**Introduction**

Correlation, often computed as part of descriptive statistics, is a statistical tool used to study the relationship between two variables, that is, whether and how strongly couples of variables are associated.

Correlations are measured between only 2 variables at a time. Therefore, for datasets with many variables, computing correlations can become quite cumbersome and time consuming.

**Correlation matrix**

A solution to this problem is to compute correlations and display them in a correlation matrix, which shows correlation coefficients for all possible combinations of two variables in the dataset.

For example, below is the correlation matrix for the dataset mtcars (which, as described by the help documentation of R, comprises fuel consumption and 10 aspects of automobile design and performance for 32 automobiles):[1](https://www.statsandr.com/blog/correlogram-in-r-how-to-highlight-the-most-correlated-variables-in-a-dataset/#fn1)

dat <- mtcars

round(cor(mtcars), 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

## 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

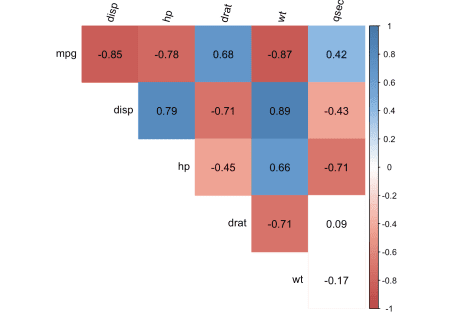
## carb -0.55 0.53 0.39 0.75 -0.09 0.43 -0.66 -0.57 0.06 0.27 1.00

Even after rounding the correlation coefficients to 2 digits, you will conceive that this correlation matrix is not easily and quickly interpretable.

If you are using R Markdown, you can use the pander() function from the {pander} package to make it slightly more readable, but still, we must admit that this table is not optimal when it comes to visualizing correlations between several variables of a dataset, especially for large datasets.

**Correlogram**

To tackle this issue and make it much more insightful, let’s transform the correlation matrix into a correlation plot. A correlation plot, also referred as a correlogram, allows to highlight the variables that are most (positively and negatively) correlated. Below an example with the same dataset presented above:



The correlogram represents the correlations for all pairs of variables. Positive correlations are displayed in blue and negative correlations in red. The intensity of the color is proportional to the correlation coefficient so the stronger the correlation (i.e., the closer to -1 or 1), the darker the boxes. The color legend on the right hand side of the correlogram shows the correlation coefficients and the corresponding colors.

As a reminder, a negative correlation implies that the two variables under consideration vary in opposite directions, that is, if one variable increases the other decreases and vice versa. A positive correlation implies that the two variables under consideration vary in the same direction, that is, if one variable increases the other increases and if one variable decreases the other decreases as well. Furthermore, the stronger the correlation, the stronger the association between the two variables.

**Correlation test**

Finally, a white box in the correlogram indicates that the correlation is not significantly different from 0 at the specified significance level (in this example, at \(\alpha = 5\)%) for the couple of variables. A correlation not significantly different from 0 means that there is **no linear** relationship between the two variables considered (there could be another kind of association, but other than linear).

To determine whether a specific correlation coefficient is significantly different from 0, a correlation test has been performed. Remind that the null and alternative hypotheses of this test are:

* \(H\_0\): \(\rho = 0\)
* \(H\_1\): \(\rho \ne 0\)

where \(\rho\) is the correlation coefficient. The correlation test is based on two factors: the number of observations and the correlation coefficient. The more observations and the stronger the correlation between 2 variables, the more likely it is to reject the null hypothesis of no correlation between these 2 variables.

In the context of our example, the correlogram above shows that the variables cyl (the number of cylinders) and hp (horsepower) are positively correlated, while the variables mpg (miles per gallon) and wt (weight) are negatively correlated (both correlations make sense if we think about it). Furthermore, the variables gear and carb are not correlated. Even if the correlation coefficient is 0.27 between the 2 variables, the correlation test has shown that we cannot reject the hypothesis of no correlation. This is the reason the box for these two variable is white.

Although this correlogram presents exactly the same information than the correlation matrix, the correlogram presents a visual representation of the correlation matrix, allowing to quickly scan through it to see which variables are correlated and which are not.

**Code**

For those interested to draw this correlogram with their own data, here is the code of the function I adapted based on the corrplot() function from the {corrplot} package (thanks again to all contributors of this package):

The main arguments in the corrplot2() function are the following:

* data: name of your dataset
* method: the correlation method to be computed, one of “pearson” (default), “kendall”, or “spearman”. As a rule of thumb, if your dataset contains quantitative continuous variables, you can keep the Pearson method, if you have qualitative ordinal variables, the Spearman method is more appropriate
* sig.level: the significance level for the correlation test, default is 0.05
* order: order of the variables, one of “original” (default), “AOE” (angular order of the eigenvectors), “FPC” (first principal component order), “hclust” (hierarchical clustering order), “alphabet” (alphabetical order)
* diag: display the correlation coefficients on the diagonal? The default is FALSE
* type: display the entire correlation matrix or simply the upper/lower part, one of “upper” (default), “lower”, “full”
* tl.srt: rotation of the variable labels
* (note that missing values in the dataset are automatically removed)

You can also play with the arguments of the corrplot2 function and see the results thanks to this [R Shiny app](https://antoinesoetewey.shinyapps.io/correlogram/).

Thanks for reading. I hope this article will help you to visualize correlations between variables in a dataset and to make correlation matrices more insightful and more appealing.