Visualising Bivariate Patterns using Association Measures

Amit Chinwan

A thesis presented for the degree of Research Masters

Supervisor: Dr. Catherine Hurley

Department of Mathematics and Statistics

Maynooth University

Maynooth Co. Kildare, Ireland

Abstract

Correlation matrix displays are valuable tools for investigating bivariate associations, typically showcasing Pearson's correlation—a linear association measure—for pairs of numerical variables. However, these displays have limitations in capturing complex non-linear associations and associations involving categorical variables. This thesis addresses these limitations by introducing alternative association measures that accommodate pairs of numerical, ordinal, and categorical variables, as well as mixed pairs where one variable is categorical and the other is numerical. For numerical variables, we incorporate modern non-linear association measures like distance correlation and the maximal information coefficient (MIC). Notably, our displays present multiple association measures for each variable pair, revealing patterns beyond linear associations or associations dependent on levels of a grouping variable. To address space issues with high-dimensional datasets, we also offer a linear layout display, showing one or more association measures for each variable pair. Furthermore, we employ seriation for matrix displays and importance sorting for linear displays to emphasize highly-associated variables or pairs with significant differences, making them easier to discern and interpret. These improvements enhance the effectiveness and efficiency of data analysis, allowing for a more comprehensive understanding of associations in various datasets.

Table of Contents

List of Figures

List of Tables

List of Code Snippets

CHAPTER 1

Introduction

The focus of this thesis is on using graphical displays to investigate patterns and relationships between variables that may not be evident from numerical summaries alone. These visual summaries are frequently employed to generate hypotheses or research questions that can be further examined in subsequent analysis stages. Given the prevalence of various types of relationships in data, it is essential to employ summary measures that can capture relationships beyond linear ones. Moreover, it is particularly valuable to explore continuous and categorical variables together using a range of summary measures. The aim of this research is to examine the intriguing relationships that exist between different types of variables within a dataset using extensions of correlation matrix displays. Our proposed techniques are implemented in the new *corVis* package in R.

1.1 Correlation Matrix Display

Correlation matrices are widely used in data analysis to study relationships between variables. Traditionally, these matrices are displayed as tables, which can be difficult to read and interpret, especially for larger datasets. [Friendly, 2002] and [Murdoch

and Chow, 1996]proposed methods for displaying correlation matrices using visual representations.

The correlation displays offer a more intuitive and informative way of visualizing correlation matrices. These displays consist of a grid of colored cells, with each cell representing the correlation between a pair of variables. The cells are colored based on the magnitude and direction of the correlation. The intensity of the color corresponds to the strength of the correlation, with darker colors indicating stronger correlations.

In addition to correlation rendering, Friendly also showed the importance of variable ordering in correlation displays for quickly highlighting interesting patterns and relationships in the data. He used the angular ordering of the first two eigenvectors of the correlation matrix for ordering the variables in the display. The ordering placed highly correlated pairs of variables nearby, making it easier to quickly identify groups of variables with high mutual correlation.

The R package *corrplot* [Wei and Simko, 2021] provides an implementation of the methods discussed in Friendly [2002] and is a tool for visualizing correlation matrices in a matrix display. The package provides options for choosing different methods to represent the correlation coefficients in the plot. The glyphs such as circles and squares are included in the package where their areas are proportional to the absolute value of correlation coefficients. Additionally, it provides several methods for ordering the variables in the correlation matrix including hierarchical clustering and the method of angular ordering of eigenvectors discussed in Friendly [2002].

1.2 Thesis outline

While correlation matrix displays are helpful in displaying bivariate associations, they primarily use Pearson's correlation - a measure of linear association between pairs of numerical variables. For complex non-linear associations and categorical variables, other measures are required. Moreover, correlation displays usually adopt a matrix layout, which consume significant space for datasets with a high dimensionality.

This thesis presents *corVis*, an R package that aims to overcome drawbacks of existing correlation matrix displays. The structure of the thesis is as follows:

- Chapter 2 discusses existing R packages which are used for producing correlation matrix displays and reviews the literature on association measures other than Pearson's correlation coefficient.
- Chapter 3 introduces new correlation displays available in the R package *corVis*. These novel displays show multiple association measures and conditional association measures, making them highly useful for data analysis.
- Chapter 4 briefly discusses the use of corVis for identifying patterns such as Simpson's paradox in datasets. The chapter also explores scagnostics patterns using corVis.
- Chapter 5 concludes the thesis and offers insights into possible avenues for further research.

CHAPTER 2

Association Measures and Existing Correlation Displays

2.1 Introduction

The first stage in data analysis includes exploring numerical and graphical variable summaries. Correlation matrix displays are useful for showing bivariate associations. Generally, these show Pearson's correlation, a measure of linear association for pairs of numerical variables. Alternative measures are needed to capture complex non-linear associations, and associations involving categorical variables. In addition, correlation displays commonly use a matrix layout which requires a lot of space for high-dimensional datasets.

This chapter presents a thorough literature review of the existing R packages available for correlation displays, highlighting their limitations. Additionally, it offers a comprehensive review of association measures applicable to pairs of numerical, ordinal, and categorical variables. Moreover, the chapter includes measures specifically designed for mixed pairs of variables, where one variable is categorical and the other is numerical. For numerical variable pairs, the chapter also covers measures

capable of effectively capturing non-linear patterns.

In the upcoming section, we will offer an introduction to association measures and outline the R packages utilized for their computation. Subsequently, we will conduct a comprehensive review of the existing R packages designed for displaying correlations. Finally, we will conclude the chapter by providing a summary of key points discussed.

2.2 Association Measures

An association measure is defined as a numerical summary quantifying the relationship between two or more variables. These measures enable a user to identify dependencies, uncover hidden patterns, and make informed decisions. For example, finding highly correlated variables with a response can help in improving the accuracy of a predictive model. Also, finding highly correlated variables in a regression setting can help in avoiding multicollinearity.

A measure is symmetric if its value is invariant to the order of inputs. For example, Pearson's correlation coefficient is a symmetric measure with a value in the range is $[-1, 1]$ summarising the strength and direction of the linear relationship between two numeric variables. The values of -1 and 1 for Pearson's correlation represent a perfect linear relationship, while 0 indicates no linear relationship.

Kendall's and Spearman's rank correlation coefficients are other popular symmetric measures assessing monotonic association between numeric variables. This section starts with a review of the association measures for numeric pairs, followed by measures for nominal and ordinal pairs and then finally measures suitable for mixed variable pairs.

As Pearson's correlation is a measure of linear association only, it is less useful for other relationships. The recently developed measures such as distance correlation [Székely et al., 2007] and MIC [Reshef et al., 2011] were proposed to overcome this limitation and are more suitable for datasets with both linear and non-linear patterns.

Distance correlation coefficient is a symmetric measure taking a value in $[0, 1]$ and summarises the relationship between two numeric variables using the distances

between observations of these variables. The distance correlation is 0 if and only if the variables are independent and 1 if and only if the variables are perfectly linear. The R package *energy* [Rizzo and Szekely, 2022] implements distance correlation.

The maximal information coefficient (MIC) is an information theory measure which uses mutual information among 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 discretise 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 gives a value 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. MIC and related maximal information based non-parametric exploration statistics have been implemented in the R package *minerva* [Albanese et al., 2012].

Simon and Tibshirani [2014] simulated pairs of variables with different relationships at varying levels of noise and show that distance correlation has more statistical power than MIC for discovering dependencies between variable pairs. They also show that in some cases MIC is less powerful than Pearson correlation. Reshef et al. [2011] defined a statistic to be *equitable* which gives similar scores to equally noisy relationships of different types. Kinney and Atwal [2014] showed that MIC is actually not equitable, a feature which Reshef et al. [2011] claimed. They also note that MIC could have a value of 1 for differing noise levels.

The alternating conditional expectations (ACE) algorithm [Breiman and Friedman, 1985] estimates optimal transformations between response and predictor variables during the regression analysis. For numeric variable pairs, the algorithm calculates the maximal correlation coefficient among the transformed variables, which summarizes the strength of the non-linear relationship between them. It is a symmetric measure and takes a value in the interval $[0, 1]$. It is equal to zero if and

only if the variables are independent. The R package *acepack* [Spector et al., 2016] provides an implementation of ACE algorithm.

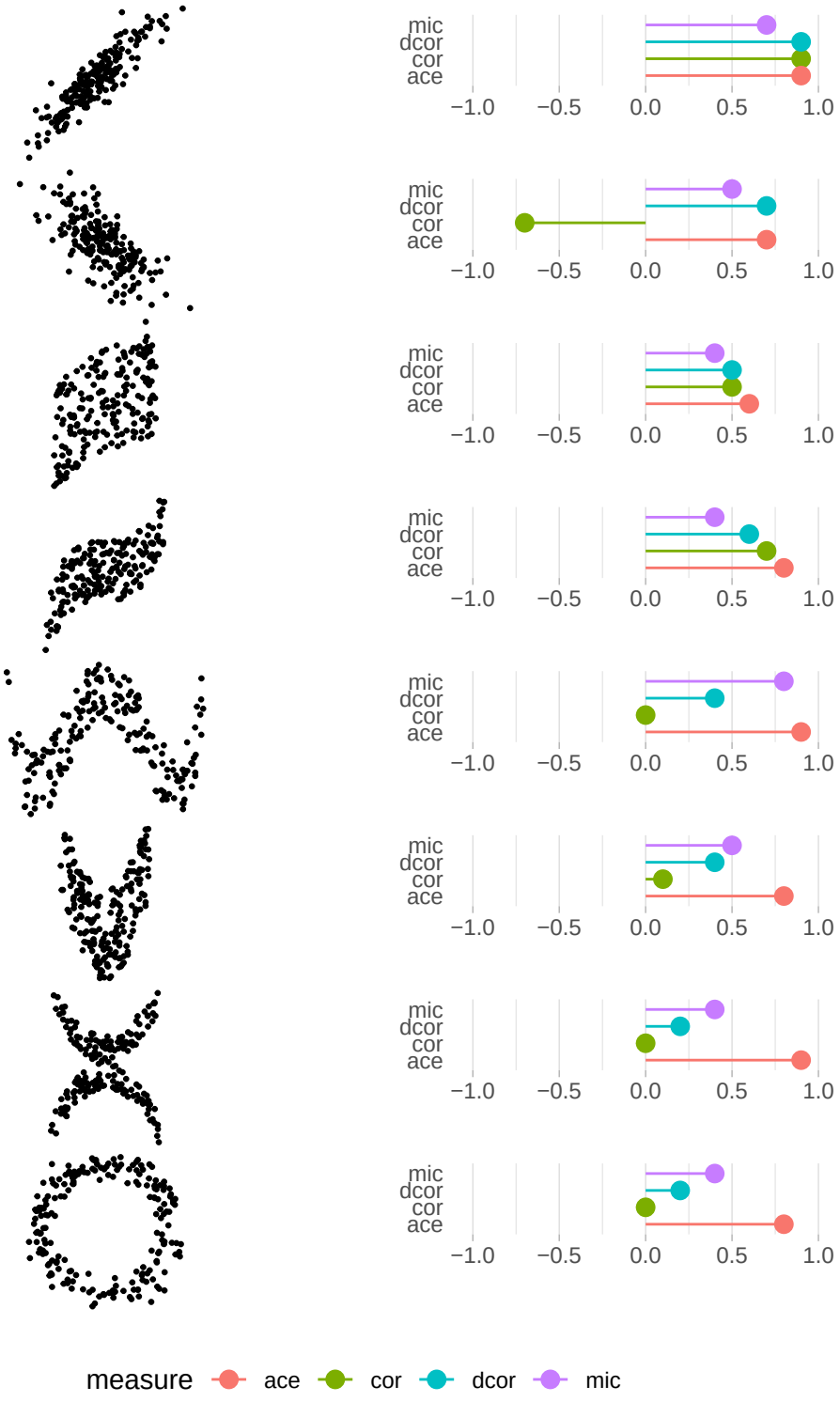


Figure 2.1: Comparison of multiple association measures for simulated patterns

In Figure 2.1, a plot depicting simulated linear and non-linear patterns [Clark,

2013] is presented. Each row showcases a pattern alongside corresponding values for the measures maximal information coefficient (mic), distance correlation (dcor), Pearson's correlation (cor) and ace correlation (ace). The results indicate that all four measures effectively summarize patterns with a linear relationship. However, for the non-linear patterns, distance correlation, MIC, and ace demonstrate greater proficiency in detecting underlying relationships compared to Pearson's correlation. This suggests that when exploring relationships among variables in datasets, it is advisable to utilize association measures such as distance correlation, MIC, and others in conjunction with Pearson's correlation.

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 then scaled to the interval $[0, 1]$. The uncertainty coefficient 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. Both measures are provided in the R package *DescTools* Andri et mult. al. [2022].

Agresti [2010] provides an overview of measures which are used for exploring the association between two 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 discordances in paired observations 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 summarises the association in $[-1, 1]$. Both Kendall's tau-b and polychoric correlation are symmetric and are available in the R packages *DescTools* [Andri et mult. al., 2022] and *polycor* [Fox, 2022].

Normalized Mutual Information (NMI) [Strehl and Ghosh, 2002] serves as a valuable metric for summarizing the association between various pairs of variables. It measures the level of information gained about one variable when observing another. The package *linkspotter* [Samba, 2020] provides an implementation of Maximal Nor-

malized Mutual Information for different variable pairs. The numeric variables are first discretized and then used for calculating mutual information. We use this measure for numeric, nominal and mixed pairs in *corVis*. Canonical correlation [Hotelling, 1992] is used to assess relationships for mixed variable pairs by converting nominal variables into sets of dummy variables, which are then assigned scores to find the maximal correlation. For two numeric variables, this measure is identical to absolute Pearson's correlation, for two factors the correlation is identical to that obtained from correspondence analysis. We provide an implementation of canonical correlation in *corVis*.

ACE, as an association measure, is versatile and can be applied to non-numeric variables as well. The process involves first converting factor variables into numerical representations using encoding techniques and subsequently computing the association measure.

2.3 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 2.1 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. 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

Package	Display	Mixed Variables
corrplot	heatmap	
mbgraphic	heatmap	
corrr	heatmap/network	
corrgrapher	network	
linkspotter	network	✓
correlation	heatmap/network	
corVis	heatmap/matrix/linear	✓

Table 2.1: List of the R packages dealing with correlation or correlation displays with information on the display layouts available in these packages and whether the package show mixed variables in a single plot

for matrix displays along with the method implemented in [Friendly, 2002].

The correlation matrix plot for the penguins dataset, depicting various measurements and attributes for three penguin species (Adelie, Chinstrap, and Gentoo) from the *palmerpenguins* [Horst et al., 2020] package, is showcased in Figure 2.2. This visualization is generated using the *corrplot* package. The plot exhibits circular glyphs within each cell, where the glyph's area is proportional to the absolute value of the correlation coefficient between the variables associated with that cell. Additionally, the color intensity of each glyph corresponds to the magnitude of the correlation coefficients.

Due to its limitation of being designed exclusively for numeric data, the *corrplot* package does not display the categorical variables within the dataset. This poses a constraint for analysts seeking to visualize associations between categorical or mixed-type variables. The limited space available for displaying correlations in the matrix layout becomes a constraint to visualise large correlation matrices.

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].

