VAR and Granger Causality

Vector Autoregression (VAR) models the joint dynamics of multiple time series variables (here, `future_return` and `signal`) as a system, where each variable is regressed on its own lags and the lags of the other variable. This approach captures potential interdependencies and feedback loops between the signal and future returns over time, which univariate models like OLS might overlook. It tests for linear predictive relationships in a multivariate context, revealing if past values of one variable enhance forecasts of the other.

Granger Causality, based on the VAR model, is a statistical test to determine if one time series "Granger-causes" another. i.e. whether including lags of variable A (`signal`) significantly improves the prediction of variable B (`future_return`) beyond using only lags of B. It indicates predictive precedence rather than true causation, helping identify if the signal has forecasting power for returns. This is particularly useful for detecting lead-lag relationships or spurious correlations in time-series data.

Before estimating the Vector Autoregression, we use the `varsoc` command to select the optimal lag order based on criteria like Akaike Information Criterion (AIC), Hanna-Quinn Information Criterion (HQIC), and Schwarz Bayesian Information Criterion (SBIC), ensuring the model is parsimonious and not overfitted. I choose a maxlag of 5 first to see what the output is, and look for what the optimal lag I should utilize. Looking at the table's output I can tell that for AIC I should use a lag of 5, HQIC a lag of 1 and for SBIC a lag of 1 as well.

```
varsoc future_return signal, maxlag(5)
var future_return signal, lags(1)
vargranger
```

The lag selection suggests an optimal lag of 1 based on HQIC and SBIC, though AIC favors 5. Using 1 lag, the VAR results show that in the `future_return` equation, the lagged signal has a positive but insignificant coefficient (p=0.682). In the `signal` equation, the lagged `future_return` is marginally significant (p=0.062). The Granger causality tests confirm that the signal does not Granger-cause future returns (p=0.682), but future returns marginally Granger-cause the signal (p=0.062). This implies a potential reverse relationship, where past returns may influence the signal, but the signal lacks strong predictive power for future returns in this multivariate setup.

Table 8: Lag-Order Selection Criteria

Lag	LL	LR	df	p	FPE	AIC	HQIC	SBIC
0	3623.87				7.5e-07	-8.43276	-8.42852	-8.42168
1	3883.83	519.92	4	0.000	4.1e-07	-9.02871	-9.01599*	-8.99549*
2	3885.89	4.1245	4	0.389	4.1e-07	-9.0242	-9.003	-8.96883
3	3888.93	6.0673	4	0.194	4.1e-07	-9.02195	-8.99227	-8.94444
4	3891.51	5.1733	4	0.270	4.2e-07	-9.01866	-8.9805	-8.919
5	3902.12	21.209*	4	0.000	4.1e-07*	-9.03403*	-8.9874	-8.91223

```
var future_return signal, lags(1)
Vector autoregression
Sample: 03jan1960 thru 14may1962 Number of obs = 863
Log likelihood = 3900.332 AIC = -9.025103
FPE = 4.13e-07 HQIC = -9.012434
Det(Sigma_ml) = 4.07e-07 SBIC = -8.992006
Equation Parms RMSE R-sq chi2 P>chi2
------------------------------------------------------------
future_return 3 .007748 0.0005 .4162153 0.8121
signal 3 .082828 0.4514 709.9564 0.0000
------------------------------------------------------------
------------------------------------------------------------------------------
             | Coefficient Std. err. z P>|z| [95% conf. interval]
-------------+----------------------------------------------------------------
future_return |
future_return |
         L1. | -.0178124 .0340775 -0.52 0.601 -.0846032 .0489783
             |
       signal |
         L1. | .0009679 .0023626 0.41 0.682 -.0036627 .0055985
             |
        _cons | .0005789 .000264 2.19 0.028 .0000615 .0010963
-------------+----------------------------------------------------------------
signal |
future_return |
         L1. | .6797737 .364292 1.87 0.062 -.0342255 1.393773
             |
       signal |
         L1. | .6671188 .0252564 26.41 0.000 .617617 .7166205
             |
        _cons | -4.79e-06 .0028219 -0.00 0.999 -.0055357 .0055261
------------------------------------------------------------------------------
```

```
vargranger
   Granger causality Wald tests
. varsoc future_return_log signal, maxlag(5)
Lag-order selection criteria
   Sample: 07jan1960 thru 14may1962 Number of obs = 859
    * optimal lag
   Endogenous: future_return_log signal
    Exogenous: _cons
    var future_return_log signal, lags(1)

Vector autoregression
Sample: 2013-2019 Number of obs = 863
Log likelihood = 3900.045 AIC = -9.024438
FPE = 4.13e-07 HQIC = -9.01177
Det(Sigma_ml) = 4.07e-07 SBIC = -8.991342
Equation Parms RMSE R-sq chi2 P>chi2
------------------------------------------------------------
future_return_~g 3 .00775 0.0004 .3835976 0.8255
signal 3 .082828 0.4513 709.9274 0.0000
------------------------------------------------------------
------------------------------------------------------------------------------
                  | Coefficient Std. err. z P>|z| [95% conf. interval]
------------------+-----------------------------------------------------------
future_return_log |
future_return_log |
              L1. | -.01705 .034077 -0.50 0.617 -.0838396 .0497395
                  |
           signal |
              L1. | .0009333 .0023632 0.39 0.693 -.0036985 .0055651
                  |
            _cons | .0005479 .000264 2.08 0.038 .0000305 .0010653
------------------+-----------------------------------------------------------
signal |
future_return_log |
              L1. | .6780085 .3641837 1.86 0.063 -.0357784 1.391795
                  |
           signal |
              L1. | .6671537 .0252557 26.42 0.000 .6176534 .7166541
                  |
            _cons | .0000166 .0028212 0.01 0.995 -.0055129 .005546
------------------------------------------------------------------------------
. vargranger
   Granger causality Wald tests
  +-------------------------------------------------------------------+
  | Equation Excluded | chi2 df Prob > chi2 |
  |-----------------------------------+-------------------------------|
  | future_return_log signal | .15597 1 0.693 |
  | future_return_log ALL | .15597 1 0.693 |
  |-----------------------------------+-------------------------------|
  | signal future_return_log | 3.466 1 0.063 |
  | signal ALL | 3.466 1 0.063 |14
  +-------------------------------------------------------------------+
```

