Norges Handelshøyskole

FIN11 Trading and Market Microstructure

FIN11 Term paper

Predefined Information

Start date:	04-11-2025 09:00 CET
End date:	08-12-2025 12:15 CET
Examination type:	Term paper (70%)
Term:	2025 12
Grading scale:	Norwegian 6-point scale (A-F)
Flow code:	191 FIN11 1 O3 2025 12
External assessor:	External assessor 1
Internal assessor:	Internal assessor 1

Participant

Candidate No.:	1100

Information from participant

Can the submission be used anonymously for educational purposes?:
Yes

I am aware of what constitutes cheating and the consequences of cheating on an exam. I hereby declare that I will NOT cheat on this exam. *:
Yes

Group

Group name:	(Anonymised)
Group number:	7
Other members:	1040, 1003, 1046

In which language did you write your answer paper? *

English



When Insiders Buy… Should We? - Evidence From Sweden



Total Words: 6,597
(excluding abstract, tables, references and appendix)

Contents

ABSTRACT **2**

DEFINITIONS **3**

1. INTRODUCTION **4**

2. THEORY **5**

2.1 Classification of Insiders 5
2.2 Regulations on Insider Trading 6
2.3 Informed Traders and Asymmetric Information 6
2.4 Efficient Market Hypothesis (EMH) 7

3. DATA AND METHODOLOGY **8**

3.1 Data Sources 8
3.2 Return Calculation 9
3.3 Data Constraints and Practical Considerations 10
3.4 Data Filtering and Sample Selection 11
3.5 Return Measurement 12

4. SUB-QUESTIONS AND HYPOTHESES **13**

5. RESULTS **15**

5.1 General Results 15
5.1.1 Following all Insider Trades *15*
5.1.2 Average Return per Year *15*
5.1.3 Return Distributions. *16*
5.2 Differentiation by Market Capitalization 17
5.3 Differentiation by Transaction Value 19
5.4 Buying the Dip? 22
5.5 Insider Position 24

6. STATISTICAL ANALYSIS **25**

6.1 One sample T-test of Mean Excess return > 0 26
6.2 Mann-Whitney U-Tests: Market Cap Segment Analysis 27
6.3 Mann-Whitney U-Tests: Transaction Value 29
6.4 Mann-Whitney U-Tests: Insider Position 30
6.5 Mann-Whitney U-Tests: Buying the Dip 31

7. DISCUSSION **32**

7.1 Interpretation of Findings 32
7.2 Limitations 33
7.2.1 Data limitations *33*
7.2.2 Analytical limitations *33*
7.3 Practical Implications 34

8. CONCLUSION **35**

REFERENCES **37**

APPENDIX **39**

Interview with Sigurd Sveen, translated from Norwegian to English 39
Companies included in Analysis 41
Python Code 42

Abstract

This study investigates whether primary insider purchases in the Swedish stock market generate excess returns and whether specific trade or firm characteristics strengthen this signal. Using approximately 5,000 insider transactions from 400 companies (2016–2025), we simulate investor returns by replicating trades on the first trading day following public disclosures. Excess returns are measured relative to the OMX Stockholm 30 Index across 90-, 180-, and 365-day horizons.

The results show statistically significant but modest positive excess returns overall, indicating that insider purchases contain informational value. Performance is strongly concentrated in mid-cap firms (1–5B SEK), which outperform both small- and large-cap firms. In contrast, transaction size provides no reliable incremental signal, and purchases made by CEOs/CFOs underperform those by other insiders. Trades executed after major price declines (≥20% in 30 days) systematically underperform normal insider purchases. Overall, insider buying contains value-relevant information not fully incorporated into market prices at disclosure.

Definitions

MAR: Market Abuse Regulation*, the EU regulatory framework on market abuse (Regulation (EU) No. 596/2014), covering insider dealing, unlawful disclosure of inside information, market manipulation, and related disclosure and reporting obligations.*

PDMR: Person *Discharging Managerial Responsibilities. Individuals in senior managerial positions who have regular access to inside information and the authority to make decisions affecting the company's future, such as CEOs, CFOs and board members (as defined under MAR Article 3(1)(25)).*

ER: *The excess return(s) is defined as the stock's return minus the return of our benchmark, the OMX Stockholm 30 Index, over the same horizon.*

MER: *The mean excess return across all trades for a given holding period, calculated as the mean of the individual excess returns* $ER^{(i)}$.

OMXS30: *OMX Stockholm 30 Index which is a market-capitalization-weighted index comprising the 30 most actively traded stocks on Nasdaq Stockholm. Widely used as a benchmark for the Swedish equity market.*

1. Introduction

Primary insiders, such as executives, board members and major shareholders, possess superior knowledge about their firm's operations. This includes detailed insight into financial condition, operational performance and future prospects. Their personal trading may therefore contain information not yet reflected in stock prices (Seyhun, 1986). Academic research on primary insider trading dates back to the 1960s, establishing a foundation for understanding these information asymmetries. Investors seek to translate this insight into returns by interpreting insider transactions as potential buy signals.

In Norway, this topic has gained considerable public attention through *Innsideporteføljen*, a portfolio constructed and managed by Sigurd Sveen in the financial newspaper *Finansavisen*. The portfolio holds five stocks which typically rotates every 1–3 months. Stock selection is based on an evaluation of recent insider purchases on the Oslo Stock Exchange. To be included, a trade must exceed NOK 1 million and represent a substantial increase in the insider's holdings. To complement the academic perspective with practical insight, we conducted an interview with Sveen about the construction and performance of the portfolio. In this paper, however, we use *Innsideporteføljen* primarily as a motivating example rather than as an object of formal analysis. The interview in its entirety can be found in the appendix.

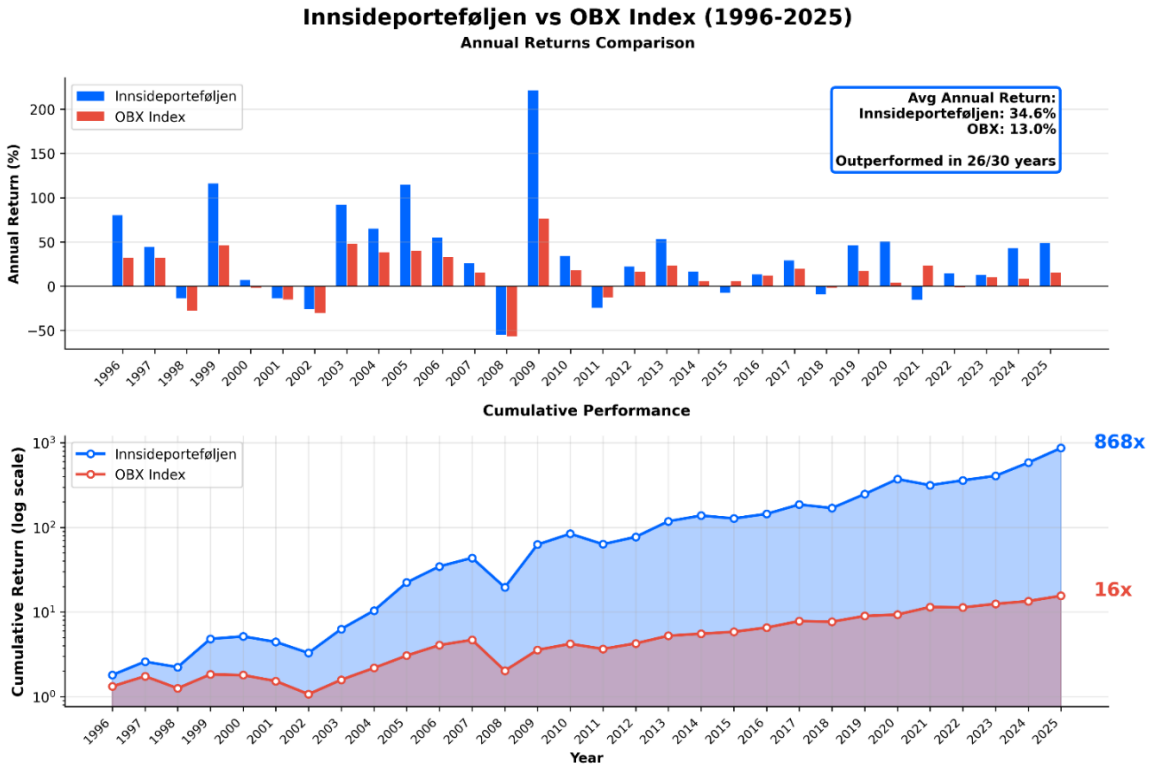


Figure 1: Innsideporteføljen vs OBX annual and cumulative return based on data from Finansavisen.

Since its inception in 1996, the portfolio has outperformed the index 26 out of 30 years. The portfolio's exceptional results have sparked a widespread debate regarding the predictive value of insider trades in Norway.

Motivated by the strong performance and rising public interest in primary insider strategies, we investigate whether similar dynamics appear in a comparable market. To explore this, we will analyze all primary insider transactions across 400 companies listed on Nasdaq Stockholm since 2016. Further, we examine which types of trades provide the strongest signals, across transaction size, insider categories and firm characteristics.

Based on these considerations we formulate the following research question:

Can following primary insider purchases in the Swedish stock market generate excess returns, and do any insider categories or trade characteristics elevate this return?

2. Theory

This chapter reviews the literature on legal insider trading to provide a theoretical basis for assessing both its profitability and its potential to generate excess returns.

2.1 Classification of Insiders

To contextualize our analysis, it is necessary to define what constitutes an insider under the regulatory framework. This paper focuses exclusively on corporate primary insiders, meaning individuals within the organization who hold positions that grant them regular access to value-relevant internal information.

According to Article 3(1)(25) of the EU Market Abuse Regulation (MAR, EU No. 596/2014), a "person discharging managerial responsibilities" (PDMR) is defined as: (a) a member of the administrative, management or supervisory body of the issuer (e.g., CEOs, CFOs and board members), or (b) a senior executive who, although not part of bodies referred to in (a), has regular access to inside information.

In this study, we use reported transactions from Finansinspektionen's official MAR reporting system in Sweden (Finansinspektionen, n.d.). These disclosures include trades from PDMRs (as defined under Article 3(1)(25)).

2.2 Regulations on Insider Trading

Understanding the regulatory framework is important for assessing whether outside investors can replicate disclosed insider purchases to achieve excess returns. Sweden is subject to the EU MAR which regulates the use and disclosure of insider information. Article 7 of MAR defines inside information as precise, non-public, and price-sensitive information. Article 17 of MAR requires issuers to disclose such information to the public as soon as possible.

In Sweden, these disclosures are handled through Finansinspektionen's official reporting system. According to Article 19, PDMRs have to report their transactions within three business days. These filings are made publicly available, creating a transparent and accessible dataset for outside investors. Article 19(11) of MAR additionally imposes a 30-day closed period before financial reports, during which PDMRs are prohibited from trading. This regulatory structure ensures timely and standardized disclosure of insider behavior, which makes it accessible for the broader market.

2.3 Informed Traders and Asymmetric Information

"Markets operate under asymmetric information: Insiders know more than everyone else, and they legally exploit that edge." – Sigurd Sveen (interview)

This observation, made by Sveen, captures the core idea of informed trading under asymmetric information.

Asymmetric information arises when some investors possess superior information about a firm's value than others. In equity markets, such information gaps can cause prices to deviate from fundamental value, because not all relevant information is fully incorporated despite public disclosures and analyst coverage.

