AMProject_Sorted

Laura Gullicksen, Erich Gozebina, Daria Palitzsch

23/05/2025

#TODO: Insert Captions for all Figures and Tables

1. Introduction

#TODO: write Intro

2. Data & Descriptive Analysis

Data Aggregation and Strategy Frequency

The raw dataset provides CHAINLINK price data at *hourly frequency.* While such high-frequency data offers more granular insights, we chose to **aggregate the data to daily frequency** for the following reasons:

1. **Alignment with Trading Strategy**: Our core trading strategy is based on a **7-day momentum signal**, which inherently reflects **weekly price trends**. Applying such a signal at an hourly resolution would not be consistent with the strategy's time horizon.
2. **Noise Reduction**: Hourly crypto data can be highly volatile and noisy. Aggregating to daily returns reduces **microstructure noise, short-term reversals**, and **Whale-driven price spikes**, improving the signal-to-noise ratio.
3. **Practical Execution Perspective**: A strategy that rebalances daily is **more realistic to implement**, considering gas fees, latencyn and operational constraints on decentralized exchanges or CEX APIs.
4. **Interpretability and Robustness**: Daily returns are more interpretable and robust across backtests. Most financial and technical indicators (e.g., RSI, MACD, SMA) are commonly applied on daily charts.

Our strategy issues long/short signals based on the past 7-day log return of CHAINLINK, i.e.,

$$\text{Momentum}_t^{(7)} = \log\left(\frac{P_t}{P_{t-7}}\right)$$

This naturally assumes daily data, as each observation reflects the cumulative return over the previous seven days.

In summary, aggregating to daily frequency is a theoretically and practically sound choice. It ensures consistency between our signal construction, model estimation, and backtesting logic.

To illustrate the effects of our aggregation decision and provide initial insights into the behavior of CHAINLINK, we present two visualizations below:

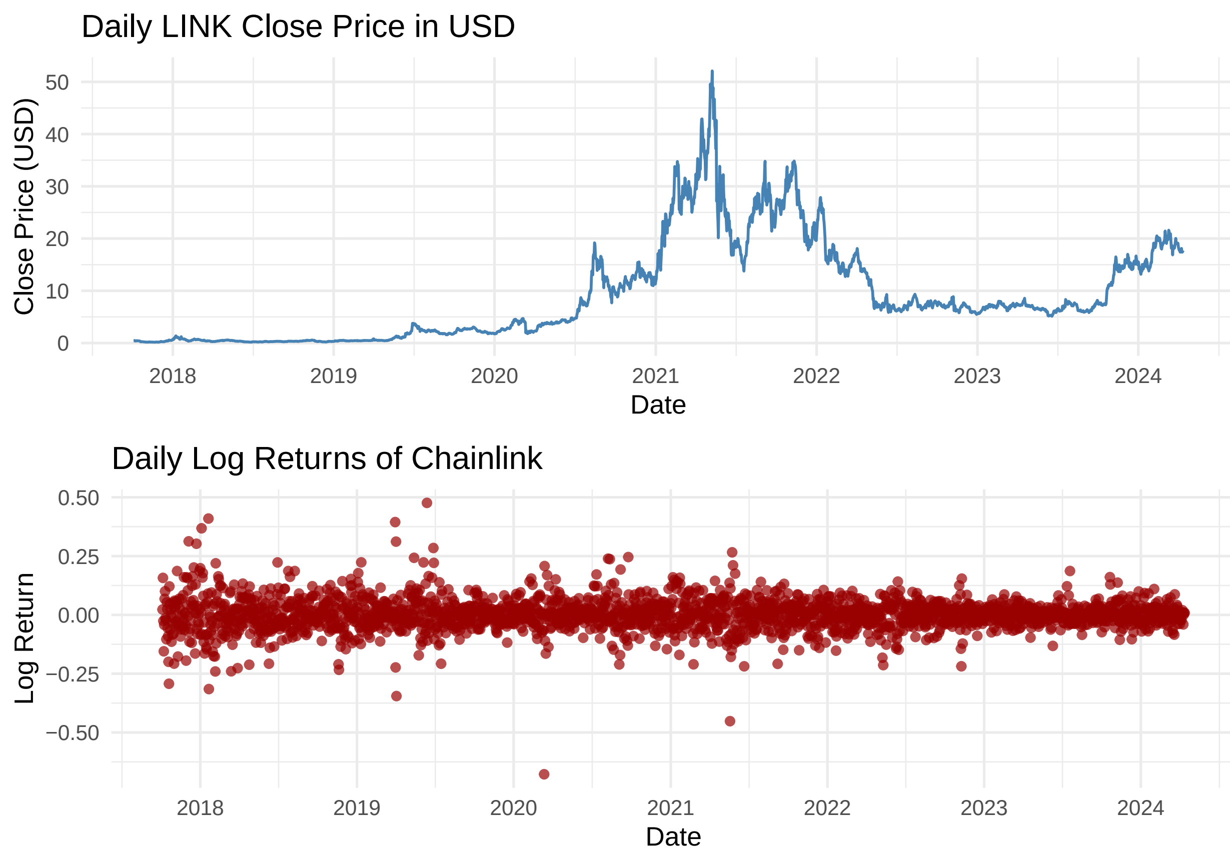

Figure 1: Close Price and Log Return of LINK

Figure 1 shows two plots:

- The **upper panel** displays the daily closing price of CHAINLINK in USD since 2018, highlighting the asset's strong bull run during the 2020–2021 crypto boom and its subsequent volatility.
- The **lower panel** shows the corresponding daily log returns, capturing the day-to-day fluctuations in price.

Having a first glance on the LINK's price data, we see that there is not much movement until 2020, followed by sharp increases by factor ten in the consecutive nine months. The price reaches its peak of over 50$/LINK in May 2021. In the second half of 2021 until April 2022 the data shows volatile behavior but with significant decrease on average. A trendless period of relatively low volatility, starting in May 2022 and ending in October 2023, shows prices between 5 and 10$/LINK. Finally, we observe another increase at the end of 2023 and beginning of 2024.

Moving away from non-stationary price data to stationary log returns, we observe a higher dispersion in the earlier years, suggesting a higher volatility then. Furthermore, most of points cluster around zero, indicating that there is no significant long term drift. Outliers, both positive and negative, imply extreme relative price movements, especially in the earlier period. Another important insight is the changing variance since the density of the points increases in the second half of the time window.

Why did we chose log returns over canonical (arithmetic) returns? Using log returns instead of canonical returns is a standard practice in financial econometrics and modeling.

$$\tilde{r}_t = \log(r_t + 1) = \log\left(\frac{p_t}{p_{t-1}}\right)$$

The underlying reason is the assumption that prices of an financial asset are log-normally distributed. This is reasonable since the log-normal distribution does not allow for negative values, which is also true for most asset prices (particularly for crypto currencies). Moreover, historical data provides evidence that the log-normal distribution gives a good fit for the prices of many financial assets. In reverse, since the logarithm function amplifies returns that are close to -1 more than positive returns, log-returns are distributed more symmetrically than canonical returns and indeed follow a normal distribution. Additionally, if returns are small, log returns approximate canonical returns very well. For x close to zero, it holds that

$$\log(x + 1) \approx x.$$

We can expect small returns since we shorten the considered time interval. Another important property is the additivity of log returns. It allows us to aggregate returns over multiple periods by summing up the pointwise log returns - a property that canonical returns miss. These properties make log returns more suitable for linear regression models, hypothesis testing, and machine learning regressors.

To better understand the characteristics of the Chainlink price and return series, we compute a set of descriptive statistics based on the daily close prices and the corresponding log returns. These statistics provide a first impression of the dataset's distribution, dispersion, and extreme values, and help assess whether further preprocessing or transformation steps are necessary before applying predictive models.

The summary statistics in Table 1 reveal that the mean daily log return of Chainlink is close to zero, while the standard deviation is relatively high, reflecting the well-known volatility of cryptocurrency markets. The minimum and maximum returns further highlight the presence of large price swings. The wide range between the minimum and maximum close prices illustrates the strong appreciation potential, but also the riskiness of the asset over the observation period.

