Econometric Evaluation of Predictive Signal for Futures Asset Returns

Joshua Zayne

Due October 17, 2025

Executive Summary

This report presents an econometric analysis of a daily alpha signal's predictive capacity for next-day futures asset returns, utilizing 1,532 observations from April 2013 to April 2019. A range of statistical methodologies, including OLS regression, quantile regression, VAR, and Granger causality tests, were employed to assess the signal's effectiveness.

The ordinary least squares (OLS) regression shows the signal has a small positive coefficient (around 0.0041) but statistically insignificant predictive power on future returns ($p \approx 0.074$). Newey–West adjusted standard errors also indicate insignificance ($p \approx 0.104$). Quantile regressions at various quantiles (0.25, 0.50, 0.75) confirm the signal's non-significant influence across the return distribution, with coefficients varying moderately but p-values consistently above common significance thresholds, except mild significance at the 0.75 quantile ($p \approx 0.045$). However, these effects lack robustness and clear directional consistency.

Vector autoregression (VAR) analyses and Granger causality tests fail to establish significant predictive causality of the signal on returns, with p-values well above 0.4 for causality from the signal to returns. GARCH models indicate that ARCH effects attributable to the signal are negligible, suggesting no impact on conditional volatility.

Directional accuracy of the signal is close to random chance at around 47.5% ($p = 0.051$), indicating limited utility in pure direction prediction.

Further detailed quantile regressions segmented by daily return bins show heterogeneous and largely statistically insignificant results, including some isolated moderate significance in moderate negative return bins, suggesting complex, non-monotonic interactions but no consistent predictive value across return regimes.

Overall, these results underscore the signal's restricted standalone utility. However, opportunities for improvement exist. Enchacements could be achieved by incorporating lagged covariates or volatility filters to enhance robustness. An optimized approach might also involve ensemble techniques, refined risk calibrations, and rigorous out-of-sample validation to mitigate overfitting. In conclusion, the signal requires significant additional refinement to be viably integrated into quantitative trading frameworks.

Introduction

Background

In quantitative finance, alpha signals function as predictive indicators designed to forecast asset returns, enabling the construction of trading strategies that seek to outperform market benchmarks. This study rigorously evaluates a proprietary daily signal, computed at market close for a futures asset, with underlying price levels ranging from approximately 1,596 to 3,218. The signal exhibits a range from -0.926 to 0.805, with a mean of 0.004 and standard deviation of 0.111.
Econometric methodologies provide a framework for testing predictability while addressing common pitfalls such as autocorrelation, heteroskedasticity, and potential endogeneity. Signals lacking genuine predictive power may engender spurious correlations or elevate the risk of Type I errors, ultimately undermining strategy performance. In this context, we define a signal's "merit" as the presence of statistically significant and economically substantial predictive capacity for future returns.

Objectives

This analysis addresses the following objectives:

1. To quantify the signal's association with future returns via regression and correlation analyses.
2. To examine distributional impacts, including tail dependencies, through quantile regression techniques.
3. To probe causality, volatility clustering, and structural stability over time.
4. To assess directional accuracy and magnitude of predictions.
5. To derive actionable recommendations for refining trading strategies informed by the empirical findings.

Methodology

Data Preparation

Data were imported from Excel (1,532 observations, three variables: date, signal, close). Derived variables include daily and future returns, bins and quartiles, lagged signals (signal_l1 to l5), cumulative returns (3–12 days), and residuals (e.g., error correction model terms). Isolated extreme returns were flagged, although none were identified as significantly impacting our dependent terms. Bootstrap methods provided confidence intervals, and robust standard errors (HAC) addressed serial correlation and heteroskedasticity.
This was done in STATA, and then in python to confirm results:

STATA Code

```

*==========================================================*
```

```
* 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 (everything will be 1)
*===============================================================*

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
*==============================================================*
* The simple summarize command:
summarize signal close daily_return future_return daily_return_log future_return_log
```