In this setting, informed traders are those who trade strategically on their informational advantage. Harris (2003, p. 10-1) defines informed traders as: "speculators who acquire and act on information about fundamental values". Within this framework, value traders are particularly relevant. Value traders form their own assessment of fundamental value and are willing to hold positions until market prices converge toward that valuation. These valuations can be based on both private and public information.

Primary insiders, especially CEOs and CFOs, are naturally viewed as informed traders. As part of the organization, they have detailed knowledge of firm fundamentals, operational developments and risk exposures. Although regulations prohibit trading on material non-public information, several studies find that legal insider purchases still predict positive excess returns (e.g. Lakonishok & Lee, 2001; Seyhun, 1986). A plausible explanation is that insiders act not only on specific news events, but on a broader understanding of their firm's long-term prospects and performance. Such knowledge may not meet the legal definition of inside information under MAR, yet it still provides insiders with a persistent information advantage over outside investors.

This perspective motivates our examination of insider transactions as informed trading signals and underpins our hypothesis that insider purchases should, on average, be followed by positive excess returns.

Spence's (1973) signaling theory framework was originally developed in labor economics. It has since been generalized to information-asymmetric markets, including finance, where costly actions can function as credible signals. In this setting, insider purchases require insiders to commit personal capital and accept regulatory oversight, which makes such trades a costly action. According to signaling theory, more costly actions convey stronger and more credible signals. Applied to insider trading, markets may therefore interpret larger personal purchases as stronger indications of confidence in the firm's valuation, even if insiders do not intend to signal.

2.4 Efficient Market Hypothesis (EMH)

For primary-insider trades to generate excess returns, markets cannot be fully efficient.

Market efficiency refers to the extent to which asset prices reflect available information. Fama's Efficient Market Hypothesis (EMH) distinguishes between weak, semi-strong and strong forms of efficiency (Fama, 1970). In the weak form, prices incorporate all historical price information. In the semi-strong form, prices fully reflect all publicly available information and adjust immediately to new announcements. In the strong form, prices also reflect non-public information, implying that no group of investors, including insiders, can systematically earn excess returns.

However, the EMH has been challenged both empirically and theoretically. Behavioral finance has documented several "anomalies" that appear inconsistent with market efficiency. These include the momentum effect (Jegadeesh & Titman, 1993) and post-earnings announcement drift (Bernard & Thomas, 1989). These anomalies suggest that markets do not instantly incorporate all available information into prices. In response, Fama (1991) refined his view of efficiency by explicitly incorporating trading and information-acquisition costs. Prices are efficient as long as no investor can earn abnormal profits after these costs, and new information is reflected in prices only when its value exceeds the cost of obtaining it.

In the context of insider trading, the semi-strong form is the relevant benchmark. If markets were perfectly semi-strong-form, insider purchases would not predict future excess returns. This because all public value-relevant information would already be incorporated into prices. However, this assumes that insiders trade in line with MAR. Any systematic return pattern following insider transactions therefore suggests a deviation from strict semi-strong efficiency. In Fama's (1991) cost-based view, however, such patterns can still be compatible with efficiency if insiders face lower information-acquisition costs than other investors.

3. Data and Methodology

3.1 Data Sources

Our analysis combines two primary data components; insider transaction disclosures reported to *Finansinspektionen* in Sweden and historical market price data obtained from Yahoo Finance.

The *Finansinspektionen* database compiles all trades reported under article 19 of MAR. Our dataset covers all insider trades from 2016 to 2025 and contains the transaction date, publication date, insider role, instrument type, volume, price, and ISIN. The MAR framework requires PDMRs to report trades within three business days and ensures standardized disclosures.

Historical stock prices and index values were retrieved with Python programming using the Yahoo Finance library. This source provides adjusted prices that account for stock splits, dividends, and other corporate actions, ensuring consistency across the sample period.

3.2 Return Calculation

Each insider transaction is treated as an independent event, producing a single observation of stock performance over a fixed horizon. We computed the return an external investor would have achieved by replicating each trade individually.

We evaluate performance over three holding periods: 90, 180 and 365 days. The 90-day window captures the short-term information incorporation, while the 180- and 365-day periods examine the predictive effect over a longer period. Using multiple timeframes ensures the simulated return are not dependent on a single horizon, creating an opportunity for a broader assessment.

Entry Date and Price

To implement this procedure, the entry date of the trade was identified as the first trading day following the public disclosure (Publication + 1). This ensures that the simulated return reflects a realistic scenario for an outside investor. The entry price is defined as the midpoint between the intraday high-low prices on entry day:

$$P_{\text{entry}} = \frac{P_{\text{high}} + P_{\text{low}}}{2}$$

Due to stock splits and other corporate actions, the insider's manually reported stock price may not align with Yahoo Finance's adjusted price data for the transaction date. To account for this when calculating returns, the Python code validated each reported price against the trading range for that day. If the reported transaction price fell within the intraday range, it was used directly. Otherwise, the midpoint of the trading range that day was applied.

Exit Price

The exit price is the adjusted closing price after the specified holding period of 90, 180 or 365 days. The stock return for trade ***i*** is:

$$R_{\text{stock}}^{(i)} = \frac{P_{\text{exit}}^{(i)} - P_{\text{entry}}^{(i)}}{P_{\text{entry}}^{(i)}}.$$

Excess Return

To measure outperformance against the Swedish stock market, we calculate an equivalent return on the OMXS30 Index using the same strategy as for the insider trades. The benchmark return is:

$$R_{\text{OMX}}^{(i)} = \frac{I_{\text{exit}}^{(i)} - I_{\text{entry}}^{(i)}}{I_{\text{entry}}^{(i)}}.$$

The excess return (ER) for each trade ***i*** is defined as the difference between the stock return and the index return:

$$\text{ER}^{(\text{i})} = \text{R}_{\text{stock}}^{(\text{i})} - \text{R}_{\text{OMX}}^{(\text{i})}$$

3.3 Data Constraints and Practical Considerations

Our analysis relies exclusively on historical price data from Yahoo Finance. Prices are automatically adjusted for stock splits, dividends and other corporate actions. Our return calculations use the adjusted prices for both entry and exit, therefore dividends are not reflected in our strategy returns. However, the OMXS30 Index includes dividends, which creates an asymmetry in our comparison that should be considered when interpreting results.

Transaction costs are excluded in the return calculations. The average transaction cost per trade in Sweden typically ranges from 0.069 – 0.25% per trade, which would reduce net returns in a practical implementation (Avanza, 2025). The annual index management fee is excluded from the analysis, amounting to approximately 0.25% per year.

Our analysis does not adjust for systematic risk (beta) as Yahoo Finance does not provide consistent historical estimates across our sample. Observed excess returns may therefore partly reflect compensation for higher risk.

3.4 Data Filtering and Sample Selection

Insider disclosures were collected from Finansinspektionen on a company-by-company basis and consolidated into a comprehensive dataset. The raw data contained approximately 60,000 mandatory insider disclosures covering the period from 2016 to 2025. To isolate transactions suitable for return analysis we systematically filtered the data and applied the following criteria to the dataset:

Security Characteristics

We restricted the sample to acquisitions of ordinary shares, excluding sales. The analysis was limited to trades denominated in Swedish Krona (SEK) executed on exchange. Transactions classified as “outside a trading venue” or involving derivatives (options, warrants, RSUs) were excluded to ensure homogeneity.

Transaction Size

We imposed a minimum transaction value of 500,000 SEK to focus on trades with a substantial size. Our threshold was inspired by Sigurd Sveen’s *Innsideporteføljen*, which operates with a 1,000,000 NOK minimum. The limit excludes minor purchases that may reflect other motives than informed conviction, while retaining sufficient observations for the analysis.

Data Quality and Verification

Several data quality issues required manual verification and systematic exclusion. Due to corporate actions, such as stock splits, mergers or recapitalizations, ISIN numbers occasionally change. To ensure accurate price tracking, the ISIN consistency for affected securities was verified. Observations with missing or invalid dates were excluded, as exact timing is important for calculating the returns. We imposed a temporal constraint based on the holding period for each study, making sure only trades with complete forward-looking price data were retained.

A structural limitation in the Finansinspektionen database is that disclosures reported in 2016 and parts of 2017 did not include instrument type. Since these filings may include options, warrants and RSUs, they cannot be reliably classified as purchases of ordinary shares. While this restriction reduces the number of observations, it ensures a homogeneous sample for the analyzed period.

Given the size of the dataset, occasional data errors or inconsistencies may remain despite our verification efforts. Such errors are most likely to affect the segmented analysis with multiple filters, rather than the general analysis.

Final Sample Composition

After applying all filters and verification of the data, our final sample composition consists of approximately 5,000 valid transactions across 254 unique companies. The sample sizes vary slightly by the different holding periods, because of the complete forward-looking price requirement.

3.5 Return Measurement

For each insider purchase, we calculate excess return across different holding periods. Performance will be evaluated using three primary metrics: Mean excess return, median excess return and win rate. Each insider trade contributes to one observation per horizon, reflecting how much each trade out- or underperformed the OMXS30.

(1) Mean Excess Return (MER)

For each horizon $T \in \{90,180,365\}$, we calculate:

$$\bar{ER}(T) = \frac{1}{N}\sum_{i=1}^{N} E\,R^{(i)}(T)$$

This represents the average performance relative to the index and serves as our primary indicator of whether insider purchases contain systematically predictive information.

(2) Median Excess Return

To provide a more robust measure of central tendency, we calculate the median excess return:

$$\text{Median}(ER(T)) = \text{med}\,\{\,ER^{(1)}(T), \ldots, ER^{(N)}(T)\,\}.$$

A positive median indicates that outperformance is not solely driven by a few extreme winners but is widespread across the sample.

(3) Win Rate

To assess consistency, we calculate the proportion of trades generating positive ER:

$$\text{Win Rate}(T) = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}(E\,R^{(i)}(T) > 0).$$

A win rate above 50% indicates that trades outperform the market more often than not, supporting the presence of a systematic informational advantage.

4. Sub-questions and Hypotheses

Building on prior research and theoretical framework, we examine the following sub-questions:

1) Does replicating insider purchases as a whole generate excess returns?

This question forms the foundation of our analysis. Before examining specific trade and firm characteristics, we first assess whether insider purchases, in aggregate, generate ER relative to a passive investment in the OMXS30. Prior research generally finds that the ER from following insider purchases in the Swedish market are modest, but positive (Sundling & Verschuur, 2014).

Hypothesis 1:
Insider purchases generate positive ER relative to the OMXS30 Index.

2) Does market capitalization affect insider trading returns?

Information asymmetry varies systematically across firms. Smaller firms exhibit lower analyst coverage, thinner trading, and slower price incorporation (Bhushan, 1989; Brennan & Subrahmanyam, 1995). Because insiders in smaller firms likely faces less efficient pricing, their information should translate more directly into ER.

Hypothesis 2:
ER are stronger in companies with a lower market capitalization.

3) Does transaction size influence predictive power?

Signal theory suggests that costly actions convey stronger information (Spence, 1973). For insiders, a large personal purchase represents a credible signal of private information or strong belief about future performance. Seyhun (1986) finds that ER increase with transaction size, suggesting that insiders trade larger volumes when they possess more valuable information.

Hypothesis 3:
Larger transactions generate higher ER.

4) Do insider purchases reported after sudden price declines outperform other insider trades?

