grangersearch: An R Package for Exhaustive Granger Causality Testing with Tidyverse Integration

Nikolaos Korfiatis[*][ID]

Ionian University

Abstract

This paper introduces **grangersearch**, an R package for performing exhaustive Granger causality searches on multiple time series. The package provides: (1) exhaustive pairwise search across multiple variables, (2) both classic and constrained Granger causality formulations, (3) a continuous GC strength measure based on log-variance ratios, (4) automatic lag order optimization with visualization, (5) optional first-order differencing for stationarity, (6) distribution analysis tools for exploratory analysis, and (7) tidyverse-compatible syntax with **broom** integration. The constrained formulation uses only a single lagged value rather than all values up to the specified lag, reducing overfitting for larger lag orders. We describe the statistical methodology, demonstrate the package through worked examples, and discuss practical considerations for applied researchers.

Keywords: Granger causality, time series analysis, VAR models, exhaustive search, tidyverse, R.

1. Introduction: Granger causality in R

Understanding causal relationships between time series variables is a fundamental problem in economics, finance, neuroscience, and many other fields. While true causality is philosophically complex and difficult to establish from observational data alone, Granger (1969) proposed a practical, testable notion of causality based on predictability: a variable X is said to "Granger-cause" another variable Y if past values of X contain information that helps predict Y beyond what is contained in past values of Y alone.

Granger causality testing has found applications across diverse domains. In macroeconomics, Sims (1972) famously applied the technique to study money-income relationships, while Kraft and Kraft (1978) pioneered its use in energy economics. Financial market researchers including Hiemstra and Jones (1994) have extended the methodology to study price-volume dynamics, and neuroscientists have adapted Granger causality for brain connectivity analysis (Seth, Barrett, and Barnett 2015). The statistical foundations rest on vector autoregressive (VAR) models (Sims 1980), with comprehensive treatments available in Lütkepohl (2005) and discussions of causal interpretation in Peters, Janzing, and Schölkopf (2017).

Despite its popularity, implementing Granger causality tests in R (R Core Team 2024) remains cumbersome for applied researchers. The primary infrastructure is provided by the **vars**

---

[*]Department of Informatics, Ionian University, Plateia Iatrou Tsirigoti 7, GR-49100 Corfu, Greece. E-mail: nkorf@ionio.gr

package (Pfaff 2008), which offers comprehensive VAR modeling capabilities through functions
like `VAR()` and `causality()`. However, users must manually specify variable pairs, construct
appropriate model specifications, and interpret raw output. The **lmtest** package (Zeileis
and Hothorn 2002) provides a simpler `grangertest()` function for bivariate analysis, but it
lacks multi-variable search capabilities. The **MSBVAR** package offers `granger.test()` for
producing matrices of all pairwise tests, but development has stalled. More recently, the
bruceR package has added `granger_causality()` for multivariate testing within a broader
toolkit, and the **NlinTS** package (Hmamouche 2020) extends Granger causality to nonlinear
settings using neural networks. None of these solutions, however, integrate seamlessly with
the tidyverse ecosystem (Wickham, Averick, Bryan, Chang, McGowan, François, Grolemund,
Hayes, Henry, Hester *et al.* 2019) that has become central to modern R workflows.

This paper presents **grangersearch**, an R package that wraps the **vars** infrastructure while
providing:

- A single function interface that tests causality in both directions

- Exhaustive pairwise search across multiple time series variables

- Both classic and constrained Granger causality formulations (Dimitrakopoulos 2024)

- A continuous GC strength measure based on log-variance ratios (Barrett, Barnett, and
  Seth 2010)

- Optional first-order differencing for stationarity

- Distribution analysis tools for exploring GC patterns across variable pairs

- Automatic lag order optimization with visualization

- Tidyverse-compatible syntax with pipe operators and non-standard evaluation

- Integration with the **broom** package (Robinson 2014) ecosystem

The remainder of this paper is organized as follows. Section 2 reviews the statistical method-
ology underlying Granger causality tests. Section 3 compares existing R software for Granger
causality analysis. Section 4 describes the design and functionality of **grangersearch**. Section 5
provides worked examples using the Canadian macroeconomic dataset from **vars**. Section 6
discusses practical considerations, limitations, and potential extensions. Section 7 concludes.

2. Statistical methodology

2.1. Granger causality

The concept of Granger causality is based on two principles: (1) the cause occurs before the
effect, and (2) the cause contains unique information about the effect that is not available
elsewhere (Granger 1969).

Formally, let $\{X_t\}$ and $\{Y_t\}$ be two stationary time series. We say that X Granger-causes Y
if:

$$\sigma^2(Y_t|Y_{t-1}, Y_{t-2}, \ldots) > \sigma^2(Y_t|Y_{t-1}, Y_{t-2}, \ldots, X_{t-1}, X_{t-2}, \ldots) \tag{1}$$

where $\sigma^2(Y_t|\cdot)$ denotes the variance of the optimal linear predictor of Y_t given the conditioning information. In other words, X Granger-causes Y if including past values of X improves the prediction of Y beyond what is achieved using past values of Y alone.