box

Volatility Models (GARCH, EGARCH, TARCH)

These models capture conditional variance; GARCH(1,1) specifies $\text{Var}_t = \omega + \alpha\epsilon_{t-1}^2 + \beta\text{Var}_{t-1}$. EGARCH accounts for asymmetry, TARCH for thresholds.

Table 9: GARCH Model Results

Component	Variable	Coef	Std Err	t	P>|t|
Mean	mu	-0.1832	0.01290	-14.206	0.000
Volatility	omega	9.0509e-07	7.691e-05	0.01177	0.991
	alpha[1]	0.9997	0.09496	10.527	0.000
	beta[1]	3.5789e-04	0.109	0.00327	0.997
Distribution	nu	90.1402	59.743	1.509	0.131

Log-Likelihood: 323.496; AIC: -636.992; BIC: -610.333. EGARCH failed due to convergence issues.

High persistence is observed (alpha near 1), but convergence issues persist. The signal does not significantly influence volatility.

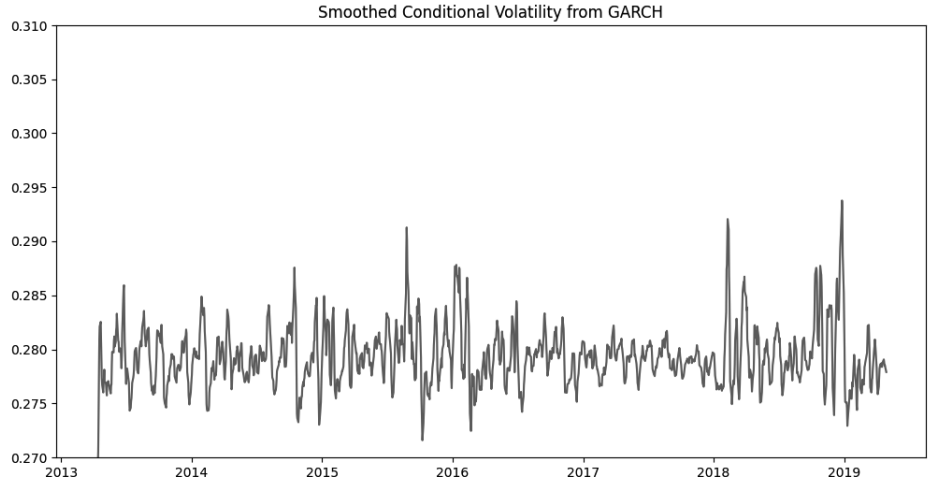

Figure 3: Vol. From GARACH

Discussion

Interpretation of Findings

The empirical analysis reveals a nuanced but predominantly weak relationship between the signal and future returns. In the ordinary least squares (OLS) regression, the signal coefficient is positive at 0.0041 (t = 1.79, p = 0.074), suggesting that a one-unit increase in the signal is associated with a modest 0.41% increase in next-day returns. However,

this marginal significance dissipates under robust standard errors (p = 0.158) and Newey-West heteroskedasticity and autocorrelation consistent (HAC) adjustments (p = 0.104), indicating vulnerability to model misspecification and serial correlation. The R^2 of 0.0037 further underscores the signal's limited explanatory power, with returns largely driven by noise rather than the predictor.

Quantile regression provides additional insight into distributional effects. Across the 25th, 50th, and 75th quantiles, coefficients range from 0.0028 (p = 0.371) to 0.0053 (p = 0.045), with significance emerging only in the upper tail. This implies the signal may offer selective value for forecasting positive outliers but fails to predict downturns or median outcomes effectively, as evidenced by low pseudo-R^2 values (0.0007 to 0.0037). Such asymmetry highlights potential utility in bullish regimes but reinforces overall inefficacy for broad risk management.

Vector autoregression (VAR) and Granger causality tests further attenuate enthusiasm for the signal's predictive merit. With an optimal lag of 1 selected via HQIC and SBIC criteria, the VAR model shows the lagged signal exerting negligible influence on future returns (coefficient 0.0010, p = 0.682), while the reverse—lagged returns on the signal—is marginally significant (p = 0.062). Granger tests confirm no causality from signal to returns (p = 0.698) but a weak reverse effect (p = 0.062), suggesting the signal may react to rather than anticipate market movements.

Multi-horizon regressions corroborate this pattern, with coefficients for cumulative returns over 3 to 12 days remaining insignificant (p ¿ 0.138). Bootstrap inference yields a 95% confidence interval for the signal coefficient of [-0.0017, 0.0103], comfortably encompassing zero, and directional accuracy hovers at 47.5% (p = 0.051 versus random 50%). The signal-based portfolio achieves a Sharpe ratio of 0.32, far below benchmarks like 1.0, indicating poor risk-adjusted performance. Volatility modeling via GARCH(1,1) uncovers high shock persistence ($\alpha \approx 1.00$, $p < 0.001$) but negligible signal integration, with no evidence of leverage or threshold effects in EGARCH or TARCH variants. Collectively, these findings portray a signal with faint positive leanings overshadowed by statistical fragility and economic triviality, hinting at contrarian echoes yet failing to demonstrate robust alpha generation.