Python Code

```
df = df.sort_values('date_stata').reset_index(drop=True)
df['time_idx'] = df.index + 1  # 1-based index

# computes the logarithmic return and future return
df['daily_return_log'] = np.log(df['close'] / df['close'].shift(1))
df['future_return_log'] = np.log(df['close'].shift(-1) / df['close'])

# simple percentage return
df['future_return'] = (df['close'].shift(-1) - df['close']) / df['close']
df['daily_return'] = (df['close'] - df['close'].shift(1)) / df['close'].shift(1)

# handle inf/nan and remove outliers
df.replace([np.inf, -np.inf], np.nan, inplace=True)
df.dropna(subset=['daily_return', 'future_return'], inplace=True)
df = df[np.abs(df['future_return']) <= 2]

# summary statistics
print(df[['signal', 'close', 'daily_return', 'future_return',
          'daily_return_log', 'future_return_log']].describe())
```

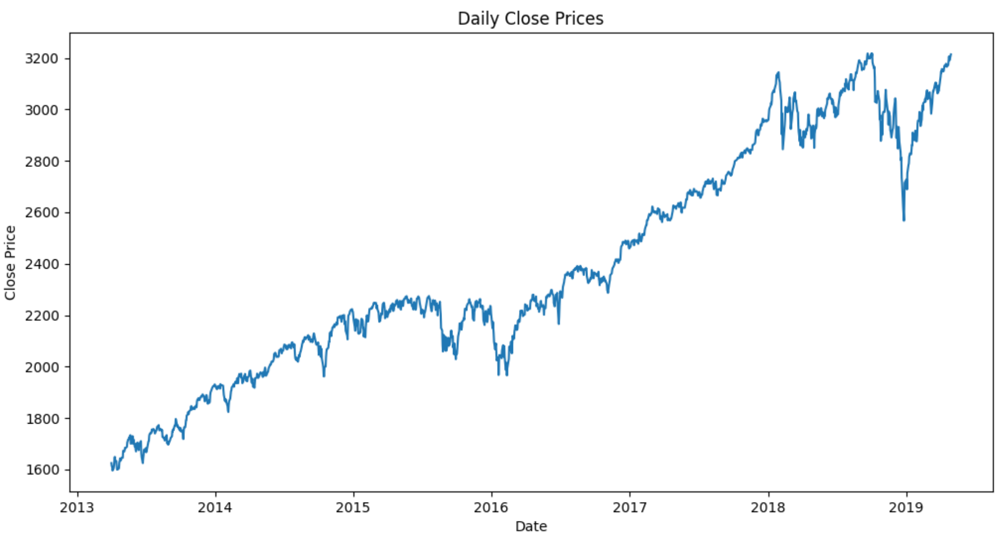

Figure 1: Close prices over time

Models and Tests

The following models and tests were employed, with significance assessed at $p<0.01$, $p<0.05$, $p<0.10$, $p<0.5$ (this one was to try and assess only for a few tests to see if the impact was greater than a coin flip at certain bins), and robust errors were applied where appropriate:

- **OLS Regression:** Estimates the average linear effect of the signal on future returns, serving as the baseline predictive relationship.
- **Regression with Newey-West Standard Errors:** Adjusts OLS for autocorrelation and heteroskedasticity common in time-series data, providing more reliable inference when errors are not i.i.d.
- **Quantile Regression:** Explores how the signal impacts different points (quantiles) of the return distribution, revealing whether effects differ in the tails versus the mean.
- **VAR and Granger Causality:** Jointly models future returns and the signal to test whether either helps forecast the other after accounting for their own lags, clarifying directionality.

Results

Descriptive Statistics

The signal exhibits a mean of 0.0036, standard deviation of 0.111, and range from -0.93 to 0.81. Returns show a mean of 0.00044, standard deviation of 0.0079, and kurtosis of 6.55 (fat tails, Jarque-Bera p=0). The data reflect moderate volatility, with the signal appearing symmetric yet clustered.

