The Impact of Credit Market Stress on Asset Volatility: A Quantile Regression Approach for Bitcoin and NASDAQ

Lukas Rueda

May 25, 2025

1 Introduction

The complex, jump-prone dynamics of asset return volatility are critical for risk management, asset pricing, and forecasting, often challenging traditional models and highlighting the importance of understanding sudden volatility changes [1]. This study draws inspiration from Caporin, Rossi, and Santucci de Magistris [1], who demonstrated with their HAR-Volatility-Jump model that credit risk indicators can predict volatility jumps identified using high-frequency data [1].

This paper investigates how credit market stress, proxied by the BAA-AAA corporate bond spread, impacts the volatility of both a digital asset (Bitcoin) and a traditional one (NASDAQ). We hypothesize that higher credit spreads increase subsequent asset volatility ($H_a : \beta_1 > 0$), and further, that Bitcoin exhibits greater sensitivity to such stress than NASDAQ.

Given the unavailability of high-frequency data used by Caporin, Rossi, and Santucci de Magistris [1], this study employs a panel framework with publicly available daily data for Bitcoin and NASDAQ. Log-volatility (`LogVol_20d`) is approximated as the natural logarithm of the annualized 20-day rolling standard deviation of daily log returns (using $\sqrt{252}$ for annualization). This proxy aims to capture the core of Caporin et al.'s (2016) volatility concept within daily data constraints [1].

The primary innovation of this research is the use of asset-specific quantile regressions to explore how the impact of credit stress (and other factors) varies across low, medium, and high conditional volatility regimes for both Bitcoin and NASDAQ. This nonlinear approach, extending traditional Ordinary Least Squares (OLS) analysis, seeks a more nuanced understanding of risk transmission and volatility dynamics in diverse financial markets.

2 Data and Summary Statistics

This study uses daily data (March 23, 2021 - April 9, 2025) for Bitcoin (BTC) prices (sourced from Yahoo Finance) and its DVOL implied volatility index (from Deribit), alongside NASDAQ Composite Index prices (Yahoo Finance) and its VXN implied volatility index (from FRED). FRED also provided data for the Baa-Aaa credit spread and the 10-year Treasury minus 3-month Treasury bill spread. After data cleaning, necessary transformations (including log return and volatility calculations), lagging of independent variables, and listwise deletion to ensure a complete dataset for regression, the final panel dataset available for the core analysis consists of 1,144 daily observations. This sample accounts for typical gaps found in financial market data, such as weekends and holidays.

2.1 Variable Definitions

The key variables constructed and used in this analysis are detailed below.

2.1.1 Dependent Variable

The dependent variable, log volatility (`logvol_20d`), measures recent realized volatility for both Bitcoin and NASDAQ. Inspired by the methodology of Caporin, Rossi, and Santucci de Magistris [1], it is calculated as the natural logarithm of the 20-day rolling standard deviation of daily log returns, which is then annualized by multiplying by the square root of 252.

2.1.2 Independent Variables

To explain variations in log volatility, several one-day lagged independent variables are employed, chosen based on financial theory and empirical evidence and to mitigate endogeneity concerns. Key among these is the **`L_baa_aaa_spread`**, representing the lagged Baa-Aaa corporate bond yield spread, which serves as a proxy for credit market stress. Another important factor is the **`L_treasury_10y_3m_spread`**, the lagged difference between 10-year Treasury and 3-month Treasury bill rates, reflecting the yield curve slope. Asset-specific market expectations of future fluctuations are captured by **`L_implied_vol`**, which is the lagged implied volatility (DVOL_BTC for Bitcoin, VXN for NASDAQ). Finally, to account for the leverage effect where negative returns often precede higher volatility, **`L_neg_log_ret`** is included, representing lagged negative daily log returns.

2.2 Descriptive Insights from the Data

Table 1: Overall Descriptive Statistics of Key Variables

Variable	Obs	Mean	Std. Dev.	Min	Max
Log Volatility (20-day)	1,144	-1.2077	0.5797	-2.8033	0.2115
L. Baa-Aaa Spread	1,144	0.8561	0.1959	0.5800	1.2300
L. Treasury Spread (10Y-3M)	1,144	0.0603	1.1621	-1.7300	2.2100
L. Implied Volatility	1,144	44.8203	24.8807	11.2000	139.9800
L. Negative Log Return	1,144	-0.0086	0.0161	-0.1549	0.0000

Note: Statistics are for the 1,144 observations used in regression analysis. ‘L.‘ denotes a one-period lag.

Descriptive statistics are presented in Table 1 (overall regression sample, N=1,144), Table 2 (by individual asset), and Table 3 (panel data decomposition), reveal important data characteristics. From Table 1, which summarizes the 1,144 observations used in the regression analysis for all listed variables, the average `logvol_20d` was -1.2077. Key predictors from this same sample include `L_baa_aaa_spread` (mean 0.8561), `L_treasury_10y_3m_spread` (mean 0.0603 with considerable variation), `L_implied_vol` (mean 44.8203 with high standard deviation), and the leverage effect proxy, `L_neg_log_ret` (mean -0.0086).

Table 2 highlights notable distinctions between the assets. Bitcoin, on average, exhibits higher mean log volatility (-0.794) compared to NASDAQ (-1.628) and also shows considerably higher and more dispersed lagged implied volatility. Furthermore, instances of negative returns were, on average, slightly larger in magnitude for Bitcoin than for NASDAQ, underscoring their unique risk-return profiles.

Table 2: Descriptive Statistics by Asset

Asset	Variable	N	Mean	Std. Dev.	Min	Max
Bitcoin	Log Volatility (20-day)	1,007	-0.794	0.393	-2.279	0.231
	L. Implied Volatility	787	65.954	18.040	32.740	139.980
	L. Negative Log Return	572	-0.012	0.020	-0.155	0.000
NASDAQ	Log Volatility (20-day)	1,007	-1.628	0.428	-2.992	-0.311
	L. Implied Volatility	787	23.897	5.818	11.200	41.430
	L. Negative Log Return	572	-0.005	0.009	-0.057	0.000
Total	Log Volatility (20-day)	2,014	-1.211	0.585	-2.992	0.231
(Pooled)	L. Implied Volatility	1,574	44.925	24.940	11.200	139.980
	L. Negative Log Return	1,144	-0.009	0.016	-0.155	0.000

Note: 'L.' denotes a one-period lag. N for each variable by asset reflects available observations. Total statistics for L. Negative Log Return reflect the regression sample count.

The panel data characteristics (Table 3, derived from 'xtsum') offer further insights. For log volatility, the variation *between* Bitcoin and NASDAQ is greater than the average variation *within* each asset over time, indicating inherent differences in their average volatility levels. As expected, macroeconomic indicators like the Baa-Aaa and Treasury spreads show no variation between assets (being common factors), while asset-specific measures such as implied volatility and negative log returns demonstrate considerable variability both between the assets and within each asset over the period.

Table 3: Panel Data Summary Statistics (xtsum)

Variable	Dimension	Mean	Std. Dev.	Min	Max	Observations
Log Volatility (20-day)	Overall	-1.2110	0.5850	-2.9921	0.2309	N = 2014
	Between		0.5891	-1.6275	-0.7945	n = 2
	Within		0.4107	-2.6955	0.1055	T-bar = 1007
L. Baa-Aaa Spread	Overall	0.8551	0.1967	0.5800	1.2300	N = 1574
	Between		0.0000	0.8551	0.8551	n = 2
	Within		0.1967	0.5800	1.2300	T-bar = 787
L. Treasury Spread (10Y-3M)	Overall	0.0589	1.1569	-1.7300	2.2100	N = 1574
	Between		0.0000	0.0589	0.0589	n = 2
	Within		1.1569	-1.7300	2.2100	T-bar = 787
L. Implied Volatility	Overall	44.9255	24.9404	11.2000	139.9800	N = 1574
	Between		29.7391	23.8968	65.9542	n = 2
	Within		13.3991	11.7113	118.9513	T-bar = 787
L. Negative Log Return	Overall	-0.0086	0.0161	-0.1549	0.0000	N = 1144
	Between		0.0047	-0.0119	-0.0053	n = 2
	Within		0.0157	-0.1516	0.0033	T-bar = 572

Note: 'L.' denotes a one-period lag. N denotes total observations, n denotes number of panels (assets), T-bar denotes average observations per panel.

In essence, the descriptive statistics indicate that Bitcoin generally presents a higher and more variable volatility profile than the NASDAQ index. Both assets exhibit characteristics consistent with the leverage effect. The macroeconomic spreads behave as common factors, while asset-specific measures demonstrate unique behaviors. This interplay affirms the suitability of a panel data approach for this study, and the final dataset of 1,144 complete daily observations provides a solid foundation for the subsequent regression analysis.

3 Econometric Methods

This study uses a panel data approach with daily observations for Bitcoin and NASDAQ (March 23, 2021 - April 9, 2025) to examine how credit market stress and other factors impact 20-day log volatility (`logvol_20d`). To mitigate potential endogeneity and model the influence of prior conditions, all explanatory variables (detailed in Section 2.1.2) are lagged by one day.

3.1 Baseline Model: Pooled Ordinary Least Squares (OLS)

While initially considering fixed-effects, diagnostic tests on the panel data guided the selection of Pooled Ordinary Least Squares (OLS) as the more statistically appropriate baseline estimator. Both the Breusch-Pagan LM test (Prob > chibar2 = 1.0000, favoring Pooled OLS over Random Effects) and an F-test for fixed effects (Prob > F = 0.2002) suggested that distinct, time-invariant asset-specific effects were not dominant after controlling for the included regressors.

