corVis: An R Package for Visualising Associations and Conditional Associations

by Amit Chinwan and Catherine Hurley

Abstract We present **corVis**, an R package for visualizing association and conditional association ~~while exploring patterns in a dataset.~~ The package provides matrix and linear layout~~s~~for displaying bivariate association and bivariate association grouped by levels of a categorical variable, using measures suitable for numerical, ordinal and nominal variables. With these displays, an analyst can gain a quick overview about the interesting structure and underlying relationships in the data. We provide a detailed look at the package functions and discuss the implemented design choices. We also provide an illustration of the package on an example dataset.

Section 1: Introduction

As a first stage in an analysis, numerical and graphical variable summaries are explored. Bivariate patterns can be explored with correlation matrices and scatterplot matrices. Typically these are applied to numerical variables only, though there are generalisations of scatterplot matrices (Emerson et al., 2013), implemented in **GGally**, where both numerical and categorical variables are used. For correlation matrices, popularized by Friendly (2002) as corrgram, an analyst would remove the non-numeric variables and then calculate and plot correlations in a matrix. Many datasets include a mix of variable types, and our goal is to investigate and visualise associations among these. The new package **corVis** fills this gap and extends the correlation matrix for different variable types.

In this work, we produce displays of bivariate association, using measures suitable for numerical, ordinal and nominal variables. For numeric variables we go beyond Pearson, Spearman and Kendall correlation, and use methods such as distance correlation (Székely et al., 2007), information theory measure maximal information coefficient (MIC) (Reshef et al., 2011), ace from alternating conditional expectations algorithm and more, which are capable of capturing non-linear patterns. We use association measures for pairs of factor and ordered-factor (ordinal) variables and also for mixed pairs where one variable is numeric and the other is a factor. For numeric-ordinal and nominal-ordinal pair of variables, we do not consider the ordering of the ordinal variable and treat them as mixed and factor pair respectively.

Our first display uses a matrix-layout, similar to corrgram. A novel feature is that our version can show multiple association measures for each pair of variables, so that patterns other than linear association can be uncovered. For high-dimensional datasets, matrix layouts become unwieldy and run out of space, so our second display uses a linear layout, showing one or more association measures for each pair of variables. This is especially useful when the analyst wishes to limit the display to pairs of variables showing non-negligible association.

We also present displays of bivariate association grouped by levels of a categorical variable, again using matrix and linear layouts. These displays are useful for discovering variable pairs showing different relationships at different levels. For our displays, we use seriation methods such that highly-associated variables or variable pairs with high difference among the measures are placed nearby and are easier to identify.

In this paper, we introduce the R package **corVis** where these displays are implemented. 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 multiple association measures and conditional associations. We conclude with a discussion and future work.

Section 2: Background

In this section we provide a brief review of existing packages used for correlation displays and association measure used in the package **corVis**.

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.

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) and produces displays in matrix layout. The package **corrr** (Kuhn et al., 2020) organises correlations as tidy data first, so leveraging the data manipulation and visualisation tools of the **tidyverse** (Wickham et al., 2019), which then can be displayed in a matrix format.

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 **corrr** (Kuhn et al., 2020) also offers network displays where line-thickness encodes correlation magnitude, with a filtering option to discard low-correlation edges. Another package for plotting correlations in a network layout is **linkspotter** (Samba, 2020) which 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.

Friendly (2002) also focused on ordering of the variables for correlation displays where the variables were 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. The package **corrplot** (Wei and Simko, 2021) provides various ordering techniques for matrix displays along with the method implemented in Friendly (2002).

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 on a scatterplot with respect to some user-selected variables of interest, driven by a geometric interpretation of correlation and encoding the correlation as vertical gridlines in the plot. The package allows user 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 is defined as a numerical summary quantifying the relationship between two or more variables. The measure is called symmetric if its value is invariant to the choice of independent or dependent variable during the calculation. For example, Pearson's correlation coefficient summarizes the strength and direction in the range $[-1, 1]$ of the linear relationship present between two numeric variables and is symmetric. Kendall's or Spearman's rank correlation coefficient are other popular measures which assess monotonic relationship in interval $[-1, 1]$ among two numeric variables and are symmetric measures.

Pearson's correlation is a popular association measure because of its easier interpretability but its limitations such as influence of outliers on its magnitude and measuring only linear dependencies

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	
corrr	heatmap/network	
corrgrapher	network	
linkspotter	network	Yes
correlation	heatmap/network	
corVis	heatmap/matrix/linear	Yes

makes it a less useful measure. The recently developed measures such as distance correlation (Székely et al., 2007) and MIC (Reshef et al., 2011) overcome these limitations and are more suitable for datasets with both linear and non-linear patterns.

The distance correlation coefficient (Székely et al., 2007) is an association measure which looks for any relationship among two numeric variables using the distances between observations of these variables and summarizes the relationship in $[0, 1]$. The distance correlation is 0 only when the variables are independent and is a symmetric measure.

The maximal information coefficient (MIC) (Reshef et al., 2011) is an information theory measure which uses mutual information among the two variables for its calculation. The main idea is to find a grid out of possible grids on a scatterplot of two numeric variables, in order to discretize the variables, which maximises the mutual information for the two variables. A normalisation technique is used to make the mutual information from different grids comparable. Referred as 'a correlation of 21st century' (Speed, 2011), MIC is capable of summarizing different types of relationships, not just linear or monotonic, between numeric variables and is in range $[0, 1]$. Reshef et al. (2011) used MIC and other related statistics to explore pairwise relationships in large data sets such as major-league baseball, gene expression, global health, and the human gut microbiota.

Both distance correlation and MIC have advantages of detecting non-linear and complex relationship but these measures aren't perfect yet. Simon and Tibshirani (2014) showed that distance correlation has more statistical power than MIC. Also, distance correlation is not an approximation when compared to MIC. On the other hand, distance correlation computation is slower as compared to the conventional association measures such as Pearson's correlation for datasets with high number of cases.

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 other variable types.