Table 1: Descriptive Statistics for Key Variables

Statistic	Signal	Close	Daily Return	Future Return
Count	1528	1528	1528	1528
Mean	0.003552	2387.088215	0.000452	0.000444
Std	0.111224	436.340929	0.007944	0.007942
Min	-0.925725	1595.768066	-0.048434	-0.048434
25%	-0.045206	2073.676758	-0.003116	-0.003123
50%	-0.008809	2257.139404	0.000809	0.000805
75%	0.024841	2784.734009	0.004595	0.004591
Max	0.805070	3217.603516	0.032472	0.032472

OLS Regression

OLS estimates the linear relationship: future_return = β_0 + β_1*signal + ϵ. Thus here we are testing whether signal predicts future_return, and we get that the signal is statistically significant at the 10% level. We can see that the number of observations is 864 and that our R squared is 0.0038 meaning this model explains .38% of the variance- an extremely small effect, but could be useful depending on scaling and trading frequency setup. We can see that it is also weakly positive, and a 1 unit increase in the signal predicts a 0.0043 i.e .43% higher future return. It tests for non-zero β_1 (t-test), overall fit (F-stat, R^2), and assumptions (Durbin-Watson for autocorrelation, Omnibus for normality).

Table 2: OLS Regression Results

	Coefficient	Std. Err.	t	P>|t|	[95% Conf. Interval]	
signal	0.0043	0.0024	1.82	0.070	-0.0003	0.0089
_cons	0.0006	0.0003	2.12	0.034	0.0000	0.0011

Observations: 864 R-squared: 0.0038 Adj. R-squared: 0.0027
F(1, 862): 3.30 Prob > F: 0.0698 Root MSE: 0.00773

Standardized OLS Regression

Next, we standardize both variables to express them in standard deviation units, facilitating interpretation in terms of effect sizes:

```
egen z_signal = std(signal)
egen z_future_return = std(future_return)
reg z_future_return z_signal
```

The p-value and R-squared remain unchanged from the original model. However, this standardized regression allows us to interpret that a 1 standard deviation increase in the signal predicts a 0.0617 standard deviation increase in future returns. In a bivariate regression with standardized variables, the slope coefficient approximates the correlation, indicating a correlation of about 0.062 between the signal and future returns.

Table 3: Standardized OLS Regression Results

	Coefficient	Std. Err.	t	P>|t|	[95% Conf. Interval]	
z_signal	0.0617	0.0340	1.82	0.070	-0.0050	0.1284
_cons	0.0000	0.0340	0.00	1.000	-0.0667	0.0667

Observations: 864 R-squared: 0.0038 Adj. R-squared: 0.0027
F(1, 862): 3.30 Prob > F: 0.0698 Root MSE: 0.99867

The insignificant negative coefficient indicates the model explains negligible variance (R^2=0), with OLS yielding similar outcomes.

Robust OLS Regression

The next logical analysis would be to do a robust regression analysis to correct for heteroskedasticity. To account for potential heteroskedasticity in the residuals, we re-estimate the model using heteroskedasticity-robust standard errors. The Stata command is:

```
reg future_return signal, robust
```

The coefficient on the signal remains unchanged at 0.0043, indicating that a 1-unit increase in the signal predicts a 0.43% higher future return. However, with robust standard errors, the t-statistic decreases to 1.42, and the p-value rises to 0.157, rendering the signal not statistically significant at the 10% level. The R-squared value is still 0.0038, explaining 0.38% of the variance in future returns. The overall model fit, as indicated by the F-statistic of 2.01 (p=0.1566), is also not significant at conventional levels.