This analysis is inspired by the well-documented short-term reversal effect, which suggests that investors tend to overreact to negative information in the short run (De Bondt & Thaler, 1985). If such overreactions occur, insiders with superior information might exploit these mispricing by purchasing shares.

Hypothesis 4:
Insider purchases made after substantial short-term price declines generate higher ER than other insider purchases.

5) Do certain insider positions provide stronger signals? (CEO, CFO, board members)

According to Harris (2003), traders vary in the quality and relevance of information they possess. This implies that insiders also differ in informational advantage depending on their role and involvement in the firm. Previous research on differing returns from inside trading for CEOs and CFOs (Wang, Shin & Francis, 2012; Ozkan & Trzeciakiewicz, 2014) raised the question of signal significance depending on the insider's position.

Hypothesis 5:
Purchases by CEOs and CFOs generate higher ER than purchases by other insiders.

5. Results

5.1 General Results

5.1.1 Following all Insider Trades

Our analysis of the valid transactions, covering 4,037-4,813 trades, produced the following results:

Holding Period	Number of Transactions	Strategy Return	Median Return	Avg. Index Return	MER	Win Rate (Positive)	Win Rate (vs Index)	Std. Dev. (Strategy Return)
90 days	4,813	3.26%	2.68%	2.48%	0.78%	56.5%	48.5%	22.29%
180 days	4,574	6.23%	5.37%	4.23%	2.00%	58.5%	52.1%	32.82%
365 days	4,037	10.43%	10.45%	9.20%	1.22%	59.1%	50.8%	47.29%

Figure 2: Overview of results for the strategy and the benchmark index.

Our results reveal consistent patterns in insider trading performance. We see that the inside strategy outperforms the index across all holding periods. Additionally, 2/3 periods achieved a win rate above 50% versus index, indicating that insider purchases contain meaningful information about future price movements. The high standard deviation is expected, as the analysis is conducted at the transactions level, reflecting the volatility of a single stock position.

An **important** methodological consideration arises when comparing the returns across different holding periods. While the 90- and 180-day strategies show lower absolute returns per trade, one might assume that annualizing these returns would result in a superior compound annual performance. However, this assumption requires that trading signals are evenly distributed throughout the year, to allow continuous reinvestment. Given the limited number of qualifying signals in our dataset, this condition is unlikely to hold in practice. The observed returns for 90- and 180-day strategies remain valid in their own timeframe but annualizing them would overstate realistically achievable performance given the signals.

5.1.2 Average Return per Year

Figure 3 illustrates the annual average return of each insider trade compared to OMXS30 across the three holding periods. The insider strategy outperforms the benchmark in the majority of years, beating it 7/9 years for the 90- and 180-day horizon, and 6/8 for the 365-day horizon. Moreover, we see substantial variation in performance over the years. The exceptional performance during 2020 stands out in all holding periods, suggesting insiders take advantage of periods with uncertainty and high volatility.

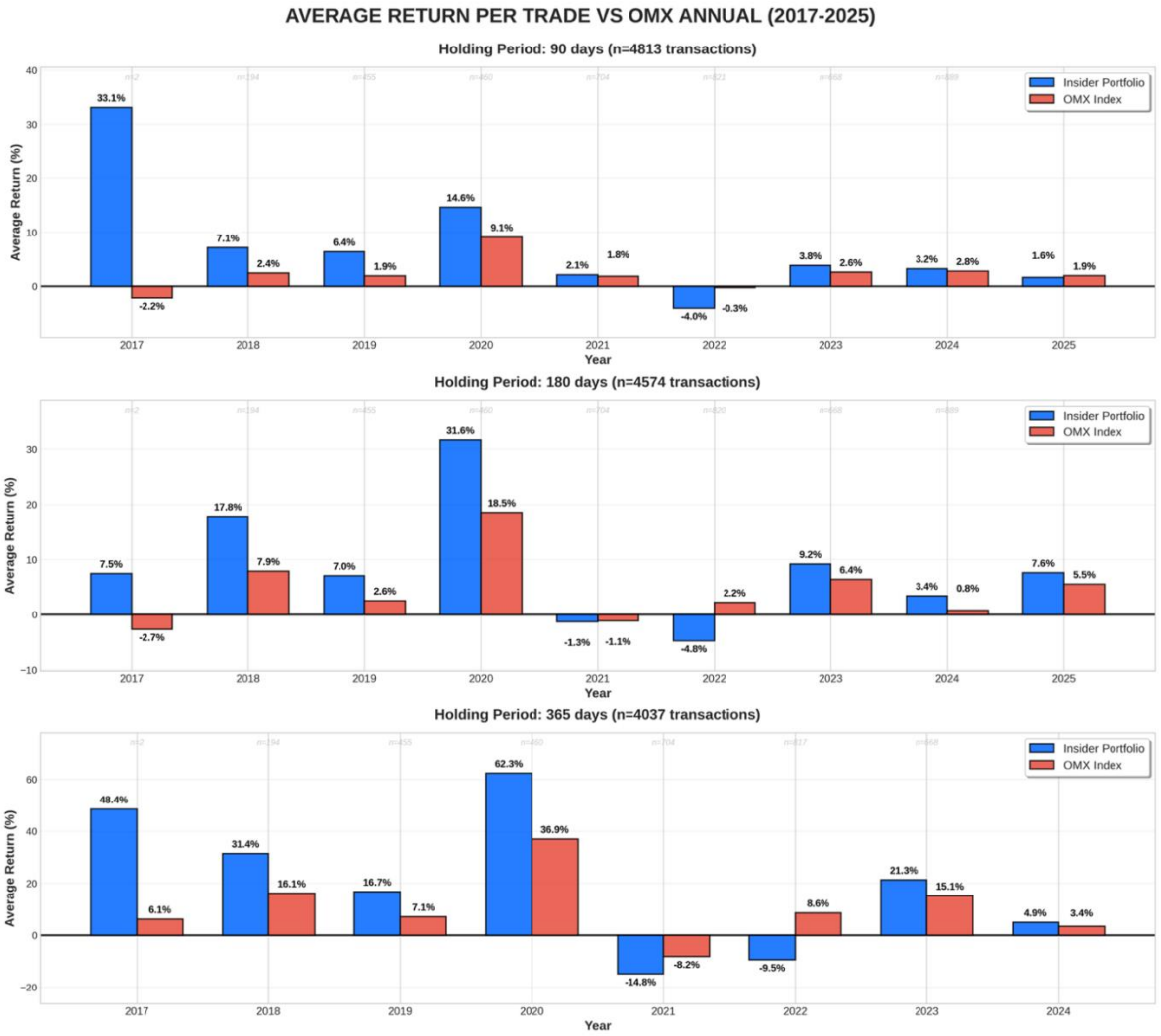


Figure 3: Average return per trade versus OMX Stockholm 30 Index annually.

Notably, the index returns vary from year to year. They reflect the average index movement during the holding periods of trades initiated that year, rather than the calendar-year performance. Another important note is the methodological distinctions: Figure 2 shows weighted averages across all trades, while figure 3 presents the unweighted annual means. This approach is better for illustrating year-to-year temporal variations in strategy performance and prevent periods with high transactions amounts from dominating the visualization.

5.1.3 Return Distributions.

The display of Return distribution (Figure 4) for each holding period helps us to examine the composition of the returns. Across all three distributions, it becomes evident that a relatively small number of exceptional gains have a substantial impact on the overall performance. While most trades generate modest returns around the median, occasionally outsized winners significantly elevate the average.

We see that the probability of capturing the exceptional "home run" trades increase substantially with longer holding periods. The 365 days strategy shows the most pronounced right tail with several positive outcomes in Figure 4 beyond 150%. This effect diminishes with shorter holding periods. Longer horizons give insider information more time to materialize in stock prices, resulting in a greater number of exceptional gains. However, this comes at the cost of increased overall dispersion and exposure to both upside and downside volatility.

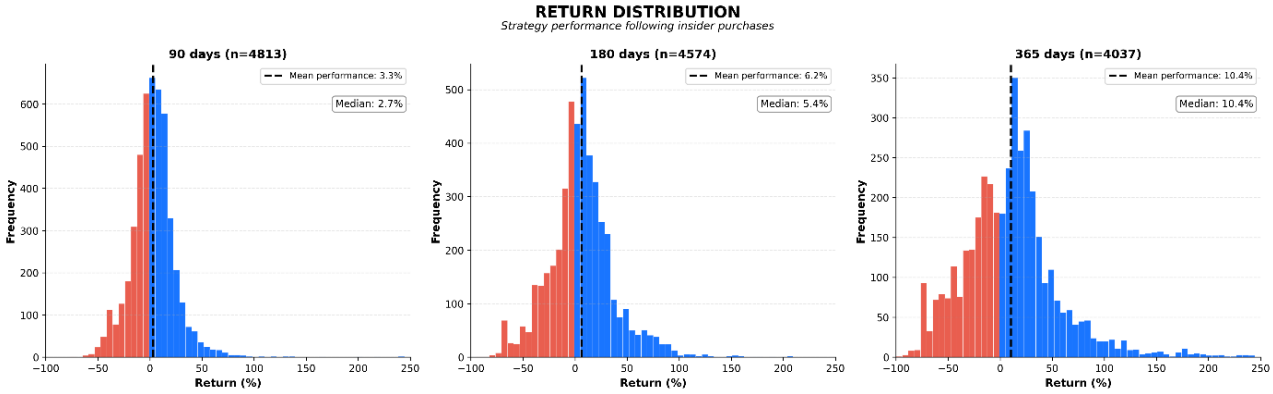


Figure 4: Return distribution for the different time horizons.

5.2 Differentiation by Market Capitalization

Having established that, in aggregate, insider purchases generate ER, we next examine whether these returns vary systematically across firm characteristics.

To test this, we segmented our dataset into four market capitalization groups for the different holding periods as presented in Figure 5.

Note: As mentioned in 5.1 holding periods represent different time horizons and the returns should not be compared directly. Compare market cap segments within each holding period (within columns).

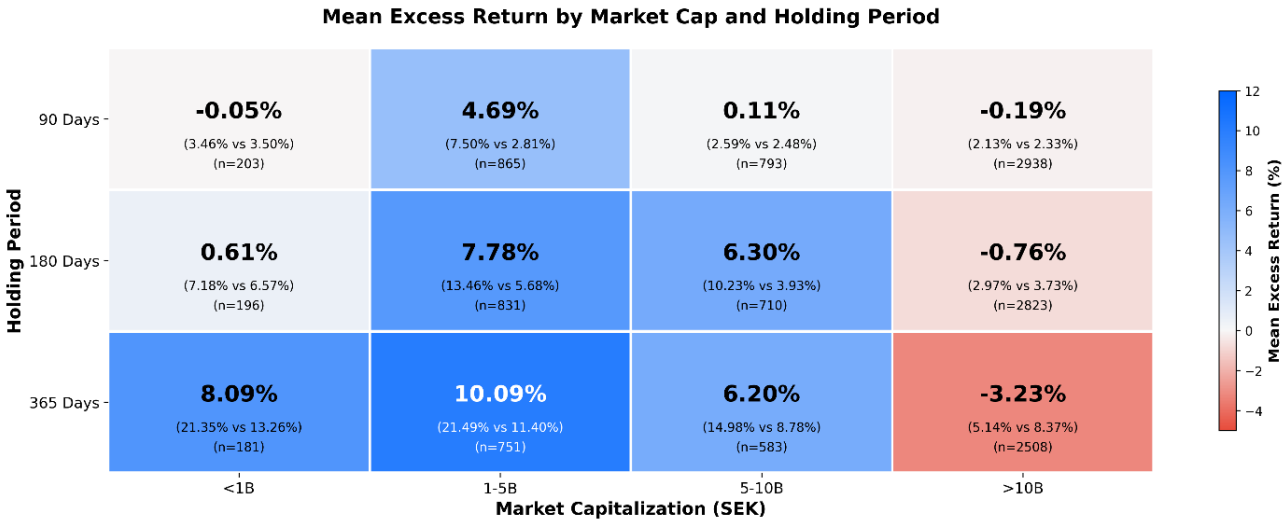


Figure 5: Mean Excess Return for each market cap segment across the different holding periods.

