Empirical Analysis of the S&P 500 Index and the Industrial Production Index, 1991-2001

1948853

02/04/2025

Introduction

Between 1991 and 2001, the U.S. economy experienced steady growth alongside transformative technological advancements and economic globalisation. While financial markets remained broadly stable, this period also encompassed the dot-com bubble—a phase of significant market exuberance and subsequent correction. These dynamics provide a unique context to investigate the relationship between industrial production and the stock market, particularly how long-run equilibrium persists amid short-term speculative distortions. Using a Vector Error Correction Model (VECM), this analysis examines the interplay between the Industrial Production Index and the S&P 500 Index.

1 Literature Review

The relationship between the financial markets and real economy is well-established. Fama [1990] highlights the role of equity markets as leading indicators of economic activity. Underpinning many of these empirical findings is the cointegration framework developed by Engle and Granger [1987], which provides a methodological foundation for modelling long-run relationships between nonstationary financial and macroeconomic variables. This framework shows that even if individual series are nonstationary, a linear combination of them can be stationary, thereby establishing the basis for error correction models that capture both short-run dynamics and long-run equilibria.

A key distinction between financial markets and real economic activity lies in their inherent adjustment speeds. Financial markets are relatively frictionless due to negligible transaction costs, allowing for rapid responses to new information. In contrast, adjustments in industrial production are often hampered by inherent frictions such as time-to-build constraints and convex adjustment costs, which prevent quick changes in capacity [Brooks, 2001].

Kim [2003]'s application of Johansen's cointegration method, demonstrating a positive link between the S&P 500 and industrial production, provides a direct methodological precedent for this research. Furthermore, Choi et al. [1999] examined the link between real stock returns and industrial production growth across G7 countries, finding evidence of long-run equilibrium but noting limitations in forecasting performance. This study builds upon these foundations by investigating these dynamics focused on the run up of the dot-com bubble, aiming to provide insights into how speculative market conditions can influence the long-term relationship between stock indices and industrial production, and will further contribute by addressing methodological concerns that are often overlooked such as rigorous stationarity testing and heteroskedasticity-robust inference.

2 Data

2.1 Data Overview

The dataset includes monthly observations of the S&P 500 closing prices from Yahoo Finance and the Industrial Production Index (IPI) from FRED, spanning from January 1991 to January 2001. In cases where S&P 500 data was unavailable for a specific IPI date, the closest available date was used instead. Both series underwent log transformation to stabilise variance and facilitate percentage change interpretation, following Choi et al. [1999]. Unless otherwise specified, any reference to these series pertains to their log-transformed versions. The S&P 500 represents the top 500 leading companies in the United States, while the IPI serves as an aggregate measure of industrial output.

Year	S&P 500 Annual Change %	IPI Annual Change %
1991	27.77	1.18
1992	4.42	4.68
1993	7.14	3.00
1994	-1.33	6.54
1995	34.16	2.39
1996	19.33	6.62
1997	31.67	8.09
1998	26.07	3.21
1999	19.64	4.88
2000	-9.27	1.02

Table 1: Year-by-Year Annual Returns for S&P 500 and IPI (1991–2001)

Over the sample period, as seen in Table 1 and Figure 1, the S&P 500 grew fast with significant annual fluctuations, as returns ranged from -9.27% in 2000 to 34.16% in 1995. The IPI, in contrast, showed more slow and stable growth, peaking at 8.09% in 1997 and reaching a low of 1.02% in 2000.

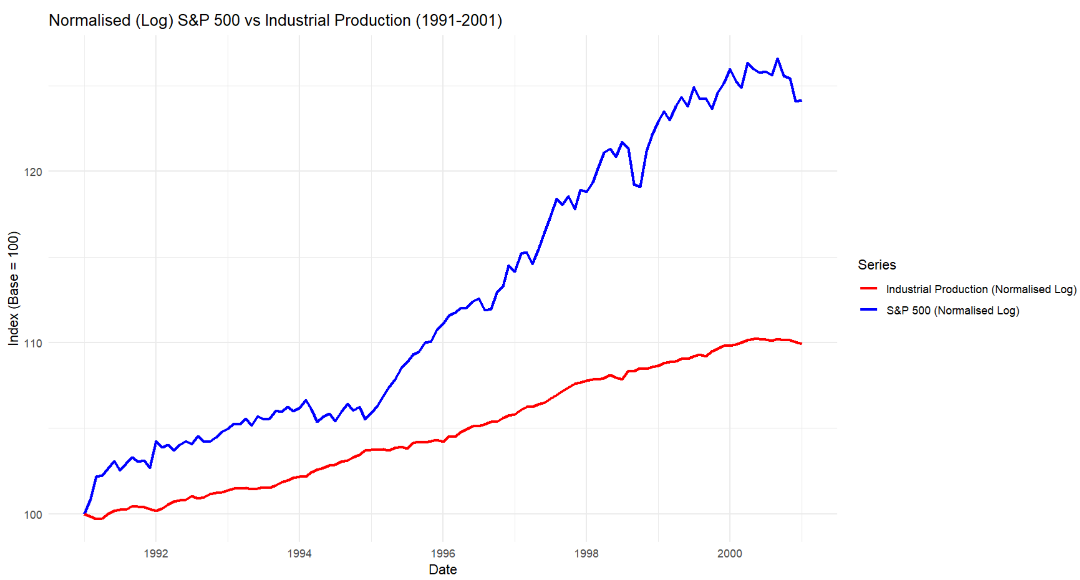


Figure 1: Normalised Logged Series for 1991-2001

2.2 Stationarity Testing

Testing the stationarity of the two series is critical to avoid spurious regression results when non-stationary time series are analysed.

Therefore, to determine the presence of a unit root in either the IPI or S&P 500 series, the Augmented Dickey–Fuller (ADF) test, established in [Dickey and Fuller, 1979], was applied under three specifications: without a constant ("None"), with an intercept ("Intercept"), and with an intercept and linear trend ("Trend"), with the results in Table 2. The findings reveal that the level data for both series fail to reject the null hypothesis of a unit root at conventional significance levels (5% and 1%). Upon first-differencing, however, the test statistics become significantly negative at the 1% significance level, thereby indicating that both series are integrated of order one.

Table 2: ADF Test Results

	Levels			First Differences		
Variable	**None**	**Drift**	**Trend**	**None**	**Drift**	**Trend**
IPI	6.482	-0.318	-1.65	-4.465***	-7.401***	-7.337***
S&P 500	3.46	-0.511	-1.606	-7.134***	-8.11***	-8.076***

Notes: All variables are converted into natural logarithm. ***, **, * denote significance at 1%, 5%, and 10% levels respectively. ADF test includes 1 lag based on AIC criterion. Case *'None'* includes neither an intercept nor a trend in the test specification; case *'Drift'* includes an intercept but no trend; and case *'Trend'* includes both an intercept and a trend. The critical values at 1%, 5%, and 10% significance levels are -2.58, -1.95 and -1.62 (*'None'*), -3.46, -2.88 and -2.57 (*'Drift'*) and -3.99, -3.43 and -3.13 (*'Trend'*) from Hamilton [1994].

The lag structure for the ADF tests was determined using the Akaike Information Criterion (AIC), which indicated a lag length of 1 most appropriate. Despite this relatively low lag order, a Ljung-Box test confirmed the absence of serial correlation in most residuals (see Table A1). The sole exception was observed in the first-differenced IPI series when modelled without a constant or trend. This exception suggests that including a constant term is necessary for stationarity for the differenced IPI.

The Phillips–Perron (PP) and Kwiatkowski–Phillips–Schmidt–Shin (KPSS) tests were also conducted (see Tables 3 and 4). The PP test indicates stationarity in first-differences for both variables under all specifications, while the KPSS test indicates non-stationarity under the "Trend" specification for the IPI first-difference. However, considering the results from the PP and ADF tests suggesting stationarity in the first-differences, the KPSS results under the "Trend" specification may reflect sensitivity to deterministic trends rather than a genuine lack of stationarity.

Table 3: Phillips-Perron Test Results

	Levels			First Differences		
Variable	**None**	**Intercept**	**Trend**	**None**	**Intercept**	**Trend**
IPI	7.334	-0.042	-2.306	-8.091***	-11.003***	-10.942***
S&P 500	3.897	-0.657	-1.556	-10.637***	-11.74***	-11.685***

Notes: All variables are converted into natural logarithm. ***, **, * denote significance at 1%, 5%, and 10% levels respectively. Phillips-Perron Z_tau test statistics are reported using short-term lag selection (4 lags). Case *'None'* includes neither an intercept nor a trend; case *'Intercept'* includes an intercept but no trend; and case *'Trend'* includes both an intercept and a trend. The critical values at 1%, 5%, and 10% significance levels are -2.58, -1.95 and -1.62 (*'None'*), -3.46, -2.88 and -2.57 (*'Intercept'*) and -3.99, -3.43 and -3.13 (*'Trend'*).

