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 using measures of association. The package provides matrix and linear layout for displaying bivariate association (and 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 patterns 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.

1 Introduction

The first stage in data analysis includes exploring numerical and graphical variable summaries using different statistical tools. One such tool to explore bivariate patterns is a correlation matrix display which is applied to numerical variables only. These displays are typically used with Pearson's correlation coefficient, a measure of linear dependency, and will mislead an analyst about relationships that aren't linear. In addition, correlation displays commonly use a matrix layout which becomes inefficient when used with high-dimensional datasets.

In this paper, we introduce the R package **corVis** which addresses the challenges of existing correlation matrix display. Our goal in this package is to investigate and visualise associations using measures of association. In **corVis**, we focus on:

1. Association measures for different variable types to go beyond numeric variables
2. Multiple association measures to explore linear and non-linear associations.
3. Conditional association measures to explore patterns at different levels of a conditioning variable
4. Displays for the above three situations in the matrix and linear layouts

As many datasets are a mix of variable types, we use association measures suitable for both numerical and categorical variables. This helps in exploring association for all the pairs of variables in a single plot. We focus on four types of variable pairs which are numeric, factor, ordinal and mixed. We consider a variable pair to be mixed when one of the variables is numeric and the other is categorical.

While exploring association among variables, it is possible to have pairs of variables which are highly associated and not picked by Pearson's correlation, which is a measure of linear association. Because of this, we go beyond Pearson, Spearman and Kendall correlation, and look at alternative measures such as distance correlation (Székely, Rizzo, and Bakirov 2007), maximal information coefficient (MIC) (Reshef et al. 2011)], ace (Wang and Murphy 2005) from alternating conditional expectations algorithm and more, which are capable of capturing non-linear patterns. We compare these multiple measures for each pair of variables so that patterns other than linear can be uncovered.

There may exist variable pairs which show different patterns at different levels of a categorical variable. In order to report these variable pairs to an analyst, we calculate association measures at different levels of a categorical variable for every variable pair in the dataset. On comparison of these measures, we can discover variable pairs showing different structures at different levels of the conditioning or grouping variable.

The above measures are displayed using different layouts. Our first display uses a matrix-layout, similar to existing correlation matrix displays. 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 associations.

We also use seriation methods for both matrix and linear displays such that highly-associated variables or variable pairs with high differences among the measures are placed nearby and are easier to identify. It has been shown that ordering provides a better perception of visualisations and makes it easier to identify patterns in the data. Bertin (1983) showed that the understanding of a matrix is simplified by reordering rows and columns of the matrix. Friendly (2002) demonstrated ordered correlation displays so that groups of variables with high mutual correlation are quickly identified and Catherine B. Hurley (2004) ordered variables in a scatterplot matrix so that interesting panels were positioned close to the main diagonal.

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

In addition to calculating and displaying association measures, we also provide a new tibble data structure for the output of association measures for different variable types, multiple measures of association and conditional association measures. These data structures can be easily explored further using data manipulation and visualization tools of **tidyverse** (Wickham et al. 2019).

In this paper, we introduce the R package **corVis** where we calculate the measures of association for pairs of variables in a tidy data structure and use this structure to produce displays. The next section provides a review of existing packages which deal with correlation displays and a background on association measures and the packages used for calculating them. Then we describe our approach to calculating the association measures, followed by illustrations of the proposed displays. We conclude with a discussion and future work.

2 Background

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

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 a matrix layout. The package **corrr** (Kuhn, Jackson, and Cimentada 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. Grimm (2017) in the package **mbgraphic** explored correlation structure of numeric variables using an interactive shiny application in matrix arrangement. She extended the display to general measures like scagnostics, which are measures characterizing a scatterplot based on trend or density, along with two more measures, based on smoothing splines and distance correlation and compared these measures for variable selection.

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, Jackson, and Cimentada 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.

The R package **correlationfunnel** offers a novel linear 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.

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.

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 relationships in interval $[-1, 1]$ among two numeric variables and are symmetric measures.

Pearson's correlation coefficient is generally used with correlation displays to understand bivariate relationships. But its limitations such as the influence of outliers on its magnitude and measuring only linear dependencies make it a less useful measure. The recently developed measures such as distance correlation (Székely, Rizzo, and Bakirov 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, Rizzo, and Bakirov 2007) is an association measure which looks for any relationship between two numeric variables using the distances between observations of these variables and summarizes the relationship in $[0, 1]$. The distance correlation is 0 when the variables are independent and 1 when the variables are perfectly linear, 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. A normalisation technique is used to make the mutual information from different grids comparable. Referred to 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 the interval $[0, 1]$. MIC is a symmetric measure where a zero value indicates independence among the variables and a value of 1 represents a noiseless functional relationship. 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.

Correlation displays commonly use numeric variables while exploring bivariate associations in a dataset and often ignore categorical variables. There are measures available in the literature for exploring associations involving categorical variables. For nominal variable pairs, measures such as Pearson's contingency coefficient and Uncertainty coefficient (Theil 1970) are used to quantify the association. Pearson's contingency coefficient is a symmetric measure which uses the χ^2 value from Pearson's χ^2 test for independence and is scaled in the interval $[0, 1]$. The uncertainty coefficient (Theil 1970) measures the proportion of uncertainty in one variable which is explained by the other. The uncertainty coefficient 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 an independent variable once.

Agresti (2010) provides an overview of the measures which are used for exploring the association between ordinal variables. Kendall's tau-b (Kendall 1945) is a measure of the strength and direction of the association between two ordinal variables. It is based on the number of concordances and

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

discordances in paired observations of the variables and summarizes the association in the range $[-1, 1]$. The polychoric correlation (Olsson 1979) measures the correlation between two ordinal variables by assuming two normally distributed latent variables and summarizes the association in $[-1, 1]$. Both of these measures are symmetric.

In **corVis**, we provide multiple association measures which can discover non-linear patterns and interesting associations featuring categorical variables.

Seriation

Careful ordering of graphical displays makes it easier to identify patterns and structures. For example, in a barplot of Covid death rate by country, sorting by death rate (instead of alphabetical order) helps identify groups of countries with high (or low) death rates. Other complex ordering examples include Friendly (2002) who demonstrated ordered 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 correlation displays along with the method implemented in Friendly (2002).

The above examples illustrate how seriation, a term to describe the ordering of objects, is useful to reveal interesting patterns. In **corVis**, we use seriation methods from **DendSer** (Catherine B. Hurley and Earle 2022) package to seriate matrix displays and use importance sorting for ordering the linear displays.

3 Overview of `corVis`

The R package **corVis** offers a flexible framework to investigate and visualise associations using measures of association, in datasets with mixed variable types. The calculation and visualisation of association measures are carried out separately in **corVis**, making it an open-ended package for both data structure and display of association measures. This allows users to leverage the widely used **tidyverse** and **ggplot2** framework for exploring these data structures.

