The Journal of Open Source Software

corrp: An R package for multiple correlation-like analysis and clustering in mixed data

Igor Dornelles Schoeller Siciliani [1,2] **and Paulo Henrique dos Santos** [1,2]

1 Meantrix, Brazil **2** Universidade Federal de Santa Catarina, Brazil

DOI: 10.21105/joss.07319

Software

- Review
- Repository
- Archive

Editor: Julia Romanowska

Reviewers:

- @devSJR
- @jromanowska

Submitted: 26 August 2024
Published: 27 May 2025

Summary

Correlation-like analysis provides an important statistical measure that describes the size and direction of an association between variables. However, there are few R packages that can efficiently perform this type of analysis on large datasets with mixed data types. The `corrp` package provides a full suite of solutions for computing various correlation-like measures, such as Pearson correlation (Pearson, 1895), Distance Correlation (Székely et al., 2007), Maximal Information Coefficient (MIC) (Reshef et al., 2011), Predictive Power Score (PPS) (Wetschoreck et al., 2020), Cramér's V (Cramér, 1946), and the Uncertainty Coefficient (Theil, 1972). These methods support the analysis of data frames with mixed classes (integer, numeric, factor, and character).

Additionally, it offers a C++ implementation of the Average Correlation Clustering Algorithm (ACCA) (Bhattacharya & De, 2010), which was originally developed for genetic studies using Pearson correlation as a similarity measure. In general, ACCA is an unsupervised clustering method, as it identifies patterns in the data without requiring predefined labels. Moreover, it requires the K parameter to be defined, similar to k-means. One of its main differences compared to other clustering methods is that it operates based on correlations rather than traditional distance metrics, such as Euclidean or Mahalanobis distance.

In this package, the ACCA algorithm has been extended to work directly with correlation matrices derived from different association methods, depending on the data types and user preferences. Furthermore, the package is designed for parallel processing in R, making it highly efficient for large datasets.

Statement of need

The `corrp` package is an R package that provides a flexible and efficient way of performing correlation-like analysis on mixed-type data frames. These datasets can contain different variable types, such as continuous (numeric), ordinal (ordered categorical), and nominal (unordered categorical) variables, which frequently arise in practical scenarios.

Moreover, most traditional correlation methods in R — such as `cor()` from the *stats package* (R Core Team, 2024), which computes Pearson, Spearman, or Kendall correlations between vectors, matrices, or data frames; and `rcorr()` from the *Hmisc package* (Jr, 2025), which efficiently computes Pearson or Spearman correlation matrices — are primarily designed for specific data types and may not generalize well to mixed data or large-scale applications. In this sense, `corrp` extends these capabilities by handling mixed data types, integrating various association methods, and offering clustering directly from the resulting correlation matrix..

The `corrp` package automatically detects the variable types present in the dataset. However, manual intervention is needed to select the appropriate correlation measure for each detected

variable pair (numeric pairs, categorical pairs, and numeric-categorical pairs) from the available options, as explained in more detail below.

The package is particularly useful for researchers and data scientists working with complex datasets who require robust and scalable tools for both association analysis and clustering.

Implementation

The `corrp` package integrates R and C++ to combine the flexibility of R with the speed of C++, optimizing key operations. Its core functionalities include the selection of correlation-like methods based on pairs of variable types (numeric pairs, numeric and categorical pairs, etc.). Users can create correlation matrices, remove variables based on significance, and cluster the correlation matrix using the ACCA clustering algorithm. This algorithm has been modified to support mixed data types and various correlation methods. Also, the package supports parallel processing through the `foreach` package, significantly improving performance on large datasets.

As mentioned before, one can choose between the following options based on the type of variable pair:

- **Numeric pairs (integer/numeric):**
  - Pearson correlation coefficient (Pearson, 1895), a widely used measure of the strength and direction of linear relationships.
  - Distance Correlation or distance covariance (Székely et al., 2007), based on the idea of expanding covariance to distances, can measure both linear and nonlinear associations between variables.
  - Maximal Information Coefficient (MIC) (Reshef et al., 2011), an information-based nonparametric based method that can detect linear or non-linear relationships between variables.
  - Predictive Power Score (PPS) (Wetschoreck et al., 2020), a metric used to assess predictive relations between variables.
- **Numeric and categorical pairs (integer/numeric - factor/categorical):**
  - Square root of the R^2 coefficient from linear regression (Cohen, 1983).
  - PPS (Wetschoreck et al., 2020).
- **Categorical pairs (factor/categorical):**
  - Cramér's V (Cramér, 1946), a measure of association between nominal variables.
  - Uncertainty Coefficient (Theil, 1972), a measure of nominal association between two variables.
  - PPS (Wetschoreck et al., 2020).