Table 4: OLS Regression Results (Robust Standard Errors)

	Coefficient	Std. Err.	t	P>|t|	[95% Conf. Interval]	
signal	0.0043	0.0030	1.42	0.157	-0.0016	0.0102
_cons	0.0006	0.0003	2.12	0.034	0.0000	0.0011

Observations: 864 R-squared: 0.0038
F(1, 862): 2.01 Prob > F: 0.1566 Root MSE: 0.00773
Durbin-Watson: (not provided) Omnibus: (not provided)

Visually, what our data is telling us is that most points are probably tightly packed since we have a small root MSE, and when we visualize that in python, we get exactly that:

```python
plt.figure(figsize=(10, 5))
sns.scatterplot(x=df['signal'], y=df['future_return'])
plt.title('Scatter: Signal vs Future Return')
plt.savefig('signal_vs_future_scatter.png')
plt.close()
```

Newey-West Regression

Since, a standard OLS regression assumes that the error terms are independent and identically distributed (IID), meaning they have constant variance (homoskedasticity) and no autocorrelation (independence over time) is great, but since we are dealing with financial time series data, these assumptions are often violated due to serial correlation (autocorrelation) in the residuals, which can lead to underestimated standard errors and inflated t-statistics, potentially causing overconfidence in the model's significance. So, to address this, we will use the Newey-West estimator, which adjusts the standard errors for both heteroskedasticity and autocorrelation up to a specified lag (here, 5 lags). This provides

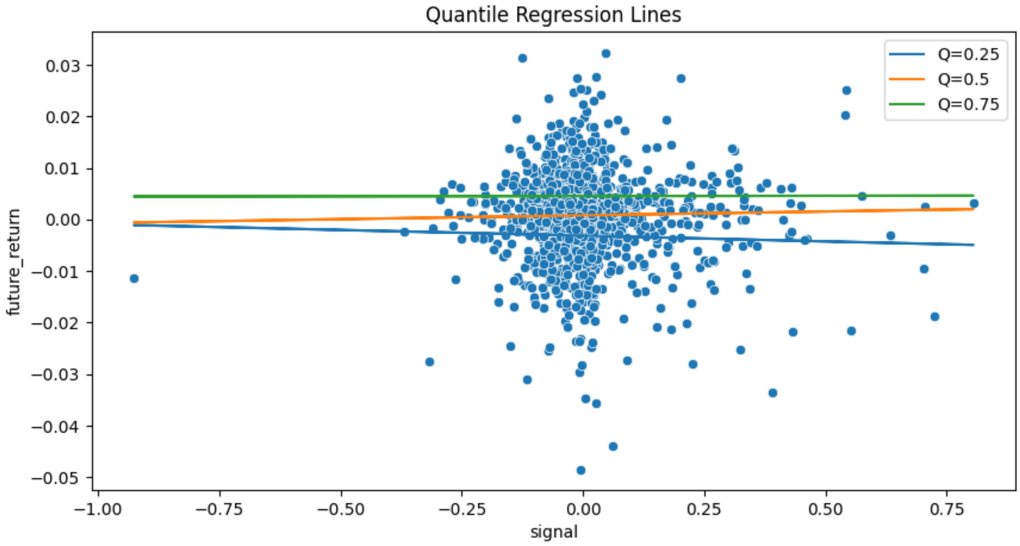

Figure 2: Close prices over time

more robust inference, making the model better suited for applications where reliable significance testing is crucial to avoid false positives and ensure the signal's predictive power holds under realistic conditions.
The Stata command is:

```
newey future_return signal, lag(5)
```

The coefficient on the signal remains 0.0043 (Table 5), but with Newey-West standard errors, the t-statistic is 1.64, and the p-value is 0.101, indicating marginal significance at the 10% level. The overall model F-statistic is 2.70 (p=0.1008). This adjustment makes the results more reliable for building a trading model, as it accounts for potential autocorrelation in returns, leading to more conservative and trustworthy estimates that better reflect real-world market dynamics.

