

# corVis: An R Package for Visualising Associations and Conditional Associations

by Amit Chinwan and Catherine Hurley

**Abstract** Correlation matrix displays are important tools to explore multivariate datasets. These displays with other measures of association can summarize interesting patterns to an analyst and assist them in framing questions while performing exploratory data analysis. In this paper, we present new visualisation techniques to visualise association between all the variable pairs in a dataset in a single plot, which is something existing displays lack. Also, we propose new methods to visualise relationship among variable pairs using conditioning. We use different layouts like matrix or linear for our displays. We use seriation in our displays which helps in highlighting interesting patterns easily. The R package corVis provides an implementation.

## Section 1: Introduction

Correlation matrix display is a popular tool to visually explore correlations among variables while performing Exploratory Data Analysis (EDA) on a multivariate dataset. Popularized by [Friendly \(2002\)](#) as corgram, these displays are produced by first calculating the correlation among the variables and then plotting these calculated values in a matrix display. With effective ordering techniques, these displays quickly highlight variables which are highly correlated and an analyst interested in building a predictive model could use these displays to remove correlated variables and avoid multicollinearity.

The correlation displays are generally used with one of the Pearson's, Spearman's or Kendall's correlation coefficient and are therefore limited to quantitative variables. An analyst can use one-hot encoding of the qualitative variables in order to use these displays but will need to deal with the high dimensions as a result of the encoding. In addition to the dimensionality problem, it is not easy to assess the overall correlation when using the one-hot encoding. The existing methods to quickly explore association among qualitative variables in a dataset include using proportions or counts with different graphical displays like boxplots or barplots. Using association measures for qualitative pairs similar to correlation for quantitative pairs will help in summarizing the relationship, which then can be displayed like the correlation displays.

Tukey and Tukey introduced scagnostics which are measures for scatterplots ([Tukey and Tukey, 1985](#)). Along with scagnostics, they proposed a scagnostics scatterplot matrix which is a visual display to explore and compare these measures for all the variable pairs in a dataset. By comparing multiple measures at once, the unusual variable pairs could be identified and looked at in more detail. In a similar manner, a display comparing association measures will help in finding interesting variable pairs. Many association measures have been proposed to summarize different types of relationships. The most commonly used measure is Pearson's correlation coefficient which captures any linear trend present between the variables. Other popular measures include Kendall's or Spearman's rank correlation coefficient which are non-parametric measures and looks for monotonic relationship. Distance correlation ([Székely et al., 2007](#)) is an important measure useful in exploring non-linear relationships. The information theory measure maximal information coefficient (MIC) ([Reshef et al., 2011](#)) is capable of summarizing complex relationships. With effective displaying techniques, the multiple measures of association provide a comparison tool that assist an analyst to reveal structure present in the data.

Small multiples (or Trellis display) is a simple yet powerful approach to compare partitions of data and understand multidimensional datasets ([Tufte, 1986](#)). The display is produced by splitting the data into groups by a conditioning variable and then plotting the data for each group. Such displays allow analysts to quickly infer about the impact of the conditioning variable. A similar idea applied to displays of association measures (correlation plot) will help uncover underlying patterns in the data. One such pattern is Simpson's paradox which can be detected by comparing Pearson's correlation for data at overall level versus individual levels of the conditioning variable.

In this paper, we propose extensions of the correlation plot and new visualizations which look at variables of mixed type, multiple association measures and conditional associations. These displays are implemented in the R package [corVis](#). The next section provides a review of existing packages which deal with correlation displays and a quick background on association measures and the packages used for calculating them. Then we describe our approach to calculate the association measures, followed by visualizations of associations and conditional associations. We conclude with a summary and future work.

## Section 2: Background

In this section we provide a brief review of existing packages used for correlation displays and association measure calculation.

### Section 2.1: Literature Review on Correlation Displays

According to [Hills \(1969\)](#), “the first and sometimes only impression gained by looking at a large correlation matrix is its largeness”. To overcome this, [Murdoch and Chow \(1996\)](#) proposed a display for large correlation matrices which uses a matrix layout of ellipses where the parameters of the ellipses are scaled to the correlation values. [Friendly \(2002\)](#) expanded on this idea by rendering correlation values as shaded squares, bars, ellipses, or circular ‘pac-man’ symbols. The variables in the matrix displays were optionally ordered using the angular ordering of the first two eigen vectors of the correlation matrix. The ordering places highly-correlated pairs of variables nearby, making it easier to quickly identify groups of variables with high mutual correlation.

Nowadays, there are many R packages devoted to correlation visualisation. Table 1 provides a summary, listing the displays offered, and whether these extend to factor variables or mixed numeric-factor pairs.

The R package [corrplot](#) ([Wei and Simko, 2021](#)) provides an implementation of the methods in [Friendly \(2002\)](#). The package [corrr](#) ([Kuhn et al., 2020](#)) organises correlations as tidy data, so leveraging the data manipulation and visualisation tools of the [tidyverse](#) ([Wickham et al., 2019](#)). In addition to various matrix displays, the package offers network displays where line-thickness encodes correlation magnitude, with a filtering option to discard low-correlation edges.

The package [corrgrapher](#) ([Morgen and Biecek, 2020](#)) uses a network plot for exploring correlations, where the nodes close to each other have high correlation magnitude, edge thickness encodes the absolute correlation value and edge color indicates the sign of correlation. The package also handles mixed type variables by using association measures obtained as transformations of  $p$ -values obtained from Pearson’s correlation test in the case of two numeric variables, Kruskal’s test for numerical and factor variables, and a chi-squared test for two categorical variables.

The package [linkspotter](#) ([Samba, 2020](#)) offers a variety of association measures (distance correlation, MIC, maximum normalized mutual information) in addition to correlation, where the measure used depends on whether the variables are both numerical, categorical or mixed. The results are visualized in a network plot, which may be packaged into an interactive shiny application.

Our own package [corVis](#) offers a variety of displays, and has new features not available elsewhere, in particular simultaneous display of multiple association measures, and association displays stratified by levels of a grouping variable. This will be described in the following sections.

