AssociationExplorer: A user-friendly Shiny application for exploring associations and visual patterns

Antoine Soetewey[a,b,*], Cédric Heuchenne[a,b], Arnaud Claes[c], Antonin Descampe[c]

[a] *HEC Liège, ULiège, Rue Louvrex 14, 4000 Liège, Belgium*
[b] *The Center for Applied Public Economics (CAPE), UCLouvain Saint-Louis Bruxelles, Boulevard du Jardin Botanique 43, 1000 Brussels, Belgium*
[c] *Observatory for Research on Media and Journalism (ORM), UCLouvain, Ruelle de la Lanterne Magique 14, 1348 Louvain-la-Neuve*

Abstract

AssociationExplorer is an interactive R Shiny web application designed to help non-technical users explore statistical associations within multivariate datasets. Aimed particularly at journalists, educators, and engaged citizens, the tool facilitates the discovery and interpretation of meaningful patterns between variables without requiring programming or statistical expertise. Users can upload structured data (e.g., from surveys or open government datasets), select relevant variables, and dynamically visualize relationships via a correlation network and contextual bivariate plots. To illustrate its capabilities, we present a case study based on the European Social Survey (ESS), showcasing how users can investigate links between attitudes, behaviors, and socio-demographic indicators across countries. The app supports a range of association measures adapted to variable types (Pearson's r, Eta, and Cramer's V), ensuring both flexibility and statistical rigor. The visual interface enables users to adjust thresholds for association strength and examine results through interactive graphs and summary tables, making the app particularly well-suited for data storytelling, exploratory research, and public communication. AssociationExplorer demonstrates how open-source statistical tooling can enhance transparency, accessibility, and insight in the interpretation of complex social data.

Keywords: R Shiny, Exploratory data analysis, Correlation network

Nr.	Code metadata description	Metadata
C1	Current code version	v3.5.4
C2	Permanent link to code/repository used for this code version	`https://github.com/AntoineSoetewey/AssociationExplorer`
C3	Permanent link to Reproducible Capsule	For example: `https://codeocean.com/capsule/0270963/tree/v1` xxx
C4	Legal Code License	MIT License
C5	Code versioning system used	Git
C6	Software code languages, tools, and services used	R, R Shiny
C7	Compilation requirements, operating environments & dependencies	xxx
C8	If available link to developer documentation/manual	`https://github.com/AntoineSoetewey/AssociationExplorer/tree/main/documentation` to do write doc xxx
C9	Support for questions or issues	`https://github.com/AntoineSoetewey/AssociationExplorer/issues`

Table 1: Code metadata

Metadata

The metadata associated with the current version of the software is summarized in Table 1.

1. Motivation and significance

The growing availability of large, complex, and high-dimensional datasets in the social sciences and public policy domains offers unprecedented opportunities for insight but also presents significant challenges for exploration and interpretation, particularly for non-specialist audiences. Journalists, educators, and engaged citizens often struggle to identify and interpret meaningful

*Corresponding author.
Email addresses: antoine.soetewey@uliege.be (Antoine Soetewey), cedric.heuchenne@uclouvain.be (Cédric Heuchenne), arnaud.claes@uclouvain.be (Arnaud Claes), antonin.descampe@uclouvain.be (Antonin Descampe)

relationships between variables without the aid of programming skills or formal statistical training. This barrier limits the broader societal impact of open data initiatives, which are designed to promote transparency, accountability, and informed public discourse.

To address this gap, we developed AssociationExplorer, a free, open-source R Shiny [3] application that enables intuitive and statistically grounded exploration of multivariate associations. The tool guides users through a visual journey of variable relationships by automatically computing appropriate bivariate association measures–Pearson's r, Eta, and Cramer's V–depending on variable types, and presenting the results in an interactive correlation network. Users can set thresholds for the strength of association and explore linked bivariate plots or tables with descriptive labels. This workflow supports transparent, reproducible, and non-technical exploratory data analysis (EDA).

Our software is particularly suited to survey-based datasets and public opinion studies. As an illustrative case, we apply AssociationExplorer to the European Social Survey (ESS), a cross-national survey that collects attitudinal, behavioral, and socio-demographic data across European countries. The tool allows users to uncover associations between trust in institutions, policy preferences, media usage, and demographic characteristics without any coding. This type of interactive analysis can empower journalists to build data-driven narratives, educators to teach statistical thinking, and citizens to explore evidence underlying public debate.

While several tools and libraries exist for correlation analysis (e.g., `corrr` [7], `GGally` [13], `corrplot` [17], `ggstatsplot` [11], `correlation` [9, 10], `lares` [8] and `Hmisc` [6] in R [12], or Python packages like `seaborn` [16] and `pingouin` [15]), they typically require programming proficiency and focus primarily on numerical associations. Most of these tools do not handle nominal categorical variables directly; if included, such variables are often transformed using one-hot or dummy encoding, which can transform their original structure and limit interpretations.

