Package: phoring (via r-universe)

November 8, 2026

Type Package

Title Unified Econometric Estimation, Specification Testing, and Diagnostics

Version 0.1.0

Description A unified econometric toolkit providing automated specification testing, robust covariance estimation, and publication-grade academic visualizations. Implements ordinary and weighted least squares with robust, cluster-robust, and heteroskedasticity and autocorrelation consistent (HAC) standard errors; instrumental variables two-stage least squares (2SLS); panel data estimators (within fixed effects, two-way fixed effects, random effects GLS, first differences); binary response models (logit and probit with average marginal effects); tobit censored regression; count models (Poisson and negative binomial); causal inference methods (difference-in-differences, sharp regression discontinuity); quantile regression; dynamic time series models (autoregressive distributed lag, Koyck distributed lag, Prais-Winsten AR(1), error correction models, vector autoregression); and time series unit root batteries (augmented Dickey-Fuller, Phillips-Perron, KPSS, Engle-Granger cointegration, Granger causality, ARCH). Estimation routines integrate comprehensive diagnostic batteries, parameter stability tests, and academic visualizations with zero external package dependencies.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Depends R (>= 3.5.0)

Imports graphics, grDevices, stats

Suggests testthat (>= 3.0.0)

URL https://github.com/abidsiddique/phoring

BugReports https://github.com/abidsiddique/phoring/issues

Config/roxygen2/version 8.0.0

Repository https://abidsiddique.r-universe.dev

Date/Publication 2026-11-07 17:39:02 UTC

RemoteUrl https://github.com/abidsiddique/phoring

RemoteRef HEAD

RemoteSha f70278bac877237818048eae17d82e83135e4e8d

Contents

arch_test . . . 3
ardl . . . 4
chow_test . . . 5
coint . . . 5
college_distance . . . 6
corr . . . 7
count_reg . . . 7
crime_panel . . . 8
desc . . . 9
diagnostics . . . 9
did . . . 10
durbin_h . . . 11
dynamics . . . 12
ecm . . . 12
econ_plot . . . 13
firm_investment . . . 14
gq_test . . . 15
granger . . . 15
house_prices . . . 16
interpret . . . 17
iv . . . 17
j_test . . . 18
koyck . . . 19
labor_supply . . . 20
logit . . . 20
macro_data . . . 22
min_wage . . . 22
outliers . . . 23
panel . . . 24
plot.econ_arch . . . 25
plot.econ_ardl . . . 26
plot.econ_chow . . . 27
plot.econ_coint . . . 27
plot.econ_corr . . . 28
plot.econ_count . . . 29
plot.econ_desc . . . 29
plot.econ_desc_balance . . . 30

plot.econ_diagnostics . . . 31
plot.econ_did . . . 31
plot.econ_durbin_h . . . 32
plot.econ_ecm . . . 33
plot.econ_gq . . . 33
plot.econ_granger . . . 34
plot.econ_hyp . . . 35
plot.econ_iv . . . 35
plot.econ_jtest . . . 36
plot.econ_koyck . . . 37
plot.econ_logit . . . 37
plot.econ_outliers . . . 38
plot.econ_panel . . . 39
plot.econ_prais . . . 40
plot.econ_qreg . . . 41
plot.econ_rdd . . . 41
plot.econ_tobit . . . 42
plot.econ_unit_root . . . 43
plot.econ_var . . . 44
prais_winsten . . . 45
qreg . . . 46
rdd . . . 47
reg . . . 48
test_hyp . . . 49
timeseries . . . 49
tobit . . . 50
unit_root . . . 50
var_model . . . 51
wage_data . . . 52
word_table . . . 53

Index **54**

`arch_test` *Autoregressive Conditional Heteroskedasticity (ARCH) Test*

Description

Engle's Lagrange Multiplier test for ARCH effects / volatility clustering in time series models.

Usage

```
arch_test(model, lags = 2, digits = 4)
```

Arguments

`model`	A fitted model object from `reg` or `lm`.
`lags`	Number of squared residual lags to test (default is 2).
`digits`	Number of decimals to display (default is 4).

Value

An object of class `econ_arch`.

Examples

```
data(macro_data, package = "phoring")
m <- reg(inflation ~ fed_funds, data = macro_data, diagnostics = FALSE)
arch_test(m, lags = 2)
```

`ardl` *Autoregressive Distributed Lag (ARDL) Model*

Description

Estimates an ARDL(p, q) model and calculates both short-run impact multipliers and long-run cumulative propensity multipliers with delta-method standard errors.

Usage

```
ardl(formula, data, p = 1, q = 1, digits = 4)
```

Arguments

`formula`	Model formula y ~ x.
`data`	A data.frame containing the time series.
`p`	Number of autoregressive lags of y (default is 1).
`q`	Number of distributed lags of x (default is 1).
`digits`	Number of decimals to display (default is 4).

Value

An object of class `econ_ardl`.

Examples

```
data(macro_data, package = "phoring")
ardl(inflation ~ unemployment, data = macro_data, p = 1, q = 1)
```

chow_test *Chow Test for Parameter Stability and Structural Breaks*

Description

Tests whether regression coefficients are constant across two subsamples (e.g. before/after a policy shock, male vs. female, or high vs. low income).

Usage

```
chow_test(formula, split_by, data = NULL, digits = 4)
```

Arguments

formula	Regression formula (or an existing econ_reg or lm model).
split_by	A binary variable or condition in data defining the two subsamples.
data	A data.frame containing the variables (required if formula is used).
digits	Number of decimal places to format (default is 4).

Value

An object of class econ_chow.

Examples

```
data(wage_data, package = "phoring")
# Test if wage equation differs by gender
chow_test(wage ~ educ + exper, split_by = "female", data = wage_data)
```

coint *Engle-Granger Cointegration Test*

Description

Tests whether two or more non-stationary time series share a stationary long-run equilibrium cointegrating relationship.

Usage

```
coint(formula, data, lags = 1, digits = 4)
```

Arguments

formula	Model formula specifying cointegrating regression (e.g., y ~ x or log(y) ~ log(x)).
data	A data.frame containing the time series variables.
lags	Number of lags for residual ADF test (default is 1).
digits	Number of decimal places to display (default is 4).

Value

An object of class econ_coint.

Examples

```
data(macro_data, package = "phoring")
coint(inflation ~ unemployment + fed_funds, data = macro_data)
```

---

college_distance *Returns to Schooling and Proximity to College Dataset*

Description

A dataset of 800 individuals designed for Instrumental Variables (2SLS) estimation, based on David Card's (1995) landmark study using geographic proximity to a 4-year college as an exogenous instrument for educational attainment.

Usage

```
college_distance
```

Format

A data frame with 800 rows and 9 variables:

wage Hourly wage in US dollars

educ Years of completed education (endogenous regressor)

exper Years of potential labor market experience

tenure Years with current employer

near_college Indicator: 1 if grew up near a 4-year college (excluded instrument)

mother_educ Mother's years of education (excluded instrument)

father_educ Father's years of education (excluded instrument)

south Indicator: 1 if residing in the South

black Indicator: 1 if Black

Source

Modeled after Card (1995), "Using Geographic Variation in College Proximity".

corr *Correlation Matrix with Significance Stars*

Description

Computes pairwise Pearson correlation coefficients with significance stars.

Usage

```
corr(data, vars = NULL, digits = 3)
```

Arguments

data	A data.frame.
vars	Optional character vector of column names.
digits	Number of decimal places (default is 3).

Value

An object of class `econ_corr`.

Examples

```
data(mtcars)
corr(mtcars, vars = c("mpg", "hp", "wt", "qsec"))
```

count_reg *Count Data Models (Poisson and Negative Binomial Regression)*

Description

Fits Poisson or Negative Binomial regression for count data, computes Incidence Rate Ratios (IRR) with 95% confidence intervals, robust standard errors, and conducts Cameron & Trivedi's overdispersion test or Likelihood Ratio test against Poisson.

Usage

```
count_reg(formula, data, model = c("poisson", "negbin"), digits = 4)
```

Arguments

formula	Model formula (e.g., `docvis ~ educ + age + female`).
data	A data.frame containing the variables.
model	Model type: `"poisson"` (default) or `"negbin"`.
digits	Number of decimals to display (default is 4).

Value

An object of class `econ_count`.

Examples

```
set.seed(42)
n <- 300
x1 <- rnorm(n); x2 <- rbinom(n, 1, 0.5)
y <- rpois(n, lambda = exp(0.5 + 0.3 * x1 - 0.4 * x2))
df <- data.frame(y, x1, x2)
count_reg(y ~ x1 + x2, data = df, model = "poisson")
count_reg(y ~ x1 + x2, data = df, model = "negbin")
```

---

crime_panel *County-Level Crime Rates Panel Dataset*

Description

A balanced panel dataset of 50 counties observed over 4 years (2020-2023), totaling 200 observations. Useful for testing the economic model of crime with Fixed Effects, Random Effects, and entity-clustered standard errors.