Table 1: Summary Statistics for Chainlink Price and Returns

Statistic	Value
Number of Observations	2,383.0000
Mean Close Price	9.1484
Std. Dev. Close Price	9.5407
Minimum Close Price	0.1453
Maximum Close Price	52.1000
Mean Return	0.0016
Std. Dev. Return	0.0676
Minimum Return	-0.6776
Maximum Return	0.4762

To evaluate the temporal dependence structure of Chainlink's daily log returns, we plot the autocorrelation function (ACF). The ACF helps determine whether past returns exhibit statistically significant correlation with future returns — a key consideration when assessing the potential for return predictability.

#TODO: don't show 0 on x-axis

The autocorrelation function (ACF) of daily log returns in Figure 2 shows no statistically significant linear dependence at any lag, indicating that past returns do not linearly predict future returns. This finding supports the weak-form Efficient Market Hypothesis (EMH). However, it does not rule out the presence of exploitable patterns captured by non-linear or directional indicators. Therefore, we proceed with a momentum-based trading strategy, leveraging the sign of multi-day past returns to generate long or short signals.

3. Standard Model

We develop a basic model that will serve as a starting point for an extended model. Our first approach to predict future returns is a simple linear regression. Why linear regression? First, this technique is the underlying mechanism of many advanced models that are often generalizations of the linear case. Therefore, it is a good fit for a starting point. In general, linear regression aims to identify linear relationships between input data and the target dimension. In our case the input data is price data, trading volume, market capitalization, and every predictor that is derived from those - returns for instance. The target dimension that we are going to predict is the return of the next day. The simplicity of linear mappings make results easy to interpret, whereby the model still remains powerful since many observed relationships are indeed of linear nature. Moreover, linear regression indicates the strength of those linear ties what makes it a helpful tool for decision making.

7-Day Momentum Signal Strategy

To quantify short-term trends in Chainlink's price, we construct a **7-day momentum signal** defined as the logarithmic return over the past seven trading days:

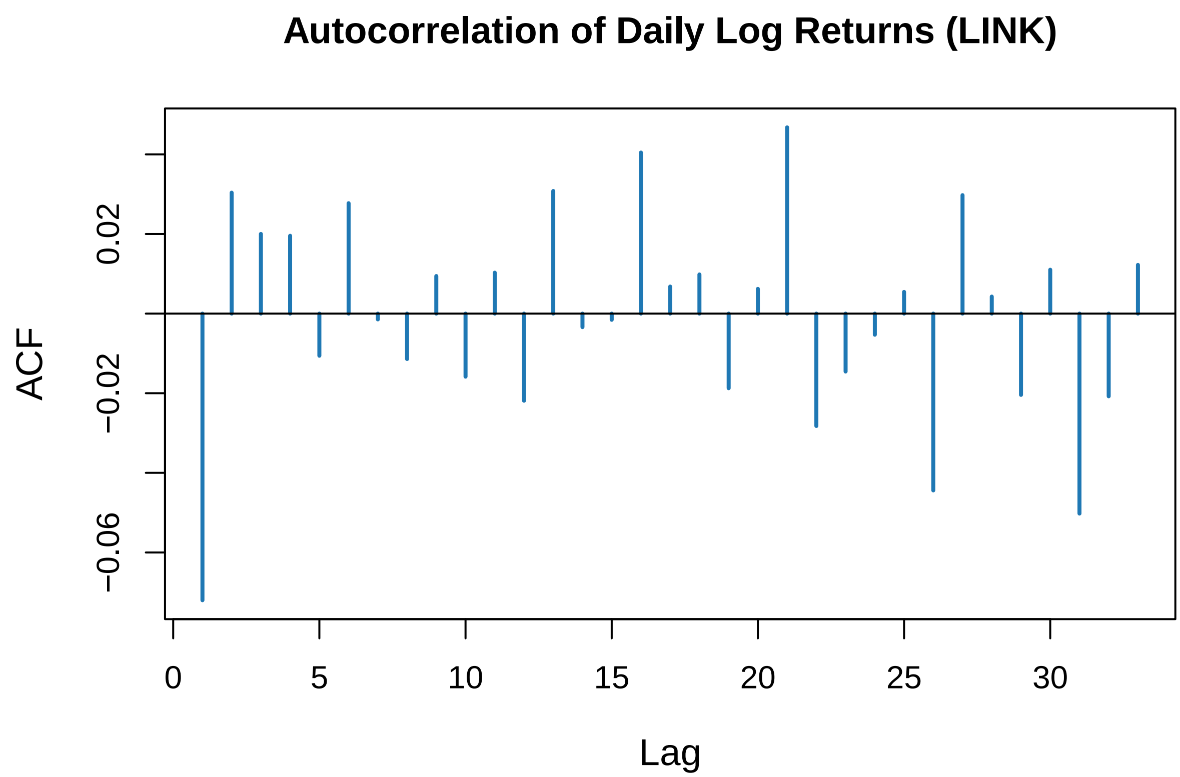

Figure 2: ACF of Daily Log Returns for Chainlink

$$\text{Momentum}_t^{(7)} = \log\left(\frac{P_t}{P_{t-7}}\right)$$

This momentum measure captures the cumulative price movement over one week and serves as a central predictive feature in our empirical analysis.

In parallel, we define the **target return** as the one-day-ahead log return:

$$r_{t+1} = \log\left(\frac{P_{t+1}}{P_t}\right)$$

This setup aligns with our objective of forecasting next-day returns based on recent trends and other features. While a simple rule-based strategy could be derived by taking positions based on the sign of the predicted return, our approach instead relies on data-driven prediction models such as linear regression and LASSO to generate trading signals and evaluate strategy performance.

Baseline Predictive Model: 7d-Momentum-Only

Table 2: Regression Results: 7-Day Momentum Strategy

	Dependent variable:
	Strategy Return
Intercept	0.0022
	(0.0088)
7-Day Momentum	0.0013
	(0.0017)
Observations	1,900
R^2	0.00003
Adjusted R^2	-0.0005
Residual Std. Error	0.0731 (df = 1898)
F Statistic	0.0605 (df = 1; 1898)
Note:	*p<0.1; **p<0.05; ***p<0.01

Draft from Erich:

The linear regression results in an estimation for the intercept of 0.001 having a p-value of 0.427 and a slope of 0.002 having a p-value of 0.806 Since the p-value for the slope clearly exceeds 0.05 (the threshold, which is commonly used for acceptance), we can not conclude a linear relation between the 7-day momentum and the future day return. Actually, this aligns with our finding in the ACF analysis, where no linear dependency between a lag of 7 days and the future day return was indicated. Thus, this result calls for an extended approach of future return prediction.

4. Extension

Extension of our OLS

To enhance the predictive power of the benchmark model, we extend it by incorporating a broader set of explanatory variables that capture not only short- and medium-term price dynamics, but also market sentiment, technical indicators, and inter-asset relationships. These include:

- Momentum indicators over 3, 7, and 14 days,
- Lagged daily returns (1-day and 2-day),
- A 7-day rolling volatility measure,
- Technical indicators such as the 14-day Relative Strength Index (RSI), MACD value and histogram, Simple Moving Average difference, and Average True Range (ATR),

- Day-of-week dummy variables to capture potential calendar effects,
- BTC-based predictors: daily BTC return, 7-day BTC momentum, and 7-day BTC volatility,
- ETH-based predictors: daily ETH return, 7-day ETH momentum, and 7-day ETH volatility,
- ETH trading volume: daily ETH volume return, 7-day ETH volume momentum, and 7-day ETH volume volatility,
- ETH market capitalization: daily ETH market capitalization return, 7-day ETH market capitalization momentum, and 7-day ETH market capitalization volatility,
- Ethereum gas fees: daily gas return, 7-day gas momentum, and 7-day gas volatility.

The extended predictive regression model is specified as:

$$r_{t+1} = \alpha + \sum_{h \in \{3,7,14\}} \beta_h \cdot \text{Momentum}_t^{(h)} + \gamma_1 \cdot r_t + \gamma_2 \cdot r_{t-1} + \delta \cdot \text{Volatility}_t^{(7)} + \sum_j \theta_j \cdot X_t^{(j)} + \varepsilon_{t+1}$$

where $X_t^{(j)}$ represents the set of technical indicators (RSI, MACD, ATR, SMA), weekday dummies, and BTC-based predictors.