In contrast, AssociationExplorer is designed to handle both quantitative and qualitative variables (including nominal factors) natively and transparently. It provides a guided, end-to-end workflow that begins with data upload and preprocessing, continues through variable selection and association filtering, and ends with interpretable visualizations. This structured process is intuitive and accessible for users of all backgrounds, making the app especially suitable for those without programming experience or formal statistical training. By lowering the technical barrier for statistical exploration, AssociationExplorer contributes to a more inclusive data culture and supports data-driven discovery in both academic and public-facing contexts.

2. Software description

2.1. Software architecture

The AssociationExplorer application is a web-based graphical user interface built with the R programming language using the Shiny framework. It adopts a modular, reactive structure where data inputs and user selections dynamically trigger updates to the visualizations and underlying computations. The user interface is styled using the `bslib` package [14] with a modern flat theme and enhanced interactivity through `shinyjs` [2] and `visNetwork` [1]. The app is structured into distinct tabs: data upload, variable selection, correlation network visualization, pairs plots, and a help section.

Upon upload, the dataset is preprocessed to exclude variables with zero variance, as these variables do not vary across observations and therefore cannot contribute to meaningful associations or visualizations. Removing them helps reduce noise and ensures that only informative variables are included in the analysis. Optionally, the user can provide a variable description file, which is integrated and used to annotate visual elements. The backend computes association measures tailored to the variable types: Pearson's r for numeric pairs, Cramer's V for categorical pairs, and the correlation ratio (eta) for mixed pairs. Associations are filtered using user-defined thresholds and represented in a correlation network and complementary bivariate plots. The app handles both CSV and Excel files and supports large datasets of up to 100 MB.

2.2. Software functionalities

The major functionalities of the AssociationExplorer application include:

- **Data upload and cleaning:** The app supports CSV and Excel files. It automatically removes variables with only one unique value, as they lack variability and cannot contribute to association analyses. Additionally, it can optionally integrate user-supplied descriptions of variables, which are used to enhance the clarity and interpretability of visualizations, particularly for non-technical users.

- **Variable selection interface:** Users can interactively choose which variables to explore. When a description file is provided, a summary table links variable names to their descriptions.

- **Dynamic association filtering:** The app computes pairwise association measures between all selected variables, using a method tailored to the types of variables involved:

- For pairs of numeric variables X and Y, the app calculates Pearson's correlation coefficient (r), and retains the association if the coefficient of determination (R^2) exceeds a user-defined threshold:

$$R^2 = r^2 = (\mathrm{cor}(X, Y))^2 \tag{1}$$

where the Pearson's correlation coefficient $\mathrm{cor}(X, Y)$ is defined as:

$$r(X, Y) = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^{n}(X_i - \bar{X})^2}\sqrt{\sum_{i=1}^{n}(Y_i - \bar{Y})^2}} \tag{2}$$

where $\bar{X}$ and $\bar{Y}$ are the sample means of X and Y, respectively, and n is the number of observations.

- For pairs of categorical variables, it computes Cramer's V, a normalized measure of association derived from the chi-squared statistic:

$$V = \sqrt{\frac{\chi^2}{n \cdot \min(k-1, r-1)}} \tag{3}$$

where χ^2 is the chi-squared statistic, n is the total number of observations, and k, r are the number of categories in each variable.

- For mixed pairs (one numeric and one categorical variable), the app computes the correlation ratio (η), which quantifies how much of the variance in the numeric variable is explained by the grouping structure of the categorical variable. It is defined as:

$$\eta = \sqrt{\frac{\mathrm{SS}_{\text{between}}}{\mathrm{SS}_{\text{total}}}} \tag{4}$$

where:

  - $\mathrm{SS}_{\text{total}}$ is the *total sum of squares* of the numeric variable:

$$\mathrm{SS}_{\text{total}} = \sum_{i=1}^{n}(y_i - \bar{y})^2$$

    with y_i the observed numeric values and $\bar{y}$ their overall mean.

  - $\mathrm{SS}_{\text{between}}$ is the *between-group sum of squares*, computed as:

$$\mathrm{SS}_{\text{between}} = \sum_{g=1}^{G} n_g(\bar{y}_g - \bar{y})^2$$

where G is the number of groups (categories), n_g is the number of observations in group g, $\bar{y}_g$ is the group mean, and $\bar{y}$ is the overall mean.

This formulation captures the proportion of the total variance in the numeric variable that can be attributed to differences between the categorical groups. A pair is retained only if η^2 exceeds the numeric threshold defined by the user.