Usage

```
crime_panel
```

Format

A data frame with 200 rows and 6 variables:

county_id County identifier (1 to 50)

year Year of observation (2020 to 2023)

crime_rate Reported crimes per 1,000 residents

prbarr Probability of arrest (ratio of arrests to reported offenses)

police Police officers per 1,000 residents

unemp County unemployment rate (percentage)

Source

Modeled after Cornwell and Trumbull (1994) panel crime study.

desc *Publication-Ready Summary Statistics and Covariate Balance for Economics*

Description

Generates standard summary statistics (Obs, Mean, Std Dev, Min, P25, Median, P75, Max, Missing) for numeric variables in a dataset, or computes a full "Table 1" covariate balance test across two treatment/subsample groups with normalized differences and t-tests.

Usage

```
desc(data, vars = NULL, by = NULL, digits = 3)
```

Arguments

data	A data.frame.
vars	Optional character vector of column names to summarize.
by	Optional character string specifying a binary grouping column (e.g. treatment, gender) for two-group covariate balance testing ("Table 1").
digits	Number of decimal places to display (default is 3).

Value

A data.frame of class econ_desc or econ_desc_balance.

Examples

```
data(wage_data, package = "phoring")
# Standard descriptive statistics
desc(wage_data, vars = c("wage", "educ", "exper"))

# Covariate balance table ("Table 1") by gender
desc(wage_data, vars = c("wage", "educ", "exper"), by = "female")
```

diagnostics *Econometric Diagnostic Tests*

Description

Computes a complete battery of econometric diagnostic tests for linear models, including Breusch-Pagan, White, Durbin-Watson, Breusch-Godfrey, Ramsey RESET, Multicollinearity (VIF), and Jarque-Bera normality tests.

Usage

```
diagnostics(model, ...)
```

Arguments

model	A fitted model object from ‘reg()‘ or ‘stats::lm()‘.
...	Additional arguments.

Value

An object of class ‘econ_diagnostics‘ containing test statistics, p-values, decisions, and individual VIF values.

Examples

```
data(mtcars)
m <- reg(mpg ~ wt + hp + qsec, data = mtcars)
diagnostics(m)
```

---

did *Difference-in-Differences (DiD) Policy Evaluation*

Description

Fits a Difference-in-Differences (DiD) model with optional control covariates, computes cluster-robust standard errors, and outputs the classic 2x2 pre/post treatment table alongside full regression estimates.

Usage

```
did(
  formula,
  data,
  treat = NULL,
  post = NULL,
  time = NULL,
  cluster = NULL,
  digits = 4,
  ...
)
```

Arguments

formula	Model formula specifying outcome, treatment, and post periods, optionally with control covariates (e.g., y ~ treat * post or y ~ treat * post + x1 + x2).
data	A data.frame containing the variables.

treat	Column name for treatment indicator (optional, automatically inferred from interaction).
post	Column name for post-period indicator (optional, automatically inferred from interaction).
time	Optional name of time period variable (e.g. "year") for multi-period trajectories and event-study plotting.
cluster	Optional column name for cluster-robust standard errors (recommended: state/unit level).
digits	Number of decimals to display (default is 4).
...	Additional arguments passed to reg.

Value

An object of class econ_did containing the DiD effect, 2x2 table, and estimates.

Examples

```
set.seed(42)
n <- 400
state <- sample(1:20, n, replace = TRUE)
treat <- ifelse(state <= 10, 1, 0)
post <- sample(0:1, n, replace = TRUE)
x1 <- rnorm(n)
y <- 10 + 2 * treat + 1.5 * post + 3.2 * (treat * post) + 0.5 * x1 + rnorm(n, sd = 2)
df <- data.frame(y, treat, post, state, x1)
did(y ~ treat * post + x1, data = df, cluster = "state")
```

---

durbin_h *Durbin's h Test for Autocorrelation in Autoregressive Models*

Description

Tests for first-order autocorrelation in dynamic models containing a lagged dependent variable, where the standard Durbin-Watson d test is biased toward 2 (Durbin, 1970).

Usage

```
durbin_h(model, lag_var = NULL, digits = 4)
```

Arguments

model	A fitted econ_reg or lm model object containing a lagged dependent variable.
lag_var	Optional character name of the lagged dependent variable. If NULL, auto-detected by looking for terms containing "lag" or matching the dependent variable name.
digits	Number of decimal places to format (default is 4).

Value

An object of class `econ_durbin_h`.

Examples

```
data(macro_data, package = "phoring")
n <- nrow(macro_data)
macro_data$inf_lag <- c(NA, macro_data$inflation[-n])
m <- reg(inflation ~ inf_lag + unemployment, data = macro_data[-1, ])
durbin_h(m, lag_var = "inf_lag")
```

`dynamics` *Dynamic Econometric Models: Koyck, Durbin's h, Prais-Winsten, ECM, and VAR*

Description

Implements dynamic time series models, autoregressive distributed lags, serial correlation corrections, error correction models, and vector autoregressions.

`ecm` *Error Correction Model (ECM)*

Description

Estimates an Error Correction Model based on the Granger Representation Theorem (Engle and Granger, 1987), capturing both short-run dynamic responses and the speed of adjustment towards long-run cointegrating equilibrium.

Usage

```
ecm(formula, data, lags = 1, digits = 4)
```

Arguments

`formula`	Model formula (e.g., `y ~ x`).
`data`	A data.frame containing the cointegrated time series.
`lags`	Number of short-run difference lags to include (default is 1).
`digits`	Number of decimal places to format (default is 4).

Value

An object of class `econ_ecm`.

Examples

```
data(macro_data, package = "phoring")
ecm(inflation ~ unemployment + fed_funds, data = macro_data, lags = 1)
```

econ_plot *Academic Publication-Quality Econometric Diagnostic Plots*

Description

Generates clean, publication-grade academic diagnostic plots with serif typography, L-frame axes, and standardized econometric styling across all models in the package.

Usage

```
econ_plot(
  x,
  which = c("all", "resid_fitted", "qq", "scale_location", "cooks", "leverage",
    "added_var", "component_resid", "dfbetas", "forest", "vif", "histogram",
   "observed_fitted", "cusum", "cusum_sq", "cusum_squares", "stability", "influence",
    "influence_plot"),
  ...
)

## S3 method for class 'econ_reg'
plot(
  x,
  y = NULL,
  which = c("all", "resid_fitted", "qq", "scale_location", "cooks", "leverage",
    "added_var", "component_resid", "dfbetas", "forest", "vif", "histogram",
   "observed_fitted", "cusum", "cusum_sq", "cusum_squares", "stability", "influence",
    "influence_plot"),
  ...
)
```

Arguments

x	An object of class econ_reg (from reg) or standard lm.
which	Character string or vector specifying which plot(s) to draw: "all" (default 2x2 core grid), "resid_fitted", "qq", "scale_location", "cooks", "leverage", "added_var", "component_resid", "dfbetas", "forest", "vif", "histogram", "observed_fitted", "cusum", "cusum_sq", "stability", or "influence".
...	Additional graphical parameters.
y	Unused, for compatibility with generic.

Value

Invisibly returns x.

Examples

```
data(wage_data, package = "phoring")
mod <- reg(wage ~ educ + exper, data = wage_data)
econ_plot(mod)
plot(mod, which = "leverage")
plot(mod, which = "cusum_sq")
```

`firm_investment` *Corporate Investment Panel Dataset (Grunfeld Style)*

Description

A balanced panel dataset of 40 manufacturing firms observed annually over 5 years (2019-2023), totaling 200 observations. Modeled after the classic Grunfeld investment panel. Ideal for Fixed Effects (Within), Random Effects (GLS), and the Hausman specification test.

Usage

```
firm_investment
```

Format

A data frame with 200 rows and 5 variables:

firm_id Firm identifier (1 to 40)

year Year of observation (2019 to 2023)

inv Gross corporate capital investment

value Market value of the firm

capital Real value of plant and equipment capital stock

Source

Modeled after Grunfeld (1958) corporate investment panel.

gq_test *Goldfeld-Quandt Test for Heteroskedasticity*

Description

Tests for heteroskedasticity by ordering the sample according to a regressor, omitting central observations, and testing whether the error variance differs significantly between the lower and upper subsamples (Goldfeld and Quandt, 1965).