Consequently, our baseline Pooled OLS model is specified as:

$$\begin{aligned} \texttt{logvol\_20d}_{it} = \beta_0 &+ \beta_1 \texttt{L\_baa\_aaa\_spread}_{it} \\ &+ \beta_2 \texttt{L\_treasury\_10y\_3m\_spread}_{it} \\ &+ \beta_3 \texttt{L\_implied\_vol}_{it} \\ &+ \beta_4 \texttt{L\_neg\_log\_ret}_{it} + \epsilon_{it} \end{aligned} \tag{1}$$

In this equation, $\texttt{logvol\_20d}_{it}$ is regressed on the lagged explanatory variables. The term β_0 is the intercept, β_1 through β_4 are the impact coefficients, and ϵ_{it} is the error term. Robust standard errors are employed to ensure reliable statistical inferences.

3.2 Quantile Regression

To explore complex, heterogeneous impacts of explanatory factors on log volatility beyond the average effects captured by OLS, this study employs quantile regression as its innovative analytical step. This approach allows modeling how predictors affect different parts (quantiles) of the conditional log volatility distribution. The quantile regression model for a specific quantile τ (where $0 < \tau < 1$) is formulated as:

$$\begin{aligned} Q_{\texttt{logvol\_20d}_{it}}(\tau|X_{it}) = \beta_0(\tau) &+ \beta_1(\tau)\texttt{L\_baa\_aaa\_spread}_{it} \\ &+ \beta_2(\tau)\texttt{L\_treasury\_10y\_3m\_spread}_{it} \\ &+ \beta_3(\tau)\texttt{L\_implied\_vol}_{it} \\ &+ \beta_4(\tau)\texttt{L\_neg\_log\_ret}_{it} \end{aligned} \tag{2}$$

Here, $Q_{\texttt{logvol\_20d}_{it}}(\tau|X_{it})$ is the τ^{th} conditional quantile of log volatility for asset i at time t, given the lagged explanatory variables X_{it}. The coefficients, $\beta_0(\tau)$ to $\beta_4(\tau)$, are estimated for each quantile τ of interest.

We analyze the 25th, 50th (median), and 75th percentiles of conditional log volatility. To capture potentially distinct responses to market factors, these quantile models are estimated separately for Bitcoin and NASDAQ using simultaneous quantile regression (via Stata's `sqreg` command). Standard errors for these coefficients are obtained through bootstrapping, a reliable method that does not depend on strict assumptions about how the errors are distributed. This asset-specific quantile approach facilitates a detailed examination of how predictor impacts vary across volatility states and between the two distinct assets.

4 Results

This section discusses empirical findings from the econometric models detailed in Section 3. We begin with the average relationships identified by the baseline Pooled Ordinary Least Squares (OLS) model, then delve into insights from the asset-specific Quantile Regression (QR) analysis, which uncovers how these relationships vary for Bitcoin and NASDAQ across different levels of conditional log volatility.

4.1 Baseline Model: Pooled OLS Results

Table 4: Pooled OLS Regression of Log Volatility

Variable	Coefficient	Std. Err.	z	P>|z|	[95% Conf. Interval]
L_baa_aaa_spread	0.268^{***}	0.052	5.18	0.000	0.166 – 0.369
L_treasury_10y_3m_spread	-0.013	0.009	-1.37	0.172	-0.031 – 0.005
L_implied_vol	0.019^{***}	0.000	45.79	0.000	0.018 – 0.020
L_neg_log_ret	-2.569^{***}	0.613	-4.19	0.000	-3.771 – -1.366
Constant	-2.314^{***}	0.048	-48.50	0.000	-2.408 – -2.221
Observations	1,144				
Overall R-squared	0.698				
Wald $\chi^2(4)$	2632.86^{***}				

Note: *** $p < 0.001$, ** $p < 0.05$, * $p < 0.10$. Standard errors from 'xtreg, re' (GLS) in parentheses for coefficients if shown, here reported in a separate column. 'L.' denotes one-period lagged variables. Confidence interval presented as [Lower Bound – Upper Bound].

Table 4 presents results from the Pooled OLS regression (Equation 1), estimated with robust standard errors to determine the average impact of lagged predictors on the 20-day log volatility of Bitcoin and NASDAQ. The findings indicate that the lagged Baa-Aaa spread (`L_baa_aaa_spread`) has a statistically significant positive average effect on log volatility (coefficient 0.268, $p < 0.001$). Similarly, lagged implied volatility (`L_implied_vol`) significantly increases volatility (0.019, $p < 0.001$), and the leverage effect, proxied by `L_neg_log_ret`, is also significant (-2.569, $p < 0.001$). In this pooled model, the lagged Treasury spread (`L_treasury_10y_3m_spread`) was found to be statistically insignificant ($p = 0.172$).

Overall, the Pooled OLS model (R-squared 0.6980) explains approximately 69.8% of the variation in log volatility across the panel. However, its assumption of uniform predictor impacts across both assets and all market conditions motivates our extension to asset-specific Quantile Regression analysis to explore these potential variations.

4.2 Asset-Specific Quantile Regression Results

To explore how the impact of explanatory variables might change at different levels of log volatility and differ specifically between Bitcoin and NASDAQ, we estimated simultaneous quantile regression models separately for each asset. This was done for the 25th (Q25), 50th (Q50 - median), and 75th (Q75) percentiles of the conditional log volatility distribution for both Bitcoin (Table 5) and NASDAQ (Table 6). Figures 1 through 2 visualize these quantile-specific coefficients and their 95% confidence intervals against the pooled OLS estimates.

Table 5: Quantile Regression of Log Volatility for Bitcoin

Variable	Q25 (BTC) Coef. (SE)	Q50 (BTC) Coef. (SE)	Q75 (BTC) Coef. (SE)
L_baa_aaa_spread	-0.224***	-0.146**	-0.183*
	(0.076)	(0.074)	(0.101)
L_treasury_10y_3m_spread	-0.021	-0.026	-0.052***
	(0.023)	(0.022)	(0.013)
L_implied_vol	0.016***	0.015***	0.014***
	(0.001)	(0.001)	(0.001)
L_neg_log_ret	-2.536***	-2.178***	-1.866**
	(0.713)	(0.598)	(0.575)
Constant	-1.883***	-1.671***	-1.350***
	(0.100)	(0.097)	(0.141)
Observations	572	572	572
Pseudo R-sq (Q25)	0.314		
Pseudo R-sq (Q50)		0.248	
Pseudo R-sq (Q75)			0.241

Note: *** $p < 0.001$, ** $p < 0.05$, * $p < 0.10$. Bootstrap standard errors (reps 50) in parentheses. 'L.' denotes one-period lagged variables.

Table 6: Quantile Regression of Log Volatility for NASDAQ

Variable	Q25 (NASDAQ) Coef. (SE)	Q50 (NASDAQ) Coef. (SE)	Q75 (NASDAQ) Coef. (SE)
L_baa_aaa_spread	0.046	-0.175^{*}	-0.225^{**}
	(0.172)	(0.104)	(0.091)
L_treasury_10y_3m_spread	-0.115^{***}	-0.095^{***}	-0.058^{**}
	(0.028)	(0.021)	(0.025)
L_implied_vol	0.065^{***}	0.065^{***}	0.056^{***}
	(0.005)	(0.003)	(0.005)
L_neg_log_ret	-1.649	-2.693^{*}	-0.675
	(2.171)	(1.587)	(1.096)
Constant	-3.411^{***}	-3.032^{***}	-2.568^{***}
	(0.098)	(0.092)	(0.071)
Observations	572	572	572
Pseudo R-sq (Q25)	0.381		
Pseudo R-sq (Q50)		0.367	
Pseudo R-sq (Q75)			0.403

Note: *** $p < 0.001$, ** $p < 0.05$, * $p < 0.10$. Bootstrap standard errors (reps 50) in parentheses. 'L.' denotes one-period lagged variables.

4.2.1 Discussion of Quantile Regression Results by Variable

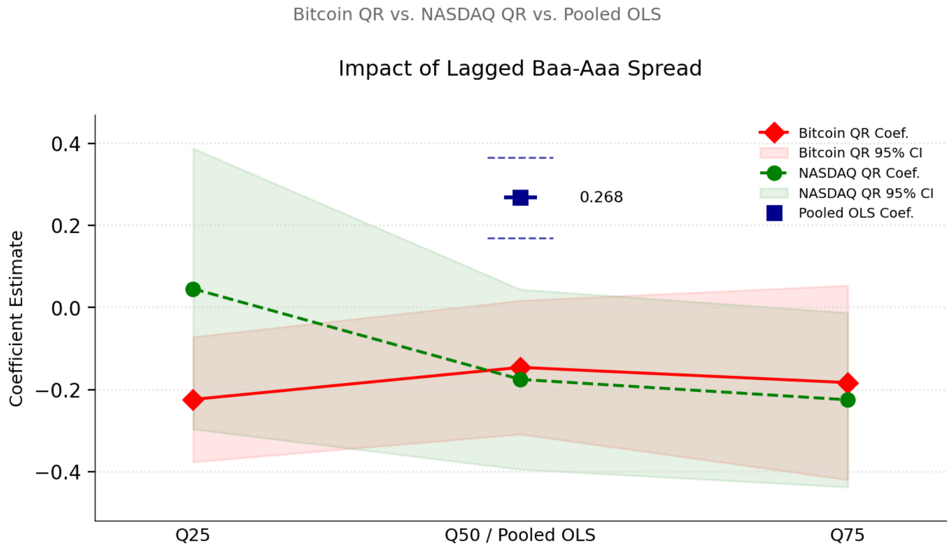

Figure 1: Quantile Regression Coefficients for Lagged Baa-Aaa Spread: Bitcoin (red diamonds) and NASDAQ (green circles) versus Pooled OLS (blue square). Shaded areas represent 95% confidence intervals for QR coefficients; dashed lines for OLS CI.