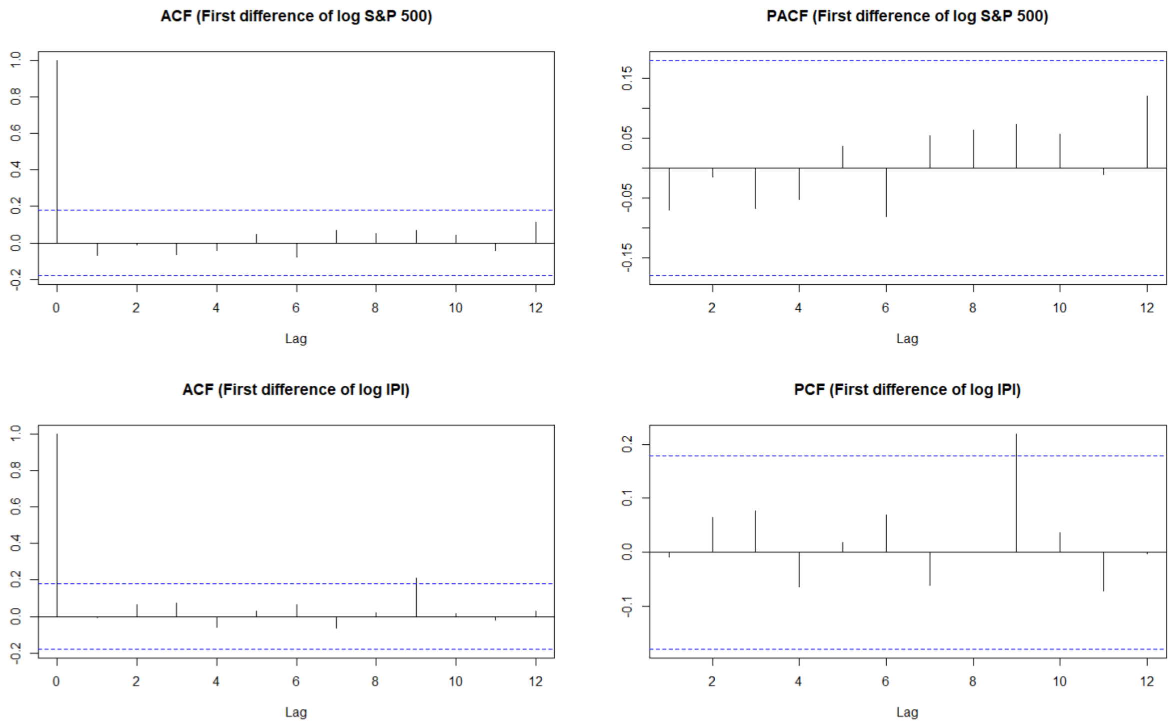
Table 4: KPSS Test Results

	Levels		First Differences	
Variable	**Intercept**	**Trend**	**Intercept**	**Trend**
IPI	2.513***	0.341***	0.213*	0.205**
S&P 500	2.453***	0.465***	0.131*	0.141*

Notes: All variables are converted into natural logarithm. ***, **, * denote significance at 1%, 5%, and 10% levels respectively. KPSS test statistics are reported using the short-term lag selection of 4. Note that for KPSS tests, the null hypothesis is that the series is stationary (rejection indicates non-stationarity). Case *'Intercept'* corresponds to testing stationarity around a level; and case *'Trend'* corresponds to testing stationarity around a deterministic trend.

To ensure robustness, alternative lag lengths were tested with the ADF test ($T^{0.25}(3)$ (as suggested by Mills and Markellos [2008]), 6 lags (bi-yearly), and 12 lags (yearly)). While first differences remained stationary at lower lags, non-stationarity emerged at 12 lags (Table A3). However, given general consensus across the lower lag ADF, PP and KPSS tests, we can justifiably assume both series are I(1).

To further assess the presence of autocorrelation in the first-differenced IPI series, as initially identified by the Ljung-Box Test on the residuals of the ADF Test with 1 lag (see Table A1), the autocorrelation functions depicted in Figure 2 can be inspected. A notable spike at lag 9 is evident in the differenced IPI series, which may account for the autocorrelation detected in the Ljung-Box Test.



Notes: The blue horizontal lines represent 95% confidence intervals. Autocorrelation coefficients extending beyond these lines are statistically significant at the 5% level.

Figure 2: Autocorrelation Functions

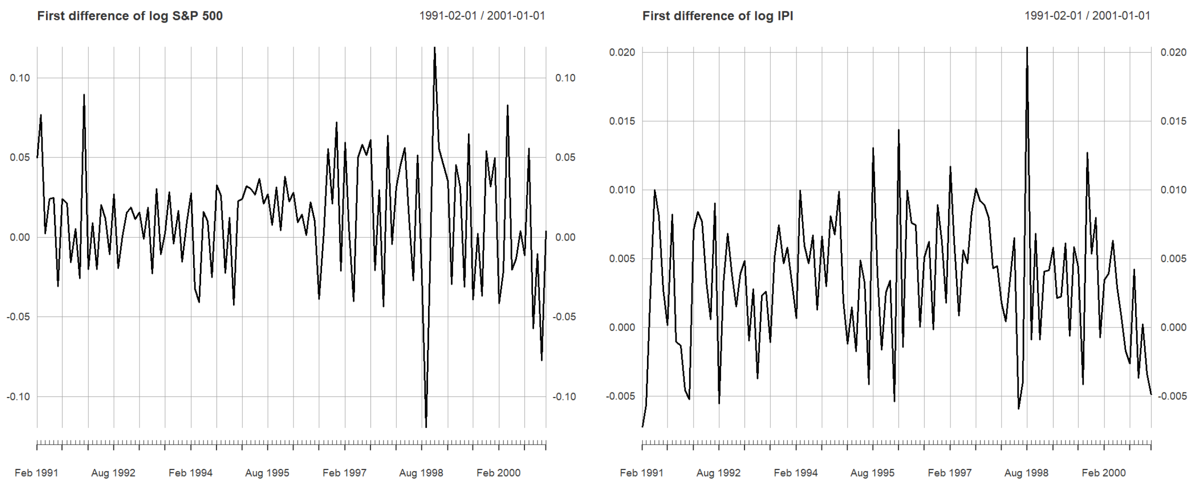


Figure 3: First-Differenced Plots

Importantly, the absence of autocorrelation in the constant and trend specification suggests that the observed autocorrelation at lag 9 potentially stems from an omitted deterministic component rather than a true stochastic dependence. Supporting this interpretation, the first-differenced series in Figure 3 reveals a pattern of fluctuation around a value above zero, indicating the presence of drift in the series. A t-test revealed a statistically significant non-zero mean in the first-differenced IPI series and also the S&P series.

In summary, this stationarity testing supports that both the S&P 500 and IPI series are integrated of order 1. However, a constant should be included specifically to account for the suggested drift in the first-differenced IPI series which also addresses autocorrelation that appears when a constant is omitted.

3 Methodology

3.1 Introduction and Overview

This analysis will use the Vector Error Correction Model (VECM). The strength of the VECM lies in its ability to separate long-run equilibrium relationships from short-run dynamics. It provides an error correction term that measures the speed of adjustment towards equilibrium following exogenous shocks.

3.2 Comparison with Alternative Methodologies

Alternative methods were considered but rejected. Vector Autoregression (VAR) models in levels produce inconsistent coefficient estimates when applied to non-stationary series with cointegration. GARCH captures conditional heteroskedasticity but is aimed at volatility analysis, therefore would fail to address the mean-reverting properties essential to this research. Given that both series are confirmed as I(1) and a theoretical expectation of cointegration, the VECM represents the most appropriate methodology.

3.3 Cointegration Testing

To establish the presence of a long-run equilibrium relationship, Johansen's cointegration test was conducted, with the results in Table 5. The null hypothesis of no cointegration ($r = 0$) is strongly rejected, as the trace statistic (50.74) exceeds the 1% critical value (24.60), providing strong evidence for at least one cointegrating relationship. The test for at most one cointegrating vector ($r \leq 1$) fails to reject the null, as the trace statistic (6.31) is below the 10% critical value (7.52). Since the rank is determined as $r = 1$, this confirms the presence of a single cointegrating vector.