Grimm [2017] in the package *mbgraphic* explored correlation structure of numeric variables using interactive matrix displays. She extended the display to general

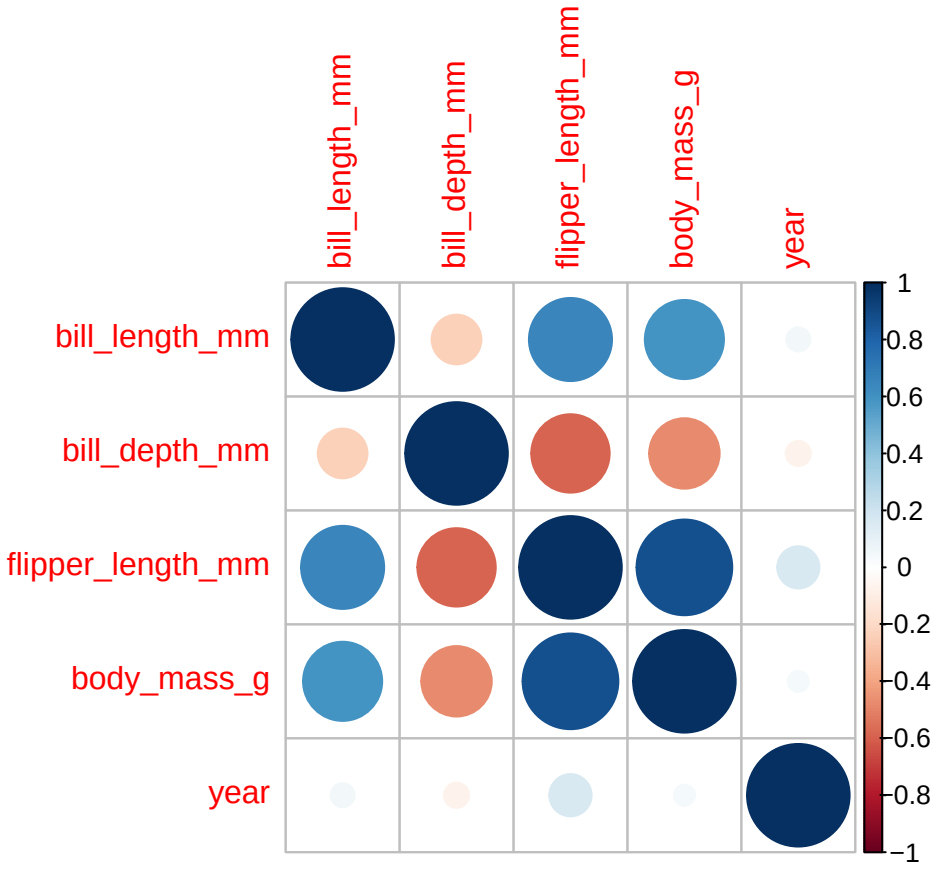
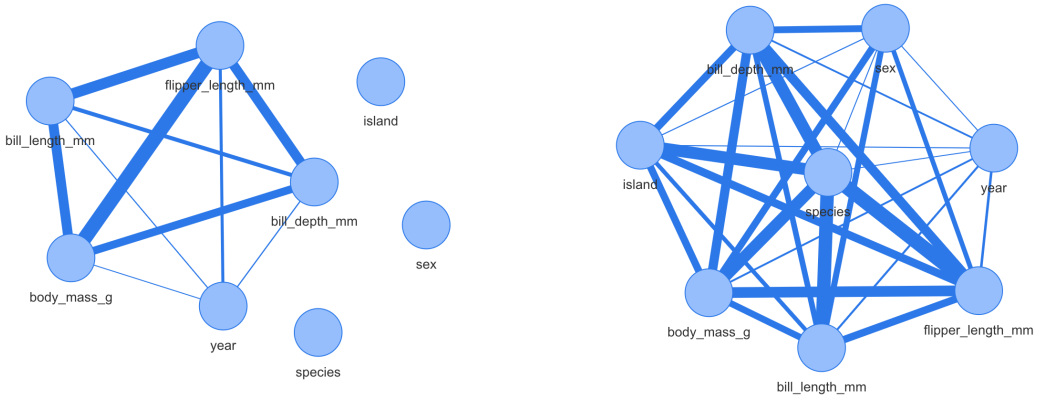


Figure 2.2: Correlation plot using R package *corrplot*. The area of the circle within each cell corresponds to the Pearson's correlation value between the two variables, and the direction of correlation is indicated by the diverging color scale

measures like scagnostics [Wilkinson et al., 2005], which are measures characterizing a scatterplot, along with two more measures, one which is distance correlation and the other based on smoothing splines.

The package *corrgrapher* [Morgen and Biecek, 2020] uses a network plot for exploring correlations, where nodes close to each other have high correlation magnitude, edge thickness encodes absolute correlation and edge color indicates the correlation sign. 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* 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. Figure 2.3 illustrates network plots based on Pearson's correlation and Maximum NMI for the penguins dataset. In these plots, the nodes represent variables, and the edge thickness is proportional to the absolute value of the correlation or NMI. If the measure is undefined for a pair of variables, there is no connecting edge between them. Additionally, these network plots can be incorporated into an interactive shiny application.



(a) Pearson correlation plot

(b) Maximum NMI plot

Figure 2.3: Association measures plot using *linkspotter* package

The R package *correlation* can also be used to explore multiple measures of association. It allows for the inclusion of factors by converting them into numeric variables when computing correlations or other association measures. These measures can then be effectively visualized using packages like *corrplot*. In Figure 2.4,

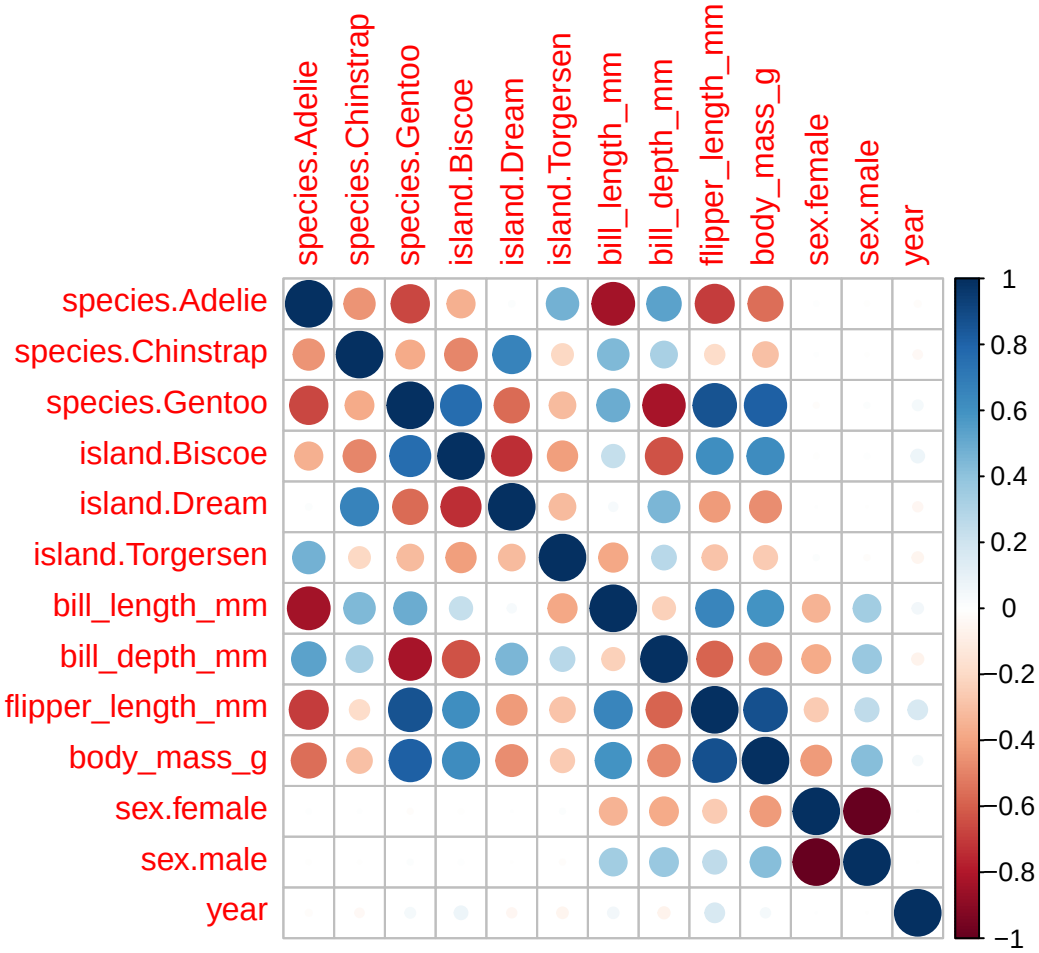


Figure 2.4: Correlation plot for all the variables in the penguins dataset using R packages *correlation* and *corrplot*. The size of the circle within each cell reflects the Pearson's correlation value between the two variables, while the direction of correlation is depicted by the diverging color scale

we can observe a correlation matrix display for both numeric and factor variables in the penguins dataset. Each category within factor variables is transformed into a numeric variable using one-hot encoding, and subsequently, the correlation is computed. However, a limitation of this display is that it doesn't offer insights into the overall relationship of a factor variable with other variables.

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* [McKenna

et al., 2016] 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* [Dancho, 2020] 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. In the resulting dataset, all variables, now categorized, undergo one-hot encoding. This means that each unique category is represented as a binary vector, and Pearson's correlation is computed 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 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 chapter.

CHAPTER 3

corVis

3.1 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* for exploring these data structures and *ggplot2* [Wickham, 2016] framework for visualizing them.

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 display components so that the variable pairs with a strong association or a high difference in measures are placed at prominent positions.

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

Table 3.1: List of main functions in *corVis* package

Table 3.1 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 explain package functionality in more detail and provide examples in the following sections.

3.2 Calculating Association Measures in *corVis*

For exploring associations using `corVis`, the first step is to calculate association measures for variable pairs in a dataset. The functions available in the package for computing measures of association, along with details on the types of variable pairs they are applicable to, are listed in Table 3.2. In the table, nn denotes numeric variable pairs, ff represents factor variable pairs, oo indicates ordinal pairs, and nf corresponds to variable pairs consisting of one numeric and one factor variable. 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. To transform asymmetric measures into symmetric ones, we compute the mean of the measures derived by treating each variable within a variable 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 directly implemented in `corVis`.

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

Table 3.2: List of the functions available in the package for calculating different association measures along with the packages used for calculation

We use the Daily Bike Sharing dataset [Fanaee-T and Gama, 2014] from the R package *timetk* [Dancho and Vaughan, 2022] which contains daily counts of rental bike transactions in the years 2011 and 2012 in the Capital bikeshare system. The dataset also includes information on daily weather (humidity, temperature and windspeed), the season, whether the day is a holiday and whether the day is a working day. Table 3.3 provides a listing of variables and their types. We use the dataset throughout this and the next section for illustrative usage of our package.

The functions listed in Table 3.2 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 which 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 circu-

Variable	Description	Variable Type
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

Table 3.3: Variable description of the Daily Bike Sharing dataset

lar 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 in a tibble or dataframe format as input and return a data structure of class `pairwise`. 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.

3.2.1 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 visualisation tools of *tidyverse* 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 3.1 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 3.3.3.

```
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)
```

Listing 3.1: Calculating association measures for whole dataset

`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 `tbl_*`'s are specifiecd by `default_assoc()` which uses `tbl_cor` (Pearson's correlation) if both variables are numeric, `tbl_gkGamma` (Goodman and Kruskal's gamma) if both variables are ordinal, and `tbl_cancor` (canonical correlation) for a mixed factor, numeric pair.

```
default_measures <- default_assoc()
default_measures
# A tibble: 4    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>
```

Listing 3.2: Default association measures

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 code segment 3.3.

```
updated_assoc <- update_assoc(default_measures,
                              num_pair = "tbl_cor",
                              num_pair_argList = "spearman",
                              mixed_pair = "tbl_ace",
                              factor_pair = "tbl_nmi")
```

Listing 3.3: Updating association measures

The input `handle.na` for `calc_assoc` specifies how `NA` (missing values) in the data should be treated. The default value is set to `TRUE` which uses pairwise complete observations for the two variables and calculates a measure of association between them.

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. The input `coerce_types` is used to convert variable types. The code segment 3.4 demonstrates how nominal factors such as weathersit can be converted into ordinal. The variable weathersit has a natural order to it.

```
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")))
```

Listing 3.4: Updating variable types

3.2.2 Calculating conditional association measures

The function `calc_assoc` is also used to calculate association measures for all 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*. The code section 3.5 calculates conditional association measures for the variable pairs in the subset of bike data at different levels of grouping variable workingday.

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

Listing 3.5: Calculation of conditional association measures

3.2.3 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 code section 3.6 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 are not included in the result.

```
bike_s_assoc_multi <- calc_assoc_all(d = bike_s,
                                     measures = c("pearson",
                                                  "dcor",
                                                  "cancor"))
```

Listing 3.6: Multiple association measures

3.3 Visualising Association Measures in *corVis*

This section provides a detailed description of the displays offered 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.

3.3.1 Association Measures Plot

For association analysis, we start with calculating the default association measures for the bike sharing data (we drop dteday and weekday as both are circular variables. Also, we drop atemp as it is highly correlated with temp and cnt as it is the sum of casual and registered) using `calc_assoc` and then plot this result using `plot_assoc_matrix` in a matrix layout in Figure 3.1.

The diagonal cells in Figure 3.1 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 value of the association measure. The argument `limits` (defaulting to $[-1, 1]$) specifies the range of measure values to be mapped to colors. The display in Figure 3.1 is similar to the correlation plot produced by the package `corrplot`.

We also offer ordering of the variables in this display so that highly-associated variables are arranged closer to each other. The argument `var_order` is used for ordering the variables in the matrix display. We use a seriation algorithm from `DendSer` [Hurley and Earle, 2022] package to order variables. The methods used for ordering the displays in the package will be discussed in more detail in the next section 3.4.

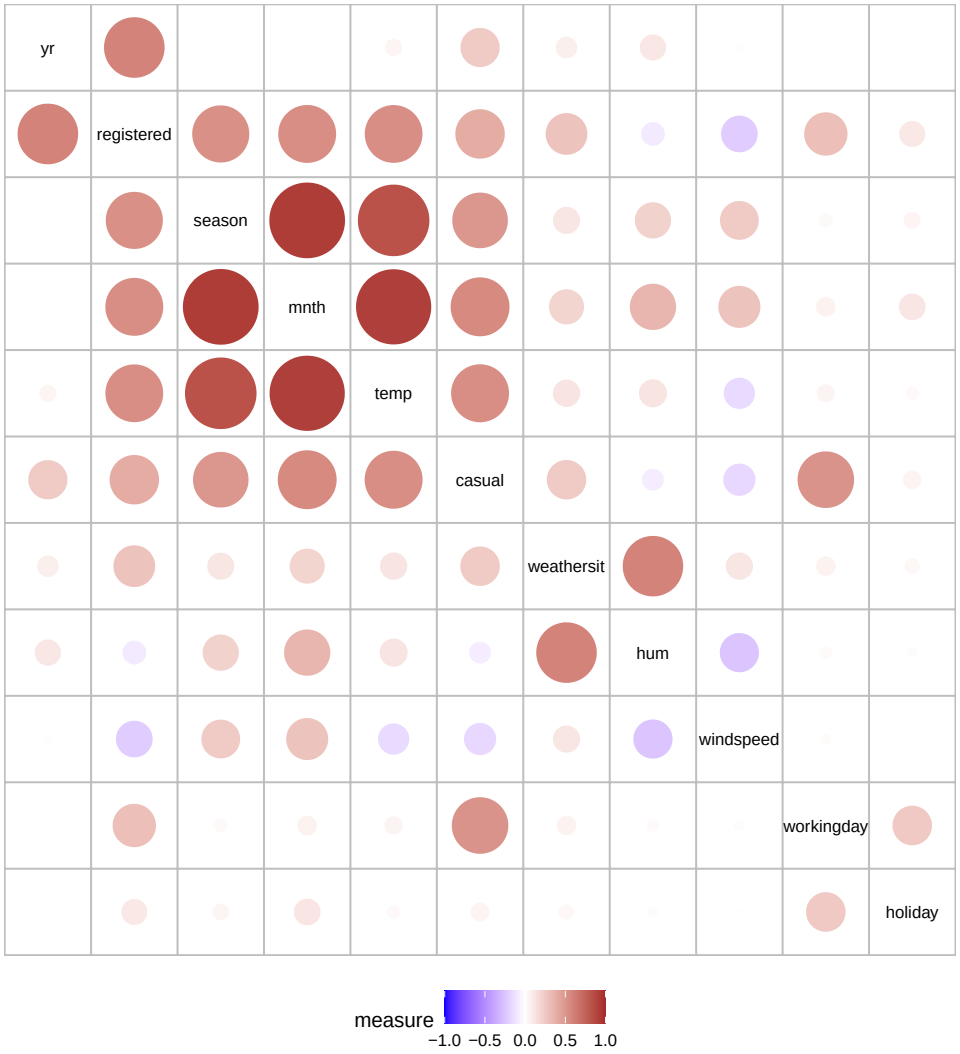


Figure 3.1: Association matrix display for bike sharing data showing Pearson's correlation for the numeric pairs and canonical correlation for factor pairs and mixed pairs. The off diagonal cells show the measure value for a variable pair using a square glyph. The color of every circle is mapped with the measure value for the pair and the area of the circle is mapped to absolute measure value for the corresponding variable pair. The plot shows many pairs 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 3.1 presents the novel feature of our display showing all the variables of a dataset in the same plot compared to displays from `corrgram` (2.2) which only shows association between numeric pairs. With canonical correlation as the association measure for factor pairs or mixed pairs, we can observe from Figure 3.1 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 segment 3.4 shows the implementation and this result is used to produce 3.2 with some factor variables as ordinals.