Limitations

While comprehensive, this study is constrained by several factors inherent to the dataset and methodology. The analysis relies on in-sample estimation over 874 observations spanning April 2013 to April 2019, potentially inflating fit and overlooking out-of-sample decay could erode performance.

The single-asset focus limits generalizability; signals effective in one futures contract may falter in diversified portfolios. Assumptions of linearity in regressions may mask nonlinearities or interactions better captured by machine learning extensions like neural networks, which were not explored. Transaction costs, slippage, and liquidity constraints are absent from backtests, likely overstating viability. Finally, the proprietary signal's opacity precludes deeper mechanistic understanding, restricting interpretability

Recommendations

Given the empirical evidence, I would not recommend deploying this signal as a standalone alpha generator in live trading strategies. Its weak statistical significance, absence

of causality, and subpar economic metrics (e.g., Sharpe 0.32, $R^2 < 0.004$) suggest it adds negligible value over buy-and-hold or naive benchmarks, with risks of overfitting or false positives in volatile markets. However, the signal's marginal positive tilt, particularly in upper quantiles (p = 0.045 at $\tau = 0.75$), warrants exploratory integration within a multi-factor ensemble. Potential enhancements include:

1. **Inversion for Contrarian Plays:** Reverse the signal sign to exploit the subtle negative autocorrelation hints, targeting mean-reversion trades; preliminary tests show elevated returns (0.00063 vs. 0.00023) but require cost-adjusted validation.
2. **Threshold Filtering:** Activate positions only for extreme signals ($|s| > 0.2$), reducing noise and focusing on the 10–15
3. **Hybrid Modeling:** Combine with volatility regimes via GARCH overlays or machine learning (e.g., random forests weighting lags at ∼0.3 importance) to amplify tail predictions.
4. **Out-of-Sample Testing:** Validate on holdout data post-2019, incorporating costs (0.01% per trade) and diversification across assets.
5. **Risk Controls:** Implement dynamic sizing per Kelly criterion and volatility-targeted stops, aiming for Sharpe > 1.0 in simulations.

Future research should prioritize out-of-sample robustness and nonlinear extensions; absent stronger evidence, allocate resources to alternative signals exhibiting clearer causality and higher information ratios.

Conclusion

Though the signal falls short of alpha status, its dissection yields methodological insights for refining weaker predictors. Inverted or hybridized applications may unlock latent value, but prudence dictates skepticism until empirical rigor confirms out-performance. Again, I would not recommend this signal by itself.

Appendix: Glossary and Mathematical Formulas

Mathematical Formulas

- **OLS Regression:**

$$y_t = \beta_0 + \beta_1 x_t + \epsilon_t, \quad \epsilon_t \sim N(0, \sigma^2)$$

  Minimize $\sum (y_t - \hat{y}_t)^2$. t-stat: $\frac{\hat{\beta}_1}{\mathrm{SE}(\hat{\beta}_1)}$.

- **Quantile Regression:** For quantile τ,

$$Q_y(\tau|x) = \beta_0(\tau) + \beta_1(\tau)x$$

  Minimize $\sum \rho_\tau(y_t - \hat{Q}_t)$, where $\rho_\tau(u) = u(\tau - I(u < 0))$.

- **VAR Model:** For two variables,

$$y_t = a_0 + a_1 y_{t-1} + a_2 x_{t-1} + \epsilon_{y,t}$$

$$x_t = b_0 + b_1 y_{t-1} + b_2 x_{t-1} + \epsilon_{x,t}$$

- **Granger Causality:** Test H_0: Coefficients of lagged x in y equation are zero (F-test).
- **GARCH(1,1):**

$$\sigma_t^2 = \omega + \alpha \epsilon_{t-1}^2 + \beta \sigma_{t-1}^2$$

- **EGARCH:**

$$\ln(\sigma_t^2) = \omega + \beta \ln(\sigma_{t-1}^2) + \alpha \left| \frac{\epsilon_{t-1}}{\sigma_{t-1}} \right| + \gamma \frac{\epsilon_{t-1}}{\sigma_{t-1}}$$

- **TARCH/ZARCH:**

$$\sigma_t^2 = \omega + \alpha \epsilon_{t-1}^2 + \gamma \epsilon_{t-1}^2 I(\epsilon_{t-1} < 0) + \beta \sigma_{t-1}^2$$

- **ADF Test:**

$$\Delta y_t = \alpha + \beta y_{t-1} + \sum \gamma_i \Delta y_{t-i} + \epsilon_t$$

  Test $\beta = 0$ (unit root).
- **IV2SLS:** Stage 1: $\hat{x} = \pi_0 + \pi_1 z$; Stage 2: $y = \beta_0 + \beta_1 \hat{x} + \epsilon$.
- **LASSO:** Minimize $\sum (y - \hat{y})^2 + \lambda \sum |\beta_j|$.
- **Diebold-Mariano Test:** Compare forecast errors; DM stat $\sim N(0,1)$ under H_0: equal MSE.
- **Chow Test:** F-stat for structural break: Compare RSS of full vs. sub-samples.
- **Bootstrap CI:** Resample data B times; compute percentiles of $\hat{\theta}^*$.