Impact of Lagged Baa-Aaa Spread (`L_baa_aaa_spread`) Figure 1 (with details in Tables 5 and 6) reveals that asset-specific quantile effects of the lagged Baa-Aaa spread differ markedly from the positive and significant Pooled OLS estimate. For **Bitcoin**, an increase in this credit spread is linked to a statistically significant *decrease* in subsequent log volatility across all its examined quantiles (25th, 50th, and 75th). For **NASDAQ**, the impact of the credit spread is not statistically significant at its lower (25th) quantile; however, at its median (50th) and upper (75th) quantiles, representing medium to high volatility states for NASDAQ, an increased credit spread is associated with significantly lower subsequent log volatility. Thus, where significant, the asset-specific quantile effects of credit stress primarily suggest a volatility-dampening role for both assets, contrasting with the average positive impact found by Pooled OLS.

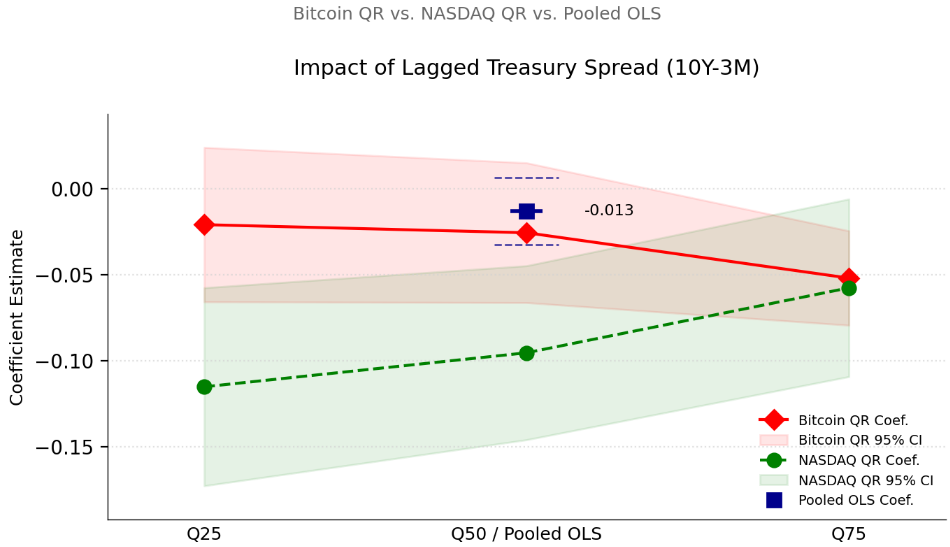

Figure 2: Quantile Regression Coefficients for Lagged Treasury Spread (10Y-3M): Bitcoin (red diamonds) and NASDAQ (green circles) versus Pooled OLS (blue square). Shaded areas represent 95% confidence intervals for QR coefficients; dashed lines for OLS CI.

Impact of Lagged Treasury Spread (`L_treasury_10y_3m_spread`) The lagged Treasury spread (Figure 2), found insignificant in the Pooled OLS model, reveals significant relationships in the asset-specific quantile analysis. For Bitcoin, a steeper yield curve significantly *lowers* subsequent volatility, but only when Bitcoin is already in a high volatility state (75th quantile). Conversely, for NASDAQ, a steeper yield curve is consistently associated with significantly lower future volatility across all quantiles, with the most pronounced dampening effect in its lower volatility states. This highlights how quantile regressions can uncover relationships missed by average-effect models.

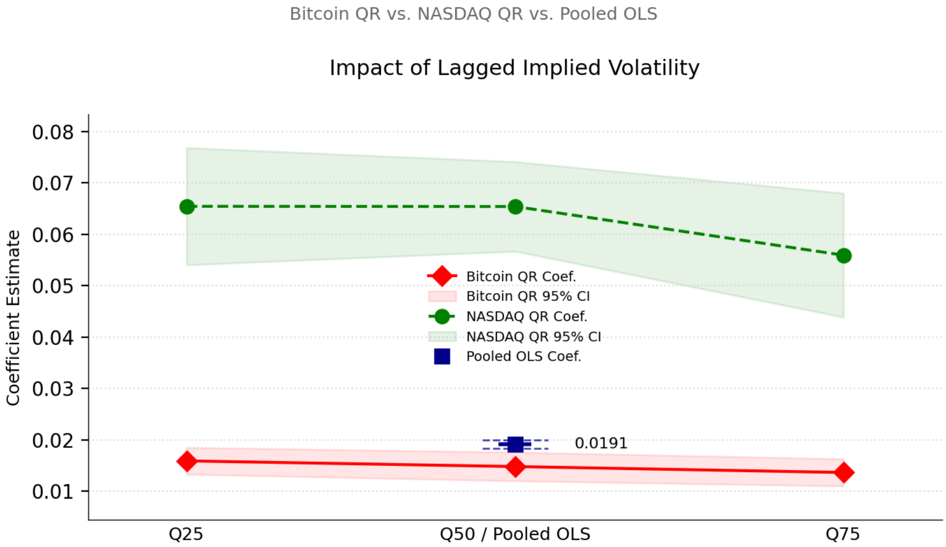

Figure 3: Quantile Regression Coefficients for Lagged Implied Volatility: Bitcoin (red diamonds) and NASDAQ (green circles) versus Pooled OLS (blue square). Shaded areas represent 95% confidence intervals for QR coefficients; dashed lines for OLS CI.

Impact of Lagged Implied Volatility (`L_implied_vol`) As shown in Figure 3, lagged implied volatility is a consistently positive and highly significant predictor of future log volatility for both Bitcoin and NASDAQ across all examined quantiles (see Tables 5 and 6). While this effect is remarkably stable for Bitcoin, its magnitude is substantially larger for NASDAQ. Consequently, the Pooled OLS coefficient appears to reflect Bitcoin's lower sensitivity rather than NASDAQ's stronger response.

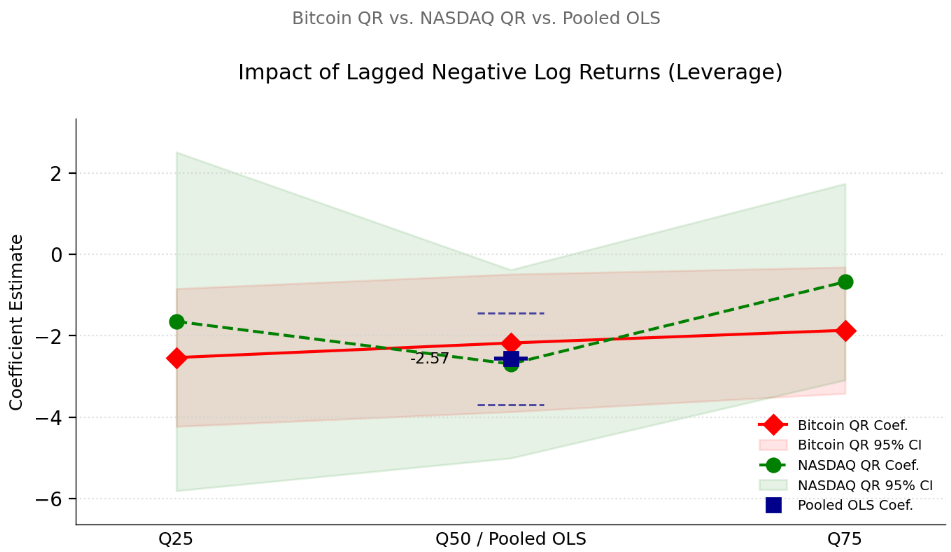

Figure 4: Quantile Regression Coefficients for Lagged Negative Log Returns: Bitcoin (red diamonds) and NASDAQ (green circles) versus Pooled OLS (blue square). Shaded areas represent 95% confidence intervals for QR coefficients; dashed lines for OLS CI.

Impact of Lagged Negative Log Returns (`L_neg_log_ret` - Leverage Effect) The leverage effect, proxied by `L_neg_log_ret`, also demonstrates asset-specific patterns (Figure 4). For Bitcoin, past negative returns significantly increase subsequent volatility across all its quantiles, with the effect's magnitude slightly decreasing at higher volatility states. For NASDAQ, this effect is statistically significant primarily at its median (50th) volatility quantile, with less certainty at its distribution's tails. The Pooled OLS estimate aligns with some of these findings but does not capture the distinct patterns of significance and magnitude for each asset.

4.2.2 Summary of Insights from Asset-Specific Quantile Regression

In sum, asset-specific quantile regressions reveal that the impacts of financial indicators on volatility are frequently not uniform, as suggested by Pooled OLS. Instead, these effects often vary significantly between Bitcoin and NASDAQ and across their respective low, medium, and high volatility states. This granular view uncovers complex dynamics, such as credit and Treasury spreads sometimes having volatility-dampening effects, and differing magnitudes or consistency of implied volatility and leverage effects for each asset. Such nuanced relationships are crucial for a more complete understanding of risk dynamics in diverse markets.

5 Conclusion

This study demonstrates that asset-specific quantile regressions unveil markedly different volatility responses for Bitcoin and NASDAQ to market indicators like credit and Treasury spreads, often revealing conditional, volatility-dampening effects that contrast sharply with simpler pooled OLS estimates. While lagged implied volatility's positive impact on future volatility was substantially larger for NASDAQ than for Bitcoin, the leverage effect showed more consistent significance for Bitcoin across its volatility distribution compared to NASDAQ's more median-concentrated sensitivity. These distinct, state-dependent sensitivities highlight the critical importance of employing asset-specific, non-linear models to achieve a more accurate and nuanced understanding of risk transmission and volatility dynamics in diverse financial markets.