Usage

```
gq_test(formula, split_by = NULL, data = NULL, omit_fraction = 0.2, digits = 4)
```

Arguments

formula	Model formula (e.g., wage ~ educ + exper) or a fitted model object.
split_by	Character name of the explanatory variable to sort by. If NULL, uses the first continuous predictor.
data	A data.frame containing the variables (required if formula is used).
omit_fraction	Fraction of central observations to omit (default is 0.20, roughly 1/5th).
digits	Number of decimal places to format (default is 4).

Value

An object of class econ_gq.

Examples

```
data(wage_data, package = "phoring")
gq_test(wage ~ educ + exper, split_by = "educ", data = wage_data)
```

granger *Bidirectional Granger Causality Test*

Description

Tests whether past values of X forecast Y, and whether past values of Y forecast X.

Usage

```
granger(formula, data, lags = 2, digits = 4)
```

Arguments

formula	Model formula specifying bivariate relationship y ~ x.
data	A data.frame containing the time series.
lags	Number of autoregressive lags to include (default is 2).
digits	Number of decimals to display (default is 4).

Value

An object of class econ_granger.

Examples

```
data(macro_data, package = "phoring")
granger(inflation ~ fed_funds, data = macro_data, lags = 2)
```

---

house_prices *Hedonic Housing Prices and Urban Environmental Quality*

Description

A cross-sectional dataset of 506 housing tracts modeled after the classic Boston housing study (Harrison & Rubinfeld, 1978). Ideal for multiple regression, heteroskedasticity testing (Breusch-Pagan, White), and Ramsey RESET functional form tests.

Usage

```
house_prices
```

Format

A data frame with 506 rows and 7 variables:

price Median value of owner-occupied homes in $1,000s

rooms Average number of rooms per dwelling

crime Per capita crime rate by town

nox Nitric oxides concentration (parts per 10 million)

dist Weighted distances to five employment centres

tax Full-value property-tax rate per $10,000

ptratio Pupil-teacher ratio by town school district

Source

Modeled after Harrison and Rubinfeld (1978) hedonic housing study.

interpret *Plain-English Econometric Model Interpretation*

Description

Automatically inspects regression functional forms (log-log, log-level, level-log, level-level) and provides human-readable, plain-English economic interpretations for every coefficient, including elasticities, semi-elasticities, percentage effects, and quadratic turning points.

Usage

```
interpret(model, digits = 4)
```

Arguments

model	An econ_reg object produced by reg or a standard lm model.
digits	Number of decimal places to format (default is 4).

Value

An object of class econ_interpret containing interpretation summaries.

Examples

```
data(wage_data, package = "phoring")
# Quadratic wage equation
mod <- reg(wage ~ educ + exper + I(exper^2), data = wage_data)
interpret(mod)

# Log-log elasticity model
data(house_prices, package = "phoring")
mod_log <- reg(log(price) ~ log(rooms) + crime, data = house_prices)
interpret(mod_log)
```

iv *Instrumental Variables (2SLS) Regression with Automated Diagnostics*

Description

Fits a Two-Stage Least Squares (2SLS) instrumental variables regression model, calculates robust or clustered standard errors, and automatically runs diagnostic tests (Weak Instruments First-Stage F, Wu-Hausman Endogeneity, and Sargan Overidentifying Restrictions).

Usage

```
iv(
  formula,
  data,
  se = c("robust", "standard", "cluster"),
  cluster = NULL,
  digits = 4,
  ...
)
```

Arguments

formula	Two-part formula separated by |. Supports standard forms: y ~ exog + endo | exog + iv or y ~ exog | endo ~ iv. Example: wage ~ exper + educ | exper + mother_educ + father_educ.
data	A data.frame containing the variables in the model.
se	Type of standard errors to compute: "robust" (White HC1, default), "standard" (classical 2SLS), or "cluster" (cluster-robust).
cluster	Vector or column name indicating cluster grouping.
digits	Number of decimal places to display in output tables (default is 4).
...	Additional arguments.

Value

An object of class econ_iv containing 2SLS estimates, diagnostics, and standard errors.

Examples

```
data(wage_data, package = "phoring")
iv(wage ~ exper + educ | exper + mother_educ, data = wage_data)
iv(wage ~ exper | educ ~ mother_educ + father_educ, data = wage_data)
```

---

j_test *Davidson-MacKinnon J-Test for Non-Nested Models*

Description

Tests competing non-nested econometric models (neither model is a subset of the other), following the specification testing framework of Davidson and MacKinnon (1981).

Usage

```
j_test(model1, model2, data = NULL, digits = 4)
```

Arguments

model1	Fitted model object or formula for Model 1.
model2	Fitted model object or formula for Model 2.
data	A data.frame containing variables (required if formulas are passed).
digits	Number of decimal places to format (default is 4).

Value

An object of class econ_jtest.

Examples

```
data(wage_data, package = "phoring")
# Model 1: Education & Experience vs. Model 2: Tenure & Mother's Education
m1 <- reg(wage ~ educ + exper, data = wage_data)
m2 <- reg(wage ~ tenure + mother_educ, data = wage_data)
j_test(m1, m2)
```

---

koyck *Koyck Distributed Lag Model*

Description

Estimates geometric distributed lag models using the Koyck transformation (Koyck, 1954), computing the Short-Run Multiplier, Long-Run Multiplier, Rate of Decay, Median Lag, and Mean Lag.

Usage

```
koyck(formula, data, method = c("ols", "iv"), digits = 4)
```

Arguments

formula	Model formula with one explanatory variable (e.g., y ~ x).
data	A data.frame containing the time series.
method	Estimation method: "ols" (default) or "iv" (Liviatan IV using lagged X as instrument).
digits	Number of decimal places to format (default is 4).

Value

An object of class econ_koyck.

Examples

```
data(macro_data, package = "phoring")
koyck(inflation ~ unemployment, data = macro_data, method = "ols")
koyck(inflation ~ unemployment, data = macro_data, method = "iv")
```

labor_supply *Female Labor Force Participation Dataset (Mroz Style)*

Description

A microeconometric cross-section of 753 married women modeled after Mroz (1987). Ideal for binary choice estimation via Logit, Probit, and computing Average Marginal Effects (AME).

Usage

```
labor_supply
```

Format

A data frame with 753 rows and 7 variables:

inlf Binary indicator: 1 if woman worked in the labor force, 0 otherwise

nwifeinc Non-wife family household income (in $1,000s)

educ Years of schooling

exper Years of past work experience

age Age in years

kidslt6 Number of young children under 6 years old

kidsge6 Number of older children between 6 and 18 years old

hours Annual hours worked (0 for non-working women, ideal for Tobit)

docvis Annual doctor visits (integer count, ideal for Poisson regression)

Source

Modeled after Mroz (1987), "The Sensitivity of an Empirical Model of Married Women's Hours of Work".

logit *Binary Choice Models (Logit and Probit) with Marginal Effects*

Description

Fits logistic regression (Logit) or probit regression (Probit) models, computes robust Huber-White or clustered standard errors, reports odds ratios (for logit), ROC-AUC, classification accuracy, and calculates Average Marginal Effects (AME) with delta-method standard errors.

Usage

```
logit(
  formula,
  data,
  se = c("robust", "standard", "cluster"),
  cluster = NULL,
  digits = 4,
  ...
)

probit(
  formula,
  data,
  se = c("robust", "standard", "cluster"),
  cluster = NULL,
  digits = 4,
  ...
)
```

Arguments

`formula`	Model formula (e.g., `admit ~ gre + gpa + rank`).
`data`	A data.frame containing the variables.
`se`	Type of standard errors: `"robust"` (Huber-White sandwich, default), `"standard"` (classical MLE), or `"cluster"` (cluster-robust).
`cluster`	Vector or column name for cluster-robust standard errors.
`digits`	Number of decimal places to display (default is 4).
`...`	Additional arguments passed to `glm`.

Value

An object of class `econ_logit` containing index estimates, odds ratios, average marginal effects (AME), and model diagnostics.

Examples

```
data(mtcars)
logit(am ~ mpg + hp, data = mtcars)
probit(am ~ mpg + hp, data = mtcars)
```

macro_data *Quarterly Macroeconomic Time Series (Phillips Curve)*

Description

A quarterly macroeconomic time series dataset spanning 120 quarters (1994Q1 to 2023Q4). Designed for time series regressions, serial correlation testing (Durbin-Watson, Breusch-Godfrey), and Newey-West HAC standard errors.