Table 5: Johansen Cointegration Test Results

No. of Cointegrating Vectors	Trace Statistic	10%	5%	1%	Eigenvalue
$r \leq 1$	6.31	7.52	9.24	12.97	0.0516
$r = 0$	50.74***	17.85	19.96	24.60	0.3116

Notes: ***, **, * denote significance at 1%, 5%, and 10% levels respectively. Test conducted with lag order K = 2, transitory specification with constant in cointegration, and N = 121 observations. Variables included are the natural logarithm of the S&P 500 Index (LC) and the natural logarithm of the Industrial Production Index (ID).

The Johansen test produces cointegrating vectors from which we can derive the long-run equation:

$$\mathrm{LC}_t = -6.991 + 3.226\mathrm{ID}_t + \varepsilon_t, \tag{1}$$

where LC_t is the logarithm of S&P 500 index, ID_t is the logarithm of the Industrial Production Index, and ε_t represents deviations from the long-run equilibrium. This relationship suggests that a 1% increase in industrial production is associated with approximately a 3.226% increase in the S&P 500 index in the long run. The elasticity exceeding 1 may reflect the forward-looking nature of equity markets, aligning with Fama [1990]'s argument that equity prices anticipate future economic activity. Between 1991 and 2001, significant price-to-earnings ratios - decoupled from fundamental earnings growth - led equity indices to react disproportionately to modest improvements in economic indicators [Shiller, 2000]. For example, Kim [2003] estimated an industrial production coefficient of 1.72% between 1974 and 1998. By incorporating a longer timespan, this analysis smooths out the anomalous market behaviour observed during the dot-com bubble. This suggests that the higher coefficient may be attributable to this specific period's market dynamics. Consequently, the estimated long-run relationship appears more pronounced than in periods with stronger ties between economic fundamentals and equity prices.

3.4 Model Specification and Lag Selection

With cointegration established, the VECM can be specified. Both the Akaike Information Criterion (AIC) and the Bayesian Information Criterion (BIC) suggest 1 lag with a constant for the VAR representation (VECM(0)). However, a lag order of 2 for the VAR (corresponding to VECM(1)) was selected for several reasons.

First, the Johansen Procedure implementation in R requires a minimum VECM lag order of 1 [Johansen, 1995]. Second, even when information criteria favour a zero-lag specification, a VECM with no lags inherently excludes short-run dynamics, potentially omitting key aspects of the system's behaviour. Third, the lag orders suggested by the information criteria were sensitive to model specification—suggesting a lag order of 3 when the constant term was removed. To ensure specification rigour, a stepwise lag reduction approach was implemented, beginning with the higher lag order (K=3) and testing it against the alternative (K=2). This no statistically significant improvement from the additional lag parameters (LR statistic = -4.51, p-value = 1.00). This finding was reinforced by substantial improvements in information criteria metrics when moving to the more parsimonious specification, as evidenced by 'ΔAIC = 12.51' and 'ΔBIC = 23.52'. Given these results, VECM(1) was the chosen specification.

While Otero and Smith [2000] recommend 12 lags for monthly data to prevent spurious cointegration findings, this approach would substantially reduce degrees of freedom in the sample of 121 observations, potentially compromising estimation precision. Furthermore, diagnostic testing (Section 3.5) confirms the absence of residual autocorrelation with VECM(1), indicating that additional lags are unnecessary for capturing the period's temporal dynamics.

3.5 Model Diagnostics

Residual diagnostics confirm the viability of the VECM(1) specification. Table 6 presents these results.

Table 6: VECM Diagnostic Tests

Test	Chi-Squared	df	p-value
Portmanteau Test	48.15	58	0.818
JB-Test	3.34	4	0.503
Skewness	1.16	2	0.559
Kurtosis	2.17	2	0.337
ARCH Test	84.99	45	0.000

Notes: Asymptotic Portmanteau test for residual autocorrelation with 16 lags. Multivariate Jarque-Bera test (JB) with separate components for Skewness and Kurtosis. ARCH-LM test with 5 lags for conditional heteroskedasticity. All tests applied to VECM(1) residuals.

The Portmanteau test rejects residual autocorrelation (p-value = 0.818), consistent with earlier Ljung-Box results for the ADF test residuals (Table A1) where autocorrelation was resolved by including a constant term. Normality tests (JB, Skewness, and Kurtosis) indicate that the residuals are normally distributed. However, the ARCH test reveals significant conditional heteroskedasticity (p-value <0.001), necessitating heteroskedasticity-robust inference. To address heteroskedasticity concerns, the HC3 heteroskedasticity-consistent covariance matrix estimator developed by MacKinnon and White [1985] was implemented in Section 4's estimation to provide robust standard errors.

4 Empirical Estimation Results and Discussion

4.1 VECM Estimation Results

The VECM results in Table 7 reveal several important insights about the dynamics between the S&P 500 and industrial production. The error correction terms (ECT) for both equations are negative and statistically significant at the 0.1% level, confirming the existence of a long-run equilibrium relationship between the two series as identified in the cointegration analysis.

The ECT coefficient of -0.04 for the S&P 500 equation indicates that approximately 4% of any disequilibrium is corrected each month, while the smaller coefficient of -0.007 for the IPI equation suggests a slower adjustment rate of about 0.7% per month.

Table 7: VECM Estimation Results

	ΔLC (S&P 500)	ΔID (IPI)
Error Correction Term	-0.0400^{***}	-0.0070^{***}
	(0.0093)	(0.0015)
ΔLC_{t-1}	-0.0647	0.0216
	(0.0963)	(0.0120)
ΔID_{t-1}	-1.4140	-0.0058
	(1.0234)	(0.1121)
R^2	0.1508	0.3513
Adj. R^2	0.1289	0.3346
Num. obs.	119	119
RMSE	0.0349	0.0048

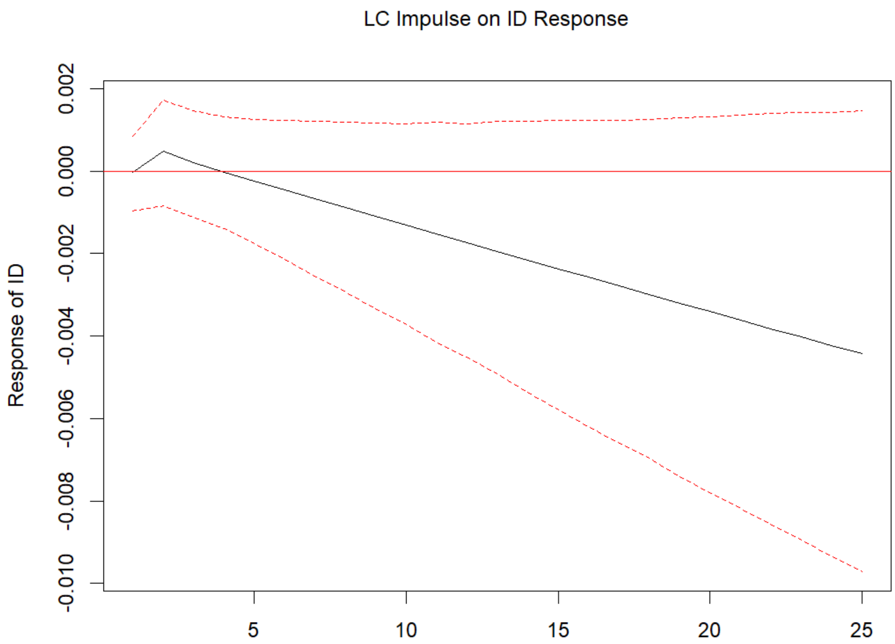
Notes: ***, **, * denote significance at 1%, 5%, and 10% levels respectively. LC refers to log S&P 500; ID refers to log Industrial Production Index. Standard errors in parentheses are HC3 heteroskedasticity-consistent estimates based on MacKinnon and White (1985).

Using the error correction coefficients, the half-life of a shock is approximately 17 months for the S&P 500 Index and 99 months for the Industrial Production Index. This disparity in adjustment speed could reflect fundamental differences in structural adjustment mechanisms between the financial sector and the real economy, where industrial production has inherent frictions as outlined in Brooks [2001].

The R^2 values indicate that the model explains about 15% of the variation in S&P 500 returns and 35% of the variation in industrial production growth. Although both variables are influenced by various factors, the higher explanatory power for industrial production suggests that, within this framework, real economic activity exhibits a more structured pattern than financial market returns, which may subject to a wider range of influences.