In**corVis**, we extend existing correlation displays beyond numeric variables by including mixed variable types and propose displays for multiple association measures useful for uncovering non-linear patterns or associations depending on the levels of a grouping variable. While designing these displays we consider matrix and linear layouts. Linear layouts are useful for high-dimensional datasets and allow a user to limit the display to variable pairs showing strong associations. We also order these displays so that the variable pairs with a strong association or a high difference in measures are placed at prominent positions.

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

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

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.

Data Structures

We provide three tidy data structures which are explored using functions in **corVis** or can be manipulated or visualized by leveraging tools in **tidyverse** These data structures are: `pairwise`, `multi_pairwise` and `cond_pairwise`. Here, we provide an example of these three data structures along with how they are beneficial when used in the **tidyverse** environment.

The `pairwise` data structure is a data format which contains scores for pairs of variables in a dataset for which a specified measure is defined. For example, the function `tbl_dcor` calculates the distance correlation measure for every numeric variable pair in the dataset. Each row of the `pairwise` object is characterized by the variable pair and association measure type.

```
# a subset of bike dataset
bike_s <- bike |>
  dplyr::select(temp, windspeed, registered,weathersit, workingday)
dcor_bike <- tbl_dcor(bike_s)
dcor_bike
```

```
#> # A tibble: 3 x 5
#>   x          y          measure measure_type pair_type
#>   <chr>      <chr>        <dbl> <chr>        <chr>
#> 1 windspeed  temp         0.181 dcor         nn
#> 2 registered temp         0.531 dcor         nn
#> 3 registered windspeed    0.208 dcor         nn
```

```
class(dcor_bike)
```

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

We define a `multi_pairwise` data structure as a data format which consists of multiple scores for variable pairs in the dataset. Similar to the `pairwise` object, every row of the `multi_pairwise` object is defined by the variable pair and measure type. An analyst interested in finding how different the association measure values are for each pair can use this object with **tidyverse** tools to find the range of measures.

```
multi_bike <- calc_assoc_all(bike_s,
                             c("pearson","nmi","ace","uncertainty"))
class(multi_bike)
```

```
#> [1] "multi_pairwise" "pairwise"       "tbl_df"         "tbl"
#> [5] "data.frame"

# calculating range of measures
multi_bike |>
  group_by(x,y) |>
  summarise(range=max(measure)-min(measure),.groups = "drop") |>
  arrange(desc(range))

#> # A tibble: 10 x 3
#>    x          y           range
#>    <chr>      <chr>       <dbl>
#>  1 registered windspeed  0.458
#>  2 windspeed  temp       0.415
#>  3 registered temp       0.374
#>  4 weathersit registered 0.278
#>  5 workingday registered 0.188
#>  6 weathersit temp       0.137
#>  7 weathersit windspeed  0.115
#>  8 workingday weathersit 0.0585
#>  9 workingday windspeed  0.0409
#> 10 workingday temp       0.0194
```

The final data structure in **corVis** is `cond_pairwise` which contains scores for every variable pair at different levels of a conditioning variable. In this case, each row is uniquely identified by a variable pair and a level of the conditioning variable. For exploring pairwise differences in groups, an analyst can easily use data manipulation tools available in **tidyverse** ecosystem.

```
cond_bike <- calc_assoc(bike_s,
                            by="weathersit")
class(cond_bike)

#> [1] "cond_pairwise" "pairwise"      "tbl_df"        "tbl"
#> [5] "data.frame"

# calculating difference in the group
cond_bike |>
  group_by(x,y) |>
  summarise(diff=max(measure)-min(measure),.groups = "drop") |>
  arrange(desc(diff))

#> # A tibble: 6 x 3
#>   x          y            diff
#>   <chr>      <chr>       <dbl>
#> 1 registered windspeed  0.421
#> 2 workingday windspeed  0.411
#> 3 windspeed  temp       0.385
#> 4 workingday registered 0.294
#> 5 workingday temp       0.205
#> 6 registered temp       0.0777
```

It is worth noting that both `multi_pairwise` and `cond_pairwise` objects inherit the `pairwise` class and satisfies the definition of the `pairwise` data structure.

4 corVis: Calculating Association Measures

For exploring associations using **corVis**, the first step is to calculate association measures for variable pairs in a dataset. Table 4 lists the functions provided in the package to calculate measures of association along with the information on the type of variable pairs they can be applied with. It also includes 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 ones by taking either the mean or the maximum of the measures calculated by

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]

treating each variable from the pair as an independent variable. The functions `tbl_ace` and `tbl_cancor` which calculate the maximal correlation coefficient among the transformed variables and canonical correlation respectively have been implemented in **corVis**.

The functions listed in Table 4 for calculating association measures provide functionality for handling missing values or `NA` in the dataset. Each of these functions either has 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. In **corVis**, 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`.

The `tbl_*` functions require a dataset as an input and return a `pairwise` data structure. The output includes the pairs of variables for which the `tbl_*` function is defined, the type of association measure, measure value and the type of variable pair. Our display functions `plot_assoc_matrix` or `plot_assoc_linear` can be used to plot this output in a matrix or linear layout respectively.

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 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 useful for further exploration of pairwise associations. The output of `calc_assoc` is a `pairwise` data structure with one measure for each pair of variables in the dataset.

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 demonstrate 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 5
#>    x          y          measure measure_type pair_type
```

```
#>    <chr>      <chr>        <dbl> <chr>        <chr>
#>  1 windspeed  temp       -0.158  pearson      nn
#>  2 registered temp        0.540  pearson      nn
#>  3 weathersit temp        0.121  cancor       nf
#>  4 workingday temp        0.0527 cancor       nf
#>  5 registered windspeed  -0.217  pearson      nn
#>  6 weathersit windspeed   0.120  cancor       nf
#>  7 workingday windspeed   0.0188 cancor       nf
#>  8 weathersit registered  0.282  cancor       nf
#>  9 workingday registered  0.304  cancor       nf
#> 10 workingday weathersit  0.0613 cancor       ff
```

calc_assoc uses tbl_* functions to calculate a measure for every variable pair. 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 calculate Pearson's correlation, tbl_gkGamma if both the variables are ordinal and compute Goodman and Kruskal's gamma and tbl_cancor for a factor pair and mixed pair and calculate the 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_gkGamma ordered ordered <NULL>
#> 3 tbl_cancor  factor  factor  <NULL>
#> 4 tbl_cancor  factor  numeric <NULL>
```