Figure 3.2 shows a strong negative association for (workingday, holiday) as holidays are not working days.

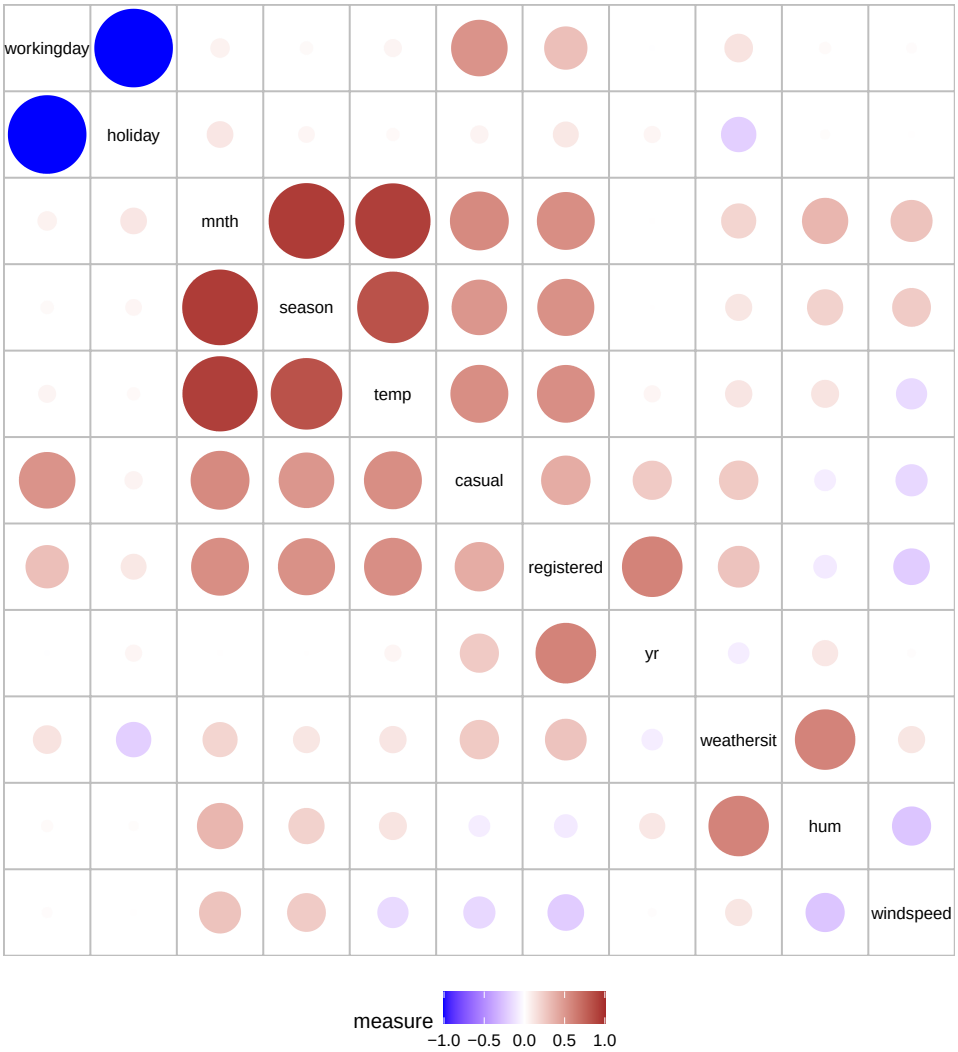


Figure 3.2: Association matrix display for bike sharing data showing Pearson's correlation for the numeric pairs, Goodman Kruskal's gamma measure for ordered pairs and canonical correlation for factor pairs and mixed pairs. The variables workingday, yr, weathersit and holiday have been converted to ordinals. The plot shows a strong negative association for (workingday, holiday) because no holiday is a working day

We use function `show_assoc` to explore associated variable pairs graphically. Figure 3.3 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 days. The barplot for variable pair (workingday, holiday) confirms that no working day was a holiday.

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

We use a seriation algorithm which brings panels with high variation in measures close to the diagonal. This will be discussed in section 3.4. We also order multiple measure types in each cell of the multiple measures display. This locates measure types with high average values at the start of each cell.

Figure 3.4 shows a multiple association measures plot in matrix layout for the 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 `ace`, `cancor`, `dcor`, `kendall`, `mic`, `nmi`, `pearson` and `spearman`.

It is evident from the Figure 3.4 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 3.5. It is evident from the scatterplots that both (casual,mnth) and (mnth,temp) show

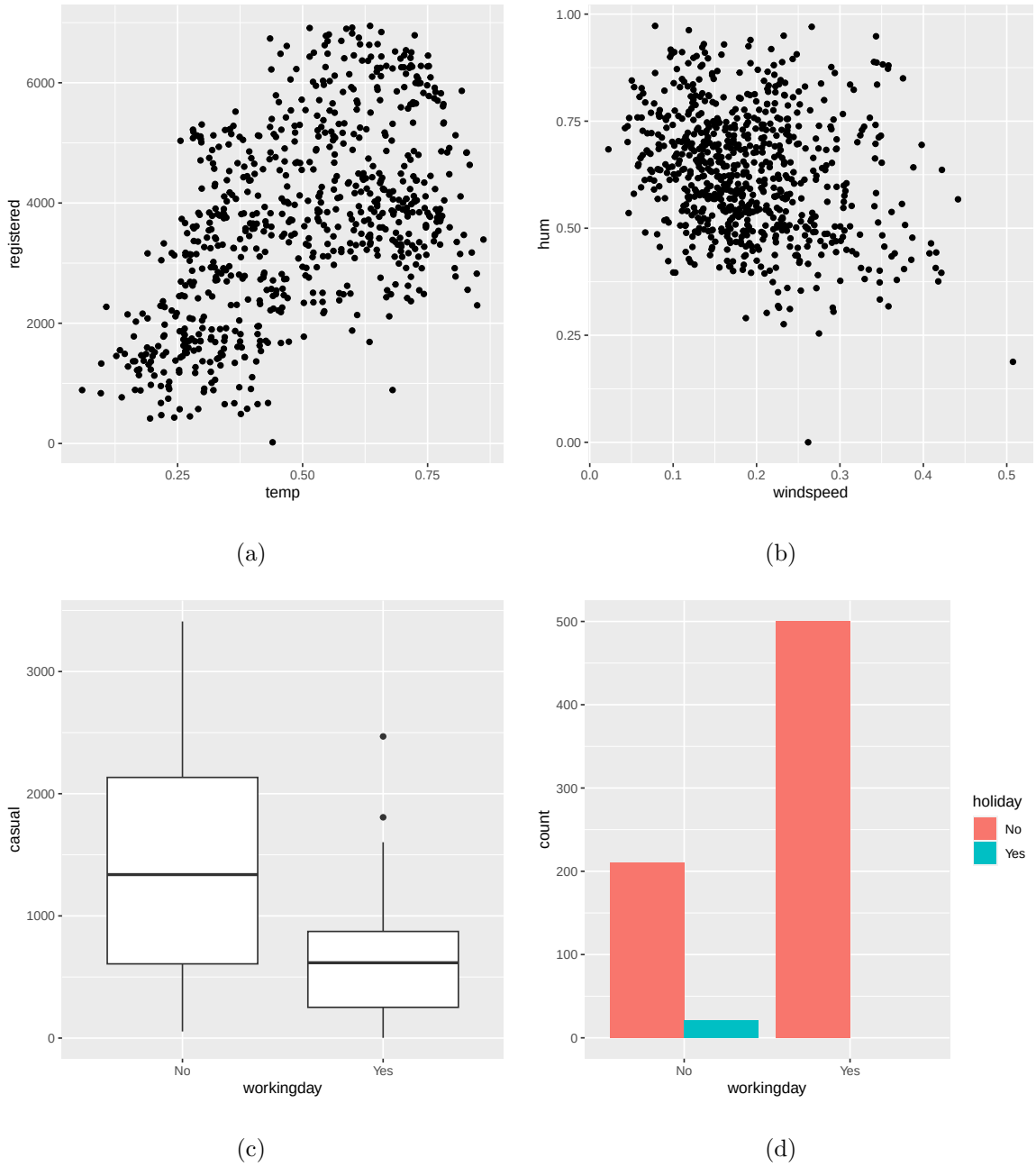


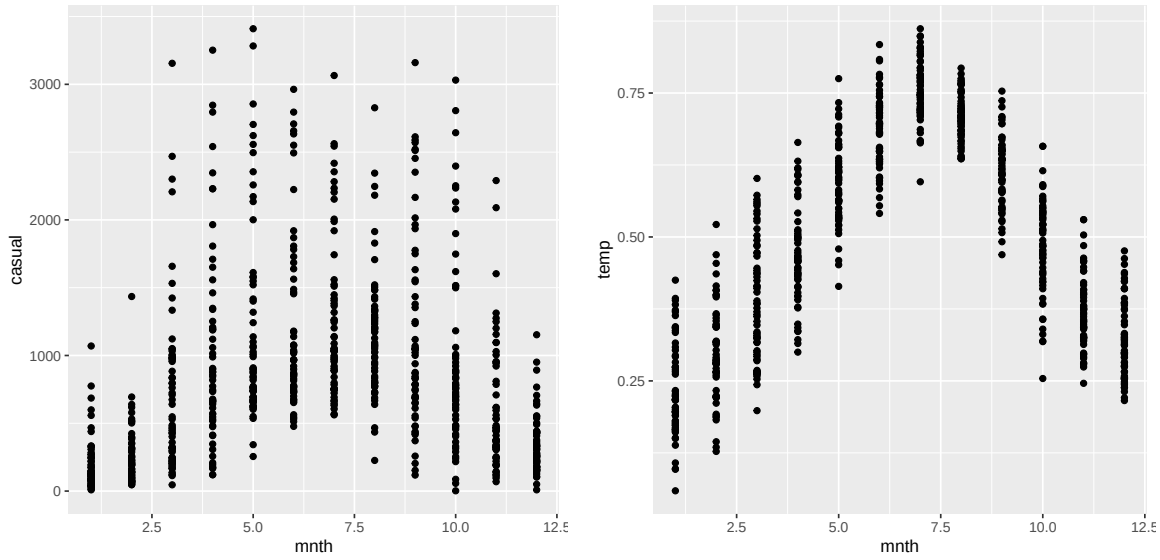
Figure 3.3: Scatterplots for numeric pair (temp,registered) and (windspeed,hum) in (a) and (b) respectively, boxplot for mixed pair (workingday,casual) in (c) and a barplot for the binary pair (workingday, holiday) in (d) showing association between the pairs of variables

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 patterns.

Figure 3.4: 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

3.3.3 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



(a) (b)

Figure 3.5: Scatterplot for variable pairs (casual,mnth) in (a) and (mnth,temp) in (b) showing a non-linear relationship for these pairs

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 a `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 represent the value of the association measure and level of the conditioning variable respectively. The overall (unconditional) value of the association measure is shown as a pink horizontal line.

For ordering the variables, we use a similar strategy as discussed above for matrix displays. The only difference is that the range of association measure values at different levels of the conditioning variable for a variable pair is used for ordering each cell. 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 3.6 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

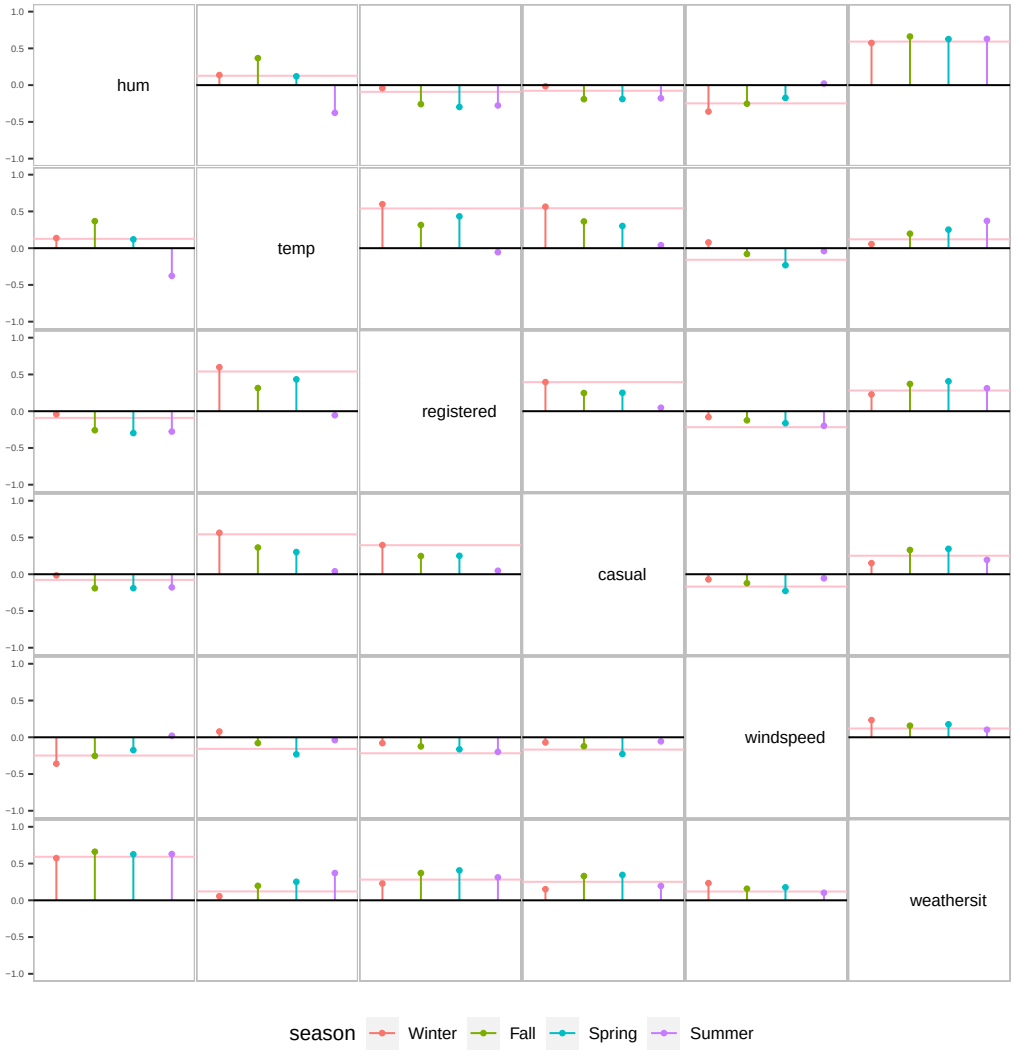


Figure 3.6: Conditional association measures plot for bike sharing data showing Pearson's correlation for numeric pairs and canonical correlation for factor or mixed pairs. The lollipops in each cell represent the value for asssociation measure colored by the conditioning variable season. The pink horizontal 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

and Kruskal's gamma for ordinal pair, canonical correlation for nominal or mixed pairs) calculated at the levels of conditioning variable `season`. 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.