$$\begin{aligned}
r_{t+1} &:= \log\left(\frac{P_{t+1}}{P_t}\right) \quad \text{(one-day-ahead LINK return)} \\
\text{Momentum}_t^{(h)} &:= \log\left(\frac{P_t}{P_{t-h}}\right) \quad \text{for } h \in \{3, 7, 14\} \\
\text{Volatility}_t^{(7)} &:= \text{std}\left(r_{t-6}, \ldots, r_t\right) \\
\text{BTC return}_t &:= \log\left(\frac{P_t^{\text{BTC}}}{P_{t-1}^{\text{BTC}}}\right) \\
\text{BTC Momentum}_t^{(7)} &:= \log\left(\frac{P_t^{\text{BTC}}}{P_{t-7}^{\text{BTC}}}\right) \\
\text{BTC Volatility}_t^{(7)} &:= \text{std}\left(r_{t-6}^{\text{BTC}}, \ldots, r_t^{\text{BTC}}\right)
\end{aligned}$$

The ETH-based predictors are constructed analogously to the BTC-based predictors. The parametrization of this model is estimated via Ordinary Least Squares (OLS) on the in-sample period. By incorporating this rich feature set, we aim to capture a range of return drivers including price trends, market overreaction, volatility clustering, inter-market dependencies, and behavioral biases tied to trading weekdays.

#TODO: Description and interpretation of output -> Laura

Lasso Model

Table 3: Extended Regression Model: Predicting LINK Returns with Crypto Features

	Dependent variable:
	Target Return
Intercept	-0.0816^{**}
	(0.0339)
Momentum (3d)	-0.0156
	(0.0194)
Momentum (7d)	-0.0192
	(0.0175)
Momentum (14d)	0.0796^{**}
	(0.0331)
Lagged Return (1d)	0.0920^{**}
	(0.0360)
Lagged Return (2d)	-0.0710
	(0.0548)
Volatility (7d)	0.0004
	(0.0003)
RSI (14)	-0.0010
	(0.0018)
SMA Diff	0.0004
	(0.0005)
MACD Value	0.0011
	(0.0017)
MACD Histogram	-0.0020
	(0.0014)
ATR (14)	0.0063
	(0.0048)
Monday	-0.0010
	(0.0045)
Tuesday	0.0014
	(0.0045)
Wednesday	0.0041
	(0.0045)

To prevent overfitting and perform automatic variable selection, we extend our linear modeling approach using the Lasso (Least Absolute Shrinkage and Selection Operator). The Lasso adds a penalty term to the standard OLS loss function, shrinking some coefficient estimates toward zero. This results in a sparse model that may improve predictive performance, particularly when dealing with multiple correlated predictors. Furthermore, reducing the number of relevant features allows for a better interpretation of simulation results.

The Lasso estimator is defined as the solution to the following optimization problem:

$$\hat{\beta}^{\text{lasso}} = \arg\min_{\beta_0,\beta} \left\{ \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right\}$$

where:

- y_i is the target variable (e.g., one-day-ahead return),
- x_{ij} are the predictor variables,
- β_j are the coefficients,
- $\lambda \geq 0$ is the tuning parameter controlling the strength of the penalty.

As λ increases, more coefficients are shrunk toward zero. For $\lambda = 0$, the solution coincides with OLS.

We use 10-fold cross-validation to select the optimal λ that minimizes the mean squared prediction error on held-out data.

Optimal Lambda from Cross-Validation: $\lambda^* = 0.002780$

Given this optimal lambda, we now run the LASSO regression, including all variables from the previous OLS regression:

Table 4: Non-Zero Coefficients from LASSO Regression

Predictor	Coefficient
Intercept	0.001414
Bitcoin Daily Return	-0.955208
Ethereum 7-Day Volume Momentum	0.001887

The LASSO regression identified two non-zero predictors for explaining Chainlink (LINK) returns:

1. Bitcoin Daily Return (Coefficient: -0.9552): This variable has a large and negative coefficient, indicating that when Bitcoin's daily return increases by 1 unit (in our scaled units), the predicted return of our LINK-based trading strategy decreases by approximately 0.9552 units, all else equal. This suggests a strong inverse relationship

between BTC movements and our strategy, potentially due to hedging behavior or negative spillovers.

