corVis: An R Package for Visualising Associations and Conditional Associations

by Amit Chinwan and Catherine Hurley

Abstract Correlation matrix displays are important tools to explore multivariate datasets prior to modeling. These displays with other measures of association can summarize interesting patterns to an analyst and assist them in framing right questions while performing exploratory data analysis. In this paper, we present new visualisation techniques to visualise association between all the variable pairs in a dataset in a single plot, which is something existing displays lack. We extend these displays to regression and classification settings, where these could be used to find out variables with high predictive power. Also, we propse new methods to visualise trivariate relationship summaries using conditioning. We use different layouts like: matrix or linear, to name a few, for our displays which have their own advantages and disadvantages. We use seriation in our displays which helps in highlighting interesting patterns easily. The R package *corVis* provides an implementation.

Introduction

Exploratory Data Analysis (EDA) is an important step to explore multivariate datasets prior to modeling. One of the important tools used for EDA is correlation matrix display, popularly known as *corrgram*(Friendly, 2002). This display is produced by first calculating the association measure (correlation in this case) and then plotting these calcualted values in a matrix display. The display is useful to quickly find highly associated variables which can be explored further and can be taken into consideration during the modeling step. Figure 1 shows the *corrgram* for the *penguins* dataset from the *palmerpenguins* package where the diagonal cells have the variable names of the numeric variables in the dataset and the non-diagnonal cell corresponding to a variable pair is colored by the value of Pearson's correlation coefficient between the two variables. It seems that there is a strong positive correlation between *flipper_length_mm* and *body_mass_g*, and a strong negative correlation between *bill_depth_mm* and *flipper_length_mm* of the penguins suggesting a linear trend for these two variable pairs.

These displays are generally used with Pearson's correlation coefficient and are therefore limited to only numeric variables. In this paper, we propose an extension of the existing *corrgram* which includes a variety of association measures and where mixed type variables can also be used. We introduce new visualisations which look at multiple association measures for pairs of variables and can assist an analyst to discover interesting variable pairs. We also present displays for conditional associations at different levels of a factor variable which can help to find interesting trivariate relationships. The paper is organised in the following way: first a review of existing packages/methods to calculate and display association, then a quick overiew of some association measures used in the package, then our approach to calculate and visualize associations, followed by a summary.

Background

There have been extensions to corrgrams like: (Buja et al., 2016) and (McKenna et al., 2016), which have been proposed mainly for exploring correlations among the numeric variables for a high dimensional dataset. We introduce a display which includes all the variables of a dataset, irrespective of the data type, in a conventional corrgram plot displaying every pairwise association. This saves the effort and time of an analyst for exploring relationship among all the variable pairs. (Kuhn et al., 2013) have proposed display techniques to compare multiple association measures for every pair of output variable and a predictor to measure the importance of each predictor. This can help in summarizing a complex relationship more efficiently as compared to using just one measure like Pearson's correlation which can only find linear associations. In a similar way, we propose different visualization techniques to compare multiple association measures for all the variable pairs in a dataset which can assist a user in finding interesting patterns.

Association Measures

An association measure can be defined as a numerical summary quantifying relationship between two or more variables. For example, Pearson's correlation coefficient summarizes the strength and

