CCRVAM: A Python Package

for Model-Free Exploratory Analysis of Multivariate Discrete Data with an Ordinal Response Variable

Dhyey Mavani
Advisor: Prof. Shu-Min Liao
Readers: Prof. Nick Horton, Prof. Amy Wagaman

Department of Statistics
Amherst College

STAT Honors Thesis Defense
April 2025

Table of Contents

Table of Contents

Breaking Down CCRVAM

Model-Free

No assumptions about underlying models; purely data-driven.

Exploratory Analysis

Discover patterns, trends, and relationships without formal hypotheses.

Multivariate Discrete Data

Multiple variables, each with separate categories (not continuous.)

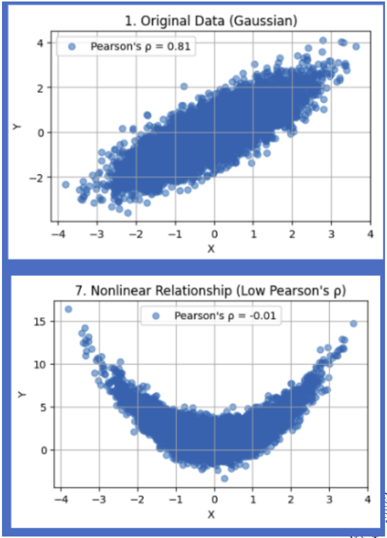
Ordinal Response Variable

Outcome is ordered (e.g., low/medium/high) but not necessarily equally spaced.

Why Not Just Use Correlation?

- **Pearson correlation** only captures linear relationships.
- It is sensitive to marginals.
- **Limitations:**
  - Doesn't always exist (e.g., non-linear relationships).
  - Not invariant under transformations.
  - $\rho = 0$ does **not** imply independence.
- *We need something* ***model-free*** *and* ***margin-independent****.*

Idea: Use copulas to separate *margins* from *dependence structure*!



Copulas: Capturing Dependence, Margin-Free

- **Copula:** A function that joins univariate margins into a joint distribution. By **Sklar's Theorem:** $F(x, y) = C(F_X(x), F_Y(y))$.
- **Invariance:** C stays the same under monotonic transforms
- Capture complex (nonlinear) dependencies.

Challenge

EDA for Categorical Data is a real challenge especially due to the **curse of high-dimensionality.**

Why Checkerboard Copulas?

For **discrete data**, copulas aren't unique.
Checkerboard Copulas fill the "gaps" through bilinear-extension:

- A valid, unique extension & Piecewise-constant density
- A path to regression & association measures

Table of Contents

Checkerboard Copula Regression: Key Concepts

What is it?

A nonparametric, model-free method for exploring how an **ordinal response** relates to **categorical predictors.**

- **Checkerboard Copula (CC)**
  - discretized copula built from contingency tables
- **Copula Scores (CCS)** — numeric values for ordered categories:

$$s^j_{i_j} = \frac{u^j_{i_j-1} + u^j_{i_j}}{2}, \quad \text{where } u^j_{i_j} = \sum_{k_j=1}^{i_j} P(X_j = x_{k_j})$$

- **CCR Function** — conditional mean of copula scores:

$$r_{U_j|\mathbf{U}_{-j}}(\mathbf{u}_{-j}) = \sum_{i_j} P(i_j|\mathbf{i}_{-j}) \cdot s^j_{i_j}$$

- **Interpretation:** Average rank of response $\Rightarrow$ ordinal prediction

Prediction & Association from Checkerboard Copulas

Prediction (CCR)

- CCR + marginal distribution $\implies$ Predicted response category
- Works for any mix of ordinal and nominal predictors

Association Measure (S)CCRAM

$$\rho^2 = 12 \cdot \mathbb{E}\left[\left(r_{U_j|\mathbf{U}_{-j}} - 0.5\right)^2\right], \quad \rho^{2*} = \frac{\rho^2}{12\sigma^2_{S_j}}$$

- ρ^2: strength of regression dependence
- ρ^{2*}: normalized (scaled to [0,1]) (R^2**-equivalent**)

- Bootstrap $\rightarrow$ prediction / CCRAM's uncertainty
- Permutation test $\rightarrow$ statistical significance of a given CCRAM

Table of Contents

`ccrvam`: Novel Python Package for Categorical EDA

Motivation

- **Lack of robust tools** for multivariate categorical data with ordinal responses (especially for methods in [39] and [22])

Our Goals

- Bridge copula theory, multivariate categorical analysis, and parallel computation
- Provide end-to-end EDA workflows in Python

PyPI Published Link: `https://pypi.org/project/ccrvam/`
Documentation: `https://ccrvam.readthedocs.io`
Install: `pip install ccrvam`

Flexible Data Loading in ccrvam

Supported Formats

- **Case Form:** One row per observation of a category-combination
- **Frequency Form:** Category tuple + count embedding
- **Contingency Table Form:** Multidimensional NumPy array
- Internal input validation pre-implemented for safety

```python
# Load data from the cases
contingency_table = DataProcessor.load_data(
    "caseform.pain.txt", "cases_form", (2, 3, 2, 6),
    ↪  var_list_4d, category_map_4d, named=True
)
# Initialize the GenericCCRVAM object
ccrvam_obj =
↪  GenericCCRVAM.from_contingency_table(contingency_table)
```

Key Features of `ccrvam`

Core Analytical Capabilities

- CCRVAM objects from multi-way contingency tables
- Calculation of marginal and cumulative distributions
- Checkerboard Copula Regression (CCR), CCRAM and SCCRAM
- Statistical inference via bootstrap and permutation tests
- Heatmaps, prediction plots, and confidence plots