The distribution of observations across market cap segments is uneven. The >10B segment contains the largest number with 2508-2938 trades, compared to the <1B segment with only 181-203 trades. However, a larger sample does not imply stronger returns. The >10B segment underperforms the index across all holding periods, with MER ranging from -0.19% to -3.23%.

The results support the information asymmetry hypothesis, with the 1-5 billion SEK segment generating ER across all holding periods. In this segment the analyst coverage is limited yet the liquidity remains sufficient for meaningful positions. This finding suggests that insider information retains predictive value in mid-cap firms, where the gap between what the insiders know and what the market has priced in remains substantial.

A win rate exceeding 50% compared to index represents a genuine informational advantage. As shown in Figure 6, the 1-5B segments maintain this edge across all holding periods. This consistency, over a total of 751-865 trades per period, confirms that insider purchases carry predictive value in companies with a medium-sized market cap.

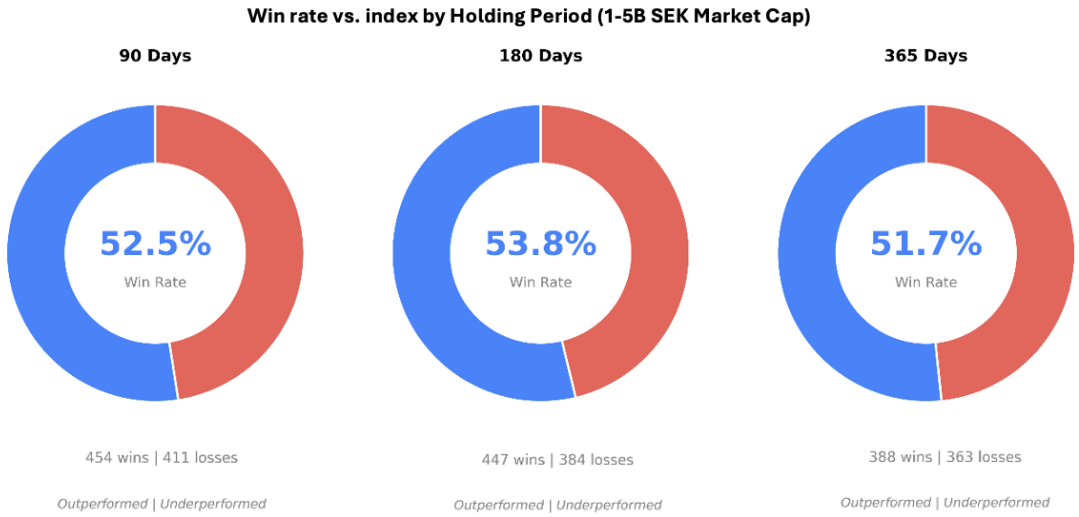


Figure 6: Proportions of trades outperforming the OMX Stockholm 30 Index within the 1-5B SEK market cap.

5.3 Differentiation by Transaction Value

Having established that market capitalization influences the return of the strategy, we further investigate whether transaction size influences these returns. The signaling hypothesis suggests that larger purchases reflect stronger conviction. To examine whether transaction size influences return across different market caps, we segmented the dataset into four transaction value categories across all holding periods, as presented in Figure 7.

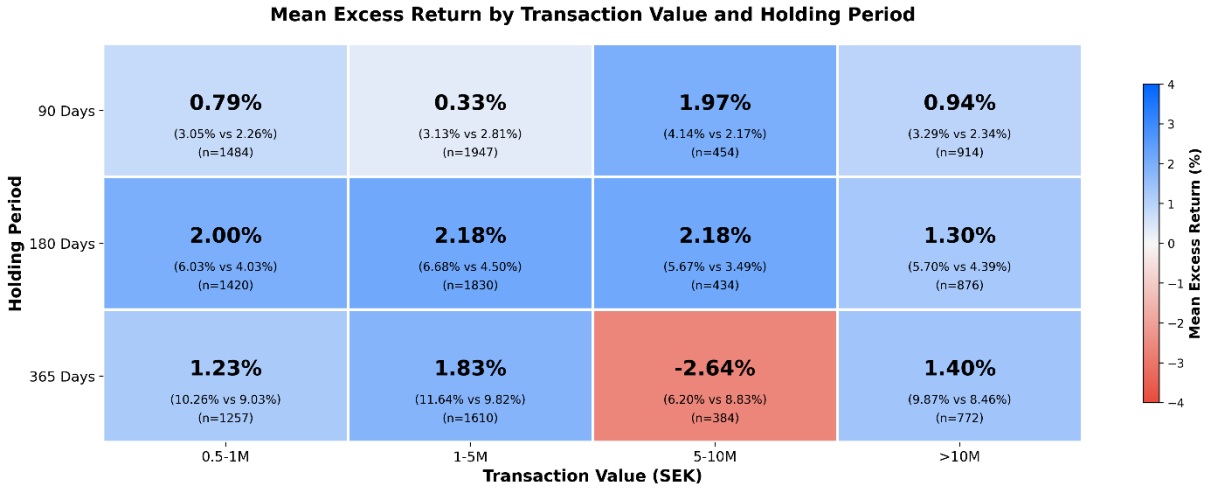


Figure 7: Mean Excess Return for each transaction value across the different holding periods.

The lower returns in Figure 7 compared to Figure 5 reflect an aggregation effect. Because Figure 7 pools transactions across all market caps, the negative returns from the >10B companies dilute the positive returns from medium-sized companies. To account for this, a nested heatmap was constructed. Figure 8 displays ER across holding periods and transaction value. Cells with fewer than 75 observations are hidden. The heatmap reveals a clear pattern. The >10B segment consistently underperforms with a large number of samples, while positive returns are concentrated in the segments with lower market cap.

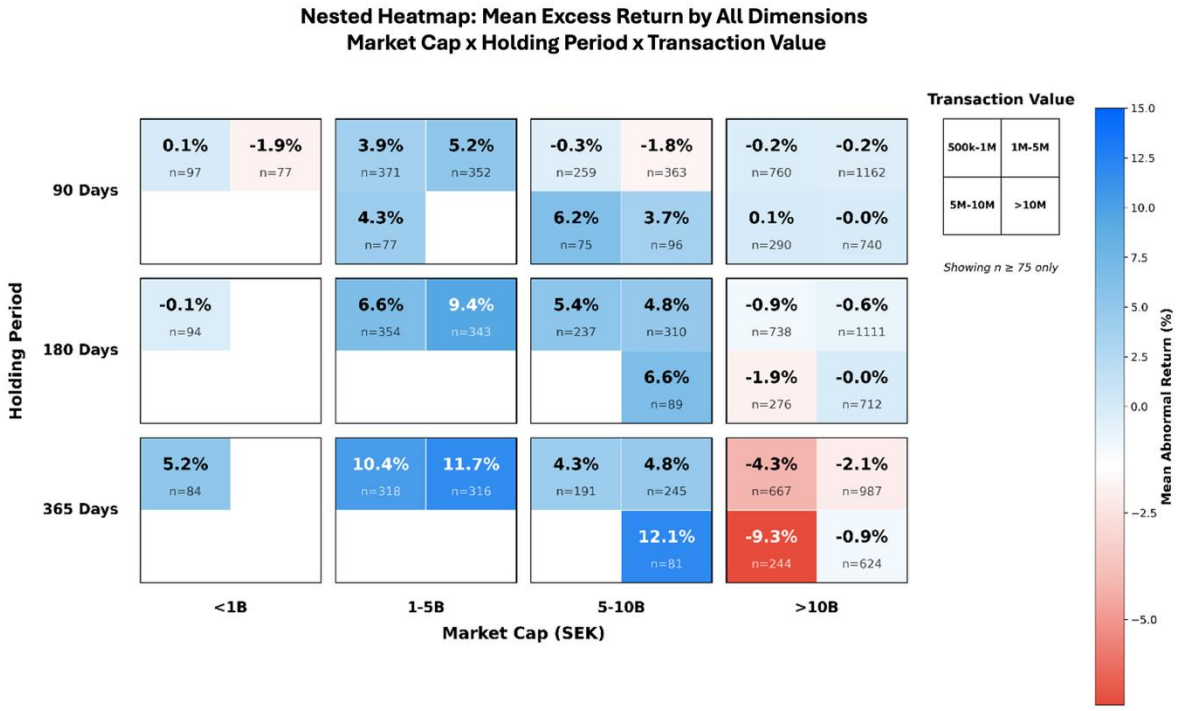


Figure 8: Mean Excess Return for market cap and transaction value over the different holding periods.

The 5-10B segment shows strong performance, but relative to the 1-5B segment the sample size is notably smaller. Within the 1-5B market cap column, six out of seven cells exceed 300 observations. Given this combination of strong returns and sufficient data, we proceed with a closer examination of this segment.

The following figures presents scatter plots of ER for the 1-5B SEK segment across all three holding periods.

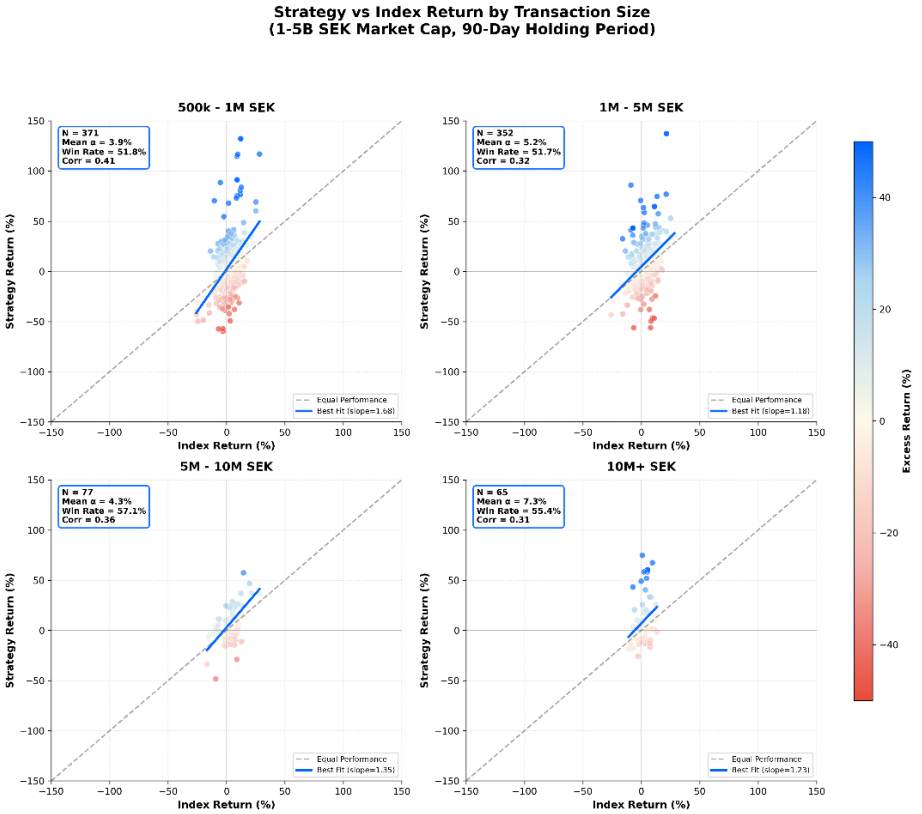


Figure 9a: Scatter plot visualizing Strategy Return vs Index across different transaction value intervals. 90-day holding period and 1-5B market cap.