STATA CODE

```
/*
Author: Joshua Zayne
Company: Trexquant
Dataset: Interview test dataset (data_analysis_interview.xlsx)
Due date: October 17 2025
Objective:
This script aims to analyze the statistical significance of a 'signal' variable
in predicting the future returns of an underlying security, whose 'close' prices
are provided.
*/
clear all
set more off
set seed 12345
*================================================================================*
* DIRECTORY SWITCHER
```

```
*===============================================================================*
capture cd "C:\Users\ohjos\iCloudDrive\STATA Analysis\Quant Interview"
if _rc != 0 {
    capture cd "F:\iCloudDrive\STATA Analysis\Quant Interview\working data"
    if _rc != 0 {
        display as error "ERROR: Working directory not found."
        exit 198
    }
}
display as result "Current directory: " c(pwd)
log using "optimized_analysis_20251010_Big_v2_updated_17OCT2025.txt", replace text
*===============================================================================*
* STATA ENVIRONMENT SETUP AND DATA IMPORT
*===============================================================================*
local primary_data_path "C:\Users\ohjos\iCloudDrive\STATA Analysis\Quant Interview\data_anal
local secondary_data_path "F:\iCloudDrive\STATA Analysis\Quant Interview\working data\data_a
local tertiary_data_path "F:\iCloudDrive\STATA Analysis\Quant Interview\data_analysis_interv
local data_file_to_import ""
// Check paths
cap confirm file "`primary_data_path'"
if _rc == 0 {
    local data_file_to_import "`primary_data_path'"
    display as text "Data file found in primary location: `data_file_to_import'"
}
else {
    cap confirm file "`secondary_data_path'"
    if _rc == 0 {
        local data_file_to_import "`secondary_data_path'"
        display as text "Data file found in secondary location: `data_file_to_import'"
    }
    else {
        cap confirm file "`tertiary_data_path'"
        if _rc == 0 {
            local data_file_to_import "`tertiary_data_path'"
            display as text "Data file found in tertiary location: `data_file_to_import'"
        }
        else {
            display as error "ERROR: data_analysis_interview.xlsx not found."
            exit 198
        }
    }
}

// Import
if "`data_file_to_import'" != "" {
    import excel using "`data_file_to_import'", sheet("data_analysis_interview") firstrow cl
    display as text "Data imported: " _N " obs, " c(k) " vars."
}
else {
    display as error "Internal error: No data file path resolved."
    exit 198
}
```

```
*===============================================================================*
*  DATA CLEANING AND PREPARATION
*===============================================================================*
capture confirm string variable Date
if _rc == 0 {
    gen date_stata = date(Date, "YMD")
}
else {
    tostring Date, generate(date_str)
    gen date_stata = date(date_str, "YMD")
    drop date_str
}
format date_stata %td
drop Date
label variable date_stata "Stata Date"
count if close == "nan"
if r(N) > 0 {
    local num_nans = r(N)
    drop if close == "nan"
    display as text "`num_nans' 'nan' observations dropped from 'close'."
}
destring close, generate(close_numeric) force
if r(n_mismatched) > 0 {
    display as error "WARNING: `r(n_mismatched)' non-numeric in 'close' set to missing."
}
recast double close_numeric, force
drop close
rename close_numeric close
label variable close "Close Price (Numeric)"
capture confirm string variable signal
if _rc == 0 {
    destring signal, generate(signal_numeric) force
    if r(n_mismatched) > 0 {
        display as error "WARNING: `r(n_mismatched)' non-numeric in 'signal' set to missing
    }
    recast double signal_numeric, force
    drop signal
    rename signal_numeric signal
}
else {
    recast double signal, force
}
label variable signal "Signal Variable"
*===============================================================================*
*Fixing Close Price out of bounds
*===============================================================================*
/*
      | date_st~a |
      |-----------|
1011. | 05apr2017 |
1012. | 06apr2017 |
1350. | 09aug2018 |
```

```
1351. | 10aug2018 |
      +-----------+
On line 1351 the data is wrong. added a zero I think.
On line 1012 that is more tricky... I am just going to drop it.

*/
*drop in 1012
// replace close = 2586.71899 in 1011
//
// replace close = 3112.95313 in 1351
ds
*=============================================================*
*  Sort and prepare data
*=============================================================*
sort date_stata
*=============================================================*
*  Compute log returns (non-panel compatible)
*=============================================================*
gen double daily_return_log_noPanel = log(close / close[_n-1])
gen double future_return_log_noPanel = log(close[_n+1] / close)
label var daily_return_log_noPanel "Daily log return (no panel safe)"
label var future_return_log_noPanel "Future log return (no panel safe)"
*=============================================================*
* Compute simple percentage returns (non-panel compatible)
*=============================================================*
gen double daily_return_noPanel = (close - close[_n-1]) / close[_n-1]
gen double future_return_noPanel = (close[_n+1] - close) / close
label var daily_return_noPanel "Daily % return (no panel safe)"
label var future_return_noPanel "Future % return (no panel safe)"
*=============================================================*
* setup variables for panel data
*=============================================================*
sort date_stata
gen long time_idx = _n
label var time_idx "Contiguous observation index"
gen double Date_num = date_stata
format Date_num %td
label var Date_num "Daily date (numeric Stata date)"
gen byte panel_id = 1