6 Do Files

6.1 Data Preparation and Regressions

```
*******************************************************************************
* Purpose: Analyze credit stress impact on asset volatility (LogVol_20d)
* for Bitcoin and NASDAQ. Includes panel diagnostics & regressions.
* Date: May 24, 2025
*******************************************************************************
log using "C:\Users\melin\OneDrive\Documents\FINAL PROJECT\\\
    REGS_asset_specific_minimal.smcl", replace
clear all
set more off

* --- 1. Setup and Data Import ---
ssc install asrol, replace

local data_path "C:\Users\melin\OneDrive\Documents\FINAL \\\
    PROJECT\DATA\"
local filename "bitcoin_dataset_clean.csv"
cd "`data_path'"
import delimited "`filename'", clear

* --- 2. Data Cleaning and Log Volatility Generation ---
gen date_stata = daily(date, "YMD")
format date_stata %td
drop date
rename date_stata date
sort date
tsset date

gen btc_log_ret = ln(btc_price / L.btc_price)
gen nasdaq_log_ret = ln(nasdaq_close / L.nasdaq_close)

asrol btc_log_ret, stat(sd) window(date 20) gen(btc_vol_raw_20d)
asrol nasdaq_log_ret, stat(sd) window(date 20) gen(nasdaq_vol_raw_20d)

gen btc_logvol_20d = ln(btc_vol_raw_20d * sqrt(252))
gen nasdaq_logvol_20d = ln(nasdaq_vol_raw_20d * sqrt(252))

* --- 3. Prepare Data (Reshape to Long Format) ---
rename btc_price btc_close_price
rename nasdaq_close nasdaq_close_price
rename btc_logvol_20d logvol_20d_btc
rename nasdaq_logvol_20d logvol_20d_nasdaq
rename btc_log_ret log_ret_btc
rename nasdaq_log_ret log_ret_nasdaq
rename btc_close_price close_price_btc
rename nasdaq_close_price close_price_nasdaq

gen implied_vol_btc = dvol_btc
gen implied_vol_nasdaq = vxn
drop dvol_btc vxn

reshape long logvol_20d log_ret close_price implied_vol, \\\
    i(date) j(asset_id) string

encode asset_id, gen(asset_numeric_id)
xtset asset_numeric_id date

* --- 4. Generate Lagged Explanatory Variables & Leverage Term ---
gen L_baa_aaa_spread = L.baa_aaa_spread
gen L_treasury_10y_3m_spread = L.treasury_10y_3m_spread
gen L_implied_vol = L.implied_vol
gen L_log_ret = L.log_ret
gen L_neg_log_ret = L_log_ret * (L_log_ret < 0)

* --- 4.A Diagnostic Tests for Panel Model Choice ---
display ""
display "--- DIAGNOSTIC TESTS FOR PANEL MODEL CHOICE (FULL PANEL) ---"
display "--- Fixed Effects Model (F-test for u_i=0) ---"
xtreg logvol_20d L_baa_aaa_spread L_treasury_10y_3m_spread \\\
```

```
    L_implied_vol L_neg_log_ret, fe

display ""
display "--- Random Effects Model ---"
xtreg logvol_20d L_baa_aaa_spread L_treasury_10y_3m_spread \\\
    L_implied_vol L_neg_log_ret, re
estimates store re_model_full_panel

display ""
display "--- Breusch-Pagan LM Test (after xtreg, re) ---"
xttest0

display ""
display "--- Hausman Test (FE vs. RE) ---"
quietly xtreg logvol_20d L_baa_aaa_spread L_treasury_10y_3m_spread \\\
    L_implied_vol L_neg_log_ret, fe
estimates store fe_model_full_panel
hausman fe_model_full_panel re_model_full_panel, sigmamore
display "--- END OF DIAGNOSTIC TESTS ---"
display ""

* --- 5. Model Estimation - ASSET SPECIFIC OLS ---
display ""
display "--- OLS Regression for BITCOIN Volatility ---"
regress logvol_20d L_baa_aaa_spread L_treasury_10y_3m_spread \\\
    L_implied_vol L_neg_log_ret if asset_id == "_btc", vce(robust)
estimates store ols_btc

display ""
display "--- OLS Regression for NASDAQ Volatility ---"
regress logvol_20d L_baa_aaa_spread L_treasury_10y_3m_spread \\\
    L_implied_vol L_neg_log_ret if asset_id == "_nasdaq", vce(robust)
estimates store ols_nasdaq

* --- 6. Innovation: Quantile Regression - ASSET SPECIFIC ---
display ""
display "--- Quantile Regression for BITCOIN Volatility (Q25, Q50, Q75) ---"
sqreg logvol_20d L_baa_aaa_spread L_treasury_10y_3m_spread \\\
    L_implied_vol L_neg_log_ret if asset_id == "_btc", \\\
        quantiles(25 50 75) reps(50)
estimates store sqreg_btc

display ""
display "--- Quantile Regression for NASDAQ Volatility (Q25, Q50, Q75) ---"
sqreg logvol_20d L_baa_aaa_spread L_treasury_10y_3m_spread \\\
    L_implied_vol L_neg_log_ret if asset_id == "_nasdaq", \\\
        quantiles(25 50 75) reps(50)
estimates store sqreg_nasdaq

* --- 7. Data Saving ---
compress
save "bitcoin_nasdaq_panel_analysis_ready.dta", replace
di "Analysis-ready dataset saved as bitcoin_nasdaq_panel \\\
    _analysis_ready.dta in `c(pwd)'"

log close
```

6.2 Data Preparation and Regressions: Log Output

```
------------------------------------------------------------------------------
      name:  <unnamed>
       log:  C:\Users\melin\OneDrive\Documents\FINAL PROJECT\REGS_asset_specific_
> minimal.smcl
  log type:  smcl
 opened on:  24 May 2025, 21:23:55

. clear all

. set more off

.
. * --- 1. Setup and Data Import ---
. ssc install asrol, replace
checking asrol consistency and verifying not already installed...
all files already exist and are up to date.

.
. local data_path "C:\Users\melin\OneDrive\Documents\FINAL PROJECT\DATA\"

. local filename "bitcoin_dataset_clean.csv"

. cd "`data_path'"
C:\Users\melin\OneDrive\Documents\FINAL PROJECT\DATA

. import delimited "`filename'", clear
(encoding automatically selected: ISO-8859-2)
(24 vars, 1,009 obs)

.
. * --- 2. Data Cleaning and Log Volatility Generation ---
. gen date_stata = daily(date, "YMD")

. format date_stata %td

. drop date

. rename date_stata date

. sort date

. tsset date

Time variable: date, 23mar2021 to 09apr2025, but with gaps
        Delta: 1 day

.
. gen btc_log_ret = ln(btc_price / L.btc_price)
(222 missing values generated)

. gen nasdaq_log_ret = ln(nasdaq_close / L.nasdaq_close)
(222 missing values generated)

.
. asrol btc_log_ret, stat(sd) window(date 20) gen(btc_vol_raw_20d)

. asrol nasdaq_log_ret, stat(sd) window(date 20) gen(nasdaq_vol_raw_20d)

.
. gen btc_logvol_20d = ln(btc_vol_raw_20d * sqrt(252))
(2 missing values generated)

. gen nasdaq_logvol_20d = ln(nasdaq_vol_raw_20d * sqrt(252))
(2 missing values generated)

.
. * --- 3. Prepare Data (Reshape to Long Format) ---
. rename btc_price btc_close_price

. rename nasdaq_close nasdaq_close_price

```

```
. rename btc_logvol_20d logvol_20d_btc

. rename nasdaq_logvol_20d logvol_20d_nasdaq

. rename btc_log_ret log_ret_btc

. rename nasdaq_log_ret log_ret_nasdaq

. rename btc_close_price close_price_btc

. rename nasdaq_close_price close_price_nasdaq

.
. gen implied_vol_btc = dvol_btc

. gen implied_vol_nasdaq = vxn

. drop dvol_btc vxn

.
. reshape long logvol_20d log_ret close_price implied_vol, i(date) j(asset_id) st
> ring
(j = _btc _nasdaq)

Data                                   Wide   ->   Long
-----------------------------------------------------------------------------
Number of observations              1,009   ->   2,018
Number of variables                    30   ->   27
j variable (2 values)                       ->   asset_id
xij variables:
       logvol_20d_btc logvol_20d_nasdaq   ->   logvol_20d
             log_ret_btc log_ret_nasdaq   ->   log_ret
     close_price_btc close_price_nasdaq   ->   close_price
     implied_vol_btc implied_vol_nasdaq   ->   implied_vol
-----------------------------------------------------------------------------

.
. encode asset_id, gen(asset_numeric_id)

. xtset asset_numeric_id date

Panel variable: asset_numeric_id (strongly balanced)
 Time variable: date, 23mar2021 to 09apr2025, but with gaps
         Delta: 1 day

.
. * --- 4. Generate Lagged Explanatory Variables & Leverage Term ---
. gen L_baa_aaa_spread = L.baa_aaa_spread
(444 missing values generated)

. gen L_treasury_10y_3m_spread = L.treasury_10y_3m_spread
(444 missing values generated)

. gen L_implied_vol = L.implied_vol
(444 missing values generated)

. gen L_log_ret = L.log_ret
(874 missing values generated)

. gen L_neg_log_ret = L_log_ret * (L_log_ret < 0)
(874 missing values generated)

.
. * --- 4.A Diagnostic Tests for Panel Model Choice ---
. display ""


. display "--- DIAGNOSTIC TESTS FOR PANEL MODEL CHOICE (FULL PANEL) ---"
--- DIAGNOSTIC TESTS FOR PANEL MODEL CHOICE (FULL PANEL) ---

. display "--- Fixed Effects Model (F-test for u_i=0) ---"
--- Fixed Effects Model (F-test for u_i=0) ---