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 which summarizes the relationship in range $[-1, 1]$ between two ordinal variables. The polychoric correlation (Olsson, 1979) measures the correlation between two ordinal variables by assuming two normally distributed latent 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 χ^2 value from the Pearson's χ^2 test for independence and scale it to summarize the association in $[0, 1]$ between two nominal variables. 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. The uncertainty coefficient measure is in the range $[0, 1]$ and is not symmetric. A symmetric version is used by taking the mean of the uncertainty coefficients obtained by treating each variable as independent variable once.

Section 3: Introducing corVis

corVis is an R package which calculates measures of association for every variable pair in a dataset and provides visualizations for displaying associations. Most of the existing correlation displays are limited to numeric pairs of variables. This package extends these displays to every variable pair. The main goal of our work is to propose displays for multiple association measures and conditional associations display which are useful for uncovering interesting patterns in the data. This will help in

Table 2: List of functions in corVis package

Function	Usage	Description
calc_assoc	Calculation	Calculates association measures
calc_assoc_all	Calculation	Calculates all the association measures available in package
plot_assoc_matrix	Visualization	Visualize assocition and conditional association in matrix plot
plot_assoc_linear	Visualization	Visualize assocition and conditional association in linear plot
show_assoc	Visualization	Association (or conditional) plot for a pair of variables

Table 3: Variable description of the Daily Bike Sharing dataset

Variable	Description	VariableType
dteday	date	date
season	season with categories Winter, Spring, Summer and Fall	nominal
yr	year of day with categories 2011 and 2012	nominal
mnth	month of day with months as categories	nominal
holiday	whether day is a holiday or not	nominal
weekday	day of the week	nominal
workingday	if day is neither weekend nor holiday it is Yes, otherwise is No	nominal
weathersit	weather situation of the day with categories clear, cloudy, lightP	nominal
temp	normalized temperature in Celsius	numeric
atemp	normalized feeling temperature in Celsius	numeric
hum	normalized humidity	numeric
windspeed	normalized windspeed	numeric
casual	count of casual users	numeric
registered	count of registered users	numeric
cnt	count of total rental bikes including both casual and registered	numeric

identifying variable pairs which shows a type of relationship or pattern in a dataset with large number of variables.

While designing these displays we consider matrix and linear layouts. A matrix layout reduces the effort in looking up for a variable pair corresponding to a cell or panel, and different measures may be displayed on the upper and lower triangle of the matrix. On the other hand, the filtering of variable pairs, for example pairs having measure value greater than a threshold, is easier with linear layouts in comparison to matrix layouts.

Table 2 provides a list of the functions available in the package. The functions `calc_assoc` and `calc_assoc_all` are responsible for calculating association measures which are used as input for the `plot_assoc_matrix` and `plot_assoc_linear` functions. The functions `plot_assoc_matrix` and `plot_assoc_linear` produces association display, multiple association measures display and conditional association display, in a matrix and linear layout respectively. We provide detailed examples on calculation and visualisation of association and conditional association in next sections.

Example: Data

We use the Daily Bike Sharing dataset (Fanaee-T and Gama, 2014) from the R package **timetk** (Dancho and Vaughan, 2022) which contains daily count of rental bike transactions between years 2011 and 2012 in Capital bikeshare system. The dataset also includes corresponding daily weather information such as humidity, temperature and windspeed, and seasonal information such as season, whether the day is a holiday and whether the day is a working day.

Table 3 provides a brief description of Daily Bike Sharing data along with the types of variables present in the dataset. We use the dataset throughout this paper for illustrative usage of the package.

Section 4: corVis: Calculating Association

This section describes the calculation of association measures in package **corVis**. The package provides a standard interface for calculating a collection of measures which quantifies the relationship between two variables. The measures available in the package are not limited to numeric variables and are used

Table 4: List of the functions available in the package for calculating different association measures along with the packages used for calculation.

name	nn	ff	oo	nf	from	range
tbl_cor	y				stats::cor	[-1,1]
tbl_dcor	y				energy::dcor2d	[0,1]
tbl_mine	y				minerva::mine	[0,1]
tbl_ace	y	y		y	corVis	[0,1]
tbl_cancor	y	y		y	corVis	[0,1]
tbl_nmi	y	y		y	linkspotter::maxNMI	[0,1]
tbl_polycor			y		polycor::polychor	[-1,1]
tbl_tau			y		DescTools::KendalTauA,B,C,W	[-1,1]
tbl_gkGamma			y		DescTools::GoodmanKruskalGamma	[-1,1]
tbl_gkTau			y		DescTools::GoodmanKruskalTau	[0,1]
tbl_uncertainty		y			DescTools::UncertCoef	[0,1]
tbl_chi		y			DescTools::ContCoef	[0,1]

with nominal, ordinal and mixed variable pairs as well. The package also provides a functionality for handling missing value or `NA` while calculating these association measures.

Table 4 lists different functions provided in the package to calculate measures along with the information on type of variable pairs they can be used with. It also include details about the external package functions used to calculate and the range for these measures. The association measures available in **corVis** are symmetric. We convert asymmetric measures to symmetric by taking either the mean or the maximum of the measures calculated by treating each variable from the pair as independent variable. The functions in Table 4 such as `tbl_ace` and `tbl_cancor` which calculates maximal correlation coefficient among the transformed variables and canonical correlation respectively, have been implemented in **corVis**.

For numeric pairs, the package provides popular correlation coefficients like Pearson, Spearman or Kendall and are calculated using `tbl_cor` function. The measures such as distance correlation or MIC for detecting non-linear patterns are implemented using `tbl_dcor` or `tbl_mine` respectively. For ordinal pairs, the measures such as polychoric correlation and Kendall's coefficients are used to find association and are computed by `tbl_polycor` or `tbl_tau` respectively. For nominal pairs , the functions `tbl_uncertainty`, `tbl_chi` or `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 score 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.

The functions listed in Table 4 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 automatically uses pairwise complete observations (depending on the package used for calculation) for taking care of missing values present in the data.