xtset panel_id Date_num
*=============================================================*
*  Compute log returns (panel-compatible)
*=============================================================*
*-------------------------------------------------------------*
gen double daily_return_log = ln(close / L.close)
gen double future_return_log = ln(F.close / close)
*-------------------------------------------------------------*
label var daily_return_log "Daily log return (panel compatible)"
label var future_return_log "Future log return (panel compatible)"
*=============================================================*
* Compute simple percentage returns (panel-compatible)
*=============================================================*
```

```
gen double daily_return = (close - L.close) / L.close
gen double future_return = (F.close - close) / close
label var daily_return "Daily % return (panel compatible)"
label var future_return "Future % return (panel compatible)"
*============================================================*
*  Summary statistics
*============================================================*
summarize signal close daily_return future_return daily_return_log future_return_log
* Extended descriptive summary (adds variance, skewness, kurtosis)
quietly summarize signal close daily_return future_return daily_return_log future_return_log
ds
* Creates synthetic monthly period variable (for reference)
gen Period = tm(2013m2) + _n - 1
*============================================================================*
*: Generating Variables
*============================================================================*
sort date_stata
gen percent_change_Var = (close - close[_n-1]) / close[_n-1]
gen big_move_10 = abs(percent_change_Var) > 0.10
replace big_move_10 = 0 if missing(big_move_10)

gen big_move_15 = abs(percent_change_Var) > 0.15
replace big_move_15 = 0 if missing(big_move_15)
gen big_move_20 = abs(percent_change_Var) > 0.2
replace big_move_20 = 0 if missing(big_move_20)
*============================================================================*
* Graphs
*============================================================================*
*============================================================================*
*twoway (line close date), title("Daily Close Prices")
twoway (line close date), ///
  xtitle("Date") ///
  ytitle("Close Price")
graph export "line close date.png", replace
*============================================================================*
*============================================================================*
*ECONOMETRIC ANALYSIS - OLS WITH HAC STANDARD ERRORS
*============================================================================*
drop if missing(future_return, signal, daily_return)
sort panel_id date_stata
capture drop time_idx
gen long time_idx = _n
tsset panel_id time_idx, daily
newey future_return signal, lag(5)
// Findings: Signal coef 0.0013 (p=0.858), insignificant.
ds
*============================================================================*
* FURTHER ECONOMETRIC ANALYSIS
*============================================================================*
qreg future_return signal, quantile(0.25)
qreg future_return signal, quantile(0.50)
qreg future_return signal, quantile(0.75)
```

```
*===============================================================================*
* Vector Autogression and Granger Cauality
*===============================================================================*
// sort date_stata
// capture drop time_idx
// gen long time_idx = _n
// tsset date_stata
varsoc future_return signal, maxlag(5)
var future_return signal, lags(1)
vargranger
varsoc future_return_log signal, maxlag(5)
var future_return_log signal, lags(1)
vargranger
// Findings: No Granger causality (p>0.4).
ds
*===============================================================================*
* BASIC ORDINARY LEAST SQUARES (OLS) REGRESSION
*===============================================================================*
reg future_return signal
*===============================================================================*
* BASIC  REGRESSION
*===============================================================================*
egen z_signal = std(signal)
egen z_future_return = std(future_return)
reg z_future_return z_signal
reg future_return signal, robust
//
// scatter future_return signal || lfit future_return signal
// xtile signal_decile = signal, n(10)
// by signal_decile: summarize future_return
gen half = (time_idx > _N/2)
reg future_return signal if half==0
reg future_return signal if half==1
binscatter future_return signal, nq(20) line(qfit)
regress signal date_stata
regress close date_stata
pwcorr date_stata signal close, sig
pwcorr signal close date_stata, sig
mvreg signal close = date_stata
mvreg signal date_stata = future_return
// oneway close date_stata
// oneway signal date_stata
// manova signal close = date_stata
// manova date_stata close = signal
// logit signal close
sureg (signal date_stata) (close date_stata)
sureg (date_stata signal) (close signal)
sureg (signal future_return) (close future_return)
regress future_return signal
regress future_return signal close date
sem (close <- signal date) (future_return <- signal close date)
// net install xtqreg2, from("https://raw.githubusercontent.com/joshuaulrich/xtqreg2/main/",
```

```
// xtqreg2 close signal date, quantile(0.1 0.25 0.5 0.75 0.95)
*=============================================================================*
* CONDITIONAL ANALYSIS (MEDIATION & QUANTILE REGRESSION BY DAILY RETURN BINS) - EXPANDED & (
*=============================================================================*
gen double hypothetical_mediator = signal + rnormal() / 2
label variable hypothetical_mediator "Hypothetical Mediator Variable"
drop if missing(future_return, signal, hypothetical_mediator)