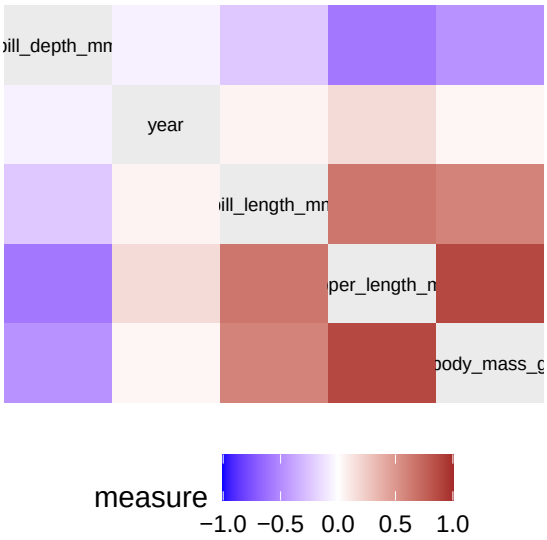

Figure 1: Example correlation matrix display for penguins data

direction of the linear relationship present between two *numeric* variables and is in the range $[-1, 1]$. Similarly, distance correlation coefficient measures the non-linear association between two *numeric* variables and summarizes it in $[0, 1]$ where 0 suggests no non-linear relationship and 1 suggests very high non-linear relationship. The package provides a collection of various measures of association which can be used to quantify the relationship between two variables and could be used to explore patterns prior to modeling. The measures available in the package are not limited to *numeric* variables only and can be used with *categorical* and *ordinal* variables as well.

- Pearson's correlation
- Spearman's rank correlation.
- Kendall's rank correlation.
- Distance correlation.
- Canonical correlations.
- Maximal-information based non-parametric exploration (MINE) statistics.

The package provides a collection of various measures of association which can be used to quantify the relationship between two variables and could be used to explore patterns prior to modeling. The measures available in the package are not limited to *numeric* variables only and can be used with *categorical* and *ordinal* variables as well. Table @ref(tab:association_measures) lists the different measures of association provided in the package with the variable types they can be used with, the package used for calculation, the information on whether the measure is symmetric, and the minimum and maximum value of the measure.

Calculating Association

We introduce a method which creates a tibble structure for the variable pairs in a dataset along with calculated association measure. The package contains various functions (shown in Table 1) for different association measures in the form `tbl_*` to calculate them. For example, a user might be interested in calculating distance correlation for numeric pair of variables in a dataset. This can be done by using `tbl_dcor`.

```
#> Rows: 10
#> Columns: 4
#> $ x            <chr> "bill_depth_mm", "flipper_length_mm", "body_mass_g", "yea~
#> $ y            <chr> "bill_length_mm", "bill_length_mm", "bill_length_mm", "bi~
#> $ measure      <dbl> 0.38720211, 0.66645577, 0.58713186, 0.07842516, 0.7039636~
#> $ measure_type <chr> "dcor", "dcor", "dcor", "dcor", "dcor", "dcor", "dcor", "~
```

Similarly, one can use `tbl_nmi` to calculate normalised mutual information for numeric, nominal and mixed pair of variables.

```
#> Rows: 28
#> Columns: 4
#> $ x            <chr> "island", "bill_length_mm", "bill_depth_mm", "flipper_len~
#> $ y            <chr> "species", "species", "species", "species", "species", "s~
#> $ measure      <dbl> 5.069605e-01, 3.525698e-01, 3.146613e-01, 3.431497e-01, 3~
#> $ measure_type <chr> "nmi", "nmi", "nmi", "nmi", "nmi", "nmi", "nmi", "nmi", "~
```

These functions return 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 output of these functions can then be used with packages like *dplyr* , *ggplot2* for further exploration of association measures.

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

In some applications, a matrix structure for a measure is more useful than dataframe or tibble. The function `matrix_assoc` helps in converting the tibble of association measure to matrix structure. The function takes a tibble or dataframe of the variable pairs of the dataset along with the calculated association measures and returns a symmetric matrix of the variables.

```
#>                   bill_length_mm bill_depth_mm flipper_length_mm body_mass_g
#> bill_length_mm                NA     0.3872021         0.6664558   0.5871319
#> bill_depth_mm         0.38720211            NA         0.7039636   0.6141631
#> flipper_length_mm     0.66645577     0.7039636                NA   0.8674122
#> body_mass_g           0.58713186     0.6141631         0.8674122          NA
#> year                  0.07842516     0.1117057         0.1643876   0.0790560
#>                           year
#> bill_length_mm    0.07842516
#> bill_depth_mm     0.11170568
#> flipper_length_mm 0.16438763
#> body_mass_g       0.07905600
#> year                      NA
```