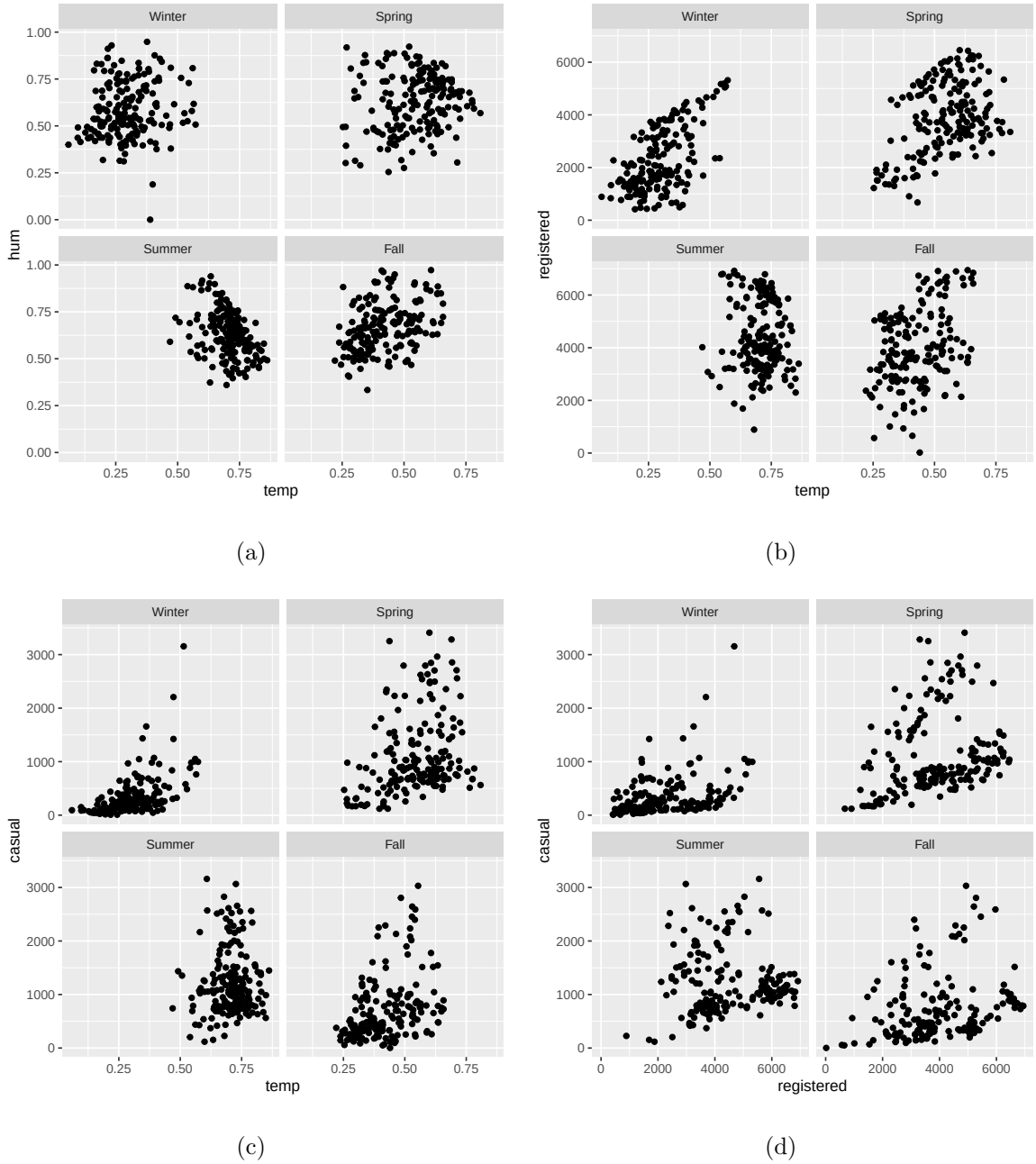


Figure 3.7: Scatterplots for variable pairs (temp, hum), (temp, registered), (temp, casual) and (registered, casual) in (a), (b), (c) and (d) respectively faceted by conditioning variable season

We explore these variable pairs in more detail using `show_assoc`. Figure 3.7 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 tempera-

ture 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.

3.3.4 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.

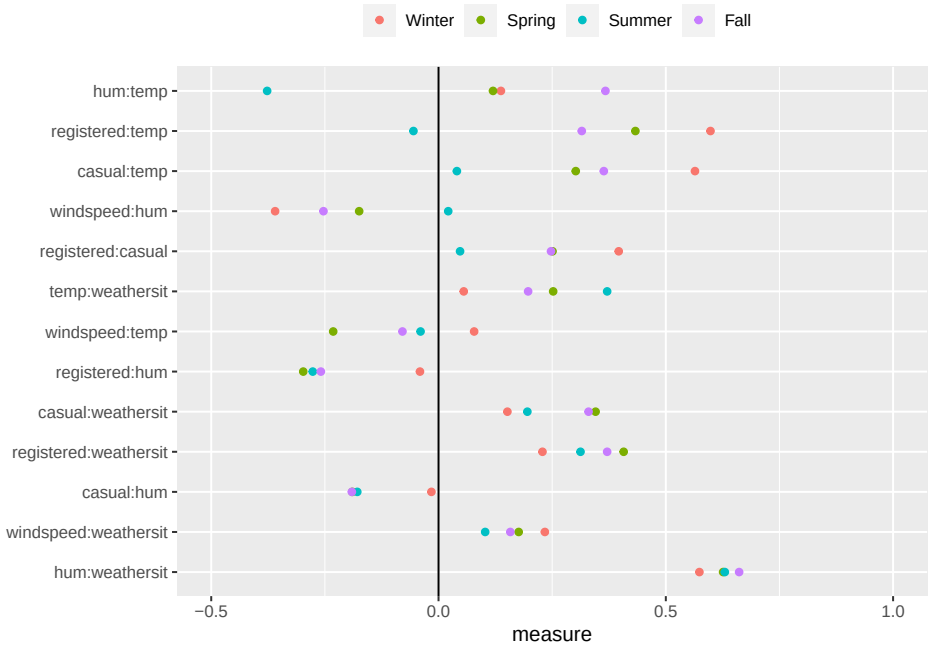


Figure 3.8: Conditional association measures plot for bike sharing data in linear layout.The display has variable pairs on the Y-axis and the value of association measures on the X-axis. The points corresponding to every variable pair represents the value of association measure for different levels of the conditioning variable and the overall value of association measure

Figure 3.8 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. 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 have little effect of conditioning on their association.

Figure 3.8 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.

3.4 Seriation in *corVis*

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 so that groups of variables with high mutual correlation are easily identified and Hurley [2004] who ordered variables in a scatterplot matrix so that interesting panels were positioned close to the main diagonal. All the above cases illustrate how seriation, a term to describe ordering of objects, is useful to reveal interesting patterns. This section focuses on the seriation techniques used for linear and matrix displays in `corVis`.

We use the American Community Survey (2012) from the R package `openintro` [Çetinkaya Rundel et al., 2022] which contains results from the US Census American Community Survey in 2012 in this section. The dataset is a demographics survey

program conducted by the U.S. Census Bureau and includes information on participants' citizenship, educational attainment, income, language proficiency, migration, disability and employment. Table 3.4 provides a description of the variables present.

Variable	Description
inc	Annual income
emp	Employment status with categories not in labor force, unemployed, employed
hrs_w	Hours worked per week
race	Race of the participant with categories white, black, asian or other
age	Age of the participant in years
gen	Gender with categories male or female
citz	Whether the person is a U.S. citizen
time_w	Travel time to work, in minutes
lang	Language spoken at home with categories english or other
marr	Whether the person is married
edu	Education level with categories hs or lower, college, grad
dis	Whether the person is disabled

Table 3.4: Variable description of the acs12 dataset

While exploring the data for missing values, it is found that the dataset includes information about individuals aged between 0 and 2 years. For these individuals, entry for variables `income`, `emp`, `hrs_w`, `time_w`, `lang` and `edu` is `NA`. In `corVis`, we provide functionality for handling missing values or `NA` in data by using pairwise complete observations while calculating the association measures.

3.4.1 Seriation in Linear Displays

We provide linear displays in the form of dot plots or heatmaps for plotting association measures and conditional association measures in `corVis`. In these displays,

the items ordered are variable pairs. We sort the variable pairs in decreasing order by either the maximum or the range between the association measure values for each pair.

We use the maximum absolute value for ordering the variable pairs in multiple measures display. The display of multiple association measures proves beneficial in identifying pairs exhibiting non-linear associations, prompting further detailed exploration. The display contrasts the values of association measures, including ace, cancor, chi, dcor, kendall, mic, nmi, pearson, spearman, and uncertainty, for each variable pair within the dataset. The variable pairs are ordered in descending order by the maximum absolute measure value of the available measures for each pair. This produces a display with highly associated variable pairs for any measure at the top, simplifying the task to identify associated variables.

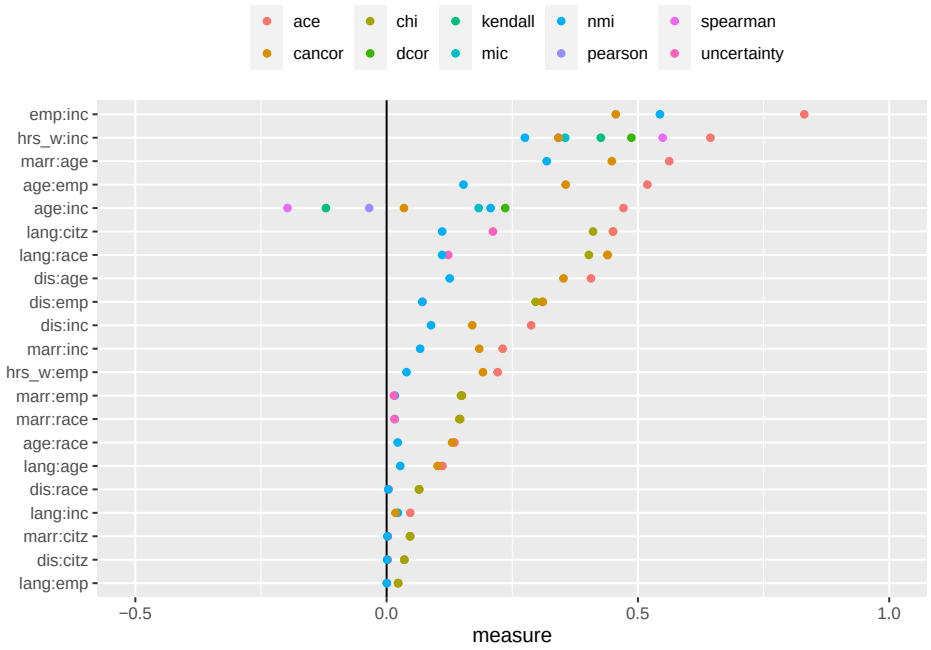


Figure 3.9: Multiple measures display in linear layout with variable pair ordered by maximum absolute measure value. For numeric variable pairs, measures such as ace, dcor, kendall, mic, pearson, spearman have been used to identify pairs with non-linear association

Figure 3.9 shows the seriated multiple measures display in linear layout for `acs12` data where it is easier to find highly associated variable pairs in the plot. The plot shows filtered variable pairs having a maximum measure value of 0.4 or greater. The

variable pair (emp, inc) is placed at the top of the display showing that the income and employment status of an individual is highly associated. This is expected as employed individuals will have a high income compared to unemployed participants. Another highly associated variable pair evident from the plot is (hrs_w,inc) showing that individuals working more hours earn more money.

We use the range of measures to order variable pairs for conditional measures display. 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.

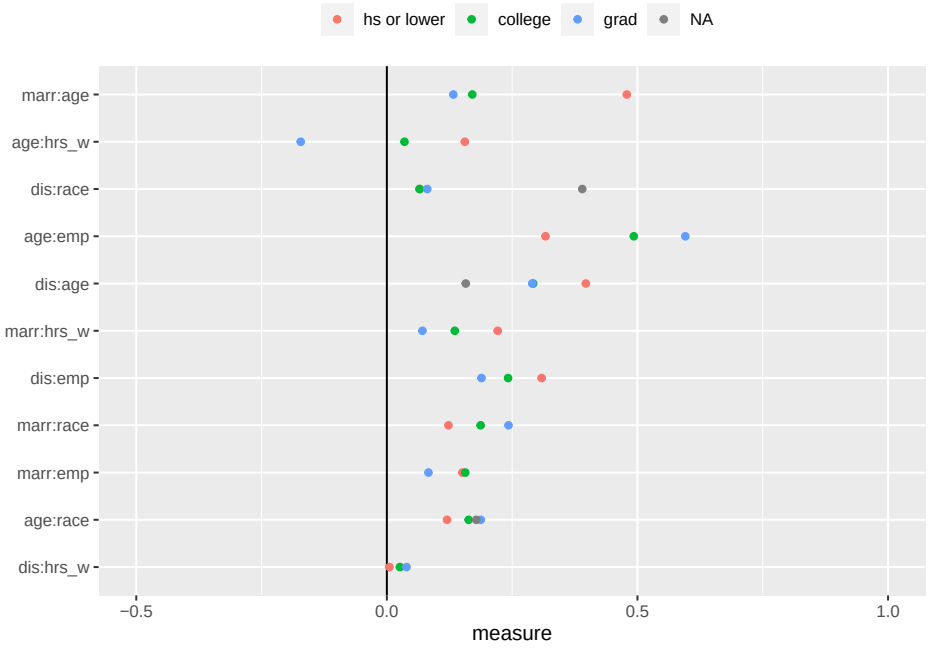


Figure 3.10: Conditional association measure display in linear layout with variable pair ordered by maximum difference value at each level. The plot displays Pearson's correlation for numeric variable pairs and canonical correlation for factor pairs and mixed pairs. It selectively shows variable pairs with a difference in measure equal to or exceeding 0.25

Figure 3.10 shows a seriated conditional measure display using the range of the measures. 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 the numeric variable pairs and canonical correlation for factor pairs and mixed pairs. It

is easier to spot the variable pairs with high differences among measures at different levels of conditioning variable `edu` in the plot. The levels of education in the data are high school or lower, college or graduate. There are also individuals in the data whose education level is missing and are participants who haven't started school yet.

The variable pair (`marr`, `age`) is placed at the top of the display and has the highest difference between the measures at different levels of `edu`. The two variables marriage and age are strongly associated at the education level high school or lower. This is expected as a high proportion of individuals with high school or lower education are usually younger and not married. The canonical correlation for the variable pair at `NA` level of education is not defined as all of the participants with missing education are not married.