Scalability, Customization, and Production Readiness

- **Optimized for speed:** vectorized, cache-aware & parallel-ready
- **Highly customizable:** flexible variable naming & display options
- **Well-tested:** full unit test suite with Pytest, 93%+ coverage
- **Rich documentation** via Sphinx, hosted on ReadTheDocs
- **Example workflows** using Jupyter Notebooks

Modular Software Architecture

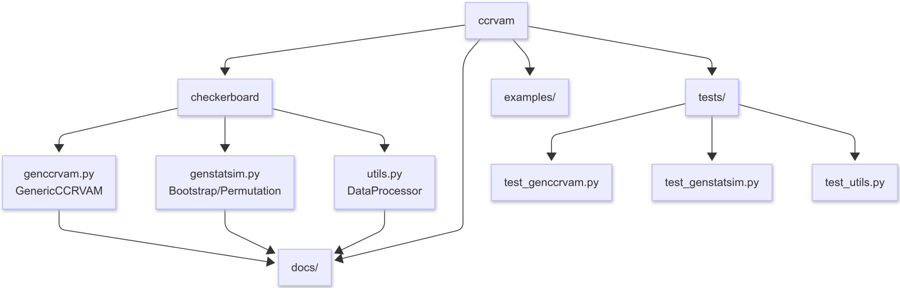


Figure: Modular organization of core engine, simulations, and utilities

User Workflow Overview

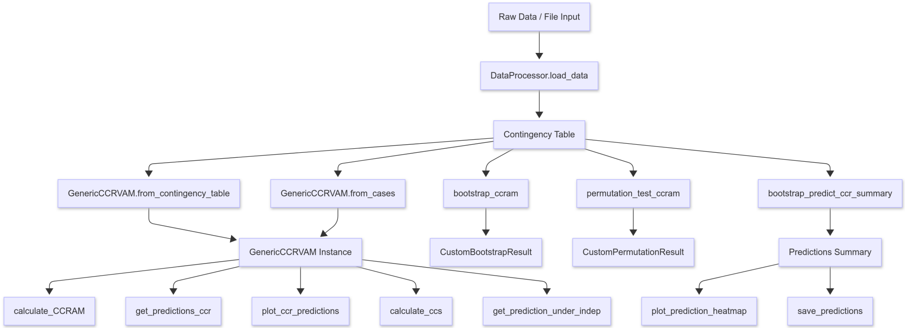


Figure: From data to insight — a streamlined EDA pipeline

CI/CD Pipeline

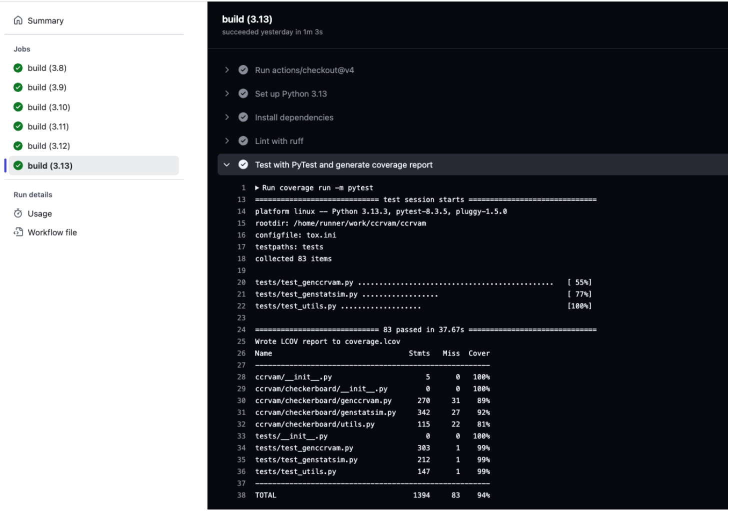
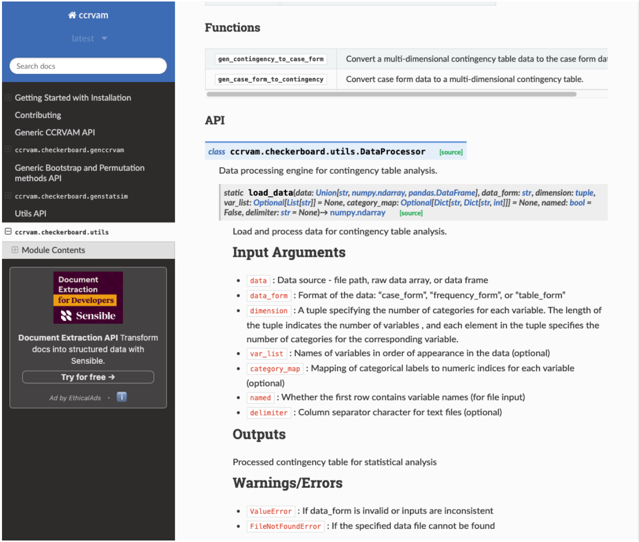


Figure: Continuous Integration and Testing Pipeline through GitHub Actions supporting Python versions from 3.8 to 3.13

ReadTheDocs Documentation



ccrvam

latest

Search docs

Getting Started with Installation
Contributing
Generic CCRVAM API
ccrvam.checkerboard.genccrvam
Generic Bootstrap and Permutation methods API
ccrvam.checkerboard.genstatsim
Utils API
ccrvam.checkerboard.utils
Module Contents

Functions

`gen_contingency_to_case_form`	Convert a multi-dimensional contingency table data to the case form dat
`gen_case_form_to_contingency`	Convert case form data to a multi-dimensional contingency table.

