

# Correlation matrix : A quick start guide to analyze, format and visualize a correlation matrix using R software

- [What is a correlation matrix ?](#)
- [Correlation analysis in R](#)
- [Data for correlation analysis](#)
- [Correlation matrix](#)
- [Correlation significance levels \(p-value\)](#)
- [A simple function to format the correlation matrix](#)
- [Visualization of the correlation matrix](#)
  - [Visualize a correlation matrix using the symnum function](#)
  - [Visualize a correlation matrix using a correlogram](#)
  - [Visualize a correlation matrix using scatter plots](#)
  - [Visualize correlation matrix using a heatmap](#)
- [Online software to analyze and visualize a correlation matrix](#)
- [conclusions](#)
- [Infos](#)

## What is a correlation matrix ?

A correlation matrix is used to investigate the dependence between multiple variables at the same time. The result is a table containing the correlation coefficients between each variable and the others.

There are different methods for correlation analysis : Pearson parametric correlation test, Spearman and Kendall rank-based correlation analysis. These methods are discussed in the next sections.

Correlation matrix can be visualized using a [correlogram](#).

The aim of this R tutorial is to show you how to compute and visualize a correlation matrix in R.

Note that an online software is also available here : Take me to the [Online software](#) to compute a correlation matrix and to plot a correlogram without any installation. This application is described at the end of this article.

## Correlation analysis in R

As you may know, The R function `cor()` can be used to compute a correlation matrix. A simplified format of the function is :

```
# x is a matrix or data.frame
cor(x, method = c("pearson", "kendall", "spearman"))
```

The argument method indicates the correlation coefficient to be computed. The default is [pearson correlation coefficient](#) which measures the linear dependence between two variables. [kendall and spearman](#) correlation methods are non-parametric rank-based correlation test.

## Data for correlation analysis

The `mtcars` data is used in the following examples to compute a 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

## Correlation matrix

```
mcor <- cor(mtcars)
mcor
```

	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
qsec	0.42	-0.59	-0.43	-0.71	0.09	-0.17	1.00	0.74	-0.23	-0.21	-0.66
vs	0.66	-0.81	-0.71	-0.72	0.44	-0.55	0.74	1.00	0.17	0.21	-0.57
am	0.60	-0.52	-0.59	-0.24	0.71	-0.69	-0.23	0.17	1.00	0.79	0.06
gear	0.48	-0.49	-0.56	-0.13	0.70	-0.58	-0.21	0.21	0.79	1.00	0.27
carb	-0.55	0.53	0.39	0.75	-0.09	0.43	-0.66	-0.57	0.06	0.27	1.00

In the table above correlations coefficients between the possible pairs of variables are shown.

Note that, if your data contain missing values, use the following R code to handle missing values by case-wise deletion.

```
cor(mtcars, use = "complete.obs")
```

## Correlation significance levels (p-value)

The output of the function `cor()` is the correlation coefficients between each variable and the others. Unfortunately, this function doesn't display the correlation significance levels (or p-value). In the next section, we will use Hmisc R package to calculate the correlation p-values.

The function `rcorr()` from [Hmisc](#) package can be used to compute the significance levels for pearson and spearman correlations. It returns both the correlation coefficients and the p-value of the correlation for all possible pairs of columns in the data table.

The simplified format of this function is :

```
rcorr(x, type=c("pearson", "spearman"))
```

x should be a matrix. The correlation type can be either pearson or spearman.