Usage

```
macro_data
```

Format

A data frame with 120 rows and 5 variables:

quarter Calendar quarter (e.g., "1994Q1" to "2023Q4")

inflation Annualized CPI inflation rate (percentage)

unemployment Civilian unemployment rate (percentage)

fed_funds Federal funds effective interest rate (percentage)

gdp_growth Annualized real GDP growth rate (percentage)

Source

Simulated based on US macroeconomic time series dynamics.

min_wage *State Minimum Wage Policy Evaluation (Card & Krueger DiD)*

Description

A Difference-in-Differences evaluation dataset of 400 fast-food restaurants in New Jersey (treated state) and eastern Pennsylvania (control state) observed before and after New Jersey's minimum wage increase.

Usage

```
min_wage
```

Format

A data frame with 400 rows and 5 variables:

store_id Store identifier (1 to 400)

chain Fast-food chain name (Burger King, KFC, Roy Rogers, Wendys)

treated State indicator: 1 if New Jersey (treated), 0 if Pennsylvania (control)

post Time period: 1 if after the wage hike, 0 if before

fte Full-time equivalent employment at the restaurant

Source

Modeled after Card and Krueger (1994) minimum wage study.

---

outliers *Detect Outliers and Influential Observations in Regression Models*

Description

Automatically computes Cook's Distance, studentized residuals, and leverage (hat values) for regression models, flagging influential data points that could distort econometric estimates.

Usage

```
outliers(model, top_n = 5, digits = 4)
```

Arguments

`model`	An `econ_reg` or `lm` model object.
`top_n`	Number of top influential observations to display (default is 5).
`digits`	Number of decimal places to format (default is 4).

Value

An object of class `econ_outliers`.

Examples

```
data(wage_data, package = "phoring")
mod <- reg(wage ~ educ + exper, data = wage_data)
outliers(mod)
```

`panel` *Comprehensive Panel Data Econometrics with Two-Way FE and Specification Tests*

Description

Fits panel data models (Fixed Effects, Two-Way Fixed Effects, Random Effects, First Differences, Between Estimator, or Pooled OLS) with customizable standard errors (clustered by entity id, White robust, or classical standard) and automatically executes the Hausman test, Breusch-Pagan LM test, and Wooldridge panel autocorrelation test.

Usage

```
panel(
  formula,
  data,
  id,
  time = NULL,
  model = c("fe", "twfe", "re", "fd", "between", "pooling"),
  se = c("cluster", "robust", "standard"),
  digits = 4,
  ...
)
```

Arguments

`formula`	Model formula (e.g., `inv ~ value + capital` or `log(inv) ~ value + I(capital^2)`).
`data`	A data.frame containing the panel data.
`id`	Column name (or unquoted symbol) indicating the entity/group identifier.
`time`	Column name (or unquoted symbol) indicating the time period (optional, required for TWFE and FD).
`model`	Model type: `"fe"` (One-way Fixed Effects / Within, default), `"twfe"` (Two-Way Fixed Effects: entity + time), `"re"` (Random Effects / Swamy-Arora GLS), `"fd"` (First Differences), `"between"` (Between / Group Means Estimator), or `"pooling"` (Pooled OLS).
`se`	Type of standard errors: `"cluster"` (clustered by entity id, default), `"robust"` (White HC1), or `"standard"` (classical homoskedastic).
`digits`	Number of decimal places to display (default is 4).
`...`	Additional arguments.

Value

An object of class `econ_panel` containing estimates, standard errors, and panel diagnostics.

Examples

```
data(firm_investment, package = "phoring")
# 1. Entity Fixed Effects (Within)
panel(inv ~ value + capital, data = firm_investment, id = "firm_id", time = "year", model = "fe")

# 2. Two-Way Fixed Effects (Entity + Time FE)
panel(inv ~ value + capital, data = firm_investment, id = "firm_id", time = "year", model = "twfe")

# 3. Random Effects (GLS)
panel(inv ~ value + capital, data = firm_investment, id = "firm_id", time = "year", model = "re")
```

`plot.econ_arch` *Plot Autoregressive Conditional Heteroskedasticity (ARCH) Diagnostics*

Description

Generates an academic publication-grade ARCH volatility suite: (1) `"residuals"`: Raw regression residuals, (2) `"squared_residuals"`: Squared residuals e_t^2 (volatility clustering), (3) `"acf_sq"`: Autocorrelations of squared residuals, (4) `"dist"`: Residual histogram with fitted normal curve, (5) `"qq"`: Normal Q-Q plot of residuals, (6) `"rolling_sd"`: Rolling standard deviation of residuals over time.

Usage

```
## S3 method for class 'econ_arch'
plot(
  x,
  y = NULL,
  which = c("all", "residuals", "squared_residuals", "acf_sq", "dist", "qq",
    "rolling_sd"),
  ...
)
```

Arguments

`x`	An object of class `econ_arch`.
`y`	Unused, for compatibility with generic.
`which`	Character string or vector specifying which plot to produce: `"all"` (default 2x2 grid), `"residuals"`, `"squared_residuals"`, `"acf_sq"`, `"dist"`, `"qq"`, or `"rolling_sd"`.
`...`	Additional graphical parameters.

Value

Invisibly returns `x`.

plot.econ_ardl *Plot Autoregressive Distributed Lag (ARDL) Model Diagnostics*

Description

Generates an academic publication-grade ARDL dynamics suite: (1) "all": 6-panel comprehensive academic dashboard (Actual vs. Fitted, Residuals, ACF, Multipliers, CUSUM, CUSUM of Squares), (2) "fit": Actual vs. fitted trajectories over time, (3) "residuals": ARDL residuals over time, (4) "acf": Autocorrelation of residuals, (5) "multipliers": Short-run impact vs. long-run multiplier bar comparison, (6) "cusum": CUSUM parameter stability test with 5% critical bounds, (7) "cusum_sq" / "cusum_squares": Brown-Durbin-Evans CUSUM of squares variance & parameter stability test with 5% critical bounds, (8) "stability": Side-by-side CUSUM and CUSUM of squares stability panel, (9) "bounds": Pesaran-Shin-Smith (2001) Bounds Testing cointegration diagnostic diagram, (10) "cumulative_multipliers": Cumulative interim dynamic multiplier curve.

Usage

```
## S3 method for class 'econ_ardl'
plot(
  x,
  y = NULL,
  which = c("all", "fit", "residuals", "acf", "multipliers", "cusum", "cusum_sq",
    "cusum_squares", "stability", "bounds", "cumulative_multipliers"),
  ...
)
```

Arguments

x	An object of class econ_ardl.
y	Unused, for compatibility with generic.
which	Character string or vector specifying which plot to produce: "all" (default 2x3 comprehensive grid), "fit", "residuals", "acf", "multipliers", "cusum", "cusum_sq", "cusum_squares", "stability", "bounds", or "cumulative_multipliers".
...	Additional graphical parameters.

Value

Invisibly returns x.

plot.econ_chow *Plot Chow Structural Break Test*

Description

Generates an academic publication-grade Chow structural stability suite: (1) "scatter": Outcome vs. predictor scatter showing separate subgroup regression slopes vs. pooled line, (2) "f_stat": Chow F-statistic comparison vs. 5% critical F value.

Usage

```
## S3 method for class 'econ_chow'
plot(x, y = NULL, which = c("all", "scatter", "f_stat"), ...)
```

Arguments

x	An object of class econ_chow.
y	Unused, for compatibility with generic.
which	Character string or vector specifying which plot to produce: "all" (default 2x1 grid), "scatter", or "f_stat".
...	Additional graphical parameters.

Value

Invisibly returns x.

plot.econ_coint *Plot Cointegration and Long-Run Equilibrium Diagnostics*

Description

Generates an academic publication-grade cointegration diagnostic suite: (1) "comovement": Normalized series comovement overlay, (2) "residuals": Cointegrating error-correction residuals with MacKinnon boundary, (3) "acf_residuals": Autocorrelation of cointegrating residuals, (4) "scatter_regression": Scatter of outcome vs. predictor with long-run regression line.

Usage

```
## S3 method for class 'econ_coint'
plot(
  x,
  y = NULL,
 which = c("all", "comovement", "residuals", "acf_residuals", "scatter_regression"),
  ...
)
```

Arguments

x	An object of class econ_coint.
y	Unused, for compatibility with generic.
which	Character string or vector specifying which plot to produce: "all" (default 2x2 grid), "comovement", "residuals", "acf_residuals", or "scatter_regression".
...	Additional graphical parameters.

Value

Invisibly returns x.

---

plot.econ_corr *Plot Correlation Matrix Heatmap*

Description

Displays an academic publication-grade correlation matrix heatmap with serif typography, numeric values overlaid, and significance indicators.

Usage

```
## S3 method for class 'econ_corr'
plot(x, y = NULL, which = c("all", "heatmap"), ...)
```

Arguments

x	An object of class econ_corr.
y	Unused, for compatibility with generic.
which	Character string specifying which plot to produce ("all" or "heatmap").
...	Additional graphical parameters.