There have been other extensions to correlation displays which are useful when dealing with high dimensional datasets. [Hills \(1969\)](#) proposed a QQ plot of the  $z$ -transform of the entries of the correlation matrix to discover correlation coefficients too large to come from a normal distribution with mean zero. [Buja et al. \(2016\)](#) proposed Association Navigator which is an interactive visualization tool for large correlation matrices with upto 2000 variables. The R package [scorrplot](#) ([Gerber, 2022](#)) produces an interactive scatterplot for exploring pairwise correlations in a large dataset by projecting variables as points and encoding the correlations as space between these points. The package provides a functionality to update variable of interest which creates tour of the correlation space between different projections of the data.

The R package [correlationfunnel](#) offers a novel display which assists in feature selection in a setting with a single response and many predictor variables. All numeric variables including the response are binned. All (now categorical) variables in the resulting dataset are one-hot encoded and Pearson’s correlation calculated with the response categories. The correlations are visualised in a dot-plot display, where predictors are ordered by maximum correlation magnitude. Correlations between one-hot encoded variables are challenging to interpret, especially as the number of levels increase. In [corVis](#) we offer a similar dot-plot display, but showing multiple correlation or association measures, or alternatively measures stratified by a grouping variable.

### Section 2.2: Literature Review on Association Measures

An association measure can be defined as a numerical summary quantifying the relationship between two or more variables. For example, Pearson’s correlation coefficient summarizes the strength and direction of the linear relationship present between two numeric variables and is in the range  $[-1, 1]$ . Kendall’s or Spearman’s rank correlation coefficient are other popular measures which assess monotonic relationship among two numeric variables and are in the range  $[-1, 1]$ . As these measures are limited

**Table 1:** List of the R packages dealing with correlation or correlation displays with information on whether the plots display multiple measures, conditional display of measures and mixed variables in a single plot

Package	Display	MixedVariables
corrplot	heatmap	
corr	heatmap/network	
corrgrapher	network	
linkspotter	network	Yes
correlation	heatmap/network	
corVis	heatmap/matrix/linear	Yes

to linear or monotonic relationships, there's a need to use association measures which are able to capture complex relationships. In addition to association measures for numeric variables, association measures for ordinal, nominal and mixed variable pairs are useful in exploring a multivariate dataset. We now give an overview of available association measures.

For a pair of numeric variables, various measures of association have been proposed in literature. The distance correlation coefficient (Székely et al., 2007) is an association measure which looks for the non-linear association between two numeric variables and summarizes it in  $[0, 1]$ . Similarly, MIC (Reshef et al., 2011) is capable of summarizing non-linear as well as periodic relationships between numeric variables and is in range  $[0, 1]$ .

Agresti (2010) provides an overview of the association measures which are used for exploring association between ordinal variables. Kendall's tau-b (Kendall, 1945) is an association measure useful in summarizing the relationship between two ordinal variables in the range  $[-1, 1]$ . It is a relatively stable measure than Goodman and Kruskal's gamma with respect to the changes in categories of any variable i.e. if two categories are merged to make a single category. The polychoric correlation (Olsson, 1979) measures the correlation between two ordinal variables by assuming two normally distributed latent variables for a contingency table of two ordinal variables and summarizes the association in  $[-1, 1]$ .

The association measures for the case of nominal pair of variables should be invariant to the order in which the categories appear. Pearson's contingency coefficient uses the  $\chi^2$  value from the Pearson's  $\chi^2$  test for independence and is a useful measure to summarize the association between two nominal variables in  $[0, 1]$ . Another measure for nominal variable pair is the Uncertainty coefficient (Theil, 1970) measuring the proportion of uncertainty in one variable which is explained by the other variable.

The canonical correlation is a measure useful in exploring association among mixed variables. The goal of the canonical correlation analysis is to maximize the association between the low-dimensional projections of two sets of variables (Härdle and Simar, 2019). Each of these measures are consistent with respect to the order of the categories of the nominal variable.

### Section 3: Introducing corVis

**corVis** is an R package which calculates measures of association for every variable pair in a dataset and helps in visualising these associations in different ways. Most of the existing correlation displays are limited to numeric pairs of variables. This package extends these displays to every variable pair.

The package offers new visualisation technique such as display with multiple measures for the association for every variable pair in the dataset. We also introduce conditional association displays which are useful in uncovering conditional structure present in the data and are produced by splitting the data by a partitioning variable, calculating association for the variable pairs at every level of partitioning variable and then plotting them. These displays are efficient for discovering differences among the groups in the data.

Efficient seriation techniques have been included to order and highlight variables with high value for an association measure. These ordered association and conditional association displays are useful in finding interesting patterns, such as Simpson's Paradox. These new displays also help an analyst to quickly discover any unusual variable pair(s) in the dataset.

Table 2 provides a list of the functions available in the package which are useful for calculating association measures among variable pairs and visualising these associations using novel displays. The function `calc_assoc` calculates the association measures for variable pairs in a dataset and is described in detail in Section 4. The remaining functions in table 2 are useful for displaying pairwise

**Table 2:** List of the available functions in corVis package with input arguments and outputs

Function	Usage	Input
<b>calc_assoc</b>	Calculation	Dataframe Types of association measures NA handler Name of the grouping variable Include overall value or not
<b>association_heatmap</b>	Visualization	Measures dataframe for lower/upper triangle Variable order Limits of the scale
<b>pairwise_2d_plot</b>	Visualization	Measures dataframe for lower/upper triangle Name of the grouping variable fill variable Variable order Limits of the scale
<b>pairwise_1d_compare</b>	Visualization	Dataframe Measures for the display Variable order Limits of the scale
<b>pairwise_1d_plot</b>	Visualization	Measures dataframe Name of the grouping variable for display fill variable Variable order Limits of the scale

association and their usage has been illustrated in Section 5.