We do not handle date times, or circular variables (usually time related). The only association measure which handles circular variables is ace, but we are not so far using this feature. In the bike data the circular variables are `season`, `month` and `weekday`.

Calculating association measures for whole dataset

The `calc_assoc` function calculates association measures for every variable pair in a dataset. The variable pairs in the output are unique pairs in the dataset where `x` $\neq$ `y`. Because of the tidy structure of the output, the data manipulation and visualization tools of **tidyverse** (Wickham et al., 2019) are applicable 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 code snippet below shows the calculation of association measures for a subset of the bike sharing data. We select three numeric (temp, windspeed, registered) and two nominal variables

(weathersit, workingday) from the original dataset to demostrate the usage of `calc_assoc`. We include all of the function arguments for the below example and describe how these are useful. The inputs such as `by` and `include.overall` will be described in the section Calculating conditional association.

```
bike_s <- bike |>
  dplyr::select(temp, windspeed, registered,weathersit, workingday)
bike_s_assoc <- calc_assoc(d = bike_s,
                           by = NULL,
                           types = default_assoc(),
                           include.overall = NULL,
                           handle.na = TRUE,
                           coerce_types = NULL)
bike_s_assoc
```

```
#> # A tibble: 10 x 4
#>    x          y          measure measure_type
#>    <chr>      <chr>        <dbl> <chr>
#>  1 windspeed  temp       -0.158  pearson
#>  2 registered temp        0.540  pearson
#>  3 weathersit temp        0.121  cancor
#>  4 workingday temp        0.0527 cancor
#>  5 registered windspeed  -0.217  pearson
#>  6 weathersit windspeed   0.120  cancor
#>  7 workingday windspeed   0.0188 cancor
#>  8 weathersit registered  0.282  cancor
#>  9 workingday registered  0.304  cancor
#> 10 workingday weathersit  0.0613 cancor
```

The `types` argument is a tibble of the `tbl_*` functions for different types of variable pairs. The default is `default_assoc()` which includes `tbl_cor` if both the variables are numeric and calculates Pearson's correlation , `tbl_gkGamma` if both the variables are ordinal and computes Goodman and Kruskal's gamma and `tbl_cancor` for a factor pair and mixed pair and calculates canonical correlation.

```
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_cancor  factor  factor  <NULL>
#> 3 tbl_gkGamma ordered ordered <NULL>
#> 4 tbl_cancor  factor  numeric <NULL>
```

The default tibble of measures is updated using the `update_assoc` function. The argument `default` has `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_measures,
                              num_pair = "tbl_cor",
                              num_pair_argList = "spearman",
                              mixed_pair = "tbl_cancor",
                              factor_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_nmi     factor  factor  <NULL>
#> 3 tbl_gkGamma ordered ordered <NULL>
#> 4 tbl_cancor  factor  numeric <NULL>
```

```
updated_bike_s_assoc <- calc_assoc(d = bike_s,
                                   types = updated_assoc)
updated_bike_s_assoc
```

```
#> # A tibble: 10 x 4
#>    x          y           measure measure_type
#>    <chr>      <chr>         <dbl> <chr>
#>  1 windspeed  temp       -0.147   spearman
#>  2 registered temp        0.531   spearman
#>  3 weathersit temp        0.121   cancor
#>  4 workingday temp        0.0527  cancor
#>  5 registered windspeed  -0.203   spearman
#>  6 weathersit windspeed   0.120   cancor
#>  7 workingday windspeed   0.0188  cancor
#>  8 weathersit registered  0.282   cancor
#>  9 workingday registered  0.304   cancor
#> 10 workingday weathersit  0.00275 nmi
```

The input `handle.na` for `calc_assoc` manages the `NA` or missing values in the data. The default value is set to `TRUE` for using pairwise complete observations for calculating a measure of association between two variables.

Sometimes an analyst might want to treat a factor as ordered. This will also be useful for pairs of binary variables where it will then be possible to see the direction of association. Alternatively, binary variables are treated as numerical. The input `coerce_types` is used to convert variable types. The code segment below demonstrates how nominal factors can be converted into ordinal.

```
bike_s_assoc <- calc_assoc(d = bike_s,
                           by = NULL,
                           types = default_assoc(),
                           include.overall = NULL,
                           handle.na = TRUE,
                           coerce_types = list(ordinal=c("workingday", "weathersit")))
bike_s_assoc
```

```
#> # A tibble: 10 x 4
#>    x          y           measure measure_type
#>    <chr>      <chr>         <dbl> <chr>
#>  1 windspeed  temp       -0.158   pearson
#>  2 registered temp        0.540   pearson
#>  3 weathersit temp        0.121   cancor
#>  4 workingday temp        0.0527  cancor
#>  5 registered windspeed  -0.217   pearson
#>  6 weathersit windspeed   0.120   cancor
#>  7 workingday windspeed   0.0188  cancor
#>  8 weathersit registered  0.282   cancor
#>  9 workingday registered  0.304   cancor
#> 10 workingday weathersit  0.133   gkGamma
```

Calculating conditional association

The function `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 finding out variable pairs showing different associations at different levels of the conditioning variable. The function has a `by` argument which is used as the grouping variable and needs to be categorical. The tibble output in the conditional setting has a similar structure as `calc_assoc` without a `by` argument. An additional `by` column representing the levels of the categorical variable is added to 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.

```
bike_s_assoc_by <- calc_assoc(d = bike_s,
                              by = "workingday",
                              include.overall = TRUE)
bike_s_assoc_by
```

```
#> # A tibble: 18 x 5
#>    x          y           measure measure_type by
#>    <chr>      <chr>         <dbl> <chr>        <fct>
```

```
#>  1 windspeed  temp       -0.198  pearson      No
#>  2 registered temp        0.564  pearson      No
#>  3 weathersit temp        0.108  cancor       No
#>  4 registered windspeed  -0.259  pearson      No
#>  5 weathersit windspeed   0.228  cancor       No
#>  6 weathersit registered  0.214  cancor       No
#>  7 windspeed  temp       -0.137  pearson      Yes
#>  8 registered temp        0.550  pearson      Yes
#>  9 weathersit temp        0.136  cancor       Yes
#> 10 registered windspeed  -0.210  pearson      Yes
#> 11 weathersit windspeed   0.0795 cancor       Yes
#> 12 weathersit registered  0.349  cancor       Yes
#> 13 windspeed  temp       -0.158  pearson      overall
#> 14 registered temp        0.540  pearson      overall
#> 15 weathersit temp        0.121  cancor       overall
#> 16 registered windspeed  -0.217  pearson      overall
#> 17 weathersit windspeed   0.120  cancor       overall
#> 18 weathersit registered  0.282  cancor       overall
```

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 the `include.overall` argument to `FALSE`.