The default association measures are updated using the update_assoc function. For example, an analyst interested in calculating Spearman's rank correlation for numeric pairs, ace measure for mixed pairs and nmi measure for factor pairs can update these measures as shown in the below code segment.

```
updated_assoc <- update_assoc(default_measures,
                              num_pair = "tbl_cor",
                              num_pair_argList = "spearman",
                              mixed_pair = "tbl_ace",
                              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    ordered ordered <NULL>
#> 3 tbl_cancor factor  factor  <NULL>
#> 4 tbl_ace    factor  numeric <NULL>
```

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

```
#> # A tibble: 10 x 5
#>    x          y          measure measure_type pair_type
#>    <chr>      <chr>        <dbl> <chr>        <chr>
#>  1 windspeed  temp       -0.147  spearman     nn
#>  2 registered temp        0.531  spearman     nn
#>  3 weathersit temp        0.189  ace          nf
#>  4 workingday temp        0.0560 ace          nf
#>  5 registered windspeed  -0.203  spearman     nn
#>  6 weathersit windspeed   0.143  ace          nf
#>  7 workingday windspeed   0.0568 ace          nf
```

```
#>  8 weathersit registered  0.329  ace          nf
#>  9 workingday registered  0.357  ace          nf
#> 10 workingday weathersit  0.0613 cancor       ff
```

The input `handle.na` for `calc_assoc` manages the `NA` or missing value 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 an ordered variable. 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 5
#>    x          y          measure measure_type pair_type
#>    <chr>      <chr>        <dbl> <chr>        <chr>
#>  1 windspeed  temp       -0.158  pearson      nn
#>  2 registered temp        0.540  pearson      nn
#>  3 weathersit temp        0.121  cancor       nf
#>  4 workingday temp        0.0527 cancor       nf
#>  5 registered windspeed  -0.217  pearson      nn
#>  6 weathersit windspeed   0.120  cancor       nf
#>  7 workingday windspeed   0.0188 cancor       nf
#>  8 weathersit registered  0.282  cancor       nf
#>  9 workingday registered  0.304  cancor       nf
#> 10 workingday weathersit  0.133  gkGamma      oo
```

Calculating conditional association measures

The function `calc_assoc` is also used to calculate association measures for all the variable pairs at different levels of a categorical variable. This is useful in exploring the conditional associations and finding out variable pairs showing different associations at different levels of the grouping 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` with an additional `by` column representing the levels of the categorical variable. The output data structure has a `cond_pairwise` class attribute which is used for displaying conditional measures. The data structure is also suitable for tidy operations with tools available in **tidyverse** (Wickham et al. 2019).

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

```
#> # A tibble: 18 x 6
#>    x          y          measure measure_type by      pair_type
#>    <chr>      <chr>        <dbl> <chr>        <fct>   <chr>
#>  1 windspeed  temp       -0.198  pearson      No      nn
#>  2 registered temp        0.564  pearson      No      nn
#>  3 weathersit temp        0.108  cancor       No      nf
#>  4 registered windspeed  -0.259  pearson      No      nn
#>  5 weathersit windspeed   0.228  cancor       No      nf
#>  6 weathersit registered  0.214  cancor       No      nf
#>  7 windspeed  temp       -0.137  pearson      Yes     nn
#>  8 registered temp        0.550  pearson      Yes     nn
#>  9 weathersit temp        0.136  cancor       Yes     nf
#> 10 registered windspeed  -0.210  pearson      Yes     nn
```

```
#> 11 weathersit windspeed   0.0795 cancor       Yes     nf
#> 12 weathersit registered  0.349  cancor       Yes     nf
#> 13 windspeed  temp       -0.158  pearson      overall nn
#> 14 registered temp        0.540  pearson      overall nn
#> 15 weathersit temp        0.121  cancor       overall nf
#> 16 registered windspeed  -0.217  pearson      overall nn
#> 17 weathersit windspeed   0.120  cancor       overall nf
#> 18 weathersit registered  0.282  cancor       overall nf
```

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

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 vector of measures as input and outputs a tibble structure with multiple measures of association for every variable pair. The data structure has `multi_pairwise` class attribute which is used for plotting multiple measure displays in matrix and linear layouts. The code section below calculates `pearson`, `dcor` and `cancor` measures for the variable pairs in subset of bike sharing data. The pairs for which a measure is not defined is not included in the result.

```
#> # A tibble: 38 x 5
#>    x          y          measure measure_type pair_type
#>    <chr>      <chr>        <dbl> <chr>        <chr>
#>  1 windspeed  temp        0.158  cancor       nn
#>  2 registered temp        0.540  cancor       nn
#>  3 weathersit temp        0.121  cancor       nf
#>  4 workingday temp        0.0527 cancor       nf
#>  5 registered windspeed   0.217  cancor       nn
#>  6 weathersit windspeed   0.120  cancor       nf
#>  7 workingday windspeed   0.0188 cancor       nf
#>  8 weathersit registered  0.282  cancor       nf
#>  9 workingday registered  0.304  cancor       nf
#> 10 workingday weathersit  0.0613 cancor       ff
#> # ... with 28 more rows
```

5 corVis: Visualising Association Measures

This section provides a detailed description of the novel displays proposed in the package **corVis**. These displays show multiple association measures to identify variable pairs with non-linear patterns or pairs of variables showing different patterns at different levels of a grouping variable. The package includes functions `plot_assoc_matrix` and `plot_assoc_linear` to produce these displays in a 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 pairs, a bar plot for ordered and factor pairs, and a box plot for mixed variable pairs.

Matrix Displays

Conventionally, correlations have been displayed using a matrix layout. Matrix displays assist in analysing the multivariate structure between the variables. In **corVis**, the `plot_assoc_matrix` function constructs a plot in the matrix layout displaying variable associations.

Careful ordering of matrix displays makes it easier to identify patterns and structures. Some examples include Friendly (2002) who demonstrated ordered correlation displays so that groups of variables with high mutual correlation are easily identified and hurley2004clustering who ordered variables in a scatterplot matrix so that interesting panels were positioned close to the main diagonal. They illustrate how seriation, a term to describe the ordering of objects, is useful to reveal interesting patterns. In **corVis**, we use seriation methods from **DendSer** (Catherine B. Hurley and Earle 2022) package to seriate matrix displays. In many of the seriation algorithms for matrix displays, the first step is to produce a dissimilarity or similarity matrix of objects to be clustered. Here, objects are the variables of a dataset. We use the Lazy path length (LPL) cost function proposed in **DendSer** to obtain an ordering using the dissimilarity matrix of variables. This method is efficient in making interesting pairs more prominent in matrix displays by placing them at the start and top-left position.

We now illustrate the use of the `plot_assoc_matrix` function with `pairwise`, `multi_pairwise` and `cond_pairwise` data structures.

Association Measures Plot

For an association measure display, 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 the `pairwise` output using `plot_assoc_matrix` in a matrix layout.

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

bike_assoc <- calc_assoc(d = bike)
plot_assoc_matrix(lassoc = bike_assoc,var_order = names(bike)) #default ordering


plot_assoc_matrix(lassoc = bike_assoc, var_order = "default") # DendSer ordering
```

The argument `var_order` is used for ordering the variables. For ordering `pairwise` object, a dissimilarity matrix is constructed first where the dissimilarity is measured by $-|m_{ij}|$, where m_{ij} is the association measure value for a variable pair (i, j). This is followed by hierarchical clustering using the seriation weights (similar to the dissimilarity measure), which produces an order such that the LPL cost function is minimised. A user can also supply their own ordering perhaps obtained from other algorithms by specifying the `var_order` argument.