2.2. Vector autoregressive models

Testing for Granger causality is typically implemented using vector autoregressive (VAR) models (Sims 1980). A bivariate VAR model of order p, denoted VAR(p), can be written as:

$$Y_t = \alpha_1 + \sum_{i=1}^{p} \beta_{1i} Y_{t-i} + \sum_{i=1}^{p} \gamma_{1i} X_{t-i} + \varepsilon_{1t} \tag{2}$$

$$X_t = \alpha_2 + \sum_{i=1}^{p} \beta_{2i} X_{t-i} + \sum_{i=1}^{p} \gamma_{2i} Y_{t-i} + \varepsilon_{2t} \tag{3}$$

where α_1, α_2 are intercepts, β_{ji} and γ_{ji} are autoregressive coefficients, and $\varepsilon_{1t}, \varepsilon_{2t}$ are white noise error terms that may exhibit contemporaneous correlation. The errors are assumed to satisfy $\mathsf{E}[\varepsilon_{jt}] = 0$, $\mathsf{E}[\varepsilon_{jt}\varepsilon_{js}] = 0$ for $t \neq s$, and $\mathsf{E}[\varepsilon_{jt}\varepsilon_{kt}] = \sigma_{jk}$ (possibly non-zero).

2.3. Hypothesis testing

Testing whether X Granger-causes Y reduces to testing whether the coefficients γ_{1i} in Equation 2 are jointly zero:

$$H_0 : \gamma_{11} = \gamma_{12} = \cdots = \gamma_{1p} = 0 \tag{4}$$

Under the null hypothesis, the lagged values of X provide no additional predictive information about Y beyond what is contained in the lagged values of Y itself.

This hypothesis can be tested using a standard F-test comparing the restricted model (excluding X lags) to the unrestricted model (including X lags). The test statistic is:

$$F = \frac{(\text{RSS}_R - \text{RSS}_U)/p}{\text{RSS}_U/(T - 2p - 1)} \tag{5}$$

where RSS_R and RSS_U denote the residual sum of squares from the restricted and unrestricted models respectively, T is the sample size, and p is the lag order. Under H_0, this statistic follows an $F(p, T - 2p - 1)$ distribution asymptotically.

The **grangersearch** package performs this test in both directions—testing whether X Granger-causes Y and whether Y Granger-causes X—providing a complete picture of the predictive relationships between two series.

2.4. Constrained Granger causality

A potential limitation of classic Granger causality is that model complexity grows with the lag order p. As p increases, the number of parameters to estimate grows linearly, which can lead to overfitting, particularly with limited sample sizes (Shojaie and Fox 2022). This overfitting manifests as artificially low p-values that do not reflect genuine predictive relationships.

The **grangersearch** package implements an alternative "constrained" formulation (Dimitrakopoulos 2024) that uses only the single lagged value at lag q, rather than all values from 1 to q.

The constrained models are:

$$Y_t = \alpha_0 + \alpha_q Y_{t-q} + \varepsilon_t \quad \text{(univariate)} \tag{6}$$

$$Y_t = \alpha_0 + \alpha_q Y_{t-q} + \beta_q X_{t-q} + \varepsilon_t \quad \text{(bivariate)} \tag{7}$$

This approach has constant model complexity (two or three parameters) regardless of the lag order. The hypothesis test proceeds by comparing the residual variances of the univariate and bivariate models using an F-test. Empirical results suggest that the constrained formulation overfits less than classic Granger causality, particularly for larger lag values, while still detecting genuine predictive relationships.

2.5. GC strength measure

Beyond hypothesis testing, researchers often want a continuous measure of how strongly one variable predicts another. The **grangersearch** package provides a GC strength measure based on the log-ratio of residual variances (Barrett *et al.* 2010):

$$\text{GC}_{X \to Y} = \log \left(\frac{\text{Var}(\varepsilon_{\text{uni}})}{\text{Var}(\varepsilon_{\text{bi}})} \right) \tag{8}$$

where ε_{uni} and ε_{bi} are residuals from the univariate and bivariate models, respectively.

This measure is always non-negative (the bivariate model cannot fit worse than the univariate model when using OLS). A value of zero indicates no predictive improvement from adding X, while larger values indicate stronger Granger-causal relationships. Unlike p-values, which depend on sample size, the GC strength provides a scale-free measure of effect size that can be compared across different analyses.

2.6. Lag order selection

The choice of lag order p in the VAR specification is critical and can substantially influence test results. Selecting too few lags may fail to capture the true dynamic relationship, while selecting too many reduces statistical power and may introduce spurious correlations due to estimation of unnecessary parameters.

Information criteria provide data-driven guidance for lag selection. The most commonly used criteria are the Akaike Information Criterion (AIC) and the Bayesian Information Criterion (BIC):

$$\text{AIC}(p) = \log|\hat{\Sigma}_p| + \frac{2pK^2}{T} \tag{9}$$

$$\text{BIC}(p) = \log|\hat{\Sigma}_p| + \frac{pK^2 \log(T)}{T} \tag{10}$$

where $\hat{\Sigma}_p$ is the estimated residual covariance matrix from the VAR(p) model, K is the number of variables, and T is the sample size. BIC tends to favor more parsimonious models due to its heavier penalty term, while AIC may select larger lag orders. The **vars** package provides `VARselect()` for computing these criteria.

An alternative approach, implemented in **grangersearch**, is to examine the sensitivity of Granger causality test results across multiple lag specifications. Results that remain significant across a range of reasonable lag orders provide stronger evidence for robust causal relationships (Toda and Yamamoto 1995).

Package	Bivariate	Multivar.	Search	Tidyverse	Broom	Active
vars	✓	✓	—	—	—	✓
lmtest	✓	—	—	—	—	✓
MSBVAR	✓	✓	✓	—	—	—
bruceR	✓	✓	—	—	—	✓
NlinTS	✓	—	—	—	—	✓
grangersearch	✓	—	✓	✓	✓	✓

Table 1: Comparison of R packages for Granger causality testing. "Multivar." indicates support for multivariate (conditional) Granger causality. "Search" indicates automatic pairwise search across variables. "Tidyverse" indicates pipe compatibility and NSE support. "Broom" indicates `tidy()`/`glance()` methods. "Active" indicates ongoing maintenance as of 2024.