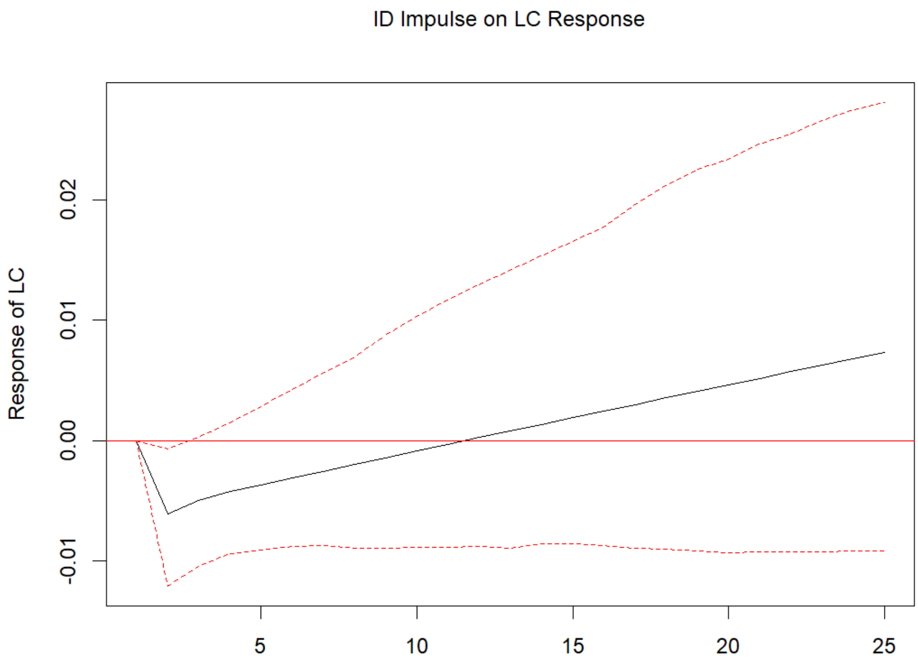
4.2 Impulse Response Analysis

To assess short term dynamics impulse response functions have been generated. Figure 4 shows how Industrial Production reacts to a shock in S&P 500. Although the point estimate (black line) turns negative after the initial period, the confidence intervals consistently include zero, therefore we cannot statistically reject a zero effect. By contrast, in Figure 5, there is a brief dip below zero in the second period where the confidence band departs from 0, therefore suggesting a significant short-run negative effect. Over the subsequent periods, the response turns positive, but again the intervals straddle zero, indicating the stock-market effect is not significantly different from zero in the medium to long term. It is important to note that these IRFs were generated without heteroskedasticity-robust standard errors due to operational complexity in R. The loss of statistical significance for the IPI lag in the S&P 500 model—observed in the robust Table 7 compared to the non-robust results (see Table A2)—suggests that the confidence intervals in the robust model would also fail to statistically reject a zero effect for the IPI IRF.



Notes: Red dotted line = bootstrapped 95% confidence intervals, 1000 runs. LC refers to log S&P 500; ID refers to log Industrial Production Index.

Figure 4: Impulse Response of Industrial Production to a Shock in S&P 500



Notes: Red dotted line = bootstrapped 95% confidence intervals, 1000 runs. LC refers to log S&P 500; ID refers to log Industrial Production Index.

Figure 5: Impulse Response of S&P 500 to a Shock in Industrial Production

4.3 Dynamic Behaviour

Table 8: Variance Decomposition

	Months	S&P 500	IPI
S&P 500	1	100	0
	6	98.11	1.89
	12	98.71	1.29
	24	97.42	2.58
	36	89.20	10.80
	48	73.38	26.62
IPI	1	0	100
	6	0.37	99.63
	12	2.75	97.25
	24	11.74	88.26
	36	20.52	79.48
	48	27.45	72.55

Notes: The table shows the percentage of forecast error variance that is attributed to the S&P 500 or IPI.

Table 8 displays the variance decomposition for the VECM model. This shows us how the stock market and industrial production index respond to shocks, by 'decomposing' how much of the forecast error variance (FEV) we can attribute to either variable. The S&P 500 Index maintains near-complete exogeneity throughout the first 24 months (97-100% self-explained variance). Thereafter, the IPI accelerates in influence rising from 2.58% at month 24 to 26.62% by month 48.

Conversely, the IPI displays a more consistent rate of integration of responsivity to S&P 500 innovations. By month 24, 11.74% of its variance is explained by the S&P 500 and by month 48, 27.45%.

5 Conclusion and Future Research

This analysis provides evidence of a long-run equilibrium relationship between the S&P 500 and Industrial Production Index where a 1% increase in industrial production is associated with a 3.226% increase in the S&P 500 index. This elasticity is substantially higher than the 1.72% effect found by Kim [2003] for 1974-1998, suggesting that while Kim's analysis partially captured the nascent stages of the dot-com phenomenon, the focused timeframe in this analysis reveals more amplified valuation dynamics that characterised the bubble cycle.

The variance decomposition analysis reveals temporal patterns supporting Shiller [2000]'s irrational exuberance thesis. The S&P 500 maintained substantial exogeneity through month 24 before gradually acknowledging the influence of industrial production—a significant departure from Kim [2003]'s findings where industrial production explained 6.75% of S&P 500 variance by month 12 compared to 1.2%. This extended decoupling between market valuations and economic fundamentals provides quantitative evidence of the psychological detachment characteristic of bubble formations.

The asymmetric adjustment mechanism in the error correction model further quantifies this disconnect. With the S&P 500 adjusting at -0.04 compared to IPI's -0.007, financial markets moved toward equilibrium nearly six times faster than industrial production, yet still required a 17-month half-life. This combined with the model's explanatory power disparity (15% for S&P 500 versus 35% for IPI), may align with how the dot-com era market was heavily driven by psychological factors—creating conditions for both explosive growth and subsequent correction.

Future research should incorporate time-variant parameters into the VECM as the current assumption of time-invariance is restrictive and may not reflect reality due to shifting market and valuation dynamics during this period. Furthermore, incorporating additional macroeconomic variables, resolving methodological tensions in lag selection and investing structural breaks would enhance robustness.

References

Ms. Petya Koeva Brooks. Time-To-Build and Convex Adjustment Costs. IMF Working Papers 2001/009, International Monetary Fund, January 2001. URL https://ideas.repec.org/p/imf/imfwpa/2001-009.html.

J. Choi, Shmuel Hauser, and Kenneth J. Kopecky. Does the stock market predict real activity? time series evidence from the g-7 countries. *Journal of Banking and Finance*, 23:1771–1792, 1999. doi: 10.1016/S0378-4266(99)00020-5.

D. Dickey and Wayne Fuller. Distribution of the estimators for autoregressive time series with a unit root. *JASA. Journal of the American Statistical Association*, 74, 06 1979. doi: 10.2307/2286348.

Robert F. Engle and C. W. J. Granger. Co-integration and error correction: Representation, estimation, and testing. *Econometrica*, 55(2):251–276, 1987. ISSN 00129682, 14680262. URL http://www.jstor.org/stable/1913236.

Eugene F Fama. Stock Returns, Expected Returns, and Real Activity. *Journal of Finance*, 45(4):1089–1108, September 1990. URL https://ideas.repec.org/a/bla/jfinan/v45y1990i4p1089-1108.html.

James Douglas Hamilton. *Time Series Analysis*. Princeton University Press, 1994. URL https://www.worldcat.org/title/time-series-analysis/oclc/1194970663&referer=brief_results.

Soren Johansen. *Likelihood-Based Inference in Cointegrated Vector Autoregressive Models*. Number 9780198774501 in OUP Catalogue. Oxford University Press, 1995. ISBN ARRAY(0x71b977b0). URL https://ideas.repec.org/b/oxp/obooks/9780198774501.html.

Ki-Ho Kim. Dollar exchange rate and stock price: evidence from multivariate cointegration and error correction model. *Review of Financial Economics*, 12:301–313, 2003. doi: 10.1016/S1058-3300(03)00026-0.

James G. MacKinnon and Halbert White. Some heteroskedasticity-consistent covariance matrix estimators with improved finite sample properties. *Journal of Econometrics*, 29(3):305–325, September 1985. URL https://ideas.repec.org/a/eee/econom/v29y1985i3p305-325.html.

Terence C. Mills and Raphael N. Markellos. *The Econometric Modelling of Financial Time Series*. Cambridge University Press, United Kingdom, 2008. ISBN 9780521883818.

Jesus Otero and Jeremy Smith. Testing for cointegration: power versus frequency of observation – further Monte Carlo results. *Economics Letters*, 67(1):5–9, April 2000. URL https://ideas.repec.org/a/eee/ecolet/v67y2000i1p5-9.html.

R.J. Shiller. *Irrational Exuberance*. Scribe Publications, 2000. ISBN 9780908011582. URL https://books.google.co.uk/books?id=wJvnmRfz04QC.

