PARTIAL ASSOCIATION EXPLORER: COMPARING MARGINAL AND CONDITIONAL DEPENDENCE STRUCTURES IN MIXED SOCIAL-SCIENCE DATA

Thaddée D'haegeleer, Cédric Heuchenne, Antoine Soetewey

LIDAM Discussion Paper ISBA
2026 / 32

UCLouvain

ISBA | LIDAM
Louvain Institute of Data Analysis and Modeling in economics and statistics

ISBA

Voie du Roman Pays 20 - L1.04.01

B-1348 Louvain-la-Neuve

Email : lidam-library@uclouvain.be

https://uclouvain.be/en/research-institutes/lidam/isba/publication

Partial Association Explorer: Comparing Marginal and Conditional Dependence Structures in Mixed Social-Science Data

Thaddée D'haegeleer[a], Cédric Heuchenne[a,b], Antoine Soetewey[a,b]

[a] CAPE, UCLouvain Saint-Louis Bruxelles
[b] HEC Liège, Université de Liège

Corresponding author: thaddee.dhaegeleer@uclouvain.be

Abstract

Partial Association Explorer is an open-source R (R Core Team, 2025) shiny (Chang et al., 2025) application for exploratory association analysis in mixed-type social-science datasets. Many applied researchers work with data combining survey items, socio-demographic variables, behavioral indicators, and plausible confounders. Standard correlation matrices are limited to numerical variables, while marginal association summaries can be misleading when variables are jointly related to background factors such as age, education, or region. Partial Association Explorer addresses this problem by computing pairwise association measures, significance tests, and local pair diagnostics for numerical–numerical, numerical–categorical, and categorical–categorical variable pairs. Its central feature is the comparison of unconditional and conditional association networks: users can add control variables and immediately identify associations that persist after adjustment, disappear as likely confounded links, or emerge only once a masking variable is accounted for. The application combines filtered interactive networks, pair plots adapted to each variable type, exportable summaries, and likelihood-based diagnostics for categorical pairs. We describe the statistical framework implemented in the software and illustrate the workflow using Belgian data from the European Social Survey.

Keywords: R, shiny, exploratory data analysis, mixed-type data, conditional association, association networks, survey data, confounding.

1 Introduction

Exploratory social-science analysis often begins before a formal model has been specified. Researchers may have a dataset containing survey items, socio-demographic indicators, behavioral measures, institutional variables, and several plausible confounders. A first question is then simple to formulate but difficult to answer in practice: which variables appear to be associated, and which of these associations remain after adjusting for relevant background factors?

This question is complicated by two features of many social-science datasets. First, variables are often mixed-type. A single dataset may contain numerical variables such as age or income, ordinal survey items, binary indicators, and multi-category factors. Classical correlation matrices provide a compact view only for numerical variables, while contingency-table summaries do not extend naturally to all mixed pair types. Second, marginal associations can be misleading. A relationship visible in the raw data may disappear after controlling for age, education, region, or another background factor. Conversely, a relationship that is weak or invisible marginally may become clearer once a masking variable is held fixed.

Partial Association Explorer addresses this problem as a computational workflow. The app computes association measures and significance tests for every selected pair of variables, displays

retained associations as an interactive network, and provides local pair plots that explain each edge. Its distinctive feature is the direct comparison between unconditional and conditional association structures. Users can first inspect the marginal network, then add one or several controls and immediately see which edges remain, disappear, or newly appear after adjustment.

This comparison matters for applied work. Without it, exploratory analysis can easily turn descriptive associations into substantive narratives too quickly. For example, a visible link between two survey responses may reflect the age composition of the sample rather than a direct substantive relation between the responses themselves. In other cases, the raw table or scatter plot may hide a relationship because two age, education, or region gradients offset each other. Partial Association Explorer makes this adjustment step visible and reproducible: the user can compare two network views and then inspect the pair plots behind the changing edges.

The application builds on AssociationExplorer (Soetewey et al., 2026), which showed how an accessible shiny interface can lower the barrier to marginal association exploration in mixed-type data. Partial Association Explorer extends that workflow by adding conditional or partial association analysis through a common control-variable interface, formal significance tests for every selected pair, likelihood-based categorical–categorical association measures, enhanced local pair plots, and exportable association summaries.

From a statistical perspective, the three pair types are handled by matched global measures and tests: Pearson's r and partial r for numerical–numerical pairs, η^2 and partial η^2 for numerical–categorical pairs, and likelihood-ratio-based coefficients V_L and $V_{L|Z}$ for categorical–categorical pairs. These measures are not presented as interchangeable quantities. Rather, they provide pair-type specific summaries that support a common exploratory workflow. The statistical framework is described in Section 3, and additional computational details are collected in the Appendix.

All screenshots and illustrative examples in the article use the Belgian data from the 2011 European Social Survey (ESS). The ESS is a cross-national survey covering social attitudes, behaviors, and socio-demographic characteristics. The Belgian extract is useful here because it combines numerical variables, categorical survey items, and plausible background controls such as age.

The remainder of the article is organized as follows. Section 2 describes the software workflow and outputs. Section 3 presents the implemented statistical framework for all pair types. Section 4 details the most important implementation and interface choices in the app. Section 5 gives an illustrative workflow showing how the software can both remove a confounded association and reveal a hidden one. Section 6 situates the app relative to related software, and Section 7 concludes with strengths, limitations, and possible extensions. Additional mathematical details are collected in the Appendix.

2 Software workflow and outputs

Partial Association Explorer is a standalone shiny application distributed through a public GitHub repository. The interface is organized around a complete exploratory sequence rather than a single model-fitting command. Users upload a `.csv`, `.xls`, or `.xlsx` file and may optionally upload a two-column variable-description file. The software then:

1. removes variables whose non-missing values take only a single distinct value;
2. detects pair types from the imported R classes, treating variables for which `is.numeric()` is true as numerical and the others as categorical;
3. lets the user choose analysis variables and optional control variables;
4. computes all pairwise associations, associated tests, and local diagnostics;

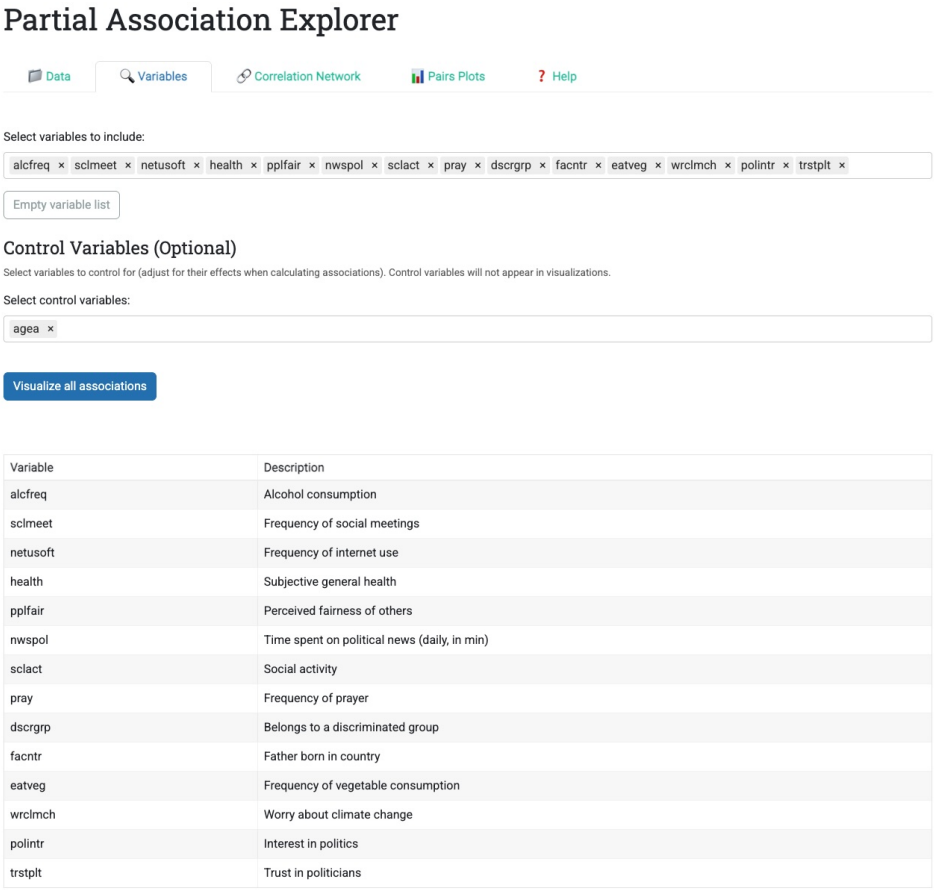
5. displays the retained associations in a filtered network;

6. allows the user to inspect each retained edge through pair plots.

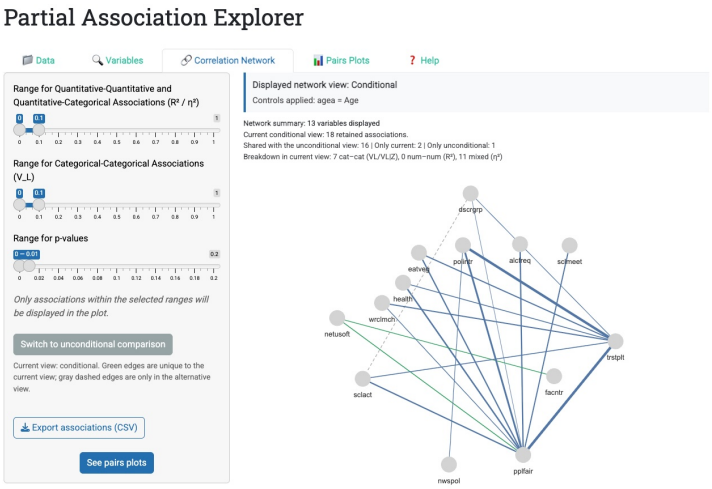
Selected analysis variables appear as nodes in the association network. Control variables are used for adjustment but do not appear as nodes. Thresholds are applied after all pairwise quantities have been computed: the user can filter numerical–numerical and numerical–categorical edges by a range for R^2 or η^2, categorical–categorical edges by a range for V_L, and all edges by a range of p-values. Thus the app is not simply computing a matrix of association measures. It structures an exploratory workflow from variable selection to network-level comparison and edge-level interpretation.