```
bike_s_assoc_by <- calc_assoc(d = bike_s,
                              by = "workingday",
                              include.overall = FALSE)
bike_s_assoc_by
```

```
#> # A tibble: 12 x 5
#>    x          y          measure measure_type by
#>    <chr>      <chr>        <dbl> <chr>        <fct>
#>  1 windspeed  temp       -0.198  pearson      No
#>  2 registered temp        0.564  pearson      No
#>  3 weathersit temp        0.108  cancor       No
#>  4 registered windspeed  -0.259  pearson      No
#>  5 weathersit windspeed   0.228  cancor       No
#>  6 weathersit registered  0.214  cancor       No
#>  7 windspeed  temp       -0.137  pearson      Yes
#>  8 registered temp        0.550  pearson      Yes
#>  9 weathersit temp        0.136  cancor       Yes
#> 10 registered windspeed  -0.210  pearson      Yes
#> 11 weathersit windspeed   0.0795 cancor       Yes
#> 12 weathersit registered  0.349  cancor       Yes
```

Calculating multiple association measures

The comparison of multiple association measures help discover patterns other than linear. We calculate multiple measures with `calc_assoc_all` function in the package. The function takes a dataset and a list of measures as input and outputs a tibble structure with multiple measures of association for every variable pair. This output is used by the plotting functions to produce multiple measures display in different layout.

```
#> # A tibble: 16 x 4
#>    x          y          measure measure_type
#>    <chr>      <chr>        <dbl> <chr>
#>  1 windspeed  temp       -0.158  pearson
#>  2 registered temp        0.540  pearson
#>  3 registered windspeed  -0.217  pearson
#>  4 windspeed  temp        0.158  cancor
#>  5 registered temp        0.540  cancor
#>  6 weathersit temp        0.121  cancor
#>  7 workingday temp        0.0527 cancor
#>  8 registered windspeed   0.217  cancor
```

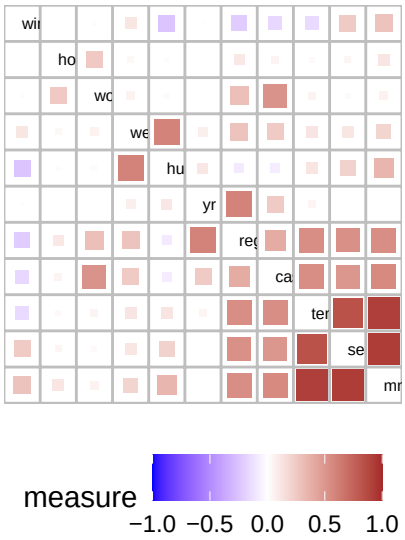

Figure 1: Association matrix display for bike sharing data showing Pearson's correlation for the numeric pairs, Goodman Kruskal's gamma measure for ordered pairs, canonical correlation for factor pairs and mixed pairs. The off diagonal cells show the measure value for a variable pair using a square glyph. The color of every square is mapped with the measure value for the pair and the area of the square is mapped by absolute measure value for the corresponding variable pair. The plot shows many variable pairs with strong association for example (casual, temp), (registered,yr), (weathersit,hum). Also, there is a negative association for (windspeed,registered) suggesting the number of registered users decreased with increase in windspeed.

```
#>  9 weathersit windspeed   0.120  cancor
#> 10 workingday windspeed   0.0188 cancor
#> 11 weathersit registered  0.282  cancor
#> 12 workingday registered  0.304  cancor
#> 13 workingday weathersit  0.0613 cancor
#> 14 windspeed  temp        0.181  dcor
#> 15 registered temp        0.531  dcor
#> 16 registered windspeed   0.208  dcor
```

Section 5: corVis: Visualising Association

This section provides a detailed description of the novel visualisation techniques proposed in the package **corVis**. These methods display association and conditional association for every variable pair in a dataset in a single plot and show multiple bivariate measures of association simultaneously. The package includes functions such as `plot_assoc_matrix` and `plot_assoc_linear` to produce these displays in matrix and linear layout respectively. In addition, the package also provides a function `show_assoc` for a quick graphical overview of the relationship between two variables. It displays a scatterplot for numeric pair, bar plot for ordered and factor pair, and box plot for mixed variable pair.

Association plots

For association analysis, we start with calculating the default association measures for the bike sharing data (we drop variables dteday, weekday, atemp and cnt) using `calc_assoc` and then plot this result using `plot_assoc_matrix` in a matrix layout in Figure 1.