Figure **??** compares the default ordering of the variables in the dataset with ordering obtained by seriation using LPL cost function. The plot shows Pearson's correlation for the numeric pairs, Goodman Kruskal's gamma measure for ordered pairs and canonical correlation for factor pairs and mixed pairs. The plot on the right shows highly associated variables at the top left corner or along the diagonal of the display, making it easier for an analyst to identify associated pairs instantly.

The diagonal cells represent the variables present in the data. Every off diagonal cell contains a glyph, circle 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 `circle` or `square` and is only used for object with `pairwise` class. The radius of the circle is mapped to absolute

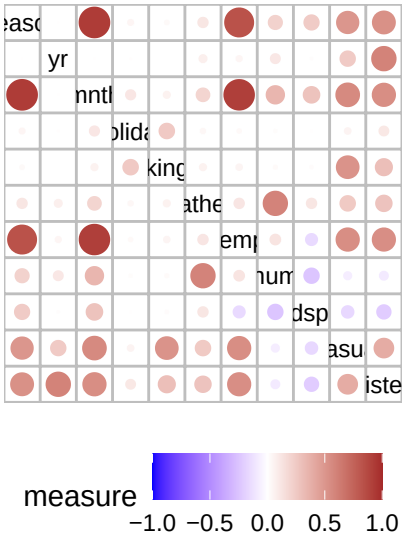

Figure 1: Left: variables in default order of the data; Right: variables ordered by LPL cost function. Association measures displays for bike sharing data showing Pearson's correlation for the numeric pairs, Goodman Kruskal's gamma measure for ordered pairs and canonical correlation for factor pairs and mixed pairs. The off diagonal cells show the measure value for a variable pair using a circle glyph. The color of every circle is mapped with the measure value for the pair and the radius of the circle is mapped to absolute measure value for the corresponding variable pair. It is easier to identify pairs on the right-hand plot with strong association, for example (casual, temp), (registered,yr) and (weathersit,hum). Also, there is a negative association for (windspeed,registered) suggesting the number of registered users decreased during windy days.

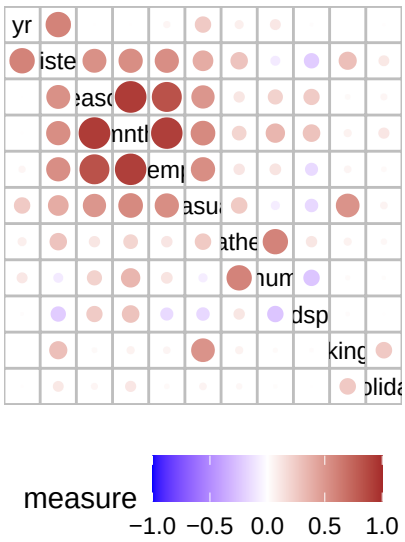

Figure 2: Left: variables in default order of the data; Right: variables ordered by LPL cost function. Association measures displays for bike sharing data showing Pearson's correlation for the numeric pairs, Goodman Kruskal's gamma measure for ordered pairs and canonical correlation for factor pairs and mixed pairs. The off diagonal cells show the measure value for a variable pair using a circle glyph. The color of every circle is mapped with the measure value for the pair and the radius of the circle is mapped to absolute measure value for the corresponding variable pair. It is easier to identify pairs on the right-hand plot with strong association, for example (casual, temp), (registered,yr) and (weathersit,hum). Also, there is a negative association for (windspeed,registered) suggesting the number of registered users decreased during windy days.

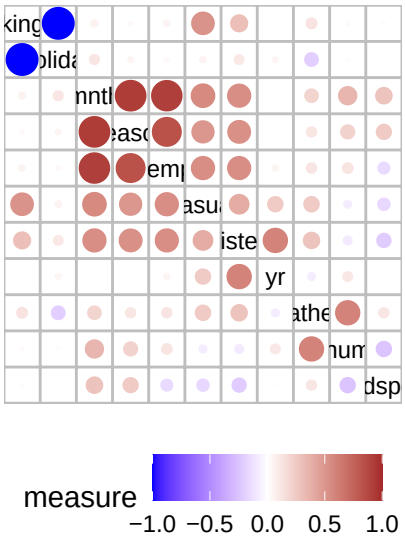

Figure 3: Association matrix display for bike sharing data showing Pearson's correlation for the numeric pairs, Goodman Kruskal's gamma measure for ordered pairs, canonical correlation for factor pairs and mixed pairs. The color of every circle is mapped with the measure value for the pair and the area 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.

value of the association measure. The argument `limits` specify the range of measure values to be mapped to colors. The default value is in the range $[-1, 1]$.

Figure **??** 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 **??** 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 3 shows a strong negative association for (workingday, holiday) as holidays are not working days.

We use function `show_assoc` to explore associated variable pairs graphically. Figure 4 display scatterplots for pairs (temp,registered) and (windspeed,hum) showing a strong positive and negative trend respectively. The boxplot for pair (workingday,casual) shows a high number of casual users using bikes on days that were not working day. The barplot for variable pair (workingday, holiday) confirms that no working day was a holiday.