When controls are selected, the app can display the conditional network together with a visual comparison to the corresponding unconditional view. The user can switch the displayed network from the unconditional structure to the conditional one and back again. In that comparison mode, the edge colors are interpreted relative to the specific view that is currently shown: green edges are present only in the displayed view, whereas muted gray dotted edges belong only to the alternative view and are shown for comparison.

The core workflow is summarized in Table 1. The software uses different statistical models depending on the pair type, but the outputs are deliberately harmonized: each pair receives one global association measure, one formal test, and one local plot adapted to the data type.



Variable and control selection.



Filtered association network.

Figure 1: Workflow screens from Partial Association Explorer using the 2011 Belgian wave of the European Social Survey. The interface moves from variable selection to a filtered network of retained associations.

Pair type	Unconditional measure and test	Conditional measure and test	Local pair plot
Numerical–numerical	Pearson's r and a t-test	Partial r and a partial-correlation t-test	Scatter plot or added-variable plot
Numerical–categorical	η^2 and an ANOVA F-test	Partial η^2 and an extra-sum-of-squares F-test	Group means or residualized group means
Categorical–categorical	V_L and a likelihood-ratio χ^2 test	$V_{L\|Z}$ and a conditional likelihood-ratio χ^2 test	Contingency table with observed counts and Pearson residuals

Table 1: Global association measures, significance tests, and local pair plots implemented in Partial Association Explorer.

Beyond the network and pair-plot views, the app also offers several quality-of-life features that are useful in practice: variable descriptions can be displayed throughout the interface, retained pairwise summaries can be exported as `.csv`, and when controls are selected the pair-plot panel can show the conditional plot together with its unconditional counterpart so that the effect of adjustment can be inspected visually.

3 Implemented statistical framework

3.1 Numerical–numerical associations

Let X and Y be numerical variables observed on n units. Without controls, the software summarizes the association by Pearson's product-moment correlation coefficient (Pearson, 1896)

$$r_{XY} = \frac{\sum_{t=1}^{n}(X_t - \bar{X})(Y_t - \bar{Y})}{\sqrt{\sum_{t=1}^{n}(X_t - \bar{X})^2}\sqrt{\sum_{t=1}^{n}(Y_t - \bar{Y})^2}}, \tag{1}$$

and filters the network on the corresponding $R^2 = r_{XY}^2$. The null hypothesis $H_0 : r_{XY} = 0$ is tested against $H_1 : r_{XY} \neq 0$ with the usual statistic

$$t = r_{XY}\sqrt{\frac{n-2}{1-r_{XY}^2}},$$

which is compared to a Student distribution with $n-2$ degrees of freedom.

With controls $Z = (Z_1, \ldots, Z_q)$, the app computes a partial correlation through the added-variable principle. Specifically, X and Y are separately regressed on Z and the resulting residuals $\tilde{X}$ and $\tilde{Y}$ are correlated:

$$r_{XY\cdot Z} = \mathrm{cor}(\tilde{X}, \tilde{Y}).$$

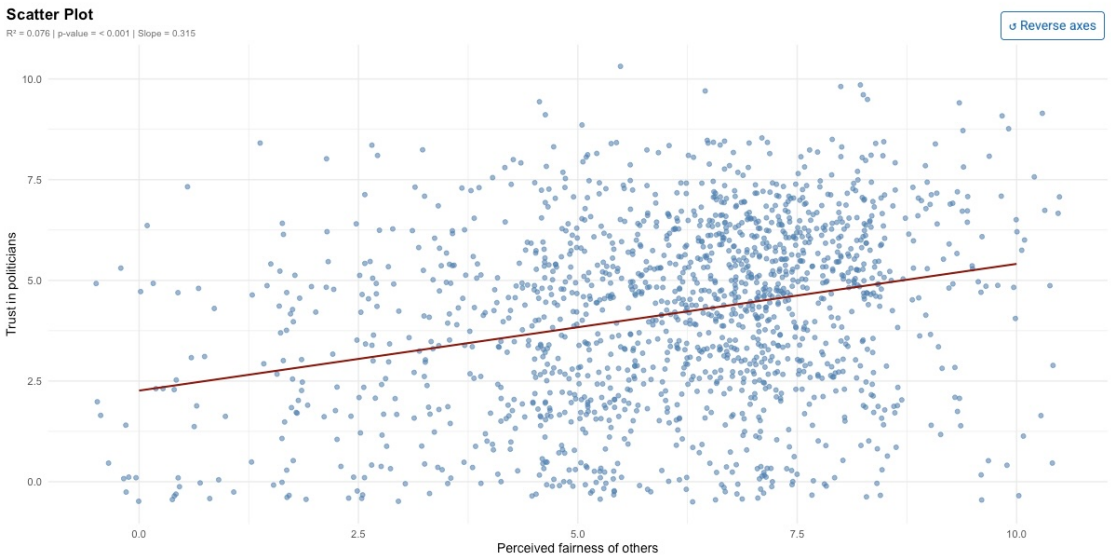
Again the network is filtered by the squared quantity $R^2_{XY\cdot Z}$. Under $H_0 : r_{XY\cdot Z} = 0$, against $H_1 : r_{XY\cdot Z} \neq 0$, the test statistic is

$$t = r_{XY\cdot Z}\sqrt{\frac{n-q-2}{1-r^2_{XY\cdot Z}}},$$

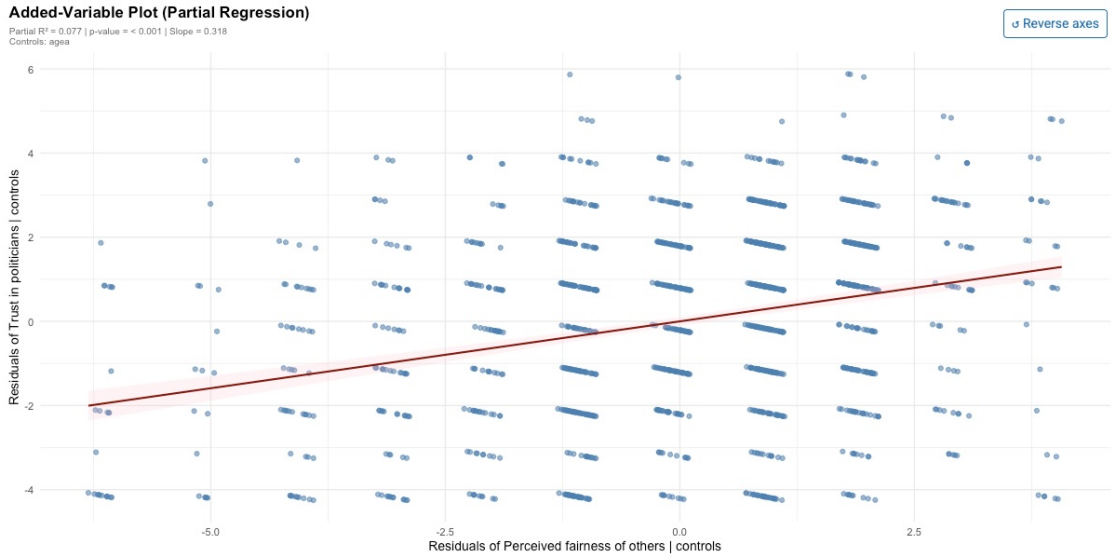
with $n-q-2$ degrees of freedom.

The associated pair plots are designed to keep the interpretation stable across cases. Without controls, the local display is a scatter plot with fitted linear trend. With controls, it becomes an added-variable plot, that is, a scatter plot of the residuals of X and Y after removing the linear contribution of Z. Additional computational details for the unconditional and conditional numerical–numerical cases are collected in Appendix A.

Reading these plots is straightforward. In the unconditional panel, the slope and the overall cloud summarize the marginal association: an upward fitted line means that larger values of one variable tend to go with larger values of the other. In Figure 2, the positive pattern suggests that respondents who give higher answers on perceived fairness also tend to report greater trust in politicians. In the conditional panel, the same reading applies to the residualized variables: the slope now reflects the remaining association after the linear effect of the control has been removed from both axes.



Unconditional pair plot.



Conditional pair plot with age of respondent (`agea`) as control.

Figure 2: Illustrative numerical–numerical pair plots from the 2011 Belgian ESS for perceived fairness of others (`pplfair`) and trust in politicians (`trstplt`). The conditional version replaces the raw scatter with the added-variable view used to compute the partial association after adjusting for age of respondent (`agea`).

3.2 Numerical–categorical associations

Let Y be numerical and let X be categorical with J groups. In the unconditional case, the software uses the classical ANOVA effect size

$$\eta^2 = \frac{\mathrm{SSB}}{\mathrm{SST}},$$

where SSB is the between-group sum of squares and SST is the total sum of squares. The null hypothesis of equal group means is tested by the usual ANOVA statistic

$$F = \frac{\mathrm{SSB}/(J-1)}{\mathrm{SSW}/(n-J)},$$

with SSW denoting the within-group sum of squares.

With controls, the app uses an ANCOVA/extra-sum-of-squares construction. A reduced model

$$Y_t = \alpha' + \sum_{k=1}^{q} \gamma'_k Z_{kt} + \varepsilon'_t$$

is compared to a full model

$$Y_t = \alpha + \sum_{j=2}^{J} \beta_j \mathbf{1}\{X_t = j\} + \sum_{k=1}^{q} \gamma_k Z_{kt} + \varepsilon_t.$$

If RSS_{red} and RSS_{full} are the residual sums of squares, the software reports

$$\eta^2_{X|Z} = \frac{\mathrm{RSS}_{red} - \mathrm{RSS}_{full}}{(\mathrm{RSS}_{red} - \mathrm{RSS}_{full}) + \mathrm{RSS}_{full}},$$

which is equivalent to the proportion of the residual variation left after adjusting for Z that is further explained by the group structure induced by X. The corresponding F-test is

$$F = \frac{(\mathrm{RSS}_{red} - \mathrm{RSS}_{full})/(J-1)}{\mathrm{RSS}_{full}/\mathrm{df}_{full}}.$$