## Section 4: corVis: Calculating Association

This section describes the calculation of association measures in our package **corVis**. The package provides a collection of various measures of association which quantifies the relationship between two variables. The association measures available in the package are not limited to numeric variables and are used with nominal, ordinal and mixed variable pairs as well. Table 3 lists different functions provided in the package to calculate various measures of association. The funName column represents the function name used to calculate measure(s) of associations in this package. The typeX and typeY columns provide the information on types of variables which can be used with the corresponding functions. The X or Y variable is one of the numeric, nominal, ordinal or any type. The from column corresponds to the package functions used to calculate the association measures by the function under funName. The symmetric column represents if the measure is symmetric i.e. if the value of measure is same regardless of the order of variables. The last column provides the range of values for these measures. The function tbl\_easy calculates association measures available in the R package **correlation** and is suitable for different variable types. The functions in Table 3 with corVis entries under from column calculate the association measures which have been implemented in this package.

For numeric pairs of variables, this package provides a range of association measures. The popular correlation coefficients like Pearson's or Spearman's or Kendall's are calculated using tbl\_cor function. The measures such as distance correlation or MIC which assess more complex relationship are calculated using tbl\_dcor or tbl\_mine respectively. The association measures available in the package for the ordinal pairs of variables are polychoric correlation and Kendall's coefficients which are calculated using tbl\_polycor or tbl\_tau respectively. For nominal pairs of variables, the functions like tbl\_gkTau, tbl\_gkGamma, tbl\_uncertainty, tbl\_chi, tbl\_cancor are used for exploring association among the variables.

The function tbl\_cancor calculates a measure of association based on canonical correlations for mixed pairs of variables. Nominal variables are converted into sets of dummy variables, which are then assigned scored to find the maximal correlation. For two numeric variables this measure is identical to absolute correlation, for two factors the correlation is identical to that obtained from correspondence analysis.

**Table 3:** List of the functions available in the package for calculating different association measures along with the packages used for calculation.

funName	typeX	typeY	from	symmetr
tbl_cor	numerical	numerical	stats::cor	TRUE
tbl_dcor	numerical	numerical	energy::dcor2d	TRUE
tbl_mine	numerical	numerical	minerva::mine	TRUE
tbl_polycor	ordinal	ordinal	polycor::polychor	TRUE
tbl_tau	ordinal	ordinal	DescTools::KendalTauA,B,C,W	TRUE
tbl_gkTau	nominal	nominal	DescTools::GoodmanKruskalTau	FALSE
tbl_gkGamma	nominal	nominal	DescTools::GoodmanKruskalGamma	TRUE
tbl_uncertainty	nominal	nominal	DescTools::UncertCoef	TRUE
tbl_chi	nominal	nominal	DescTools::ContCoef	TRUE
tbl_cancor	nominal/numerical	nominal/numerical	corVis	TRUE
tbl_nmi	any	any	corVis	TRUE
tbl_easy	any	any	correlation::correlation	TRUE

The functions listed in 3 for calculating association measures provide a functionality for handling missing value or NA in the dataset. Each of these functions either have a `handle.na` argument or have package functions which automatically uses pairwise complete observations for taking care of missing values present in the data.

### Calculating association for a single type of variable pairs

We introduce a method which creates a tibble structure for the variable pairs in a dataset along with calculated association measure. The package contains various functions (shown in Table 3) for different association measures in the form `tbl_*` to calculate them. For example, someone interested in calculating distance correlation for numeric pair of variables in a dataset is done by using `tbl_dcor`.

```
df <- penguins
distance <- tbl_dcor(df)
distance

#> # A tibble: 10 x 4
#>   x           y      measure measure_type
#>   <chr>      <chr>      <dbl> <chr>
#> 1 bill_depth_mm bill_length_mm 0.387 dcor
#> 2 flipper_length_mm bill_length_mm 0.666 dcor
#> 3 body_mass_g      bill_length_mm 0.587 dcor
#> 4 year             bill_length_mm 0.0784 dcor
#> 5 flipper_length_mm bill_depth_mm 0.704 dcor
#> 6 body_mass_g      bill_depth_mm 0.614 dcor
#> 7 year             bill_depth_mm 0.112 dcor
#> 8 body_mass_g      flipper_length_mm 0.867 dcor
#> 9 year             flipper_length_mm 0.164 dcor
#> 10 year            body_mass_g      0.0791 dcor
```

Similarly, `tbl_nmi` is used to calculate normalised mutual information for numeric, nominal and mixed pair of variables.

```
nmi <- tbl_nmi(df)
nmi

#> # A tibble: 28 x 4
#>   x           y      measure measure_type
#>   <chr>      <chr>      <dbl> <chr>
#> 1 island      species 0.507      nmi
#> 2 bill_length_mm species 0.353      nmi
#> 3 bill_depth_mm species 0.315      nmi
#> 4 flipper_length_mm species 0.343      nmi
#> 5 body_mass_g      species 0.300      nmi
```

```
#> 6 sex                species 0.0000854 nmi
#> 7 year                species 0.00441 nmi
#> 8 bill_length_mm      island 0.248 nmi
#> 9 bill_depth_mm      island 0.230 nmi
#> 10 flipper_length_mm island 0.217 nmi
#> # ... with 18 more rows
```

The tibble output for the functions mentioned in Table 3 has the following structure:

- x and y representing a pair of variables
- measure representing the calculated value for association measure
- measure\_type representing the association measure calculated for x and y pair.

### Calculating association measures for whole dataset

calc\_assoc is used to calculate association measures for all the variable pairs in the dataset at once in a tibble structure. The variable pairs in the output are unique pairs and a subset of all the pairs of variables in a dataset where  $x \neq y$ . Because of the tidy structure of the output, the data manipulation and visualisation tools of [tidyverse](#) (Wickham et al., 2019) are applicable to and are useful for further exploration of pairwise associations. In addition to tibble structure, the output also has pairwise and data.frame class which are important class attributes for producing visual summaries in this package.