Multiple Association Measures Plot

The multiple measures plot compares association measures for variable pairs in a dataset. This display is useful in detecting pairs showing non-linear association which then can be explored further in more detail. The first step in producing the display is to calculate multiple pairwise association measures for a dataset using the `calc_assoc_all` function. The `multi_pairwise` output of the function is then fed into `plot_assoc_matrix` to produce a multiple measures display. The `plot_assoc_matrix` function

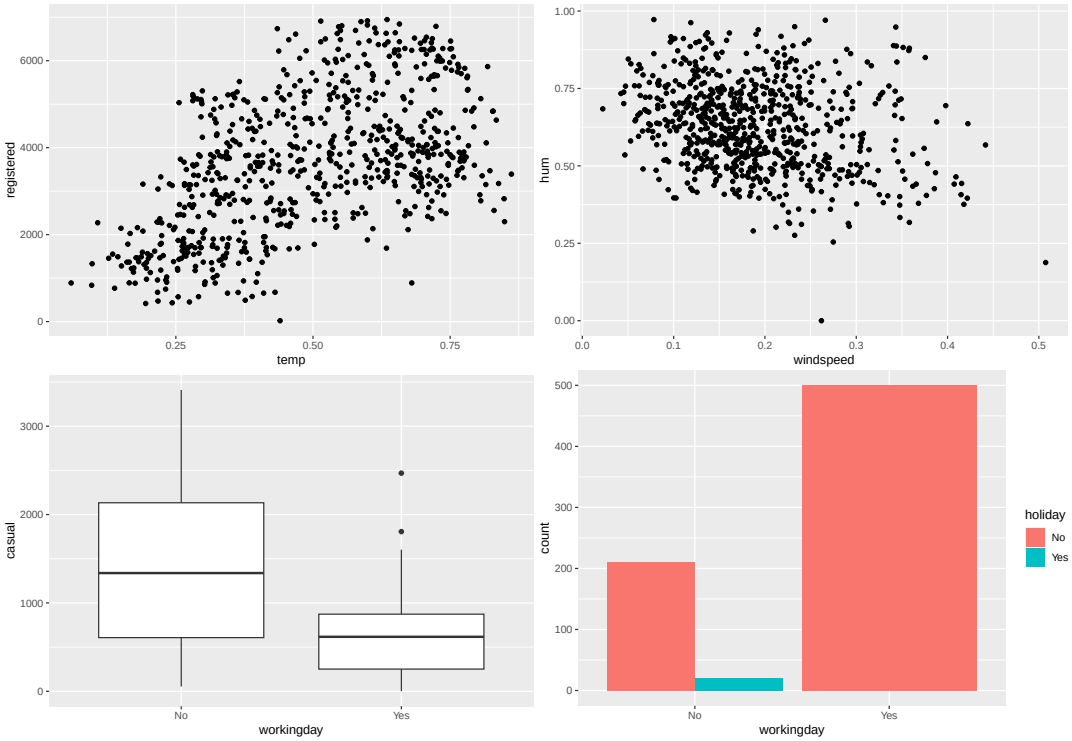

Figure 4: Scatterplots for numeric pair (temp,registered) and (windspeed,hum), boxplot for mixed pair (workingday,casual) and barplot for ordinal pair ((workingday, holiday) showing association between the pairs of variables.

constructs a matrix display where the diagonal cells represent the variables and off-diagonal cells show variable pairs with multiple association measures as lollipops. The height and colour of the lollipops are mapped with the absolute value and by the type of association measure.

For the seriation of this display, a similarity matrix is first obtained by taking the maximum association measure value for a variable pair. This is followed by steps similar to the seriation of the matrix display discussed above. We also order multiple measure types in each cell of the multiple measures display. We use a simple sorting approach by ordering the measure types in decreasing order of their average measure value. This locates measure types with high average values at the start of each cell.

Figure 5 shows a multiple association measures plot in matrix layout for bike sharing dataset. We convert the factor variable `mnth` into numeric and use only numeric variables to produce the display. The plot compares the absolute values of association measures such as `ace`, `cancor`, `dcor`, `kendall`, `mic`, `nmi`, `pearson` and `spearman` for every variable pair in the dataset.

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

biken_assoc_all <-calc_assoc_all(biken)
plot_assoc_matrix(biken_assoc_all)
```

It is evident from the Figure 5 that pairs (casual,mnth) and (mnth,temp) have higher value for `ace` compared to other measures. The measures `nmi`, `dcor` and `mic` have similar values but higher than `pearson`, `spearman` and `kendall`. This suggests the presence of a non-linear pattern for these pairs.

We use `show_assoc` to explore the pattern for these variable pairs in Figure 6. It is evident from the scatterplots that both (casual,mnth) and (mnth,temp) show a non-linear relationship which measures such as Pearson, Kendall or Spearman correlation failed to capture. The `ace` measure detects this non-linear association efficiently as the `ace` algorithm estimates the transformations of variables which leads to maximal correlation for the variable pair and uncovers non-linear pattern.

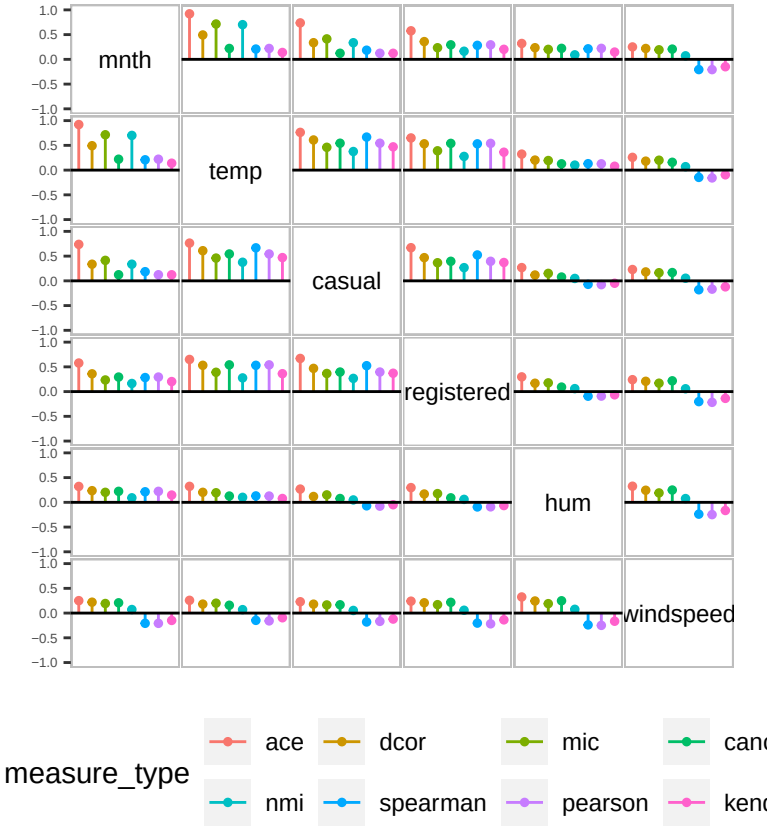

Figure 5: Seriated multiple association measures plot in a matrix layout for numeric variables in bike sharing data. The lollipops in each cell represent the value of the association measure colored by the type of measure. The variable pairs are ordered by the maximum value of association measures such that cells with highest value for any measure are close to the diagonal. The plot shows that pairs (casual,mnth) and (mnth,temp) might have a non-linear association which can be explored further.

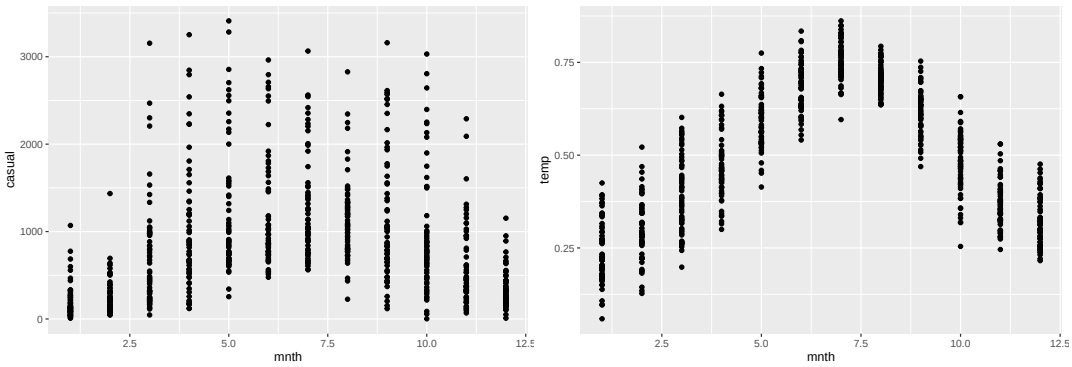

Figure 6: Scatterplot for variable pairs (from left to right) (casual,mnth) and (mnth,temp) showing a non-linear relationship for these pairs.

Conditional Association Measures Plot

The conditional association measures plot explores bivariate association at different levels of a categorical variable. This display is useful for identifying pairs of variables showing different patterns at different levels of a grouping variable. To produce this display, the first step is to calculate association measures for the variable pairs using `calc_assoc` function at each level of the grouping variable which is specified using the `by` argument. The `cond_pairwise` output is then used as input to `plot_assoc_matrix` to produce a conditional measures display.

When supplied with `cond_pairwise` object, the `plot_assoc_matrix` function constructs a conditional association measures display where the diagonal cells represent the variables and off-diagonal cells show variable pairs with association measures as lollipops for levels of a conditioning variable. The height and colour of the lollipops are mapped with the value of association measure and level of the conditioning variable respectively. The overall value of the association measure for the corresponding pair of variable in a cell is represented by a pink horizontal line.

For ordering the variables, we use a similar strategy as discussed above for matrix displays. The only difference is that a similarity matrix is constructed by taking the range of association measure values at different levels of the conditioning variable for a variable pair. For ordering levels of the conditioning variable in a cell, we follow a similar approach used for ordering measure types in multiple measures display.

Figure 7 shows a conditional association plot for the bike sharing data in matrix layout. Each cell corresponding to a variable pair shows four lollipops 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 `season`. The horizontal pink line represents the overall association measure for the pair. The plot shows a low overall correlation between hum and temp. This is also true in Spring and Winter, but the association is positive in fall and negative in Summer. Also, the overall correlation between registered and temp is moderate and positive. This is also true in each season except for summer where the correlation is about 0. The same pattern also holds for casual.