Value

Invisibly returns x.

plot.econ_count *Plot Count Model Diagnostics (Poisson / NegBin)*

Description

Generates an academic publication-grade count regression diagnostic suite: (1) "rootogram": Kleiber-Zeileis hanging rootogram of observed vs. expected count frequencies, (2) "freq_bars": Grouped bar comparison of observed vs. expected counts, (3) "mean_variance": Binned mean vs. variance curve (testing equidispersion), (4) "residuals": Pearson residuals vs. fitted counts, (5) "irr_forest": Incidence Rate Ratio (IRR) forest plot $\exp(\beta) \pm 95\%$ CI.

Usage

```
## S3 method for class 'econ_count'
plot(
  x,
  y = NULL,
 which = c("all", "rootogram", "freq_bars", "mean_variance", "residuals", "irr_forest"),
  ...
)
```

Arguments

x	An object of class econ_count.
y	Unused, for compatibility with generic.
which	Character string or vector specifying which plot to produce: "all" (default 2x2 grid), "rootogram", "freq_bars", "mean_variance", "residuals", or "irr_forest".
...	Additional graphical parameters.

Value

Invisibly returns x.

plot.econ_desc *Plot Descriptive Distributions*

Description

Generates an academic publication-grade distribution grid for numeric variables in descriptive summaries: (1) "histogram": Histograms with fitted kernel density curves, (2) "boxplot": Standardized boxplots across variables.

Usage

```
## S3 method for class 'econ_desc'
plot(x, y = NULL, which = c("all", "histogram", "boxplot"), ...)
```

Arguments

x	An object of class econ_desc.
y	Unused, for compatibility with generic.
which	Character string specifying which plot to produce: "all" (default histogram grid), "histogram", or "boxplot".
...	Additional graphical parameters.

Value

Invisibly returns x.

---

plot.econ_desc_balance

Plot Covariate Balance Love Plot

Description

Generates an academic publication-grade Love plot showing standardized mean differences (normalized differences) between treatment and control groups alongside the ± 0.10 balance threshold.

Usage

```
## S3 method for class 'econ_desc_balance'
plot(x, y = NULL, which = c("all", "love"), ...)
```

Arguments

x	An object of class econ_desc_balance.
y	Unused, for compatibility with generic.
which	Character string specifying which plot to produce ("all" or "love").
...	Additional graphical parameters.

Value

Invisibly returns x.

plot.econ_diagnostics *Plot Comprehensive Econometric Diagnostics Dashboard*

Description

Generates an academic publication-grade summary dashboard of diagnostic test results: (1) `"pvalues"`: Horizontal bar chart of $-\log_{10}(p\text{-value})$ for all specification tests (RESET, Breusch-Pagan, White, Jarque-Bera, Durbin-Watson, Breusch-Godfrey) with the 0.05 critical boundary, (2) `"vif"`: Variance Inflation Factors (VIF) bar chart with 5 and 10 multicollinearity reference lines.

Usage

```
## S3 method for class 'econ_diagnostics'
plot(x, y = NULL, which = c("all", "pvalues", "vif"), ...)
```

Arguments

`x`	An object of class `econ_diagnostics`.
`y`	Unused, for compatibility with generic.
`which`	Character string or vector specifying which plot to produce: `"all"` (default 2x1 grid), `"pvalues"`, or `"vif"`.
`...`	Additional graphical parameters.

Value

Invisibly returns `x`.

plot.econ_did *Plot Difference-in-Differences (DiD) Trajectories and Diagnostics*

Description

Generates an academic publication-grade DiD suite: (1) `"trajectory"`: 2x2 Pre/Post trajectories with treatment effect counterfactual, (2) `"means_ci"`: 2x2 Cell group means with 95% confidence intervals, (3) `"comparison"`: DiD estimate comparison vs. naive post-only and pre/post estimates, (4) `"event_study"`: Group averages over calendar time and event-study dynamics.

Usage

```
## S3 method for class 'econ_did'
plot(
  x,
  y = NULL,
  which = c("all", "trajectory", "means_ci", "comparison", "event_study"),
  ...
)
```

Arguments

x	An object of class econ_did.
y	Unused, for compatibility with generic.
which	Character string or vector specifying which plot to produce: "all" (default 2x2 grid), "trajectory", "means_ci", "comparison", or "event_study".
...	Additional graphical parameters.

Value

Invisibly returns x.

---

plot.econ_durbin_h *Plot Durbin's h Autoregressive Serial Correlation Test*

Description

Displays standard normal null density with rejection regions and Durbin's h test statistic.

Usage

```
## S3 method for class 'econ_durbin_h'
plot(x, y = NULL, which = c("all", "density"), ...)
```

Arguments

x	An object of class econ_durbin_h.
y	Unused, for compatibility with generic.
which	Character string specifying which plot to produce ("all" or "density").
...	Additional graphical parameters.

Value

Invisibly returns x.

plot.econ_ecm *Plot Error Correction Model (ECM) Diagnostics*

Description

Generates an academic publication-grade ECM dynamics suite: (1) "fit": Actual vs. fitted first differences Δy_t, (2) "residuals": ECM dynamic residuals, (3) "decay_path": Speed of adjustment disequilibrium decay path $(1+\alpha)^h$, (4) "acf_residuals": Autocorrelation function of ECM residuals, (5) "cusum": CUSUM parameter stability test with 5% bounds, (6) "cusum_sq": CUSUM of squares stability test with 5% bounds, (7) "stability": Side-by-side CUSUM and CUSUMSQ stability panel.

Usage

```
## S3 method for class 'econ_ecm'
plot(
  x,
  y = NULL,
  which = c("all", "fit", "residuals", "decay_path", "acf_residuals", "cusum",
    "cusum_sq", "cusum_squares", "stability"),
  ...
)
```

Arguments

x	An object of class econ_ecm.
y	Unused, for compatibility with generic.
which	Character string or vector specifying which plot to produce: "all" (default 2x2 grid), "fit", "residuals", "decay_path", "acf_residuals", "cusum", "cusum_sq", or "stability".
...	Additional graphical parameters.

Value

Invisibly returns x.

plot.econ_gq *Plot Goldfeld-Quandt Heteroskedasticity Test*

Description

Generates an academic publication-grade Goldfeld-Quandt diagnostic plot: (1) "residuals": Residuals ordered by the sorting variable, highlighting the two subsamples and omitted central observations, (2) "distribution": Theoretical F-distribution with the GQ test statistic and rejection threshold.

Usage

```
## S3 method for class 'econ_gq'
plot(x, y = NULL, which = c("all", "residuals", "distribution"), ...)
```

Arguments

x	An object of class econ_gq.
y	Unused, for compatibility with generic.
which	Character string or vector specifying which plot to produce: "all" (default 2x1 grid), "residuals", or "distribution".
...	Additional graphical parameters.

Value

Invisibly returns x.

---

plot.econ_granger *Plot Granger Causality Diagnostics*

Description

Generates an academic publication-grade Granger causality suite: (1) "timeseries": Standardized time series trajectories, (2) "ccf": Cross-correlation function (CCF) across leads and lags, (3) "f_stats": Bidirectional F-statistic comparison vs. 5% critical F value.

Usage

```
## S3 method for class 'econ_granger'
plot(x, y = NULL, which = c("all", "timeseries", "ccf", "f_stats"), ...)
```

Arguments

x	An object of class econ_granger.
y	Unused, for compatibility with generic.
which	Character string or vector specifying which plot to produce: "all" (default 3 panels), "timeseries", "ccf", or "f_stats".
...	Additional graphical parameters.

Value

Invisibly returns x.

plot.econ_hyp *Plot General Linear Hypothesis & Wald Test*

Description

Displays the theoretical null distribution (F or Chi-Square) with the rejection region and Wald statistic.

Usage

```
## S3 method for class 'econ_hyp'
plot(x, y = NULL, which = c("all", "density"), ...)
```

Arguments

x	An object of class econ_hyp.
y	Unused, for compatibility with generic.
which	Character string specifying which plot to produce ("all" or "density").
...	Additional graphical parameters.

Value

Invisibly returns x.

plot.econ_iv *Plot Instrumental Variables 2SLS Model*

Description

Generates an academic publication-grade 2SLS identification suite: (1) "first_stage": First-Stage Fit: Endogenous X vs. Instrument Z with F-stat, (2) "second_stage": Second Stage: Outcome Y vs. Predicted Endogenous $\hat{X}$, (3) "residuals": 2SLS Residuals vs. Fitted values, (4) "orthogonality": Instrument Exogeneity Check: Residuals vs. Instrument Z, (5) "forest_compare": OLS vs. 2SLS Coefficient Comparison Forest Plot, (6) "f_stat": First-stage F-statistic vs. Stock-Yogo rule of thumb (F = 10), (7) "reduced_form": Reduced form regression of Y on Z.