```

```
. xtreg logvol_20d L_baa_aaa_spread L_treasury_10y_3m_spread L_implied_vol L_neg_
> log_ret, fe

Fixed-effects (within) regression               Number of obs     =      1,144
Group variable: asset_nume~d                    Number of groups  =          2

R-squared:                                      Obs per group:
     Within  = 0.3851                                         min =        572
     Between = 1.0000                                         avg =      572.0
     Overall = 0.6978                                         max =        572

                                                F(4, 1138)        =     178.18
corr(u_i, Xb) = 0.8380                          Prob > F          =     0.0000

--------------------------------------------------------------------------------
> ---------
              logvol_20d | Coefficient  Std. err.      t    P>|t|     [95% conf.
> interval]
-------------------------+------------------------------------------------------
> ---------
        L_baa_aaa_spread |   .2766598   .0521547     5.30   0.000     .1743296
>  .3789899
L_treasury_10y_3m_spread |  -.0048743   .0110083    -0.44   0.658    -.0264731
>  .0167245
           L_implied_vol |   .0180661   .0009078    19.90   0.000      .016285
>  .0198472
           L_neg_log_ret |  -2.617046   .6143316    -4.26   0.000    -3.822396
> -1.411696
                   _cons |  -2.276626   .0560846   -40.59   0.000    -2.386667
> -2.166585
-------------------------+------------------------------------------------------
> ---------
                 sigma_u |  .03804034
                 sigma_e |  .31900741
                     rho |  .01402023   (fraction of variance due to u_i)
--------------------------------------------------------------------------------
> ---------
F test that all u_i=0: F(1, 1138) = 1.64                     Prob > F = 0.2002

.
. display ""


. display "--- Random Effects Model ---"
--- Random Effects Model ---

. xtreg logvol_20d L_baa_aaa_spread L_treasury_10y_3m_spread L_implied_vol L_neg_
> log_ret, re

Random-effects GLS regression                   Number of obs     =      1,144
Group variable: asset_nume~d                    Number of groups  =          2

R-squared:                                      Obs per group:
     Within  = 0.3848                                         min =        572
     Between = 1.0000                                         avg =      572.0
     Overall = 0.6980                                         max =        572

                                                Wald chi2(4)      =    2632.86
corr(u_i, X) = 0 (assumed)                      Prob > chi2       =     0.0000

--------------------------------------------------------------------------------
> ---------
              logvol_20d | Coefficient  Std. err.      z    P>|z|     [95% conf.
> interval]
-------------------------+------------------------------------------------------
> ---------
        L_baa_aaa_spread |   .2677381   .0517026     5.18   0.000     .1664029
>  .3690734
L_treasury_10y_3m_spread |  -.0125939   .0092167    -1.37   0.172    -.0306583
>  .0054706
           L_implied_vol |   .0190995   .0004171    45.79   0.000      .018282
>   .019917
           L_neg_log_ret |  -2.568606   .6133408    -4.19   0.000    -3.770732
```

```
>  -1.36648
                _cons |  -2.314421   .0477185   -48.50   0.000    -2.407948
> -2.220895
-------------+----------------------------------------------------------------
> ---------
              sigma_u |          0
              sigma_e |  .31900741
                  rho |          0   (fraction of variance due to u_i)
------------------------------------------------------------------------------
> ---------

. estimates store re_model_full_panel

.
. display ""


. display "--- Breusch-Pagan LM Test (after xtreg, re) ---"
--- Breusch-Pagan LM Test (after xtreg, re) ---

. xttest0

Breusch and Pagan Lagrangian multiplier test for random effects

        logvol_20d[asset_numeric_id,t] = Xb + u[asset_numeric_id] + e[asset_numer
> ic_id,t]

        Estimated results:
                         |       Var     SD = sqrt(Var)
                ---------+-----------------------------
               logvo~20d |   .3360129       .5796662
                       e |   .1017657       .3190074
                       u |          0              0

        Test: Var(u) = 0
                             chibar2(01) =     0.00
                          Prob > chibar2 =   1.0000

.
. display ""


. display "--- Hausman Test (FE vs. RE) ---"
--- Hausman Test (FE vs. RE) ---

. quietly xtreg logvol_20d L_baa_aaa_spread L_treasury_10y_3m_spread L_implied_vo
> l L_neg_log_ret, fe

. estimates store fe_model_full_panel

. hausman fe_model_full_panel re_model_full_panel, sigmamore

Note: the rank of the differenced variance matrix (1) does not equal the number
        of coefficients being tested (4); be sure this is what you expect, or
        there may be problems computing the test.  Examine the output of your
        estimators for anything unexpected and possibly consider scaling your
        variables so that the coefficients are on a similar scale.

                 ---- Coefficients ----
             |      (b)          (B)            (b-B)     sqrt(diag(V_b-V_B))
             |  fe_model_f~l re_model_f~l    Difference       Std. err.
-------------+----------------------------------------------------------------
L_baa_aaa_~d |    .2766598     .2677381        .0089216        .0069633
L_treasury~d |   -.0048743    -.0125939        .0077196        .0060251
L_implied_~l |    .0180661     .0190995       -.0010334        .0008066
L_neg_log_~t |   -2.617046    -2.568606       -.0484395         .037807
------------------------------------------------------------------------------
                          b = Consistent under H0 and Ha; obtained from xtreg.
           B = Inconsistent under Ha, efficient under H0; obtained from xtreg.

Test of H0: Difference in coefficients not systematic

    chi2(1) = (b-B)'[(V_b-V_B)^(-1)](b-B)
```

```
               =    1.64
Prob > chi2 = 0.2001
(V_b-V_B is not positive definite)

. display "--- END OF DIAGNOSTIC TESTS ---"
--- END OF DIAGNOSTIC TESTS ---

. display ""


.
. * --- 5. Model Estimation - ASSET SPECIFIC OLS ---
. display ""


. display "--- OLS Regression for BITCOIN Volatility ---"
--- OLS Regression for BITCOIN Volatility ---

. regress logvol_20d L_baa_aaa_spread L_treasury_10y_3m_spread L_implied_vol L_ne
> g_log_ret if asset_id == "_btc", vce(robust)

Linear regression                               Number of obs     =        572
                                                F(4, 567)         =     117.37
                                                Prob > F          =     0.0000
                                                R-squared         =     0.4643
                                                Root MSE          =     .28664

--------------------------------------------------------------------------------
> ---------
                         |               Robust
              logvol_20d | Coefficient  std. err.      t    P>|t|     [95% conf.
> interval]
-------------------------+------------------------------------------------------
> ---------
        L_baa_aaa_spread |   -.184296   .0648807    -2.84   0.005    -.3117318
> -.0568602
L_treasury_10y_3m_spread |  -.0403694    .014851    -2.72   0.007    -.0695392
> -.0111996
           L_implied_vol |   .0153455   .0009759    15.72   0.000     .0134287
>  .0172624
           L_neg_log_ret |  -1.797929   .5312523    -3.38   0.001    -2.841392
> -.7544664
                   _cons |   -1.66431   .0842858   -19.75   0.000     -1.82986
> -1.498759
--------------------------------------------------------------------------------
> ---------

. estimates store ols_btc

.
. display ""


. display "--- OLS Regression for NASDAQ Volatility ---"
--- OLS Regression for NASDAQ Volatility ---

. regress logvol_20d L_baa_aaa_spread L_treasury_10y_3m_spread L_implied_vol L_ne
> g_log_ret if asset_id == "_nasdaq", vce(robust)

Linear regression                               Number of obs     =        572
                                                F(4, 567)         =     211.36
                                                Prob > F          =     0.0000
                                                R-squared         =     0.5989
                                                Root MSE          =     .26778

--------------------------------------------------------------------------------
> ---------
                         |               Robust
              logvol_20d | Coefficient  std. err.      t    P>|t|     [95% conf.
> interval]
-------------------------+------------------------------------------------------
> ---------
        L_baa_aaa_spread |  -.1294672   .0829026    -1.56   0.119     -.292301
```

```
>  .0333665
L_treasury_10y_3m_spread |   -.101856   .0158702    -6.42   0.000    -.1330276
> -.0706844
           L_implied_vol |    .065843   .0033961    19.39   0.000     .0591725
>  .0725134
           L_neg_log_ret |  -.4085538   1.115485    -0.37   0.714    -2.599542
>  1.782434
                   _cons |  -3.079702   .0596557   -51.62   0.000    -3.196875
> -2.962529
--------------------------------------------------------------------------------
> ---------

. estimates store ols_nasdaq

.
. * --- 6. Innovation: Quantile Regression - ASSET SPECIFIC ---
. display ""


. display "--- Quantile Regression for BITCOIN Volatility (Q25, Q50, Q75) ---"
--- Quantile Regression for BITCOIN Volatility (Q25, Q50, Q75) ---

. sqreg logvol_20d L_baa_aaa_spread L_treasury_10y_3m_spread L_implied_vol L_neg_
> log_ret if asset_id == "_btc", quantiles(25 50 75) reps(50)
(fitting base model)

Bootstrap replications (50): .........10.........20.........30.........40.......
> .50 done

Simultaneous quantile regression                    Number of obs =        572
  bootstrap(50) SEs                                 .25 Pseudo R2 =     0.3142
                                                    .50 Pseudo R2 =     0.2483
                                                    .75 Pseudo R2 =     0.2408

--------------------------------------------------------------------------------
> ---------
                         |              Bootstrap
              logvol_20d | Coefficient  std. err.      t    P>|t|     [95% conf.
> interval]
-------------------------+------------------------------------------------------
> ---------
q25                      |
        L_baa_aaa_spread |  -.2241067   .0762805    -2.94   0.003    -.3739336
> -.0742799
L_treasury_10y_3m_spread |  -.0208172   .0226941    -0.92   0.359    -.0653919
>  .0237575
           L_implied_vol |   .0158764   .0011615    13.67   0.000      .013595
>  .0181578
           L_neg_log_ret |  -2.535739   .7129643    -3.56   0.000    -3.936112
> -1.135365
                   _cons |   -1.88291   .1003566   -18.76   0.000    -2.080026
> -1.685794
-------------------------+------------------------------------------------------
> ---------
q50                      |
        L_baa_aaa_spread |  -.1458748   .0737999    -1.98   0.049    -.2908294
> -.0009201
L_treasury_10y_3m_spread |  -.0255172   .0219918    -1.16   0.246    -.0687126
>  .0176781
           L_implied_vol |   .0147713   .0011934    12.38   0.000     .0124273
>  .0171154
           L_neg_log_ret |  -2.177763   .5984748    -3.64   0.000    -3.353261
> -1.002264
                   _cons |  -1.671142   .0974686   -17.15   0.000    -1.862585
> -1.479698
-------------------------+------------------------------------------------------
> ---------
q75                      |
        L_baa_aaa_spread |  -.1830934   .1007274    -1.82   0.070    -.3809377
>  .0147509
L_treasury_10y_3m_spread |  -.0518503   .0128282    -4.04   0.000    -.0770468
> -.0266537
           L_implied_vol |   .0136191   .0013634     9.99   0.000     .0109413
```

```
>   .016297
           L_neg_log_ret |  -1.866006   .5747911    -3.25   0.001    -2.994986
> -.7370268
                   _cons |  -1.349618   .1411836    -9.56   0.000    -1.626924
> -1.072311
--------------------------------------------------------------------------------
> ---------

. estimates store sqreg_btc

.
. display ""


. display "--- Quantile Regression for NASDAQ Volatility (Q25, Q50, Q75) ---"
--- Quantile Regression for NASDAQ Volatility (Q25, Q50, Q75) ---

. sqreg logvol_20d L_baa_aaa_spread L_treasury_10y_3m_spread L_implied_vol L_neg_
> log_ret if asset_id == "_nasdaq", quantiles(25 50 75) reps(50)
(fitting base model)

Bootstrap replications (50): .........10.........20.........30.........40.......
> .50 done

Simultaneous quantile regression                     Number of obs =        572
  bootstrap(50) SEs                                  .25 Pseudo R2 =     0.3808
                                                     .50 Pseudo R2 =     0.3667
                                                     .75 Pseudo R2 =     0.4032

--------------------------------------------------------------------------------
> ---------
                         |               Bootstrap
              logvol_20d | Coefficient  std. err.      t    P>|t|     [95% conf.
> interval]
-------------------------+------------------------------------------------------
> ---------
q25                      |
         L_baa_aaa_spread |   .0457559   .1719271     0.27   0.790    -.2919359
>  .3834476
L_treasury_10y_3m_spread |  -.1150368   .0276333    -4.16   0.000    -.1693129
> -.0607608
           L_implied_vol |   .0654167    .004749    13.77   0.000      .056089
>  .0747444
           L_neg_log_ret |  -1.648527   2.171402    -0.76   0.448      -5.9135
>  2.616446
                   _cons |  -3.411395   .0984948   -34.64   0.000    -3.604854
> -3.217936
-------------------------+------------------------------------------------------
> ---------
q50                      |
         L_baa_aaa_spread |  -.1751987   .1044568    -1.68   0.094    -.3803682
>  .0299709
L_treasury_10y_3m_spread |  -.0952706   .0213874    -4.45   0.000    -.1372788
> -.0532624
           L_implied_vol |    .065382   .0031351    20.85   0.000     .0592242
>  .0715398
           L_neg_log_ret |  -2.693149   1.586661    -1.70   0.090    -5.809601
>  .4233017
                   _cons |  -3.031873   .0922355   -32.87   0.000    -3.213038
> -2.850708
-------------------------+------------------------------------------------------
> ---------
q75                      |
         L_baa_aaa_spread |  -.2250522   .0910275    -2.47   0.014    -.4038444
>   -.04626
L_treasury_10y_3m_spread |  -.0575422   .0252523    -2.28   0.023    -.1071418
> -.0079427
           L_implied_vol |     .05591   .0053965    10.36   0.000     .0453103
>  .0665096
           L_neg_log_ret |  -.6752768   1.096396    -0.62   0.538    -2.828771
>  1.478218
                   _cons |  -2.567685   .0713248   -36.00   0.000    -2.707778
> -2.427592
```

```
----------------------------------------------------------------------------------
> ---------

. estimates store sqreg_nasdaq

.
. * --- 7. Data Saving ---
. compress
  variable date was float now int
  variable asset_numeric_id was long now byte
  (10,090 bytes saved)

. save "bitcoin_nasdaq_panel_analysis_ready_minimal_comments.dta", replace
(file bitcoin_nasdaq_panel_analysis_ready_minimal_comments.dta not found)
file bitcoin_nasdaq_panel_analysis_ready_minimal_comments.dta saved

. di "Analysis-ready dataset saved as bitcoin_nasdaq_panel_analysis_ready_minimal
> _comments.dta in `c(pwd)'"
Analysis-ready dataset saved as bitcoin_nasdaq_panel_analysis_ready_minimal_comme
> nts.dta in C:\Users\melin\OneDrive\Documents\FINAL PROJECT\DATA

.
. log close
----------------------------------------------------------------------------------
```