API

class **`ccrvam.checkerboard.utils.DataProcessor`** [source]

Data processing engine for contingency table analysis.

static **`load_data`**(*data: Union[str, numpy.ndarray, pandas.DataFrame], data_form: str, dimension: tuple, var_list: Optional[List[str]] = None, category_map: Optional[Dict[str, Dict[str, int]]] = None, named: bool = False, delimiter: str = None*)→ numpy.ndarray [source]

Load and process data for contingency table analysis.

Input Arguments

- `data` : Data source - file path, raw data array, or data frame
- `data_form` : Format of the data: “case_form”, “frequency_form”, or “table_form”
- `dimension` : A tuple specifying the number of categories for each variable. The length of the tuple indicates the number of variables , and each element in the tuple specifies the number of categories for the corresponding variable.
- `var_list` : Names of variables in order of appearance in the data (optional)
- `category_map` : Mapping of categorical labels to numeric indices for each variable (optional)
- `named` : Whether the first row contains variable names (for file input)
- `delimiter` : Column separator character for text files (optional)

Outputs

Processed contingency table for statistical analysis

Warnings/Errors

- `ValueError` : If data_form is invalid or inputs are inconsistent
- `FileNotFoundError` : If the specified data file cannot be found

Figure: Sphinx-Powered Detailed API Library Documentation

Table of Contents

Performance Benchmarking

Key Optimization Strategies

- **Vectorized operations**: Replace Python loops + interoperability.
- **Cached statistics**: Reuses conditionals to avoid redundant work.
- **Sparse-aware structures**: Low-latency access for large tables.

Scalability in Practice

- (S)CCRAM bootstrap on 2×3×2×6 table: $<$ **5 seconds** on MacOSX.
- `bootstrap_predict_ccr_summary()`: (15 seconds)
  - Native parallelism via `ProcessPoolExecutor`
  - **3× speedup** on 7 cores (5 seconds)
  - **15× speedup** on 63 cores (FrostByte HPC) (1 second)
- **GPU-Ready:** Supports CuPy for seamless CUDA acceleration.

Efficient. Scalable. Ready for High-Dimensional Data.

Reflections & Lessons Learned

What Implementation Taught Us

- Translating theory into code deepened understanding.
- Visualization development improved **user-friendliness** especially since Python was more verbose than R.
- **Unit testing** became a form of **error-proofing** our software and it often revealed gaps in reasoning.

Statistical Thinking as Engineering

- Building reusable components encouraged **abstractions.**
- Constraints sharpened insight into impact of **optimizations.**

Lesson: *Theory & Code Co-Evolve—They Clarify Each Other*

Table of Contents

Dataset Overview

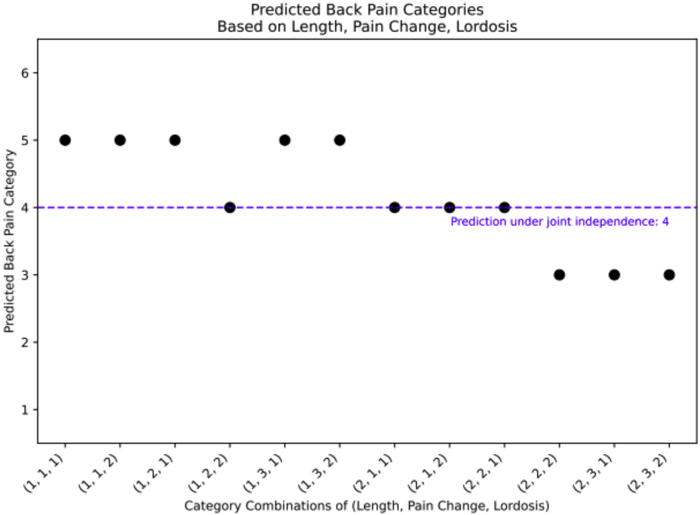
Clinical Back Pain Study [1]

- **Response (Ordinal):** Pain outcome — 6 ordered levels {Worse, Same, SI, MODI, MARI, Complete Relief})
- **Predictors (Categorical):**
  - X_1: Length of Previous Attack (Short, Long)
  - X_2: Pain Change (Better, Same, Worse)
  - X_3: Lordosis (Absent/Decreasing, Present/Increasing)
- Total predictors' combinations: $2 \times 3 \times 2 = 12$

***Goal:** Understand how categorical predictors affect ordered outcomes.*

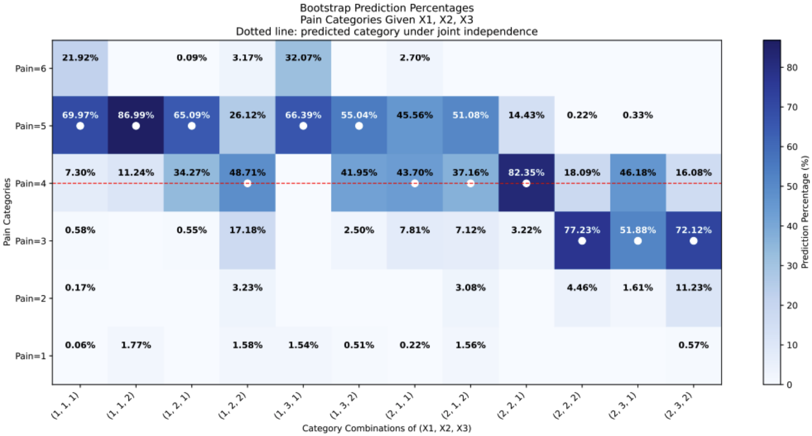
CCR Predictions

- $X_1 = 1$ (Short) patients $\rightarrow$ generally better outcomes.
- Joint Independence Baseline is Moderate Improvement (MODI)



Bootstrap Confidence in Predictions

- Most confident prediction: (`Short`, `Better`, `Present`) → Category 5
- Least confident prediction: (`Long`, `Better`, `Present`)
  → Split between Categories 4 & 5



(S)CCRAM Uncertainty & Significance Testing

Point Estimation Results

- **CCRAM:** 0.2576
- **SCCRAM:** 0.2687 (normalized)
- ∼27% of pain outcome variability explained by the 3 predictors.
- Scaled version aids in cross-study comparisons.