A Tables

Table A1: Ljung-Box Test Results for ADF Test Residuals

ADF Test Specification	Levels		First Differences	
	LC	ID	LC_diff	ID_diff
None (no constant)	0.8983	0.7141	0.7525	0.0201**
Drift (with constant)	0.8784	0.6935	0.8498	0.6669
Trend (with constant & trend)	0.8968	0.7053	0.8553	0.6456

Notes: Table reports p-values from Ljung-Box Q-tests with 12 lags on residuals from ADF unit root tests. ADF tests were conducted using the AIC for optimal lag selection. LC refers to log S&P 500; ID refers to log Industrial Production Index; LC_diff and ID_diff are their first differences. *** p<0.01, ** p<0.05, * p<0.1

Table A2: VECM Estimation Results (Non-Robust)

	ΔLC (S&P 500)	ΔID (IPI)
Error Correction Term	-0.0400^{***}	-0.0070^{***}
	(0.0089)	(0.0012)
ΔLC_{t-1}	-0.0647	0.0216
	(0.0883)	(0.0122)
ΔID_{t-1}	-1.4140^{*}	-0.0058
	(0.6763)	(0.0931)
R^2	0.1508	0.3514
Adj. R^2	0.1289	0.3346
Num. obs.	119	119
RMSE	0.0349	0.0048

Notes: *** p <0.001; ** p <0.01; * p <0.05. LC = natural logarithm of the S&P 500 Index; ID = natural logarithm of the Industrial Production Index. Standard errors in parentheses.

Table A3: Sensitivity of ADF Test Results to Lag Selection

	Levels						First Differences					
	LC			ID			LC_diff			ID_diff		
Lag Selection	None	Drift	Trend	None	Drift	Trend	None	Drift	Trend	None	Drift	Trend
AIC	3.460	-0.512	-1.606	6.482	-0.318	-1.65	-7.1336***	-8.11***	-8.0755***	-4.4646***	-7.4006***	-7.337***
$T^{0.25} = 3$	3.2864	-0.2156	-1.6713	4.3108	-0.7455	-1.3585	-4.6701***	-5.8755	-5.8515***	-2.5935***	-4.6459***	-4.5978***
Fixed lag = 6	3.0131	-0.1068	-1.579	2.6033	-0.4899	-2.0415	-2.733***	-3.8226***	-3.7736***	-1.6729*	-3.217**	-3.11**
Fixed lag = 12	0.8586	-0.811	-2.2382	1.8457	-1.4299	-1.1161	-1.2863	-1.6701	-1.3308	-1.0423	-2.5515	-2.136

Notes: *** $p<0.01$, ** $p<0.05$, * $p<0.1$. LC refers to log S&P 500; ID refers to log Industrial Production Index; LC_diff and ID_diff are their first differences. T is sample size. The critical values at 1%, 5%, and 10% significance levels are -2.58, -1.95 and -1.62 (*'None'*), -3.46, -2.88 and -2.57 (*'Drift'*) and -3.99, -3.43 and -3.13 (*'Trend'*) from Hamilton [1994].