2. Ethereum 7-Day Volume Momentum (Coefficient: 0.0019): This predictor captures short-term trends in Ethereum's trading volume. The positive but small coefficient implies that higher recent momentum in ETH trading volume is weakly associated with increased LINK returns, possibly due to spillover effects from rising market activity in related tokens.

3. Intercept (Coefficient: 0.0014): The intercept represents the model's baseline prediction when all predictors are zero. Here, it suggests a small positive base return, though in practice this often has less interpretive value than the covariates.

5. Forecasting & Backtesting

In-Sample testing

To evaluate the performance of our predictive models, we begin by conducting in-sample (IS) testing. This involves fitting each model on a fixed training sample and evaluating how well the model explains historical variation in the data.

We assess in-sample performance using the following criteria:

- **Mean Squared Error (MSE)**: Measures the average squared difference between predicted and actual returns.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2$$

- **Adjusted R^2**: Indicates the proportion of variance explained by the model, adjusted for the number of predictors.

$$R^2_{\text{adj}} = 1 - \frac{\text{RSS}/(n-p-1)}{\text{TSS}/(n-1)}$$

- **Directional Accuracy**: The fraction of times the predicted direction matches the actual direction of returns.

$$\text{Accuracy} = \frac{1}{n}\sum_{i=1}^{n}\mathbb{1}\left(\text{sign}(\hat{y}_i) = \text{sign}(y_i)\right)$$

These metrics are computed for all three models:

1. Benchmark (7-day momentum only),
2. Extended linear model with multiple features,
3. Lasso-regularized regression with automatic feature selection.

Table 5: In-Sample Performance of Benchmark, Extended, and Lasso Models

Model	MSE	Directional_Accuracy	Adj_R2
Benchmark	0.0053	0.5051	-0.0005
Extended	0.0035	0.7110	0.3310
Lasso	0.0036	0.7142	0.3229

#TODO: interpret the In-Sample Results

Out-of-sample testing:

After performing the in-sample analysis, we now test our models Out-Of-Sample on the test data set:

Table 6: Out-of-Sample Model Evaluation

Model	MSE	Directional_Accuracy	R2_OS
Benchmark	0.0014	0.5191	0.0000
Extended	0.0012	0.7277	0.1881
Lasso	0.0010	0.7362	0.3363

#TODO Interpret OOS Results

The benchmark model fails to generalize out-of-sample (R^2_OS = 0), confirming the weak predictive power of raw 7-day momentum. In contrast, the LASSO model improves out-of-sample accuracy substantially (R^2_OS = 0.34), likely due to its ability to regularize noise and select relevant predictors.

To evaluate the trading performance of our predictive models beyond return forecasting, we implement a simple In-and-Out (I/O) strategy. This rule-based strategy enters the market ("IN") only when the model predicts strong positive returns—i.e., when the predicted value exceeds the 75th percentile of the respective model's prediction distribution. Otherwise, the model stays out of the market ("OUT").

Signal Construction:

We construct separate trading signals for:

- the Benchmark model (7-day momentum),
- the Extended Linear Regression model,
- the LASSO model, and
- the actual target returns (for reference).

The procedure:

- 1 (enter position) if the predicted return exceeds the threshold

- 0 (stay out) otherwise

Compute daily strategy returns by multiplying the lagged signal with the actual log return. Apply a fixed fee per position change to simulate realistic trading costs.

This approach emphasizes selective participation in the market based on the model's confidence and allows us to assess whether models can not only predict returns but also generate economically meaningful signals.