6.3 Tables

```
********************************************************************************
* Purpose: Generate summary statistics and regression tables (LaTeX).
* Uses: Analysis-ready dataset from 01_data_prep...do
* Date: May 24, 2025
********************************************************************************

log using "C:\Users\melin\OneDrive\Documents\FINAL PROJECT\TABLES_ \\\
    asset_specific_minimal.smcl", replace
clear all
set more off

* --- 0. Install necessary packages ---
ssc install estout, replace
ssc install logout, replace

* --- 1. Define Paths and Load Processed Data ---
local project_base_path "C:\Users\melin\OneDrive\Documents\FINAL PROJECT\"
local data_path "`project_base_path'DATA\"
local tables_path "`project_base_path'TABLES\"
local analysis_file "bitcoin_nasdaq_panel_analysis_ready_long_with_diagnostics.dta"

capture mkdir "`tables_path'"
cd "`data_path'"
use "`analysis_file'", clear

* --- 2. Generate Descriptive Statistics Tables (.tex) ---

display ""
display "--- Generating: Table_Descriptive_Overall.tex ---"
estpost summarize logvol_20d L_baa_aaa_spread L_treasury_10y_3m_spread L_implied_vol L_neg_log_ret, listwise
esttab . using "`tables_path'Table_Descriptive_Overall.tex", ///
    cells("mean(fmt(%9.3f)) sd(fmt(%9.3f)) min(fmt(%9.3f)) max(fmt(%9.3f)) count(fmt(%9.0gc)
    label(N))") ///
    replace booktabs nonumber nomtitles label ///
    title("Overall Descriptive Statistics of Key Variables") ///
    addnote("Statistics are for the 1,144 observations used in regression analysis.")

display ""
display "--- Generating: Table_Descriptive_ByAsset.tex (table body) ---"
logout, save("`tables_path'Table_Descriptive_ByAsset") tex replace : ///
    tabstat logvol_20d L_implied_vol L_neg_log_ret, by(asset_id) stats(n mean sd min max)
    format(%9.3f) columns(stats)

display ""
display "--- Generating: Table_Descriptive_Panel.tex (table body) ---"
logout, save("`tables_path'Table_Descriptive_Panel") tex replace : ///
    xtsum logvol_20d L_baa_aaa_spread L_treasury_10y_3m_spread L_implied_vol L_neg_log_ret

* --- 3. Re-run Models for Regression Table Generation ---
// Pooled OLS - using xtreg
xtreg logvol_20d L_baa_aaa_spread L_treasury_10y_3m_spread \\\
    L_implied_vol L_neg_log_ret, re
estimates store ols_pooled_tab

// Quantile Regression for Bitcoin
sqreg logvol_20d L_baa_aaa_spread L_treasury_10y_3m_spread \\\
    L_implied_vol L_neg_log_ret if asset_id == "_btc", quantiles(25 50 75) reps(50)
estimates store qreg_btc_tab

// Quantile Regression for NASDAQ
sqreg logvol_20d L_baa_aaa_spread L_treasury_10y_3m_spread \\\
    L_implied_vol L_neg_log_ret if asset_id == "_nasdaq", quantiles(25 50 75) reps(50)
estimates store qreg_nasdaq_tab

* --- 4. Generate Formatted Regression Tables (.tex) ---

display ""
display "--- Generating: Table_OLS_Log_Volatility.tex (Pooled OLS) ---"
// Note: esttab for xtreg, re will show Wald chi2 instead of F-stat, and overall R2.
esttab ols_pooled_tab using "`tables_path'Table_OLS_Log_Volatility.tex", ///
```

```
    replace booktabs ///
    b(%9.3f) se(%9.3f) ///
    title("Pooled OLS Regression of Log Volatility (GLS SEs from RE Model)") ///
    keep(L_baa_aaa_spread L_treasury_10y_3m_spread L_implied_vol L_neg_log_ret _cons) ///
    stats(r2_o N, fmt(%9.3f %9.0gc) labels("Overall R-squared" "Observations")) ///
    starlevels(* 0.10 ** 0.05 *** 0.001) ///
    mgroups("Log Volatility (20-day)", pattern(1) prefix(\multicolumn{@span}{c}{) \\\
        suffix(}) span erepeat(\cmidrule(lr){@span})) ///
    nonumbers nodepvars nomtitles ///
    addnote("Standard errors in parentheses. Data source: `analysis_file'. \\\
        Variables are lagged one period.")

display ""
display "--- Generating: Table_Quantile_Bitcoin.tex ---"
esttab qreg_btc_tab using "`tables_path'Table_Quantile_Bitcoin.tex", ///
    replace booktabs unstack ///
    b(%9.3f) se(%9.3f) ///
    title("Quantile Regression of Log Volatility for Bitcoin") ///
    keep(L_baa_aaa_spread L_treasury_10y_3m_spread L_implied_vol L_neg_log_ret _cons) ///
    stats(N r2_p_q1 r2_p_q2 r2_p_q3, ///
          fmt(%9.0gc %9.3f %9.3f %9.3f) ///
          labels("Observations" "Pseudo R-sq (Q25)" "Pseudo R-sq (Q50)" "Pseudo R-sq (Q75)") ///
          nostar) ///
    starlevels(* 0.10 ** 0.05 *** 0.001) ///
    mtitles("Q25 (BTC)" "Q50 (BTC)" "Q75 (BTC)") ///
    nonumbers nodepvars ///
    addnote("Standard errors in parentheses. Data source: `analysis_file'. \\\
        Variables are lagged. reps(50) for SEs.")

display ""
display "--- Generating: Table_Quantile_NASDAQ.tex ---"
esttab qreg_nasdaq_tab using "`tables_path'Table_Quantile_NASDAQ.tex", ///
    replace booktabs unstack ///
    b(%9.3f) se(%9.3f) ///
    title("Quantile Regression of Log Volatility for NASDAQ") ///
    keep(L_baa_aaa_spread L_treasury_10y_3m_spread L_implied_vol L_neg_log_ret _cons) ///
    stats(N r2_p_q1 r2_p_q2 r2_p_q3, ///
          fmt(%9.0gc %9.3f %9.3f %9.3f) ///
          labels("Observations" "Pseudo R-sq (Q25)" "Pseudo R-sq (Q50)" \\\
            "Pseudo R-sq (Q75)") ///
          nostar) ///
    starlevels(* 0.10 ** 0.05 *** 0.001) ///
    mtitles("Q25 (NASDAQ)" "Q50 (NASDAQ)" "Q75 (NASDAQ)") ///
    nonumbers nodepvars ///
    addnote("Standard errors in parentheses. Data source: `analysis_file'. \\\
        Variables are lagged. reps(50) for SEs.")

* --- End of Script ---
capture log close
```