The function outputs a matrix even if any variable pair is missing in the input tibble with *NA* for corresponding variable pair cell in the matrix output.

```
#>                   bill_length_mm bill_depth_mm flipper_length_mm body_mass_g
#> bill_length_mm                NA            NA         0.6664558   0.5871319
#> bill_depth_mm                 NA            NA         0.7039636   0.6141631
#> flipper_length_mm     0.66645577     0.7039636                NA   0.8674122
#> body_mass_g           0.58713186     0.6141631         0.8674122          NA
#> year                  0.07842516     0.1117057         0.1643876   0.0790560
#>                           year
#> bill_length_mm    0.07842516
#> bill_depth_mm     0.11170568
#> flipper_length_mm 0.16438763
#> body_mass_g       0.07905600
#> year                      NA
```

The function has an additional argument called *group* which represents the level of the grouping categorical variable for which the matrix output needs to be calculated and is set to *overall* as default.

Calculating association measures for whole dataset

`calc_assoc` can be used to calculate association measures for all the variable pairs in the dataset at once in a tibble structure. In addition to tibble structure, the output also has *paiwise* and *data.frame* class which are important class attributes for producing visual summaries in this package.

```
#> Rows: 28
#> Columns: 4
#> $ x            <chr> "island", "bill_length_mm", "bill_depth_mm", "flipper_len~
#> $ y            <chr> "species", "species", "species", "species", "species", "s~
#> $ measure      <dbl> 0.81328762, 0.84131393, 0.82447508, 0.88217284, 0.8183348~
#> $ measure_type <chr> "cancor", "cancor", "cancor", "cancor", "cancor", "cancor~

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

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

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

An analyst can update these measures using the `update_assoc` function where one can specify a `tbl_*` function to calculate association measure depending on the variable pair in the dataset and a method if it calculates more than one measure.

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

```
#> Rows: 28
#> Columns: 4
#> $ x            <chr> "island", "bill_length_mm", "bill_depth_mm", "flipper_len~
#> $ y            <chr> "species", "species", "species", "species", "species", "s~
#> $ measure      <dbl> 5.069605e-01, 8.413139e-01, 8.244751e-01, 8.821728e-01, 8~
#> $ measure_type <chr> "nmi", "cancor", "cancor", "cancor", "cancor", "nmi", "ca~
```

The tibble output for `calc_assoc` has the following structure:

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

The variable pairs in the output are unique pairs and a subset of all the variable pairs of a dataset where $x \neq y$. As explained earlier, the `measure_type` represents the association measure calculated for a specific type of variable pair. A user can be interested in calculating multiple association measures for a type of variable pair. This can be done by using the `calc_assoc` and `update_assoc` together for calculating different association measures and then merging the output tibbles.

Calculating conditional association

`calc_assoc_by` can be used to calculate association measures for all the variable pairs at different levels of a categorical variable. This can help in exploring the conditional associations and find out interesting patterns in the data prior to modeling. The output of this function is a tibble structure with *pairwise* and *data.frame* as additional class attributes. The `by` argument is used for the grouping variable which needs to be categorical.

The function also has a `types` argument which can be updated similarly to `calc_assoc`.