B Code

```
setwd("C:/Users/lollc/OneDrive - University of Bristol/Applied Financial
    Econometrics/Coursework")
# setwd("/Users/leo/Library/CloudStorage/OneDrive-UniversityofBristol/
   Applied Financial Econometrics/Coursework")

library(dplyr)
library(ggplot2)
library(zoo)
library(forecast)
library(urca)
library(xts)
library(tidyverse)
library(lubridate)
library(vars)
library(aTSA)
library(tseries)
library(stats)
library(tsDyn)

########################################################################
# SECTION 3: DATA
########################################################################

# --- 3A. Data Overview ---

df1 <- read.csv("S&P 500.csv")
df1 <- df1[, c("Date", "Close")]
df1 <- rename(df1, LC = Close)
df1$Date <- as.Date(df1$Date, "%Y-%m-%d")
df1$LC <- log(df1$LC)

df2 <- read.csv("INDPRO.csv")
df2 <- rename(df2, ID = INDPRO)
df2$Date <- as.Date(df2$observation_date, "%Y-%m-%d")
df2 <- df2[, c("Date", "ID")]
df2$ID <- log(df2$ID)

date_start <- as.Date("1991-01-01")
date_end <- as.Date("2001-01-01")
df1 <- df1 %>% filter(Date >= date_start & Date <= date_end)
df2 <- df2 %>% filter(Date >= date_start & Date <= date_end)

get_closest_value <- function(date, df_sp) {
  closest_idx <- which.min(abs(as.numeric(df_sp$Date - date)))
  return(df_sp$LC[closest_idx])
}

df <- df2 %>% mutate(LC = sapply(Date, function(d) get_closest_value(d,
   df1)))

# --- Table 1: Year-by-Year Annual Returns (Original Levels) ---

df_sp_levels <- read.csv("S&P 500.csv")
df_sp_levels$Date <- as.Date(df_sp_levels$Date, "%Y-%m-%d")
df_sp_levels <- df_sp_levels %>% filter(Date >= as.Date("1991-01-01") &
```

```
                                            Date <= as.Date("2001-01-01"))

df_ipi_levels <- read.csv("INDPRO.csv")
df_ipi_levels$Date <- as.Date(df_ipi_levels$observation_date, "%Y-%m-%d
   ")
df_ipi_levels <- df_ipi_levels %>%
  dplyr::select(Date, INDPRO) %>%
  filter(Date >= as.Date("1991-01-01") & Date <= as.Date("2001-01-01"))

df_sp_ann <- df_sp_levels %>%
  mutate(Year = year(Date)) %>%
  arrange(Date) %>%
  group_by(Year) %>%
  summarise(SP_Start = first(Close), SP_End = last(Close)) %>%
  mutate(SP_Return = (SP_End / SP_Start - 1) * 100)

df_ipi_ann <- df_ipi_levels %>%
  mutate(Year = year(Date)) %>%
  arrange(Date) %>%
  group_by(Year) %>%
  summarise(IPI_Start = first(INDPRO), IPI_End = last(INDPRO)) %>%
  mutate(IPI_Return = (IPI_End / IPI_Start - 1) * 100)

df_returns <- merge(
  df_sp_ann %>% dplyr::select(Year, SP_Return),
  df_ipi_ann %>% dplyr::select(Year, IPI_Return),
  by = "Year"
)

print("--- Table 1: Year-by-Year Annual Returns for S&P 500 and IPI
   (1991-2001) ---")
print(df_returns)

# --- Figure 1: Normalised Logged Series (1991-2001) ---

plot_data <- df %>%
  mutate(norm_LC = (LC / first(LC)) * 100,
         norm_ID = (ID / first(ID)) * 100)

# Create the plot (corresponds to Figure 1)
print( # Print ggplot object to display it in R viewer
  ggplot(plot_data, aes(x = Date)) +
    geom_line(aes(y = norm_LC, color = "S&P 500 (Normalised Log)"), size
        = 1) +
    geom_line(aes(y = norm_ID, color = "Industrial Production (
       Normalised Log)"), size = 1) +
    labs(title = "Normalised (Log) S&P 500 vs Industrial Production
       (1991-2001)",
         x = "Date", y = "Index (Base = 100)", color = "Series") +
    theme_minimal() +
    scale_color_manual(values = c("S&P 500 (Normalised Log)" = "blue",
                                  "Industrial Production (Normalised Log
                                     )" = "red"))
)

LC_xts <- xts(df$LC, order.by = df$Date)
ID_xts <- xts(df$ID, order.by = df$Date)
```

```
df_diff <- diff(cbind(LC_xts, ID_xts))
df_diff <- df_diff[-1,]

# --- 3B. Stationarity Testing ---

adf_star_string <- function(urdf_obj) {
  test_stat <- urdf_obj@teststat[1]
  cvals <- urdf_obj@cval[1, ]
  star <- ""
  if (!is.na(test_stat) && !is.na(cvals["1pct"]) && test_stat < cvals["1
     pct"]) {
    star <- "***"
  } else if (!is.na(test_stat) && !is.na(cvals["5pct"]) && test_stat <
     cvals["5pct"]) {
    star <- "**"
  } else if (!is.na(test_stat) && !is.na(cvals["10pct"]) && test_stat <
     cvals["10pct"]) {
    star <- "*"
  }
  paste0(round(test_stat, 3), star)
}

LC_diff <- diff(LC_xts) %>% na.omit()
adf_level_none <- ur.df(LC_diff, type = "none", selectlags = "AIC")
print("ADF Test Results for LC_diff (no constant, no trend):")
print(summary(adf_level_none))

extract_adf_results <- function(x_xts) {
  adf_level_none <- ur.df(x_xts, type = "none", selectlags = "AIC")
  adf_level_drift <- ur.df(x_xts, type = "drift", selectlags = "AIC")
  adf_level_trend <- ur.df(x_xts, type = "trend", selectlags = "AIC")

  print(summary(adf_level_none))
  print(summary(adf_level_drift))
  print(summary(adf_level_trend))

  x_diff <- diff(x_xts) %>% na.omit()
  adf_diff_none <- ur.df(x_diff, type = "none", selectlags = "AIC")
  adf_diff_drift <- ur.df(x_diff, type = "drift", selectlags = "AIC")
  adf_diff_trend <- ur.df(x_diff, type = "trend", selectlags = "AIC")

  print(summary(adf_diff_none))
  print(summary(adf_diff_drift))
  print(summary(adf_diff_trend))

  c(
    Level_None = adf_star_string(adf_level_none),
    Level_Intercept = adf_star_string(adf_level_drift),
    Level_Intercept_Trend = adf_star_string(adf_level_trend),
    Diff_None = adf_star_string(adf_diff_none),
    Diff_Intercept = adf_star_string(adf_diff_drift),
    Diff_Intercept_Trend = adf_star_string(adf_diff_trend)
  )
}

var_names <- c("ID", "LC")
results_list_adf <- lapply(var_names, function(var) {
```

```
  extract_adf_results(get(paste0(var, "_xts")))
  }) %>% setNames(var_names)

adf_results_df <- data.frame(
  Variable = var_names,
  Level_None = sapply(results_list_adf, function(x) x["Level_None"]),
  Level_Intercept = sapply(results_list_adf, function(x) x["
     Level_Intercept"]),
  Level_Intercept_Trend = sapply(results_list_adf, function(x) x["
     Level_Intercept_Trend"]),
  Diff_None = sapply(results_list_adf, function(x) x["Diff_None"]),
  Diff_Intercept = sapply(results_list_adf, function(x) x["
     Diff_Intercept"]),
  Diff_Intercept_Trend = sapply(results_list_adf, function(x) x["
     Diff_Intercept_Trend"]),
  row.names = NULL
)

print(adf_results_df)

# PP Test Function
extract_pp_results <- function(x_xts, type = "Z_tau") {
  x <- as.numeric(coredata(x_xts))
  pp_level <- aTSA::pp.test(x, type = type)
  x_diff <- diff(x) %>% na.omit()
  pp_diff <- aTSA::pp.test(x_diff, type = type)

  add_stars <- function(pval) {
    if (is.na(pval)) return("")
    if (pval <= 0.01) return("***")
    else if (pval <= 0.05) return("**")
    else if (pval <= 0.10) return("*")
    else return("")
  }

  level_none <- paste0(round(pp_level[1, 2], 3), add_stars(pp_level[1,
     3]))
  level_drift <- paste0(round(pp_level[2, 2], 3), add_stars(pp_level[2,
     3]))
  level_trend <- paste0(round(pp_level[3, 2], 3), add_stars(pp_level[3,
     3]))

  diff_none <- paste0(round(pp_diff[1, 2], 3), add_stars(pp_diff[1, 3]))
  diff_drift <- paste0(round(pp_diff[2, 2], 3), add_stars(pp_diff[2, 3])
     )
  diff_trend <- paste0(round(pp_diff[3, 2], 3), add_stars(pp_diff[3, 3])
     )

  c(
    Level_None = level_none,
    Level_Intercept = level_drift,
    Level_Intercept_Trend = level_trend,
    Diff_None = diff_none,
    Diff_Intercept = diff_drift,
    Diff_Intercept_Trend = diff_trend
  )
}
```

```
results_list_pp <- lapply(var_names, function(var) {
  extract_pp_results(get(paste0(var, "_xts")), type = "Z_tau")
}) %>% setNames(var_names)

pp_results_df <- data.frame(
  Variable = var_names,
  Level_None = sapply(results_list_pp, function(x) x["Level_None"]),
  Level_Intercept = sapply(results_list_pp, function(x) x["
     Level_Intercept"]),
  Level_Intercept_Trend = sapply(results_list_pp, function(x) x["
     Level_Intercept_Trend"]),
  Diff_None = sapply(results_list_pp, function(x) x["Diff_None"]),
  Diff_Intercept = sapply(results_list_pp, function(x) x["Diff_Intercept
     "]),
  Diff_Intercept_Trend = sapply(results_list_pp, function(x) x["
     Diff_Intercept_Trend"]),
  row.names = NULL
)

print(pp_results_df)

extract_kpss_results <- function(x_xts) {
  x <- as.numeric(coredata(x_xts))

  add_stars <- function(pval) {
    if (is.na(pval)) return("")
    if (pval <= 0.01) return("***")
    else if (pval <= 0.05) return("**")
    else if (pval <= 0.10) return("*")