```
bike$dteday <- NULL
bike$weekday <- NULL
bike$atemp <- NULL
bike$cnt <- NULL

bike_assoc <- calc_assoc(d = bike)
plot_assoc_matrix(lassoc = bike_assoc)
```

The diagonal cells in Figure 1 represent the variables present in the data. Every off diagonal cell

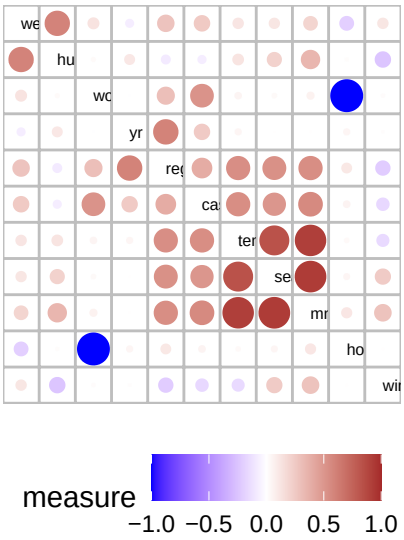

Figure 2: Association matrix display for bike sharing data showing Pearson's correlation for the numeric pairs, Goodman Kruskal's gamma measure for ordered pairs, canonical correlation for factor pairs and mixed pairs. The color of every circle is mapped with the measure value for the pair and the area of the circle is mapped by absolute measure value for the corresponding variable pair. The variables workingday, yr, weathersit and holiday have been converted to ordinals. The plot shows a strong negative association for (workingday, holiday) because no holiday is a working day. The plot also shows a negative association for (weathersit,holiday) suggesting weather wasn't good during the holidays.

contains a glyph, square in this plot, which is filled with a divergent color scale representing the value of corresponding association measure for a variable pair. The `glyph` argument can be either `square` or `circle`. The area of the square is mapped to absolute value of the association measure which quickly highlights the associated pairs of variables. We also offer ordering of the variables in this display so that highly-associated variables are arranged closer to each other and the task of detecting patterns or relations becomes easier. The argument `var_order` is used for the variables in the matrix display. The function uses average linkage hierarchical clustering of the association matrix for ordering the variables, which clusters the highly associated variables together and arranges them nearby.

Figure 1 presents the novel feature of our display showing all the variables of a dataset in the same plot compared to corrgram which only shows association between numeric pairs. With canonical correlation as association measure for factor pairs or mixed pairs, we can observe from Figure 1 that pairs such as (weathersit, humidity), (workingday, registered), (yr,casual), (season, casual) and (season, registered) are strongly associated.

In some cases, an analyst might want to handle some factors as ordinals to see the direction of association. As discussed in the previous section, we can convert variable types by specifying the `coerce_types` argument. The code below shows an implementation and produces a display with some factor variables as ordinals.

```
bike_assoc_o <- calc_assoc(bike,
                           coerce_types=list(ordinal=c("workingday",
                                                       "yr",
                                                       "weathersit",
                                                       "holiday")))

plot_assoc_matrix(bike_assoc_o, glyph="circle")
```

Figure @ref(fig:coerce_types_display) shows a strong negative association for (workingday, holiday) as holidays are not working days. Also, there is a negative association for (weathersit,holiday) as many holidays often fall during the winters.

In order to explore associated variable pairs, the function `show_assoc` is used to plot a scatterplot for numeric pairs, bar plot for factor and ordered pairs, and boxplot for mixed pairs in Figure 3.

```
show_assoc(d = bike,
           x = "temp",
           y = "registered")
```

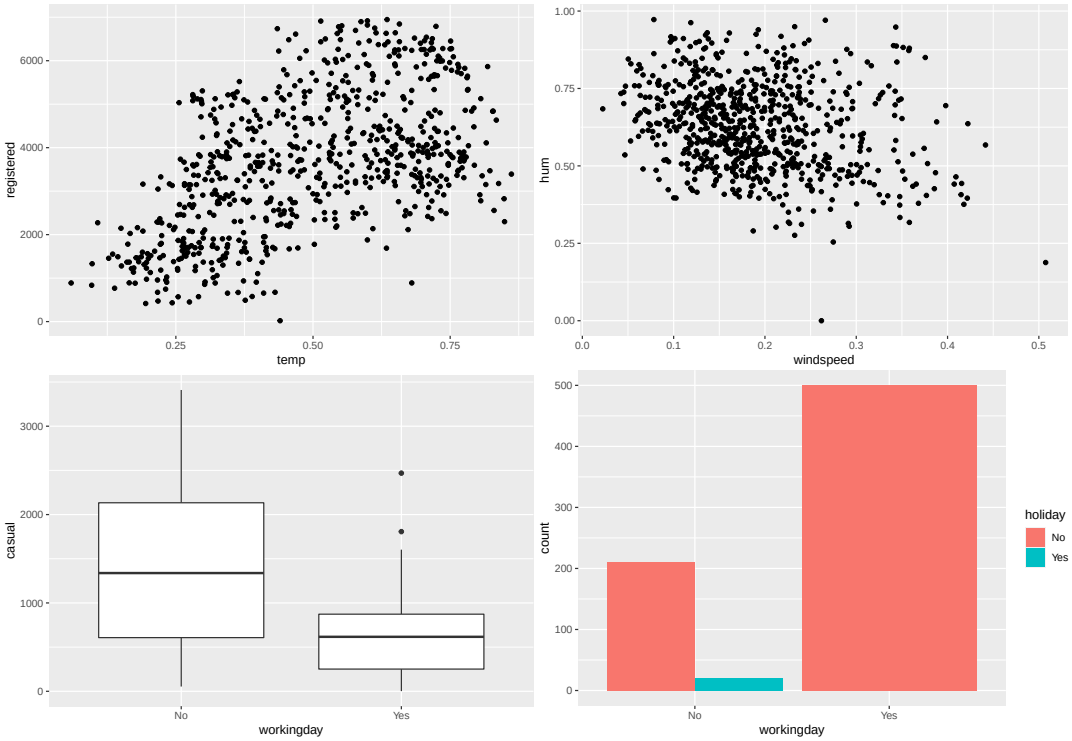

Figure 3: Scatterplot and barplot for numeric variable pair (wtdesire, height) and ordinal variable pairs (genhlth, age), (genhlth,smoke100) and (smoke100, age) showing association between these pair of variables.

```
show_assoc(d = bike,
           x = "windspeed",
           y = "hum")

show_assoc(d = bike,
           x = "workingday",
           y = "casual")

show_assoc(d = bike,
           x = "workingday",
           y = "holiday")
```

Multiple Association Measures Plot

The multiple association measures plot compares association measures for variable pairs in a dataset. This display is useful in detecting pairs of variables showing non-linear relationship which then can be explored further in more detail.