```
#> Rows: 84
#> Columns: 5
#> $ x            <chr> "island", "bill_length_mm", "bill_depth_mm", "flipper_len~
#> $ y            <chr> "species", "species", "species", "species", "species", "s~
#> $ measure      <dbl> 0.50167116, 0.88495341, 0.89966135, 0.91412697, 0.9106057~
#> $ measure_type <chr> "nmi", "cancor", "cancor", "cancor", "cancor", "cancor", ~
#> $ by           <fct> female, female, female, female, female, female, female, f~
```

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

The tibble output for `calc_assoc_by` has the following structure: - `x` and `y` representing a pair of variables - `measure` representing the calculated value for association measure - `measure_type` representing the association measure calculated for `x` and `y` pair. - `by` representing the levels of the categorical variable used in the function.

The variable pairs in the output are repeated for every level of `by` variable. At present the function doesn't allow multiple `by` variables to be used for conditioning but is something which can be done by using the `calc_assoc_by` function multiple times and then merging the multiple outputs.For calculating multiple measures for a specific variable type, one can use `update_assoc` with `calc_assoc_by` and then can merge these multiple tibble outputs.

Creating Your Own Association Measure

We introduce a new structure for calculating association measures which can be used to add other existing or new measures in the package. These measures can then then be analysed and visualised using the plot functions present in the package. For example, Cramer's V is a measure to summarize association between two categorical variables using the Chi-square test statistic. If a user wants to add Cramer's V to the package, they can write a simple function and then can use it for their analysis.