Each association is retained only if its corresponding strength metric–R^2, η^2, or Cramer's V–exceeds the threshold set by the user. These thresholds can be adjusted interactively through the interface, and the filtering process is reactive: updates to the thresholds immediately propagate to the network and bivariate visualizations. This allows users to dynamically control the sensitivity of the association analysis and focus on relationships of substantive interest.

- **Interactive correlation network:** The filtered associations are displayed as an interactive graph where nodes represent variables and edges represent associations. Edge thickness and length reflect the strength of the association: stronger associations are shown with thicker and shorter edges, whereas weaker associations are displayed with thinner and longer edges. This dual visual representation helps users quickly identify the most meaningful relationships in the network. Variable descriptions are displayed when the user hovers over a node in the network, allowing for quick access to additional context without cluttering the visualization. The network is built using the `visNetwork` R package, which supports interactive, customizable graph layouts; full documentation is available at `https://datastorm-open.github.io/visNetwork/`.

- **Bivariate visualization of variable pairs:** For each variable pair exceeding the threshold:

  - Scatter plots with linear regression lines are shown for numeric pairs, helping visualize the direction and strength of the relationship.

  - Colored contingency tables with marginal sums are shown for categorical pairs, where cell background colors vary in intensity according to the frequency of observations, using a blue gradient to highlight higher counts.

  - Mean plots are shown for numeric-categorical pairs, with bars ordered by mean value to make it easy to compare and rank categories based on the quantitative variable.

  Confidence intervals for the regression lines and standard errors in the mean plots are intentionally omitted to maintain a clean, uncluttered visualization that prioritizes ease of interpretation. Mean plots were selected over boxplots to avoid overwhelming non-expert users with distributional information, focusing instead on clear, accessible insights about average group differences.

- **Accessibility and user guidance:** A dedicated help section explains each step, allowing users with a limited statistical background to interactively explore their data.

2.3. Sample code snippets analysis

Below is a representative snippet from the application showing how the software selects the appropriate association measure depending on the types of the variable pair and filters associations based on user thresholds:

```
# Numeric vs numeric case
if (is_num1 && is_num2) {
    ...
  r <- cor(x, y, use = "complete.obs")
  cor_val <- ifelse(r^2 >= threshold_num, r, 0)
  cor_type <- "Pearson's r"

# Categorical vs categorical case
} else if (!is_num1 && !is_num2) {
  ...
  tbl <- table(x, y)
  ...
  n_obs <- sum(tbl)
  df_min <- min(nrow(tbl) - 1, ncol(tbl) - 1)
  if (df_min > 0) {
    v_cramer <- sqrt(chi$statistic / (n_obs * df_min))
    cor_val <- ifelse(v_cramer >= threshold_cat,
                      v_cramer, 0)
    cor_type <- "Cramer's V"
  }

# Mixed case (numeric vs categorical)
```

```
} else {
  ...
  means_by_group <- tapply(num_var, cat_var,
                           mean, na.rm = TRUE)
  overall_mean <- mean(num_var, na.rm = TRUE)
  n_groups <- tapply(num_var, cat_var, length)
  bss <- sum(n_groups * (means_by_group - overall_mean)^2,
             na.rm = TRUE)
  tss <- sum((num_var - overall_mean)^2, na.rm = TRUE)

  if (tss > 0) {
    eta <- sqrt(bss / tss)
    cor_val <- ifelse(eta^2 >= threshold_num, eta, 0)
    cor_type <- "Eta"
  }
}
```

This conditional structure ensures that the correct statistical method is applied for each type of variable pair, supporting a robust and interpretable exploration of associations.

3. Illustrative example

To demonstrate the core functionalities of AssociationExplorer, we use a curated subset of data from the European Social Survey (ESS), Round 11. The ESS is a large-scale, cross-national survey that measures attitudes, beliefs, and behaviors across European countries. The original dataset includes responses from over 46,000 individuals on topics such as politics, trust, well-being, media use, and health. The full ESS dataset, codebook, and documentation are freely available at `https://ess.sikt.no/en/` [5, 4].
For this example, we focus on the Belgian respondents, resulting in a reduced dataset of 1,594 individuals. We selected 60 variables covering areas highly relevant for understanding public opinion and everyday life in Belgium: interest in politics, confidence in institutions, lifestyle behaviors, perceived discrimination, vaccination, and more. These variables include both numbers (quantitative data) and labels or categories (qualitative data), making the dataset ideal for exploring diverse forms of associations.
This example is particularly relevant for our research project ODALON (Open multimodal Data for Automated Local News), which aims to develop a platform that supports the (semi-)automated production of local news in

Belgium. AssociationExplorer plays a key role in this effort by offering journalists, researchers, and citizens an intuitive tool to explore potentially newsworthy patterns in public and survey data, without requiring programming skills or statistical training.