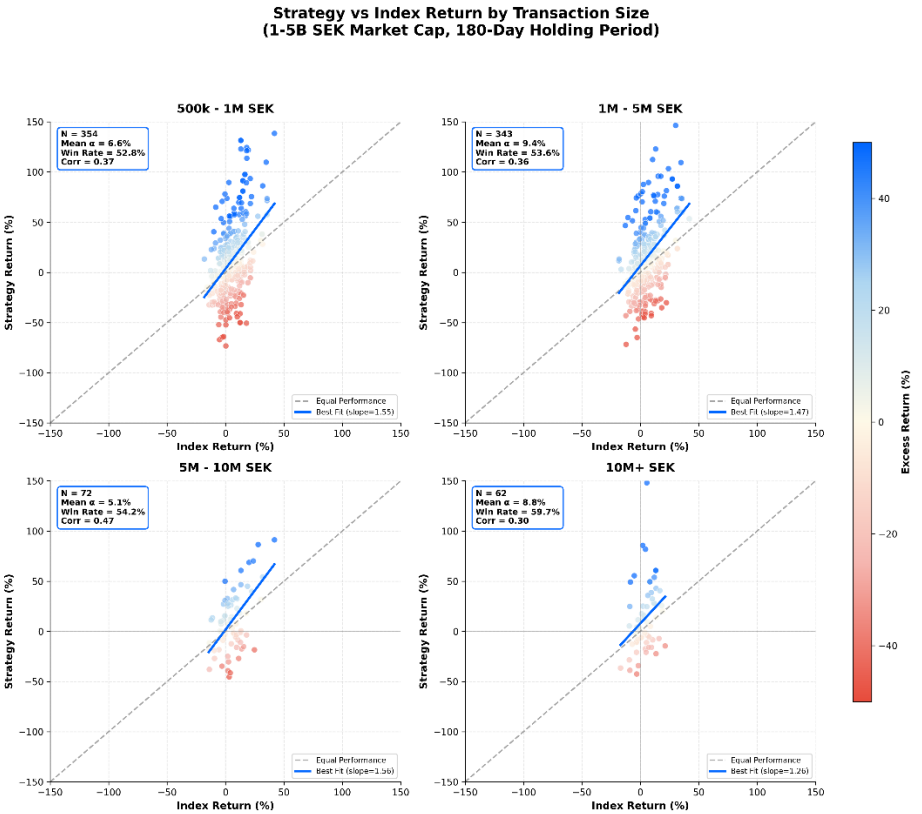


Figure 9b: Scatter plot visualizing Strategy Return vs Index across different transaction value intervals. 180-day holding period and 1-5B market cap.

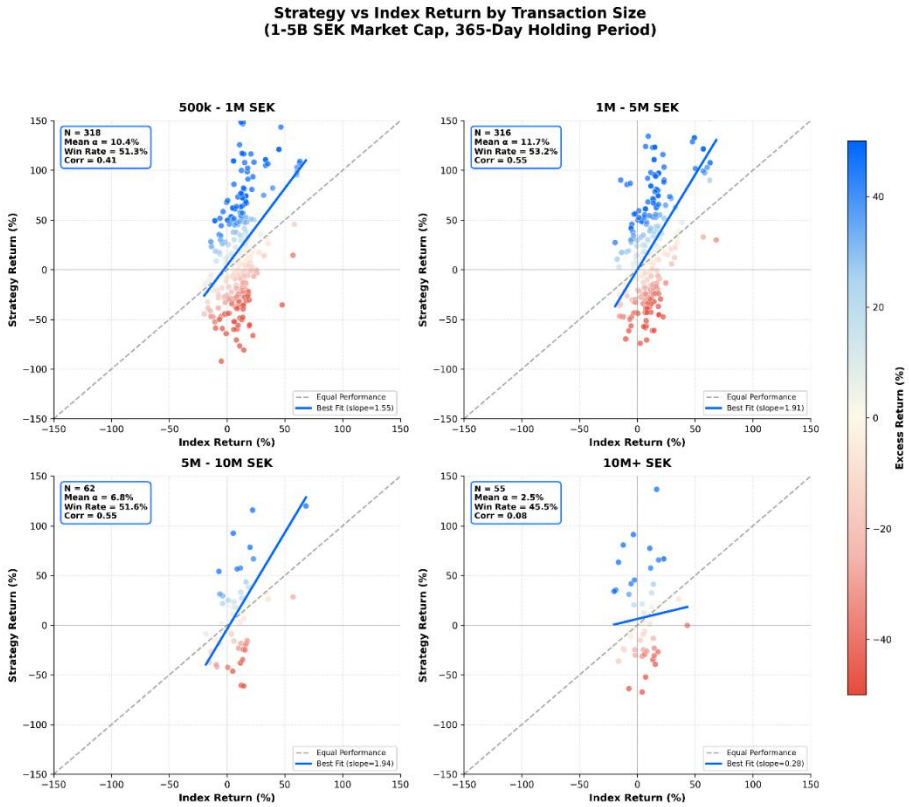


Figure 9c: Scatter plot visualizing Strategy Return vs Index across different transaction value intervals. 365-day holding period and 1-5B market cap.

Both the 500k-1M and the 1-5M transaction sizes demonstrate strong performance across the 90-, 180- and 365-day holding periods. The 1-5M category consistently holds a consistent edge, with a slightly higher win rate of 52-54% compared to 51-53% for the smaller transaction sizes.

The largest transactions (above 10M SEK) show mixed results. While performing well at the shorter horizons, with an excess return of up to 8.8% in 180 days, they underperform substantially over the 365-day period with a win rate below 50%. However, these findings should be interpreted with caution, as the 5M-10M and 10M+ categories comprise only 55-77 observations per period. In contrast the two best performing categories together represent over 630 trades per holding period, providing a more reliable sample.

5.4 Buying the Dip?

To test our hypothesis on this topic, we identified insider purchases made after a decline of >20% in the last 30 days. This identified 469-497 trades depending on holding period. We then compared their subsequent returns to the index.

Holding Period	Number of Transactions	Buy the Dip Return	Median Return	Avg. Index Return	MER	Win Rate (Positive)	Win Rate (vs Index)	Std. Dev. Buy the Dip Return
90 days	497	3.28%	4.10%	5.01%	-1.73%	54.1%	47.5%	35.13%
180 days	487	4.88%	3.54%	7.85%	-2.97%	51.5%	46.6%	50.47%
365 days	469	13.02%	3.35%	21.32%	-8.29%	51.2%	41.2%	72.02%

Figure 10: Results for "buying the dip" strategy.

Figure 10 shows that the “buying the dip” strategy consistently underperforms across all holding periods. Despite generating positive absolute returns, the strategy underperforms against index. The results are clear when looking at the win rates, 51.2% of the returns were positive in the 365-day holding period, but only 41.2% outperformed the index. However, when filtering the trades year by year, the picture is more nuanced, as shown in Figure 11.

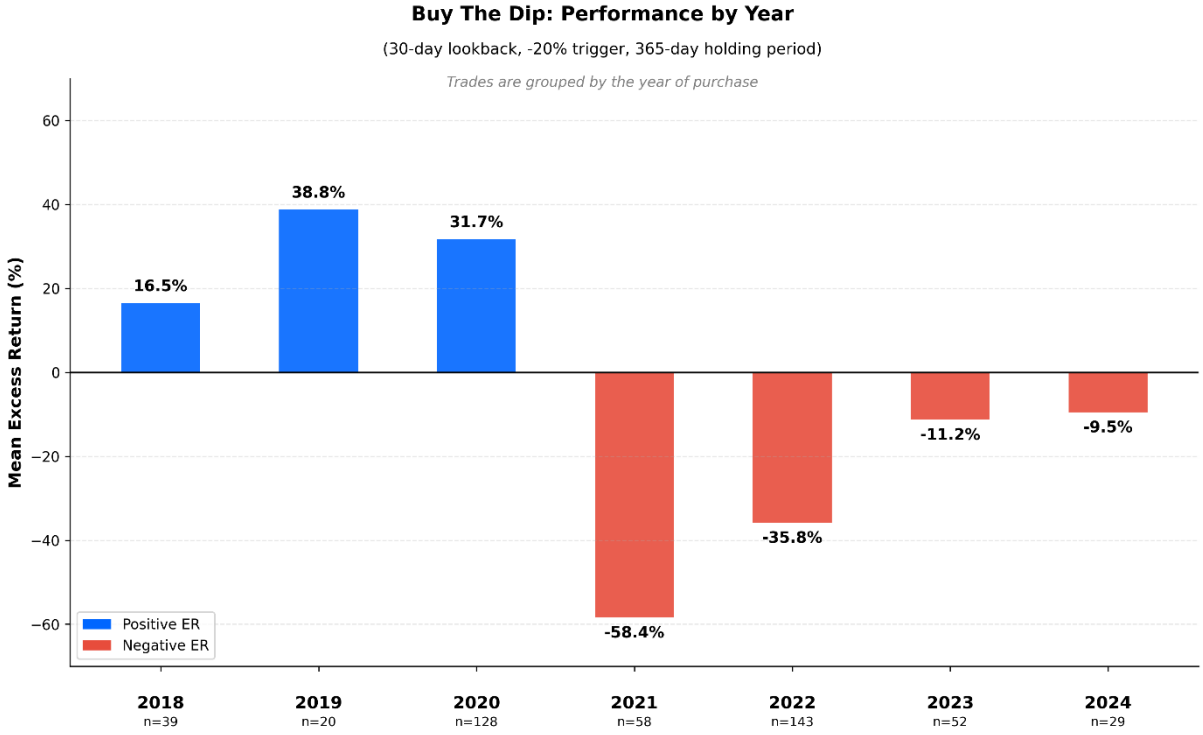


Figure 11: Annual performance of the “Buying the dip” strategy.

The 2021 results initially appeared remarkable, showing a mean excess return of -58%. Upon closer inspection of the trades within this year, the extreme underperformance was largely driven by the tech communication company Sinch AB. After rising 900% during 2020, the stock started declining from its peak in 2021. As the price fell, and triggered our -20% threshold, insiders started buying stock. The stock ultimately collapsed over 90% over the next year, which illustrates that insider purchases during a decline do not necessarily signal a turnaround.

A possible explanation is managerial hubris bias, discussed further in Section 7 (Malmendier & Tate, 2005). More broadly, buying the dip does not appear to be a reliable strategy for generating excess returns, as outcomes vary widely across years and market conditions.