Usage

```
## S3 method for class 'econ_iv'
plot(
  x,
  y = NULL,
  which = c("all", "first_stage", "second_stage", "residuals", "orthogonality",
    "forest_compare", "f_stat", "reduced_form"),
  ...
)
```

Arguments

x	An object of class econ_iv.
y	Unused, for compatibility with generic.
which	Character string or vector specifying which plot to produce: "all" (default 2x2 grid), "first_stage", "second_stage", "residuals", "orthogonality", "forest_compare", "f_stat", or "reduced_form".
...	Additional graphical parameters.

Value

Invisibly returns x.

---

plot.econ_jtest *Plot Davidson-MacKinnon J-Test*

Description

Generates an academic publication-grade J-test comparison bar chart showing the t-statistics for competing non-nested models against the ± 1.96 critical thresholds.

Usage

```
## S3 method for class 'econ_jtest'
plot(x, y = NULL, which = c("all", "bars"), ...)
```

Arguments

x	An object of class econ_jtest.
y	Unused, for compatibility with generic.
which	Character string specifying which plot to produce ("all" or "bars").
...	Additional graphical parameters.

Value

Invisibly returns x.

plot.econ_koyck *Plot Koyck Distributed Lag Multipliers and Diagnostics*

Description

Generates an academic publication-grade Koyck distributed lag suite: (1) "lag_weights": Geometric lag decay weights across horizons, (2) "cumulative": Cumulative interim multipliers approaching long-run effect, (3) "actual_fitted": Actual vs. fitted values over time, (4) "acf_residuals": Autocorrelation function of Koyck residuals.

Usage

```
## S3 method for class 'econ_koyck'
plot(
  x,
  y = NULL,
  which = c("all", "lag_weights", "cumulative", "actual_fitted", "acf_residuals",
    "cusum", "cusum_sq", "cusum_squares", "stability"),
  horizons = 12,
  ...
)
```

Arguments

x	An object of class econ_koyck.
y	Unused, for compatibility with generic.
which	Character string or vector specifying which plot to produce: "all" (default 2x2 grid), "lag_weights", "cumulative", "actual_fitted", "acf_residuals", "cusum", "cusum_sq", or "stability".
horizons	Integer specifying the number of lag horizons to plot (default 12).
...	Additional graphical parameters.

Value

Invisibly returns x.

plot.econ_logit *Plot Binary Outcome (Logit/Probit) Model Diagnostics*

Description

Generates an academic publication-grade classification and link diagnostics suite: (1) "scurve": S-Curve sigmoid probability curve vs. linear predictor η, (2) "roc": Receiver Operating Characteristic (ROC) curve with AUC annotation, (3) "distribution": Class separation boxplots of predicted probabilities by observed class, (4) "residuals": Deviance residuals vs. predicted probabilities, (5) "calibration": Hosmer-Lemeshow decile calibration plot with 45-degree reference line, (6) "thresholds": Sensitivity & Specificity curves vs. cutoff with Youden's optimal threshold, (7) "marginal": Marginal probability curve across continuous predictor, (8) "odds_ratio": Odds ratios forest plot (for logit) with 95% CIs on log scale, (9) "confusion": Confusion matrix classification tile heatmap at threshold 0.5.

Usage

```
## S3 method for class 'econ_logit'
plot(
  x,
  y = NULL,
  which = c("all", "scurve", "roc", "distribution", "residuals", "calibration",
    "thresholds", "marginal", "odds_ratio", "confusion"),
  ...
)
```

Arguments

x	An object of class econ_logit.
y	Unused, for compatibility with generic.
which	Character string or vector specifying which plot to produce: "all" (default 2x2 grid), "scurve", "roc", "distribution", "residuals", "calibration", "thresholds", "marginal", "odds_ratio", or "confusion".
...	Additional graphical parameters.

Value

Invisibly returns x.

---

plot.econ_outliers *Plot Influential Outliers & Leverage Diagnostics*

Description

Generates an academic publication-grade outlier influence suite: (1) "influence": Influence bubble plot (Studentized residuals vs. Leverage, bubble size = Cook's D), (2) "cooks": Cook's distance index barplot with 4/n threshold.

Usage

```
## S3 method for class 'econ_outliers'
plot(x, y = NULL, which = c("all", "influence", "cooks"), ...)
```

Arguments

x	An object of class econ_outliers.
y	Unused, for compatibility with generic.
which	Character string or vector specifying which plot to produce: "all" (default 2x1 grid), "influence", or "cooks".
...	Additional graphical parameters.

Value

Invisibly returns x.

---

plot.econ_panel *Plot Panel Data Regression Diagnostics*

Description

Generates an academic publication-grade panel diagnostics suite or individual plots: (1) "between": Between-entity variation in group means with between regression line, (2) "within": Within-entity demeaned scatter with within regression slope, (3) "spaghetti": Entity trajectories over time with mean trend, (4) "fe_dist": Distribution of estimated individual fixed effects $\hat{\alpha}_i$, (5) "heterogeneity": Individual entity residual boxplots, (6) "residuals": Panel residuals vs. fitted values with LOESS smoother, (7) "ar_residual": Residual vs. lagged within-entity residual $e_{i,t}$ vs $e_{i,t-1}$ (Wooldridge AR(1) test), (8) "time_effects": Period mean residuals $\pm 95\%$ CI across time.

Usage

```
## S3 method for class 'econ_panel'
plot(
  x,
  y = NULL,
  which = c("all", "between", "within", "spaghetti", "fe_dist", "heterogeneity",
    "residuals", "ar_residual", "time_effects"),
  ...
)
```

Arguments

x	An object of class econ_panel.
y	Unused, for compatibility with generic.
which	Character string or vector specifying which plot to produce: "all" (default 2x2 grid), "between", "within", "spaghetti", "fe_dist", "heterogeneity", "residuals", "ar_residual", or "time_effects".
...	Additional graphical parameters.

Value

Invisibly returns x.

---

plot.econ_prais *Plot Prais-Winsten / Cochrane-Orcutt AR(1) Diagnostics*

Description

Generates an academic publication-grade serial correlation correction suite: (1) "fitted": Actual vs. FGLS fitted values, (2) "residuals": Transformed quasi-differenced residuals, (3) "acf": Autocorrelation function of transformed residuals, (4) "ar_scatter": e_t vs. e_{t-1} scatter comparing OLS vs. FGLS residuals with $\hat{\rho}$ slope.

Usage

```
## S3 method for class 'econ_prais'
plot(
  x,
  y = NULL,
 which = c("all", "fitted", "residuals", "acf", "ar_scatter", "cusum", "cusum_sq",
    "cusum_squares", "stability"),
  ...
)
```

Arguments

x	An object of class econ_prais.
y	Unused, for compatibility with generic.
which	Character string or vector specifying which plot to produce: "all" (default 2x2 grid), "fitted", "residuals", "acf", "ar_scatter", "cusum", "cusum_sq", or "stability".
...	Additional graphical parameters.

Value

Invisibly returns x.

plot.econ_qreg *Plot Quantile Regression Process*

Description

Generates an academic publication-grade quantile process suite: (1) "process": Coefficient path $\hat{\beta}(\tau)$ across quantiles τ with 95% pointwise CI band, alongside the horizontal OLS benchmark estimate and 95% confidence interval, (2) "loss": Total check-loss across quantiles τ, (3) "fitted_scatter": Scatter of outcome vs. predictor with fitted quantile lines.

Usage

```
## S3 method for class 'econ_qreg'
plot(
  x,
  y = NULL,
  var = NULL,
  which = c("all", "process", "loss", "fitted_scatter"),
  ...
)
```

Arguments

x	An object of class econ_qreg.
y	Unused, for compatibility with generic.
var	Character string specifying which coefficient to plot. Defaults to first non-intercept predictor.
which	Character string or vector specifying which plot to produce: "all" (default 2x1 grid), "process", "loss", or "fitted_scatter".
...	Additional graphical parameters.

Value

Invisibly returns x.

plot.econ_rdd *Plot Sharp Regression Discontinuity (RDD) Diagnostics*

Description

Generates an academic publication-grade RDD identification suite: (1) "binned_rd": Binned RD scatter plot with local polynomial fits and 95% confidence bands, (2) "density": Running variable density histogram with discontinuity check around cutoff, (3) "bandwidth_sensitivity": Point estimate $\hat{\tau} \pm 95\%$ CI across bandwidths, (4) "placebo": Placebo cutoff falsification test.