The function calc\_assoc has a types argument which is basically a tibble of the association measure to be calculated for different variable pairs. The default tibble of measures is default\_assoc() which calculates Pearson's correlation if both the variables are numeric, Kendall's tau-b if both the variables are ordinal, canonical correlation if one is factor and other is numeric and canonical correlation for the rest of the variable pairs.

```
default_measures <- default_assoc()
default_measures

#> # A tibble: 4 x 4
#>   funName   typeX   typeY   argList
#>   <chr>     <chr>  <chr>  <list>
#> 1 tbl_cor   numeric numeric <NULL>
#> 2 tbl_tau   ordered ordered <NULL>
#> 3 tbl_cancor factor  numeric <NULL>
#> 4 tbl_cancor other   other   <NULL>

penguin_assoc <- calc_assoc(df, types = default_assoc())
penguin_assoc

#> # A tibble: 28 x 4
#>   x           y      measure measure_type
#>   <chr>      <chr>    <dbl> <chr>
#> 1 island     species  0.813  cancel
#> 2 bill_length_mm species  0.841  cancel
#> 3 bill_depth_mm species  0.824  cancel
#> 4 flipper_length_mm species 0.882  cancel
#> 5 body_mass_g species  0.818  cancel
#> 6 sex        species  0.0121 cancel
#> 7 year       species  0.0511 cancel
#> 8 bill_length_mm island   0.392  cancel
#> 9 bill_depth_mm island   0.632  cancel
#> 10 flipper_length_mm island  0.613  cancel
#> # ... with 18 more rows

class(penguin_assoc)

#> [1] "pairwise" "tbl_df" "tbl" "data.frame"
```

The default tibble of measures is updated using the update\_assoc function which has arguments for updating the tbl\_\* functions to calculate association measures depending on the type variable pair in the dataset and a method for tbl\_\* functions which calculates more than one measure. The update\_assoc function has an argument default which has the default\_assoc() tibble as its default value and is useful when tbl\_\* functions need to be updated for a few types of variable pairs.



```

updated_assoc <- update_assoc(default=default_assoc(),
                              num_pair = "tbl_cor",
                              num_pair_argList = "spearman",
                              mixed_pair = "tbl_cancor",
                              other_pair = "tbl_nmi")

updated_assoc

#> # A tibble: 4 x 4
#>   funName   typeX   typeY   argList
#>   <chr>     <chr>   <chr>   <list>
#> 1 tbl_cor   numeric numeric <chr [1]>
#> 2 tbl_tau   ordered ordered <NULL>
#> 3 tbl_cancor factor  numeric <NULL>
#> 4 tbl_nmi   other   other   <NULL>

```

calc\_assoc also has a handle.na argument for handling the NA or missing values which is fed into the tbl\_\* functions used with the types argument for different types of variable pairs. The default value is set to TRUE for using pairwise complete observations for calculating a measure of association between two variables.

If a user is interested in calculating multiple association measures for a type of variable pair, it can be done by using the calc\_assoc and update\_assoc together for calculating different association measures and then merging the output tibbles.

```

updated_penguin_assoc <- calc_assoc(df, types = updated_assoc)
updated_penguin_assoc

#> # A tibble: 28 x 4
#>       x                y      measure measure_type
#>   <chr>          <chr>    <dbl>   <chr>
#> 1 island                species 0.507     nmi
#> 2 bill_length_mm        species 0.841     cancel
#> 3 bill_depth_mm         species 0.824     cancel
#> 4 flipper_length_mm     species 0.882     cancel
#> 5 body_mass_g           species 0.818     cancel
#> 6 sex                  species 0.0000854 nmi
#> 7 year                  species 0.0511     cancel
#> 8 bill_length_mm        island  0.392     cancel
#> 9 bill_depth_mm         island  0.632     cancel
#> 10 flipper_length_mm    island  0.613     cancel
#> # ... with 18 more rows

```

## Calculating conditional association

calc\_assoc is also used to calculate association measures for all the variable pairs at different levels of a categorical variable. This helps in exploring the conditional associations and find out the differences between the groups of the conditioning variable. The function has a by argument which is used as the grouping variable and needs to be categorical.

```

penguin_assoc_by <- calc_assoc_by(df, by = "sex")
penguin_assoc_by

#> # A tibble: 84 x 5
#>       x                y      measure measure_type by
#>   <chr>          <chr>    <dbl>   <chr>   <fct>
#> 1 island                species 0.809     cancel  female
#> 2 bill_length_mm        species 0.885     cancel  female
#> 3 bill_depth_mm         species 0.900     cancel  female
#> 4 flipper_length_mm     species 0.914     cancel  female
#> 5 body_mass_g           species 0.911     cancel  female
#> 6 year                  species 0.0457    cancel  female
#> 7 bill_length_mm        island  0.395     cancel  female
#> 8 bill_depth_mm         island  0.669     cancel  female
#> 9 flipper_length_mm    island  0.649     cancel  female
#> 10 body_mass_g          island  0.668     cancel  female
#> # ... with 74 more rows

```

**Table 4:** Variable description of the penguins dataset along with the types of variables

Variable	Description	VariableType
species	Penguin species (Adelie, Chinstrap or Gentoo)	nominal
island	Islands in Palmer Archipelago, Antarctica (Biscoe, Dream or Torgersen)	nominal
bill_length_mm	Bill length of penguin (millimeters)	numerical
bill_depth_mm	Bill length of penguin (millimeters)	numerical
flipper_length_mm	Flipper length of penguin (millimeters)	numerical
body_mass_g	Body mass of penguin (grams)	numerical
sex	Sex of the penguin (female,male)	nominal
year	Year of the study	numerical

By default, the function `calc_assoc` calculates the association measures for all the variable pairs at different levels of the grouping variable and the pairwise association measures for the ungrouped data (overall) when used with the `by` argument. This behavior can be changed by setting `include.overall` argument to `FALSE`.

```
penguin_assoc_by <- calc_assoc_by(df, by = "sex", include.overall = FALSE)
```

The tibble output in the conditional setting has a similar structure as `calc_assoc` used with no `by` argument. When used with the `by` argument, an additional `by` column representing the levels of the categorical variable is added in the tibble output. The `x` and `y` variables in the output are repeated for every level of `by` variable. In order to have multiple `by` variables, the function `calc_assoc` is used multiple times with a different `by` variable each time and then the multiple outputs are binded row wise. For calculating multiple measures for a specific variable type, one can use `update_assoc` with `calc_assoc` and then can merge these multiple tibble outputs.

## Section 5: corVis: Visualising Association

We propose novel visualisations to display association and conditional association for every variable pair in a dataset in a single plot and show multiple bivariate measures of association simultaneously to find out interesting patterns. Efficient seriation techniques have also been included to order and highlight interesting relationships. These ordered association and conditional association displays help find interesting patterns in the dataset.