5.5 Insider Position

Finally, we examine whether trades made by the CEO and CFO generate higher mean ER than trades made by other primary insiders. After applying the CEO/CFO position filter, 1,017-1,237 trades remained across the different holding periods. Our hypothesis is that CEOs and CFOs are more hands-on in the operation, giving them superior value relevant information. However, the results tell a different story:

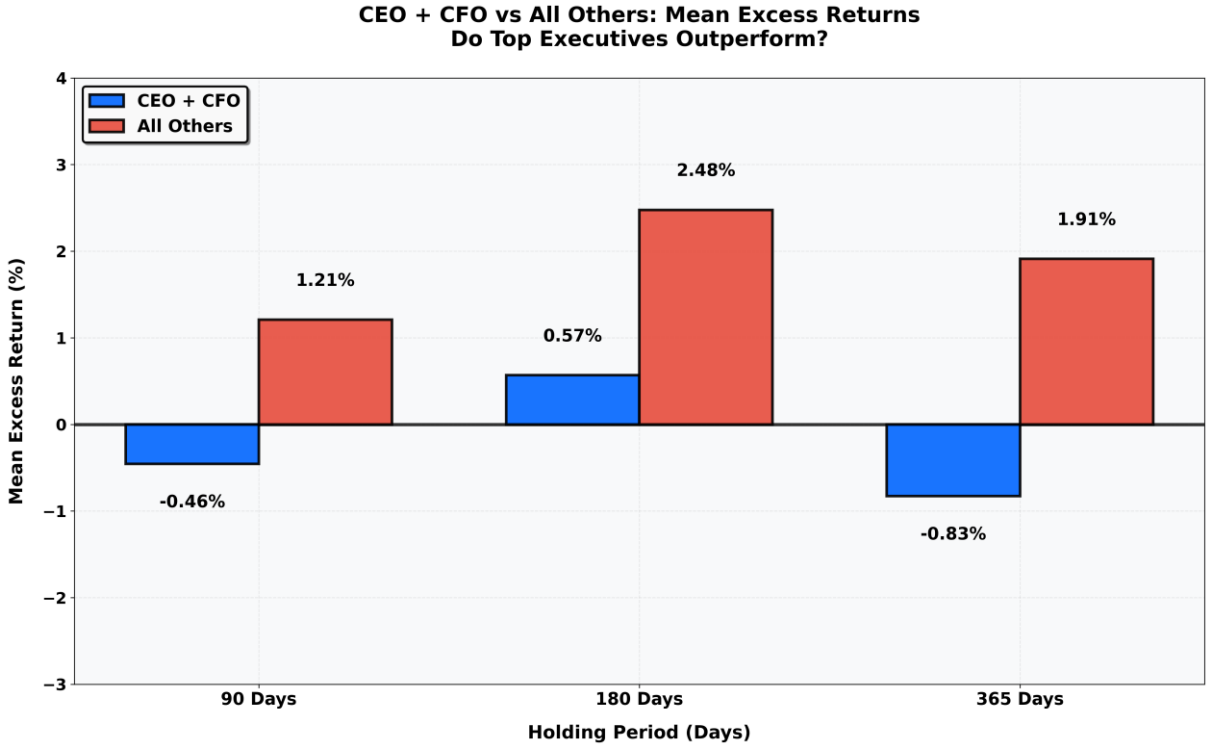


Figure 12. Average ER for the groups CEO+CFO and all other executives for all three horizons.

Figure 12 compares the average excess return for CEO + CFO against all other insider trades. Across all holding periods CEO + CFO trades underperform, with lower or negative ER. This indicates that key executives as CEO + CFO do not outperform other insiders.

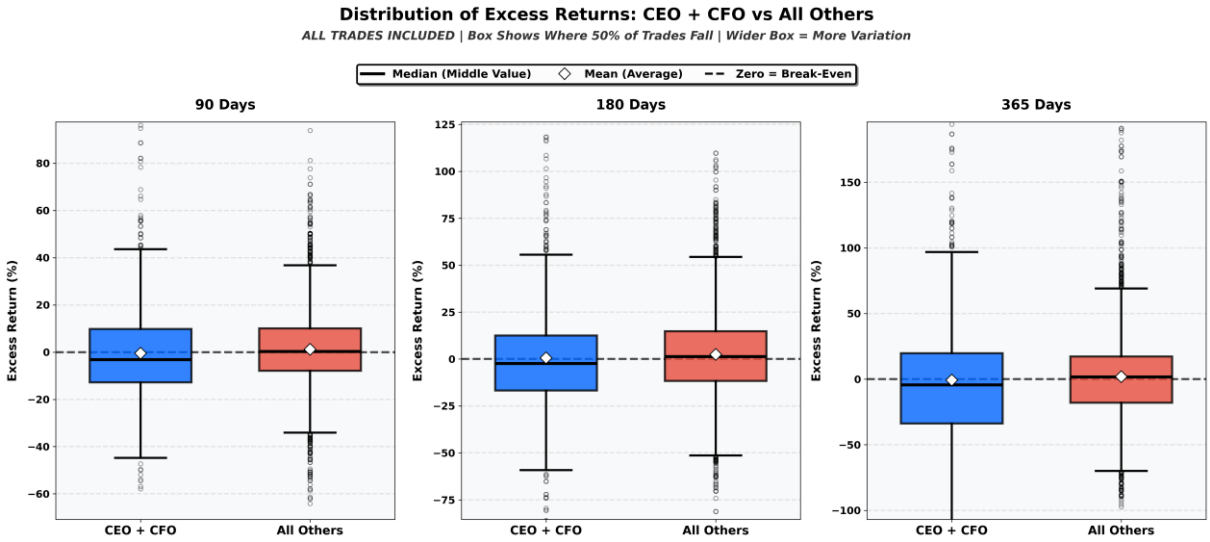


Figure 13: Comparison of ER distribution for the groups CEO + CFO with all other inside trades.

To understand this pattern more clearly, Figure 13 illustrates the full distribution of returns. The box represents the interquartile range (where 50% of trades fall), the thick black line indicates the median, and the white diamond shows the mean return. The CEO + CFO group shows wider boxes and more extreme values, indicating substantially higher volatility and less consistent outcomes compared to other insiders.

The combination of lower returns and higher volatility makes CEO + CFO trades comparatively less attractive than other insider trades.

6. Statistical analysis

To complement the descriptive findings in Section 5, we will conduct a series of statistical tests to evaluate the robustness and statistical significance of our findings. First, we test whether insider purchases as a whole generate excess returns relative to the OMXS30 using one-sample t-tests. Second, for each subgroup analysis (market capitalization, transaction value, insider position, and buying-the-dip trades), we compare groups using one-sided Mann–Whitney U-tests. Finally, for subgroups that appear to outperform in the descriptive analysis, we run additional one-sample t-tests against the index.

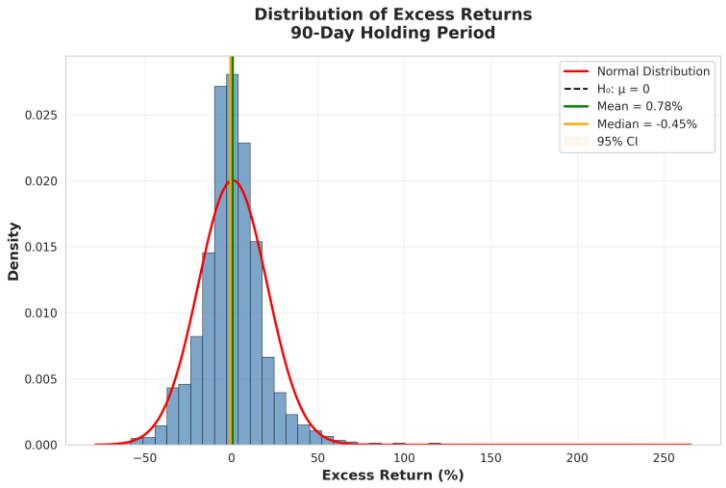
All tests are done with a significance level of $\alpha = 0.05$.

6.1 One sample T-test of Mean Excess return > 0

Having established descriptive evidence of positive excess returns in Section 5, we now test whether these returns are statistically different from zero. We evaluate this using a one-sided one-sample t-test with the following hypotheses:

H_0: average excess return ≤ 0

H_1: average excess return > 0




```
ONE-SAMPLE T-TEST
================================================

Hypothesis Testing:
  H₀: μ ≤ 0  (No excess return)
  H₁: μ > 0  (Positive excess return)

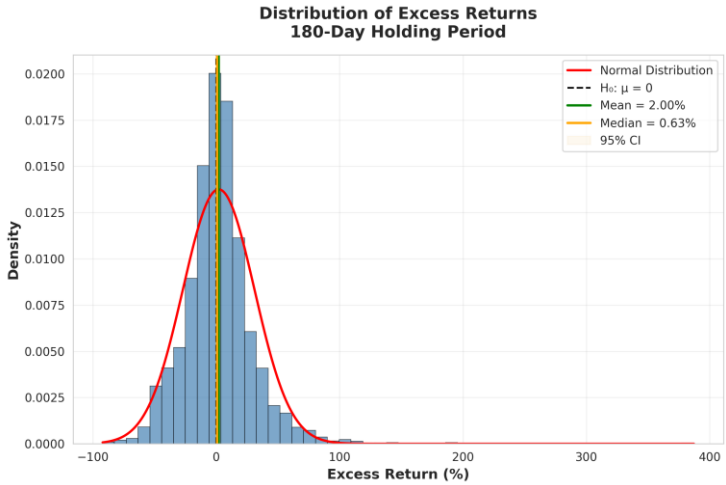
================================================
DESCRIPTIVE STATISTICS
================================================
  Sample Size (n)              4,813
  Mean Return                  0.78%
  Median Return                -0.45%
  Std. Deviation               19.91%
  Std. Error                   0.2870%

================================================
TEST STATISTICS
================================================
  t-statistic                  2.7234
  p-value (one-tailed)         0.003243

  95% Confidence Interval:
    [0.22%, 1.34%]

================================================
EFFECT SIZE
================================================
  Cohen's d                    0.0393
```

Figure 14a : Distribution of ER for the 90-day holding period.




```
ONE-SAMPLE T-TEST
================================================

Hypothesis Testing:
  H₀: μ ≤ 0  (No excess return)
  H₁: μ > 0  (Positive excess return)

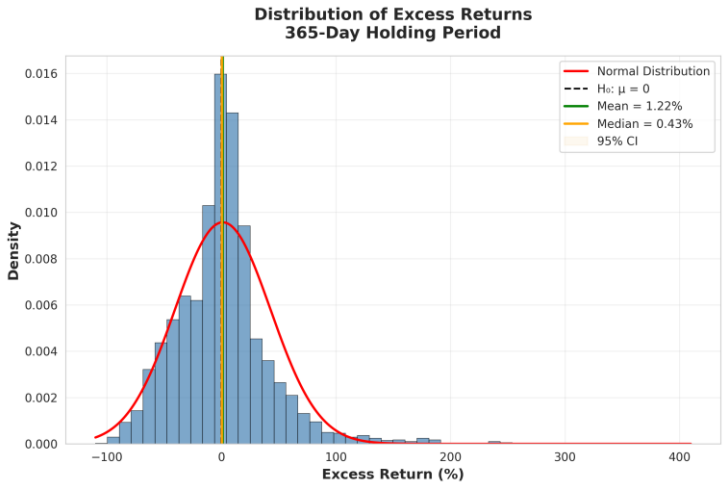
================================================
DESCRIPTIVE STATISTICS
================================================
  Sample Size (n)              4,574
  Mean Return                  2.00%
  Median Return                0.63%
  Std. Deviation               29.01%
  Std. Error                   0.4289%

================================================
TEST STATISTICS
================================================
  t-statistic                  4.6579
  p-value (one-tailed)         0.000002

  95% Confidence Interval:
    [1.16%, 2.84%]

================================================
EFFECT SIZE
================================================
  Cohen's d                    0.0689
```

Figure 14b: Distribution of ER for the 180-day holding period.




```
ONE-SAMPLE T-TEST
================================================

Hypothesis Testing:
  H₀: μ ≤ 0  (No excess return)
  H₁: μ > 0  (Positive excess return)

================================================
DESCRIPTIVE STATISTICS
================================================
  Sample Size (n)              4,037
  Mean Return                  1.22%
  Median Return                0.43%
  Std. Deviation               41.70%
  Std. Error                   0.6563%

================================================
TEST STATISTICS
================================================
  t-statistic                  1.8614
  p-value (one-tailed)         0.031381

  95% Confidence Interval:
    [-0.07%, 2.51%]

================================================
EFFECT SIZE
================================================
  Cohen's d                    0.0293
```

Figure 14c: Distribution of ER for the 365-day holding period.