Bootstrap (9999 resamples)

- CCRAM CI: (0.185, 0.476)
- SCCRAM CI: (0.069, 0.351)

Permutation Testing

- **p-value (CCRAM)** = 0.0016
- **p-value (SCCRAM)** = 0.0011

Table of Contents

Future Directions

Theoretical Extensions

- Formalize **conditional (S)CCRAM** to assess partial associations while controlling for confounders. [40]
- Extend CCR framework to **longitudinal structures.** [22]
- Explore **interpretable decompositions** of (S)CCRAM values to enhance explainability. [40]

Computational and Software Advances

- Improve scalability via **sparse structures, approximate inference**, and internal **GPU-accelerated computation.**
- Develop **interactive visualization tools + bubble-plots** for exploring multivariate dependence dynamically.

Vision: *Interpretable, Scalable, and Accessible Dependence Analysis for Categorical Data*

Acknowledgements

- Grateful to my grandparents, parents, and sister for their unwavering support and belief in me.
- Deep thanks to **Prof. Shu-Min Liao** for exceptional mentorship, and to **Prof. Daeyoung Kim** for software guidance.
- Appreciation to **Profs. Horton, Wagaman, Ishii, Riondato, Correia, Matheson, and Bailey** for nuggets of wisdom through teaching, advising, and encouragement.
- Thanks in advance to **Prof. Horton** (my college advisor), and **Prof. Correia** (honors program coordinator)!
- Grateful to **Prof. Bailey** for formatting support, and for the `.qmd` thesis template.
- Thankful for opportunities within and beyond Amherst College!
- And to everyone who helped in small or big ways—thank you!

References I

[1] J. A. Anderson. Regression and ordered categorical variables. *Journal of the Royal Statistical Society. Series B (Methodological)*, 46(1):1–30, 1984. ISSN 00359246. URL http://www.jstor.org/stable/2345457.

[2] F. J. Anscombe. Graphs in statistical analysis. *The American Statistician*, 27(1):17–21, 1973. ISSN 00031305, 15372731. URL http://www.jstor.org/stable/2682899.

[3] Ned Batchelder and Contributors to Coverage.py. Coverage.py: The code coverage tool for python. https://coverage.readthedocs.io/, 2025. Version 7.8.0. Licensed under Apache 2.0. Repository: https://github.com/nedbat/coveragepy.

[4] A. Buja, D. Cook, H. Hofmann, M. Lawrence, E.-K. Lee, D. F. Swayne, and H. Wickham. Statistical inference for exploratory data analysis and model diagnostics. *Philosophical Transactions of the Royal Society of London*, 367:4361–4383, 2009.

[5] A. C. Davison and D. V. Hinkley. *Bootstrap Methods and Their Application*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, Cambridge, 1997. ISBN 0-521-57391-2.

[6] M. Denuit and P. Lambert. Constraints on concordance measures in bivariate discrete data. *Journal of Multivariate Analysis*, 93:40–57, 2005.

[7] NumPy Developers. Numpy 2.2.4: Fundamental package for array computing in python. https://pypi.org/project/numpy/, 2025. Released: Mar 16, 2025. Powered by NumFOCUS. License: BSD.

[8] D. L. Donoho. 50 years of data science. *Journal of Computational and Graphical Statistics*, 26:745–766, 2017.

[9] B. Efron. Better bootstrap confidence intervals. *Journal of the American Statistical Association*, 82(397):171–185, 1987. doi: 10.2307/2289144.

[10] Arturo Erdely. A subcopula based dependence measure. *Kybernetika*, 53(2):231–243, 2017. URL http://eudml.org/doc/288202.

[11] Olivier P. Faugeras. Inference for copula modeling of discrete data: a cautionary tale and some facts. *Dependence Modeling*, 5(1):121–132, 2017. doi: doi:10.1515/demo-2017-0008. URL https://doi.org/10.1515/demo-2017-0008.

[12] Gery Geenens. Copula modeling for discrete random vectors. *Dependence Modeling*, 8(1):417–440, 2020. doi: doi:10.1515/demo-2020-0022. URL https://doi.org/10.1515/demo-2020-0022.

[13] A. Gelman and A. Vehtari. What are the most important statistical ideas of the past 50 years? *Journal of the American Statistical Association*, 116:2087–2097, 2021.

References II

[14] C. Genest, J. G. Nešlehová, and B. Rémillard. On the empirical multilinear copula process for count data. *Bernoulli*, 20: 1344–1371, 2014.

[15] Christian Genest, Johanna Nešlehová, and Bruno Remillard. Asymptotic behavior of the empirical multilinear copula process under broad conditions. *Journal of Multivariate Analysis*, 159, 04 2017. doi: 10.1016/j.jmva.2017.04.002.

[16] Tim Hesterberg. What teachers should know about the bootstrap: Resampling in the undergraduate statistics curriculum, 2014. URL https://arxiv.org/abs/1411.5279.

