper", order="hclust", tl.col="black", tl.srt=45)



## Combining correlogram with the significance test

### Computing the p-value of correlations

To **compute the matrix of p-value**, a custom R function is used :

# mat : is a matrix of data

# ... : further arguments to pass to the native R cor.test function

cor.mtest <- **function**(mat, **...**) {

mat <- as.matrix(mat)

n <- ncol(mat)

p.mat<- matrix(NA, n, n)

diag(p.mat) <- 0

**for** (i **in** 1:(n - 1)) {

**for** (j **in** (i + 1):n) {

tmp <- cor.test(mat[, i], mat[, j], **...**)

p.mat[i, j] <- p.mat[j, i] <- tmp$p.value

}

}

colnames(p.mat) <- rownames(p.mat) <- colnames(mat)

p.mat

}

# matrix of the p-value of the correlation

p.mat <- cor.mtest(mtcars)

**head**(p.mat[, 1:5])

mpg cyl disp hp drat

mpg 0.000e+00 6.113e-10 9.380e-10 1.788e-07 1.776e-05

cyl 6.113e-10 0.000e+00 1.803e-12 3.478e-09 8.245e-06

disp 9.380e-10 1.803e-12 0.000e+00 7.143e-08 5.282e-06

hp 1.788e-07 3.478e-09 7.143e-08 0.000e+00 9.989e-03

drat 1.776e-05 8.245e-06 5.282e-06 9.989e-03 0.000e+00

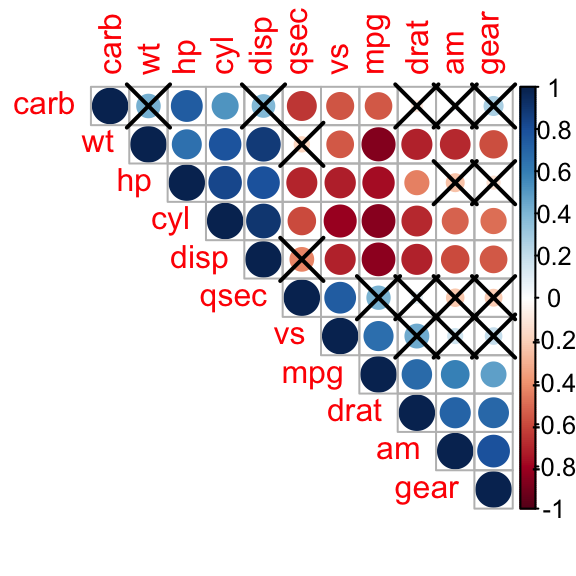
wt 1.294e-10 1.218e-07 1.222e-11 4.146e-05 4.784e-06

### Add significance level to the correlogram

# Specialized the insignificant value according to the significant level

corrplot(M, type="upper", order="hclust",

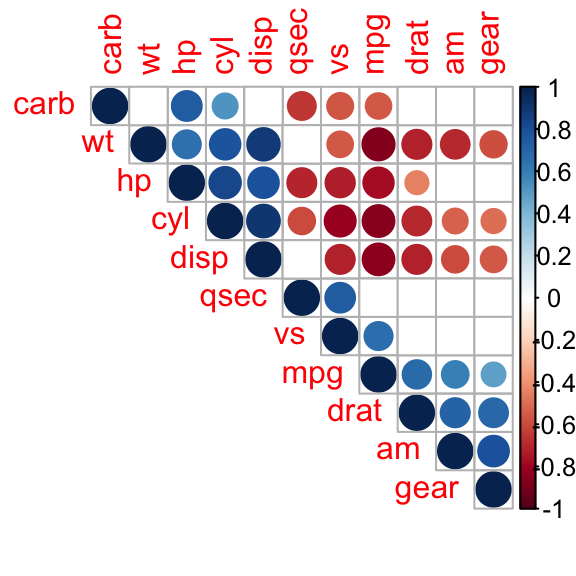
p.mat = p.mat, sig.level = 0.01)



# Leave blank on no significant coefficient

corrplot(M, type="upper", order="hclust",

p.mat = p.mat, sig.level = 0.01, insig = "blank")



In the above figure, **correlations** with **p-value** > 0.01 are considered as insignificant. In this case the **correlation coefficient values** are leaved blank or crosses are added.