The results (Figure 14) show statistically significant positive excess returns across all three holding periods. The 90- and 180-day periods are significant at the 1% level ($p = 0.003$ and $p < 0.001$), while the 365-day period is significant at the 5% level ($p = 0.031$). This indicates that insider transactions contain predictive value, with the strength and evidence varying across time horizons. Effect sizes (Cohen's d) are small in all cases, indicating that although the signal is statistically detectable, the magnitude is modest relative to the volatility of individual stock returns. While the t-tests show statistically significant excess returns, the corresponding effect sizes are small, meaning that the economic magnitude of the signal is modest.

6.2 Mann-Whitney U-Tests: Market Cap Segment Analysis

To evaluate if the excess return varies systematically across companies with different market capitalizations, we conducted a non-parametric one-sided Mann–Whitney U-test. We apply a Bonferroni correction to account for multiple comparisons, adjusting the significance level by dividing α by the number of group-wise tests performed. As the 1–5B SEK segment delivered the strongest excess returns, we compare this group against all other market-cap segments. We define our hypotheses:

H₀: *ER for the 1–5B segment ≤ ER in the comparison group*

H₁: *ER for the 1–5B segment > ER in the comparison group*

Mann-Whitney U-Test Results - 90-Day Holding Period
H_1: 1-5B SEK segment outperforms other segments (one-sided test)
Bonferroni-corrected α = 0.01667 (for 3 comparisons)

Comparison	Comparison Group MER	MER Difference	p-value	Significance	Bonferroni Significant?
1-5B vs <1B	-0.05%	+4.74%	0.031621	✓ **Significant** *	✗ No
1-5B vs 5-10B	0.11%	+4.58%	0.001296	✓✓ **Significant** **	✓ **Yes**
1-5B vs >10B	-0.19%	+4.88%	0.004093	✓✓ **Significant** **	✓ **Yes**

Figure 15a: Results of the Mann-Whitney U-tests for market cap for the 90-day holding period.

180-day Holding Period

Comparison	Comparison Group MER	MER Difference	p-value	Significance	Bonferroni Significant?
1-5B vs <1B	0.61%	+7.17%	0.001058	✓✓ **Significant** **	✓ **Yes**
1-5B vs 5-10B	6.30%	+1.48%	0.819161	✗ Not Significant	✗ No
1-5B vs >10B	-0.76%	+8.53%	0.000068	✓✓✓ **Highly Significant** ***	✓ **Yes**

Figure 15b: Results of the Mann-Whitney U-tests for market cap for the 180-day holding period.

365-day Holding Period

Comparison	Comparison Group MER	MER Difference	p-value	Significance	Bonferroni Significant?
1-5B vs <1B	8.09%	+2.00%	0.000431	✓✓✓ **Highly Significant** ***	✓ **Yes**
1-5B vs 5-10B	6.20%	+3.89%	0.380982	✗ Not Significant	✗ No
1-5B vs >10B	-3.23%	+13.32%	0.000005	✓✓✓ **Highly Significant** ***	✓ **Yes**

Figure 15c: Results of the Mann-Whitney U-tests for market cap for the 365-day holding period.

The results show that the 1-5B segment significantly outperforms both small-cap (<1B) and large-cap (>10B) segments across all holding periods. These results remain robust even after the Bonferroni correction (α-adjusted = 0.0167), with the comparison against large-cap firms showing high significance across all horizons (all $p < 0.005$). However, the small-cap comparison in the 90-day horizon does not remain significant after correction ($p = 0.032 > 0.0167$).

When comparing the 1-5B segment with the 5-10B segment, we observe a significant difference only in the 90-day period ($p = 0.0013$). This advantage disappears in the 180-day ($p = 0.82$) and 365-day ($p = 0.38$) periods. Overall, this suggests that the informational advantage of insider trading signals extends beyond the narrow 1-5B SEK range.

To test whether the outperformance of the index for the 1-5B segment is statistically significant, we conduct a one-sided t-test on ER within this market cap. The results (Figure 16) show that the mid-cap segment deliver highly significant positive ER across all periods. Moreover, we observe a stronger effect size (Cohen's d) compared to the full sample, indicating a strong and economically meaningful signal in this segment.

One-Sample t-Test Results
H_1: Mean excess returns are significantly greater than zero

Period	n	Outperformance	t-statistic	p-value	Significance	Cohen's d
90 days	865	4.69%	4.82	0.000001	***	0.1641
180 days	831	7.78%	5.63	0.000001	***	0.1953
365 days	751	10.09%	5.54	0.000001	***	0.2023

Figure 16: Results of the T-test for market cap 1-5B.

6.3 Mann-Whitney U-Tests: Transaction Value

To examine the strength of insider signals based on the transaction value, we compare the 1-5 million SEK transaction size against all other transaction-size groups. We perform the same Mann-Whitney U-test as for market cap, with the following hypothesis:

H_0: *ER for 1–5M trades ≤ ER for the comparison group*

H_1: *ER for 1–5M trades > ER for the comparison group*

Mann-Whitney U-Test Results - 90-Day Holding Period
H_1: 1-5M SEK segment outperforms other segments (one-sided test)
Bonferroni-corrected α = 0.01667 (for 3 comparisons)

Comparison	Comparison Group MER	MER Difference	p-value	Significance	Bonferroni Significant?
1-5M vs <1M	0.84%	-0.45%	0.479350	✗ Not Significant	✗ No
1-5M vs 5-10M	1.99%	-1.61%	0.945376	✗ Not Significant	✗ No
1-5M vs >10M	0.94%	-0.55%	0.975640	✗ Not Significant	✗ No

Figure 17a: Results of the Mann-Whitney U-Test for transaction volume for the 90-day holding period.

180-day Holding Period

Comparison	Comparison Group MER	MER Difference	p-value	Significance	Bonferroni Significant?
1-5M vs <1M	2.05%	+0.18%	0.387011	✗ Not Significant	✗ No
1-5M vs 5-10M	2.23%	+0.01%	0.464410	✗ Not Significant	✗ No
1-5M vs >10M	1.30%	+0.94%	0.734746	✗ Not Significant	✗ No

Figure 17b: Results of the Mann-Whitney U-Test for transaction volume for the 180-day holding period.

365-day Holding Period

Comparison	Comparison Group MER	MER Difference	p-value	Significance	Bonferroni Significant?
1-5M vs <1M	1.37%	+0.56%	0.085196	✗ Not Significant	✗ No
1-5M vs 5-10M	-2.48%	+4.42%	0.019766	✓ Significant *	✗ No
1-5M vs >10M	1.34%	+0.59%	0.804248	✗ Not Significant	✗ No

Figure 17c: Results of the Mann-Whitney U-Test for transaction volume for the 365-day holding period.

Across all three holding periods, none of the comparisons remain statistically significant after applying the Bonferroni correction. Only the comparison between 1-5M vs 5-10M shows significance at the conventional 5% level. However, it does not remain significant under the adjusted threshold ($\alpha = 0.0167$).

Since no transaction size-segment consistently outperform others, additional testing against index is not justified. Overall, these results indicate that there is no evidence that insider transactions in the 1–5M range generate higher ER than other transaction sizes.

6.4 Mann-Whitney U-Tests: Insider Position

To determine whether the observed underperformance of CEO and CFO trades is statistically significant rather than a sample-specific outcome we conduct a one-sided Mann-Whitney U-test. Since we are comparing two groups, we exclude the Bonferroni correction. We formulate the following hypothesis:

H_0: *CEO + CFO ER* $\geq$ *ER of other insiders*

H_1: *CEO + CFO ER* $<$ *ER of other insiders*

Mann-Whitney U-Test: CEO+CFO vs Other Insiders (One-Sided Test)
H_1: CEO+CFO trades yield lower excess returns than other insider trades

Days	Comparison	Comparison Group MER	MER Difference	p-value	Significance
90	CEO+CFO < Others	1.21%	-1.64%	0.000000	✓✓✓ Highly Significant ***
180	CEO+CFO < Others	2.47%	-1.86%	0.000001	✓✓✓ Highly Significant ***
365	CEO+CFO < Others	1.83%	-2.37%	0.000335	✓✓✓ Highly Significant ***

Figure 18: Results of the Mann-Whitney U-Test for different insider groups.

The results show that the underperformance of CEO + CFO trades compared to other insiders is significant. All p-values are well below the 5% threshold, allowing us to reject H_0. This implies that the observed underperformance is statistically robust.

Our focus in this subsection is on the internal performance differences within insider groups. Therefore, further tests against the index are not necessary.

6.5 Mann-Whitney U-Tests: Buying the Dip

To test whether "buying the dip" insider trades underperform normal insider trades, we conduct a one-sided Mann–Whitney U-test comparing trades made after a ≥20% price drop over a 30-day lookback window with all other insider purchases. The hypotheses are:

H_0: *ER for dip-buying trades ≥ ER for normal insider trades*

H_1: *ER for dip-buying trades < ER for normal insider trades*

Mann-Whitney U-Test: Buy After Dip vs Normal Trades (One-Sided Test)
H_1: Buying after a 20%+ dip (30d lookback) yields lower excess returns than normal trades

Days	Comparison	Comparison Group MER	MER Difference	p-value	Significance
90	Dip < Normal	1.06%	-2.79%	0.000232	✓✓✓ Highly Significant ***
180	Dip < Normal	2.58%	-4.31%	0.000042	✓✓✓ Highly Significant ***
365	Dip < Normal	2.47%	-4.20%	0.007835	✓✓ Significant **

Figure 19: Results of the Mann-Whitney U-Test for buying after the dip strategy.

The results show a clear and statistically significant pattern: buying after a major dip does not generate higher excess returns compared to normal insider trades. Across all holding periods, dip-buying yields 2.8%–4.3% lower MER**,** with all p-values well below the 5% level. This confirms that "buying the dip" is a systematically weaker signal than ordinary insider purchases and provides no support for using it as a profitable strategy.

7. Discussion

7.1 Interpretation of Findings

Our findings show that insider purchases generate statistically significant excess returns. This supports informed trader theory: insiders hold value-relevant information that is not immediately incorporated into market prices. As Sveen emphasized in our interview, insiders operate under persistent informational asymmetry, and our results confirm that this advantage has measurable, although economically small, predictive power.