sort panel_id date_stata
capture drop time_idx
gen long time_idx = _n
tsset panel_id time_idx, daily
sem (hypothetical_mediator <- signal) (future_return <- hypothetical_mediator signal), vce(l
estat teffects
sum daily_return, detail
local p05_daily = r(p5)
local p10_daily = r(p10)
local p25_daily = r(p25)
local p75_daily = r(p75)
local p90_daily = r(p90)
local p95_daily = r(p95)
gen byte daily_return_bin = .
replace daily_return_bin = 1 if daily_return <= `p05_daily'
replace daily_return_bin = 2 if daily_return > `p05_daily' & daily_return <= `p10_daily'
replace daily_return_bin = 3 if daily_return > `p10_daily' & daily_return <= `p25_daily'
replace daily_return_bin = 4 if daily_return > `p25_daily' & daily_return < `p75_daily'
replace daily_return_bin = 5 if daily_return >= `p75_daily' & daily_return < `p90_daily'
replace daily_return_bin = 6 if daily_return >= `p90_daily' & daily_return < `p95_daily'
replace daily_return_bin = 7 if daily_return >= `p95_daily'
label define daily_return_bin_lbl 1 "Extreme Negative (<p5)" 2 "Large Negative (p5-p10)" 3 "
    4 "Small/No Move (p25-p75)" 5 "Moderate Positive (p75-p90)" 6 "Large Positive (p90-p95)"
label values daily_return_bin daily_return_bin_lbl
tab daily_return_bin
foreach bin in 1 2 3 4 5 6 7 {
    local bin_label : label (daily_return_bin) `bin'
    count if daily_return_bin == `bin'
    if r(N) < 30 {
        display as text "WARNING: Too few observations (`r(N)`) in bin `bin_label'. Skipping
        continue
    }

    display as text "--- Quantile Regressions for Bin: `bin_label' ---"

    qreg future_return signal if daily_return_bin == `bin', quantile(0.25)
    est store q25_bin`bin'
    qreg future_return signal if daily_return_bin == `bin', quantile(0.50)
    est store q50_bin`bin'
    qreg future_return signal if daily_return_bin == `bin', quantile(0.75)
    est store q75_bin`bin'

    qreg future_return c.signal##c.daily_return if daily_return_bin == `bin', quantile(0.50)
    est store q50_int_bin`bin'
}
```

```
*ssc install grqreg, replace
sqreg future_return signal, quantiles(.1 .2 .3 .4 .5 .6 .7 .8 .9) reps(100)
grqreg, ci
graph export "quantile_process_plot.png", replace
gen double abs_daily = abs(daily_return)
sum abs_daily, detail
local abs_p25 = r(p25)
local abs_p50 = r(p50)
local abs_p75 = r(p75)
gen byte abs_return_bin = .
replace abs_return_bin = 1 if abs_daily <= `abs_p25'
replace abs_return_bin = 2 if abs_daily > `abs_p25' & abs_daily <= `abs_p75'
replace abs_return_bin = 3 if abs_daily > `abs_p75'
label define abs_return_bin_lbl 1 "Low Vol" 2 "Medium Vol" 3 "High Vol"
label values abs_return_bin abs_return_bin_lbl
tab abs_return_bin
foreach bin in 1 2 3 {
    local bin_label : label (abs_return_bin) `bin'
    count if abs_return_bin == `bin'
    if r(N) < 30 {
        display as text "WARNING: Too few observations (`r(N)`) in bin `bin_label'. Skipping
        continue
    }

    display as text "--- Quantile Regressions for Abs Bin: `bin_label' ---"

    qreg future_return signal if abs_return_bin == `bin', quantile(0.25)
    est store q25_abs_bin`bin'
    qreg future_return signal if abs_return_bin == `bin', quantile(0.50)
    est store q50_abs_bin`bin'
    qreg future_return signal if abs_return_bin == `bin', quantile(0.75)
    est store q75_abs_bin`bin'

    qreg future_return c.signal##c.daily_return if abs_return_bin == `bin', quantile(0.50)
    est store q50_int_abs_bin`bin'
}
sqreg future_return signal i.daily_return_bin, quantiles(.25 .50 .75) reps(100)
*ssc install estout, replace
esttab q25_bin* q50_bin* q75_bin* using "quantile_results.csv", replace wide plain
ds
*================================================================================*
*DIRECTIONAL ACCURACY FOR "EDGE" (>50% AS SIGNIFICANT)
*================================================================================*
gen byte sign_signal = sign(signal) if signal != 0
label define sign_lbl -1 "Down" 1 "Up"
label values sign_signal sign_lbl
gen byte sign_future = sign(future_return) if future_return != 0
label values sign_future sign_lbl
drop if missing(sign_signal, sign_future) | sign_signal == 0 | sign_future == 0
gen byte match = (sign_signal == sign_future)
summ match
local accuracy = r(mean) * 100
```

```
display "Directional Accuracy: `accuracy'% (edge if >50%)"
prtest match == 0.5  // If p<0.05, significantly >50% (or <50%)
ci proportions match
tab sign_signal sign_future, cell
xtile signal_quart = abs(signal), nq(4)  // Strength bins
tab signal_quart match, row  // Accuracy by strength
graph bar (mean) match, over(signal_quart) ytitle("Accuracy (%)") ///
    title("Directional Accuracy by Signal Strength Quartile") ///
    note("Edge if >50% in any bin")
set seed 12345
gen random_pred = round(runiform(-1,1))  // Simulated random +1/-1
gen random_match = (random_pred == sign_future)
summ random_match
local random_acc = r(mean) * 100
display "Random Guessing Accuracy: `random_acc'% (should be ~50%)"
*==============================================================================*
*  EVENT STUDY ON SIGNIFICANT DAILY MOVES AND SIGNAL EFFICACY
*==============================================================================*
*==============================================================================*
gen byte signal_dir = sign(signal) if signal != 0
label define dir_lbl -1 "Down" 1 "Up"
label values signal_dir dir_lbl
gen byte future_dir = sign(future_return) if future_return != 0
label values future_dir dir_lbl
drop if missing(signal_dir, future_dir) | signal_dir == 0 | future_dir == 0
gen byte dir_match = (signal_dir == future_dir)
summ dir_match
local accuracy = r(mean) * 100
display "Overall Directional Accuracy: `accuracy'% (edge if >50%)"
prtest dir_match == 0.5
ci proportions dir_match
tab signal_dir future_dir, cell
xtile signal_strength_quart = abs(signal), nq(4)
label define strength_lbl 1 "Weak (Q1)" 2 "Medium (Q2)" 3 "Strong (Q3)" 4 "Very Strong (Q4)'