In R, various statistical functions are available to measure these correlations. Below follows a list of correlation techniques and their corresponding R functions:

- **Linear Model** → `stats::lm`
- **Pearson Correlation** → `stats::cor.test`
- **Distance Correlation** → `corrp::dcor_t_test`
- **MIC** → `minerva::mine`
- **PPS** → `ppsr::score`
- **Uncertainty Coefficient** → `DescTools::UncertCoef`
- **Cramer's V** → `lsr::cramersV`

An important point to note is that some methods, such as the square root of R^2, PPS, and the Uncertainty Coefficient, are asymmetric. In other words, the correlation value between two variables, A and B, may not be the same as the correlation between B and A in the correlation matrix.

Usage

The `corrp` package provides seven main functions for correlation calculations, clustering, and basic data manipulation:

- **corrp**: Performs correlation-like analysis with user-specified methods for numeric, categorical, factor, integer and mixed pairs.
- **corr_matrix**: Generates a correlation matrix from analysis results.
- **corr_rm**: Removes variables based on p-value significance.
- **acca**: Performs the ACCA clustering algorithm with added support for mixed data types.
- **sil_acca**: A C++ implementation of the Silhouette method for interpreting and validating the consistency of clusters within ACCA clusters of data.
- **best_acca**: Determining the optimal number of clusters in ACCA clustering using the average silhouette approach.

We calculate correlations for the *eusilc* dataset using the MIC for numeric pairs, PPS for numeric/categorical pairs, and Uncertainty Coefficient for categorical pairs. This synthetic dataset represents Austrian EU-SILC data on income, demographics, and household characteristics (Alfons et al., 2024).

```
set.seed(2024)
library("laeken")
library("corrp")
data(eusilc)

eusilc <- eusilc[, c("eqSS", "eqIncome", "db040", "rb090")]
colnames(eusilc) <- c("House_Size", "Income", "State", "Sex")

results <- corrp(
  eusilc,
  cor.nn = 'dcor', cor.nc = 'lm', cor.cc = 'pps',
  verbose = FALSE
)

head(results$data)
```

infer	infer.value	stat	stat.value	isig	msg	varx	vary
Distance Correlation	1.000	P-value	0.000	TRUE		House_Size	House_Size
Distance Correlation	0.008	P-value	0.000	TRUE		House_Size	Income
Linear Model	0.146	P-value	3.57e-64	TRUE		House_Size	State
Linear Model	0.071	P-value	4.79e-18	TRUE		House_Size	Sex
Distance Correlation	0.008	P-value	0.000	TRUE		Income	House_Size
Distance Correlation	1.000	P-value	0.000	TRUE		Income	Income

When choosing correlation methods, it's important to think about their performance for different pair types. For **numeric pairs**, **Pearson** (`pearson`) is the quickest and most efficient option, while the **Maximal Information Coefficient** (`mic`) is significantly slower, making it less suitable for large datasets. **Distance correlation** (`dcor`) is a better performer than MIC but still not the fastest choice, while **Predictive Power Score** (`pps`) is efficient but may take longer than Pearson. For **numeric-categorical pairs**, the **linear model** (`lm`) typically outperforms `pps`. In **categorical pairs**, **Cramér's V** (`cramersV`), **Uncertainty Coefficient** (`uncoef`), and `pps` are options, with `uncoef` being the slowest of the three.

As the number of columns in the data increases, the runtime will also increase due to the `N * N` scaling. Therefore, the user should choose the methods wisely to ensure efficient performance.