```
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)
```

We explore these variable pairs in more detail using `show_assoc`. Figure 8 shows scatterplots for variable pairs (temp, hum), (temp, registered), (temp, casual) and (registered, casual) faceted by conditioning variable season. The faceted scatterplot for (temp, hum) show the decrease in humidity with increase in temperature in Summer and the oppsoite during Fall. Clearly, the plot for (temp, registered) and (temp, casual) show that there is no clear pattern for registered and casual with temperature in Summer compared to other seasons.

Linear Displays

We provide linear displays in the form of dot plots or heatmaps for plotting association measures and conditional association measures in `corVis`. These displays are handy for focusing on pairs of variables showing non-negligible associations. In **corVis**, the `plot_assoc_linear` function constructs a plot in the linear layout displaying variable associations. The only required input for the `plot_assoc_linear` function is a data structure of class `pairwise`, `multi_pairwise` or `cond_pairwise`. We also provide importance sorting for these displays where the items ordered are variable pairs. We sort the variable pairs in decreasing order by either the maximum or the range between the measures for each pair.

Figure 9 shows a linear display for a `cond_pairwise` data structure. The measures are displayed using dotplot (or a heatmap) where color of the dots (or each cell) is coded by the level of the partitioning variable. The plot shows filtered variable pairs having a difference in measure values equal to or greater than 0.25. It also shows Pearson's correlation for numeric pairs, Goodman and Kruskal's gamma for ordinal pairs and canonical correlation for factor or mixed pairs.

For ordering variable pairs of `cond_pairwise` objects, we use the range of measures. The pairs of variables are ordered in descending order by range. As a result of this ordering, the variable pairs with the highest difference in measures are placed on the top of the display. This makes it easier to find triples of variables showing an interesting pattern. The pair of variables for which the measures at different levels are similar show that there is no effect of conditioning on their association.