```
biken <- bike |>
  mutate(mnth=as.numeric(mnth)) |>
  select(where(is.numeric))

bn <-calc_assoc_all(biken)
# all measures are positive, so change to
plot_assoc_matrix(bn, limits=c(0,1)) # label position is now wrong!
```

Figure 4 shows a multiple association measures plot in linear layout for German election dataset. The plot compares the absolute values of association measures such as `ace`, `dcor`, `kendall`, `mic`, `pearson` and `spearman` for every variable pair in the dataset. Each cell of the plot corresponds to a variable pair and an association measure, and color intensity of each cell corresponds to the absolute value of association measure. The variable pairs in the plot are ordered by the maximum difference between the absolute value of these measures. The plot shows the variable pair (`Unemployment.03`,`CTT`) with highest

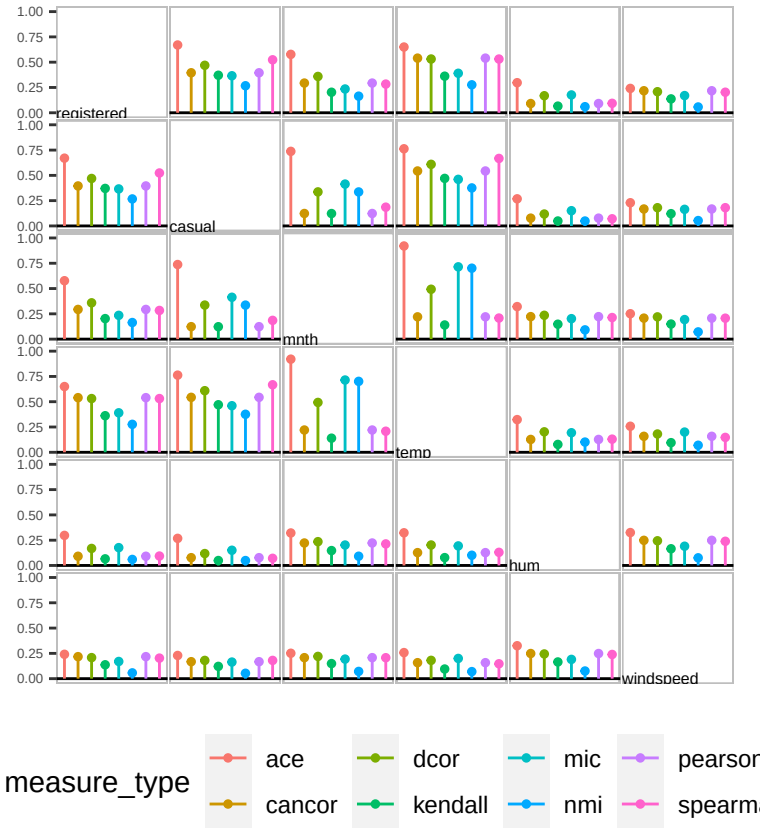

Figure 4: Multiple association measures plot in a linear layout for a subset of German election data. The plot has variable pairs on the Y-axis and association measures on the X-axis. The color intensity of each cell is proportional to the absolute value of association measure. The variable pairs on Y-axis are ordered by the maximum difference between the absolute value of association measures. The plot shows the highest difference for variable pair V.FDP.02 and V.Linke.02 which can be explored further to understand the underlying reasons for this difference.

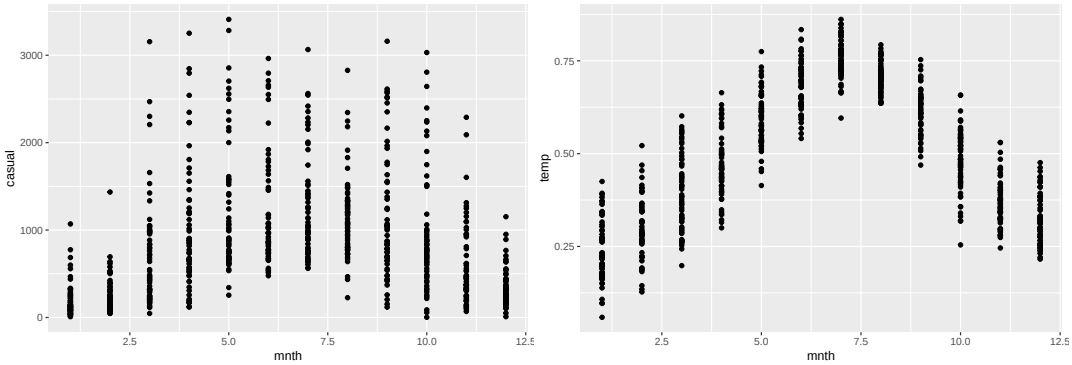

Figure 5: Scatterplot for variable pairs (from left to right) (Unemployment.03 and CTT) and ('Pop.35.60', 'Gruene.02') showing relationships between these pair of variables.

difference between the association measures. The low value for Pearson's correlation and Kendall's correlation suggest no trend but measures such as ace, distance correlation, MIC and Spearman's correlation indicates that a relationship might exist among the variables. Another interesting variable pair evident from the plot is (`Pop.35.60`, `Gruene.02`) for which the three popular measures like Pearson's, Kendall's and Spearman's correlation coefficient are almost zero but measures such as ace, distance correlation and MIC suggest presence of a pattern.

```
show_assoc(biken, "mnth", "casual")
show_assoc(biken, "mnth", "temp")
```

We use `show_assoc` to explore the relationship for the interesting variable pairs in Figure 5. It is evident from the plots that the variable pairs (`Unemployment.03`,`CTT`) and (`Pop.35.60`, `Gruene.02`) show a non-linear trend for which measures such as Pearson's correlation, Kendall's correlation and Spearman's correlation might not be suitable.

Conditional Association Plot

The conditional association plot 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.