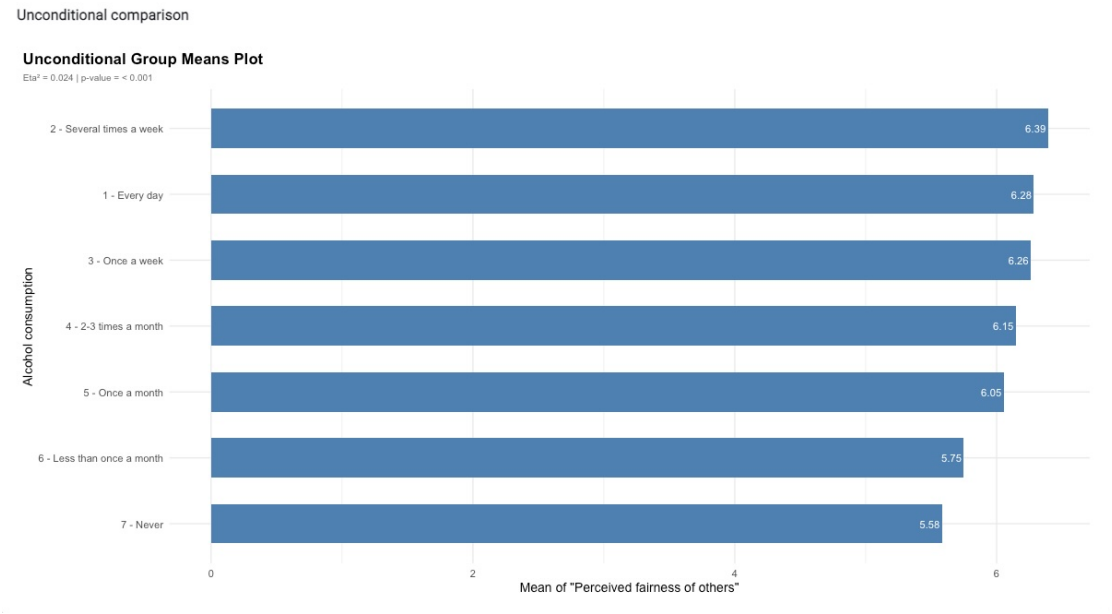
The pair plots again follow a common logic. Without controls, the local display is a group-means plot of the raw outcome. With controls, the software first fits Y on Z, extracts residuals

$$\tilde{Y}_t = Y_t - \widehat{E}(Y_t \mid Z_t),$$

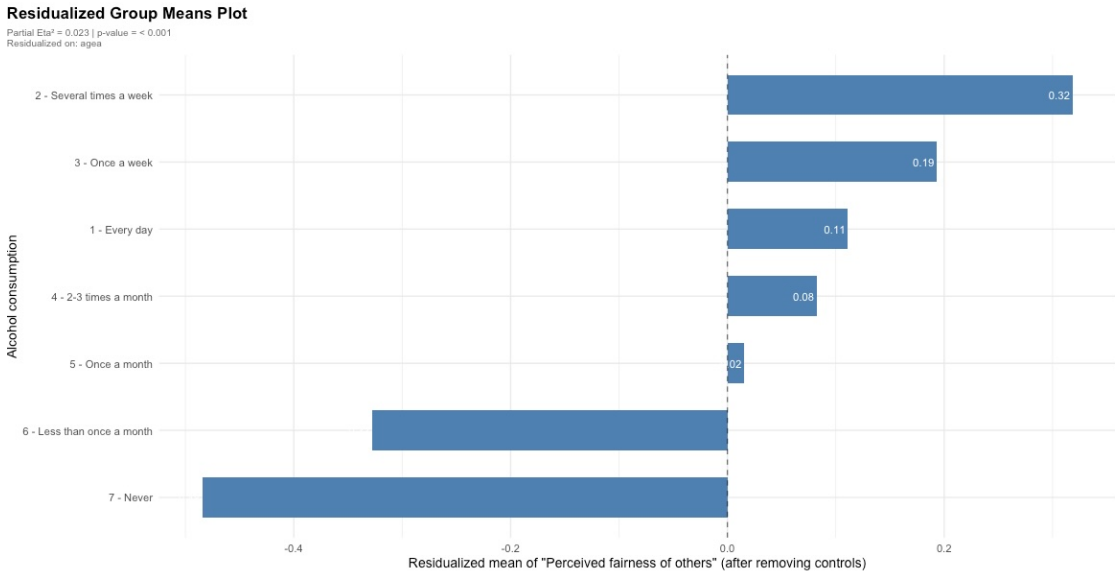
and then plots the group means of $\tilde{Y}$ across the categories of X. These residualized means show the remaining group differences after removing the linear contribution of the controls. Explicit computational formulas, including the definitions of SSB, SST, SSW, and the residualized-means construction, are collected in Appendix B.

These plots should be read vertically. Without controls, the displayed points or bars are raw group means. With controls, they are means of residualized values, so positive or negative positions indicate above- or below-expected levels of the numerical outcome after adjustment rather than positive or negative outcome values in an absolute sense. In Figure 3, differences across alcohol-frequency groups show how average responses on perceived fairness vary marginally and, in the conditional panel, how much of that variation remains once age has been taken into account.

Figure 3 also shows how the printed numbers should be read. In the unconditional panel, they are raw group means: respondents who report drinking several times a week have an average perceived fairness score of 6.39, whereas respondents who never drink have an average of 5.58. In the conditional panel, the numbers are no longer raw means but group averages of the residualized outcome after removing the linear effect of age. The zero line therefore represents the fairness level predicted by age alone. A value of 0.32 for "Several times a week" means that this group scores on average about 0.32 points above its age-predicted level, whereas the value around -0.48 for "Never" means that abstainers score about half a point below what would be expected given their age. Negative bars do not indicate negative fairness scores; they indicate below-expected fairness once age has been removed. In this example, the ordering remains broadly similar before and after adjustment, which suggests that age contributes to the pattern but does not fully explain it.



Unconditional pair plot.



Conditional pair plot with age of respondent (`agea`) as control.

Figure 3: Illustrative numerical–categorical pair plots from the 2011 Belgian ESS for perceived fairness of others (`pplfair`) by frequency of alcohol consumption (`alcfreq`). The conditional view displays residualized group means after removing the linear contribution of age of respondent (`agea`).

3.3 Categorical–categorical associations

For categorical variables X and Y with I and J levels, respectively, the software treats the pair (X, Y) as a single joint categorical outcome $W = (X, Y)$ with $K = I \times J$ categories. The modeling strategy is based on baseline-category multinomial logits relative to the first cell.

When controls are present, let $Z = (Z_1, \ldots, Z_q)$ denote the control vector after model-matrix coding of categorical controls. Under the null model of conditional independence, the

linear predictor for cell (i, j) is

$$\eta_{ij}^{(0)}(z) = \alpha_i + \beta_j + \lambda_i^\top z + \kappa_j^\top z,$$

whereas the alternative model adds an interaction term:

$$\eta_{ij}^{(1)}(z) = \alpha_i + \beta_j + \lambda_i^\top z + \kappa_j^\top z + \gamma_{ij}.$$

The unconditional case is obtained by dropping the z-dependent terms.

The terms in these predictors have a direct interpretation. The parameters α_i and β_j are category-specific baseline effects for the levels of X and Y: α_i shifts the logit for row category i of X, while β_j shifts the logit for column category j of Y. The term $\lambda_i^\top z$ is the contribution of the control vector z to row category i, and $\kappa_j^\top z$ is the analogous control contribution to column category j. Under the null model, these terms allow the controls to change the fitted margins of X and Y without introducing any residual association between them. The interaction parameter γ_{ij}, present only in the alternative model, captures the remaining cell-specific association after accounting for these main effects and control contributions. The symbol δ is used later in Appendix B for a different purpose, namely the control slopes in the residualization regression for numerical–categorical pair plots.

The implementation uses reference-category constraints. If the first level of X and the first level of Y are taken as baseline categories, then

$$\alpha_1 = \beta_1 = 0, \qquad \lambda_1 = \mathbf{0}, \qquad \kappa_1 = \mathbf{0}, \qquad \gamma_{1j} = \gamma_{i1} = 0.$$

The corresponding probabilities are

$$\pi_{ij}^{(m)}(z;\theta) = \frac{\exp(\eta_{ij}^{(m)}(z))}{\sum_{r=1}^{I}\sum_{s=1}^{J}\exp(\eta_{rs}^{(m)}(z))}, \qquad m \in \{0,1\},$$

with the baseline cell receiving linear predictor zero.

For $m \in \{0,1\}$, the observation-level log-likelihood is

$$\ell_m(\theta) = \sum_{t=1}^{n}\sum_{i=1}^{I}\sum_{j=1}^{J}\mathbf{1}\{X_t = i, Y_t = j\}\log\pi_{ij}^{(m)}(Z_t;\theta). \tag{2}$$

Let $\widehat{\theta}_0$ and $\widehat{\theta}_1$ denote the maximizers of ℓ_0 and ℓ_1, and write $\ell_0 = \ell_0(\widehat{\theta}_0)$ and $\ell_1 = \ell_1(\widehat{\theta}_1)$. The unconditional likelihood-ratio deviance is

$$G^2 = 2(\ell_1 - \ell_0),$$

which the app transforms into the bounded association measure

$$V_L = \sqrt{1 - \exp(-G^2/n)}.$$

With controls, the same construction is applied to the nested conditional multinomial models; if $G^2_{|Z}$ denotes the likelihood-ratio deviance for that conditional comparison, the app reports

$$V_{L|Z} = \sqrt{1 - \exp(-G^2_{|Z}/n)}.$$

In the unconditional contingency-table case, where the alternative model is the unrestricted joint distribution and the null model is independence, G^2/n is twice the plug-in empirical mutual information between X and Y when natural logarithms are used. Consequently,

$$V_L = \sqrt{1 - \exp[-2\widehat{I}(X;Y)]},$$

which is the plug-in sample analogue of Linfoot's informational coefficient of correlation (Linfoot, 1957). Equivalently, $V_L^2 = 1-(L_0/L_1)^{2/n}$, so the squared measure has the Cox–Snell likelihood-ratio pseudo-R^2 form (Cox & Snell, 1989), with the independence model as the null model and the association model as the alternative. The mutual-information identity is specific to the unconditional table; once controls are introduced as regression covariates, the same likelihood-ratio logic remains, but there is no equally simple contingency-table mutual-information formula.

Under regularity conditions, the associated significance test compares G^2 or $G^2_{|Z}$ to a χ^2 distribution with

$$\mathrm{df} = (I-1)(J-1),$$

that is, the number of free interaction parameters.

The fitted null model also yields expected counts

$$\widehat{E}_{ij}^{(0)} = \sum_{t=1}^{n} \Pr(X=i, Y=j \mid Z_t; \widehat{\theta}_0) = \sum_{t=1}^{n} \pi_{ij}^{(0)}(Z_t; \widehat{\theta}_0), \tag{3}$$

which reduce to the familiar independence expectations in the unconditional contingency-table case. Further computational details for the multinomial likelihood, expected counts, and local diagnostics are collected in Appendix C.