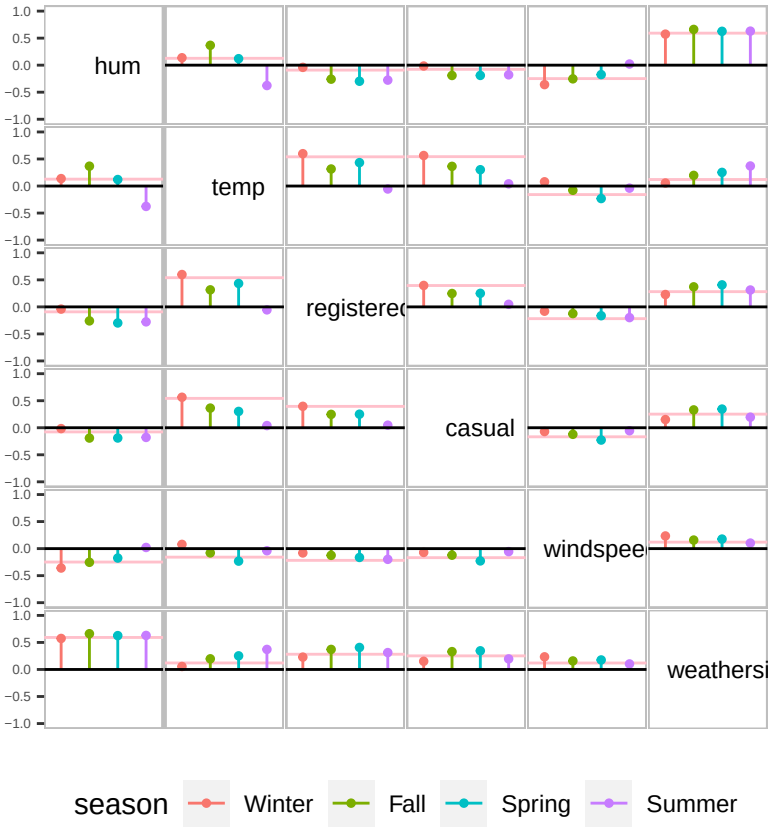

Figure 7: Seriated conditional association measures plot for bike sharing data showing Pearson's correlation for numeric pairs, Goodman and Kruskal's gamma for ordinal pair, canonical correlation for factor or mixed pairs. The lollipops in each cell represent the value for asssociation measure colored by the conditioning variable season. The pink horizontal line in each cell represents overall value of the association measure. The plot shows evident difference in measure value for pairs (temp, hum), (temp, registered), (temp, casual) and (registered, casual) for different seasons.

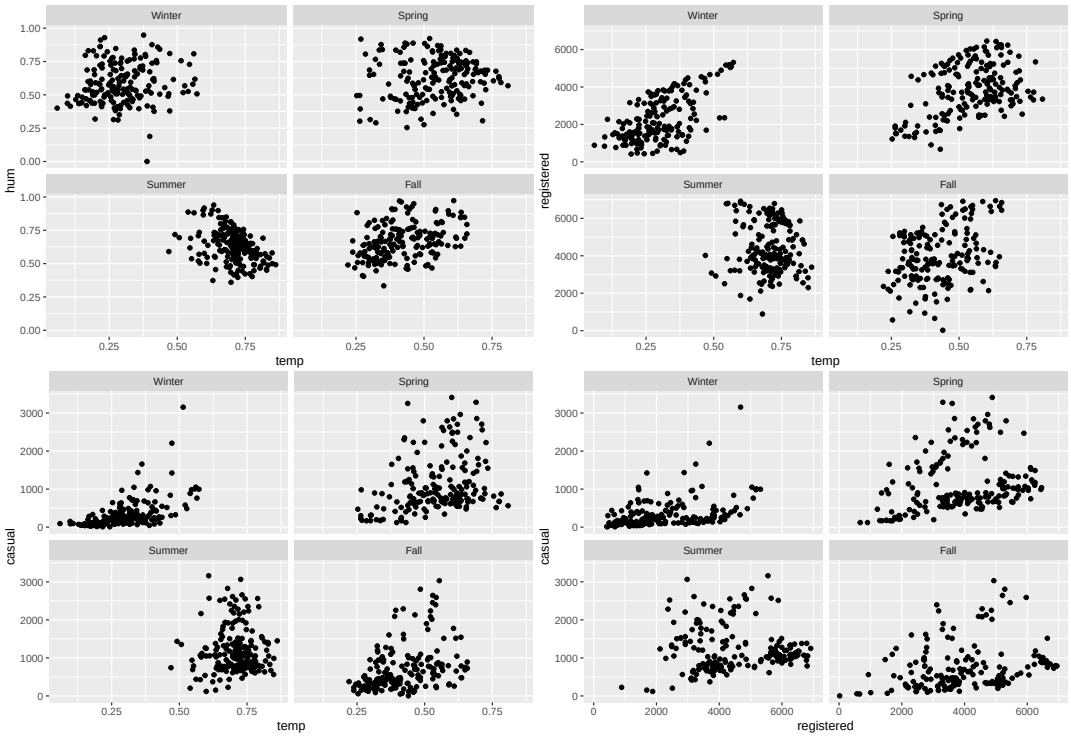

Figure 8: Scatterplots for variable pairs (temp, hum), (temp, registered), (temp, casual) and (registered, casual) faceted by conditioning variable season

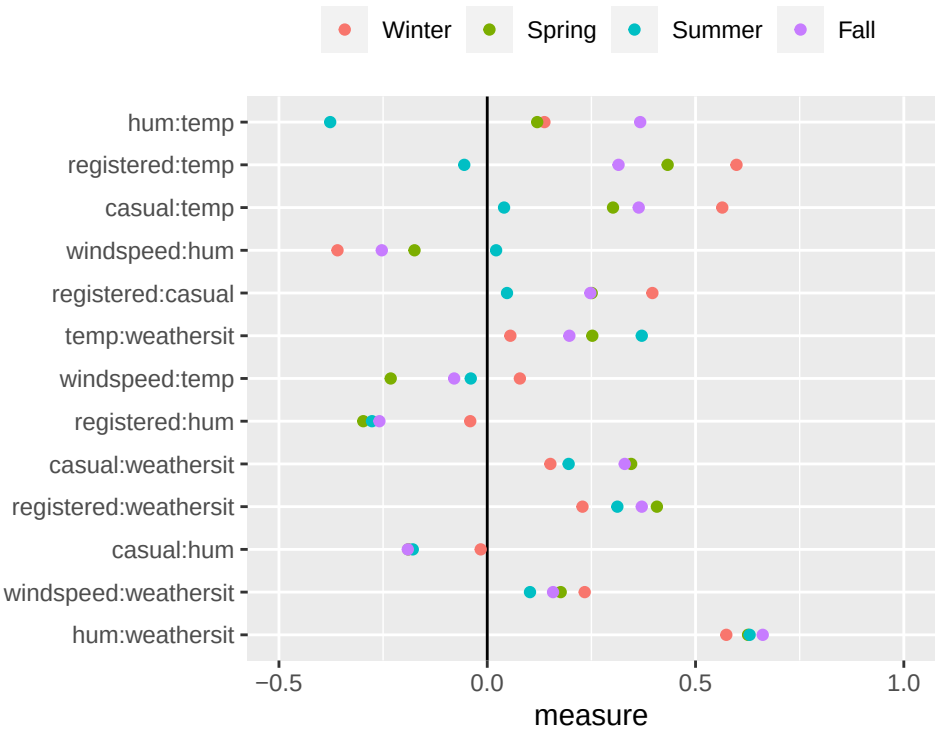

Figure 9: Conditional association measures plot for bike sharing data in linear layout.The display has variable pairs on the Y-axis and the value of association measures on the X-axis. The points corresponding to every variable pair represents the value of association measure for different levels of the conditioning variable and the overall value of association measure.

Figure 9 shows that the variable pair (hum, temp) is placed at the top of the display and has the highest difference between the measures at different levels of season. The two variables humidity and temperature show opposite trends for the summer and fall seasons. This is expected as humidity generally decreases with temperature in summer whereas it tends to increase with temperature during the fall season.

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

bike_by_assoc |>
  group_by(x,y) |>
  summarise(range=max(measure,na.rm = T)-min(measure,na.rm = T)) |>
  arrange(desc(range)) -> bike_by_assoc_rng
bike_by_assoc_rngf <- filter(bike_by_assoc_rng,range>=0.25) # filtering variable pairs with a range of 0.25 or gr
bike_by_assoc_mod <- filter(bike_by_assoc,
                            x %in% bike_by_assoc_rngf$x & y %in% bike_by_assoc_rngf$y)
plot_assoc_linear(bike_by_assoc_mod,
                  plot_type = "dotplot",
                  pair_order = "max-min",
                  limits = c(-0.5,1))
```