Visualising Association

We propose novel visualisations to display association for every variable pair in a dataset in a single plot and show multiple bivariate measures of association simultaneously to find out interesting patterns. Efficient seriation techniques have been included to order and highlight interesting relationships. These ordered association and conditional association displays can help find interesting patterns in the dataset. While designing these displays we considered matrix-type, linear and network-based layouts. A matrix-type layout simplifies lookup, and different measures may be displayed on the upper and lower diagonal. Linear layouts are more space-efficient than matrix plots, but lookup is more challenging. Variable pairs can be ordered by relevance (usually difference in measures of association or across the factor levels), and less relevant pairs can be omitted. Linear displays are also suitable to display associations between the response and predictors only. Our selection criteria for a better display were based on :

- Number of variables
- Easier pixel-variable or variable-pixel look up
- Number of levels of a factor for conditional association displays

Figure 2 shows this display for every variable pair in the *penguins* dataset from the *palmerpenguins* package. It shows a high positive Pearson's correlation among flipper_length_mm and body_mass_g, flipper_length_mm and bill_length_mm, and bill_length_mm and bodymass_g. There seems to be a strong negative Pearson's correlation between flipper_length_mm and bill_depth_mm, and bill_depth_mm and body_mass_g. The plot also shows that there is a high canonical correlation between species and other variables except year and sex, and a high canonical correlation between island and species, which traditional correlation matrix display would omit as they are limited to numeric variable pairs only. The variables in the display are ordered using average linkage clustering method to find out highly associated variables quickly.

We can also calculate multiple association measures for all the variable pairs in the dataset and compare them. This will help in finding out pairs of variables with a high difference among different measures and one can investigate these bivariate relationships in more detail. The `pairwise_summary_plot` function can be used to compare various measures using the matrix layout. It plots multiple measures among the variable pairs as bars, where each bar represents one measure of association. Figure 3 shows a matrix layout comparing Pearson's and Spearman's correlation coefficient for the numeric variable pairs in *penguins* data.

In addition to matrix layout, we can also use linear layouts for comparing multiple measures. Figure 4 shows a linear layout comparing multiple association measures for all the variable pairs in the penguins data. Linear layouts seems to be more suitable when comparing high number of association measures.

Visualising Conditional Association

The package includes a function `calc_assoc_by` which calculates the pairwise association at different levels of a categorical conditioning variable. This helps in finding out interesting variable triples

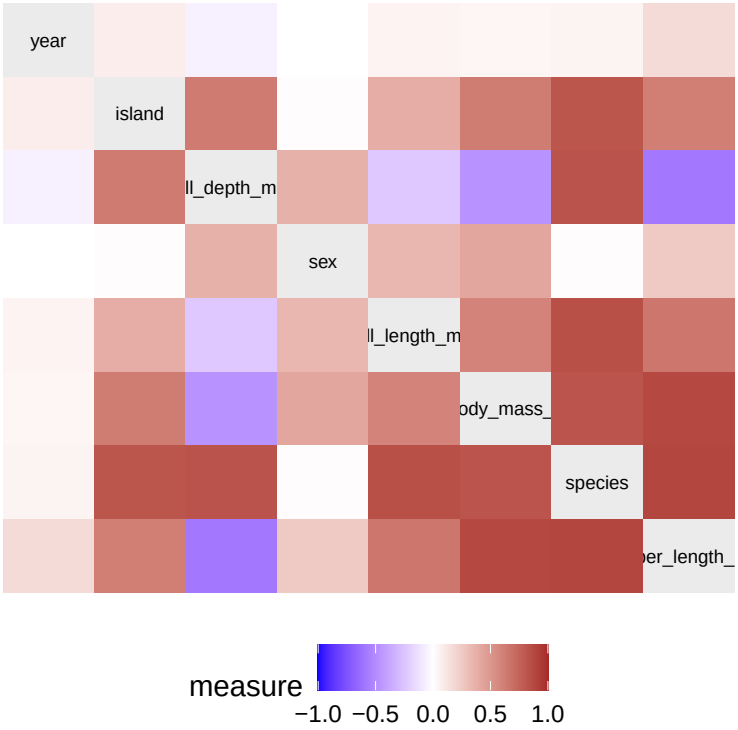

Figure 2: Association matrix display for penguins data

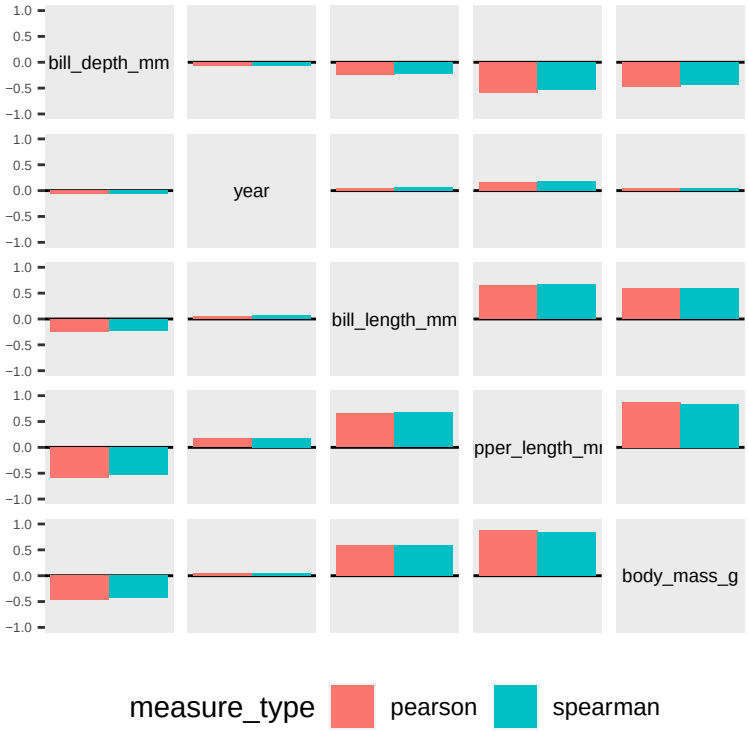