While designing these displays we considered matrix-type, linear and network-based layouts. A matrix-type layout simplifies the effort in finding variables, and different measures may be displayed on the upper and lower diagonal. Linear layouts are more space-efficient than matrix plots, but looking for variables is more challenging. Variable pairs are usually ordered in the linear layouts by relevance (usually difference in measures of association or across the factor levels) ,and it is easier to omit less relevant pairs.

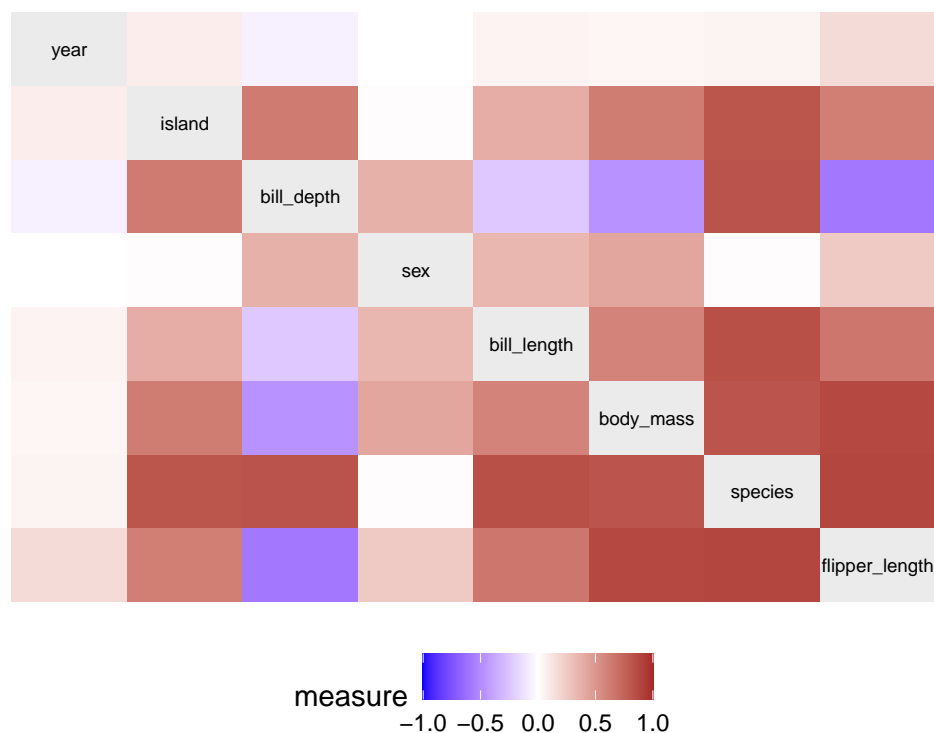
The penguins dataset from the [palmerpenguins](#) package has been used to provide illustrative examples. Table 4 provides a brief description of the variables in this dataset.

### Association Matrix plot

The function `association_heatmap` is used to display a matrix layout with association for variable pairs in the dataset. The display is similar to existing correlation matrix plots but with every variable pair in the dataset. This function `association_heatmap` takes the calculated measures of association by `calc_assoc` function as input and outputs a matrix display by rendering the magnitude of association measures with a color. The function has `lassoc` and `uassoc` arguments for a tibble of association measures for the lower triangle and the upper triangle of the matrix display respectively. The `uassoc` argument is `NULL` by default and uses the same tibble input as used by `lassoc` if not changed. The argument `var_order` is used for ordering or seriating the matrix display such that highly-associated variables are placed nearby and are easier to identify. The function also has a `limits` argument specifying the limit of the color scale.

Figure 1 shows an example of the association matrix display for every variable pair in the penguins dataset. The cells along the diagonal of the matrix display show the variables present in the dataset. Every off diagonal cell is colored using a divergent color scale with limits  $[-1, 1]$  representing the





**Figure 1:** Association matrix display for penguins data showing Pearson's correlation for numeric variable pairs, canonical correlation for mixed variable pairs and categorical variable pairs.

value of association measure between two variables. The plot shows Pearson's correlation for the numeric pair of variables and canonical correlation for mixed and nominal variable pairs.

It shows a high positive Pearson's correlation among flipper\_length and body\_mass, flipper\_length and bill\_length, and bill\_length and bodymass. There seems to be a strong negative Pearson's correlation between flipper\_length and bill\_depth, and bill\_depth and body\_mass.

The plot also shows that there is a low canonical correlation between species and sex, and island and sex of the penguins suggesting low association, which traditional correlation matrix display would omit as they are limited to numeric variable pairs only. There is a high canonical correlation between island and species suggesting a stronger association among these two variables.

The high canonical correlation between bill\_length and species, body\_mass and species, and flipper\_length and species suggests that the body measurements of a penguin depends on their species. The variables in the display are ordered using average linkage clustering method to find out highly associated variables quickly.

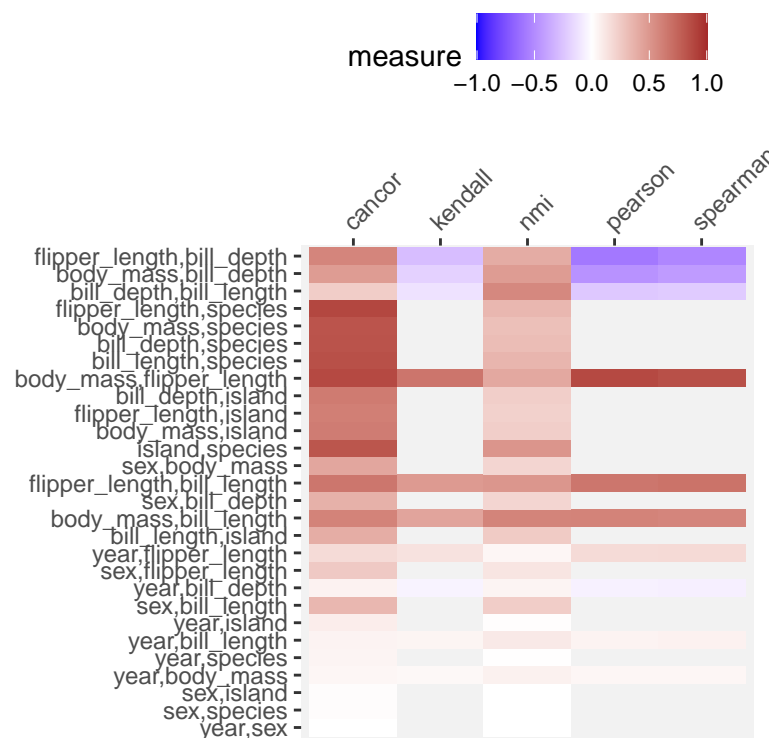
## Multiple Association Measures Plot

We also calculate multiple association measures for all the variable pairs in the dataset and compare them. This helps in identifying pairs of variables with a high difference among different measures or any unusual variable pairs which should be investigated further in more detail.