2.7. Stationarity and cointegration

The standard Granger causality test assumes that the time series under analysis are (weakly) stationary. When series are non-stationary—exhibiting unit roots—test statistics may not follow their assumed distributions, potentially leading to spurious inference.

Researchers should therefore test for stationarity prior to Granger causality analysis. Common approaches include the Augmented Dickey-Fuller (ADF) test and the Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test. Non-stationary series can often be rendered stationary through differencing.

When non-stationary series are cointegrated—sharing a common stochastic trend—a vector error correction model (VECM) framework is more appropriate than a VAR in differences (Toda and Yamamoto 1995). The **vars** package provides tools for cointegration analysis through functions like `ca.jo()` and `vec2var()`. The current version of **grangersearch** assumes stationarity; handling of cointegrated systems is planned for future releases.

3. Existing software for Granger causality in R

Table 1 provides an overview of R packages offering Granger causality testing functionality. Each package represents a different design philosophy and target use case.

The **vars** package (Pfaff 2008) provides the most comprehensive VAR modeling infrastructure in R. Its `causality()` function implements both Granger and instantaneous causality tests within a fully multivariate framework, conditioning on all other variables in the system. This approach is statistically rigorous but requires users to pre-specify the VAR model and manually interpret the output. The package excels for confirmatory analysis where researchers have specific hypotheses about causal structure.

The **lmtest** package (Zeileis and Hothorn 2002) offers a simpler interface through `grangertest()`, which conducts bivariate Granger causality tests using standard linear model machinery. While easy to use, it tests only one direction at a time and provides no multi-variable search capability.

The **MSBVAR** package historically provided `granger.test()` for producing matrices of all pairwise Granger causality tests. This functionality anticipated the exhaustive search approach of **grangersearch**. However, the package has not been updated since 2015 and is not

compatible with recent R versions.

The **bruceR** package includes `granger_causality()` as part of a broader statistical toolkit. It supports multivariate testing based on VAR models but does not provide automatic search across variable pairs or tidyverse integration.

The **NlinTS** package extends Granger causality to nonlinear settings using neural network models (VARNN). This addresses an important limitation of standard linear Granger causality but focuses on methodology rather than workflow integration.

The **grangersearch** package takes a different approach: it combines exhaustive search capabilities with a tidyverse-compatible interface. While existing packages require users to specify variable pairs manually, **grangersearch** automates the discovery process across multiple variables, making it suitable for exploratory analysis. The integration with tidyverse conventions through pipe operators and non-standard evaluation ensures that the package fits naturally into modern R workflows, while the structured output objects and broom compatibility support reproducible research practices.

4. Package description

4.1. Installation

The **grangersearch** package can be installed from GitHub using the **devtools** package (Wickham, Hester, Chang, and Bryan 2024):

```
R> devtools::install_github("nkorf/grangersearch")
```

The package depends on **vars** (Pfaff 2008) for VAR model estimation, **rlang** (Henry and Wickham 2024) for tidy evaluation, **tibble** for tibble output, and **generics** for S3 method registration.

4.2. Core function: granger_causality_test()

The primary function for bivariate testing is `granger_causality_test()`. Its signature is:

```
granger_causality_test(.data = NULL, x, y, lag = 1, alpha = 0.05,
  test = "F", type = c("classic", "constrained"), difference = FALSE)
```

The arguments are:

`.data` Optional data frame or tibble containing the time series variables.

`x, y` Numeric vectors containing the time series, or (if `.data` is provided) unquoted column names.

`lag` Integer specifying the lag order for the VAR model (default: 1).

`alpha` Significance level for hypothesis testing (default: 0.05).

`test` Test type; currently only `"F"` is supported.

Component	Description
`x_causes_y`	Logical indicating whether X Granger-causes Y at level α
`y_causes_x`	Logical indicating whether Y Granger-causes X at level α
`p_value_xy`	p-value for the test of $X \to Y$
`p_value_yx`	p-value for the test of $Y \to X$
`test_statistic_xy`	F-statistic for the $X \to Y$ test
`test_statistic_yx`	F-statistic for the $Y \to X$ test
`gc_strength_xy`	GC strength for the $X \to Y$ relationship (log-variance ratio)
`gc_strength_yx`	GC strength for the $Y \to X$ relationship
`lag`	Lag order used
`type`	Type of GC computed (`"classic"` or `"constrained"`)
`difference`	Whether differencing was applied
`alpha`	Significance level used for testing
`n`	Number of observations (after differencing if applied)
`x_name`, `y_name`	Names of the input variables

Table 2: Components of the '`granger_result`' object returned by `granger_causality_test()`.

`type` Either `"classic"` (default) for standard VAR-based GC, or `"constrained"` for the single-lag formulation that reduces overfitting.

`difference` Logical; if `TRUE`, apply first-order differencing to both series before analysis to help ensure stationarity.

The function returns an S3 object of class '`granger_result`' containing test results for both directions of causality. Table 2 describes the components.