    else return("")
  }

  level_intercept_test <- tseries::kpss.test(x, null = "Level", lshort =
      TRUE)
  level_intercept <- paste0(round(level_intercept_test$statistic, 3),
                            add_stars(level_intercept_test$p.value))

  level_trend_test <- tseries::kpss.test(x, null = "Trend", lshort =
     TRUE)
  level_trend <- paste0(round(level_trend_test$statistic, 3),
                        add_stars(level_trend_test$p.value))

  x_diff <- diff(x) %>% na.omit()
  diff_intercept_test <- tseries::kpss.test(x_diff, null = "Level",
     lshort = TRUE)
  diff_intercept <- paste0(round(diff_intercept_test$statistic, 3),
                           add_stars(diff_intercept_test$p.value))

  diff_trend_test <- tseries::kpss.test(x_diff, null = "Trend", lshort =
      TRUE)
  diff_trend <- paste0(round(diff_trend_test$statistic, 3),
                       add_stars(diff_trend_test$p.value))

  return(c(
    level_intercept = level_intercept,
    level_trend = level_trend,
```

```
    diff_intercept = diff_intercept,
    diff_trend = diff_trend
  ))
}

results_list_kpss <- list()
for (var in var_names) {
  var_data <- get(paste0(var, "_xts"))
  results_list_kpss[[var]] <- extract_kpss_results(var_data)
}

kpss_results_df <- data.frame(
  Variable = var_names,
  stringsAsFactors = FALSE
)
kpss_results_df$level_intercept <- sapply(results_list_kpss, function(x)
    x["level_intercept"])
kpss_results_df$level_trend <- sapply(results_list_kpss, function(x) x["
   level_trend"])
kpss_results_df$diff_intercept <- sapply(results_list_kpss, function(x)
   x["diff_intercept"])
kpss_results_df$diff_trend <- sapply(results_list_kpss, function(x) x["
   diff_trend"])

print("--- Appendix Table: KPSS Test Results Summary ---")
print(kpss_results_df)

# --- ADF Sensitivity Analysis with Alternative Lags ---
alt_lags <- c(6, 12, floor(nrow(LC_xts)^0.25)) # Lags: 6, 12, T^0.25

for (var in var_names) {
  var_xts <- get(paste0(var, "_xts"))
  var_diff_xts <- diff(var_xts) %>% na.omit()
  cat(paste("\n--- Variable:", var, "---\n"))

  for (lag in alt_lags) {
    cat(paste("\nLag =", lag, "\n"))
    cat("Level Tests:\n")
    print(summary(ur.df(var_xts, type = "none", lags = lag)))
    print(summary(ur.df(var_xts, type = "drift", lags = lag)))
    print(summary(ur.df(var_xts, type = "trend", lags = lag)))
    cat("\nDifference Tests:\n")
    print(summary(ur.df(var_diff_xts, type = "none", lags = lag)))
    print(summary(ur.df(var_diff_xts, type = "drift", lags = lag)))
    print(summary(ur.df(var_diff_xts, type = "trend", lags = lag)))
    cat("--------------------\n")
  }
}

# --- Figure 2: Autocorrelation Functions of First-Differenced Series
   ---
par(mfrow = c(1, 2))

acf(df_diff$LC, main = "ACF of Differenced S&P 500", na.action = na.pass
   , lag.max=12)
```

```
acf(df_diff$ID, main = "ACF of Differenced IPI", na.action = na.pass,
    lag.max=12)

par(mfrow = c(1, 1))

# --- Figure 3: Plots of First-Differenced Series ---
par(mfrow = c(2, 1))

plot(df_diff$LC, main = "First Differences of Log S&P 500 (dLC)", ylab="
    dLC", xlab="Date")
plot(df_diff$ID, main = "First Differences of Log IPI (dID)", ylab="dID
    ", xlab="Date")

par(mfrow = c(1, 1))

print(t.test(df_diff$LC, mu = 0))
print(t.test(df_diff$ID, mu = 0))
mean_LC_diff <- mean(df_diff$LC, na.rm = TRUE)
mean_ID_diff <- mean(df_diff$ID, na.rm = TRUE)
print(paste("Mean of differenced LC:", round(mean_LC_diff, 5)))
print(paste("Mean of differenced ID:", round(mean_ID_diff, 5)))


########################################################################
# SECTION 4: METHODOLOGY
########################################################################
dfx <- cbind(LC_xts, ID_xts)
# --- 4C. Cointegration Testing & 4D. Model Specification ---

lag_selection <- VARselect(df_diff, lag.max = 12, type = "const")
print(lag_selection)

lag_selection <- VARselect(df_diff, lag.max = 12, type = "none")
print(lag_selection)

# --- LR Test for Lag Length (K=3 vs K=2) ---
johansen_k3 <- ca.jo(dfx, type = "trace", K = 3, ecdet = "const", spec =
     "transitory")
vecm_k3 <- cajorls(johansen_k3, r = 1)

johansen_k2 <- ca.jo(dfx, type = "trace", K = 2, ecdet = "const", spec =
     "transitory")
vecm_k2 <- cajorls(johansen_k2, r = 1)

coef_count_k3 <- length(coef(vecm_k3$rlm))
coef_count_k2 <- length(coef(vecm_k2$rlm))

res_k3 <- residuals(vecm_k3$rlm)
res_k2 <- residuals(vecm_k2$rlm)

n_k3 <- nrow(res_k3)
n_k2 <- nrow(res_k2)

n_vars <- ncol(res_k3)

sigma_k3 <- crossprod(res_k3) / n_k3
sigma_k2 <- crossprod(res_k2) / n_k2
```

```
loglik_k3 <- -0.5 * n_k3 * (n_vars * log(2*pi) + log(det(sigma_k3)) +
   n_vars)
loglik_k2 <- -0.5 * n_k2 * (n_vars * log(2*pi) + log(det(sigma_k2)) +
   n_vars)

n_params_k3 <- coef_count_k3 + n_vars*(n_vars+1)/2
n_params_k2 <- coef_count_k2 + n_vars*(n_vars+1)/2

aic_k3 <- -2 * loglik_k3 + 2 * n_params_k3
aic_k2 <- -2 * loglik_k2 + 2 * n_params_k2

bic_k3 <- -2 * loglik_k3 + log(n_k3) * n_params_k3
bic_k2 <- -2 * loglik_k2 + log(n_k2) * n_params_k2

lr_stat <- 2 * (loglik_k3 - loglik_k2)
df <- n_params_k3 - n_params_k2
p_value <- 1 - pchisq(lr_stat, df)

print(aic_k3-aic_k2)
print(bic_k3-bic_k2)
print(p_value)
print(lr_stat)

# --- Main Cointegration Test  ---
johansen <- johansen_k2
print(summary(johansen))

test_stats <- johansen@teststat
critical_vals <- johansen@cval
lambda <- johansen@lambda
lambda_subset <- lambda[1:nrow(critical_vals)]
results <- cbind(
  test_stats,
  critical_vals,
  lambda = lambda_subset
)
results_ordered <- results[c(2, 1), , drop=FALSE]
rownames(results_ordered) <- c("r <= 1", "r = 0")

results_df <- data.frame(
  Hypothesis = rownames(results_ordered),
  Trace.Statistic = paste0(
    format(round(results_ordered[, 1], 2), nsmall = 2),
    ifelse(!is.na(results_ordered[, 1]) & !is.na(results_ordered[, 4]) &
        results_ordered[, 1] > results_ordered[, 4], "***",
           ifelse(!is.na(results_ordered[, 1]) & !is.na(results_ordered
              [, 3]) & results_ordered[, 1] > results_ordered[, 3],
              "**",
                  ifelse(!is.na(results_ordered[, 1]) & !is.na(
                     results_ordered[, 2]) & results_ordered[, 1] >
                     results_ordered[, 2], "*", "")))
  ),
  Critical.Value.10. = format(round(results_ordered[, 2], 2), nsmall =
     2),
  Critical.Value.5. = format(round(results_ordered[, 3], 2), nsmall = 2)
     ,
```

```
  Critical.Value.1. = format(round(results_ordered[, 4], 2), nsmall = 2)
     ,
  Eigenvalue = format(round(results_ordered[, 5], 4), nsmall = 4),
  stringsAsFactors = FALSE,
  row.names = NULL
)

print(results_df)
print(paste("Lag order (K):", johansen@lag))
print(paste("Deterministic terms (ecdet):", johansen@ecdet))
print(paste("Sample size:", nrow(johansen@x)))

# --- Equation 1: Estimated Long-Run Cointegrating Relationship ---
coint_vec <- johansen@V[,1]
norm_coint_vec <- coint_vec / coint_vec[1]
constant_term_eq1 <- -norm_coint_vec[3]
id_coefficient_eq1 <- -norm_coint_vec[2]

# --- VECM Estimation (K=2) ---
vecm_model <- vecm_k2

# --- 4E. Model Diagnostics ---
vecm_as_var <- vec2var(johansen, r=1)

serial_test <- serial.test(vecm_as_var, lags.pt = 16, type = "PT.
   asymptotic")

normality_test <- normality.test(vecm_as_var)

arch_test <- vars::arch.test(vecm_as_var, lags.multi = 5)

pt_stat <- as.numeric(serial_test$serial$statistic); pt_df <- as.numeric
   (serial_test$serial$parameter); pt_pval <- as.numeric(
   serial_test$serial$p.value)
jb_stat <- tryCatch(as.numeric(normality_test$jb.mul$JB$statistic[1,1]),
    error=function(e) NA); jb_df <- tryCatch(as.numeric(
   normality_test$jb.mul$JB$parameter), error=function(e) NA); jb_pval
   <- tryCatch(as.numeric(normality_test$jb.mul$JB$p.value[1,1]), error=
   function(e) NA)
sk_stat <- tryCatch(as.numeric(normality_test$jb.mul$Skewness$statistic
   [1,1]), error=function(e) NA); sk_df <- tryCatch(as.numeric(
   normality_test$jb.mul$Skewness$parameter), error=function(e) NA);
   sk_pval <- tryCatch(as.numeric(normality_test$jb.mul$Skewness$p.value
   [1,1]), error=function(e) NA)
kt_stat <- tryCatch(as.numeric(normality_test$jb.mul$Kurtosis$statistic
   [1,1]), error=function(e) NA); kt_df <- tryCatch(as.numeric(
   normality_test$jb.mul$Kurtosis$parameter), error=function(e) NA);
   kt_pval <- tryCatch(as.numeric(normality_test$jb.mul$Kurtosis$p.value