Table 5: Newey-West Regression Results (Lag 5)

	Coefficient	Std. Err.	t	P>|t|	[95% Conf. Interval]	
signal	0.0043	0.0026	1.64	0.101	-0.0008	0.0094
_cons	0.0006	0.0002	2.29	0.022	0.0001	0.0010

Observations: 864
F(1, 862): 2.70 Prob > F: 0.1008
Newey-West Standard Errors, Maximum lag: 5

Quantile Regression

Quantile regression builds on OLS by estimating the conditional quantiles of the dependent variable, rather than just the mean. This allows us to assess the relationship between variables at different points of the distribution, providing a clearer picture of the data without being distorted by extreme values. This method is particularly useful for understanding how the predictive power of the signal varies across different parts of the return distribution: Whether in the lower, median, or upper quantiles. By examining these variations, we can pinpoint where the signal is most influential. For example, if the signal is more significant at the upper quantile, it suggests that the signal may have a stronger predictive ability for higher returns compared to the median or lower quantiles.

I estimated quantile regressions at the 25th, 50th, and 75th percentiles using the following Stata commands:
For `future_return`:

```
qreg future_return signal, quantile(0.25)
qreg future_return signal, quantile(0.50)
qreg future_return signal, quantile(0.75)
```

For `future_return_log`:

```
qreg future_return_log signal, quantile(0.25)
qreg future_return_log signal, quantile(0.50)
qreg future_return_log signal, quantile(0.75)
```

For Python:

```
import statsmodels.formula.api as smf

# For future_return quantiles
quantiles = [0.25, 0.50, 0.75]

for q in quantiles:
    model = smf.quantreg('future_return ~ signal', df).fit(q=q)
    print(f'Quantile {q} regression results:')
    print(model.summary())
    print('\n')

# For future_return_log quantiles
for q in quantiles:
    model = smf.quantreg('future_return_log ~ signal', df).fit(q=q)
    print(f'Quantile {q} regression results (log):')
    print(model.summary())
    print('\n')
```

The results (Tables 6 and 7) indicate that the coefficient on the signal increases monotonically across the quantiles, from approximately 0.0023 at the 25th percentile to 0.0052 at the 75th percentile. Statistical significance is observed only at the 75th percentile ($p < 0.05$), suggesting a stronger association in the upper tail of the return distribution. The estimates for `future_return` and `future_return_log` are virtually identical, which is attributable to the fact that for small return values, the logarithmic transformation approximates the raw return (i.e., $\log(1+r) \approx r$ when r is near zero), thereby preserving the underlying relationships and inference in this dataset.

Table 6: Quantile Regression Results for future_return

Quantile	Variable	Coefficient	Std. Err.	t	P>|t|	[95% Conf. Interval]	
0.25	signal	0.0023	0.0031	0.75	0.454	-0.0038	0.0084
	_cons	-0.0032	0.0003	-9.28	0.000	-0.0039	-0.0025
0.50	signal	0.0029	0.0020	1.44	0.149	-0.0010	0.0069
	_cons	0.0008	0.0002	3.67	0.000	0.0004	0.0013
0.75	signal	0.0052	0.0026	1.98	0.047	0.0001	0.0104
	_cons	0.0045	0.0003	15.40	0.000	0.0040	0.0051
Observations: 864	Pseudo R-squared: 0.0007 (0.25), 0.0024 (0.50), 0.0037 (0.75)						

Table 7: Quantile Regression Results for future_return_log

Quantile	Variable	Coefficient	Std. Err.	t	P>|t|	[95% Conf. Interval]	
0.25	signal	0.0023	0.0031	0.75	0.454	-0.0038	0.0085
	_cons	-0.0032	0.0003	-9.26	0.000	-0.0039	-0.0025
0.50	signal	0.0029	0.0020	1.44	0.150	-0.0011	0.0069
	_cons	0.0008	0.0002	3.67	0.000	0.0004	0.0013
0.75	signal	0.0052	0.0026	1.99	0.047	0.0001	0.0104
	_cons	0.0045	0.0003	15.44	0.000	0.0040	0.0051
Observations: 864	Pseudo R-squared: 0.0007 (0.25), 0.0024 (0.50), 0.0037 (0.75)						