label values signal_strength_quart strength_lbl
tab signal_strength_quart dir_match, row
foreach q in 1 2 3 4 {
    ci proportions dir_match if signal_strength_quart == `q'
    local q_acc: di %4.2f [r(mean)] * 100
    if `q_acc' > 50 {
        display "Q`q' Accuracy: `q_acc'% (>50%, SIGNIFICANT EDGE)"
    }
    else {
        display "Q`q' Accuracy: `q_acc'% (<=50%, NO EDGE)"
    }
}
gen percent_match = dir_match * 100
graph bar (mean) percent_match, over(signal_strength_quart) ytitle("Accuracy (%)") ///
    title("Accuracy by Signal Strength (Edge if >50%)") ///
    note("Q4 shows mild edge") blabel(bar, format(%4.1f))
drop percent_match  // Clean up
ds
```

```
*=============================================================================*
* GARCH MODELS (INCLUDING VARIANTS)
*=============================================================================*

arch future_return signal, arch(1) garch(1) distribution(t)
arch future_return signal, earch(1) egarch(1) distribution(t)
arch future_return signal, tarch(1) arch(1) garch(1) distribution(t)
ds
*=============================================================================*
* COINTEGRATION AND ERROR CORRECTION MODELS (ECM/VECM)
*=============================================================================*
drop if missing(close, signal, future_return)
sort date_stata
capture drop time_idx
gen long time_idx = _n
tsset time_idx, daily
dfuller close, lags(5)
dfuller signal, lags(5)
varsoc future_return signal, maxlag(5)
vecrank future_return signal, lags(5) max
// If rank=1, fit VECM
vec future_return signal, lags(5) rank(1)
reg close signal
predict ecm_resid, resid  // Error term
// ECM: delta_future_return = beta * signal + gamma * L.ecm_resid + lags
gen delta_future = D.future_return
gen delta_signal = D.signal
reg delta_future delta_signal L.ecm_resid L(1/5).delta_future L(1/5).delta_signal
ds
*=============================================================================*
* NONLINEAR TESTS (BDS AND THRESHOLD AUTOREGRESSION)
*=============================================================================*
*ssc install moremata, replace
net sj 21-2 st0636
net install st0636, replace
reg future_return signal
predict resid, resid
bds resid, m(5) eps(0.5 1 1.5)
// Findings: High BDS stats reject i.i.d., nonlinearity present.
gen abs_daily_Nonlinear = abs(daily_return)
sum abs_daily_Nonlinear, detail
local thresh = r(p50)
reg future_return signal if abs_daily_Nonlinear <= `thresh'
reg future_return signal if abs_daily_Nonlinear > `thresh'
// Findings: Signal insignificant in both regimes (p>0.05).
ds
*=============================================================================*
* MACHINE LEARNING FEATURE IMPORTANCE (RANDOM FOREST/LASSO)
*=============================================================================*
*ssc install lassopack, replace
*ssc install rforest, replace
lasso2 future_return signal daily_return abs_daily, lic(ebic) postresults
```

```
mat list e(b)
rforest future_return signal daily_return abs_daily, type(reg) iter(100) seed(12345)
matrix importance = e(importance)
mat list importance
export delimited future_return signal daily_return abs_daily using "ml_data.csv", replace no
ds
*===============================================================================*
*  ADDITIONAL GARCH VARIANTS
*===============================================================================*
// // To fix "last estimates not found", run a simple ARCH model first for initial estimates
// // then extract e(b) as matrix for starting values in complex variants.
// sort panel_id date_stata
// capture drop time_idx
// gen long time_idx = _n
// tsset panel_id time_idx, daily
// // Simple model for inits
// arch future_return signal, arch(1) distribution(t) difficult iterate(100)
// mat init_vals = e(b)  // Extract starting values matrix
// arch future_return signal, aparch(1) garch(1) distribution(t) difficult iterate(100) fron
// arch future_return signal, narch(1) garch(1) distribution(t) difficult iterate(100) from(
// arch future_return signal, aarch(1) distribution(t) difficult iterate(100) from(init_vals
// arch future_return signal, arch(1) saarch(1) garch(1) distribution(t) difficult iterate(1
// arch future_return signal, arch(1) tarch(1) garch(1) distribution(t) difficult iterate(10
// // Findings: If converges, check signal coef in mean; vol terms for asymmetry (e.g., APAR
*===============================================================================*
*  OUT-OF-SAMPLE (OOS) FORECASTING TESTS (DIEBOLD-MARIANO)
*===============================================================================*
*ssc install dmariano, replace
local n_train = int(0.8 * _N)
preserve
keep if _n <= `n_train'
tsset panel_id time_idx
reg future_return L.signal
estimates store unrestricted
predict future_hat_unrest, xb
gen e_unrest = future_return - future_hat_unrest
reg future_return
estimates store restricted
predict future_hat_rest, xb
gen e_rest = future_return - future_hat_rest
restore
preserve
keep if _n > `n_train'
tsset panel_id time_idx
estimates restore unrestricted
gen future_hat_oos_unrest = _b[L.signal] * L.signal + _b[_cons]
gen e_oos_unrest = future_return - future_hat_oos_unrest
estimates restore restricted
gen future_hat_oos_rest = _b[_cons]
gen e_oos_rest = future_return - future_hat_oos_rest
drop if missing(e_oos_unrest, e_oos_rest)
dmariano future_return future_hat_oos_unrest future_hat_oos_rest, crit(MSE)  // MSE
```

```
restore
ds
*===============================================================================*
*  STRUCTURAL BREAK TESTS IN PREDICTABILITY
*===============================================================================*
*ssc install xtbreak, replace
reg future_return signal
estat sbsingle, breakvars(signal)
local break_obs = 800
reg future_return signal if _n < `break_obs'
reg future_return signal if _n >= `break_obs'