Make sure that the Hmisc package (required for the next step) is installed.

```
# install.packages("Hmisc")
library(Hmisc)
rcorr(as.matrix(mtcars[,1:7]))

      mpg   cyl  disp    hp  drat    wt   qsec 
mpg  1.00 -0.85 -0.85 -0.78  0.68 -0.87  0.42 
cyl -0.85  1.00  0.90  0.83 -0.70  0.78 -0.59 
disp -0.85  0.90  1.00  0.79 -0.71  0.89 -0.43 
hp   -0.78  0.83  0.79  1.00 -0.45  0.66 -0.71 
drat  0.68 -0.70 -0.71 -0.45  1.00 -0.71  0.09 
wt   -0.87  0.78  0.89  0.66 -0.71  1.00 -0.17 
qsec  0.42 -0.59 -0.43 -0.71  0.09 -0.17  1.00 

n= 32
```

```
P
      mpg   cyl  disp    hp  drat    wt   qsec 
mpg  0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0171 
cyl  0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0004 
disp 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0131 
hp   0.0000 0.0000 0.0000 0.0000 0.0100 0.0000 0.0000 
drat 0.0000 0.0000 0.0000 0.0100 0.0000 0.0000 0.6196 
wt   0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.3389 
qsec 0.0171 0.0004 0.0131 0.0000 0.6196 0.3389
```

The output of the function `rcorr()` is a list with the elements below :

- r : the correlation matrix
- n : the matrix of the number of observations used in analyzing each pair of variables
- P : the p-values corresponding to the significance levels of correlations.

## A simple function to format the correlation matrix

This section provides a simple function for formatting a correlation matrix into a table with 4 columns containing :

- Column 1 : row names (variable 1 for the correlation test)
- Column 2 : column names (variable 2 for the correlation test)
- Column 3 : the correlation coefficients
- Column 4 : the p-values of the correlations

The custom function below can be used :

```
# ++++++
# flattenCorrMatrix
# ++++++
# cormat : matrix of the correlation coefficients
# pmat : matrix of the correlation p-values
flattenCorrMatrix <- function(cormat, pmat) {
  ut <- upper.tri(cormat)
  data.frame(
    row = rownames(cormat)[row(cormat)[ut]],
    column = rownames(cormat)[col(cormat)[ut]],
    cor = (cormat)[ut],
    p = pmat[ut]
  )
}
```

Example of usage :

```
library(Hmisc)
res<-rcorr(as.matrix(mtcars[,1:7]))
flattenCorrMatrix(res$r, res$P)
```

```
      row column    cor      p
1  mpg   cyl -0.8522 6.113e-10
2  mpg  disp -0.8476 9.380e-10
3  cyl  disp  0.9020 1.803e-12
4  mpg   hp -0.7762 1.788e-07
5  cyl   hp  0.8324 3.478e-09
6 disp   hp  0.7909 7.143e-08
7  mpg  drat  0.6812 1.776e-05
8  cyl  drat -0.6999 8.245e-06
9 disp  drat -0.7102 5.282e-06
10 hp   drat -0.4488 9.989e-03
11 mpg   wt -0.8677 1.294e-10
12 cyl   wt  0.7825 1.218e-07
13 disp  wt  0.8880 1.222e-11
14 hp    wt  0.6587 4.146e-05
```

```

15 drat      wt -0.7124 4.784e-06
16 mpg      qsec 0.4187 1.708e-02
17 cyl      qsec -0.5912 3.661e-04
18 disp      qsec -0.4337 1.314e-02
19 hp       qsec -0.7082 5.766e-06
20 drat      qsec 0.0912 6.196e-01
21 wt       qsec -0.1747 3.389e-01

```

## Visualization of the correlation matrix

There are many solutions to visualize a correlation matrix in R software :

- `symnum()` function
- `corrplot()` function to plot a correlogram
- scatter plots
- heatmap

### Visualize a correlation matrix using the `symnum` function

The R function `symnum()` replaces correlation coefficients by symbols according to the level of the correlation. It takes the correlation matrix as an argument :

```

symnum(mcor)

      m  cy ds h dr w  q v a g e r
mpg    1
cyl   + 1
disp  + + 1
hp    , + , 1
drat  , , , . 1
wt    + , + , , 1
qsec  . . . , , 1
vs    + , , , . . , 1
am    . . . , , , 1
gear  . . . , . . , 1
carb  . . . , . . . 1
attr(,"legend")
[1] 0 ' ' 0.3 '.' 0.6 ',' 0.8 '+' 0.9 '*' 0.95 'B' 1

```

As indicated in the legend, the correlation coefficients between 0 and 0.3 are replaced by a space (" "); correlation coefficients between 0.3 and 0.6 are replaced by "."; etc ...