[17] Marius Hofert, Ivan Kojadinovic, Martin Maechler, and Jun Yan. *Elements of Copula Modeling with R*. Springer Use R! Series, 2018. ISBN 978-3-319-89635-9. URL http://www.springer.com/de/book/9783319896342.

[18] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007. doi: 10.1109/MCSE.2007.55.

[19] Richard Iannone, Joe Cheng, Barret Schloerke, Ellis Hughes, Alexandra Lauer, JooYoung Seo, Ken Brevoort, and Olivier Roy. *gt: Easily Create Presentation-Ready Display Tables*, 2024. URL https://gt.rstudio.com. R package version 0.11.0.9000, https://github.com/rstudio/gt.

[20] H. Joe. *Dependence Modeling with Copulas*. Chapman and Hall/CRC, New York, 2014.

[21] Holger Krekel. pytest 8.3.5: Simple powerful testing with python. https://pypi.org/project/pytest/, 2025. Released: Mar 2, 2025. License: MIT.

[22] Shu-Min Liao, Li Wang, and Daeyoung Kim. *Visualization of Dependence in Multidimensional Contingency Tables with an Ordinal Dependent Variable via Copula Regression*, pages 517–538. Springer International Publishing, Cham, 2024. ISBN 978-3-031-56318-8. doi: 10.1007/978-3-031-56318-8_21. URL https://doi.org/10.1007/978-3-031-56318-8_21.

[23] Wes Mckinney. pandas: a foundational python library for data analysis and statistics. *Python High Performance Science Computer*, 01 2011.

[24] Roger B. Nelsen. *An introduction to copulas*. Springer Science business media, 2006. ISBN 978-0-387-28659-4. URL http://www.springer.com/de/book/9780387286594.

[25] J. Nešlehová. On rank correlation measures for non-continuous random variables. *Journal of Multivariate Analysis*, 98 544–567, 2007.

References III

[26] Ryosuke Okuta, Yuya Unno, Daisuke Nishino, Shohei Hido, and Crissman Loomis. Cupy: A numpy-compatible library for nvidia gpu calculations. In *Proceedings of Workshop on Machine Learning Systems (LearningSys) in The Thirty-first Annual Conference on Neural Information Processing Systems (NIPS)*, 2017. URL `http://learningsys.org/nips17/assets/papers/paper_16.pdf`.

[27] Ludger Rüschendorf. On the distributional transform, sklar's theorem, and the empirical copula process. *Journal of Statistical Planning and Inference*, 139(11):3921–3927, 2009. ISSN 0378-3758. doi: 10.1016/j.jspi.2009.05.030. URL `https://www.sciencedirect.com/science/article/pii/S037837580900158X`. Special Issue: The 8th Tartu Conference on Multivariate Statistics The 6th Conference on Multivariate Distributions with Fixed Marginals.

[28] B. Schweizer and A. Sklar. Operations on distribution functions not derivable from operations on random variables. *Studia Mathematica*, 52:43–52, 1974.

[29] C. Shearer. The crisp-dm model: the new blueprint for data mining. *Journal of Data Warehousing*, 5:13–22, 2000.

[30] Daniel D. Sjoberg, Karissa Whiting, Michael Curry, Jessica A. Lavery, and Joseph Larmarange. Reproducible summary tables with the gtsummary package. *The R Journal*, 13:570–580, 2021. doi: 10.32614/RJ-2021-053. URL `https://doi.org/10.32614/RJ-2021-053`.

[31] M Sklar. Fonctions de repartition an dimensions et leurs marges. *Publ. inst. statist. univ. Paris*, 8:229–231, 1959.

[32] Knut Sveidqvist and Contributors to Mermaid. *Mermaid: Generate diagrams from markdown-like text*, 2014. Version 1.2.0. Repository: `https://github.com/mermaid-js/mermaid`.

[33] E. R. Tufte. *The Visual Display of Quantitative Information*. Graphics Press, Cheshire, CT, 1983.

[34] J.W. Tukey. *Exploratory Data Analysis*. AddisonWesley; Boston, MA, 1977.

[35] Adam Turner. Sphinx 8.2.3: Python documentation generator. `https://pypi.org/project/Sphinx/`, 2025. Released: Mar 2, 2025. License: BSD-2-Clause. Requires Python 3.11+.

[36] Kevin Ushey and Hadley Wickham. *renv: Project Environments*, 2024. URL `https://rstudio.github.io/renv/`. R package version 1.0.7, https://github.com/rstudio/renv.

[37] Christophe VG. pypi-template 0.8.0: Template-based common/best practices for managing a python package on pypi. `https://pypi.org/project/pypi-template/`, 2024. Released: Nov 25, 2024. License: MIT.