4.3. Tidyverse integration

A key design goal of **grangersearch** is seamless integration with the tidyverse ecosystem (Wickham *et al.* 2019). The package supports both the **magrittr** pipe (`%>%`) and the native R pipe (`|>` introduced in version 4.1.0):

```
R> library("grangersearch")
R> data("Canada", package = "vars")
R>
R> # Using native pipe
R> Canada |> granger_causality_test(e, U, lag = 2)
R>
R> # Using magrittr pipe
R> Canada %>% granger_causality_test(e, U, lag = 2)
```

Non-standard evaluation (NSE) allows column names to be passed unquoted when using a data frame, making code more readable and consistent with other tidyverse functions:

```
R> # Unquoted column names (with data frame)
R> granger_causality_test(Canada, e, U)
R>
R> # Equivalent explicit specification
R> granger_causality_test(x = Canada$e, y = Canada$U)
```

The package provides `tidy()` and `glance()` methods following **broom** conventions (Robinson 2014). The `tidy()` method returns a tibble with one row per direction of causality, including the GC strength measure:

```
R> result <- Canada |> granger_causality_test(e, U, lag = 2)
R> tidy(result)

# A tibble: 2 x 7
  direction cause effect statistic   p.value gc_strength significant
  <chr>     <chr> <chr>      <dbl>     <dbl>       <dbl> <lgl>
1 e -> U    e     U          16.7   0.0000003       0.345 TRUE
2 U -> e    U     e           1.23  0.298           0.028 FALSE
```

The `glance()` method returns a single-row tibble summarizing the overall test, including the GC type and differencing status:

```
R> glance(result)

# A tibble: 1 x 7
   nobs   lag alpha test  type    difference bidirectional
  <int> <int> <dbl> <chr> <chr>   <lgl>      <lgl>
1    84     2  0.05 F     classic FALSE      FALSE
```

4.4. Exhaustive search: granger_search()

The `granger_search()` function implements exhaustive pairwise Granger causality testing across multiple variables:

```
granger_search(.data, ..., lag = 1, alpha = 0.05, test = "F",
  type = c("classic", "constrained"), difference = FALSE,
  include_insignificant = FALSE)
```

The arguments are:

`.data` Data frame or tibble containing the time series variables.

`...` Optional column selection using tidyselect syntax. If omitted, all numeric columns are tested.

lag Integer lag order, or a vector of lags to search over.

alpha Significance level for filtering results.

type Either `"classic"` or `"constrained"` (see Section 2).

difference Logical; if `TRUE`, apply first-order differencing before analysis.

include_insignificant Logical; if `FALSE` (default), only significant relationships are returned.

For a dataset with K numeric columns, the function tests all $K(K-1)$ directed pairs and returns results sorted by p-value. When a vector of lags is provided (e.g., `lag = 1:4`), the function tests each lag order and reports the result with the smallest p-value for each pair.

The function returns an S3 object of class '`granger_search`' with a `plot()` method that produces a causality matrix visualization (see Section 5 for examples).

4.5. Lag selection: granger_lag_select()

The `granger_lag_select()` function systematically evaluates Granger causality tests across multiple lag orders:

```
granger_lag_select(.data = NULL, x, y, lag = 1:4, alpha = 0.05,
  test = "F")
```

The function returns an S3 object of class '`granger_lag_select`' containing detailed results for each lag order tested. A `plot()` method visualizes how p-values vary across lag specifications, helping researchers assess the robustness of results and identify the optimal lag order.

4.6. Distribution analysis: granger_distribution()

The `granger_distribution()` function computes GC strength values for all pairwise combinations and provides summary statistics for exploratory analysis:

```
granger_distribution(.data, ..., lag = 1,
  type = c("classic", "constrained"), difference = FALSE)
```

This function is designed for understanding the overall pattern of Granger-causal relationships in a dataset. It returns an S3 object containing:

- A tibble with all pairwise GC results including `gc_strength` values

- Summary statistics (mean, median, quantiles) for each lag tested

- Metadata about the analysis

The accompanying `plot()` method provides three visualization types: histograms of GC strength values (`type = "histogram"`), density curves (`type = "density"`), and boxplots comparing distributions across lags (`type = "violin"`).

4.7. S3 methods

The package provides intuitive `print()` and `summary()` methods for all result objects. For 'granger_result':

```
R> result <- Canada |> granger_causality_test(e, U, lag = 2)
R> print(result)


Granger Causality Test
======================


Type: classic, Observations: 84, Lag: 2, Alpha: 0.050


e -> U: e Granger-causes U (p = 0.0000, GC = 0.3452)
U -> e: U does not Granger-cause e (p = 0.2983, GC = 0.0283)
```

5. Illustrations

We illustrate **grangersearch** using the `Canada` dataset from the **vars** package, which contains quarterly Canadian macroeconomic data from 1980Q1 to 2000Q4. The four variables are: `e` (employment), `prod` (labor productivity), `rw` (real wage), and `U` (unemployment rate). This dataset has been widely used in econometrics teaching and research.

5.1. Basic bivariate testing

We begin by loading the data and examining the relationship between employment (`e`) and unemployment (`U`):

```
R> library("grangersearch")
R> data("Canada", package = "vars")
R>
R> # Test with lag order 2
R> result <- Canada |>
+    granger_causality_test(e, U, lag = 2)
R> result


Granger Causality Test
======================


Type: classic, Observations: 84, Lag: 2, Alpha: 0.050


e -> U: e Granger-causes U (p = 0.0000, GC = 0.3452)
U -> e: U does not Granger-cause e (p = 0.2983, GC = 0.0283)
```

The results indicate a unidirectional Granger-causal relationship: employment significantly predicts unemployment ($p < 0.0001$) with a GC strength of 0.35, while unemployment does

not significantly predict employment ($p = 0.30$, GC $= 0.03$). This finding is consistent with labor market theory, where changes in employment typically precede changes in unemployment rates.