6 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, Anand, and Grossman (2005) used this idea with their graph-theoretic scagnostic measures

to highlight unusual scatterplots. Similarly, Kuhn, Johnson, 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.

References

Agresti, Alan. 2010. *Analysis of Ordinal Categorical Data*. Vol. 656. John Wiley & Sons.

Bertin, Jacques. 1983. "Semiology of Graphics: Diagrams, Networks, Maps." *University of Wisconsin Press, Madison, Wisconsin*.

Dancho, Matt, and Davis Vaughan. 2022. *Timetk: A Tool Kit for Working with Time Series in r*. https://CRAN.R-project.org/package=timetk.

Fanaee-T, Hadi, and Joao Gama. 2014. "Event Labeling Combining Ensemble Detectors and Background Knowledge." *Progress in Artificial Intelligence* 2: 113–27.

Friendly, Michael. 2002. "Corrgrams: Exploratory Displays for Correlation Matrices." *The American Statistician* 56 (4): 316–24.

Grimm, Katrin. 2017. "Kennzahlenbasierte Grafikauswahl."

Hills, Michael. 1969. "On Looking at Large Correlation Matrices." *Biometrika* 56 (2): 249–53.

Hurley, Catherine B. 2004. "Clustering Visualizations of Multidimensional Data." *Journal of Computational and Graphical Statistics* 13 (4): 788–806.

Hurley, Catherine B., and Denise Earle. 2022. *DendSer: Dendrogram Seriation: Ordering for Visualisation*. https://CRAN.R-project.org/package=DendSer.

Kendall, Maurice G. 1945. "The Treatment of Ties in Ranking Problems." *Biometrika* 33 (3): 239–51.

Kuhn, Max, Simon Jackson, and Jorge Cimentada. 2020. *Corrr: Correlations in r*. https://CRAN.R-project.org/package=corrr.

Kuhn, Max, Kjell Johnson, et al. 2013. *Applied Predictive Modeling*. Vol. 26. Springer.

Morgen, Pawel, and Przemyslaw Biecek. 2020. *Corrgrapher: Explore Correlations Between Variables in a Machine Learning Model*. https://CRAN.R-project.org/package=corrgrapher.

Murdoch, Duncan J, and ED Chow. 1996. "A Graphical Display of Large Correlation Matrices." *The American Statistician* 50 (2): 178–80.

Olsson, Ulf. 1979. "Maximum Likelihood Estimation of the Polychoric Correlation Coefficient." *Psychometrika* 44 (4): 443–60.

Reshef, David N, Yakir A Reshef, Hilary K Finucane, Sharon R Grossman, Gilean McVean, Peter J Turnbaugh, Eric S Lander, Michael Mitzenmacher, and Pardis C Sabeti. 2011. "Detecting Novel Associations in Large Data Sets." *Science* 334 (6062): 1518–24.

Samba, Alassane. 2020. *Linkspotter: Bivariate Correlations Calculation and Visualization*. https://CRAN.R-project.org/package=linkspotter.

Simon, Noah, and Robert Tibshirani. 2014. "Comment on "Detecting Novel Associations in Large Data Sets" by Reshef Et Al, Science Dec 16, 2011." arXiv. https://doi.org/10.48550/ARXIV.1401.7645.

Speed, Terry. 2011. "A Correlation for the 21st Century." *Science* 334 (6062): 1502–3.

Székely, Gábor J, Maria L Rizzo, and Nail K Bakirov. 2007. "Measuring and Testing Dependence by Correlation of Distances." *The Annals of Statistics* 35 (6): 2769–94.

Theil, Henri. 1970. "On the Estimation of Relationships Involving Qualitative Variables." *American Journal of Sociology* 76 (1): 103–54.

Tukey, John W, and Paul A Tukey. 1985. "Computer Graphics and Exploratory Data Analysis: An Introduction." In *Proceedings of the Sixth Annual Conference and Exposition: Computer Graphics*, 85:773–85. 3.

Wang, Duolao, and Michael Murphy. 2005. "Identifying Nonlinear Relationships in Regression Using the ACE Algorithm." *Journal of Applied Statistics* 32 (3): 243–58.

Wei, Taiyun, and Viliam Simko. 2021. *R Package 'Corrplot': Visualization of a Correlation Matrix*. https://github.com/taiyun/corrplot.

Wickham, Hadley, Mara Averick, Jennifer Bryan, Winston Chang, Lucy D'Agostino McGowan, Romain François, Garrett Grolemund, et al. 2019. "Welcome to the tidyverse." *Journal of Open Source Software* 4 (43): 1686. https://doi.org/10.21105/joss.01686.

Wilkinson, Leland, Anushka Anand, and Robert Grossman. 2005. "Graph-Theoretic Scagnostics." In *Information Visualization, IEEE Symposium on*, 21–21. IEEE Computer Society.

Bibliography

A. Agresti. *Analysis of ordinal categorical data*, volume 656. John Wiley & Sons, 2010. [p]

A. Buja, A. M. Krieger, and E. I. George. A visualization tool for mining large correlation tables: The association navigator., 2016. [p]

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. [p]

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. [p]

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

M. Friendly. Corrgrams: Exploratory displays for correlation matrices. *The American Statistician*, 56(4): 316–324, 2002. [p]

S. Gerber. *scorr: s-CorrPlot: Visualizing Correlation*, 2022. URL http://mckennapsean.com/scorrplot/. R package version 1.0. [p]

K. Grimm. Kennzahlenbasierte grafikauswahl. 2017. [p]

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

M. G. Kendall. The treatment of ties in ranking problems. *Biometrika*, 33(3):239–251, 1945. [p]

M. Kuhn, K. Johnson, et al. *Applied predictive modeling*, volume 26. Springer, 2013. [p]

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. [p]

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. [p]

D. J. Murdoch and E. Chow. A graphical display of large correlation matrices. *The American Statistician*, 50(2):178–180, 1996. [p]

U. Olsson. Maximum likelihood estimation of the polychoric correlation coefficient. *Psychometrika*, 44 (4):443–460, 1979. [p]

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. [p]

A. Samba. *linkspotter: Bivariate Correlations Calculation and Visualization*, 2020. URL https://CRAN.R-project.org/package=linkspotter. R package version 1.3.0. [p]

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. [p]

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

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. [p]

H. Theil. On the estimation of relationships involving qualitative variables. *American Journal of Sociology*, 76(1):103–154, 1970. [p]

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. [p]

T. Wei and V. Simko. *R package 'corrplot': Visualization of a Correlation Matrix*, 2021. URL https://github.com/taiyun/corrplot. (Version 0.92). [p]

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. [p]

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

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