   [1,1]), error=function(e) NA)
arch_stat <- as.numeric(arch_test$arch.mul$statistic); arch_df <- as.
   numeric(arch_test$arch.mul$parameter); arch_pval <- as.numeric(
   arch_test$arch.mul$p.value)

diag_results_df <- data.frame(
  Test = c("Portmanteau Test", "JB-Test", "Skewness", "Kurtosis", "ARCH
     Test"),
  Chi_Squared = c(pt_stat, jb_stat, sk_stat, kt_stat, arch_stat),
```

```
  df = c(pt_df, jb_df, sk_df, kt_df, arch_df),
  p_value = c(pt_pval, jb_pval, sk_pval, kt_pval, arch_pval),
  stringsAsFactors = FALSE
)
diag_results_df$Chi_Squared <- round(diag_results_df$Chi_Squared, 2)
diag_results_df$p_value <- round(diag_results_df$p_value, 3)
print(diag_results_df)

# --- 4F. Heteroskedasticity-Robust Inference (on chosen K=2 model) ---
rlm_model <- vecm_model$rlm
coef_matrix <- coef(rlm_model)
X <- model.matrix(rlm_model$terms, rlm_model$model)
resid_matrix <- residuals(rlm_model)
n <- nrow(X)
p <- ncol(X)
df <- n - p

XtX_inv <- solve(crossprod(X) + diag(ncol(X)) * 1e-10)
robust_se_matrix <- matrix(0, nrow = nrow(coef_matrix), ncol = ncol(
   coef_matrix))
rownames(robust_se_matrix) <- rownames(coef_matrix)
colnames(robust_se_matrix) <- colnames(coef_matrix)

for (i in 1:ncol(coef_matrix)) {
  resid_i <- resid_matrix[, i]
  h <- diag(X %*% XtX_inv %*% t(X))
  u_star <- resid_i / (1 - pmin(h, 0.95))
  omega_star_diag <- u_star^2
  first_term <- t(X) %*% diag(omega_star_diag) %*% X
  second_term <- (1/n) * (t(X) %*% (u_star %*% t(u_star)) %*% X)
  middle_term <- first_term - second_term
  robust_vcov <- ((n-1)/n) * XtX_inv %*% middle_term %*% XtX_inv
  robust_se_matrix[, i] <- sqrt(diag(robust_vcov))
}

print(round(coef_matrix, 4))
print(round(robust_se_matrix, 4))

robust_p_values <- 2 * pt(abs(coef_matrix / robust_se_matrix), df = df,
   lower.tail = FALSE)
print("Robust P-values:")
print(round(robust_p_values, 4))

add_stars <- function(p) {
  if (is.na(p)) return("")
  if (p < 0.01) return("***")
  if (p < 0.05) return("**")
  if (p < 0.10) return("*")
  return("")
}
stars_matrix <- apply(robust_p_values, c(1, 2), add_stars)

combined_output <- matrix(paste0(format(round(coef_matrix, 4), nsmall =
   4), stars_matrix),
                          nrow = nrow(coef_matrix), dimnames = dimnames(
                             coef_matrix))
print(combined_output, quote = FALSE)
```

```
summary_list <- summary(rlm_model)
r_squared <- sapply(summary_list, function(s) s$r.squared)
adj_r_squared <- sapply(summary_list, function(s) s$adj.r.squared)
rmse <- sapply(1:ncol(resid_matrix), function(i) sqrt(mean(resid_matrix
   [,i]^2)))
n_obs <- nrow(X)

print(paste("Observations:", n_obs))
print(paste("R-squared (LC, ID):", round(r_squared[1], 4), ",", round(
   r_squared[2], 4)))
print(paste("Adj. R-squared (LC, ID):", round(adj_r_squared[1], 4), ",",
    round(adj_r_squared[2], 4)))
print(paste("RMSE (LC, ID):", round(rmse[1], 4), ",", round(rmse[2], 4))
   )


########################################################################
# SECTION 5: EMPIRICAL ESTIMATION RESULTS AND DISCUSSION
########################################################################

# --- 5B. Impulse Response Analysis  ---

irf_LC_ID <- vars::irf(vecm_as_var, impulse = "LC_xts", response = "
   ID_xts",
                       boot = TRUE, runs = 1000, n.ahead = 24, ci =
                          0.95)
plot(irf_LC_ID, main = "Impulse Response of ID to Shock in LC", ylab = "
   Response of ID")

irf_ID_LC <- vars::irf(vecm_as_var, impulse = "ID_xts", response = "
   LC_xts",
                       boot = TRUE, runs = 1000, n.ahead = 24, ci =
                          0.95)
plot(irf_ID_LC, main = "Impulse Response of LC to Shock in ID", ylab = "
   Response of LC")

#######################################
# APPENDIX TABLES
#######################################

# LC
adf_level_none <- ur.df(LC_xts, type = "none",lags = 1)
adf_level_drift <- ur.df(LC_xts, type = "drift", lags = 1)
adf_level_trend <- ur.df(LC_xts, type = "trend", lags = 1)
residuals_none <- residuals(adf_level_none@testreg)
residuals_drift <- residuals(adf_level_drift@testreg)
residuals_trend <- residuals(adf_level_trend@testreg)
lb_test_none <- Box.test(residuals_none, type = "Ljung-Box", lag = 12)
print("Ljung-Box Test for ADF (None) Residuals:")
print(lb_test_none)
# Ljung-Box test for 'drift' specification
lb_test_drift <- Box.test(residuals_drift, type = "Ljung-Box", lag = 12)
print("Ljung-Box Test for ADF (Drift) Residuals:")
print(lb_test_drift)
# Ljung-Box test for 'trend' specification
lb_test_trend <- Box.test(residuals_trend, type = "Ljung-Box", lag = 12)
```

```
print("Ljung-Box Test for ADF (Trend) Residuals:")
print(lb_test_trend)

# ID
ID_diff <- diff(ID_xts) %>% na.omit()
adf_level_none <- ur.df(ID_xts, type = "none",lags = 1)
adf_level_drift <- ur.df(ID_xts, type = "drift", lags = 1)
adf_level_trend <- ur.df(ID_xts, type = "trend", lags = 1)
residuals_none <- residuals(adf_level_none@testreg)
residuals_drift <- residuals(adf_level_drift@testreg)
residuals_trend <- residuals(adf_level_trend@testreg)
lb_test_none <- Box.test(residuals_none, type = "Ljung-Box", lag = 12)
print("Ljung-Box Test for ADF (None) Residuals:")
print(lb_test_none)
# Ljung-Box test for 'drift' specification
lb_test_drift <- Box.test(residuals_drift, type = "Ljung-Box", lag = 12)
print("Ljung-Box Test for ADF (Drift) Residuals:")
print(lb_test_drift)
# Ljung-Box test for 'trend' specification
lb_test_trend <- Box.test(residuals_trend, type = "Ljung-Box", lag = 12)
print("Ljung-Box Test for ADF (Trend) Residuals:")
print(lb_test_trend)

# LC diff
adf_level_none <- ur.df(LC_diff, type = "none",lags = 1)
adf_level_drift <- ur.df(LC_diff, type = "drift", lags = 1)
adf_level_trend <- ur.df(LC_diff, type = "trend", lags = 1)
residuals_none <- residuals(adf_level_none@testreg)
residuals_drift <- residuals(adf_level_drift@testreg)
residuals_trend <- residuals(adf_level_trend@testreg)
lb_test_none <- Box.test(residuals_none, type = "Ljung-Box", lag = 12)
print("Ljung-Box Test for ADF (None) Residuals:")
print(lb_test_none)
# Ljung-Box test for 'drift' specification
lb_test_drift <- Box.test(residuals_drift, type = "Ljung-Box", lag = 12)
print("Ljung-Box Test for ADF (Drift) Residuals:")
print(lb_test_drift)
# Ljung-Box test for 'trend' specification
lb_test_trend <- Box.test(residuals_trend, type = "Ljung-Box", lag = 12)
print("Ljung-Box Test for ADF (Trend) Residuals:")
print(lb_test_trend)

# ID diff
adf_level_none <- ur.df(ID_diff, type = "none",lags = 1)
adf_level_drift <- ur.df(ID_diff, type = "drift", lags = 1)
adf_level_trend <- ur.df(ID_diff, type = "trend", lags = 1)
residuals_none <- residuals(adf_level_none@testreg)
residuals_drift <- residuals(adf_level_drift@testreg)
residuals_trend <- residuals(adf_level_trend@testreg)
lb_test_none <- Box.test(residuals_none, type = "Ljung-Box", lag = 12)
print("Ljung-Box Test for ADF (None) Residuals:")
print(lb_test_none)
# Ljung-Box test for 'drift' specification
lb_test_drift <- Box.test(residuals_drift, type = "Ljung-Box", lag = 12)
print("Ljung-Box Test for ADF (Drift) Residuals:")
print(lb_test_drift)
# Ljung-Box test for 'trend' specification
```

```
lb_test_trend <- Box.test(residuals_trend, type = "Ljung-Box", lag = 12)
print("Ljung-Box Test for ADF (Trend) Residuals:")
print(lb_test_trend)

mysum <- summary((vecm_k2)$rlm)
screenreg(list(mysum[[1]], mysum[[2]]))
model_summaries <- summary(vecm_model$rlm)
screenreg(
  list(model_summaries[[1]], model_summaries[[2]]),
  custom.model.names = c("\Delta Log(S&P 500)", "\Delta Log(IPI)"),
  custom.coef.names = c(
    "Error Correction Term (ECT)",
    "Lag 1 \DeltaLog(S&P 500)",
    "Lag 1 \Delta Log(IPI)"
  ),
  include.rsquared = TRUE,
  include.adjrs = TRUE,
  include.nobs = TRUE,
  include.rmse = TRUE,
  digits = 4,
  single.row = TRUE
)

var_decomposition <- fevd(vecm_as_var, n.ahead = 48)

horizons <- c(1, 6, 12, 24, 36, 48)
sp500_decomp <- t(sapply(horizons, function(h) var_decomposition$LC_xts[
   h,]))
ipi_decomp <- t(sapply(horizons, function(h) var_decomposition$ID_xts[h
   ,]))

rownames(sp500_decomp) <- paste0("h=", horizons)
rownames(ipi_decomp) <- paste0("h=", horizons)

print("Variance Decomposition of S&P 500:")
print(round(sp500_decomp * 100, 2))
print("Variance Decomposition of Industrial Production:")
print(round(ipi_decomp * 100, 2))
```