Using the previous result, we can create a correlation matrix as follows:

```
m <- corr_matrix(results, col = 'infer.value', isig = TRUE)
m
```

	House_Size	Income	State	Sex
House_Size	1.000	0.008	0.146	0.071
Income	0.008	1.000	0.070	0.071
State	0.146	0.070	1.000	0.000
Sex	0.071	0.071	0.000	1.000

```
# attr(,"class")
# [1] "cmatrix" "matrix"
```

Finally, we can cluster the dataset variables using the ACCA algorithm and the correlation matrix. For example, consider clustering into 2 groups ($k = 2$):

```
acca.res <- acca(m, 2)
acca.res
# $cluster1
# [1] "Sex"    "Income"
#
# $cluster2
# [1] "State"      "House_Size"
#
# attr(,"class")
# [1] "acca_list" "list"
```

Performance Improvements

When using the `corrp` function with the `dcor` method for numeric pairs (i.e., `cor.nn = "dcor"`), significant improvements in both memory usage and runtime are observed. This is because the `corrp` package uses a C++ implementation of distance correlation (`dcor_t_test`), which is more efficient than the `energy::dcorT.test` function from the `energy` package.

For example, using two vectors of length 10000 and 20000, the benchmarks show the following improvements:

Method	10,000		20,000	
	Memory (MB)	Time (sec)	Memory (MB)	Time (sec)
dcor_t_test (C++)	4701.44	6.022	18440.54	25.977
energy::dcorT.test	7000.65	13.846	27598.38	60.264

This highlights a substantial reduction in both memory usage and execution time, making the `corrp` package more scalable for larger datasets when applying distance correlation methods. The memory reduction is particularly important because calculating distance correlation requires constructing a distance matrix of size N^2, where N is the length of the input vector. As N grows, the memory demands can quickly become prohibitive.

Acknowledgements

We acknowledge the contributions of the Meantrix team (https://github.com/meantrix) and thank Srikanth Komala Sheshachala (https://github.com/talegari) for the initial inspiration from the `cor2` function (Sheshachala, 2025).

References

Alfons, A., Templ, M., & Filzmoser, P. (2024). *Laeken: Estimation of indicators on social exclusion and poverty, as well as Pareto tail modeling for empirical income distributions*. https://doi.org/10.32614/CRAN.package.laeken

Bhattacharya, A., & De, R. K. (2010). Average correlation clustering algorithm (ACCA) for grouping of co-regulated genes with similar pattern of variation in their expression values. *Journal of Biomedical Informatics*, *43*(4), 560–568. https://doi.org/10.1016/j.jbi.2010.02.001

Cohen, J. (1983). *Applied multiple regression/correlation analysis for the behavioral sciences* (2nd ed.). Lawrence Erlbaum Associates. https://doi.org/10.4324/9780203774441

Cramér, H. (1946). Mathematical methods of statistics. *Princeton University Press*. https://doi.org/10.1515/9781400883868

Jr, F. E. H. (2025). *Hmisc: Harrell miscellaneous*. https://doi.org/10.32614/CRAN.package.Hmisc

Pearson, K. (1895). Notes on regression and inheritance in the case of two parents. *Proceedings of the Royal Society of London*, *58*, 240–242. https://doi.org/10.1098/rspl.1895.0041

R Core Team. (2024). *The R stats package: Statistical functions*. https://stat.ethz.ch/R-manual/R-devel/library/stats/html/00Index.html

Reshef, D. N., Reshef, Y. A., Finucane, H. K., Grossman, S. R., McVean, G., Turnbaugh, P. J., Lander, E. S., Mitzenmacher, M., & Sabeti, P. C. (2011). Detecting novel associations in large data sets. *Science*, *334*(6062), 1518–1524. https://doi.org/10.1126/science.1205438

Sheshachala, S. K. (2025). *Sidekicks: A misc set of functions for data analysis*. https://github.com/talegari/sidekicks.

Székely, G. J., Rizzo, M. L., & Bakirov, N. K. (2007). Measuring and testing independence by correlation of distances. *The Annals of Statistics*, *35*(6), 2769–2794. https://doi.org/10.1214/009053607000000505

Theil, H. (1972). *Statistical decomposition analysis: With applications in the social and administrative sciences*. North-Holland Publishing Company. ISBN: 978-0444103789

Wetschoreck, F., Krabel, T., & Krishnamurthy, S. (2020). *8080labs/ppscore: Zenodo release*. Zenodo. https://doi.org/10.5281/ZENODO.4091345