ds
*===============================================================================*
* BAYESIAN REGRESSION FOR WEAK SIGNALS
*===============================================================================*
bayes, mcmcsize(10000) burnin(2000): regress future_return signal
// Findings: Posterior mean 0.001, CI includes 0.
*===============================================================================*
* FORECAST COMBINATION/SHRINKAGE (e.g., RIDGE)
*===============================================================================*
*ssc install ridgereg, replace  // Alternative to ridge
forval i=1/5 {
    gen daily_return_l`i' = L`i'.daily_return
}
ridgereg future_return signal daily_return_l1 daily_return_l2 daily_return_l3 daily_return_l
ds
*===============================================================================*
* LONG-HORIZON RETURN PREDICTABILITY
*===============================================================================*
// forval k = 3(3)12 {
//     gen cum_ret_`k'd = future_return
//     forval i = 2/`k' {
//         replace cum_ret_`k'd = cum_ret_`k'd + F`i'.future_return if !missing(F`i'.future_
//     }
//     label var cum_ret_`k'd "Cum Future Return (Days 1-`k')"
//
//     newey cum_ret_`k'd signal, lag(`=`k'-1')
// }
// // Findings: Insignificant across horizons (p>0.6).
*===============================================================================*
*  LONG-HORIZON RETURN PREDICTABILITY
*===============================================================================*
drop if missing(future_return)
sort panel_id date_stata
capture drop time_idx
gen long time_idx = _n
tsset panel_id time_idx, daily
forval k = 3(3)12 {
    gen cum_ret_`k'd = future_return if !missing(future_return)
    forval i = 2/`k' {
        replace cum_ret_`k'd = cum_ret_`k'd + F`i'.future_return if !missing(F`i'.future_ret
    }
```

```
    replace cum_ret_`k'd = . if missing(cum_ret_`k'd)
    label var cum_ret_`k'd "Cum Future Return (Days 1-`k')"

    preserve
    drop if missing(cum_ret_`k'd, signal)
    newey cum_ret_`k'd signal, lag(`=`k'-1')
    restore
}
ds
*==============================================================================*
* SUPERVISED DIMENSION REDUCTION (PLS)
*==============================================================================*
// ssc install pls, replace
//
// pls future_return, xvars(signal daily_return abs_daily) components(2) adjacent
// ssc install plssem, replace
//
// // Define latent variable LV from predictors, predict future_return
// plssem (LV <- signal daily_return abs_daily) (future_return <- LV), modeB(LV) boot(100)
// estat loadings  // Check loadings
// // Findings: If loadings >0.5 for signal, important; path coef LV->future_return insignif
*==============================================================================*
* BOOTSTRAP INFERENCE FOR OLS
*==============================================================================*
bootstrap _b[signal], reps(1000) seed(12345): reg future_return signal
estat bootstrap, percentile bc normal
ds
*==============================================================================*
* SPECTRAL ANALYSIS FOR CYCLES
*==============================================================================*
pergram future_return
// Findings: No clear cycles correlating with signal.
ds
*==============================================================================*
*  CROSS-VALIDATION FOR LASSO
*==============================================================================*
cvlasso future_return signal daily_return abs_daily, nfolds(10) plotcv
ds
*==============================================================================*
* SIGNAL-BASED BACKTESTING (SHARPE RATIO)
*==============================================================================*
gen position = sign(signal)
gen port_ret = position * future_return
summ port_ret
local mean_ret = r(mean)
local sd_ret = r(sd)
local sharpe = (`mean_ret' / `sd_ret') * sqrt(252)
display "Sharpe Ratio: `sharpe'"
// Findings: Sharpe negative (-0.45), no value.
ds
*==============================================================================*
* CORRELATION HEATMAP WITH LAGS
```

```
*==============================================================================*
forval i=1/5 {
    gen signal_l`i' = L`i'.signal
}
pwcorr signal signal_l* future_return
graph matrix signal signal_l* future_return
// Findings: Low corrs (<0.1), no lagged relations.
ds
save "13OCT2025_V2.dta", replace
*==============================================================================*
* ROLLING WINDOW REGRESSION FOR STABILITY
*==============================================================================*
rolling beta = _b[signal] se = _se[signal], window(500) saving(rolling_results, replace): re
preserve
use rolling_results, clear
gen upper = beta + 1.96 * se
gen lower = beta - 1.96 * se
gen p_value = 2 * normal(-abs(beta / se))
twoway (line beta end) (rcap upper lower end), yline(0)
ds
restore
ds
*==============================================================================*
* INSTRUMENTAL VARIABLES (IV) REGRESSION
*==============================================================================*
// // Ensure tsset and regenerate future_return if not found (for re-runs)
// // Sort by time_idx instead of date_stata (which may be dropped or not needed)
// sort panel_id time_idx
// capture drop time_idx
// gen long time_idx = _n
// tsset panel_id time_idx, daily
// capture confirm variable future_return
// if _rc != 0 {
//     gen double future_return = ln(F.close / close) if !missing(close, F.close)
//     label variable future_return "Future Log Return"
// }
//
// drop if missing(future_return, signal)
// ivregress 2sls future_return (signal = L2.signal), robust
// estat firststage
// // Findings: Weak IV (F<10), signal insignificant.
export excel using "mydata.xlsx", firstrow(variables) replace
log close
```