References IV

[38] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, Aditya Vijaykumar, Alessandro Pietro Bardelli, Alex Rothberg, Andreas Hilboll, Andreas Kloeckner, Anthony Scopatz, Antony Lee, Ariel Rokem, C. Nathan Woods, Chad Fulton, Charles Masson, Christian Häggström, Clark Fitzgerald, David A. Nicholson, David R. Hagen, Dmitrii V. Pasechnik, Emanuele Olivetti, Eric Martin, Eric Wieser, Fabrice Silva, Felix Lenders, Florian Wilhelm, G. Young, Gavin A. Price, Gert-Ludwig Ingold, Gregory E. Allen, Gregory R. Lee, Hervé Audren, Irvin Probst, Jörg P. Dietrich, Jacob Silterra, James T Webber, Janko Slavič, Joel Nothman, Johannes Buchner, Johannes Kulick, Johannes L. Schönberger, José Vinícius de Miranda Cardoso, Joscha Reimer, Joseph Harrington, Juan Luis Cano Rodríguez, Juan Nunez-Iglesias, Justin Kuczynski, Kevin Tritz, Martin Thoma, Matthew Newville, Matthias Kümmerer, Maximilian Bolingbroke, Michael Tartre, Mikhail Pak, Nathaniel J. Smith, Nikolai Nowaczyk, Nikolay Shebanov, Oleksandr Pavlyk, Per A. Brodtkorb, Perry Lee, Robert T. McGibbon, Roman Feldbauer, Sam Lewis, Sam Tygier, Scott Sievert, Sebastiano Vigna, Stefan Peterson, Surhud More, Tadeusz Pudlik, Takuya Oshima, Thomas J. Pingel, Thomas P. Robitaille, Thomas Spura, Thouis R. Jones, Tim Cera, Tim Leslie, Tiziano Zito, Tom Krauss, Utkarsh Upadhyay, Yaroslav O. Halchenko, and Yoshiki Vázquez-Baeza. Scipy 1.0: fundamental algorithms for scientific computing in python. *Nature Methods*, 17(3):261–272, February 2020. ISSN 1548-7105. doi: 10.1038/s41592-019-0686-2. URL `http://dx.doi.org/10.1038/s41592-019-0686-2`.

[39] Zheng Wei and Daeyoung Kim. On exploratory analytic method for multi-way contingency tables with an ordinal response variable and categorical explanatory variables. *Journal of Multivariate Analysis*, 186:104793, 08 2021. doi: 10.1016/j.jmva.2021.104793.

[40] Zheng Wei, Li Wang, Shu-Min Liao, and Daeyoung Kim. On the exploration of regression dependence structures in multidimensional contingency tables with ordinal response variables. *Journal of Multivariate Analysis*, 196:105179, 2023. ISSN 0047-259X. doi: 10.1016/j.jmva.2023.105179. URL `https://www.sciencedirect.com/science/article/pii/S0047259X23000258`.

[41] Hao Zhu. *kableExtra: Construct Complex Table with 'kable' and Pipe Syntax*, 2024. URL `https://CRAN.R-project.org/package=kableExtra`. R package version 1.4.0.

Thank you for being a wonderful audience!

Let’s chat if you’re curious about anything!

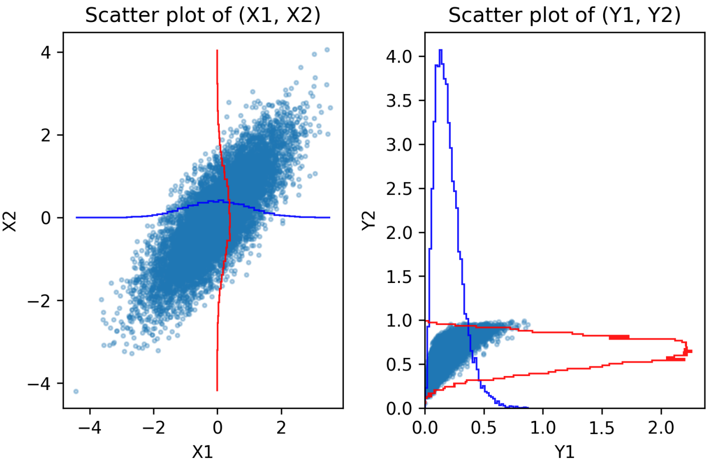
I’m all ears for questions, feedback, and collaborations.

A detailed list of references, and additional content details can be found in the thesis write-up.

Table of Contents

Motivation: Beyond Pearson Correlation

- Scatter plots of (X_1, X_2) vs (Y_1, Y_2) show different structures.
- Pearson correlations: $\rho(X_1, X_2) \approx 0.802$ and $\rho(Y_1, Y_2) \approx 0.755$
- *Limitation*: captures only **linear** dependence.



Limitations of Pearson's ρ

- Only detects linear dependence
- Sensitive to marginals and transformations
- $\rho = 0$ does not imply independence
- May not exist (e.g., heavy-tailed distributions)

Conclusion: Need margin-free, nonlinear measures.

Association vs. Dependence: A Closer Look

Dependence:

- Knowing X gives information about Y.
- Formally: $P(X, Y) \neq P(X)P(Y)$.
- Any relationship — not necessarily linear or monotonic.

Association:

- Usually monotonic (increasing/decreasing).
- Large $X \rightarrow$ large Y (+ve association) or small Y (-ve association).
- Measured by correlation, Spearman's ρ, Kendall's τ, etc.

Example: $X \sim \mathcal{N}(0, 1)$

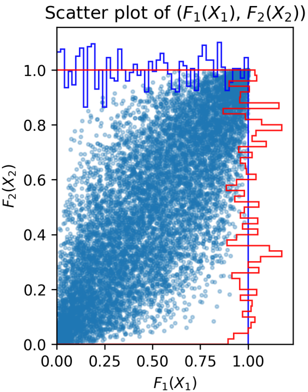
- $Y = X^2$: (perfectly dependent)
  - No association — Y does not $\uparrow$ or $\downarrow$ consistently with X.
- $Y = X^3$: (perfectly dependent too!)
  - Strong positive association (monotonic relationship).

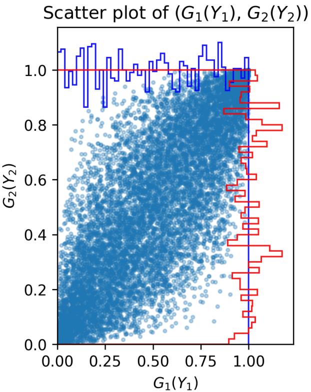
Summary: All Associations are Dependencies, but NOT vice-versa.