The function `pairwise_1d_compare` is used to display a linear layout for comparing multiple association measures calculated for variable pairs in the dataset. `pairwise_1d_compare` takes a dataset and a set of association measures as input and outputs a heatmap display by plotting all the variable pairs in the dataset on the Y-axis and the association measures on the X-axis and encoding the magnitude of association measures with a color.

The `df` argument of the function `pairwise_1d_compare` is used for a dataset. The `measures` argument is for the multiple association measures to be calculated for pairs of variables of `df`. The default measures calculated by the function are Pearson's, Spearman's and Kendall's correlation coefficient, canonical correlation and normalized mutual information. The function also come up with a `var_order` argument which is set to `max_diff` by default and orders the variable pairs on the basis of maximum difference among the magnitude of measures.

Figure 2 shows a linear layout comparing multiple association measures for all the variable pairs in the penguins data. Linear layouts are more suitable when comparing high number of association



**Figure 2:** Comparing multiple association measures using a linear layout. The display has variable pairs on the Y-axis and association measures on the X-axis. The cell corresponding to a variable pair and an association measure has been colored grey showing that the measure is not defined for corresponding pair.

measures.

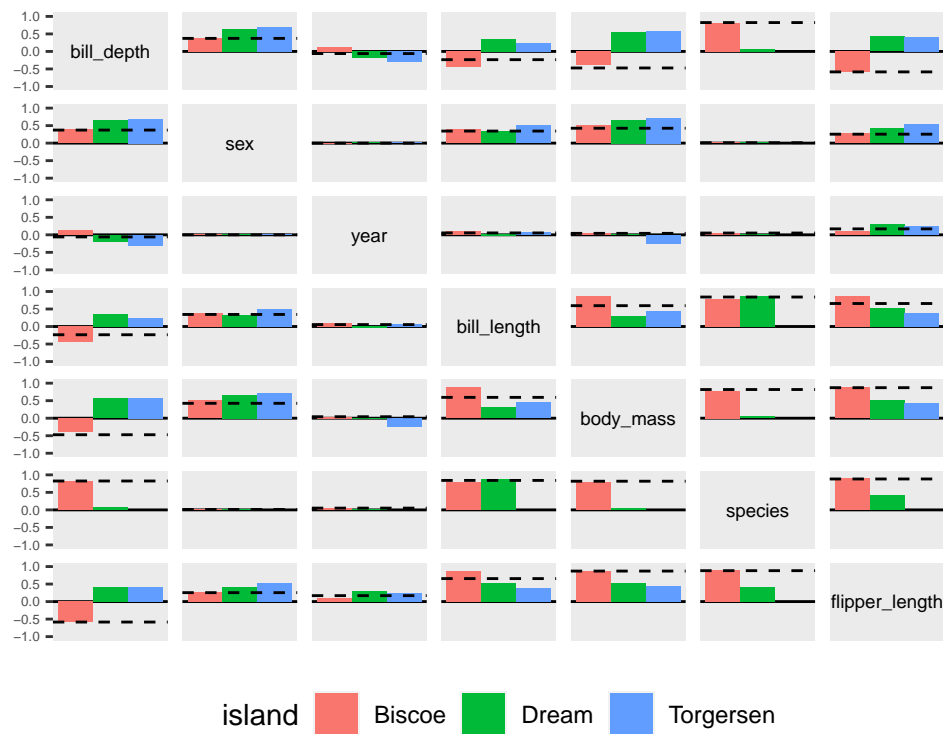
Conditional Association Plot

The function `pairwise_2d_plot` is used to display a matrix layout of the conditional association for variable pairs in the dataset. The display is produced by splitting the data by a partitioning variable and calculating association for the variable pairs at each level of partitioning variable using `calc_assoc` function with conditioning variable as the `by` argument. The calculated association measures are then displayed using bars in a matrix plot. The height and color of the bars are coded with the value of association measure and the level of the partitioning variable respectively. These displays are efficient for discovering variable pair with high differences among the levels of partitioning variable in the data.

The measures of association calculated for every variable pair at every level of conditioning serve as input to the `pairwise_2d_plot` function. The `lassoc` and `uassoc` arguments look for a tibble of association measures for the lower triangle and the upper triangle of the matrix display respectively. The `uassoc` argument is set to `NULL` by default and uses the same tibble input as used by `lassoc` if not updated. The argument `group_var` is responsible for the grouping variable when plotting the bars. The default value `by` uses the conditioning variable and `measure_type` is useful for displaying bars with height and color of the bars coded by the value of association measure and the type of association measure respectively.

Figure 3 shows a conditional association plot for the penguins data. Each cell corresponding to a variable pair shows three bars which correspond to the association measure (Pearson’s correlation for numeric pair and Normalized mutual information for other combination of variables) calculated at the levels of conditioning variable `island`. The dashed line represents the overall association measure. The plot shows that there is a high value for normalised mutual information between `bill_length_mm` and `species` for the penguins which lived in Biscoe island compared to the penguins which lived in Dream island. It can also be seen that the cell corresponding to variable pair `flipper_length_mm` and `bill_depth_mm` has a high negative overall Pearson’s correlation and for the penguins which lived in Biscoe island but positive correlation for penguins which lived in Dream and Torgersen island. This is an instance of Simpson’s paradox which can be taken into account during the modeling step.