### Visualize a correlation matrix using a correlogram

You have to install the `corrplot` package which performs a graphical display of a correlation matrix.

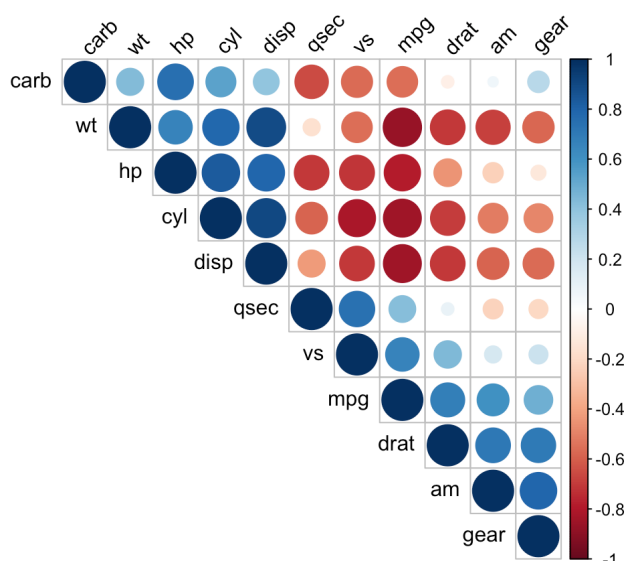
To read more about `corrplot()` function click here : [visualize a correlation matrix using corrplot](#).

The function `corrplot()` takes the correlation matrix as the first argument. The second argument (`type="upper"`) is used to display only the upper triangular of the correlation matrix.

```

# install.packages("corrplot")
library(corrplot)
corrplot(mcor, type="upper", order="hclust", tl.col="black", tl.srt=45)

```



Positive correlations are displayed in blue and negative correlations in red color. Color intensity and the size of the circle are proportional to the correlation coefficients. In the right side of the correlogram, the legend color shows the correlation coefficients and the corresponding colors.

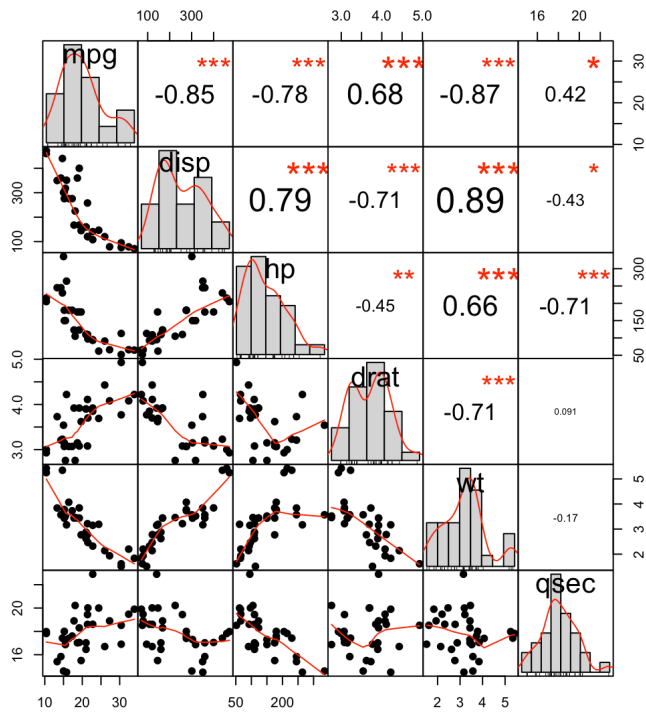
- The correlation matrix is reordered according to the correlation coefficient using “hclust” method.
- `tl.col` (for text label color) and `tl.srt` (for text label string rotation) are used to change text colors and rotations.
- Possible values for the argument type are : “upper”, “lower”, “full”