#TODO: check the code

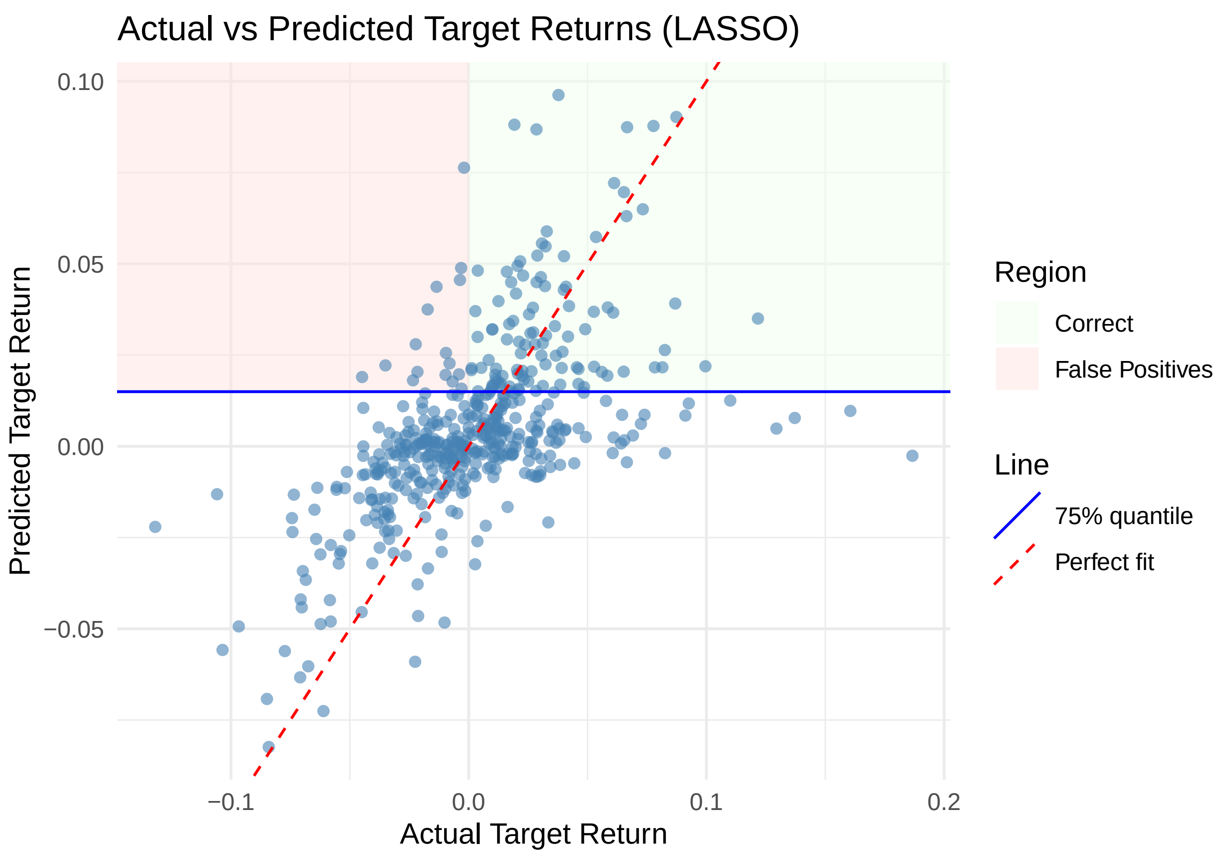

Figure 3: Actual vs Predicted Target Returns (LASSO)

Figure 3 visualizes the relationship between actual and predicted one-day-ahead returns from the LASSO regression model. Each point represents an observation in the test set, with the x-axis showing the realized return and the y-axis the model's predicted return. The dashed red line indicates the 45° line of perfect predictions, while the blue horizontal line marks the 75th percentile of predicted returns.

We observe a moderate clustering of points around the diagonal, suggesting that the LASSO model captures some predictive structure in the return dynamics, particularly for returns near zero. However, deviations from the red line highlight prediction errors, especially in the tails.

The green region denotes instances where the predicted and actual returns have the same sign—i.e., the model correctly forecasts the return direction. These cases reflect successful directional predictions, which are critical for long/short trading strategies. In contrast, the red-shaded area highlights false positives, where the model incorrectly predicts a positive return, but the actual return is negative.

While the spread around the diagonal indicates that exact return magnitudes are not always accurately predicted, the high density in the green region supports the model's strong directional accuracy (see Table 6). This suggests that the LASSO model is well-suited for sign-based trading rules, even if its point predictions are noisy.