This design was chosen after considering Poisson log-linear alternatives. Log-linear models are elegant and standard for contingency tables (Agresti, 2013). However, in an interactive app with potentially many or continuous controls, multiway-table approaches become fragile because continuous covariates must be discretized and sparse tables arise quickly. The structured multinomial approach keeps the likelihood-ratio logic while allowing numerical and categorical controls to enter as regression covariates.

Local diagnostics and pair plots. Local interpretation for categorical pairs is based on observed counts

$$O_{ij} = \sum_{t=1}^{n} \mathbf{1}\{X_t = i, Y_t = j\},$$

the observed-minus-expected deviation

$$D_{ij} = O_{ij} - \widehat{E}_{ij}^{(0)},$$

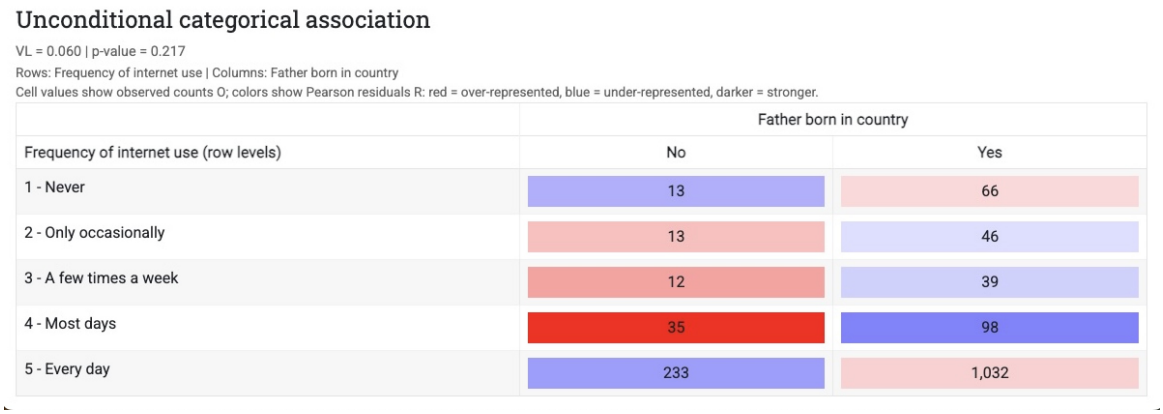
and the Pearson residual

$$R_{ij} = \frac{O_{ij} - \widehat{E}_{ij}^{(0)}}{\sqrt{\widehat{E}_{ij}^{(0)}}}.$$

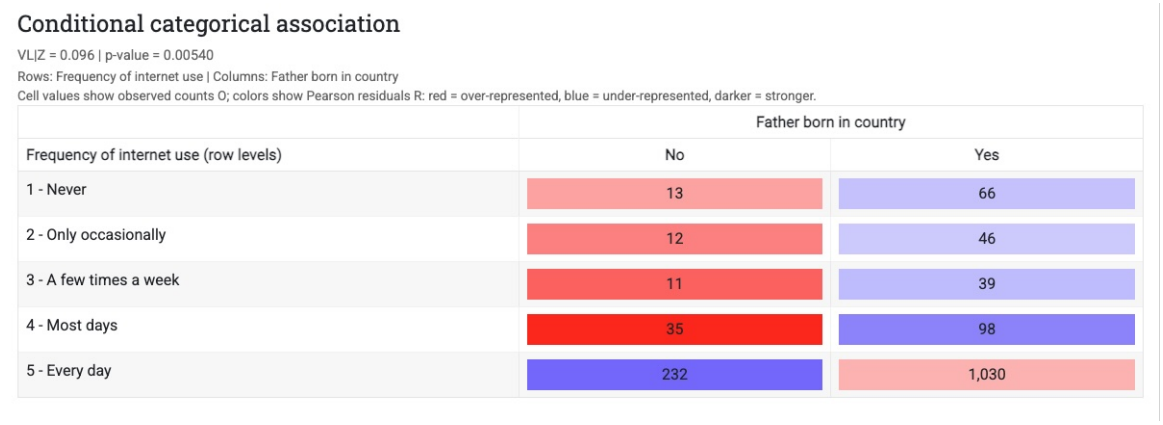
The app computes both D_{ij} and R_{ij}. In the displayed pair plot, the categorical pair plot prints the observed counts O_{ij} inside each cell and uses R_{ij} as the basis for the color scale. Positive residuals are shown in red, negative residuals in blue, and darker colors indicate larger standardized departures from the fitted null model. This choice separates the raw frequency information from the standardized model-based diagnostic.

The categorical pair plot should therefore be read in two layers. The printed count tells the reader how many observations fall in a given cell, while the color indicates whether that combination is over-represented or under-represented relative to the fitted null model. A dark red cell means that the observed count is substantially larger than the fitted expectation, whereas a dark blue cell means that the combination occurs less often than expected. In Figure 4, the printed counts do not change between the unconditional and conditional panels: there are still 13 respondents in the "Never"/"No" cell and 66 in the "Never"/"Yes" cell. What changes is the reference model used to compute the expected counts. In the unconditional panel, "Never"/"No" is blue and "Never"/"Yes" is red, meaning that marginally there are fewer non-users with a

father not born in the country and more non-users with a father born in the country than simple independence would predict. After adjustment for age, the same two cells reverse color: the "Never"/"No" combination becomes over-represented and the "Never"/"Yes" combination under-represented relative to conditional independence. A natural interpretation is that age was masking part of the association. Older respondents are both less frequent internet users and more likely to report that their father was born in the country, which pushes the unconditional table toward the "Never"/"Yes" pattern. Once comparisons are made within age groups, a different residual structure becomes visible. More generally, this figure shows why the cell values and colors should be read together: the numbers tell how many observations are in each combination, while the colors tell whether those counts are high or low relative to the fitted null model. When controls are added, the same reading applies, but the comparison is now made against conditional independence given the selected controls rather than simple marginal independence.



Unconditional pair plot.



Conditional pair plot with age of respondent (`agea`) as control.

Figure 4: Illustrative categorical–categorical pair plots from the 2011 Belgian ESS for frequency of internet use (`netusoft`) and father born in country (`facntr`). The cells print observed counts, while color encodes the Pearson residual relative to the fitted null model. The conditional panel adjusts for age of respondent (`agea`).

4 Implementation and interface choices

The app is not merely a front-end around naive repeated estimation. Several design choices in the implementation improve responsiveness and make the unconditional–conditional comparison usable in an interactive setting, especially for the categorical–categorical module.

First, the app builds a prepared problem object for each categorical pair. This object stores the cleaned factor levels, the joint outcome mapping, the control-design matrix, and grouped counts. The same prepared object is then reused by the null and alternative models. This avoids repeated preprocessing and ensures that both hypotheses are fit on exactly the same representation of the data.

Second, repeated rows of the control-design matrix are collapsed into grouped counts before evaluating the likelihood. If several observations share the same coded control vector, their contributions are accumulated once and weighted by their frequency. This does not change the model or the maximized likelihood, but it reduces the number of rows over which the categorical likelihood must be evaluated. The grouped likelihood is written explicitly in Appendix C.

Third, the categorical models are estimated with `optim()` using an analytic gradient rather than by finite-difference approximation. The alternative model is warm-started from the null-model estimate by adding zero-valued interaction parameters before optimization. This substantially reduces the amount of work needed to move from the null to the alternative model.

Fourth, the app caches pairwise results. Since users often revisit the same variables while changing filters or switching between network and pair-plot views, caching avoids refitting the same pairwise models unnecessarily.

The app also includes a specific solution for large categorical pair plots. A full contingency table is displayed as long as the number of cells does not exceed 49. Beyond that threshold, the app selects a submatrix with at most seven rows and seven columns, that is, at most a 7×7 display. The selection score is the squared Pearson residual

$$S_{ij} = R_{ij}^2 = \frac{\left(O_{ij} - \widehat{E}_{ij}^{(0)}\right)^2}{\widehat{E}_{ij}^{(0)}}, \tag{4}$$

which is the exact cellwise contribution to Pearson's chi-square statistic $X^2 = \sum_{i,j} R_{ij}^2$. It is not a cellwise contribution to the likelihood-ratio statistic G^2. However, under the null model and standard regularity conditions, X^2 and G^2 are asymptotically equivalent: both converge to the same χ^2 limit and differ only by higher-order terms. This score is therefore more suitable for display selection than cellwise likelihood-ratio terms because R_{ij}^2 is symmetric in over- and under-representation, remains informative for empty but unexpected cells, and aligns directly with the color scale shown in the pair plot. If $m_r = \min(I, 7)$ and $m_c = \min(J, 7)$ denote the display limits, the app solves

$$\max_{u,v,z} \sum_{i=1}^{I} \sum_{j=1}^{J} S_{ij} z_{ij}$$

subject to

$$\sum_{i=1}^{I} u_i = m_r, \qquad \sum_{j=1}^{J} v_j = m_c,$$

$$z_{ij} \le u_i, \qquad z_{ij} \le v_j,$$

$$z_{ij} \ge u_i + v_j - 1.$$

Here u_i and v_j indicate selected rows and columns, while z_{ij} indicates whether cell (i, j) belongs to the displayed submatrix. The exact optimization is carried out with lpSolve (Berkelaar, 2024) when available and otherwise falls back to a heuristic based on the largest positive scores S_{ij}. Because the display score and the cell value are both driven by R_{ij}, the selected submatrix emphasizes precisely the standardized local departures that the user ultimately sees. Additional details on the display score and the retained-score coverage statistic are given in Appendix D.

Finally, the interface is designed to support comparison rather than only inspection. When controls are selected, the network view can be switched between conditional and unconditional

structures, and the pair-plot section can show the unconditional comparison below the conditional plot for the same pair. This design choice is central to the app's social-science use case: it makes it easier to understand not only whether an association is retained, but also how the local structure changes after adjustment.

5 Illustrative analysis workflow

This section illustrates the app as an exploratory reasoning tool rather than as a procedure for establishing causal effects. Consider a Belgian extract of the European Social Survey and suppose a user wants to identify the dependence structure linking five variables: frequency of internet use (`netusoft`), self-rated health (`health`), frequency of alcohol consumption (`alcfreq`), frequency of social meetings (`sclmeet`), and vaccination status (`vacc19`). The aim is not to test one pre-specified hypothesis, but to obtain an interpretable overview of how these variables are related to one another within a common association network and how that network changes once a plausible background factor is introduced.