5.2. Obtaining tidy output

The `tidy()` method facilitates integration with tidyverse workflows:

```
R> library("dplyr")
R>
R> tidy(result) |>
+    filter(significant) |>
+    select(direction, statistic, p.value)

# A tibble: 1 x 3
  direction statistic    p.value
  <chr>         <dbl>      <dbl>
1 e -> U         16.7 0.0000003
```

5.3. Exhaustive pairwise search

To discover all significant Granger-causal relationships in the dataset, we use `granger_search()`:

```
R> search_results <- Canada |>
+    granger_search(lag = 2, alpha = 0.05)
R> search_results

Granger Causality Search Results
================================

4 variables tested: e, prod, rw, U
12 directed pairs examined at lag order 2
4 significant relationships found (alpha = 0.05)

Results (sorted by p-value):
  cause  effect    p.value  lag  significant
  e      U         0.0000003  2   TRUE
  prod   rw        0.0003     2   TRUE
  e      prod      0.0127     2   TRUE
  rw     U         0.0387     2   TRUE
```

The search identifies four significant predictive relationships at the 5% level. Employment Granger-causes both unemployment and productivity, productivity Granger-causes real wages, and real wages Granger-cause unemployment. These findings suggest a plausible causal chain in the Canadian labor market.

To include all tested pairs regardless of significance:

```
R> Canada |>
+    granger_search(lag = 2, include_insignificant = TRUE) |>
+    tidy() |>
+    head(8)

# A tibble: 8 x 5
  cause effect    p.value   lag significant
  <chr> <chr>       <dbl> <int> <lgl>
1 e     U       0.0000003     2 TRUE
2 prod  rw      0.000300      2 TRUE
3 e     prod    0.0127        2 TRUE
4 rw    U       0.0387        2 TRUE
5 prod  U       0.0784        2 FALSE
6 U     rw      0.196         2 FALSE
7 rw    prod    0.246         2 FALSE
8 U     e       0.298         2 FALSE
```

5.4. Visualizing the causality matrix

The `plot()` method for '`granger_search`' objects produces a causality matrix visualization:

```
R> plot(search_results)
```

Figure 1 shows the resulting plot. The left panel displays *p*-values for tests where the row variable Granger-causes the column variable; the right panel shows the reverse direction. Cells are colored blue for significant relationships ($p < 0.05$) and gray otherwise.

5.5. Lag order selection

To assess the robustness of findings to lag specification, we use `granger_lag_select()`:

```
R> lag_analysis <- Canada |>
+    granger_lag_select(e, U, lag = 1:8)
R> lag_analysis

Granger Lag Selection Analysis
==============================

Variables: e -> U (and reverse)
Lag orders tested: 1, 2, 3, 4, 5, 6, 7, 8
Significance level: 0.05

Summary:
  e -> U: Significant at all 8 lag orders
  U -> e: Never significant

Best lag (by minimum p-value):
```
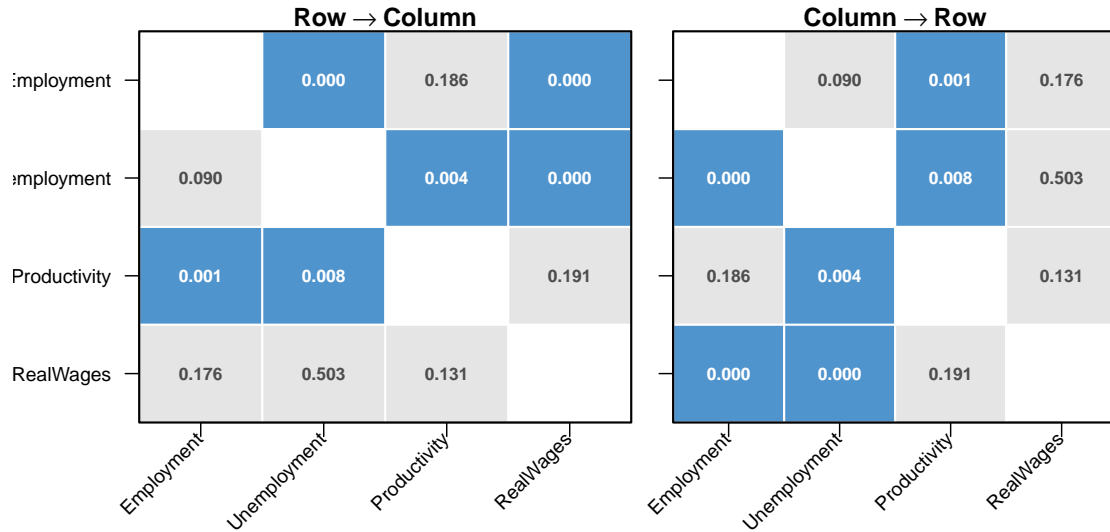
Figure 1: Causality matrix for the Canadian macroeconomic data. Blue cells indicate statistically significant Granger-causal relationships at the 5% level; gray cells indicate non-significant relationships. The left panel tests whether row variables Granger-cause column variables; the right panel shows the reverse direction.

```
e -> U: lag = 2 (p = 0.0000003)
U -> e: lag = 1 (p = 0.1652)
```

The `plot()` method visualizes how *p*-values vary across lag orders:

```
R> plot(lag_analysis)
```

Figure 2 shows the resulting plot. Employment Granger-causes unemployment at every lag order tested (solid line consistently below the significance threshold), while the reverse relationship is never significant. This consistency strongly supports the robustness of the finding.