```
bike_by_assoc <- select(bike, -workingday, -holiday, -mnth, -yr) |>
  calc_assoc(by="season",
                 coerce_types=list(ordinal=c( "weathersit")))


plot_assoc_matrix(bike_by_assoc)
```

Figure 6 shows a conditional association plot for the cdc data. Each cell corresponding to a variable pair shows two bars which correspond to the association measure (Pearson's correlation for numeric pairs, Goodman and Kruskal's gamma for ordinal pair, canonical correlation for nominal or mixed pairs) calculated at the levels of conditioning variable `genhlth`. The dotted line represents the overall association measure. The plot indicates that there is an evident difference in the Goodman and Kruskal's gamma for the variable pair (`smoke100`, `age`) for different levels of health, compared with each other and overall value. Also, the canonical correlation for variable pair (`weight`, `gender`) for individuals feeling poor or fair health is low compared to the overall value. We explore these variable pairs in more detail using `show_assoc`.

```
show_assoc(bike, "temp", "hum", by="season")

show_assoc(bike, "temp", "registered", by="season")

show_assoc(bike, "temp", "casual", by="season")

show_assoc(bike, "registered", "casual", by="season")
```

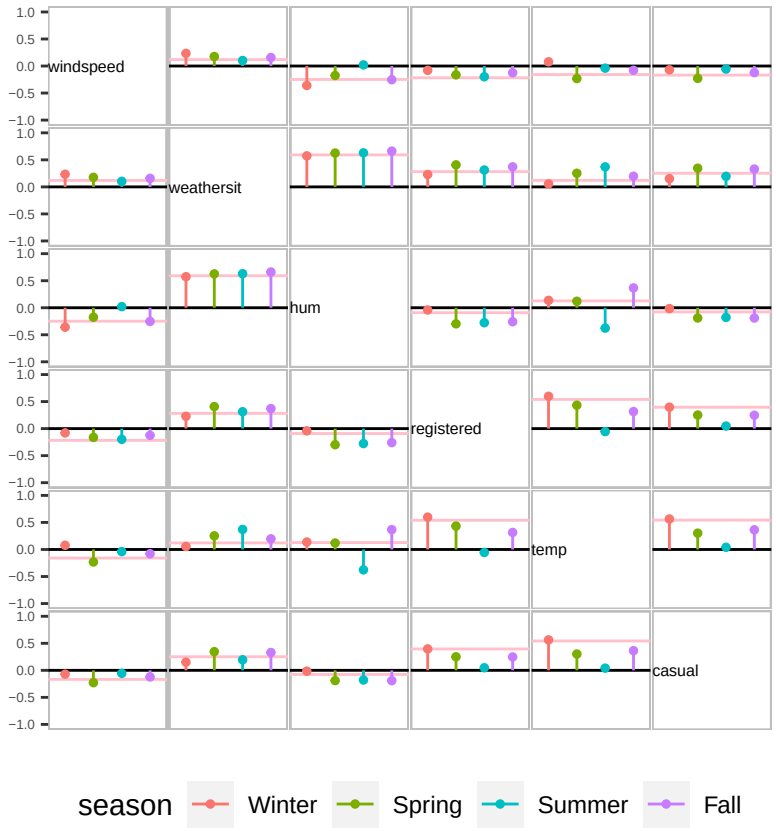

Figure 6: Conditional Association plot for cdc data showing Pearson's correlation for numeric pairs, Goodman and Kruskal's gamma for ordinal pair, canonical correlation for nominal or mixed pairs. The bars in each cell represent the value for asssociation measure colored by the conditioning variable genhlth The dotted line in each cell represents overall value of the association measure. The plot shows evident difference in measure value for pair (smoke100, age) and (weight, gender) for participants with different levels of health in the data.

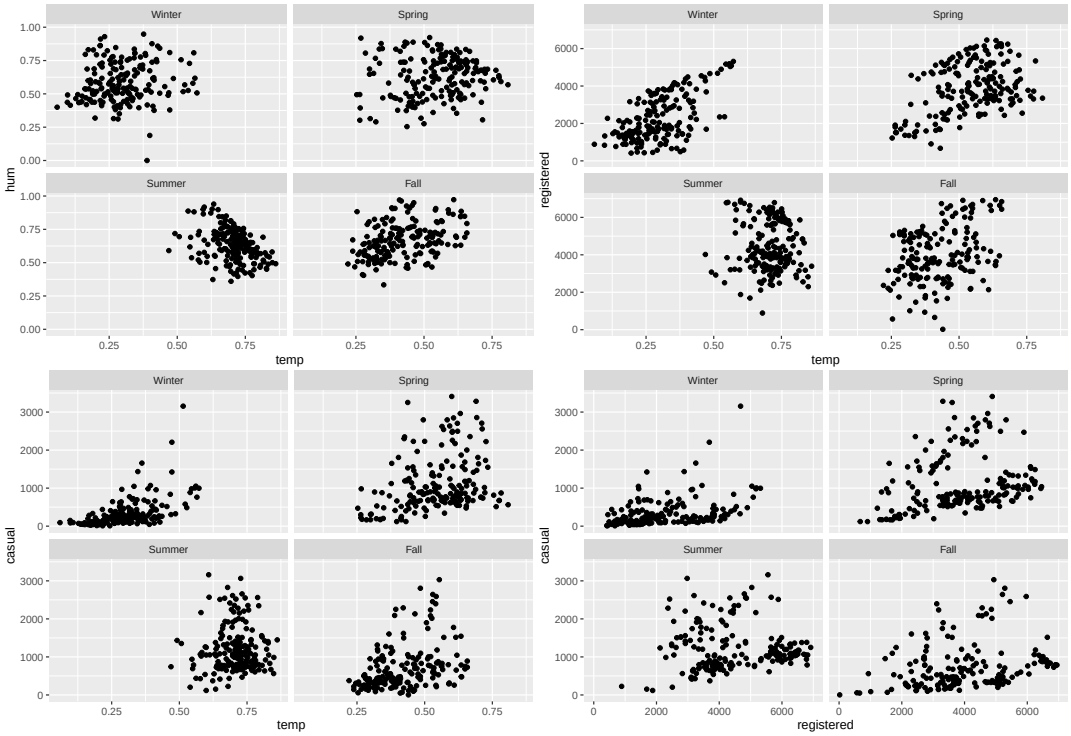

Figure 7: Barplot for variable pair (smoke100, age), and boxplot for variable pair (weight, gender) faceted by conditioning variable genhlth.

Figure 7 shows a barplot for the variable pair (`smoke100`, `age`) and a boxplot for variable pair (`weight`, `gender`) faceted by the conditioning variable `genhlth`. The faceted barplot shows that individuals who are old with smoking habits suffer with poor health more (almost twice) compared to individuals who are old and don't smoke. Interstingly, the faceted boxplot shows that healthier females have low weight compared to the females who don't feel healthy. On the other hand, the weight of the males who feel either health or unhealthy have fairly similar weight.

We also use linear layouts for displaying conditional association in the package. The function `plot_assoc_linear` 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 (or a heatmap) where color of the dots (or each cell) is 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. In comparison to matrix layout, it is easier to omit less relevant pairs of variables in linear layouts by filtering the variables pairs having a higher value for association measures than a threshold.

Figure 8 shows a linear display for conditional association measures with the variable pairs having absoulte measure value greater than 0.1 along the Y-axis, the value of association measure along X-axis and color of the points representing the level of the grouping variable. The linear layout becomes more useful over the matrix layout for conditional association display when the number of variables and number of levels of grouping variable are high.