Data preparation for the curated dataset was carried out in R and included:

- Filtering the dataset to include only Belgian respondents.
- Converting survey-specific nonresponse codes (e.g., `77`, `88`, `9999`, etc.) to `NA` values, based on the ESS codebook.
- Reversing response scales to ensure consistency (e.g., higher values always indicate stronger agreement or frequency).
- Recoding several categorical variables to have meaningful and interpretable labels (e.g., for gender, religion, political participation, or health behaviors).

The full R script used to perform this transformation is openly available in the `data` folder of the GitHub repository at `https://github.com/AntoineSoetewey/AssociationExplorer/tree/main/shiny_app/data`.

Once the dataset is uploaded into AssociationExplorer via the `Data` tab (see Figure **??**), users are guided through a step-by-step process. In addition to the main dataset, users can optionally upload a separate description file that provides human-readable explanations for each variable. In our example, this file was created using information from the official ESS codebook, allowing for clearer interpretation throughout the app interface. If no description file is provided, the application will automatically use the variable names themselves as default labels in all visualizations and summary tables. In the `Variables` tab, they select the variables they wish to explore, optionally assisted by a table of the variables' names and descriptions (see Figure **??**). Next, users can adjust the association thresholds in the `Correlation Network` tab to focus on the most meaningful relationships among the selected variables (see Figure **??**). Finally, the `Pairs Plots` tab displays detailed bivariate visualizations, including scatter plots, mean plots, and contingency tables, for each retained association (see Figure **??**).

This example shows how non-expert users, such as journalists or engaged citizens, can uncover unexpected or important relationships in a public opinion dataset. These insights can serve as the starting point for local news, public debates, or policy communication.

4. Impact

This is the main section of the article and reviewers will weight it appropriately. Please indicate:

- *Any new research questions that can be pursued as a result of your software.*
- *In what way, and to what extent, your software improves the pursuit of existing research questions.*
- *Any ways in which your software has changed the daily practice of its users.*
- *How widespread the use of the software is within and outside the intended user group (downloads, number of users if your software is a service, citable publications, etc.).*
- *How the software is being used in commercial settings and/or how it has led to the creation of spin-off companies.*

Please note that points 1 and 2 are best demonstrated by references to citable publications.

5. Conclusions

References

[1] Almende B.V. and Contributors and Thieurmel, B. (2022). *visNetwork: Network Visualization using 'vis.js' Library*. R package version 2.1.2.

[2] Attali, D. (2021). *shinyjs: Easily Improve the User Experience of Your Shiny Apps in Seconds*. R package version 2.1.0.

[3] Chang, W., Cheng, J., Allaire, J., Sievert, C., Schloerke, B., Xie, Y., Allen, J., McPherson, J., Dipert, A., and Borges, B. (2024). *shiny: Web Application Framework for R*. R package version 1.9.1.

[4] European Social Survey European Research Infrastructure (ESS ERIC) (2024a). ESS11 Data Documentation.

[5] European Social Survey European Research Infrastructure (ESS ERIC) (2024b). ESS11 integrated file, edition 3.0. [Data set].

[6] Harrell Jr, F. E. (2025). *Hmisc: Harrell Miscellaneous*. R package version 5.2-3.

[7] Kuhn, M., Jackson, S., and Cimentada, J. (2022). *corrr: Correlations in R*. R package version 0.4.4.

[8] Lares, B. (2025). *lares: Analytics & Machine Learning Sidekick*. R package version 5.2.11.

[9] Makowski, D., Ben-Shachar, M. S., Patil, I., and Lüdecke, D. (2020). Methods and algorithms for correlation analysis in r. *Journal of Open Source Software*, 5(51):2306.

[10] Makowski, D., Wiernik, B., Patil, I., Lüdecke, D., and Ben-Shachar, M. (2022). correlation: Methods for correlation analysis [r package].

[11] Patil, I. (2021). Visualizations with statistical details: The 'ggstatsplot' approach. *Journal of Open Source Software*, 6(61):3167.

[12] R Core Team (2024). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

[13] Schloerke, B., Cook, D., Larmarange, J., Briatte, F., Marbach, M., Thoen, E., Elberg, A., and Crowley, J. (2024). *GGally: Extension to 'ggplot2'*. R package version 2.2.1.

[14] Sievert, C., Cheng, J., and Aden-Buie, G. (2025). *bslib: Custom 'Bootstrap' 'Sass' Themes for 'shiny' and 'rmarkdown'*. R package version 0.9.0.

[15] Vallat, R. (2018). Pingouin: statistics in python. *Journal of Open Source Software*, 3(31):1026.

[16] Waskom, M. L. (2021). seaborn: statistical data visualization. *Journal of Open Source Software*, 6(60):3021.

[17] Wei, T. and Simko, V. (2024). *R package 'corrplot': Visualization of a Correlation Matrix*. (Version 0.95).