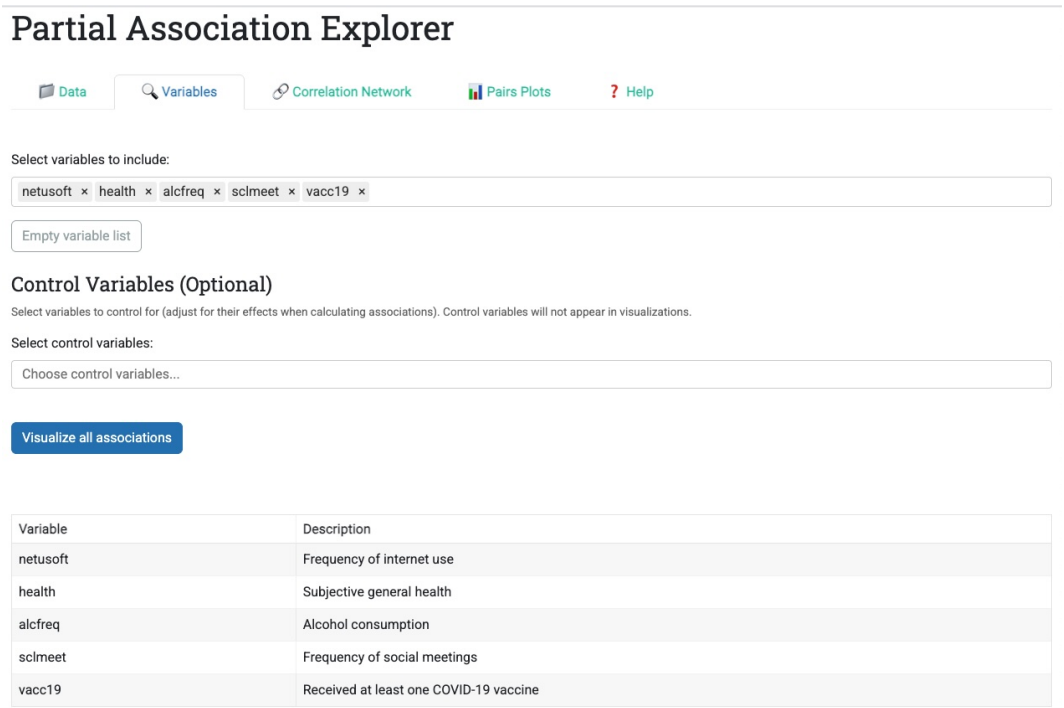
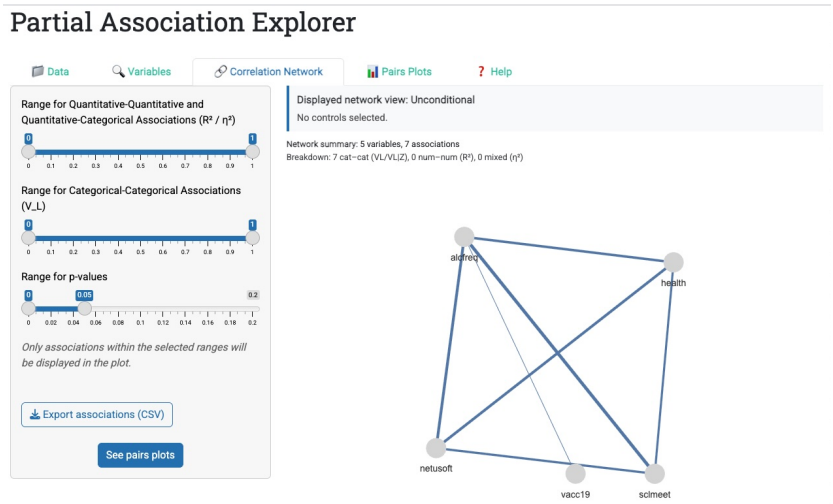
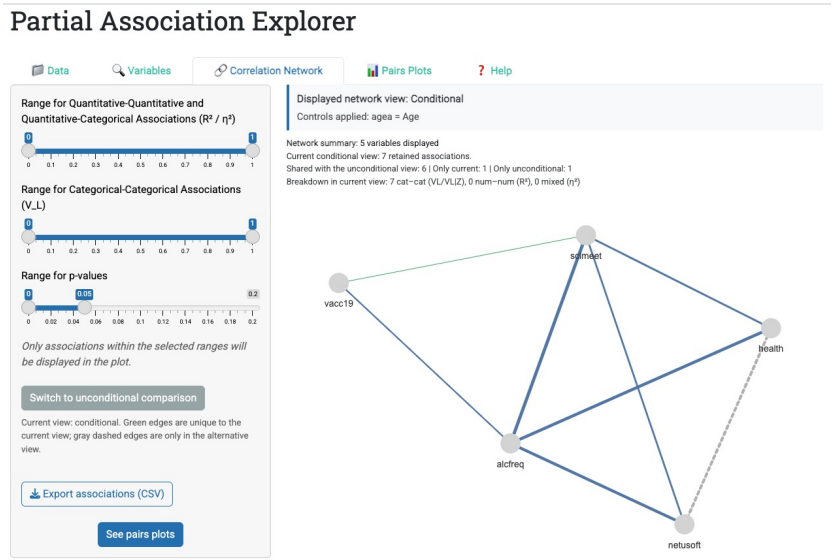


Figure 5: Variable selection screen used in the illustrative workflow. The selected variables are frequency of internet use (`netusoft`), self-rated health (`health`), frequency of alcohol consumption (`alcfreq`), frequency of social meetings (`sclmeet`), and vaccination status (`vacc19`).



Unconditional association network.



Conditional association network with age of respondent (`agea`) as control.

Figure 6: Illustrative workflow in Partial Association Explorer. Top: unconditional network. Bottom: conditional network after controlling for age of respondent (`agea`). The edge between self-rated health (`health`) and frequency of internet use (`netusoft`) is retained without controls but disappears after conditioning on age, whereas the edge between frequency of social meetings (`sclmeet`) and vaccination status (`vacc19`) appears only in the conditional view.

The user first performs the analysis without controls and filters the network by retaining only associations with p-values below 0.05. This unconditional view already yields a compact map of the dependence structure. Several links are visible, including an association between `health` and `netusoft`. At first sight this suggests that respondents who use the internet more frequently also tend to report better health. Figure 6 shows the resulting network, while the pair plot in Figure 7 clarifies the direction of the relationship. Respondents who never use the internet are under-represented in the *very good* health category and over-represented in the *fair* and *very bad* categories, whereas daily internet users show the reverse pattern. Read as a whole, the table suggests a simple marginal association: more frequent internet use is associated with better self-rated health.

This pattern is substantively plausible, but it is also a natural candidate for confounding.

The user may reasonably suspect that age plays an important role, since age is often related both to internet use and to health, as well as to other social behaviors recorded in the survey. In other words, age is a common background variable that may shape the entire dependence structure rather than a single pairwise association.

The user therefore adds age (`agea`) as a control and reruns the analysis. The conditional network in Figure 6 reveals a more surprising result than a simple global weakening. The total number of retained edges remains similar, but the composition of the graph changes. The edge between `health` and `netusoft`, clearly visible in the unconditional analysis, disappears after conditioning on age. At the same time, the edge between `sclmeet` and `vacc19`, which was not retained in the unconditional view, now enters the network.

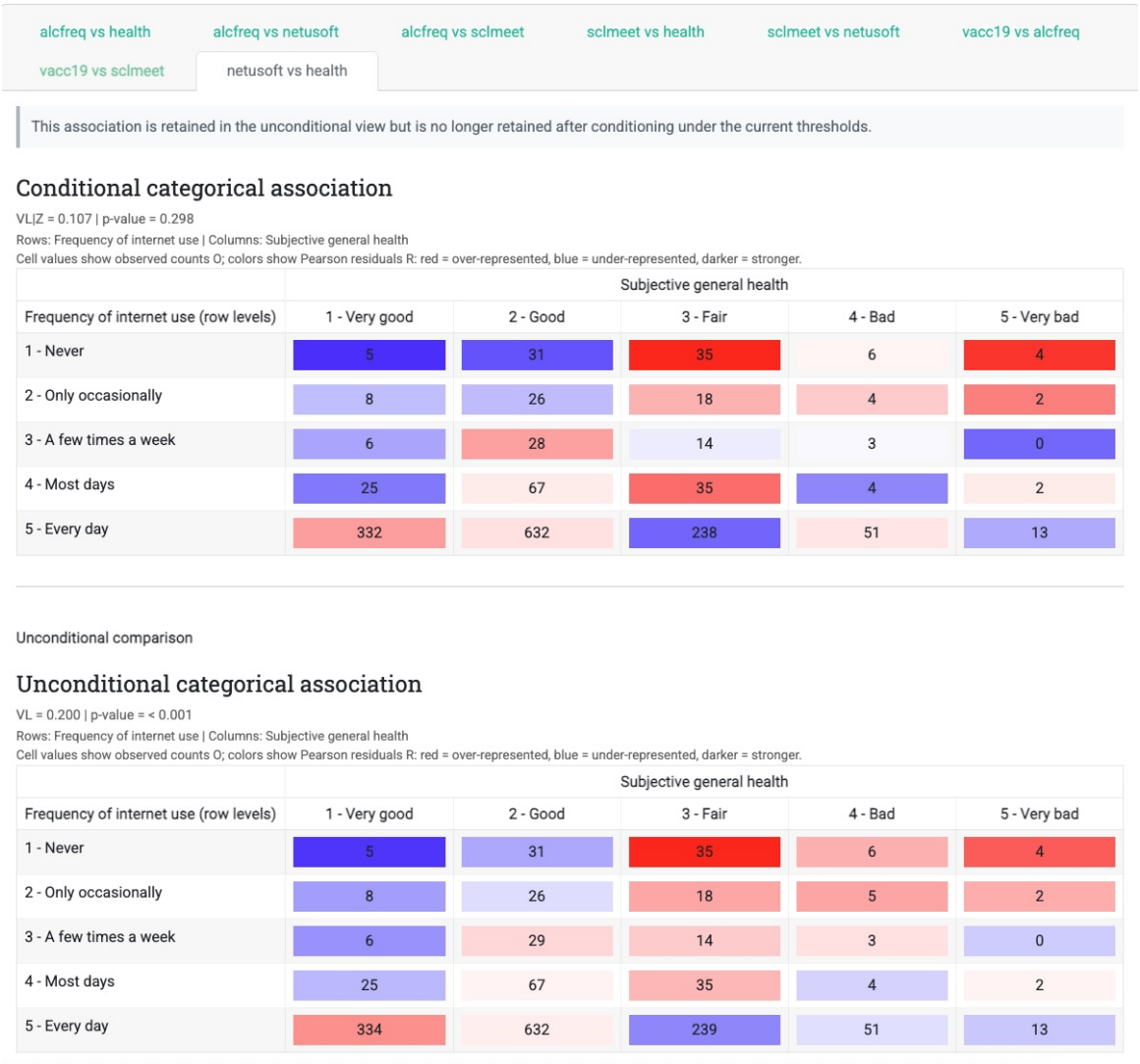


Figure 7: Conditional and unconditional pair plots for frequency of internet use (`netusoft`) and self-rated health (`health`) after adding age of respondent (`agea`) as control. The pair is retained in the unconditional view but disappears from the conditional network.