3.4.2 Seriation in Matrix Displays

We employ techniques from the `DendSer` package to arrange matrix displays within `corVis`. These techniques implement algorithms outlined in Earle and Hurley [2015]. In these algorithms, the initial step involves generating a dissimilarity or similarity matrix for objects, which are subsequently clustered. In our scenario, these objects represent variables within a dataset. We utilize the Lazy Path Length (LPL) cost function from the `DendSer` package to establish an order based on the dissimilarity matrix of variables. The LPL method is essentially a variation of the traveling salesman problem, aiming to minimize the overall distance traveled while also prioritizing the delay of longer distances, indicating a preference for shorter ones. This approach proves effective in highlighting interesting pairs in matrix displays by positioning them prominently at the beginning and top-left position.

Let π be an order obtained from the hierarchical clustering of a matrix with n variables using the seriation weights w_{ij}. Then, LPL cost function for an order π is defined as:

$LPL(\pi) = \sum_{i=1}^{n-1}(n-1)w_{\pi(i),\pi(i+1)}$

LPL is a weighted measure of path length and rewards orders with short path lengths and where the weights generally increase.

Below is a code snippet showing how the `dser` function from package `DendSer` is used to obtain an ordering for the correlation matrix of numeric variables in the

`acs12` data. The dissimilarity between variable pairs is measured by $|-\text{Pearson correlation}|$ and a dissimilarity matrix is constructed.

```
acs12_num <- select(acs12,where(is.numeric))
names(acs12_num)
[1] "inc"    "hrs_w"  "age"    "time_w"
m <- cor(acs12_num,use = "pairwise.complete.obs")
o <- DendSer::dser(as.dist(-abs(m)),cost = DendSer::costLPL)
names(acs12_num)[o]
[1] "inc"    "hrs_w"  "time_w" "age"
```

Listing 3.7: R code showing how dser function from DendSer apckage is used for ordering

Below we provide a general overview of the seriation algorithm presented in `DendSer` and LPL cost function to understand how seriation is implemented. The algorithm uses a seriation weight w_{ij} for variables i and j, which measures the importance of a cell, in the matrix and uses these weights to perform hierarchical clustering. Generally, the weights are measures of dissimilarity between the variables. The final step is to rearrange the nodes of the dendrogram obtained from clustering such that ordering minimises a cost function. This produces an arrangement where associated variable pairs are placed adjacent or nearby each other. As people generally read from left to right, placing the most important cells at the beginning or top-left corner of the display allows an analyst to immediately identify important pairs of variables. We use the LPL cost function to achieve this.

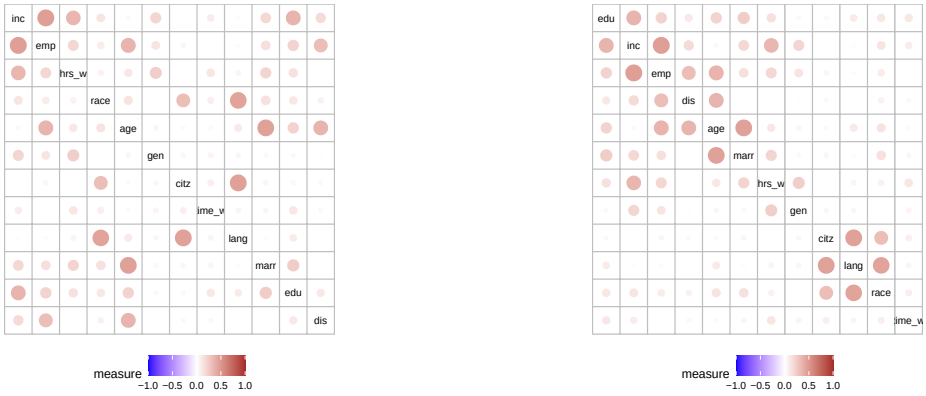
We use the sorting approach discussed above for all of our matrix displays to obtain variable ordering.

Association Measure Display

The association measure display plots a measure of association for variable pairs in a dataset. The plotting function `plot_assoc_matrix` takes a data structure with variable pairs and corresponding measures as input. For ordering the variables, 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.

Figure 3.11 compares the default ordering of the variables in the dataset with the ordering obtained by seriation using the LPL cost function. The plot shows Pearson's correlation for numeric 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. For instance, the variable pairs ('edu','inc') and ('inc', 'emp') are highly associated and are easy to discover in the plot on right compared to the plot on the left. This shows that individuals who are graduates and employed earn more money compared to other individuals.



(a) (b)

Figure 3.11: Association measure display for acs12 data. (a) variables in default order of the data; (b) variables ordered by LPL cost function

A user can also supply their own ordering perhaps obtained from other algorithm by specifying `var_order` argument in the `plot_assoc_matrix` function.

Multiple Measures Display

The multiple measures display plots multiple measures of association for every variable pair in the dataset. 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 3.12: Seriated multiple association measures display for acs12 data. Variable pairs at top left or along diagonal of the display have a high value for any of the multiple measures

Figure 3.12 shows a seriated multiple measures display. The plot displays variable pairs with high measure value(s) in the top left corner or along the diagonal of the display, making it easier to find pairs of variables where any of the measures is high. The plot shows that variable pairs (hrs_w, emp) and (inc, emp) are two of the highly associated pairs with a high value for `ace` measure. This is expected as income of an individual generally depend on the number of hours worked and their employement status.

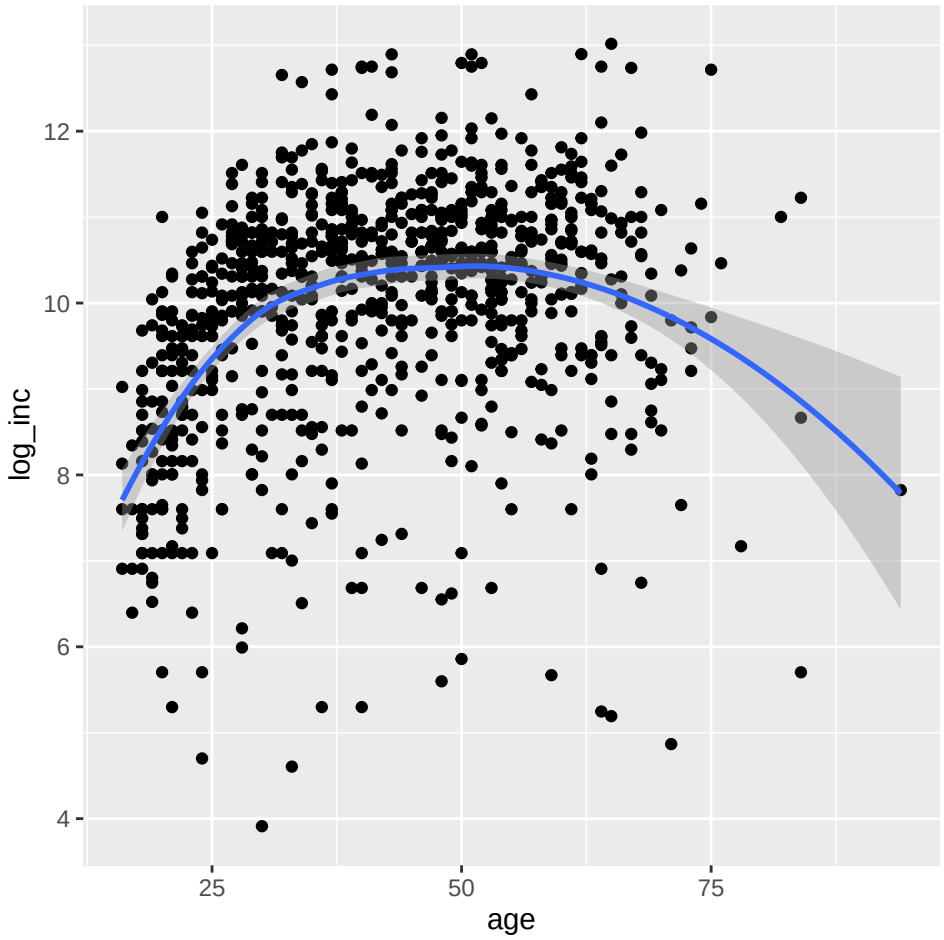


Figure 3.13: Scatterplot for variable pair (log_inc, age) showing non-linear pattern with loess smoothing function

The variable pair (inc, age) in Figure 4 is also placed close to the diagonal and shows a high value for measure `ace` compared to other measures of association. A closer look at the transformed variable pair using a scatterplot in Figure 3.13 shows the presence of non-linear association which is captured by `ace` measure.

Conditional Association Measures Display

The conditional association measure display plots pairwise association measures at different levels of a conditioning variable. 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, we follow a similar approach used

for ordering measure types in multiple measures display.

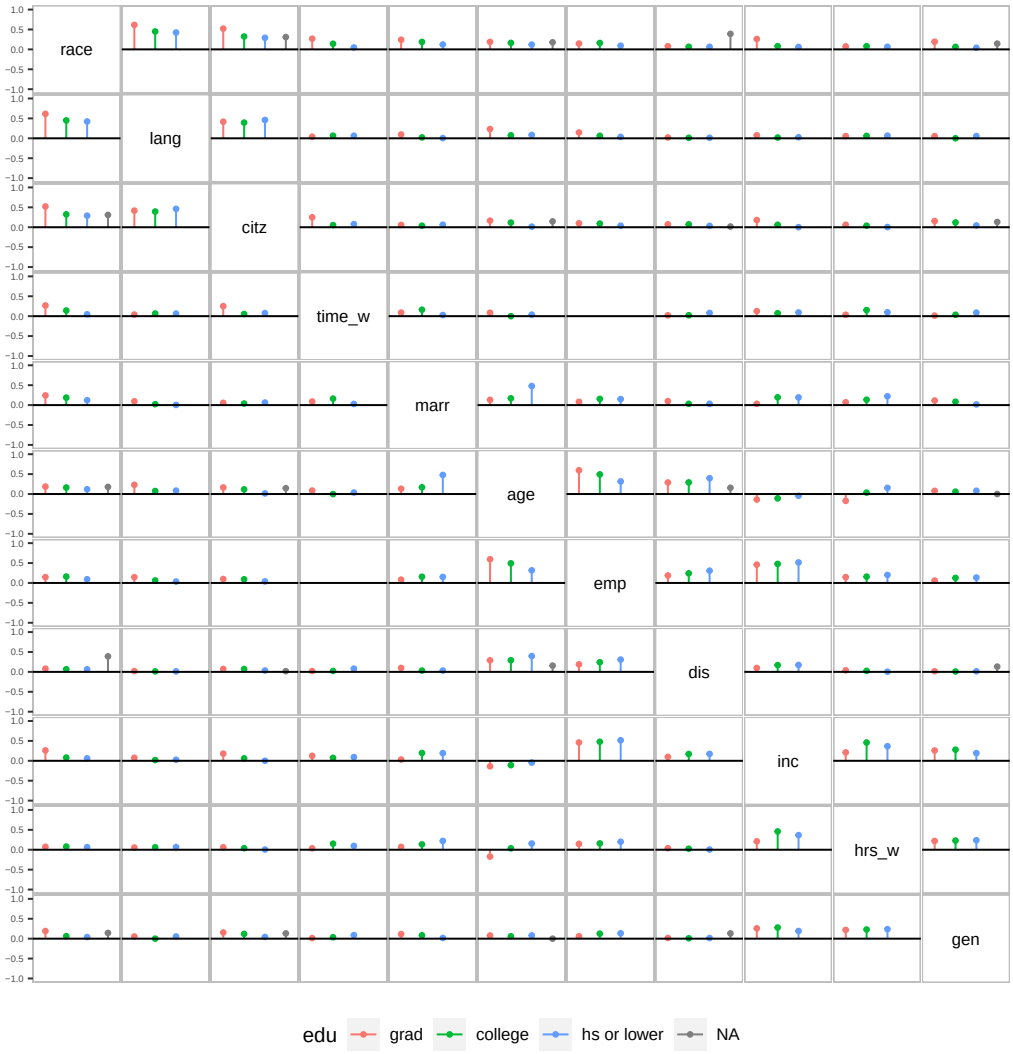


Figure 3.14: Seriated conditional association measures display for acs12 data. Variable pairs at top left or along diagonal of the display have a high difference in the measures value calculated at different levels of the conditioning variable

Figure 3.14 shows the seriated conditional association measure display. The variable pairs with high differences in measure value for a grouping variable are placed at the beginning or along the diagonal of the display. This helps in quickly finding pairs with high group differences. The plot shows variable pair (age, emp) with high differences in canonical correlation value at different levels of the conditioning variable edu.

CHAPTER 4

Exploring patterns using *corVis*

4.1 Simpson's Paradox

The change in the direction of an association/trend when the data is divided into subgroups shows the existence of Simpson's paradox [Simpson, 1951] in a dataset. The classic example of a gender discrimination suit against the University of California, Berkeley [Bickel et al., 1975] exhibits how one can lead to wrong results. The data showed a significant difference in the number of males to the number of females who got admitted to the University until the data is split by the departments to which candidates had applied. This disaggregation of data showed the existence of Simpson's paradox. In this case, the reason behind its occurrence was that more female than male candidates applied to the departments where it was hard to get admission.

Simpson's paradox is observed in a regression setting when the relationship between two variables reverses upon disaggregation by a third variable, which may act as a confounding variable. Pearson's correlation can be employed to detect instances of Simpson's paradox by capturing the reversal of linear trends. By analyzing triples of variables using Pearson's correlation, we can identify cases where

Simpson's paradox is present.

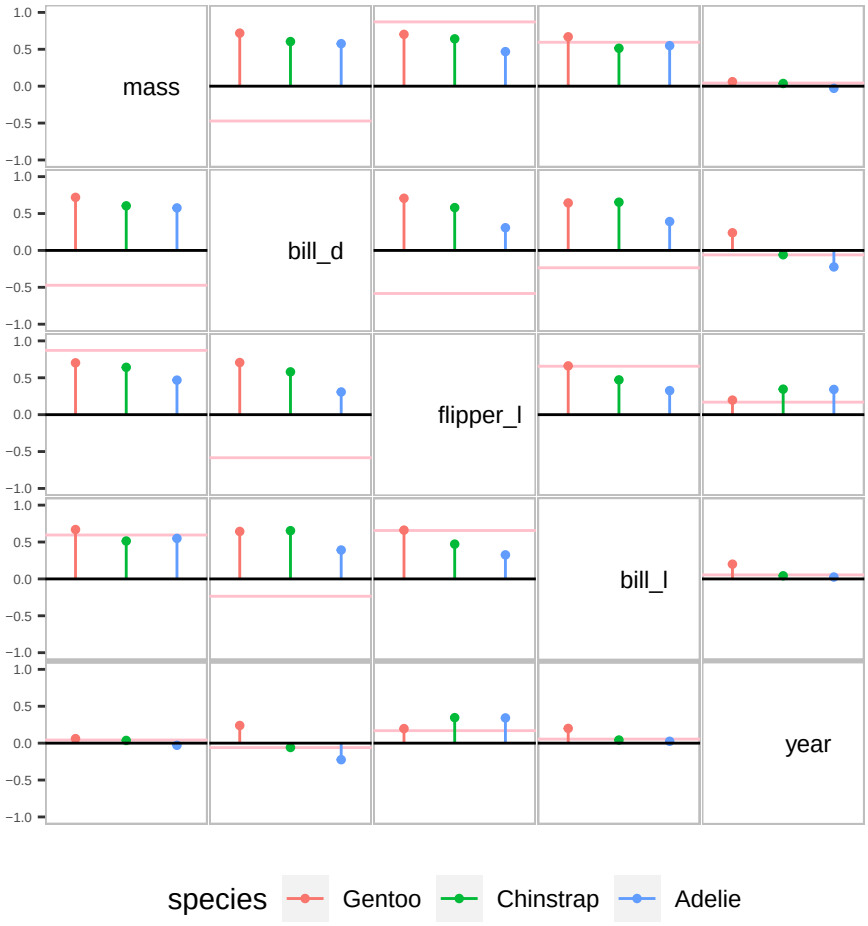


Figure 4.1: Conditional measures plot showing pairs of variables with Simpson's paradox. It can be seen clearly that variable pairs (mass, bill_d),(bill_d, flipper_l) and (bill_d, bill_l) show Simpson's paradox

The package `corVis` offers conditional association measures display that can be utilized to identify variable pairs exhibiting Simpson's paradox. As demonstrated in Figure 4.1, the conditional measures plot showcases the overall Pearson's correlation with a pink horizontal line in each cell, while lollipops illustrate correlations at different levels of the grouping variable for the corresponding variables.