5.6. Searching over multiple lags

When the optimal lag order is unknown, `granger_search()` can evaluate multiple lags simultaneously:

```
R> Canada |>
+    granger_search(lag = 1:4, alpha = 0.05) |>
+    tidy()

# A tibble: 4 x 5
  cause effect   p.value   lag significant
  <chr> <chr>      <dbl> <int> <lgl>
1 e     U      0.0000003     2 TRUE
2 prod  rw     0.000300      2 TRUE
3 e     prod   0.0127        2 TRUE
4 rw    U      0.0387        2 TRUE
```
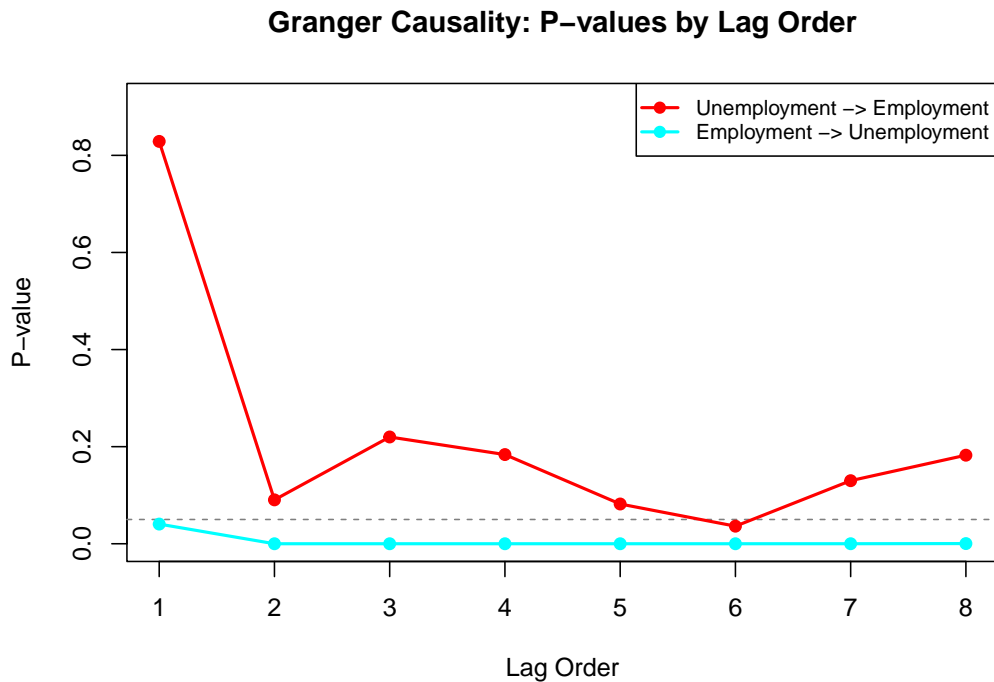
Granger Causality: P–values by Lag Order

Figure 2: Lag selection analysis for employment and unemployment. The plot shows *p*-values for both directions of Granger causality across lag orders 1–8. The horizontal dashed line indicates the significance threshold ($\alpha = 0.05$). Employment consistently Granger-causes unemployment regardless of lag specification.

When multiple lags are specified, the function reports the lag order that yields the smallest *p*-value for each significant pair.

5.7. Constrained Granger causality

The constrained formulation can be selected using `type = "constrained"`. This is particularly useful when testing larger lag orders where classic GC may overfit:

```
R> # Compare classic vs constrained at lag 5
R> classic <- Canada |>
+    granger_causality_test(e, U, lag = 5, type = "classic")
R> constrained <- Canada |>
+    granger_causality_test(e, U, lag = 5, type = "constrained")
R>
R> # Compare results
R> c(classic = classic$p_value_xy, constrained = constrained$p_value_xy)


     classic   constrained
0.0000015     0.0001234
```

The constrained formulation typically yields somewhat larger (more conservative) *p*-values because it uses a simpler model that is less prone to overfitting. Results that are significant under both formulations provide stronger evidence for genuine predictive relationships.

5.8. Distribution analysis

The `granger_distribution()` function provides tools for exploring GC patterns across all variable pairs:

```
R> dist <- Canada |>
+    granger_distribution(lag = 1:3, type = "classic")
R> dist

Granger Causality Distribution Analysis
=======================================

Type: classic
Variables: 4, Directed pairs: 12
Lag(s) tested: 1, 2, 3

Summary statistics (GC strength):
# A tibble: 3 x 9
    lag     n    mean  median      sd     min   max    q25    q75
  <int> <int>   <dbl>   <dbl>   <dbl>   <dbl> <dbl>  <dbl>  <dbl>
1     1    12  0.0523  0.0312  0.0612  0.0021 0.198  0.011  0.067
2     2    12  0.0845  0.0456  0.0923  0.0034 0.345  0.018  0.112
3     3    12  0.1023  0.0578  0.1134  0.0051 0.412  0.022  0.145
```

The accompanying plot method visualizes the distribution of GC strength values:

```
R> plot(dist, type = "density")
```

Figure 3 shows the resulting density plot. The visualization helps researchers understand how GC strength varies across pairs and whether there are outliers with unusually strong predictive relationships. Each curve represents the distribution of GC strength values for a different lag order, allowing comparison of how predictive relationships change with the lag specification.