Figure 7 explains the first change more formally. Because both `health` and `netusoft` are categorical, the app reports V_L in the unconditional case and $V_{L|Z}$ in the conditional case. Without controls, the association is moderate and statistically significant ($V_L = 0.200$, $p < 0.001$). After conditioning on age, it drops to $V_{L|Z} = 0.107$ with $p = 0.298$, so the null hypothesis of conditional independence is no longer rejected at the chosen threshold. Substantively, this suggests that much of the original relationship was driven by age composition: younger respondents

tend to use the internet more often and also tend to report better health, creating an apparent association between the two variables.

At the same time, conditioning on age reveals a second and more unexpected result. The pair `sclmeet`–`vacc19` is not retained in the unconditional analysis ($V_L = 0.085$, $p = 0.074$), so a user inspecting only the marginal network would not identify it as relevant. After controlling for age, however, the association becomes statistically significant and enters the conditional graph ($V_{L|Z} = 0.105$, $p = 0.0078$). A plausible interpretation is that age masks the relationship in the raw data: younger respondents tend to report more frequent social meetings, whereas older respondents are more likely to be vaccinated. These opposite age gradients partly offset one another in the marginal table. Once comparisons are made within age groups, a clearer pattern emerges in which respondents with more frequent social contact are also more likely to be vaccinated. This second change is illustrated by the pair-plot screenshot in Figure 8.

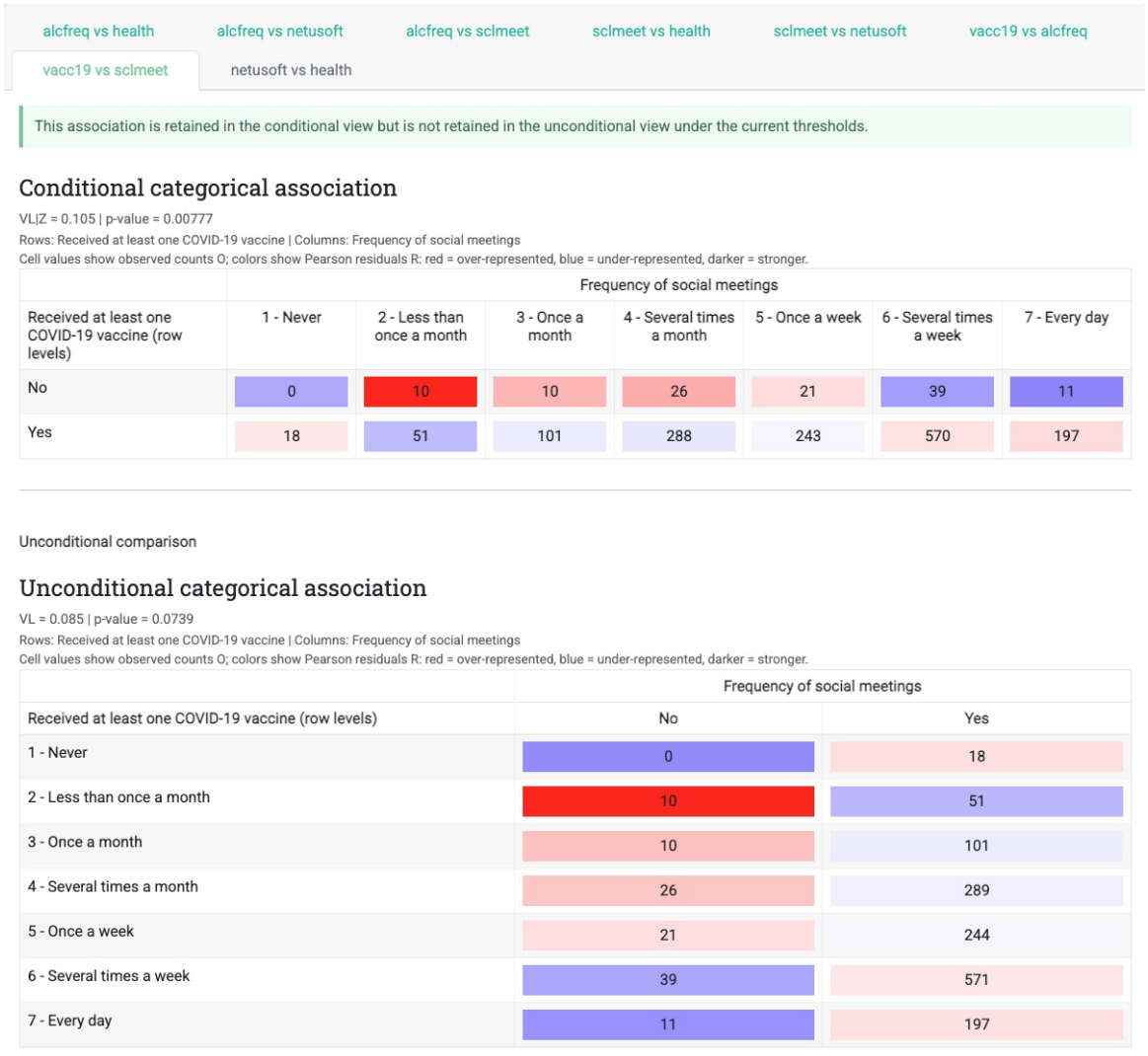


Figure 8: Conditional and unconditional pair plots for frequency of social meetings (`sclmeet`) and vaccination status (`vacc19`) after adding age of respondent (`agea`) as control. The association is not retained in the unconditional view but appears after adjustment for age.

This example captures the intended use of Partial Association Explorer. A user begins by mapping the unconditional dependence structure among a set of variables, then asks whether a plausible background variable such as age may be shaping the observed pattern. By introducing age as a control, the software turns a descriptive network into a more informative

exploratory comparison: one edge (`netusoft–health`) disappears because it was largely compositional, while another (`sclmeet–vacc19`) becomes visible only after adjustment. The substantive value of the app is precisely this side-by-side distinction between associations that are visible marginally, associations that appear robust to adjustment, and associations that are hidden until a masking variable is taken into account.

6 Related software

The closest software comparison is the earlier AssociationExplorer (Soetewey et al., 2026), from which Partial Association Explorer inherits the basic idea of exploring associations through a network and pairwise plots in a shiny interface. The main extensions are the introduction of control variables, the computation of formal tests for every selected pair, the likelihood-based treatment of categorical–categorical associations, and the expanded local diagnostics.

The relevant comparison is not only whether a program can compute an association coefficient. For the purposes of exploratory social-science analysis, five dimensions matter jointly: whether the software handles mixed variable types, whether it offers a network-level view, whether formal tests are attached to the displayed edges, whether users can introduce controls across all pair types, and whether the local structure behind each edge can be inspected. Partial Association Explorer is designed around the intersection of these dimensions.

Sirius is another relevant comparison: a Python and Django application for exploratory visualization of mixed data based on mutual information networks and backbone sparsification (Adams et al., 2021). Sirius shares with Partial Association Explorer the objective of helping users navigate heterogeneous dependence structures visually. The two tools nevertheless emphasize different tasks. Sirius uses a common information-theoretic score across mixed feature pairs and focuses on network-level exploration, whereas Partial Association Explorer adopts pair-type-specific association measures, attaches a formal test to every edge, and gives users an explicit conditional workflow in which chosen control variables can remove spurious associations or reveal masked ones. Its local diagnostics are likewise tied directly to the statistical model used for each pair type.

More generally, analysts can assemble pieces of a similar workflow from base R and general-purpose packages. Numerical correlations can be computed with standard correlation functions; numerical–categorical relationships can be assessed through linear models or ANOVA; and contingency-table analyses can be carried out with standard categorical-data tools. Assembling a workflow comparable to Partial Association Explorer usually requires moving manually between different model objects, plotting systems, and data representations. In contrast, Partial Association Explorer offers one interface in which all pair types are treated jointly, the network is filtered using a common visual grammar, and optional controls can be applied consistently across the full set of pairs.

The app should therefore be understood less as a competitor to any single-purpose package and more as a unifying exploratory layer that connects several standard statistical ideas within one workflow. Its distinctive contribution is the direct comparison of marginal and adjusted association structures in a setting where variables may be numerical, categorical, or a mixture of both.

7 Discussion

Partial Association Explorer contributes in five main ways. First, it unifies global association measures for all common pair types in a single interface. Second, it makes conditional association analysis accessible without requiring users to write regression, ANOVA, or multinomial likelihood code by hand. Third, it combines effect-size filtering with formal significance tests so that users can tune the network with strength and p-value sliders rather than relying on one

threshold alone. Fourth, it makes the unconditional–conditional comparison explicit, allowing users to detect spurious associations that disappear after adjustment and hidden associations that emerge only conditionally. Fifth, it connects global network edges to local diagnostics, which is especially important for categorical data where a single global statistic may be driven by a limited number of over- or under-represented cells.

The software is not intended to establish causal effects. Conditional associations depend on the selected controls and on the adequacy of the working models. Sparse categorical tables, rare categories, separation-like behavior in multinomial fitting, and misspecified control effects may affect the reliability of both global and local summaries. The app should therefore be understood as a transparent exploratory layer that helps users decide where more formal modeling is warranted.