From the plot, it becomes evident that variable pairs such as mass and bill depth, bill depth and flipper length, and, bill depth and bill length demonstrate instances of Simpson's paradox. In these cases, the overall trend of the correlation reverses when the data is disaggregated by the grouping variable, which, in this specific example, is the species.

In summary, by utilizing the conditional association measures provided in `corVis`, researchers can effectively identify and visualize Simpson's paradox in variable pairs where the observed associations change direction when considering different subsets based on the grouping variable.

4.2 Scagnostics

Scagnostics was Paul and John Tukey's [Friedman and Stuetzle, 2002] idea of defining measures to characterize a scatterplot. Their idea was to plot $p \times (p-1)/2$ scatterplots, where p is the number of variables, as points in a scatterplot matrix of k measures which can be used to find subsets of interesting scatterplots. They suggested that the possible measures for a scatterplot could be density, shape, trend, outliers.

Wilkinson et al. [2005] proposed graph-theoretic scagnostic measures based on three geometric graphs: the convex hull, alpha hull and minimum spanning tree (MST) to highlight unusual scatterplots. They presented nine measures which quantified the density, shape, trend or outliers in a scatterplot. These nine measures are:

- Outlying: Quantifies the presence of extreme outliers within the scatterplot.
- Skewed: Measures the skewness in a scatterplot.
- Clumpy: Evaluates the level of clustering of data points.
- Sparse: Quantifies the sparsity nature of data points.
- Striated: Measures the presence of stripes in the scatterplot.
- Convex: Quantifies the shape of points.
- Skinny: Measures how much skinny a plot is.
- Monotonic: Measures trend in a scatterplot.

4.2.1 Calculating scagnostics

The package *corVis* has a function `tbl_scag` for calculating scagnostic measures in a tibble structure for the numeric variable pairs in a dataset. The `tbl_scag`

function calculates scagnostic measures using the `scagnostics` package [Wilkinson and Anand, 2022] in R.

`tbl_scag` uses a dataset and a scagnostic measure as its two main inputs and outputs a tibble with the variable pairs and calculated scagnostic measure. By default, the scagnostic measure calculated is outlying which quantifies the presence of outliers in a scatterplot. We also provide functionality for handling missing values by using pairwise complete observations.

`tbl_scag` returns a tibble with the variable pairs and calculated measure, and also with additional classes `pairwise` and `data.frame`. With the pairwise measures of association in a tibble or dataframe structure, the outputs are used with packages like `dplyr` , `ggplot2` for further exploration of these scagnostic measures.

4.2.2 Visualising scagnostics

Once the scagnostic measures have been calculated, these are visualised in different layouts using functions in `corVis`. `association_heatmap` function is used to produce a matrix layout of a scagnostic measure for every numeric variable pair in the dataset. Figure 4.2 shows the display of monotonic scores for numeric variable pairs in the penguins dataset from the `palmerpenguins` package. The area of the circular glyph is proportional to the magnitude of the measure. It is clearly evident from Figure 4.2 that flipper length and body mass of penguins have a montonic relationship.

Comparing scagnostics

Following Tukey's idea of a scatterplot matrix display of the measures, we propose a linear display for comparison of multiple scagnostic measures for all the numeric variable pairs in a dataset. This is useful in identifying unusual scatterplots or variable pairs.

`plot_assoc_linear` produces a linear display for comparing multiple scagnostic measures. It takes the calculated scagnostic measures as input and displays the variable pairs on the Y-axis and the scagnostic measures on the X-axis. Figure 4.3 shows a comparison of scagnostic measures for penguins dataset. The plot displays nine colored dots representing the nine scagnostic measures for each pair of variables.

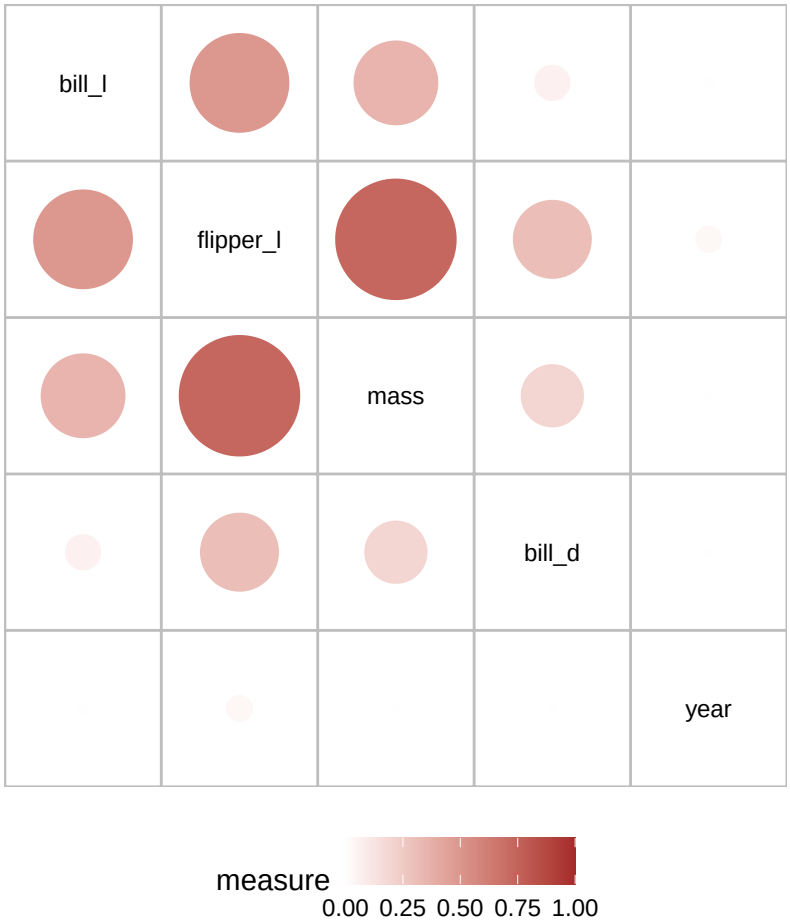


Figure 4.2: Matrix display of monotonic measure for penguins data

The variable pair flipper length and bill depth in Figure 4.3 shows a high value for clumpy and monotonic measures suggesting a presence of clusters of points and a monotonic relationship between these two variables. The low value for clumpy and a high value for monotonic scagnostic measure for the variable pair body mass and flipper length suggests a strong association among these two variables for the penguins of every species.

Conditional scagnostics

The function `calc_assoc` is used to calculate pairwise scagnostic measures at different levels of a categorical conditioning variable. This helps in finding out interesting variable triples which can be explored further prior to modeling. Figure 4.4 shows a conditional plot for the penguins data. Each cell corresponding to a variable pair shows three bars which correspond to the clumpy scagnostic measure calculated at

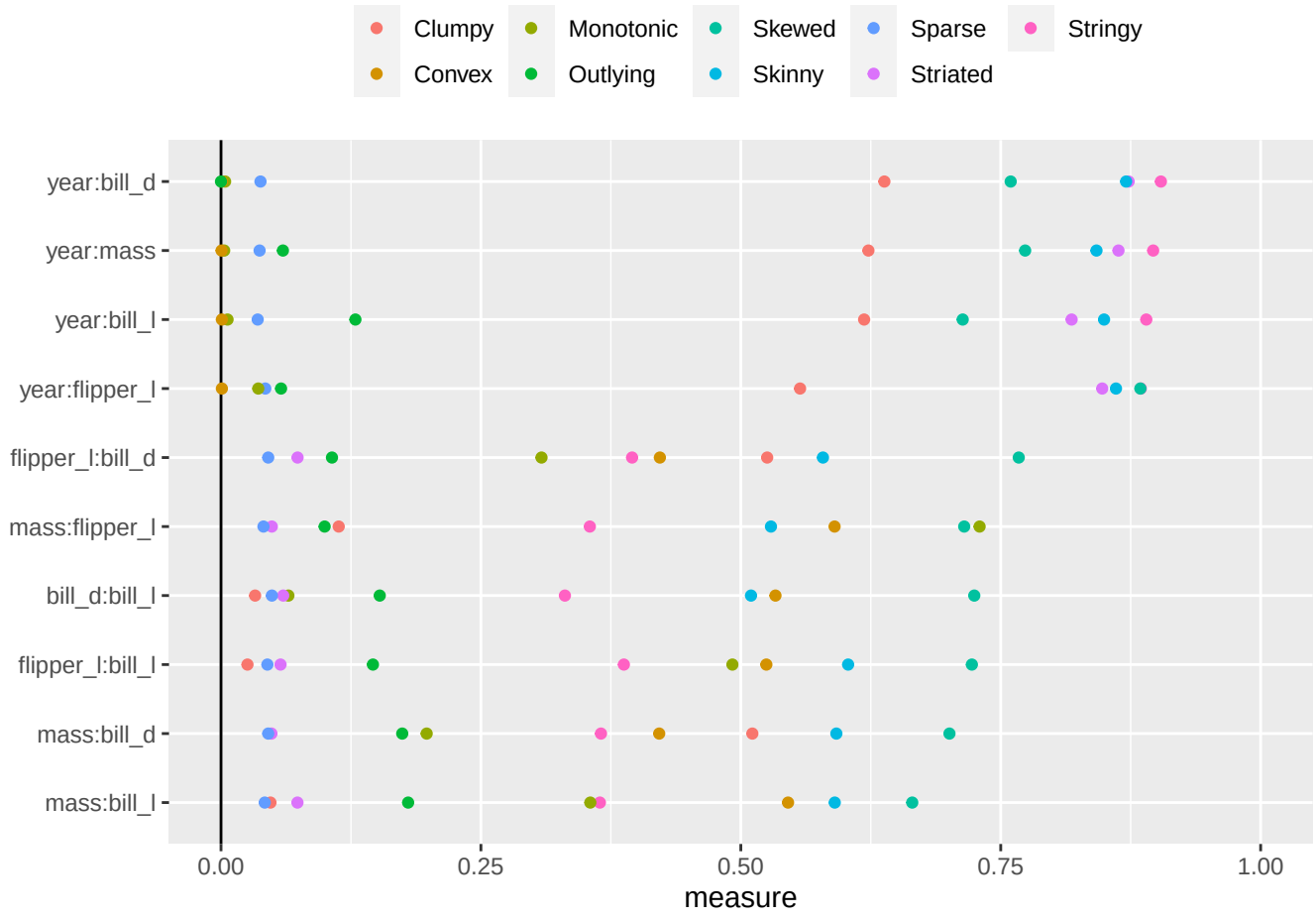


Figure 4.3: Comparison of multiple scagnostic measures using a linear layout

the levels of conditioning variable `species`. The dotted line represents the overall value of clumpy measure.

Figure 4.4 shows that there is a high value for clumpy for variable pairs flipper length and bill depth, and, body mass and bill depth suggesting a presence of cluster of data points for these pairs. The low values of clumpy for both of these pairs at different levels of species shows that different species of penguins tend to have different pairwise values for flipper length and bill depth, and, body mass and bill depth.

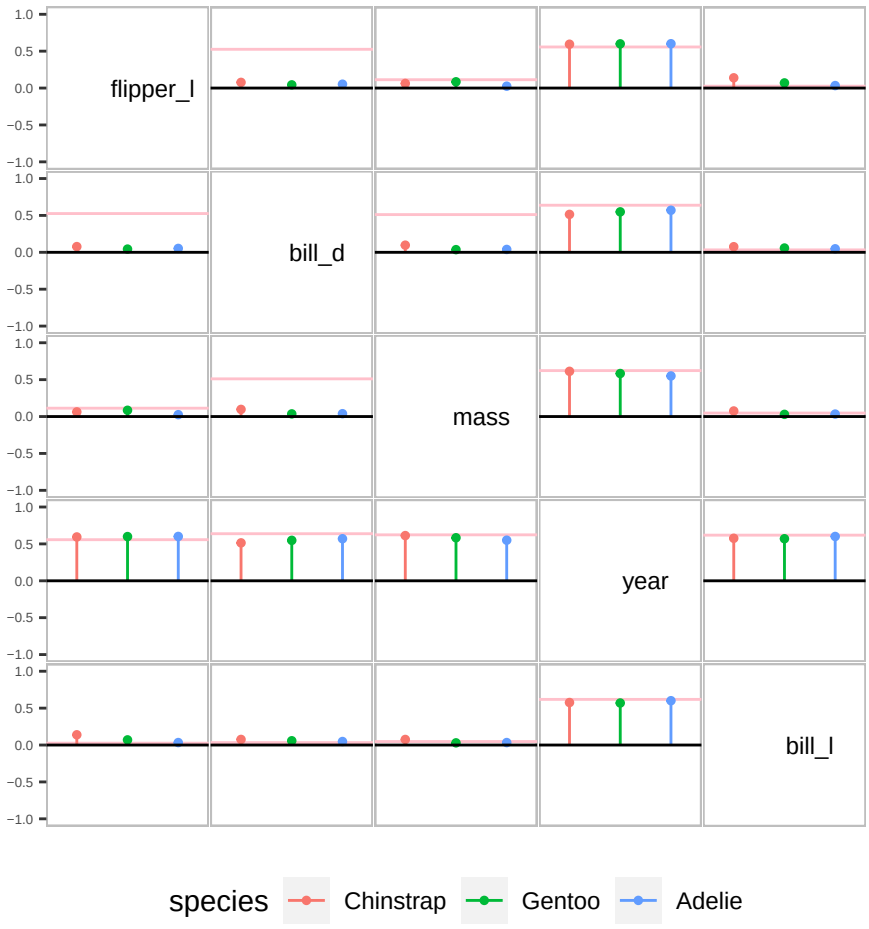


Figure 4.4: Conditional scagnostics plot showing the value of clumpy scagnostic for various pairs of variables in the penguins data

CHAPTER 5

Conclusion and Further Work

5.1 Summary

This thesis introduces a novel method for visualizing correlation, association measures or other pairwise measures such as scagnostics, emphasizing the utilization of multiple measures and conditional measures. The focus is on accommodating the diverse nature of datasets, which often consist of numerical, ordinal, and categorical variables. The aim is to offer a comprehensive display that explores the association between each variable pair in the dataset using measures suitable for different variable types.

The visualization method is implemented in R within the corVis package. Two distinct layouts are employed to display the association measures. The first layout adopts a matrix-style approach, similar to existing correlation matrix displays. However, this version stands out as it can exhibit multiple association measures for each variable pair, unveiling patterns beyond linear associations or those dependent on a grouping variable's level.

For datasets with high dimensionality, matrix layouts can become cumbersome and overcrowded. Therefore, the second layout adopts a linear design, presenting one

or more association measures for each variable pair. This becomes particularly useful when the analyst intends to focus on pairs of variables with significant associations. Linear representations are also valuable when investigating the relationship between the response variable and the exploratory variables.

