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])
                                   disp
                        cyl
            mpa
      1.0000000
                 -0.8521620
                            -0.8475514
                                        -0.7761684
     -0.8521620
                  1.0000000
                             0.9020329
                                         0.8324475
disp -0.8475514
                  0.9020329
                             1.0000000
                                         0.7909486
     -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!

```
matrix(mtcars[, 1:4]))
  Hmisc::rcorr(as
                   disp
              cyl
                             hp
       mpg
      1.00 - 0.85
                  -0.85
                         -0.78
                   0.90
     -0.85
               00
             0.90
disp -0.85
                    1.00
     -0.78
            0.83
n=32
         cyl disp hp
                     0
               0
      0
               0
                     0
           0
                     0
      0
               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**!

```
Dropping non-numeric/-boolean column(s): Species
                            Sepal.Width
                                         Petal.Length
             Sepal.Length
                                                        Petal.Width
Sepal.Length
               1.000
                              -0.118
                                           0.872***
                                                          0.818***
                                          "-0.428***"
                                                        "-0.366***"
Sepal.Width
                              1.000
               0.872***"
                            "-0.428***"
                                                          0.963***"
Petal.Length
                                           1.000
               0.818***"
                            "-0.366***"
                                           0.963***"
                                                          1.000
Petal.Width
```

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:

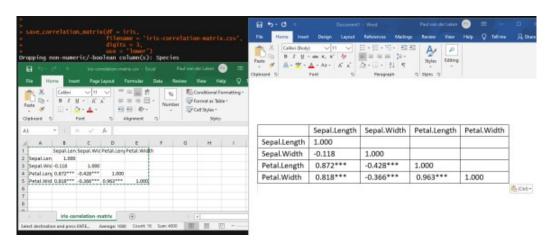
Or maybe you are interested in a **different type of correlation coefficients**, and not so much in significance levels:

For other formatting options, do have a look at the source code below.

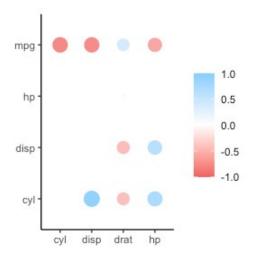
Now, to make matters **even more easy**, I wrote a second function (save_correlation_matrix) to directly save any created correlation matrices:

```
> 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!



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



I hope my functions are of help to you!

Do reach out if you get to use them in any of your research papers!

I would be super interested and feel honored.

correlation matrix

```
#' correlation matrix
#' Creates a publication-ready / formatted correlation matrix, using
`Hmisc::rcorr` in the backend.
#' @param df dataframe; containing numeric and/or logical columns to calculate
correlations for
#' @param type character; specifies the type of correlations to compute; gets
passed to `Hmisc::rcorr`; options are `"pearson"` or `"spearman"`; defaults to
#' @param digits integer/double; number of decimals to show in the correlation
matrix; gets passed to `formatC`; defaults to `3`
#' @param decimal.mark character; which decimal.mark to use; gets passed to
`formatC`; defaults to `.`
#' @param use character; which part of the correlation matrix to display;
options are `"all"`, `"upper"`, `"lower"`; defaults to `"all"`
#' @param show significance boolean; whether to add `*` to represent the
significance levels for the correlations; defaults to `TRUE`
#' @param replace diagonal boolean; whether to replace the correlations on the
diagonal; defaults to `FALSE`
#' @param replacement character; what to replace the diagonal and/or upper/lower
triangles with; defaults to `""` (empty string)
#' @return a correlation matrix
#' @export
# •
#' @examples
#' `correlation matrix(iris)`
#' `correlation_matrix(mtcars)`
correlation matrix <- function(df,</pre>
                               type = "pearson",
                               digits = 3,
                               decimal.mark = ".",
                               use = "all",
                               show significance = TRUE,
```

```
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 = )</pre>
  R <- correlation matrix$r # Matrix of correlation coeficients
  p <- correlation matrix$P # Matrix of p-value</pre>
  # 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))</pre>
  rownames(Rnew) <- colnames(x)</pre>
  colnames(Rnew) <- paste(colnames(x), "", sep =" ")
  # replace undesired values
  if (use == 'upper') {
```

replace diagonal = FALSE,

```
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)
}</pre>
```

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)
} ....
```