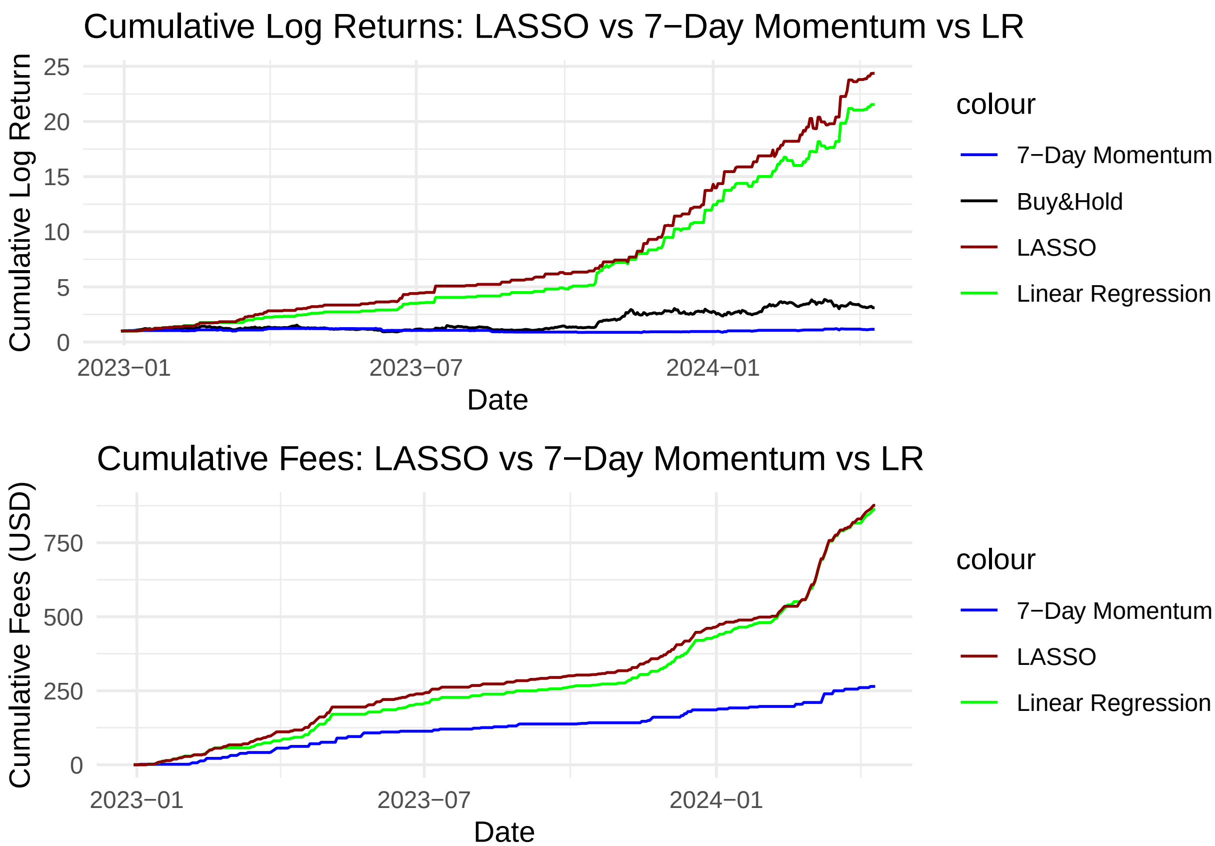

Figure 4: Cumulative Performance of Trading Strategies (Log Returns and Fees)

#TODO describe the plots

The **top panel** of Figure 4 compares the cumulative log returns of four strategies:

- LASSO model (dark red) achieves the highest return, significantly outperforming all others.
- Linear Regression (green) performs closely behind LASSO.
- The 7-Day Momentum (blue) lags far behind, reflecting its weak signal quality.
- Buy-and-Hold (black) delivers minimal returns with no active decision-making.

The **bottom panel** shows cumulative trading fees incurred due to signal changes:

- Both LASSO and Linear Regression generate more trades and hence higher fees.
- Despite these higher transaction costs, their net performance (gross return - fees) is still superior.
- The 7-Day Momentum strategy results in fewer position switches and lower fees but offers little economic value due to poor returns.

Overall, the LASSO model demonstrates superior performance, not only in terms of return predictability but also as a trading signal generator, even after accounting for transaction costs.

#TODO: Interpret plot results

Optional: Rolling Lasso Approach

6. Conclusion

#TODO: write conclusion