A more specific limitation concerns how variable types are detected. In the app, a variable is treated as numerical when `is.numeric()` returns true in R; otherwise it is handled as categorical. This rule is transparent and easy to reproduce, but it is tied to storage class rather than to measurement meaning. Ordered factors, Likert-type items, and other ordinal variables imported as factors therefore enter the categorical branch by default. When such variables have many ordered levels, this can enlarge contingency tables, reduce stability in sparse cells, and send the analysis toward likelihood-based categorical measures even when a user might prefer a numerical approximation.

A related limitation is that the three global measures are bounded on the same $[0, 1]$ scale but are not substantively identical. R^2 and η^2 are variance-explained quantities tied to linear-model decompositions, whereas V_L is a monotone transformation of a per-observation likelihood improvement. As a result, a common slider threshold remains an exploratory convenience rather than a theoretically exact cross-type standard. The issue becomes sharper when ordinal variables could plausibly be encoded either way: the same five-category pair might in principle yield a moderate V_L when treated as categorical but a much smaller R^2 if recoded numerically, because the two representations summarize different model families and different notions of association.

These limitations suggest several extensions. One is to let users override the automatically detected type more explicitly and to add dedicated ordinal handling, for example through polychoric or polyserial alternatives, ordinal regression-based measures, or side-by-side sensitivity analyses under different codings. Another is to provide a suggestion layer based on variable labels, value patterns, or external metadata, potentially assisted by language models, to flag large ordered categorical variables that may deserve special treatment.

Computational details

The application is implemented in R and distributed through a public GitHub repository at `https://github.com/Thadhaeg/Partial-association-explorer`. The user interface relies primarily on shiny (Chang et al., 2025), bslib (Sievert et al., 2025), shinyjs (Attali, 2021), and visNetwork (Almende B.V. et al., 2025). Tabular displays rely on reactable (Lin, 2023). Data import and preprocessing use readxl (Wickham & Bryan, 2026) and janitor (Firke, 2024). The network backend uses tidygraph (Pedersen, 2024). When available, exact large-table submatrix selection is performed with lpSolve (Berkelaar, 2024).

The screenshots included in this article were generated from the app as of 2026-07-10.

References

Adams, J. L., Deluca, T. F., Danforth, C. M., Dodds, P. S., Zheng, Y., Anastasakis, K., Choi, B., Min, A., & Bessey, M. M. (2021). Sirius: Visualization of mixed features as a mutual information network graph. *arXiv*. https://doi.org/10.48550/arXiv.2106.05260

Agresti, A. (2013). *Categorical data analysis* (3rd ed.). Wiley.

Almende B.V., Contributors, & Thieurmel, B. (2025). *visNetwork: Network visualization using the vis.js library* (Version 2.1.4) [R package]. Comprehensive R Archive Network. https://doi.org/10.32614/CRAN.package.visNetwork

Attali, D. (2021). *shinyjs: Easily improve the user experience of your shiny apps in seconds* (Version 2.1.0) [R package]. Comprehensive R Archive Network. https://doi.org/10.32614/CRAN.package.shinyjs

Berkelaar, M. (2024). *lpSolve: Interface to lp_solve v. 5.5 to solve linear/integer programs* (Version 5.6.23) [R package]. Comprehensive R Archive Network. https://doi.org/10.32614/CRAN.package.lpSolve

Chang, W., Cheng, J., Allaire, J. J., Sievert, C., Schloerke, B., Xie, Y., Allen, J., McPherson, J., Dipert, A., & Borges, B. (2025). *shiny: Web application framework for R* (Version 1.11.1) [R package]. Comprehensive R Archive Network. https://doi.org/10.32614/CRAN.package.shiny

Cox, D. R., & Snell, E. J. (1989). *Analysis of binary data* (2nd ed.). Chapman and Hall.

Firke, S. (2024). *janitor: Simple tools for examining and cleaning dirty data* (Version 2.2.1) [R package]. Comprehensive R Archive Network. https://doi.org/10.32614/CRAN.package.janitor

Lin, G. (2023). *reactable: Interactive data tables for R* (Version 0.4.4) [R package]. Comprehensive R Archive Network. https://doi.org/10.32614/CRAN.package.reactable

Linfoot, E. H. (1957). An informational measure of correlation. *Information and Control*, *1*(1), 85–89. https://doi.org/10.1016/S0019-9958(57)90116-X

Pearson, K. (1896). VII. Mathematical contributions to the theory of evolution. III. Regression, heredity, and panmixia. *Philosophical Transactions of the Royal Society of London. Series A*, *187*, 253–318. https://doi.org/10.1098/rsta.1896.0007

Pedersen, T. L. (2024). *tidygraph: A tidy API for graph manipulation* (Version 1.3.1) [R package]. Comprehensive R Archive Network. https://doi.org/10.32614/CRAN.package.tidygraph

R Core Team. (2025). *R: A language and environment for statistical computing* [Computer software]. R Foundation for Statistical Computing. https://www.R-project.org/

Sievert, C., Cheng, J., & Aden-Buie, G. (2025). *bslib: Custom Bootstrap Sass themes for shiny and rmarkdown* (Version 0.9.0) [R package]. Comprehensive R Archive Network. https://doi.org/10.32614/CRAN.package.bslib

Soetewey, A., Heuchenne, C., Claes, A., & Descampe, A. (2026). AssociationExplorer: A user-friendly shiny application for exploring associations and visual patterns. *SoftwareX*, *33*, Article 102483. https://doi.org/10.1016/j.softx.2025.102483

Wickham, H., & Bryan, J. (2026). *readxl: Read Excel files* (Version 1.5.0) [R package]. Comprehensive R Archive Network. https://doi.org/10.32614/CRAN.package.readxl

A Supplementary computational details for numerical–numerical associations

This Appendix records the explicit quantities computed by the app for numerical–numerical pairs in the unconditional and conditional cases.

A.1 Unconditional case

Given numerical variables X and Y observed on n units, the app first computes the same Pearson product-moment correlation coefficient as in Equation (1) (Pearson, 1896):

$$r_{XY} = \frac{\sum_{t=1}^{n}(X_t - \bar{X})(Y_t - \bar{Y})}{\sqrt{\sum_{t=1}^{n}(X_t - \bar{X})^2}\sqrt{\sum_{t=1}^{n}(Y_t - \bar{Y})^2}}.$$

The network is filtered on the squared quantity

$$R^2_{XY} = r^2_{XY}.$$

For inference, the null hypothesis $H_0 : r_{XY} = 0$ is tested with

$$T = r_{XY}\sqrt{\frac{n-2}{1-r^2_{XY}}},$$

which is equivalent to the one-degree-of-freedom F statistic

$$F = T^2 \sim F_{1,n-2}.$$

The app reports the corresponding two-sided p-value through this equivalent F representation.

A.2 Conditional case

With controls $Z = (Z_1, \ldots, Z_q)$, the app computes the partial correlation through residualization. It first fits linear regressions

$$X_t = a_X + \sum_{k=1}^{q} b_{Xk} Z_{kt} + u^X_t, \qquad Y_t = a_Y + \sum_{k=1}^{q} b_{Yk} Z_{kt} + u^Y_t,$$

and then forms the residuals

$$\tilde{X}_t = \hat{u}^X_t = X_t - \widehat{\mathbb{E}}(X_t \mid Z_t), \qquad \tilde{Y}_t = \hat{u}^Y_t = Y_t - \widehat{\mathbb{E}}(Y_t \mid Z_t).$$

The conditional association measure is the squared partial correlation

$$r_{XY|Z} = \mathrm{cor}(\tilde{X}, \tilde{Y}), \qquad R^2_{XY|Z} = r^2_{XY|Z}.$$

To test $H_0 : r_{XY|Z} = 0$, the app uses

$$T_{\mathrm{cond}} = r_{XY|Z}\sqrt{\frac{n-q-2}{1-r^2_{XY|Z}}},$$

or equivalently

$$F_{\mathrm{cond}} = T^2_{\mathrm{cond}} \sim F_{1,n-q-2}.$$

The implementation counts only controls with non-zero variation in the effective value of q.

A.3 Pair plots

Without controls, the pair plot is the raw scatter of (X_t, Y_t) with a fitted linear trend. With controls, it is the added-variable plot formed by the residual pairs $(\tilde{X}_t, \tilde{Y}_t)$. In both cases the plot gives the local pattern behind the global R^2 measure, while the sign of the slope reflects the sign of the corresponding correlation coefficient.

B Supplementary computational details for numerical–categorical associations

Let Y be numerical and let X be categorical with levels $1, \dots, J$.

B.1 Unconditional case

The app uses the ANOVA decomposition. Writing $\bar{Y}$ for the overall mean, $\bar{Y}_j$ for the mean in group j, and n_j for the group size, it computes

$$\mathrm{SST} = \sum_{t=1}^{n} (Y_t - \bar{Y})^2, \qquad \mathrm{SSB} = \sum_{j=1}^{J} n_j (\bar{Y}_j - \bar{Y})^2.$$

The global association measure is

$$\eta^2 = \frac{\mathrm{SSB}}{\mathrm{SST}}.$$

The null hypothesis of equal group means is tested with the ANOVA statistic

$$F = \frac{\mathrm{SSB}/(J-1)}{\mathrm{SSW}/(n-J)}, \qquad \mathrm{SSW} = \sum_{j=1}^{J} \sum_{t:X_t=j} (Y_t - \bar{Y}_j)^2,$$

which is compared to an $F_{J-1,n-J}$ distribution.

B.2 Conditional case

With controls $Z = (Z_1, \dots, Z_q)$, the app uses an ANCOVA/extra-sum-of-squares construction. The full model is

$$Y_t = \alpha + \sum_{j=2}^{J} \beta_j \mathbf{1}\{X_t = j\} + \sum_{k=1}^{q} \gamma_k Z_{kt} + \varepsilon_t,$$

whereas the reduced model omits the group indicators:

$$Y_t = \alpha' + \sum_{k=1}^{q} \gamma'_k Z_{kt} + \varepsilon'_t.$$

Here α is the intercept of the full ANCOVA model, β_j is the effect of category j of X relative to the reference category, and γ_k is the slope associated with control Z_k. The primed quantities in the reduced model play the same roles after the categorical predictor has been removed. Let $\mathrm{RSS}_{\mathrm{full}}$ and $\mathrm{RSS}_{\mathrm{red}}$ denote their residual sums of squares. The extra sum of squares attributable to X given Z is

$$\mathrm{SS}(X \mid Z) = \mathrm{RSS}_{\mathrm{red}} - \mathrm{RSS}_{\mathrm{full}},$$

and the app computes

$$\eta^2_{X|Z} = \frac{\mathrm{SS}(X \mid Z)}{\mathrm{SS}(X \mid Z) + \mathrm{RSS}_{\mathrm{full}}}.$$

The corresponding test statistic is

$$F_{\text{cond}} = \frac{[\text{RSS}_{\text{red}} - \text{RSS}_{\text{full}}]/(J-1)}{\text{RSS}_{\text{full}}/\text{df}_2},$$

where df_2 is the residual degrees of freedom of the full model. This is compared to an F_{J-1,df_2} distribution.