6.4 Tables: Log Output

```
-------------------------------------------------------------------------------
      name:  <unnamed>
       log:  C:\Users\melin\OneDrive\Documents\FINAL PROJECT\TABLES_asset_specific
> _minimal.smcl
  log type:  smcl
 opened on:  25 May 2025, 12:03:31

. clear all

. set more off

.
. * --- 0. Install necessary packages ---
. ssc install estout, replace
checking estout consistency and verifying not already installed...
all files already exist and are up to date.

. ssc install logout, replace
checking logout consistency and verifying not already installed...
all files already exist and are up to date.

.
. * --- 1. Define Paths and Load Processed Data ---
. local project_base_path "C:\Users\melin\OneDrive\Documents\FINAL PROJECT\"

. local data_path "`project_base_path'DATA\"

. local tables_path "`project_base_path'TABLES\"

. local analysis_file "bitcoin_nasdaq_panel_analysis_ready_long_with_diagnostics.d
> ta"

.
. capture mkdir "`tables_path'"

. cd "`data_path'"
C:\Users\melin\OneDrive\Documents\FINAL PROJECT\DATA

. use "`analysis_file'", clear

.
. * --- 2. Generate Descriptive Statistics Tables (.tex) ---
. * These commands will use the loaded '`analysis_file`' as is.
. * The 'listwise' option for estpost summarize will use observations where all
. * variables *in that specific command* are non-missing.
. * tabstat and xtsum will use all available observations for the variables they p
> rocess.
.
. display ""


. display "--- Generating: Table_Descriptive_Overall.tex ---"
--- Generating: Table_Descriptive_Overall.tex ---

. estpost summarize logvol_20d L_baa_aaa_spread L_treasury_10y_3m_spread L_implied
> _vol L_neg_log_ret, listwise

             |  e(count)   e(sum_w)    e(mean)     e(Var)      e(sd)     e(min)
>   e(max)     e(sum)
-------------+-----------------------------------------------------------------
> --------------------
  logvol_20d |      1144       1144  -1.207717   .3360129   .5796662  -2.803345
> .2114888  -1381.628
L_baa_aaa_~d |      1144       1144   .8560839   .0383888   .1959307        .58
>     1.23     979.36
L_treasury~d |      1144       1144   .0602972   1.350495   1.162108      -1.73
>     2.21      68.98
L_implied_~l |      1144       1144   44.82028   619.0489   24.88069       11.2
>   139.98    51274.4
L_neg_log_~t |      1144       1144  -.0086474   .0002582   .0160672  -.1548894
>        0  -9.892644
```

```
. esttab . using "`tables_path'Table_Descriptive_Overall.tex", ///
>     cells("mean(fmt(%9.3f)) sd(fmt(%9.3f)) min(fmt(%9.3f)) max(fmt(%9.3f)) count
> (fmt(%9.0gc) label(N))") ///
>     replace booktabs nonumber nomtitles label ///
>     title("Overall Descriptive Statistics of Key Variables") ///
>     addnote("Statistics are for the 1,144 observations used in regression analys
> is.")
(output written to C:\Users\melin\OneDrive\Documents\FINAL PROJECT\TABLES\Table_De
> scriptive_Overall.tex)

.
. display ""


. display "--- Generating: Table_Descriptive_ByAsset.tex (table body) ---"
--- Generating: Table_Descriptive_ByAsset.tex (table body) ---

. logout, save("`tables_path'Table_Descriptive_ByAsset") tex replace : ///
>     tabstat logvol_20d L_implied_vol L_neg_log_ret, by(asset_id) stats(n mean sd
>  min max) format(%9.3f) columns(stats)
```