Uniformizing Marginals: PIT and Standardization

Probability Integral Transform (PIT): $F(X) \sim$ Uniform$(0,1)$ if $X \sim F$.
Quantile Transformation: $F^{-1}(U) \sim F$, where $U \sim$ Uniform$(0,1)$

- PIT isolates pure dependence.
- Quantile match restores common marginals.





Copulas: Formal Definitions & Theorems

Subcopula (for discrete data)

Function $C^S : D_1 \times D_2 \to [0,1]$ satisfying:

- Groundedness: $C^S(u,0) = C^S(0,v) = 0$
- Marginal consistency
- 2-increasing property

Copula

Full extension to $[0,1]^2$ domain.

Sklar's Theorem: $H(x,y) = C(F(x), G(y))$

- Copula separates margins from dependence

Invariance Principle: Strictly increasing transforms preserve copulas.

Kendall's τ and Spearman's ρ (Continuous Data)

$$\tau = \mathbb{E}\left[\text{sign}((X_1 - X_1')(X_2 - X_2'))\right] \qquad \rho = \rho_{\text{pearson}}(F(X_1), F(X_2))$$

- **Rank-based**: depend only on the ordering of values, not magnitudes.
- **Margin-free**: invariant under strictly increasing transformations (PIT).
- **Interpretation**:
  - τ: probability difference between concordant and discordant pairs.
  - ρ: linear correlation of uniformized marginals.
- **Advantage**: capture monotonic relationships without assuming linearity.

Discrete Data: Challenges and Checkerboard Copula

- Discrete margins $\Rightarrow$ subcopulas only on grids.
- No unique full copula; need extensions. Some options are:
  - Adding Random Noise to discrete data-points
  - Performing multilinear extension of distributions
- **Checkerboard Copula**: fills gaps using multilinear interpolation.

$$c^{+}(\mathbf{u}) = \frac{p_{i_1,\ldots,i_d}}{\prod_j p_{+i_j+}}$$

Checkerboard Rank-Based Measures: τ^+ and ρ^+

$$\tau^+ = 4\int_{[0,1]^2} C^+(u,v)dC^+(u,v) - 1 \qquad \rho^+ = 12\int_{[0,1]^2} C^+(u,v)dudv - 3$$

- Properly normalized
- Monotonic transformation invariance
- Robust for discrete data

Limitation: Only defined and valid for bivariate cases.

Beyond Bivariate: High-Dimensional Dependence

- Real-world categorical data often multivariate.
- Need scalable, model-free association measures.
- **Checkerboard Copula Regression and (S)CCRAM** generalize these ideas.

Goal: Model-free categorical EDA with theoretical guarantees.

Table of Contents

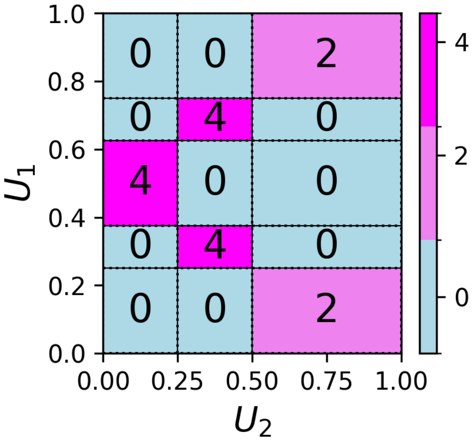
Data Example: Multivariate Discrete Data

- X_1: Treatment dose (5 levels)
- X_2: Pain severity (3 levels)

	Mild	Moderate	Severe
Very Low	0	0	$2/8$
Low	0	$1/8$	0
Medium	$2/8$	0	0
High	0	$1/8$	0
Very High	0	0	$2/8$

Checkerboard Copula Density Visualization

- Margins:
  - X_1: $\{0, 2/8, 3/8, 5/8, 6/8, 1\}$
  - X_2: $\{0, 2/8, 4/8, 1\}$
- Blockwise constant density values



Checkerboard Copula Scores (CCS)

Definition: CCS

$s^j_{i_j} = (u^j_{i_{j-1}} + u^j_{i_j})/2$ for ordinal variable X_j.

Example CCS:

- X_1: $(2/16, 5/16, 8/16, 11/16, 14/16)$
- X_2: $(2/16, 6/16, 12/16)$

Properties:

- Mean: $\mu_{S_j} = 0.5$
- Variance: $\sigma^2_{S_j} = \frac{1}{4}\sum_{i_j=1}^{I_j} u^j_{i_j-1} u^j_{i_j} p_{+i_j+}$
- Derivations can be found in [39]

CCR: Conditional Density, Regression & Prediction

$$c^+(u_j|\mathbf{u}_{-j}) = \frac{p_{i_j|\mathbf{i}_{-j}}}{p_{+i_j+}} \qquad r_{U_j|\mathbf{U}_{-j}}(\mathbf{u}_{-j}) = \sum p_{i_j|\mathbf{i}_{-j}} s^j_{i_j}$$

Example: U_2 on U_1

Regression varies blockwise over u_1 ranges. Table of $r_{U_2|U_1}$ values constructed.

Prediction process:

1. Find $\mathbf{u}^*_{-j}$ from predictor categories
2. Compute $u^*_j = r_{U_j|\mathbf{U}_{-j}}(\mathbf{u}^*_{-j})$
3. Locate predicted category via marginal CDF

Example:

$X_1 =$ Medium predicts $X_2 =$ Mild.