Similar to the work of Friendly [2002], which utilized ordered correlation displays for identifying groups of variables with strong mutual correlation, this thesis incorporates seriation for matrix displays and importance sorting for linear displays. The objective in both cases is to position highly-associated variables or variable pairs with notable differences in prominent locations, enhancing their visibility and facilitating identification.

The implementations presented in this thesis are available in R package `corVis` which is hosted at github.com/chinwan16/corVis.

5.2 Future Work

This thesis introduces research tools designed to facilitate the exploration of patterns within a dataset using association measures during data analysis. In line with this objective, there are opportunities for further improvements and newly emerged research directions to pursue. This section offers recommendations for future work that hold promise and are worth investigating.

In Section 3.3.3, conditional association plots are generated based on a grouping variable, which is a factor or an ordered factor in this thesis. Extending our method to handle continuous variables as grouping variables would be a logical progression. While our current approach seamlessly handles categorical grouping variables, dealing with continuous variables requires creating discrete partitions using disjoint binning techniques or overlapping bins (shingles). Future research could explore this process to identify meaningful bins for a continuous grouping variable, particularly in cases where interesting patterns are present.

Circular variables are data types that capture measurements in a circular or periodic fashion, signifying values situated on a circle or within a periodic interval. Examples of such variables include time of day, season of the year, weekday, angles, or compass directions. In this research, however, we do not specifically address cir-

cular variables. Although there is an association measure called `ace` that handles circular variables, we have not utilized this feature in this thesis. To expand the analysis capabilities of the package, future research could focus on exploring association measures specifically designed for the analysis of circular variables. Incorporating such measures into the package would enhance its functionality and allow for a more comprehensive analysis of datasets containing circular data.

5.3 Conclusion

In this thesis, a novel approach is introduced to visualize association and conditional association by utilizing measures of association. The method offers both a matrix and linear layout, enabling the display of bivariate associations, potentially grouped by levels of a categorical variable. Moreover, the approach employs measures that are suitable for numerical, ordinal, and nominal variables. By employing these visual displays, analysts can obtain a comprehensive overview of the intriguing structure and underlying patterns present in the data. This visualization technique facilitates the exploration and interpretation of complex relationships within the dataset, aiding in the discovery of valuable insights and meaningful associations.

Bibliography

Alan Agresti. *Analysis of ordinal categorical data*, volume 656. John Wiley & Sons, 2010.

Davide Albanese, Michele Filosi, Roberto Visintainer, Samantha Riccadonna, Giuseppe Jurman, and Cesare Furlanello. Minerva and minepy: a c engine for the mine suite and its r, python and matlab wrappers. *Bioinformatics*, page bts707, 2012.

Signorell Andri et mult. al. *DescTools: Tools for Descriptive Statistics*, 2022. URL `https://cran.r-project.org/package=DescTools`. R package version 0.99.47.

Peter J Bickel, Eugene A Hammel, and J William O'Connell. Sex bias in graduate admissions: Data from berkeley: Measuring bias is harder than is usually assumed, and the evidence is sometimes contrary to expectation. *Science*, 187(4175):398–404, 1975.

Leo Breiman and Jerome H Friedman. Estimating optimal transformations for multiple regression and correlation. *Journal of the American statistical Association*, 80(391):580–598, 1985.

Andreas Buja, Abba M Krieger, and Edward I George. A visualization tool for mining large correlation tables: The association navigator., 2016.

Michael Clark. *A Comparison of Correlation Measures*, 2013. URL `https://m-clark.github.io/docs/CorrelationComparison.pdf`.

Matt Dancho. *correlationfunnel: Speed Up Exploratory Data Analysis (EDA) with the Correlation Funnel*, 2020. URL `https://CRAN.R-project.org/package=correlationfunnel`. R package version 0.2.0.

Matt Dancho and Davis 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.

Denise Earle and Catherine B Hurley. Advances in dendrogram seriation for application to visualization. *Journal of Computational and Graphical Statistics*, 24(1): 1–25, 2015.

Hadi Fanaee-T and Joao Gama. Event labeling combining ensemble detectors and background knowledge. *Progress in Artificial Intelligence*, 2:113–127, 2014.

John Fox. *polycor: Polychoric and Polyserial Correlations*, 2022. URL `https://CRAN.R-project.org/package=polycor`. R package version 0.8-1.

Jerome H Friedman and Werner Stuetzle. John w. tukey's work on interactive graphics. *The Annals of Statistics*, 30(6):1629–1639, 2002.

Michael Friendly. Corrgrams: Exploratory displays for correlation matrices. *The american statistician*, 56(4):316–324, 2002.

Katrin Grimm. Kennzahlenbasierte grafikauswahl. 2017.

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

Allison Marie Horst, Alison Presmanes Hill, and Kristen B Gorman. *palmerpenguins: Palmer Archipelago (Antarctica) penguin data*, 2020. URL `https://allisonhorst.github.io/palmerpenguins/`. R package version 0.1.0.

Harold Hotelling. Relations between two sets of variates. In *Breakthroughs in statistics: methodology and distribution*, pages 162–190. Springer, 1992.

Catherine B Hurley. Clustering visualizations of multidimensional data. *Journal of Computational and Graphical Statistics*, 13(4):788–806, 2004.

Catherine B. Hurley and Denise Earle. *DendSer: Dendrogram Seriation: Ordering for Visualisation*, 2022. URL `https://CRAN.R-project.org/package=DendSer`. R package version 1.0.2.

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

Justin B Kinney and Gurinder S Atwal. Equitability, mutual information, and the maximal information coefficient. *Proceedings of the National Academy of Sciences*, 111(9):3354–3359, 2014.

Max Kuhn, Simon Jackson, and Jorge Cimentada. *corrr: Correlations in R*, 2020. URL `https://CRAN.R-project.org/package=corrr`. R package version 0.4.3.

Sean McKenna, Miriah Meyer, Christopher Gregg, and Samuel Gerber. s-corrplot: an interactive scatterplot for exploring correlation. *Journal of Computational and Graphical Statistics*, 25(2):445–463, 2016.

Pawel Morgen and Przemyslaw 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.

Duncan J Murdoch and ED Chow. A graphical display of large correlation matrices. *The American Statistician*, 50(2):178–180, 1996.

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

David N Reshef, Yakir A Reshef, Hilary K Finucane, Sharon R Grossman, Gilean McVean, Peter J Turnbaugh, Eric S Lander, Michael Mitzenmacher, and Pardis C Sabeti. Detecting novel associations in large data sets. *science*, 334(6062):1518–1524, 2011.

Maria Rizzo and Gabor Szekely. *energy: E-Statistics: Multivariate Inference via the Energy of Data*, 2022. URL `https://CRAN.R-project.org/package=energy`. R package version 1.7-11.

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

Noah Simon and Robert 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`.

Edward H Simpson. The interpretation of interaction in contingency tables. *Journal of the Royal Statistical Society: Series B (Methodological)*, 13(2):238–241, 1951.

Phil Spector, Jerome Friedman, Robert Tibshirani, Thomas Lumley, Shawn Garbett, and Jonathan Baron. *acepack: ACE and AVAS for Selecting Multiple Regression Transformations*, 2016. URL `https://CRAN.R-project.org/package=acepack`. R package version 1.4.1.

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

Alexander Strehl and Joydeep Ghosh. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *Journal of machine learning research*, 3 (Dec):583–617, 2002.

Gábor J Székely, Maria L Rizzo, and Nail K Bakirov. Measuring and testing dependence by correlation of distances. *The annals of statistics*, 35(6):2769–2794, 2007.

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

Taiyun Wei and Viliam Simko. *R package 'corrplot': Visualization of a Correlation Matrix*, 2021. URL `https://github.com/taiyun/corrplot`. (Version 0.92).

Hadley Wickham. *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York, 2016. ISBN 978-3-319-24277-4. URL `https://ggplot2.tidyverse.org`.

Hadley Wickham, Mara Averick, Jennifer Bryan, Winston Chang, Lucy D'Agostino McGowan, Romain François, Garrett Grolemund, Alex Hayes, Lionel Henry, Jim Hester, Max Kuhn, Thomas Lin Pedersen, Evan Miller, Stephan Milton Bache,

Kirill Müller, Jeroen Ooms, David Robinson, Dana Paige Seidel, Vitalie Spinu, Kohske Takahashi, Davis Vaughan, Claus Wilke, Kara Woo, and Hiroaki Yutani. Welcome to the tidyverse. *Journal of Open Source Software*, 4(43):1686, 2019. doi:10.21105/joss.01686.

Lee Wilkinson and Anushka Anand. *scagnostics: Compute scagnostics - scatterplot diagnostics*, 2022. URL `https://CRAN.R-project.org/package=scagnostics`. R package version 0.2-6.

Leland Wilkinson, Anushka Anand, and Robert Grossman. Graph-theoretic scagnostics. In *Information Visualization, IEEE Symposium on*, pages 21–21. IEEE Computer Society, 2005.

Mine Çetinkaya Rundel, David Diez, Andrew Bray, Albert Y. Kim, Ben Baumer, Chester Ismay, Nick Paterno, and Christopher Barr. *openintro: Data Sets and Supplemental Functions from 'OpenIntro' Textbooks and Labs*, 2022. URL `https://CRAN.R-project.org/package=openintro`. R package version 2.4.0.

APPENDIX A

Package Documentation

The documentation for `corVis` is included in this appendix.

Package 'corVis'

October 23, 2023

Title Visualising association measures and conditional association measures

Version 0.0.0.9000

Description 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.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.3

Imports stats, ggplot2, dplyr, magrittr, DescTools, forcats, labeling, energy, correlation, minerva, polycor, rlang, linkspotter, tidyverse, acepack, kableExtra, corrplot, ggraph, igraph, DendSer, openintro, scagnostics, wdm, Hmisc

Suggests rmarkdown, knitr, palmerpenguins, NHANES, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

URL https://chinwan16.github.io/corVis/

R topics documented:

`assoc_tibble` *A generic function to create a data structure for every variable pair in a dataset*

Description

Creates a data structure for every variable pair in a dataset.

Usage

```
assoc_tibble(data, measure_type = "?", pair_type = "?")

## S3 method for class 'matrix'
assoc_tibble(data, measure_type = "?", pair_type = "?")

## S3 method for class 'data.frame'
assoc_tibble(data, measure_type = NA_character_, pair_type = NA_character_)
```

Arguments

`data`	A dataframe.
`measure_type`	a character string indicating the measure of association.
`pair_type`	a character string specifying the type of variable pair.

Value

A data structure for pairs of variables with a column `measure` for measure value, `measure_type` for a type of association measure and `pair_type` for the variable pair.

Methods (by class)

- `assoc_tibble(matrix)`: assoc_tibble method
- `assoc_tibble(data.frame)`: assoc_tibble method

Examples

```
assoc_tibble(cor(iris[,1:4]), measure_type="pearson")
assoc_tibble(iris)
```

`calc_assoc` *Calculates association or conditional association measures for a dataset*

Description

Calculates association measures for every variable pair in a dataset when by is NULL. If by is a name of a categorical variable in the dataset, conditional association measures for every variable pair at different levels of the conditional variable in a dataset are calculated.

Usage

```
calc_assoc(
  d,
  by = NULL,
  types = default_assoc(),
  include.overall = TRUE,
  handle.na = TRUE,
  coerce_types = NULL
)
```

Arguments

d	data
by	a character string for the name of the conditioning variable. Set to NULL by default.
types	a tibble for the measures to be calculated for different variable types. The default is default_assoc() which calculates Pearson's correlation if the variable pair is numeric, Goodman Kruskal's gamma if variable pair is ordered factor, canonical correlation for a factor pair, and canonical correlation for mixed variable pairs.
include.overall	Useful during calculation of conditional association measures. If TRUE calculates the overall measure of association for every pair of variable, in addition to association measures for pairs at levels of the conditioning variable, and includes it in the result.
handle.na	If TRUE uses pairwise complete observations to calculate measure of association.
coerce_types	a list specifying the variables that need to be coerced to different variable types

Details

Returns a `pairwise` tibble structure with `(p(p-1))/2` variable pairs, if a dataset has p variables. The `pairwise` output contains association measures for different types of pairs of variables specified by the `types` argument. The default is set to `default_assoc()` and a user can update these measures using `update_assoc()`. The function also allows the user to change the variable type of variable using `coerce_types` argument. The function returns a `cond_pairwise` data structure when by is set to a conditioning variable. The output contains an association measure for variable pairs at different levels of the conditioning variable. An additional column by is included in the output having levels of the conditioning variable. When `include.overall` is TRUE, the output also includes measures of association for variable pairs calculated without the conditioning variable. The by column has a level "overall" for these cases.

Value

A tibble with class `pairwise` when by argument is set to NULL. When a conditioning variable is specified the function returns a tibble with class `cond_pairwise`.

Examples

```
assoc_iris <- calc_assoc(iris, types=default_assoc())

# Example for updating assocaition measures
updated_assoc <- update_assoc(mixed_pair="tbl_ace")
assoc_iris <- calc_assoc(iris, types=updated_assoc)


# Example for coercing variable types
iris1 <- dplyr::as_tibble(iris)
iris1$Sepal.Length <- cut(iris1$Sepal.Length,breaks=4) # converting Sepal.Length into a factor
iris1$Sepal.Width <- cut(iris1$Sepal.Width,breaks=4) # converting Sepal.Width into a factor
iris1$Species <- as.character(iris1$Species) # converting Species into a character vector
assoc_iris1 <- calc_assoc(iris1, coerce_types = list(ordinal=c("Sepal.Length","Sepal.Width"),
                                                     factor="Species",
                                                     numeric=NULL))
# Example for calculating conditional association measures
cond_assoc_iris <- calc_assoc(iris, by = "Species",include.overall=TRUE)
cond_assoc_iris_wo <- calc_assoc(iris, by = "Species",include.overall=FALSE) # without overall
```

`calc_assoc_all` *Calculates multiple association measures*

Description

Calculates multiple association measures for every variable pair in a dataset.

Usage

```
calc_assoc_all(
  d,
 measures = c("pearson", "spearman", "kendall", "cancor", "nmi", "dcor", "mic", "ace",
```

```
    "polycor", "tau_b", "uncertainty", "gkTau", "gkGamma", "chi"),
  handle.na = T
)
```

Arguments

d	dataframe
measures	a set of all the measures such as "pearson","spearman","kendall", "cancor","nmi","dcor", "mic", "ace, "polycor", "tau_b", "uncertainty", "gkTau", "gkGamma" and "chi" available in the package. Set to all the measures by default and can be updated to a subset of these measures.
handle.na	If TRUE uses pairwise complete observations to calculate measure of association

Value

tibble of class "multi_pairwise"

Examples

```
calc_assoc_all(iris)
```

---

default_assoc *Default association measures calculated using* calc_assoc

---

Description

Gives a tibble for association measures for different types of variable pairs in a dataset. The variable pairs (ordinal,factor) and (ordinal,numeric) are considered as factor pair and mixed pair respectively. Used by calc_assoc for its types argument.

Usage

```
default_assoc()
```

Value

tibble default association measures for pairs of variables

Examples

```
default_assoc()
spearman_assoc <- default_assoc()
spearman_assoc$argList[[1]] <- list(method="spearman")
spearman_assoc
```

matrix_assoc *A generic function to create a symmetric matrix from a tibble of association measures with different classes*

Description

Creates a symmetric association matrix from a tibble of association measures with classes such as `pairwise`, `cond_pairwise` and `multi_pairwise`. For an association measure tibble of class `multi_pairwise`, a matrix is not calculated.