Figure 3: Comparing Pearson's and Spearman's correlation coefficient

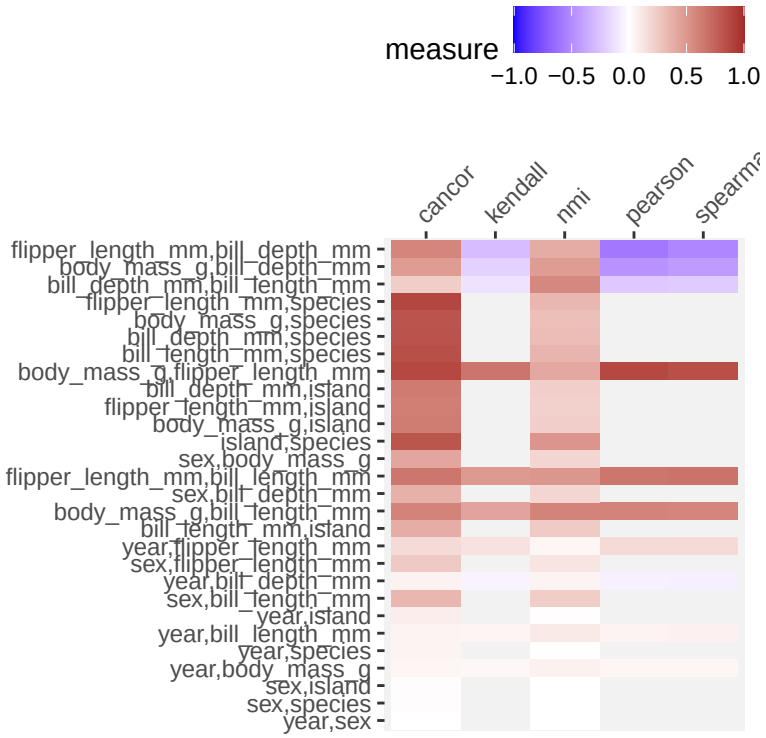

Figure 4: Comparing multiple association measures using a linear layout

which can be explored further prior to modeling. Figure 5 shows a conditional association plot for the *penguins* data. Each cell corresponding to a variable pair shows three bars which correspond to the association measure (Pearson's correlation for numeric pair and Normalized mutual information for other combination of variables) calculated at the levels of conditioning variable *island*. The dashed line represents the overall association measure. The plot shows that there is a high value for normalised mutual information between bill_length_mm and species for the penguins which lived in *Biscoe* island compared to the penguins which lived in *Dream* island. It can also be seen that the cell corresponding to variable pair flipper_length_mm and bill_depth_mm has a high negative overall Pearson's correlation and for the penguins which lived in *Biscoe* island but positive correlation for penguins which lived in *Dream* and *Torgersen* island. This is an instance of Simpson's paradox which can be taken into account during the modeling step.

We also provide a functionality for highlighting interesting patterns like Simpson's paradox. Figure 5 shows the matrix plot with highlighted cells for the variable pairs where Simpson's paradox is present.

The cells can also be highlighted on the basis of a score calculated by the user. This can be done by providing a dataframe with pairs of variables to highlight and a score for highlighting variable pairs. The cells with high score will have a thicker border compared to cells with low score. Figure 6 shows highlighted cells on the basis of a score provided for a subset of variable pairs.