We also use linear layouts for displaying conditional association in the package. The function



**Figure 3:** Conditional Association plot for penguins data showing Pearson's correlation for numeric pairs and normalised mutual information for categorical or mixed pairs. The bars in each cell represent the value for association measure colored by the conditioning variable 'island'. The dashed line in each cell represents overall value of the association measure.

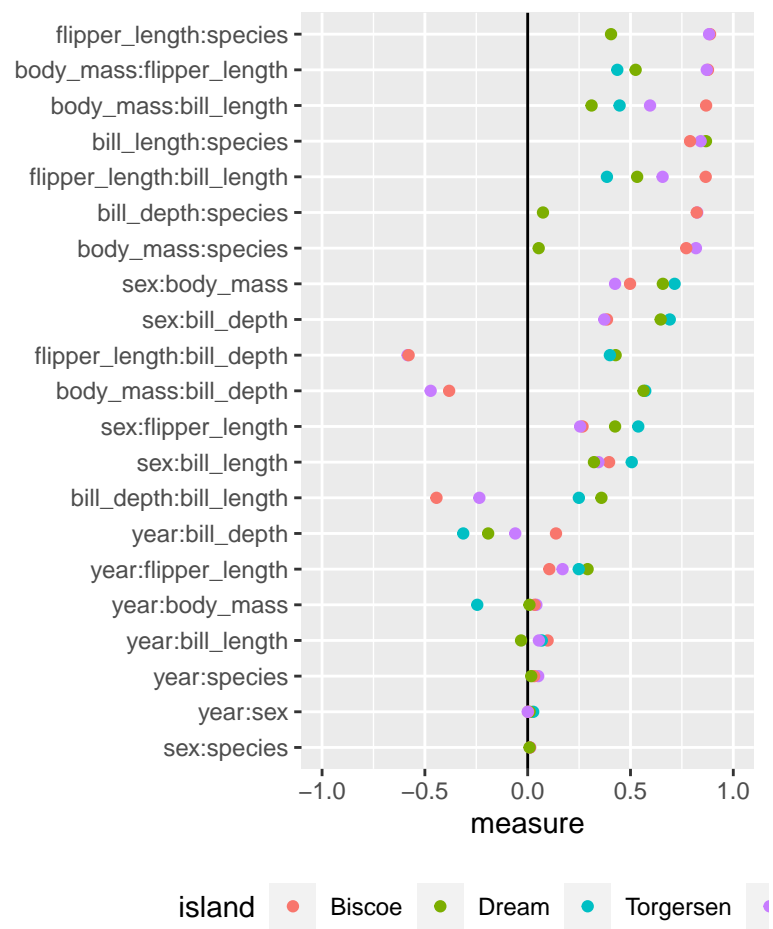
`pairwise_1d_plot` is used for displaying a linear layout of the conditional association for variable pairs in the dataset. The association measures are calculated for every variable pair at each level of partitioning variable using `calc_assoc` function with conditioning variable as the `by` argument.

The measures are then displayed using a dotplot where color of the dots are coded by the level of the partitioning variable and the variable pairs are ordered by absolute maximum value of association measure for each of the pair of variable. These displays are also efficient for discovering differences among the levels of partitioning variable in the data. With the linear layouts it is easier to omit less relevant pairs of variables by filtering the variables pairs having a higher value for association measures than a threshold.

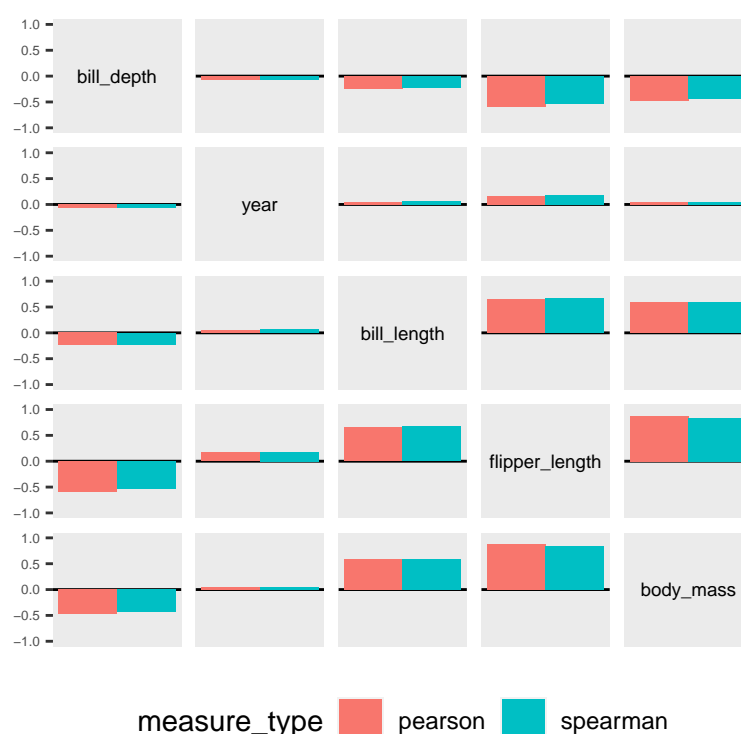
The measures of association calculated for every variable pair at every level of conditioning serve as input to the `pairwise_1d_plot` function. The `assoc` argument uses a tibble of association measures calculated using `calc_assoc` function with a `by` variable. The argument `group_var` is responsible for the grouping variable when plotting the dots. The default value `by` uses the conditioning variable and `measure_type` is useful for displaying dots with color of the dots coded by the type of association measure. The `var_order` argument is responsible for the ordering of variable pairs in the display. If set to default variable pairs are ordered alphabetically and are ordered by absolute maximum value of association measure for every variable pair when set to `max_diff`.

Figure 4 shows a funnel-like linear display for conditional association measures with all the variable pairs on the y-axis, the value of association measure on x-axis and color of the points representing the level of the grouping variable. The linear layout becomes more useful over the matrix layout when the number of variables and number of levels of grouping variable are high.

