

I remember that, when I first started using R, I found it quite difficult to generate these correlation matrices automatically.

Yes, there is the `cor` function, but it does not include significance levels.

```
> cor(mtcars[,1:4])
      mpg      cyl      disp      hp
mpg  1.0000000 -0.8521620 -0.8475514 -0.7761684
cyl -0.8521620  1.0000000  0.9020329  0.8324475
disp -0.8475514  0.9020329  1.0000000  0.7909486
hp   -0.7761684  0.8324475  0.7909486  1.0000000
> |
```

Then there the (in)famous `Hmisc` package, with its `rcorr` function. But this tool provides a **whole new range of issues**.

What's this `storage.mode`, and what are we trying to coerce again?

```
> Hmisc::rcorr(mtcars[, 1:4])
Error in storage.mode(x) <- "double" :
  'list' object cannot be coerced to type 'double'
> |
```

Soon you figure out that `Hmisc::rcorr` only takes in matrices (*thus with only numeric values*). **Hurray**, now you can run a correlation analysis on your *dataframe*, you think...

Yet, the output is **all but publication-ready!**

```
> Hmisc::rcorr(as.matrix(mtcars[, 1:4]))
      mpg      cyl      disp      hp
mpg  1.00 -0.85 -0.85 -0.78
cyl -0.85  1.00  0.90  0.83
disp -0.85  0.90  1.00  0.79
hp   -0.78  0.83  0.79  1.00

n= 32

P
      mpg      cyl      disp      hp
mpg      0      0      0
cyl      0      0      0
disp      0      0      0
hp      0      0      0
> |
```

You wanted one correlation matrix, but now you have two... **Double the trouble?**

To spare future scholars the struggle of the early day R programming, I would like to share my *custom function* [correlation_matrix](#).

My [correlation_matrix](#) takes in a *dataframe*, selects only the numeric (and boolean/logical) columns, calculates the correlation coefficients and p-values, and outputs a **fully formatted publication-ready correlation matrix!**

```
> correlation_matrix(iris)
Dropping non-numeric/-boolean column(s): Species

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 "
```

You can specify **many formatting options** in `correlation_matrix`.

For instance, you can use only 2 decimals. You can focus on the lower triangle (*as the lower and upper triangle values are identical*). And you can drop the diagonal values:

```
> correlation_matrix(mtcars[, 1:7], digits = 2, use = 'lower', replace_diagonal = TRUE)

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

Or maybe you are interested in a **different type of correlation coefficients**, and not so much in significance levels:

```
> correlation_matrix(mtcars[, 1:4], type = "spearman", show_significance = FALSE)

mpg      cyl      disp      hp
mpg " 1.000" "-0.852" "-0.848" "-0.776"
cyl "-0.852" " 1.000" " 0.902" " 0.832"
disp "-0.848" " 0.902" " 1.000" " 0.791"
hp "-0.776" " 0.832" " 0.791" " 1.000"
```

For other formatting options, do have a look at the [source code below](#).

Now, to make matters **even more easy**, I wrote a second function (`save_correlation_matrix`) to directly save any created correlation matrices:

```
> save_correlation_matrix(df = iris,
+                           filename = 'iris-correlation-matrix.csv',
+                           digits = 3,
+                           use = 'lower')
Dropping non-numeric/-boolean column(s): Species
```

Once you open your new correlation matrix file in Excel, it is **immediately ready** to be copy-pasted into Word!

The screenshot shows two applications side-by-side. On the left is Microsoft Excel with a file named 'iris-correlation-matrix.csv' open. The spreadsheet displays the correlation matrix for the iris dataset, with columns labeled 'Sepal.Length', 'Sepal.Width', 'Petal.Length', and 'Petal.Width'. The data is as follows:

	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

On the right is Microsoft Word, showing the same data copied from Excel into a table. The table structure and content are identical to the one shown in the Excel screenshot.

If you are looking for ways to **visualize** your correlations do have a look at the packages `corr` and `corrplot`.


```

        replace_diagonal = FALSE,
        replacement = ""){

# check arguments
stopifnot({
  is.numeric(digits)
  digits >= 0
  use %in% c("all", "upper", "lower")
  is.logical(replace_diagonal)
  is.logical(show_significance)
  is.character(replacement)
})
# we need the Hmisc package for this
require(Hmisc)

# retain only numeric and boolean columns
isNumericOrBoolean = vapply(df, function(x) is.numeric(x) | is.logical(x),
logical(1))
if (sum(!isNumericOrBoolean) > 0) {
  cat('Dropping non-numeric/-boolean column(s):', paste(names(
isNumericOrBoolean)[!isNumericOrBoolean], collapse = ', '), '\n\n')
}
df = df[isNumericOrBoolean]

# transform input data frame to matrix
x <- as.matrix(df)

# run correlation analysis using Hmisc package
correlation_matrix <- Hmisc::rcorr(x, type = )
R <- correlation_matrix$r # Matrix of correlation coefficients
p <- correlation_matrix$p # Matrix of p-value

# transform correlations to specific character format
Rformatted = formatC(R, format = 'f', digits = digits, decimal.mark =
decimal.mark)

# if there are any negative numbers, we want to put a space before the
positives to align all
if (sum(R < 0) > 0) {
  Rformatted = ifelse(R > 0, paste0(' ', Rformatted), Rformatted)
}

# add significance levels if desired
if (show_significance) {
  # define notions for significance levels; spacing is important.
  stars <- ifelse(is.na(p), " ", ifelse(p < .001, "****", ifelse(p < .01, "***
", ifelse(p < .05, "** ", " "))))
  Rformatted = paste0(Rformatted, stars)
}
# build a new matrix that includes the formatted correlations and their
significance stars
Rnew <- matrix(Rformatted, ncol = ncol(x))
rownames(Rnew) <- colnames(x)
colnames(Rnew) <- paste(colnames(x), "", sep = " ")

# replace undesired values
if (use == 'upper') {

```

```

    Rnew[lower.tri(Rnew, diag = replace_diagonal)] <- replacement
  } else if (use == 'lower') {
    Rnew[upper.tri(Rnew, diag = replace_diagonal)] <- replacement
  } else if (replace_diagonal) {
    diag(Rnew) <- replacement
  }

  return(Rnew)
}

```

save_correlation_matrix

```

#' save_correlation_matrix
#' Creates and save to file a fully formatted correlation matrix, using
#' `correlation_matrix` and `Hmisc::rcorr` in the backend
#' @param df dataframe; passed to `correlation_matrix`
#' @param filename either a character string naming a file or a connection open
#' for writing. "" indicates output to the console; passed to `write.csv`
#' @param ... any other arguments passed to `correlation_matrix`
#'
#' @return NULL
#'
#' @examples
#' `save_correlation_matrix(df = iris, filename = 'iris-correlation-matrix.csv')`
`
#' `save_correlation_matrix(df = mtcars, filename = 'mtcars-correlation-matrix.
#' csv', digits = 3, use = 'lower')`
save_correlation_matrix = function(df, filename, ...) {
  write.csv2(correlation_matrix(df, ...), file = filename)
}...

```