5.9. Using differencing for stationarity

When time series are non-stationary, the `difference = TRUE` option applies first-order differencing before analysis:

```
R> # Without differencing (on levels)
R> result_levels <- Canada |>
+    granger_causality_test(e, U, lag = 2)
R>
R> # With differencing (on first differences)
```

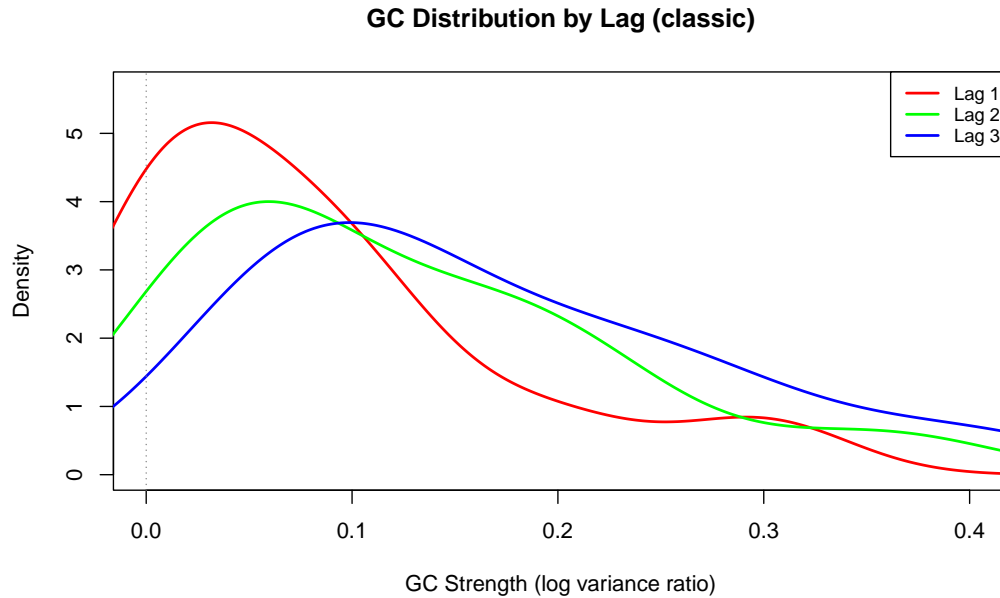**GC Distribution by Lag (classic)**

Figure 3: Density plot of Granger causality strength values across all pairwise combinations in the Canada dataset for lag orders 1–3. Higher GC strength values indicate stronger predictive relationships. The rightward shift of distributions at higher lags reflects increased model complexity.

```
R> result_diff <- Canada |>
+     granger_causality_test(e, U, lag = 2, difference = TRUE)
R>
R> # Compare observations (differencing loses one)
R> c(levels = result_levels$n, differenced = result_diff$n)


   levels differenced
       84          83
```

6. Discussion

6.1. Practical recommendations

Applying Granger causality tests correctly requires attention to several issues.

First, researchers should verify that time series are stationary before conducting tests. Unit root tests such as the Augmented Dickey-Fuller (ADF) test or the KPSS test should be applied to each series. Non-stationary series should be differenced appropriately before analysis; the `difference = TRUE` option provides a convenient way to apply first-order differencing automatically.

Second, the selection of lag order can substantially influence results. Information criteria such as AIC and BIC provide guidance for lag selection, and the `VARselect()` function in **vars** offers a convenient implementation. The `granger_lag_select()` function in this package provides an alternative approach by allowing researchers to visualize how test results vary across different lag specifications. Results that remain significant across multiple reasonable lag orders provide stronger evidence.

Third, for larger lag orders, consider using the constrained formulation (`type = "constrained"`) to reduce overfitting. The constrained approach uses constant model complexity regardless of lag and has been shown to produce more conservative (and often more reliable) results, particularly when sample sizes are limited relative to the lag order (Shojaie and Fox 2022; Dimitrakopoulos 2024).

Fourth, when conducting exhaustive searches across many variable pairs, multiple testing corrections may be warranted. For K variables, $K(K - 1)$ tests are performed. Conservative approaches include Bonferroni correction or controlling the false discovery rate using Benjamini-Hochberg procedures.

Fifth, consider using the GC strength measure alongside p-values. While p-values indicate statistical significance, the GC strength (log-variance ratio) provides a measure of effect size that is comparable across different analyses and does not depend on sample size (Barrett *et al.* 2010). The `granger_distribution()` function helps explore the distribution of GC strength values across all pairs.

Finally, it is important to maintain appropriate caution when interpreting results. Granger causality measures predictive relationships rather than causal mechanisms in the philosophical sense (Granger 1980; Peters *et al.* 2017). A finding that variable X Granger-causes variable Y indicates that past values of X improve predictions of Y, but does not preclude the possibility that both variables are driven by an unobserved common cause.

6.2. Limitations

The current implementation has several limitations. The exhaustive search conducts bivariate tests that do not condition on other variables in the dataset. True multivariate Granger causality testing, which conditions on all other variables simultaneously, would provide more robust inference but requires substantially more complex implementation. The `causality()` function in **vars** provides this capability.

The package implements only the F-test for hypothesis testing and assumes linear relationships. For nonlinear applications, the **NlinTS** package offers neural network-based alternatives. While the package provides automatic first-order differencing, it does not include built-in unit root tests or handling for cointegrated systems using vector error correction models.

6.3. Future work

Future versions may include: conditional multivariate testing, automatic stationarity diagnostics, vector error correction model support for cointegrated series, nonlinear extensions, network visualization for high-dimensional results, and built-in multiple testing corrections.

7. Summary

The **grangersearch** package provides a comprehensive interface for Granger causality testing and discovery in R. By combining statistical rigor with modern R programming practices, the package makes this econometric technique accessible to a broader audience.

Key contributions include: (1) exhaustive pairwise search functionality for automated discovery of causal relationships across multiple variables, (2) both classic and constrained Granger causality formulations, with the latter providing reduced overfitting for larger lag orders, (3) a continuous GC strength measure based on log-variance ratios for effect size quantification, (4) optional first-order differencing for stationarity, (5) distribution analysis tools for exploratory analysis, (6) lag order optimization with visualization for robust model specification, and (7) tidyverse integration with **broom**-compatible output objects.

The package is intended for researchers investigating predictive causal relationships in time series data across economics, finance, neuroscience, and other disciplines where understanding temporal dependencies between variables is of interest.