Visualization Methods

Cross-Tabulations:

- Color-coded predicted categories for 2-way tables.
- Overlay bootstrap proportions to assess uncertainty.
- *Contrast observed patterns vs. independence structure.*

Bubble Plots:

- Axes: combinations of predictors vs. response categories.
- Dark dot = predicted category; bubble size $\propto$ bootstrap proportion.
- *Highlights interaction effects in high dimensions.*

Doubledecker Plots:

- Hierarchical splits: vertical (predictors) and horizontal (response).
- Width = observed frequency; height = bootstrap-predicted proportion.
- *Ideal for temporal or ordered structures.*

Null Comparisons:

- *Visually assess significance of dependence.*

Bootstrap Procedure for CCR and (S)CCRAM

We apply a nonparametric bootstrap to quantify uncertainty.

Steps:

1. Resample with replacement from the original dataset to create a bootstrap sample (same size).
2. Estimate the CCR from the resampled table.
3. Predict the response category (for CCR) or compute (S)CCRAM from the resample.
4. Repeat steps 1–3 for B resamples (e.g., $B = 9999$).

Output:

- Empirical distribution of predicted categories or association scores.
- Estimate of bootstrap standard error.
- Confidence intervals derived from the distribution.

Bootstrap Confidence Interval Methods

From the bootstrap estimates $\{\hat{\theta}_b\}_{b=1}^{B}$ (e.g., (S)CCRAM):

1. Percentile Method [5]

- CI: $[\hat{\theta}_{\alpha/2}, \hat{\theta}_{1-\alpha/2}]$
- Simple and intuitive; uses bootstrap quantiles directly.

2. Basic (Reverse Percentile) Method

- CI: $[2\hat{\theta} - \hat{\theta}_{1-\alpha/2},\ 2\hat{\theta} - \hat{\theta}_{\alpha/2}]$
- Centers interval around original estimate.

3. BCa (Bias-Corrected and Accelerated) [9]

- Adjusts for bias and skewness using data-dependent corrections.
- CI: $[\hat{\theta}_{\text{BCa low}}, \hat{\theta}_{\text{BCa high}}]$
- Offers better coverage in small or skewed samples.

Comparison of Bootstrap CI Methods

Method	Properties	Notes
Percentile	No bias correction; direct quantiles from bootstrap distribution	Fast, simple; may miscover in skewed distributions
Basic (Reverse Percentile)	Assumes symmetric bootstrap distribution	Reflects bootstrap errors about original estimate
BCa	Adjusts for both bias and skew (acceleration)	Most accurate; slower; recommended in small or skewed samples

Recommendation:

- Use **BCa** unless the distribution is symmetric and unbiased.

Table of Contents

Prediction and Visualization

Prediction Pipeline: `get_predictions_ccr()` $\rightarrow$ `plot_ccr_predictions()`

Customization: Flexible variable naming, Side/x-axis legend styles, and Exportable high-res graphics.

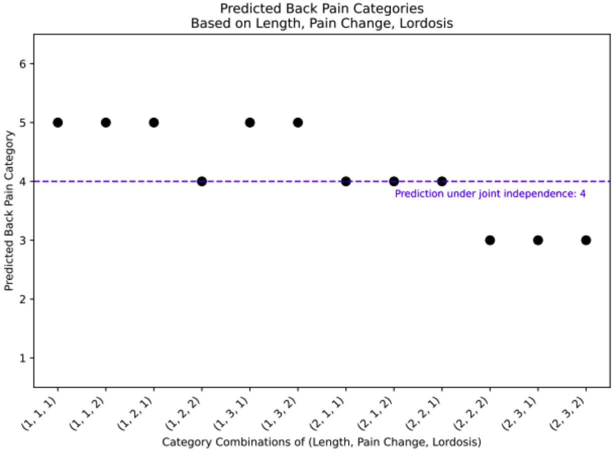


Table of Contents

A Peek into Underlying Probability Distributions

Marginal Distributions

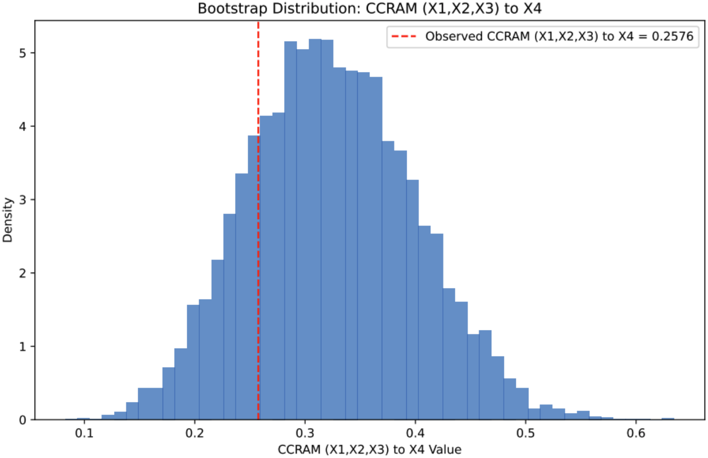
- 61.4% of patients had **long** previous attacks
- Pain change: Better (20.8%), Same (51.5%), Worse (27.7%)
- Lordosis **absent**: 63.4%
- *Outcome:* 60% reported at least **moderate improvement**

Treatment is generally effective,
but patient characteristics seemingly matter.

Bootstrap Analysis of (S)CCRAM

Uncertainty Quantification:

- CCRAM 95% BCa CI: (0.1849, 0.4762)
- SCCRAM 95% BCa CI: (0.0691, 0.3509)



Permutation Testing for Statistical Significance

Permutation Test Results:

- CCRAM p-value: 0.0016 and SCCRAM p-value: 0.0011

Conclusion: Strong evidence against H_0; observed dependence is statistically significant.