6.5 Plots: Coded in Python

```python
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import os

# Common settings
critval = 1.959963984540054 # invnormal(0.975)
pdf_width_inches = 7
pdf_height_inches = 4.2 # Adjusted slightly for potentially more legend items
png_dpi = 200

# Plotting function to avoid code repetition
def create_coefficient_plot(data_dict, var_name, y_axis_label_text, plot_title, plot_subtitle,
                            output_filename_base, figures_path="."):
    """
    Generates and saves a coefficient plot showing asset-specific QR and pooled OLS.
    """
    df = pd.DataFrame(data_dict)
    df['ci_low'] = df['coef'] - critval * df['se']
    df['ci_high'] = df['coef'] + critval * df['se']

    qreg_btc_df = df[df['model_type'] == 'QReg_BTC'].sort_values(by='x_plot_val')
    qreg_nas_df = df[df['model_type'] == 'QReg_NAS'].sort_values(by='x_plot_val')
    ols_df = df[df['model_type'] == 'OLS']

    fig, ax = plt.subplots(figsize=(pdf_width_inches, pdf_height_inches))

    # Bitcoin Quantile Regression
    if not qreg_btc_df.empty:
        ax.fill_between(qreg_btc_df['x_plot_val'], qreg_btc_df['ci_low'], qreg_btc_df['ci_high'],
                        color='red', alpha=0.1, label='Bitcoin QR 95% CI')
        ax.plot(qreg_btc_df['x_plot_val'], qreg_btc_df['coef'], color='red', linestyle='-', linewidth=1.5,
                marker='D', markersize=7, label='Bitcoin QR Coef.')

    # NASDAQ Quantile Regression
    if not qreg_nas_df.empty:
        ax.fill_between(qreg_nas_df['x_plot_val'], qreg_nas_df['ci_low'], qreg_nas_df['ci_high'],
                        color='green', alpha=0.1, label='NASDAQ QR 95% CI')
        ax.plot(qreg_nas_df['x_plot_val'], qreg_nas_df['coef'], color='green', linestyle='--', linewidth=1.5,
                marker='o', markersize=7, label='NASDAQ QR Coef.')

    # OLS Regression
    if not ols_df.empty:
        ols_point = ols_df.iloc[0]
        ols_x_pos = ols_point['x_plot_val']

        ax.plot(ols_x_pos, ols_point['coef'], color='darkblue',
                marker='s', markersize=7, label='Pooled OLS Coef.', linestyle='None')
        ax.hlines(y=ols_point['ci_low'], xmin=ols_x_pos - 0.1, xmax=ols_x_pos + 0.1,
                  color='darkblue', alpha=0.7, linestyle='--', linewidth=1)
        ax.hlines(y=ols_point['ci_high'], xmin=ols_x_pos - 0.1, xmax=ols_x_pos + 0.1,
                  color='darkblue', alpha=0.7, linestyle='--', linewidth=1)
        ax.hlines(y=ols_point['coef'], xmin=ols_x_pos - 0.05, xmax=ols_x_pos + 0.05,
                  color='darkblue', linestyle='-', linewidth=2)

        if "implied_vol" in output_filename_base:
            mlabformat_str = "{:.4f}"
            mlab_x_offset = 0.18
        elif "neg_log_ret" in output_filename_base:
            mlabformat_str = "{:.2f}"
            mlab_x_offset = -0.18
        elif "treasury_spread" in output_filename_base: # Specific formatting for treasury spread
            mlabformat_str = "{:.3f}"
            mlab_x_offset = 0.18
        else: # baa_aaa_spread (default)
            mlabformat_str = "{:.3f}"
```

```
            mlab_x_offset = 0.18

        ax.text(ols_x_pos + mlab_x_offset, ols_point['coef'],
                mlabformat_str.format(ols_point['coef']),
                verticalalignment='center', horizontalalignment='left' if mlab_x_offset > 0 else 'right',
                fontsize=8, color='black')

    all_y_values_list = []
    if not qreg_btc_df.empty:
        all_y_values_list.extend([qreg_btc_df['ci_low'], qreg_btc_df['ci_high']])
    if not qreg_nas_df.empty:
        all_y_values_list.extend([qreg_nas_df['ci_low'], qreg_nas_df['ci_high']])
    if not ols_df.empty:
        all_y_values_list.extend([ols_df['ci_low'], ols_df['ci_high']])

    if all_y_values_list:
        all_y_values = pd.concat(all_y_values_list)
        min_y = all_y_values.min()
        max_y = all_y_values.max()
        padding = (max_y - min_y) * 0.10
        ax.set_ylim(min_y - padding, max_y + padding)
    else:
        ax.set_ylim(-1, 1)

    ax.set_xticks([1, 2, 3])
    ax.set_xticklabels(["Q25", "Q50 / Pooled OLS", "Q75"], fontsize=9)
    ax.set_xlim(0.7, 3.3)
    ax.tick_params(axis='x', which='major', length=0)

    ax.set_title(plot_title, fontsize=11, loc='center', pad=20)
    fig.text(0.5, 0.93, plot_subtitle, ha='center', fontsize=9, color='dimgray')

    ax.set_ylabel(y_axis_label_text, fontsize=9)
    ax.yaxis.grid(True, linestyle=':', color='lightgrey', alpha=0.7)
    handles, labels = ax.get_legend_handles_labels()

    desired_order_map = {
        'Bitcoin QR Coef.': 0, 'Bitcoin QR 95% CI': 1,
        'NASDAQ QR Coef.': 2, 'NASDAQ QR 95% CI': 3,
        'Pooled OLS Coef.': 4
    }
    available_labels_in_order = [lbl for lbl in desired_order_map if lbl in labels]
    ordered_handles = [handles[labels.index(lbl)] for lbl in available_labels_in_order]
    ordered_labels = available_labels_in_order

    if ordered_handles:
        ax.legend(ordered_handles, ordered_labels, loc='best',
                  fontsize=7, frameon=False, ncol=1)

    ax.set_facecolor('white')
    fig.set_facecolor('white')
    for spine in ['top', 'right']:
        ax.spines[spine].set_visible(False)
    for spine in ['left', 'bottom']:
        ax.spines[spine].set_color('black')
        ax.spines[spine].set_linewidth(0.5)

    plt.tight_layout(rect=[0, 0, 1, 0.90])

    if not os.path.exists(figures_path):
        os.makedirs(figures_path)
    pdf_file = os.path.join(figures_path, f"{output_filename_base}.pdf")
    png_file = os.path.join(figures_path, f"{output_filename_base}.png")

    plt.savefig(pdf_file, bbox_inches='tight')
    plt.savefig(png_file, dpi=png_dpi, bbox_inches='tight')
    plt.close(fig)
    print(f"Exported {output_filename_base} (PDF and PNG) to {figures_path}")

# --- Data and Calls for Each Plot ---

figures_path = r"C:\Users\melin\OneDrive\Documents\FINAL PROJECT\FIGURES"
```

```python

ols_coeffs = {
    "L_baa_aaa_spread": 0.2677381,
    "L_treasury_10y_3m_spread": -0.013,
    "L_implied_vol": 0.0190995,
    "L_neg_log_ret": -2.568606
}
ols_ses = {
    "L_baa_aaa_spread": 0.0500451,
    "L_treasury_10y_3m_spread": 0.010,
    "L_implied_vol": 0.0004262,
    "L_neg_log_ret": 0.5771519
}

btc_qr_coeffs = {
    "L_baa_aaa_spread": [-0.2241067, -0.1458748, -0.1830934],
    "L_treasury_10y_3m_spread": [-0.0208172, -0.0255172, -0.0518503],
    "L_implied_vol": [0.0158764, 0.0147713, 0.0136191],
    "L_neg_log_ret": [-2.535739, -2.177763, -1.866006]
}
btc_qr_ses = {
    "L_baa_aaa_spread": [0.0777605, 0.0831789, 0.1208323],
    "L_treasury_10y_3m_spread": [0.0228912, 0.0207246, 0.0139991],
    "L_implied_vol": [0.0013332, 0.0014256, 0.0013526],
    "L_neg_log_ret": [0.8630549, 0.8622337, 0.7911354]
}

nasdaq_qr_coeffs = {
    "L_baa_aaa_spread": [0.0457559, -0.1751987, -0.2250522],
    "L_treasury_10y_3m_spread": [-0.1150368, -0.0952706, -0.0575422],
    "L_implied_vol": [0.0654167, 0.065382, 0.05591],
    "L_neg_log_ret": [-1.648527, -2.693149, -0.6752768]
}
nasdaq_qr_ses = {
    "L_baa_aaa_spread": [0.1746449, 0.1119737, 0.1084846],
    "L_treasury_10y_3m_spread": [0.0293638, 0.0257964, 0.0263304],
    "L_implied_vol": [0.0058218, 0.0044505, 0.0061582],
    "L_neg_log_ret": [2.122272, 1.180048, 1.230968]
}

# 1. Plot for L_baa_aaa_spread
var_key_baa = "L_baa_aaa_spread" # Renamed for clarity
data_baa_updated = {
    'varname': [var_key_baa]*7,
    'model_type': ["QReg_BTC"]*3 + ["QReg_NAS"]*3 + ["OLS"],
    'x_plot_val': [1, 2, 3, 1, 2, 3, 2],
    'actual_quantile': [0.25, 0.50, 0.75, 0.25, 0.50, 0.75, 0.50],
    'coef': btc_qr_coeffs[var_key_baa] + nasdaq_qr_coeffs[var_key_baa] +
    [ols_coeffs[var_key_baa]],
    'se': btc_qr_ses[var_key_baa] + nasdaq_qr_ses[var_key_baa] + [ols_ses[var_key_baa]]
}
create_coefficient_plot(data_dict=data_baa_updated,
                        var_name=var_key_baa,
                        y_axis_label_text="Coefficient Estimate",
                        plot_title="Impact of Lagged Baa-Aaa Spread",
                        plot_subtitle="Bitcoin QR vs. NASDAQ QR vs. Pooled OLS",
                        output_filename_base="py_plot_L_baa_aaa_spread",
                        figures_path=figures_path)

# 2. Plot for L_implied_vol
var_key_imp = "L_implied_vol"
data_imp_updated = {
    'varname': [var_key_imp]*7,
    'model_type': ["QReg_BTC"]*3 + ["QReg_NAS"]*3 + ["OLS"],
    'x_plot_val': [1, 2, 3, 1, 2, 3, 2],
    'actual_quantile': [0.25, 0.50, 0.75, 0.25, 0.50, 0.75, 0.50],
    'coef': btc_qr_coeffs[var_key_imp] + nasdaq_qr_coeffs[var_key_imp] +
    [ols_coeffs[var_key_imp]],
    'se': btc_qr_ses[var_key_imp] + nasdaq_qr_ses[var_key_imp] + [ols_ses[var_key_imp]]
}
create_coefficient_plot(data_dict=data_imp_updated,
                        var_name=var_key_imp,
                        y_axis_label_text="Coefficient Estimate",
```

```python
                        plot_title="Impact of Lagged Implied Volatility",
                        plot_subtitle="Bitcoin QR vs. NASDAQ QR vs. Pooled OLS",
                        output_filename_base="py_plot_L_implied_vol",
                        figures_path=figures_path)

# 3. Plot for L_neg_log_ret
var_key_neg = "L_neg_log_ret"
data_neg_updated = {
    'varname': [var_key_neg]*7,
    'model_type': ["QReg_BTC"]*3 + ["QReg_NAS"]*3 + ["OLS"],
    'x_plot_val': [1, 2, 3, 1, 2, 3, 2],
    'actual_quantile': [0.25, 0.50, 0.75, 0.25, 0.50, 0.75, 0.50],
    'coef': btc_qr_coeffs[var_key_neg] + nasdaq_qr_coeffs[var_key_neg] +
    [ols_coeffs[var_key_neg]],
    'se': btc_qr_ses[var_key_neg] + nasdaq_qr_ses[var_key_neg] + [ols_ses[var_key_neg]]
}
create_coefficient_plot(data_dict=data_neg_updated,
                        var_name=var_key_neg,
                        y_axis_label_text="Coefficient Estimate",
                        plot_title="Impact of Lagged Negative Log Returns (Leverage)",
                        plot_subtitle="Bitcoin QR vs. NASDAQ QR vs. Pooled OLS",
                        output_filename_base="py_plot_L_neg_log_ret",
                        figures_path=figures_path)

# 4. Plot for L_treasury_10y_3m_spread
var_key_tsy = "L_treasury_10y_3m_spread"
data_tsy_updated = {
    'varname': [var_key_tsy]*7,
    'model_type': ["QReg_BTC"]*3 + ["QReg_NAS"]*3 + ["OLS"],
    'x_plot_val': [1, 2, 3, 1, 2, 3, 2],
    'actual_quantile': [0.25, 0.50, 0.75, 0.25, 0.50, 0.75, 0.50],
    'coef': btc_qr_coeffs[var_key_tsy] + nasdaq_qr_coeffs[var_key_tsy] +
    [ols_coeffs[var_key_tsy]],
    'se': btc_qr_ses[var_key_tsy] + nasdaq_qr_ses[var_key_tsy] + [ols_ses[var_key_tsy]]
}
create_coefficient_plot(data_dict=data_tsy_updated,
                        var_name=var_key_tsy,
                        y_axis_label_text="Coefficient Estimate",
                        plot_title="Impact of Lagged Treasury Spread (10Y-3M)",
                        plot_subtitle="Bitcoin QR vs. NASDAQ QR vs. Pooled OLS",
                        output_filename_base="py_plot_L_treasury_spread", # New filename base
                        figures_path=figures_path)

print(f"All Python plots generated and saved in '{figures_path}' directory.")
```

References

[1] Massimiliano Caporin, Eduardo Rossi, and Paolo Santucci de Magistris. "Volatility jumps and their economic determinants". In: *Journal of Financial Econometrics* 14.1 (2016), pp. 29–80. DOI: 10.1093/jjfinec/nbu028.