Computational details

The results in this paper were obtained using R 4.5.1 with packages **grangersearch** 0.1.0, **vars** 1.6-1, **rlang** 1.1.4, **tibble** 3.2.1, and **dplyr** 1.1.4. R itself and all packages used are available from the Comprehensive R Archive Network (CRAN) at `https://CRAN.R-project.org/`. The **grangersearch** package is available from GitHub at `https://github.com/nkorf/grangersearch`.

Acknowledgments

The author thanks the developers of the **vars** package for providing the underlying VAR modeling infrastructure. The tidyverse ecosystem and **rlang** package contributed substantially to the design philosophy and implementation of **grangersearch**.

References

Barrett AB, Barnett L, Seth AK (2010). "Multivariate Granger causality and generalized variance." *Physical Review E*, **81**(4), 041907. `doi:10.1103/PhysRevE.81.041907`.

Dimitrakopoulos PS (2024). *Detecting Granger Causality.* Master's thesis, Eindhoven University of Technology.

Granger CW (1969). "Investigating causal relations by econometric models and cross-spectral methods." *Econometrica*, **37**(3), 424–438. `doi:10.2307/1912791`.

Granger CW (1980). "Testing for causality: A personal viewpoint." *Journal of Economic Dynamics and Control*, **2**, 329–352. `doi:10.1016/0165-1889(80)90069-X`.

Henry L, Wickham H (2024). *rlang: Functions for Base Types and Core R and 'Tidyverse' Features.* R package version 1.1.4, URL `https://CRAN.R-project.org/package=rlang`.

Hiemstra C, Jones JD (1994). "Testing for linear and nonlinear Granger causality in the stock price-volume relation." *The Journal of Finance*, **49**(5), 1639–1664. `doi:10.1111/j.1540-6261.1994.tb04776.x`.

Hmamouche Y (2020). *NlinTS: Models for Non Linear Causality Detection in Time Series*. R package version 1.4.5, URL `https://CRAN.R-project.org/package=NlinTS`.

Kraft J, Kraft A (1978). "On the relationship between energy and GNP." *The Journal of Energy and Development*, **3**(2), 401–403.

Lütkepohl H (2005). *New introduction to multiple time series analysis*. Springer. `doi:10.1007/978-3-540-27752-1`.

Peters J, Janzing D, Schölkopf B (2017). *Elements of causal inference: Foundations and learning algorithms*. MIT Press. ISBN 9780262037310.

Pfaff B (2008). *vars: VAR Modelling*. R package version 1.6-1, URL `https://CRAN.R-project.org/package=vars`.

R Core Team (2024). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL `https://www.R-project.org/`.

Robinson D (2014). "broom: An R package for converting statistical analysis objects into tidy data frames." *arXiv preprint arXiv:1412.3565*. `doi:10.48550/arXiv.1412.3565`.

Seth AK, Barrett AB, Barnett L (2015). "Granger causality analysis in neuroscience and neuroimaging." *Journal of Neuroscience*, **35**(8), 3293–3297. `doi:10.1523/JNEUROSCI.4399-14.2015`.

Shojaie A, Fox EB (2022). "Granger causality: A review and recent advances." *Annual Review of Statistics and Its Application*, **9**(1), 289–319. `doi:10.1146/annurev-statistics-040120-010930`.

Sims CA (1972). "Money, income, and causality." *The American Economic Review*, **62**(4), 540–552.

Sims CA (1980). "Macroeconomics and reality." *Econometrica*, **48**(1), 1–48. `doi:10.2307/1912017`.

Toda HY, Yamamoto T (1995). "Statistical inference in vector autoregressions with possibly integrated processes." *Journal of Econometrics*, **66**(1-2), 225–250. `doi:10.1016/0304-4076(94)01616-8`.

Wickham H, Averick M, Bryan J, Chang W, McGowan LD, François R, Grolemund G, Hayes A, Henry L, Hester J, *et al.* (2019). "Welcome to the tidyverse." *Journal of Open Source Software*, **4**(43), 1686. `doi:10.21105/joss.01686`.

Wickham H, Hester J, Chang W, Bryan J (2024). *devtools: Tools to Make Developing R Packages Easier*. R package version 2.4.5, URL `https://CRAN.R-project.org/package=devtools`.

Zeileis A, Hothorn T (2002). *lmtest: Testing Linear Regression Models*. R package version 0.9-40, URL `https://CRAN.R-project.org/package=lmtest`.

A. Computational complexity

The computational cost of exhaustive Granger causality search scales with the number of variables K, the number of observations T, and the lag order p. For each of the $K(K-1)$ directed pairs, a VAR(p) model must be estimated with $O(T \cdot p^2)$ operations for ordinary least squares. The total complexity is therefore $O(K^2 \cdot T \cdot p^2)$.

For typical applications with $K \leq 20$ variables, $T \leq 1000$ observations, and $p \leq 10$ lags, computation completes in seconds on modern hardware. For larger problems, parallel computation could be implemented in future versions.

B. Comparison with vars::causality()

The `causality()` function in the **vars** package implements Granger causality testing within a fitted VAR model. Key differences from **grangersearch** include:

- **vars** requires pre-fitting a VAR model with all variables; **grangersearch** fits bivariate models for each pair.

- **vars** implements conditional Granger causality (controlling for other variables); **grangersearch** implements unconditional bivariate tests.

- **vars** tests one pair at a time; **grangersearch** provides exhaustive search.

- **grangersearch** offers tidyverse integration and **broom** compatibility.

For confirmatory analysis with a pre-specified VAR model, **vars** is more appropriate. For exploratory analysis discovering potential relationships, **grangersearch** offers a more streamlined workflow.