We can also use linear layouts for displaying conditional association. Figure 7 shows a funnel-like linear display for conditional association measures with all the variable pairs on the y-axis, the value of association measure on x-axis and color of the points representing the level of the grouping variable. The linear layout becomes more useful over the matrix layout when the number of variables and number of levels of grouping variable are high.

Bibliography

A. Buja, A. M. Krieger, and E. I. George. A visualization tool for mining large correlation tables: The association navigator., 2016. [p1]

M. Friendly. Corrgrams: Exploratory displays for correlation matrices. *The American Statistician*, 56(4): 316–324, 2002. [p1]

M. Kuhn, K. Johnson, et al. *Applied predictive modeling*, volume 26. Springer, 2013. [p1]

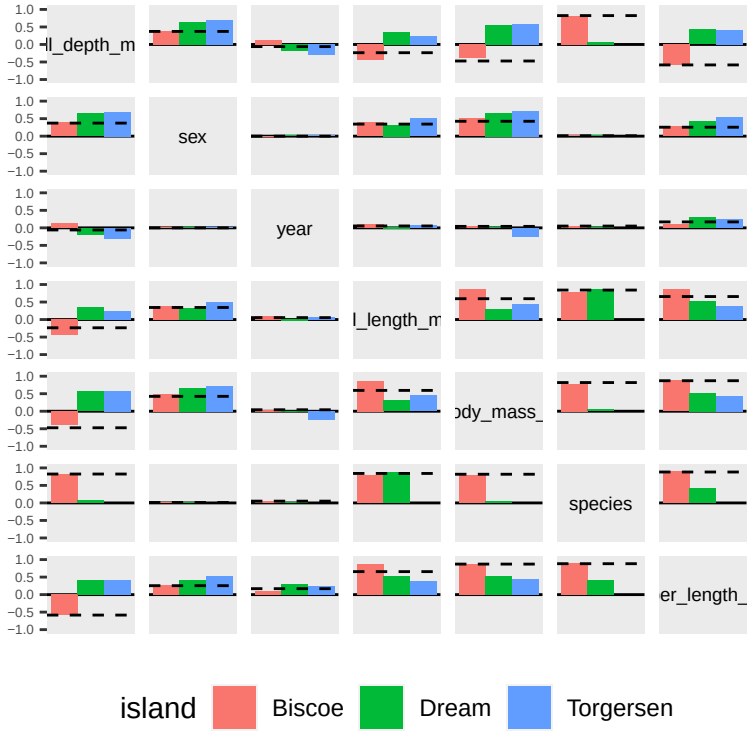

Figure 5: Conditional Association plot

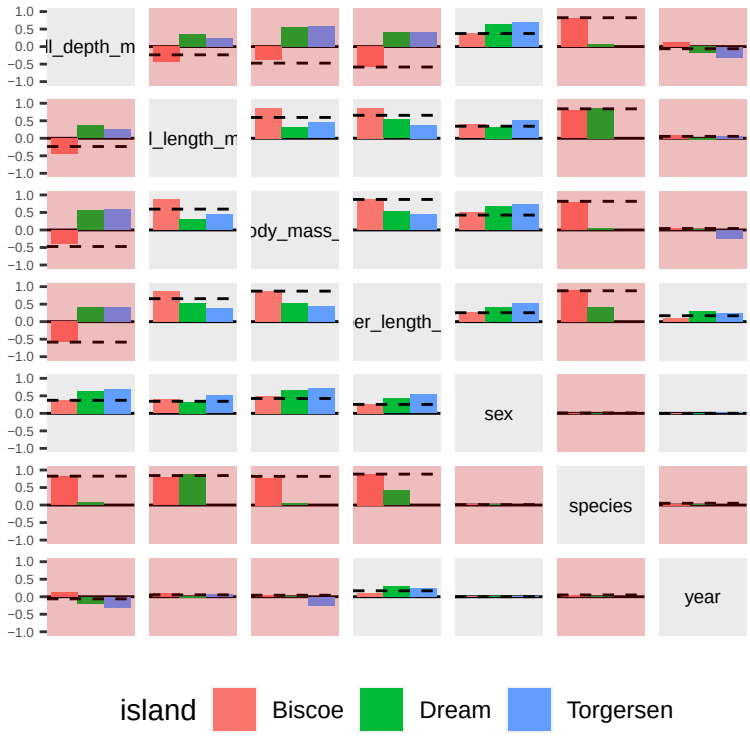

Figure 6: Conditional Association plot with Simpson's paradox

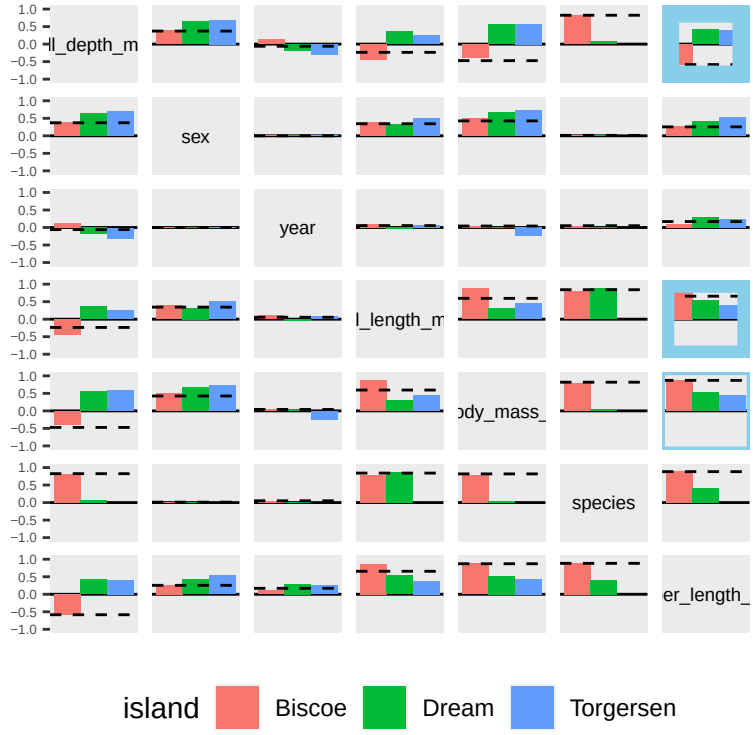

Figure 7: Conditional Association plot with manual highlighting

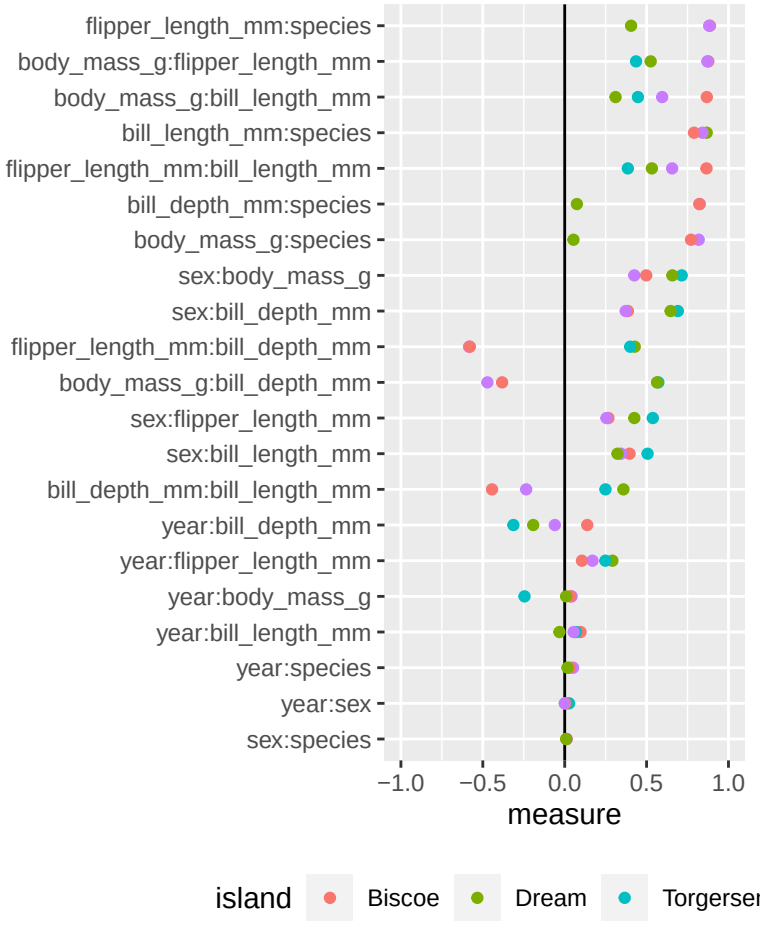

Figure 8: Conditional Association plot using linear layout

S. McKenna, M. Meyer, C. Gregg, and S. Gerber. s-corrplot: An interactive scatterplot for exploring correlation. *Journal of Computational and Graphical Statistics*, 25(2):445–463, 2016. doi: 10.1080/10618600.2015.1021926. URL https://doi.org/10.1080/10618600.2015.1021926. [p1]

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

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