## Customize the correlogram

col <- colorRampPalette(c("#BB4444", "#EE9988", "#FFFFFF", "#77AADD", "#4477AA"))

corrplot(M, method="color", col=col(200),

type="upper", order="hclust",

addCoef.col = "black", # Add coefficient of correlation

tl.col="black", tl.srt=45, #Text label color and rotation

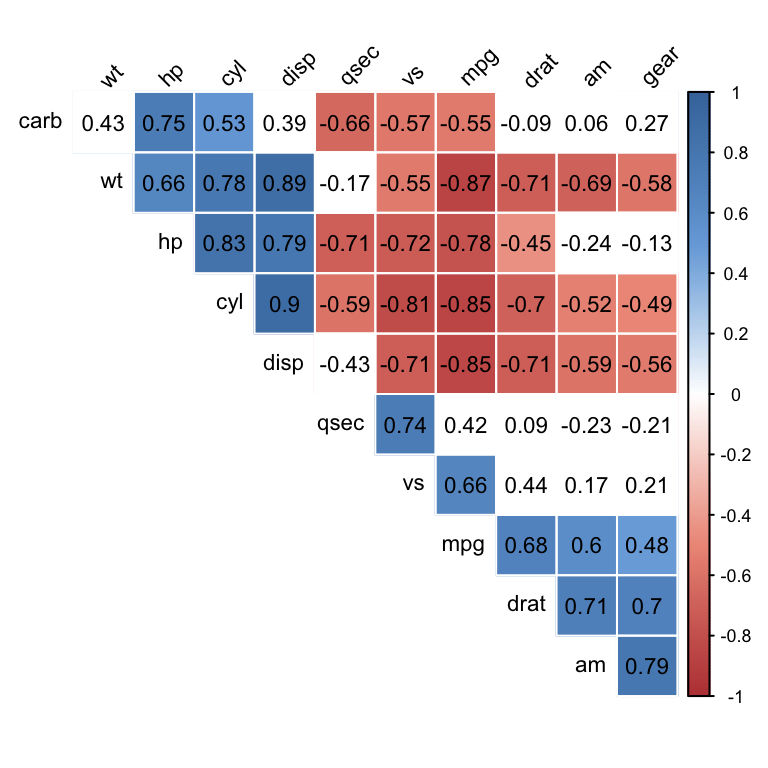
# Combine with significance

p.mat = p.mat, sig.level = 0.01, insig = "blank",

# hide correlation coefficient on the principal diagonal

diag=FALSE

)



setwd("C:\\Users\\vkumar15\\Desktop")

data = read.csv("prince\_raw\_data.csv")

data$text

Keywords <- as.character(data$text)

is.character(Keywords)

key = Keywords[1:5]

key[1]

str = ""

for(x in key){

#str = paste(str,x)

print(x)

}

d <- c("this is test word","this is r programn","hello test")

str = ""

for(x in d){

str = paste(str,x)

}

print(str)

words <- strsplit(str) # this, is ,are

table(words)

mtcars

str(mtcars)

data=as.matrix(mtcars)

heatmap(data)

heatmap(data, scale="column")

##

library(vcd)

HairEyeColor

mosaic(HairEyeColor, shade=TRUE, legend=TRUE)

library("plot3D")

x <- sep.l <- iris$Sepal.Length

y <- pet.l <- iris$Petal.Length

z <- sep.w <- iris$Sepal.Width

scatter3D(x, y, z, colvar = z, col = NULL, add = FALSE)