Usage

```
matrix_assoc(assoc, group = NULL)

## S3 method for class 'tbl'
matrix_assoc(assoc, group = NULL)

## S3 method for class 'pairwise'
matrix_assoc(assoc, group = NULL)

## S3 method for class 'cond_pairwise'
matrix_assoc(assoc, group = "overall")

## S3 method for class 'multi_pairwise'
matrix_assoc(assoc, group = NULL)
```

Arguments

`assoc`	A tibble or dataframe with calculated association measures for variable pairs.
`group`	A character string specifying the level of the conditioning variable for which a symmetric matrix needs to be create. One of "overall" (default) or a level of conditioning variable.

Value

matrix symmetric matrix of variable pairs with corresponding association measure

Methods (by class)

- `matrix_assoc(tbl)`: matrix_assoc method
- `matrix_assoc(pairwise)`: matrix_assoc method
- `matrix_assoc(cond_pairwise)`: matrix_assoc method
- `matrix_assoc(multi_pairwise)`: matrix_assoc method

Examples

```
# Symmetric matrix for assoc with class `pairwise`
a <- calc_assoc(iris)
matrix_assoc(a)

# Symmetric matrix for assoc with class `cond_pairwise`
```

```
b <- calc_assoc(iris, by="Species")
matrix_assoc(b,group="setosa")
```

`measures` *Association measure functions available in the package*

Description

A tibble of association measure functions along with the types of variable pairs these functions can be applied to. It also contains information regarding the packages used to calculate association measures and the range of the measures calculated.

Usage

```
measures
```

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 12 rows and 7 columns.

Value

tibble

`order_assoc_lollipop` *Ordering lollipops for the lollipop plot*

Description

Calculates an ordering for the lollipops in matrix layout displays

Usage

```
order_assoc_lollipop(assoc, group_var = group_var)
```

Arguments

`assoc`	A tibble with association measures for different variable pairs in the dataset. Must be of class `cond_pairwise` or `multi_pairwise`.
`group_var`	a character vector for grouping variable. One of "by" or "measure_type"

Value

character vector representing the ordering of the lollipops

Examples

```
order_assoc_lollipop(calc_assoc(iris),"by")
order_assoc_lollipop(calc_assoc_all(iris),"measure_type")
```

order_assoc_var *Ordering variables for matrix layout*

Description

Calculates an ordering for the variables for matrix layout displays.

Usage

```
order_assoc_var(assoc, group_var = group_var)
```

Arguments

assoc	A tibble with association measures for different variable pairs in the dataset. Must be of class pairwise, cond_pairwise or multi_pairwise.
group_var	a character vector for grouping variable. One of NULL, "by" or "measure_type"

Value

character vector representing the ordering of the variables

Examples

```
order_assoc_var(calc_assoc(iris))
order_assoc_var(calc_assoc(iris),"by")
order_assoc_var(calc_assoc_all(iris),"measure_type")
```

plot_assoc_linear *Pairwise plot in a linear layout*

Description

Plots the calculated measures of association among different variable pairs for a dataset in a linear layout.

Usage

```
plot_assoc_linear(
  assoc,
  pair_order = "max",
  plot_type = c("dotplot", "heatmap"),
  limits = c(-1, 1)
)
```

Arguments

assoc	A tibble with the calculated association measures for every variable pair in the dataset. Must be of class pairwise, cond_pairwise or multi_pairwise.
pair_order	a character string for ordering of the pairs of variables in linear layout. One of "max" (default) or "max-min". When set to "max", pairs are arranged in decreasing order of the absolute value of measure or measures (when multiple measures per pair are present). When set to "max-min", the ordering is only applicable to assoc with class cond_pairwise or multi_pairwise and the pairs are ordered in descending order of the range calculated among the measures.
plot_type	a character string for specifying the type of plot an analyst wants. One of "dot-plot" or "heatmap".
limits	a numeric vector specifying the limits of the scale. Default is c(-1,1)

Value

A ggplot2 object

Examples

```
plot_assoc_linear(calc_assoc(iris))
plot_assoc_linear(calc_assoc(iris,"Species"))
plot_assoc_linear(calc_assoc_all(iris))
```

---

plot_assoc_matrix *Pairwise plot in a matrix layout*

Description

Plots the calculated measures of association among different variable pairs for a dataset in a matrix layout.

Usage

```
plot_assoc_matrix(
  lassoc,
  uassoc = NULL,
  glyph = c("circle", "square"),
  var_order = "default",
  limits = c(-1, 1)
)
```

Arguments

lassoc	A tibble with the calculated association measures for the lower triangle of the matrix plot. Must be of class pairwise, cond_pairwise or multi_pairwise.
uassoc	A tibble with the calculated association measures for the upper triangle of the matrix plot. Must be of class pairwise, cond_pairwise or multi_pairwise. If *NULL* (default) the matrix plot is symmetric.
glyph	A character string for the glyph to be used for lassoc with "pairwise" class. Either "circle" (default) or "square".

var_order	A character string for the variable order. Either "default" for ordering using Dendser or a user provided variable order.
limits	a numeric vector of length specifying the limits of the scale. Default is c(-1,1)

Value

A ggplot2 object

Examples

```
plot_assoc_matrix(calc_assoc(iris))
plot_assoc_matrix(calc_assoc(iris,"Species"))
plot_assoc_matrix(calc_assoc_all(iris))
```

---

show_assoc *Association plot for a variable pair with or without a conditioning variable*

Description

Plots the interesting variable pairs of a dataset with or without a conditioning variable. For a numeric pair, mixed pair and factor pair, a scatterplot, raincloud plot and a mosaic plot is drawn respectively.

Usage

```
show_assoc(d, x, y, by = NULL)
```

Arguments

d	A dataset
x	a character string for one of the variable.
y	a character string for the second variable.
by	a character string for the grouping variable.

Examples

```
show_assoc(iris,"Sepal.Width","Species")
```

sym_assoc *A tibble structure for a symmetric association matrix*

Description

Creates a tibble with duplicated entries for each variable pair so that it can be used to create a symmetric association matrix.

Usage

```
sym_assoc(assoc)
```

Arguments

assoc	A tibble or dataframe with calculated association measures for variable pairs.

Value

tibble

Examples

```
sym_assoc(tbl_cor(iris))
```

tbl_ace *Alternating conditional expectations correlation*

Description

Calculates the maximal correlation coefficient from alternating conditional expectations algorithm for every variable pair in a dataset.

Usage

```
tbl_ace(d, handle.na = T, ...)
```

Arguments

d	A dataframe or tibble
handle.na	If TRUE uses pairwise complete observations.
...	other arguments

Details

The maximal correlation is calculated using alternating conditional expectations algorithm which find the transformations of variables such that the proportion of variance explained is maximised. The ace function from acepack package is used for the calculation.

Value

A tibble with a correlation coefficient from alternating conditional expectations algorithm for every variable pair

References

Breiman, Leo, and Jerome H. Friedman. "Estimating optimal transformations for multiple regression and correlation." Journal of the American statistical Association 80.391 (1985): 580-598.

Examples

```
tbl_ace(iris)
```

---

tbl_cancor *Canonical correlation*

Description

Calculates canonical correlation for every variable pair in a dataset.

Usage

```
tbl_cancor(d, handle.na = TRUE, ...)
```

Arguments

d	A dataframe or tibble
handle.na	If TRUE uses pairwise complete observations to calculate correlation coefficient
...	other arguments

Value

A tibble with canonical correlation for every variable pair

Examples

```
tbl_cancor(iris)
```

tbl_chi *Pearson's Contingency Coefficient*

Description

Calculates Pearson's Contingency coefficient for every nominal variable pair in a dataset.

Usage

```
tbl_chi(d, handle.na = TRUE, ...)
```

Arguments

d	A dataframe or tibble
handle.na	If TRUE uses pairwise complete observations.
...	other arguments

Details

The Pearson's contingency coefficient is calculated using ContCoef function from the DescTools package.

Value

A tibble with calculated Pearson's contingency coefficient for every nominal variable pair

Examples

```
tbl_chi(iris)
```

tbl_cor *Pearson, Spearman or Kendall correlation*

Description

Calculates one of either pearson, spearman or kendall correlation for every numeric variable pair in a dataset.

Usage

```
tbl_cor(d, method = "pearson", handle.na = TRUE, ...)
```

Arguments

d	A dataframe or tibble
method	A character string for the correlation coefficient to be calculated. Either "pearson" (default), "spearman", or "kendall"
handle.na	If TRUE uses pairwise complete observations to calculate correlation coefficient
...	other arguments

Value

A tibble with calculated association measure for every numeric variable pair

Examples

```
tbl_cor(iris)
tbl_cor(iris, method="kendall")
tbl_cor(iris, method="spearman")
```

`tbl_dcor` *Distance correlation*

Description

Calculates distance correlation for every numeric variable pair in a dataset.

Usage

```
tbl_dcor(d, handle.na = TRUE, ...)
```

Arguments

`d`	A dataframe or tibble
`handle.na`	If TRUE uses pairwise complete observations to calculate correlation coefficient
`...`	other arguments

Details

The distance correlation is calculated using `dcor2d` from `energy` package

Value

A tibble with distance correlation for every numeric variable pair

Examples

```
tbl_dcor(iris)
```

tbl_gkGamma *Goodman Kruskal's Gamma*

Description

Calculates Goodman Kruskal's Gamma coefficient for every ordinal variable pair in a dataset.

Usage

```
tbl_gkGamma(d, handle.na = TRUE, ...)
```

Arguments

d	A dataframe or tibble
handle.na	If TRUE uses pairwise complete observations.
...	other arguments

Details

The Goodman Kruskal's Gamma coefficient is calculated using GoodmanKruskalGamma function from the DescTools package.

Value

A tibble with ordinal variable pairs and Goodman Kruskal's Gamma coefficient

Examples

```
tbl_gkGamma(iris)
```

tbl_gkTau *Goodman Kruskal's Tau*

Description

Calculates Goodman Kruskal's Tau coefficient for every ordinal variable pair in a dataset.

Usage

```
tbl_gkTau(d, handle.na = TRUE, ...)
```

Arguments

d	A dataframe or tibble
handle.na	If TRUE uses pairwise complete observations.
...	other arguments

Details

The Goodman Kruskal's Tau coefficient is calculated using GoodmanKruskalTau function from the DescTools package.

Value

A tibble with Goodman Kruskal's Tau for every ordinal variable pair

Examples

```
tbl_gkTau(iris)
```

---

tbl_mine *MINE family measures*

Description

Calculates MINE family measures for every numeric variable pair in a dataset.

Usage

```
tbl_mine(d, method = "mic", handle.na = TRUE, ...)
```

Arguments

d	A dataframe or tibble
method	character string for the MINE measure to be calculated. Either "mic" (default), "mas", "mev", "mcn", or "mic-r2"
handle.na	If TRUE uses pairwise complete observations to calculate correlation coefficient
...	other arguments

Details

The measures are calculated using mine from minerva

Value

A tibble

References

Reshef, David N., et al. "Detecting novel associations in large data sets." science 334.6062 (2011): 1518-1524

Examples

```
tbl_mine(iris)
tbl_mine(iris, method="mas")
```

tbl_nmi *Normalized mutual information*

Description

Calculates normalized mutual information for every variable pair in a dataset.

Usage

```
tbl_nmi(d, handle.na = T, ...)
```

Arguments

d	A dataframe or tibble
handle.na	If TRUE uses pairwise complete observations to calculate normalized mutual information
...	other arguments

Details

The normalized mutual information is calculated using maxNMI from linkpotter package

Value

A tibble

Examples

```
tbl_nmi(iris)
```

tbl_polycor *Polychoric correlation*

Description

Calculates Polychoric correlation using from polycor for every ordinal variable pair in a dataset.

Usage

```
tbl_polycor(d, handle.na = TRUE, ...)
```

Arguments

d	A dataframe or tibble
handle.na	If TRUE uses pairwise complete observations to calculate correlation coefficient
...	other arguments

Details

The polychoric correlation is calculated using the polychor function from the polycor package

Value

A tibble with polychoric correlation for ordinal variable pairs

Examples

```
tbl_polycor(iris)
```

`tbl_scag` *Graph-theoretic scagnostics measures*

Description

Calculates scagnostic measure for every numeric variable pair in a dataset.

Usage

```
tbl_scag(d, scagnostic = "Outlying", handle.na = T, ...)
```

Arguments

`d`	A dataframe or tibble
`scagnostic`	a character string for the scagnostic to be calculated. One of "Outlying", "Stringy", "Striated", "Clumpy", "Sparse", "Skewed", "Convex", "Skinny" or "Monotonic"
`handle.na`	If TRUE uses pairwise complete observations.
`...`	other arguments

Details

The scagnostic measures are calculated using `scagnostics` function from the `scagnostics` package.

Value

A tibble with one of the nine scagnostic measures for every numeric variable pair

References

Wilkinson, Leland, Anushka Anand, and Robert Grossman. "Graph-theoretic scagnostics." Information Visualization, IEEE Symposium on. IEEE Computer Society, 2005

Examples

```
tbl_scag(iris)
```

tbl_tau *Kendall's tau A, B, C and Kendall's W*

Description

Calculates one of either Kendall's tau A, B, C or Kendall's W for every ordinal variable pair in a dataset.

Usage

```
tbl_tau(d, method = c("B", "A", "C", "W"), ...)
```

Arguments

d	A dataframe or tibble
method	A character string for the correlation coefficient to be calculated. Either "B" (default), "A", "C" or "W"
...	other arguments

Details

The association measures Kendall's tau A, B, C or Kendall's W are calculated using KendallTauA, KendallTauB, StuartTauC or KendallW respectively,from the DescTools package.

Value

A tibble with ordinal variable pairs along with one of either Kendall's tau A, B, C or Kendall's W measure

Examples

```
tbl_tau(iris)
tbl_tau(iris, method="A")
tbl_tau(iris, method="C")
tbl_tau(iris, method="W")
```

tbl_uncertainty *Uncertainty coefficient*

Description

Calculates uncertainty coefficient for every nominal variable pair in a dataset.

Usage

```
tbl_uncertainty(d, handle.na = TRUE, ...)
```

Arguments

d	A dataframe or tibble
handle.na	If TRUE uses pairwise complete observations to calculate correlation coefficient
...	other arguments

Details

The Uncertainty coefficient is calculated using UncertCoef function from the DescTools package.

Value

A tibble with every nominal variable pair and uncertainty coefficient value.

Examples

```
tbl_uncertainty(iris)
```

---

update_assoc *A user friendly function for changing association measures*

Description

Creates a tibble for different measures of association for different variable types of a dataset.

Usage

```
update_assoc(
  default = default_assoc(),
  num_pair = NULL,
  num_pair_argList = NULL,
  factor_pair = NULL,
  factor_pair_argList = NULL,
  ordered_pair = NULL,
  ordered_pair_argList = NULL,
  mixed_pair = NULL,
  mixed_pair_argList = NULL,
  ...
)
```

Arguments

default	default measure functions for different variable pairs. set to default_assoc()
num_pair	a measure(s) function for numeric pair of variables, default is NULL
num_pair_argList	a character string specifying the measure to be calculated using num_pair, default is NULL
factor_pair	a measure(s) function for factor pair of variables, default is NULL
factor_pair_argList	a character string specifying the measure to be calculated using factor_pair, default is NULL

`ordered_pair` a measure(s) function for ordered pair of variables, default is NULL

`ordered_pair_argList`
a character string specifying the measure to be calculated using ordered_pair, default is NULL

`mixed_pair` a measure(s) function for mixed pair of variables, default is NULL

`mixed_pair_argList`
a character string specifying the measure to be calculated using mixed_pair, default is NULL

`...` other arguments

Value

tibble

Examples

```
updated_assoc <- update_assoc(num_pair="tbl_cor", num_pair_argList="spearman",
ordered_pair="tbl_tau",mixed_pair="tbl_nmi",factor_pair="tbl_cancor")
calc_assoc(iris,types=updated_assoc)
```