Usage

```
## S3 method for class 'econ_rdd'
plot(
  x,
  y = NULL,
  which = c("all", "binned_rd", "density", "bandwidth_sensitivity", "placebo"),
  ...
)
```

Arguments

x	An object of class econ_rdd.
y	Unused, for compatibility with generic.
which	Character string or vector specifying which plot to produce: "all" (default 2x2 grid), "binned_rd", "density", "bandwidth_sensitivity", or "placebo".
...	Additional graphical parameters.

Value

Invisibly returns x.

---

plot.econ_tobit *Plot Tobit Censored Regression Diagnostics*

Description

Generates an academic publication-grade Tobit diagnostic suite: (1) "observed_hist": Observed Y histogram highlighting censoring pile-up, (2) "pred_observed": Observed Y vs. unconditional predicted $E[y]$, (3) "mcdonald_moffitt": McDonald-Moffitt marginal decomposition (extensive vs. intensive margin), (4) "qq_resid": Normal Q-Q plot of residuals for uncensored observations, (5) "residuals": Censored residuals vs. linear index.

Usage

```
## S3 method for class 'econ_tobit'
plot(
  x,
  y = NULL,
 which = c("all", "observed_hist", "pred_observed", "mcdonald_moffitt", "qq_resid",
    "residuals"),
  ...
)
```

Arguments

x	An object of class econ_tobit.
y	Unused, for compatibility with generic.
which	Character string or vector specifying which plot to produce: "all" (default 2x2 grid), "observed_hist", "pred_observed", "mcdonald_moffitt", "qq_resid", or "residuals".
...	Additional graphical parameters.

Value

Invisibly returns x.

---

plot.econ_unit_root *Plot Unit Root and Stationarity Tests*

Description

Generates an academic publication-grade time-series stationarity suite: (1) "levels": Series levels over time, (2) "diffs": First differences over time, (3) "acf_levels": Autocorrelation of levels, (4) "acf_diffs": Autocorrelation of first differences, (5) "pacf": Partial autocorrelations of levels and differences, (6) "dashboard": Test statistic dashboard (ADF, PP, KPSS vs. 5% critical values).

Usage

```
## S3 method for class 'econ_unit_root'
plot(
  x,
  y = NULL,
 which = c("all", "levels", "diffs", "acf_levels", "acf_diffs", "pacf", "dashboard"),
  ...
)
```

Arguments

x	An object of class econ_unit_root.
y	Unused, for compatibility with generic.
which	Character string or vector specifying which plot to produce: "all" (default 2x2 grid), "levels", "diffs", "acf_levels", "acf_diffs", "pacf", or "dashboard".
...	Additional graphical parameters.

Value

Invisibly returns x.

plot.econ_var *Plot Vector Autoregression (VAR) Diagnostics and Dynamics*

Description

Generates an academic publication-grade VAR dynamics suite: (1) "stability": Inverse roots of companion matrix characteristic polynomial on the complex unit circle, (2) "irf": Orthogonalized Impulse Response Functions (IRF) grid across horizons, (3) "fevd": Forecast Error Variance Decomposition (FEVD) stacked bars, (4) "residuals": Residuals across equations over time, (5) "correlation": Innovation covariance/correlation heatmap, (6) "acf": Autocorrelation function of residuals for each equation, (7) "granger": Granger causality p-value heatmap matrix.

Usage

```
## S3 method for class 'econ_var'
plot(
  x,
  y = NULL,
  which = c("all", "stability", "irf", "fevd", "residuals", "correlation", "acf",
    "granger"),
  horizons = 10,
  ...
)
```

Arguments

x	An object of class econ_var.
y	Unused, for compatibility with generic.
which	Character string or vector specifying which plot to produce: "all" (default 2x2 grid), "stability", "irf", "fevd", "residuals", "correlation", "acf", or "granger".
horizons	Number of horizons for IRF and FEVD (default 10).
...	Additional graphical parameters.

Value

Invisibly returns x.

`prais_winsten` *Prais-Winsten & Cochrane-Orcutt FGLS Autocorrelation Correction*

Description

Fits Feasible Generalized Least Squares (FGLS) to correct for first-order AR(1) serial correlation (Prais and Winsten, 1954; Cochrane and Orcutt, 1949), using Prais-Winsten (retaining the first observation) or Cochrane-Orcutt transformation until convergence.

Usage

```
prais_winsten(
  formula,
  data,
  method = c("prais", "cochrane"),
  max_iter = 50,
  tol = 1e-04,
  digits = 4
)
```

Arguments

`formula`	Model formula (e.g., `y ~ x1 + x2`).
`data`	A data.frame containing time series variables.
`method`	Either `"prais"` (default, Prais-Winsten) or `"cochrane"` (Cochrane-Orcutt).
`max_iter`	Maximum iterations for rho convergence (default is 50).
`tol`	Convergence tolerance (default is 1e-4).
`digits`	Number of decimal places to format (default is 4).

Value

An object of class `econ_prais`.

Examples

```
data(macro_data, package = "phoring")
prais_winsten(inflation ~ fed_funds + unemployment, data = macro_data)
```

qreg *Quantile Regression Across the Distribution*

Description

Estimates quantile regressions for single or multiple conditional quantiles (e.g., 10th, 25th, median 50th, 75th, 90th percentiles), allowing researchers to analyze how economic covariates impact different parts of the outcome distribution beyond the conditional mean.

Usage

```
qreg(
  formula,
  data,
  tau = c(0.1, 0.25, 0.5, 0.75, 0.9),
  se = c("robust", "classical"),
  digits = 4
)
```

Arguments

formula	Model formula (e.g., wage ~ educ + exper).
data	A data.frame containing the variables.
tau	Quantile(s) to estimate, between 0 and 1 (default is c(0.1, 0.25, 0.5, 0.75, 0.9)).
se	Standard error type: "robust" (Huber sandwich, default) or "classical".
digits	Number of decimal places to format (default is 4).

Value

An object of class econ_qreg.

Examples

```
data(wage_data, package = "phoring")
# Estimate median regression (tau = 0.5)
qreg(wage ~ educ + exper, data = wage_data, tau = 0.5)

# Estimate full distribution quantiles (10%, 25%, 50%, 75%, 90%)
q_all <- qreg(wage ~ educ + exper, data = wage_data, tau = c(0.25, 0.5, 0.75))
```

rdd *Sharp Regression Discontinuity Design (RDD)*

Description

Estimates causal treatment effects at a policy cutoff using Sharp Regression Discontinuity (SRD) with local linear regression, optimal bandwidth selection, triangular/uniform kernel weights, and robust standard errors.

Usage

```
rdd(
  formula,
  data,
  cutoff = 0,
  bandwidth = NULL,
  kernel = c("triangular", "uniform"),
  digits = 4
)
```

Arguments

formula	Model formula where the first RHS variable is the running variable (e.g., outcome ~ running_var or outcome ~ running_var + controls).
data	A data.frame containing the variables.
cutoff	Threshold value of the running variable that determines treatment (default is 0).
bandwidth	Bandwidth window around the cutoff. If NULL (default), computes the optimal rule-of-thumb bandwidth (Silverman/Imbens-Kalyanaraman).
kernel	Kernel weighting function: "triangular" (default, optimal for boundary estimation) or "uniform".
digits	Number of decimal places to format (default is 4).

Value

An object of class econ_rdd.

Examples

```
set.seed(42)
n <- 500
x <- runif(n, -2, 2)
d <- as.numeric(x >= 0)
y <- 2 + 3 * d + 1.5 * x + 0.8 * d * x + rnorm(n, sd = 0.5)
df <- data.frame(y = y, score = x)
rdd(y ~ score, data = df, cutoff = 0)
```

reg *One-Command Econometric Regression with Automated Diagnostics*

Description

Fits an Ordinary Least Squares (OLS) or Weighted Least Squares (WLS) regression model, calculates robust (HC0-HC3), clustered, or HAC standard errors, and automatically runs and prints a complete battery of econometric diagnostic tests.

Usage

```
reg(
  formula,
  data,
  se = c("robust", "standard", "cluster", "hac", "HC0", "HC1", "HC2", "HC3"),
  cluster = NULL,
  lag = NULL,
  weights = NULL,
  digits = 4,
  diagnostics = TRUE,
  ...
)
```

Arguments

formula	An object of class formula (e.g., y ~ x1 + x2).
data	A data.frame containing the variables in the model.
se	Type of standard errors to compute: "robust" (White HC1, default), "standard" (classical OLS), "cluster" (cluster-robust), "hac" (Newey-West autocorrelation-consistent), or specific types "HC0", "HC1", "HC2", "HC3".
cluster	Vector or column name indicating cluster grouping for clustered standard errors.
lag	Number of lags for Newey-West HAC standard errors (optional, defaults to automatic rule).
weights	Optional weights vector or column name for Weighted Least Squares (WLS).
digits	Number of decimal places to display in output tables (default is 4).
diagnostics	Logical: whether to automatically compute and display diagnostic tests (default is TRUE).
...	Additional arguments passed to lm.