```
bike_by_assoc <- select(bike, -workingday, -holiday, -mnth, -yr) |>
  calc_assoc(by="season",
                 coerce_types=list(ordinal=c( "weathersit")))
bike_by_assoc <- dplyr::filter(bike_by_assoc, abs(measure) > 0.2)
plot_assoc_linear(assoc = bike_by_assoc,
                  plot_type = "dotplot")
```

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

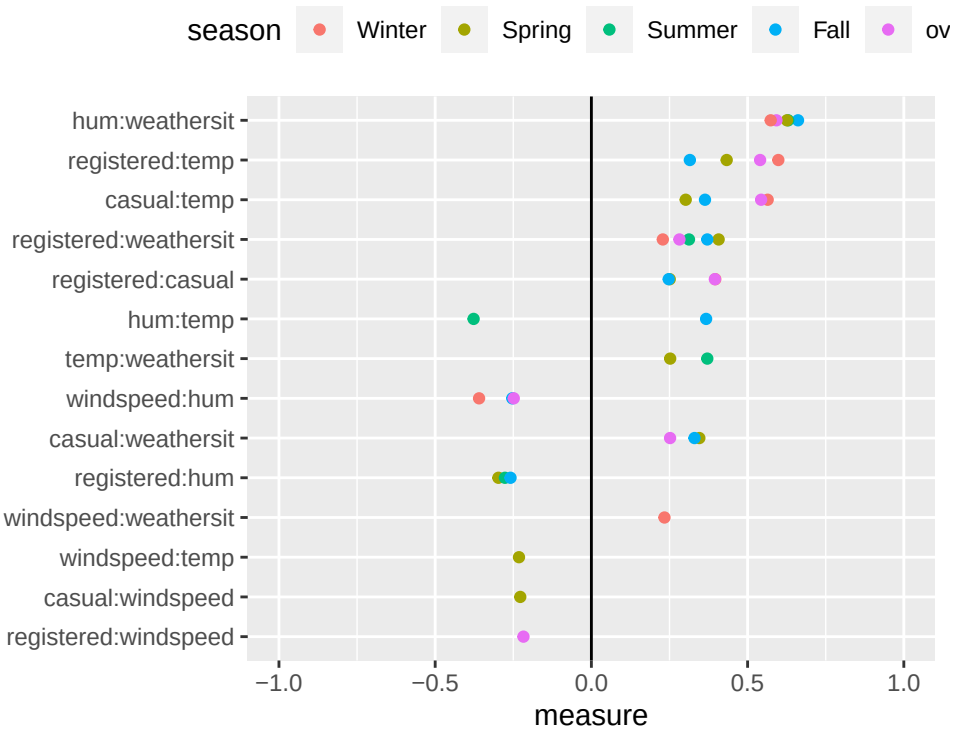

Figure 8: Conditional Association plot using linear layout.The display has variable pairs on the Y-axis and the value of association measures on the X-axis. The points corresponding to every variable pair represents the value of association measure for different levels of the conditioning variable and the overall value of association measure.

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. Dancho and D. Vaughan. *timetk: A Tool Kit for Working with Time Series in R*, 2022. URL https://CRAN.R-project.org/package=timetk. R package version 2.8.2. [p4]

J. W. Emerson, W. A. Green, B. Schloerke, J. Crowley, D. Cook, H. Hofmann, and H. Wickham. The generalized pairs plot. *Journal of Computational and Graphical Statistics*, 22(1):79–91, 2013. [p1]

H. Fanaee-T and J. Gama. Event labeling combining ensemble detectors and background knowledge. *Progress in Artificial Intelligence*, 2:113–127, 2014. [p4]

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]

M. Hills. On looking at large correlation matrices. *Biometrika*, 56(2):249–253, 1969. [p1, 2]

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. [p16]

M. Kuhn, S. Jackson, and J. Cimentada. *corrr: Correlations in R*, 2020. URL https://CRAN.R-project.org/package=corrr. 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. [p1]

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]

N. Simon and R. Tibshirani. Comment on "detecting novel associations in large data sets" by reshef et al, science dec 16, 2011, 2014. URL https://arxiv.org/abs/1401.7645. [p3]

T. Speed. A correlation for the 21st century. *Science*, 334(6062):1502–1503, 2011. [p3]

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]

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. [p16]

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. Grolemund, 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, 5]

L. Wilkinson, A. Anand, and R. Grossman. Graph-theoretic scagnostics. In *Information Visualization, IEEE Symposium on*, pages 21–21. IEEE Computer Society, 2005. [p16]

Amit Chinwan
Maynooth University
Hamilton Institute
Maynooth, Ireland
amit.chinwan.2019@mumail.ie

Catherine Hurley
Maynooth University
Department of Mathematics and Statistics
Maynooth, Ireland
catherine.hurley@mu.ie