B.3 Pair plots

Without controls, the pair plot displays the raw group means

$$\bar{Y}_j = \frac{1}{n_j} \sum_{t:X_t=j} Y_t.$$

With controls, the app first residualizes the numerical outcome by fitting

$$Y_t = a + \sum_{k=1}^{q} \delta_k Z_{kt} + u_t$$

and computing

$$\tilde{Y}_t = \hat{u}_t = Y_t - \widehat{\mathbb{E}}(Y_t \mid Z_t).$$

It then displays the residualized group means

$$\bar{\tilde{Y}}_j = \frac{1}{n_j} \sum_{t:X_t=j} \tilde{Y}_t.$$

Here a is the intercept of the residualization regression, and each δ_k is the slope attached to control Z_k. Thus δ_k measures the linear contribution of the kth control to the numerical outcome before group means are recomputed from the residuals. These bars show the remaining group differences after the linear dependence of Y on the controls has been removed.

C Supplementary computational details for categorical–categorical associations

The categorical–categorical module is based on a joint multinomial model for $W = (X, Y)$, treated as a single categorical outcome with $K = I \times J$ possible cells.

C.1 Unconditional case

Under the null model of independence, the linear predictor for cell (i, j) is

$$\eta_{ij}^{(0)} = \alpha_i + \beta_j,$$

whereas the alternative model adds interaction terms

$$\eta_{ij}^{(1)} = \alpha_i + \beta_j + \gamma_{ij}.$$

Here α_i is the baseline effect associated with level i of X, β_j is the baseline effect associated with level j of Y, and γ_{ij} is the cell-specific interaction that measures the departure from independence for combination (i, j). With first row and first column taken as references, the constraints are

$$\alpha_1 = 0, \qquad \beta_1 = 0, \qquad \gamma_{1j} = \gamma_{i1} = 0.$$

The implied probabilities are

$$\pi_{ij}^{(m)} = \frac{\exp(\eta_{ij}^{(m)})}{\sum_{r=1}^{I}\sum_{s=1}^{J}\exp(\eta_{rs}^{(m)})}, \qquad m \in \{0,1\},$$

and the observation-level log-likelihood is

$$\ell_m(\theta) = \sum_{t=1}^{n} \log \pi_{X_t Y_t}^{(m)}(\theta).$$

The likelihood-ratio statistic is

$$G^2 = 2(\ell_1 - \ell_0),$$

and the corresponding bounded association measure is

$$V_L = \sqrt{1 - \exp(-G^2/n)}.$$

Under regularity conditions, the inference step compares G^2 to a $\chi^2_{(I-1)(J-1)}$ distribution.

C.2 Conditional case

Let $Z = (Z_1, \ldots, Z_q)$ denote the controls after model-matrix coding. The null model of conditional independence is

$$\eta_{ij}^{(0)}(z) = \alpha_i + \beta_j + \sum_{k=1}^{q} \lambda_{ik} z_k + \sum_{k=1}^{q} \kappa_{jk} z_k,$$

and the alternative adds cell interactions:

$$\eta_{ij}^{(1)}(z) = \alpha_i + \beta_j + \sum_{k=1}^{q} \lambda_{ik} z_k + \sum_{k=1}^{q} \kappa_{jk} z_k + \gamma_{ij}.$$

In this conditional specification, α_i and β_j remain the baseline effects for categories of X and Y. The coefficient λ_{ik} gives the contribution of control Z_k to the logit component associated with row category i, whereas κ_{jk} gives the analogous contribution for column category j. The interaction parameter γ_{ij} again measures the residual cell-specific association once these main and control effects have been taken into account. The same reference-category structure is used, now together with

$$\lambda_{1k} = 0, \qquad \kappa_{1k} = 0 \qquad \text{for all } k.$$

The conditional probabilities are

$$\pi_{ij}^{(m)}(z) = \frac{\exp(\eta_{ij}^{(m)}(z))}{\sum_{r=1}^{I}\sum_{s=1}^{J}\exp(\eta_{rs}^{(m)}(z))}, \qquad m \in \{0,1\}.$$

To keep the main text readable, the theoretical definitions in Equations (2) and (3) are written observation by observation, indexed by t. In this Appendix, the notation changes because the app groups observations that share the same coded control vector. The statistical definition is unchanged: only the bookkeeping differs, so repeated terms are replaced by counts.

The implementation evaluates the conditional categorical likelihood on grouped control-design rows rather than observation by observation. Let $z^{(g)}$, $g = 1, \ldots, G$, denote the distinct control-design rows after model-matrix coding, and let N_{gij} be the number of observations belonging to cell (i, j) within control group g. Then the grouped conditional log-likelihood is

$$\ell_m(\theta) = \sum_{g=1}^{G}\sum_{i=1}^{I}\sum_{j=1}^{J} N_{gij} \log \pi_{ij}^{(m)}(z^{(g)};\theta), \qquad m \in \{0,1\}.$$

This is algebraically equivalent to the observation-level conditional log-likelihood, but it is much faster to evaluate whenever many observations share the same coded control vector.

The conditional likelihood-ratio statistic is

$$G^2_{|Z} = 2(\ell_1 - \ell_0),$$

and the conditional association measure is

$$V_{L|Z} = \sqrt{1 - \exp(-G^2_{|Z}/n)}.$$

The same $(I-1)(J-1)$ degrees of freedom apply to the likelihood-ratio test because the difference between the null and alternative models is exactly the interaction block.

C.3 Local diagnostics and pair plots

Let O_{ij} denote the observed cell count

$$O_{ij} = \sum_{t=1}^{n} \mathbf{1}\{X_t = i, Y_t = j\}.$$

For both the unconditional and conditional modules, the expected count used by the app is defined from the fitted null probabilities as

$$\widehat{E}^{(0)}_{ij} = \sum_{t=1}^{n} \Pr(X = i, Y = j \mid Z_t; \widehat{\theta}_0) = \sum_{t=1}^{n} \pi^{(0)}_{ij}(Z_t; \widehat{\theta}_0).$$

In the unconditional case, there are no controls and the fitted probability is constant across observations, so this reduces to

$$\widehat{E}^{(0)}_{ij} = n\pi^{(0)}_{ij}(\widehat{\theta}_0).$$

In the conditional case, the implementation groups identical control-design rows. If $z^{(g)}$, $g = 1, \ldots, G$, denotes a distinct coded control vector and $n_g = \sum_{i,j} N_{gij}$ its frequency, then the same quantity can be written as

$$\widehat{E}^{(0)}_{ij} = \sum_{g=1}^{G} n_g \pi^{(0)}_{ij}(z^{(g)}; \widehat{\theta}_0).$$

The pair plots are built from the observed-minus-expected deviation

$$D_{ij} = O_{ij} - \widehat{E}^{(0)}_{ij}$$

and the Pearson residual

$$R_{ij} = \frac{O_{ij} - \widehat{E}^{(0)}_{ij}}{\sqrt{\widehat{E}^{(0)}_{ij}}}.$$

The squared Pearson residual is the exact cellwise contribution to Pearson's chi-square statistic:

$$X^2 = \sum_{i=1}^{I}\sum_{j=1}^{J} R^2_{ij} = \sum_{i=1}^{I}\sum_{j=1}^{J} \frac{\left(O_{ij} - \widehat{E}^{(0)}_{ij}\right)^2}{\widehat{E}^{(0)}_{ij}}.$$

Thus R^2_{ij}, not R_{ij} itself, is the relevant cellwise contribution. This is not the same as a cellwise contribution to the likelihood-ratio statistic G^2, whose multinomial decomposition involves terms of the form $2\,O_{ij}\log(O_{ij}/\widehat{E}^{(0)}_{ij})$. In finite samples, X^2 and G^2 are generally different. Under the null model and standard regularity conditions, however, they are asymptotically equivalent in the sense that both converge to the same χ^2 limit and differ only by higher-order

terms from a Taylor expansion around $O_{ij} = \widehat{E}_{ij}^{(0)}$. This is why R_{ij}^2 is a principled display score even though the global test reported by the app is likelihood-ratio based.

The app prints the observed counts O_{ij} inside the displayed cells and uses R_{ij} only for the color scale. The observed-minus-expected deviations D_{ij} are still computed internally, but the standardized residuals are preferred for coloring because they provide a comparable scale across cells.

In the implementation, both null and alternative models are estimated with BFGS through `optim()` using analytic gradients. The alternative model is warm-started from the null-model estimate by inserting a block of zero-valued interaction parameters before optimization.

D Large categorical tables and coverage of the displayed submatrix

For large categorical pair plots, the app optimizes over the score matrix $S_{ij} = R_{ij}^2$ defined in Equation (4). If $\mathcal{S}$ denotes the selected submatrix, the app reports the proportion of total score retained in the display:

$$\text{Coverage} = \frac{\sum_{(i,j)\in\mathcal{S}} S_{ij}}{\sum_{i=1}^{I}\sum_{j=1}^{J} S_{ij}}.$$

This percentage tells the user how much of the standardized residual structure remains visible after the full table has been trimmed to at most a 7×7 display.

The choice of $S_{ij} = R_{ij}^2$ is deliberate. Although the pair plot prints the observed counts O_{ij} inside cells, it colors cells according to the sign and magnitude of R_{ij}. Selecting rows and columns through R_{ij}^2 therefore prioritizes the largest standardized departures regardless of sign. This is more appropriate for display than local likelihood-ratio terms such as $2\,O_{ij}\log(O_{ij}/\widehat{E}_{ij}^{(0)})$, which are asymmetric in over- and under-representation and become zero when $O_{ij} = 0$. In particular, an empty cell that is highly unexpected under the fitted null model may still deserve to be shown visually, and R_{ij}^2 preserves that information whereas the local likelihood-ratio term does not.

The app prints the observed counts O_{ij} inside the displayed cells and uses R_{ij} for the color encoding. The observed-minus-expected deviations D_{ij} are still computed internally, but the standardized residuals remain the most suitable quantity for coloring because they are comparable across cells.