The `pairwise_2d_plot` function is also useful for comparing various measures using the matrix layout. It plots multiple measures among the variable pairs as bars, where each bar represents one measure of association. Figure 5 shows a matrix layout comparing Pearson's and Spearman's correlation coefficient for the numeric variable pairs in penguins data. The plot shows that the value for both the correlation coefficients are very high for `bill_length` and `flipper_length`, `bill_length` and `body_mass`, and `flipper_length` and `body_mass` suggesting a strong linear and monotonic relationship among these variable pairs in the dataset.



**Figure 4:** Conditional Association plot using linear layout. The display has variable pairs on the Y-axis and the value of association measures on the X-axis. The points corresponding to every variable pair represents the value of association measure for different levels of the conditioning variable and the overall value of association measure.



**Figure 5:** Matrix display comparing Pearson's and Spearman's correlation coefficient. All the variable pairs have similar values for both correlations.

## Section 5: Discussion

We use multiple association measures in a single display for different variable pairs which serves as a comparison tool while exploring association in a dataset and assist in identifying unusual variable pairs. These multiple measures can be displayed in a scatterplot matrix similar to what [Tukey and Tukey \(1985\)](#) proposed. They suggested that scatterplot matrix of the scagnostics measures, which are measures summarizing a scatterplot, can be used to identify unusual scatterplots or variable pairs. [Wilkinson et al. \(2005\)](#) used this idea with their graph-theoretic scagnostic measures to highlight unusual scatterplots. Similarly, [Kuhn et al. \(2013\)](#) have used this idea in a predictive modeling context. They have produced a scatterplot matrix of the measures between the response and continuous predictors such as Pearson's correlation coefficient, pseudo- $R^2$  from the locally weighted regression model, MIC and Spearman's rank correlation coefficient to explore the predictor importance during feature selection step. These displays show the importance of comparing multiple association measures at once for different variable pairs.

## Bibliography

- A. Agresti. *Analysis of ordinal categorical data*, volume 656. John Wiley & Sons, 2010. [p3]
- A. Buja, A. M. Krieger, and E. I. George. A visualization tool for mining large correlation tables: The association navigator., 2016. [p2]
- M. Friendly. Corrgrams: Exploratory displays for correlation matrices. *The American Statistician*, 56(4): 316–324, 2002. [p1, 2]
- S. Gerber. *scorr: s-CorrPlot: Visualizing Correlation*, 2022. URL <http://mckennapsean.com/scorrplot/>. R package version 1.0. [p2]
- W. K. Härdle and L. Simar. *Applied multivariate statistical analysis*. Springer Nature, 2019. [p3]
- M. Hills. On looking at large correlation matrices. *Biometrika*, 56(2):249–253, 1969. [p2]
- M. G. Kendall. The treatment of ties in ranking problems. *Biometrika*, 33(3):239–251, 1945. [p3]
- M. Kuhn, K. Johnson, et al. *Applied predictive modeling*, volume 26. Springer, 2013. [p13]

- M. Kuhn, S. Jackson, and J. Cimentada. *corr: Correlations in R*, 2020. URL <https://CRAN.R-project.org/package=corr>. R package version 0.4.3. [p2]
- P. Morgen and P. Biecek. *corrgrapher: Explore Correlations Between Variables in a Machine Learning Model*, 2020. URL <https://CRAN.R-project.org/package=corrgrapher>. R package version 1.0.4. [p2]
- D. J. Murdoch and E. Chow. A graphical display of large correlation matrices. *The American Statistician*, 50(2):178–180, 1996. [p2]
- U. Olsson. Maximum likelihood estimation of the polychoric correlation coefficient. *Psychometrika*, 44(4):443–460, 1979. [p3]
- D. N. Reshef, Y. A. Reshef, H. K. Finucane, S. R. Grossman, G. McVean, P. J. Turnbaugh, E. S. Lander, M. Mitzenmacher, and P. C. Sabeti. Detecting novel associations in large data sets. *science*, 334(6062):1518–1524, 2011. [p1, 3]
- A. Samba. *linkspotter: Bivariate Correlations Calculation and Visualization*, 2020. URL <https://CRAN.R-project.org/package=linkspotter>. R package version 1.3.0. [p2]
- G. J. Székely, M. L. Rizzo, and N. K. Bakirov. Measuring and testing dependence by correlation of distances. *The annals of statistics*, 35(6):2769–2794, 2007. [p1, 3]
- H. Theil. On the estimation of relationships involving qualitative variables. *American Journal of Sociology*, 76(1):103–154, 1970. [p3]
- E. R. Tufte. *The Visual Display of Quantitative Information*. Graphics Press, USA, 1986. ISBN 096139210X. [p1]
- J. W. Tukey and P. A. Tukey. Computer graphics and exploratory data analysis: An introduction. In *Proceedings of the sixth annual conference and exposition: computer graphics*, volume 85, pages 773–785, 1985. [p1, 13]
- T. Wei and V. Simko. *R package 'corrplot': Visualization of a Correlation Matrix*, 2021. URL <https://github.com/taiyun/corrplot>. (Version 0.92). [p2]
- H. Wickham, M. Averick, J. Bryan, W. Chang, L. D. McGowan, R. François, G. Golemund, A. Hayes, L. Henry, J. Hester, M. Kuhn, T. L. Pedersen, E. Miller, S. M. Bache, K. Müller, J. Ooms, D. Robinson, D. P. Seidel, V. Spinu, K. Takahashi, D. Vaughan, C. Wilke, K. Woo, and H. Yutani. Welcome to the tidyverse. *Journal of Open Source Software*, 4(43):1686, 2019. doi: 10.21105/joss.01686. [p2, 6]
- L. Wilkinson, A. Anand, and R. Grossman. Graph-theoretic scagnostics. In *Information Visualization, IEEE Symposium on*, pages 21–21. IEEE Computer Society, 2005. [p13]

Amit Chinwan  
 Maynooth University  
 Hamilton Institute  
 Maynooth, Ireland  
[amit.chinwan.2019@mumail.ie](mailto:amit.chinwan.2019@mumail.ie)

Catherine Hurley  
 Maynooth University  
 Department of Mathematics and Statistics  
 Maynooth, Ireland  
[catherine.hurley@mu.ie](mailto:catherine.hurley@mu.ie)