Value

An object of class econ_reg containing the regression results, standard errors, coefficient table, model statistics, and diagnostic tests.

Examples

```
data(mtcars)
reg(mpg ~ wt + hp + qsec, data = mtcars)
reg(mpg ~ wt + hp, data = mtcars, cluster = "cyl")
```

---

test_hyp *General Linear Hypothesis & Wald F-Test*

Description

Tests linear hypotheses of model parameters using the robust or clustered covariance matrix from the estimated model. Supports interaction terms, factor levels, polynomial terms, and multi-parameter restrictions.

Usage

```
test_hyp(model, hypothesis, ...)
```

Arguments

model	A fitted model object from reg, iv, panel, logit, tobit, or lm.
hypothesis	Character vector of linear restrictions (e.g., "educ + exper = 1", "treat:post = 0", or c("wt = 0", "hp = 0")).
...	Additional arguments.

Value

An object of class econ_hyp containing the Wald test statistic, degrees of freedom, and p-value.

Examples

```
data(mtcars)
m <- reg(mpg ~ wt + hp + qsec, data = mtcars)
test_hyp(m, "wt + hp = 0")
test_hyp(m, c("wt = 0", "hp = 0"))
```

---

timeseries *Complete Time Series Econometric Toolkit*

Description

Provides a comprehensive suite for time series econometrics: 1. ‘unit_root()‘: Multi-test stationarity battery (ADF, Phillips-Perron, KPSS). 2. ‘coint()‘: Engle-Granger two-step cointegration test. 3. ‘granger()‘: Bidirectional Granger causality test. 4. ‘arch_test()‘: Engle’s ARCH LM test for volatility clustering. 5. ‘ardl()‘: Autoregressive Distributed Lag model with short-run and long-run multipliers.

tobit — *Censored Regression (Tobit Model) with McDonald-Moffitt Decomposition*

Description

Fits a Tobit regression model (left-censored at 0 or a user-specified threshold), calculates robust or standard standard errors, and automatically produces the McDonald-Moffitt decomposition of marginal effects.

Usage

```
tobit(formula, data, left = 0, se = c("robust", "standard"), digits = 4)
```

Arguments

formula	Model formula (e.g., hours ~ educ + exper + age).
data	A data.frame containing the variables.
left	Left-censoring cutoff point (default is 0).
se	Type of standard errors: "robust" (sandwich, default) or "standard" (MLE Hessian).
digits	Number of decimal places to display (default is 4).

Value

An object of class econ_tobit.

Examples

```
data(labor_supply, package = "phoring")
labor_supply$hours <- pmax(labor_supply$hours, 0)
tobit(hours ~ educ + exper + age, data = labor_supply, left = 0)
```

unit_root — *Multi-Test Stationarity Battery (ADF, Phillips-Perron, KPSS)*

Description

Executes Augmented Dickey-Fuller (ADF), Phillips-Perron (PP), and KPSS tests simultaneously with MacKinnon response surface p-values.

Usage

```
unit_root(x, lags = NULL, type = c("trend", "drift", "none"), digits = 4)
```

Arguments

x	A numeric vector or time series.
lags	Number of lags for ADF test (optional; automatically selected by AIC if NULL).
type	Model specification: "trend" (constant and linear trend, default), "drift" (constant only), or "none" (neither constant nor trend).
digits	Number of decimals to display (default is 4).

Value

An object of class econ_unit_root.

Examples

```
data(macro_data, package = "phoring")
unit_root(macro_data$inflation, type = "trend")
unit_root(macro_data$unemployment, type = "drift")
```

---

var_model *Vector Autoregression (VAR) Model*

Description

Fits a Vector Autoregression system (Sims, 1980) for multiple time series, computing system equation estimates, information criteria (AIC, BIC), and pairwise Granger causality summary matrices.

Usage

```
var_model(data, vars, p = 1, digits = 4)
```

Arguments

data	A data.frame containing numeric time series variables.
vars	Character vector of column names to include in the VAR system.
p	Lag order (default is 1).
digits	Number of decimal places to format (default is 4).

Value

An object of class econ_var.

Examples

```
data(macro_data, package = "phoring")
var_model(macro_data, vars = c("inflation", "unemployment", "fed_funds"), p = 1)
```

wage_data *Cross-Sectional and Panel Wage Determination Dataset*

Description

A simulated labor economics dataset modeled after the Mincerian wage equation and empirical labor economics literature. Includes individual characteristics, parental education instruments, firm identifiers for clustering, and panel time periods.

Usage

```
wage_data
```

Format

A data frame with 600 observations and 11 variables:

wage Hourly wage in US dollars

educ Years of education

exper Years of labor market experience

tenure Years with current employer

female Indicator variable: 1 if female, 0 if male

married Indicator variable: 1 if married, 0 otherwise

mother_educ Mother's years of education (instrument for educ)

father_educ Father's years of education (instrument for educ)

firm_id Firm identifier for clustering and panel analysis

year Year of observation (2020-2023)

inlf Indicator variable: 1 if in labor force / high wage, 0 otherwise

Source

Simulated based on standard labor econometrics specifications.

word_table *Side-by-Side Model Comparison & Word Table Export*

Description

Compiles multiple econometric models into a single publication-quality table with standard errors in parentheses and significance stars, and exports directly to Microsoft Word (.docx/.doc), HTML, or Markdown.

Usage

```
word_table(
  ...,
  labels = NULL,
  file = NULL,
  title = "Regression Results",
  format = c("text", "html", "word"),
  digits = 3
)
```

Arguments

...	One or more fitted model objects (from reg, iv, panel, logit, tobit, count_reg, did, koyck, prais_winsten, ecm, ardl, or lm).
labels	Optional character vector of custom column labels for each model (e.g., c("OLS", "FE", "2SLS")).
file	Optional filename for export (e.g., "table1.docx" or "table1.html").
title	Title for the table.
format	Output display format: "text" (console, default), "html", or "word".
digits	Number of decimal places to format (default is 3).

Value

A list of the input models (invisibly).

Examples

```
data(mtcars)
m1 <- reg(mpg ~ wt, data = mtcars)
m2 <- reg(mpg ~ wt + hp, data = mtcars)
word_table(m1, m2, labels = c("Simple OLS", "Multiple OLS"))
```

Index

∗ **datasets**
college_distance, 6
crime_panel, 8
firm_investment, 14
house_prices, 16
labor_supply, 20
macro_data, 22
min_wage, 22
wage_data, 52

arch_test, 3
ardl, 4, *53*

chow_test, 5
coint, 5
college_distance, 6
corr, 7
count_reg, 7, *53*
crime_panel, 8

desc, 9
diagnostics, 9
did, 10, *53*
durbin_h, 11
dynamics, 12

ecm, 12, *53*
econ_plot, 13

firm_investment, 14
formula, *48*

glm, *21*
gq_test, 15
granger, 15

house_prices, 16

interpret, 17
iv, 17, *49*, *53*

j_test, 18

koyck, 19, *53*

labor_supply, 20
lm, *3*, *48*, *49*, *53*
logit, 20, *49*, *53*

macro_data, 22
min_wage, 22

outliers, 23

panel, 24, *49*, *53*
plot.econ_arch, 25
plot.econ_ardl, 26
plot.econ_chow, 27
plot.econ_coint, 27
plot.econ_corr, 28
plot.econ_count, 29
plot.econ_desc, 29
plot.econ_desc_balance, 30
plot.econ_diagnostics, 31
plot.econ_did, 31
plot.econ_durbin_h, 32
plot.econ_ecm, 33
plot.econ_gq, 33
plot.econ_granger, 34
plot.econ_hyp, 35
plot.econ_iv, 35
plot.econ_jtest, 36
plot.econ_koyck, 37
plot.econ_logit, 37
plot.econ_outliers, 38
plot.econ_panel, 39
plot.econ_prais, 40
plot.econ_qreg, 41
plot.econ_rdd, 41
plot.econ_reg *(*econ_plot*)*, 13
plot.econ_tobit, 42
plot.econ_unit_root, 43
plot.econ_var, 44

prais_winsten, 45, *53*
probit *(*logit*)*, 20

qreg, 46

rdd, 47
reg, *3*, *11*, *13*, *17*, 48, *49*, *53*

test_hyp, 49
timeseries, 49
tobit, *49*, 50, *53*

unit_root, 50

var_model, 51

wage_data, 52
word_table, 53