Read more : [visualize a correlation matrix using corplot.](#)

## Visualize a correlation matrix using scatter plots

The function `chart.Correlation()`, from the package `PerformanceAnalytics`, can be used to display a chart of a correlation matrix. Install the package if you don't have it.

```
library(PerformanceAnalytics)
mydata <- mtcars[, c(1,3,4,5,6,7)]
chart.Correlation(mydata, histogram=TRUE, pch=19)
```



- The distribution of each variable is shown on the diagonal.
- On the bottom of the diagonal : the bivariate scatter plots with a fitted line are displayed
- On the top of the diagonal : the value of the correlation plus the significance level as stars

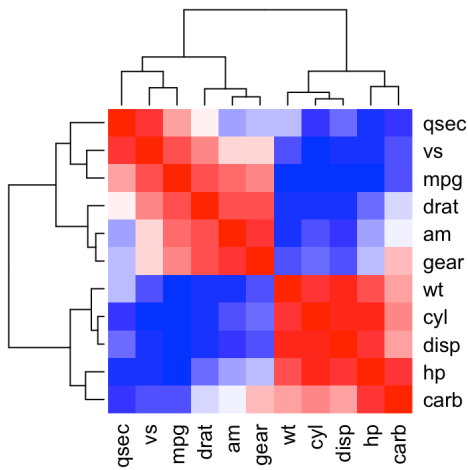
Each significance level is associated to a symbol :

```
p-values(0, 0.001, 0.01, 0.05, 0.1, 1) <=> symbols( "***", "**", "*", ".", " ")
```

## Visualize correlation matrix using a heatmap

```
# Get some colors
col<- colorRampPalette(c("blue", "white", "red"))(20)

heatmap(x = mcor, col = col, symm = TRUE)
```



- x : the correlation matrix to be plotted
- col : color palettes
- symm : logical indicating if x should be treated symmetrically; can only be true when x is a square matrix.

## Online software to analyze and visualize a correlation matrix

A web application for computing and visualizing a correlation matrix is available here without any installation : [online software for correlation matrix](#).





(Designed by Freepik)

Take me to the [correlation matrix calculator](#)

The software can be used as follow :

1. Go to the web application : [correlation matrix calculator](#)
2. Upload a .txt tab or a CSV file containing your data (columns are variables). The supported file formats are described [here](#). You can use the demo data available on the calculator web page by clicking on the corresponding link.
3. After uploading, an overview of a part of your file is shown to check that the data are correctly imported. If the data are not correctly displayed, please make sure that the format of your file is ok [here](#).
4. Click on the 'Analyze' button and select at least 2 variables to calculate the correlation matrix. By default, all variables are selected. Please, deselect the columns containing texts. You can also select the correlation methods (Pearson, Spearman or Kendall). Default is the Pearson method.
5. Click the OK button
6. Results : the output of the software includes :
  - The correlation matrix
  - The visualization of the correlation matrix as a correlogram
  - A web link to export the results as .txt tab file

Note that, you can specify the alternative hypothesis to use for the correlation test by clicking on the button "Advanced options".

Choose one of the 3 options :

- Two-sided
- Correlation < 0 for "less"
- Correlation > 0 for "greater"

Default value is Two-sided.

## conclusions

- Use `cor()` function for simple correlation analysis
- Use `rcorr()` function from `Hmisc` package to compute matrix of correlation coefficient and matrix of p-value in single step.

- Use `symnum()`, `corplot()`, [from `corrplot` package], `chart.Correlation()` [from `PerformanceAnalytics` package], or `heatmap()` functions to visualize a correlation matrix.

## Infos

This analysis was performed using R (ver. 3.1.0).