The concentration of returns in the 1–5B SEK segment aligns with the information-asymmetry mechanism. Small- and mid-cap firms typically face limited analyst coverage and less frequent public attention, increasing the gap between insider knowledge and market expectations. These results are in line with the original formulation of EMH Under Fama's (1991) cost-based view of efficiency, such deviations are consistent with efficiency, as insiders face far lower information-acquisition costs than outside investors in these firms. This contrasts with Fama's (1970) semi-strong formulation, which does not account for information acquisition costs and would therefore view such excess returns as deviations from efficiency.

While signal theory suggests that following trades with higher transaction values should produce higher excess returns, our results show no such effect. One possible explanation is that executing large transactions is harder in small-and-midcap companies, precisely the segment we found made the highest excess return.

The underperformance of CEO and CFO trades relative to other insiders challenges the intuitive hierarchy of informational value. MAR's 30-day closed period restricts the timing of executive trades, reducing their ability to act strategically. In addition, executive purchases may be more visible to the market, leading to faster price incorporation before outside investors can act. The stronger performance of other insiders suggests that informational advantages are context-dependent rather than purely positional.

Finally, trades executed after large price declines underperform normal insider purchases. This pattern is consistent with hubris bias, where insiders overestimate their own judgment and interpret sharp drawdowns as temporary mispricing rather than signals of deteriorating fundamentals (Malmendier & Tate, 2005). These results suggest that insiders' informational advantage is most reliable in normal trading conditions and not in periods dominated by drawdowns.

Overall, the evidence indicates that insider transactions convey information, but the strength of this signal varies systematically across firms, insiders, and trading contexts. The findings support informed trader theory, highlight limits to semi-strong efficiency, and demonstrate that informational value is segmented rather than uniform.

7.2 Limitations

7.2.1 Data limitations

Our dataset relies on disclosures from Finansinspektionen and historical prices from Yahoo Finance. Corporate actions, such as delisting and exclusion of companies not found with Yahoo Finance may introduce survivorship bias. Return calculations exclude dividends from the returns, while the benchmarked index includes them. This makes our excess return estimates conservative. The transaction cost and index management fee are also excluded.

Yahoo Finance prices are adjusted for stock splits and corporate actions. When the insider's reported transaction price falls outside the adjusted daily range, we use the midpoint to calculate their return. This introduces minor inaccuracy in insider return calculations but does not affect strategy returns, which use publication day + 1 as entry.

7.2.2 Analytical limitations

Each transaction is analyzed independently. This isolates the informational content in each trade, yet it fails to reflect a realistic portfolio with capital constraints or compounding returns.

Several segmented analyses suffer from small samples. In addition, the sample period (2016-2025) includes the COVID-19 crash, during which insider performance was exceptional. Results may therefore differ in other market conditions.

The analysis compares each trade's return with the index over identical holding periods, ensuring that ER are measured consistently across different timeframes. However, the 90- and 180-day strategies cannot simply be annualized, since this would assume a steady flow of trading signals that allows continuous reinvestment. Given the limited number of qualifying trades, this assumption is unlikely to hold, and annualized figures would overstate realistically achievable portfolio performance. This deviates from a passive portfolio where an investor would hold the index continuously rather than buying and selling alongside each trade.

Statistical inference in our study is subject to several limitations. Return distributions are non-normal and heavily tailed, making parametric tests such as the t-test sensitive to outliers and deviations from normality. In the segmented analysis small-sized companies reduce statistical power and increase the likelihood of failing to detect true effects. Moreover, testing multiple subgroups introduces a risk of false positives, even after applying Bonferroni corrections.

7.3 Practical Implications

"If you beat the market more than 50% of the time, you're good. If you beat it 51% of the time, you're extremely good. And if you beat it 52%, you're phenomenal."
– Peder Viervoll, Head of Quantitative Trading, NBIM

Results show that optimal trading strategies achieve win rates of up to 52-53%, and 50.8% annually for all trades. These results suggest that insider transactions contain exploitable information even in a modern regulatory Nordic environment.

Our findings imply that insider signals should not be treated as uniformly informative. The predictive value in our analysis is concentrated in mid-cap firms, where analyst coverage is more limited than for large-cap companies. In other words, the market appears more efficient in firms with broad analyst coverage. This has implications for how market participants interpret disclosed transactions and shows that insider trades are not equally informative.

The persistent underperformance of “buying-the-dip” insider trades suggests that large price declines do not strengthen the insider signal and should not be interpreted as unusually attractive opportunities. Additionally, CEO and CFO trades show weaker informational value than trades by other insiders and should not be prioritized when using insider transactions as a screening tool.

Overall, our strong and significant results present several practical implications for outside investors. Our findings suggest that firm size matters more than specific insider positions such as CEO and CFO. Investors seeking to maximize their gains over a 365-day period should, according to our findings, follow insider trades in the mid-cap segment. However, implementing this strategy requires following a large number of trades and diversifying across many firms, as the edge only materializes in the aggregate.

Finally, these implications apply to Sweden during the 2016–2025 period. Market structure, disclosure practices, and coverage intensity differ across countries, meaning that insider-based strategies may not yield similar outcomes elsewhere. Practical implications must also recognize that patterns found in historical data do not necessarily persist in future market conditions.

8. Conclusion

The aim of this study was to examine whether trades made by primary insiders are followed by excess returns, and whether specific trade or firm characteristics strengthen the signal. Across all three holding periods, insider purchases are on average associated with modest positive excess returns, indicating that such trades contain limited but detectable informational value.

The signal is not uniform across all insiders or firms. Purchases in mid-sized firms show consistently stronger performance than those in smaller and larger firms. However, transaction size and purchase timing provide little incremental explanatory power. Furthermore, trades made by CEOs and CFOs underperform those made by other insiders, suggesting that executives offer weaker informational signals than commonly assumed.

Overall, the findings are consistent with previous research and indicate that following primary insider transactions can generate significant excess returns.

To build on these findings, future research could incorporate a more explicit treatment of risk. This could be done by constructing an insider-based portfolio, such as *Innsideporteføljen* and evaluating risk-adjusted performance using standard asset pricing models. In addition, further studies could examine whether insider *sell* transactions carry the same informational content as purchases. Although this topic has been researched previously, updated evidence is needed to assess whether the documented patterns persist in more recent data and across different market conditions.

References

Avanza. (n.d.). *Courtageklasser.* https://www.avanza.se/konton-lan-prislista/prislista/courtageklasser.html

Bernard, V. L., & Thomas, J. K. (1989). Post-earnings-announcement drift: Delayed price response or risk premium? *Journal of Accounting Research, 27*(Supplement), 1–36. https://doi.org/10.2307/2491062

Bhushan, R. (1989). Firm characteristics and analyst following. *Journal of Accounting and Economics, 11*(2–3), 255–274. https://doi.org/10.1016/0165-4101(89)90008-6

Brennan, M. J., & Subrahmanyam, A. (1996). Market microstructure and asset pricing: On the compensation for illiquidity in stock returns. *Journal of Financial Economics, 41*(3), 441–464. https://doi.org/10.1016/0304-405X(95)00870-K

De Bondt, W. F. M., & Thaler, R. (1985). Does the stock market overreact? *The Journal of Finance, 40*(3), 793–805. https://doi.org/10.2307/2327804

European Parliament and Council. (2014). *Regulation (EU) No 596/2014 of 16 April 2014 on market abuse (Market Abuse Regulation).* Official Journal of the European Union, L 173, 1–61. https://eur-lex.europa.eu/eli/reg/2014/596/oj

Fama, E. F. (1970). Efficient capital markets: A review of theory and empirical work. *The Journal of Finance, 25*(2), 383–417. https://doi.org/10.2307/2325486

Finansinspektionen. (n.d.). *Marknadssök – Official MAR disclosure database.* https://marknadssok.fi.se/publiceringsklient

Harris, L. (2003). *Trading and exchanges: Market microstructure for practitioners* (Draft ed.). Oxford University Press. https://www.acsu.buffalo.edu/~keechung/MGF743/Readings/Trading-Exchanges-Market-Microstructure-Practitioners%20Draft%20Copy.pdf

Jegadeesh, N., & Titman, S. (1993). Returns to buying winners and selling losers: Implications for stock market efficiency. *The Journal of Finance, 48*(1), 65–91. https://doi.org/10.1111/j.1540-6261.1993.tb04702.x

Lakonishok, J., & Lee, I. (2001). Are insider trades informative? *Review of Financial Studies, 14*(1), 79–111. https://doi.org/10.1093/rfs/14.1.79

Malmendier, U., & Tate, G. (2005). CEO overconfidence and corporate investment. *The Journal of Finance, 60*(6), 2661–2700. https://doi.org/10.1111/j.1540-6261.2005.00813.x

Nasdaq. (n.d.). *OMXS30 index fact sheet.* https://indexes.nasdaqomx.com/docs/FS_OMXS30.pdf

Ozkan, A., & Trzeciakiewicz, A. (2014). Informative content of insider purchases: Evidence from the financial crisis. *Review of Behavioral Finance, 6*(1), 26–45. https://doi.org/10.1108/rbf-01-2014-0008

Seyhun, H. N. (1986). Insiders' profits, costs of trading, and market efficiency. *Journal of Financial Economics, 16*(2), 189–212. https://doi.org/10.1016/0304-405X(86)90060-7

Spence, M. (1973). Job market signaling. *The Quarterly Journal of Economics, 87*(3), 355–374. https://doi.org/10.2307/1882010

Sundling, C., & Verschuur, C. (2014). *How to yield abnormal return by replicating insider trades: A study on the Swedish stock market* (Bachelor thesis). University of Gothenburg, School of Business, Economics and Law. https://gupea.ub.gu.se/bitstream/handle/2077/36369/gupea_2077_36369_1.pdf

Wang, W., Shin, Y.-C., & Francis, B. B. (2012). Are CFOs' trades more informative than CEOs' trades? *Journal of Financial and Quantitative Analysis, 47*(4), 743–762. https://doi.org/10.1017/S0022109012000257

Appendix

Interview with Sigurd Sveen, translated from Norwegian to English

Q1: How is "Innsideporteføljen" structured?

The portfolio consists of five different companies listed on the Oslo Stock Exchange, where primary insiders invested a substantial amount. The five companies are equally weighed at all times, which means that we have to rebalance the portfolio when the stock prices move. Additionally, we exclude transactions costs. We only change the portfolio once a week on Friday at 12 o'clock. We use the last traded price, as long as it does not fall outside the interval between the bid and ask prices when the portfolio is frozen. If the bid price is higher than the last traded price, we use the bid price, and vice versa for the ask price. In 2001 we introduced stop-loss, which means that if the price of a stock on any given day has a closing price that is 30% or more below the entry price in the portfolio, it is automatically removed with a 30% loss and replaced with an interest-free cash amount until the next update.

Q2: What do you look at when considering what stocks to include in the portfolio?

We have some criteria but can deviate from them if we find it reasonable. One criterion is that the purchase has to be of a certain size. We currently have a requirement of around 1 million NOK. Additionally, the transaction has to be a pure acquisition of shares, not a part of an incentive scheme. Another aspect is that the transaction has to lead to a substantial increase in the total holdings of the stock. For example, if John Fredriksen buys Frontline shares for 10 million NOK, we disregard that, as this would be a minor increase. At last, we only consider stocks listed on Oslo Stock Exchange.

Q3: Are there certain types of trades you follow more than others or is the portfolio rule-based?

We try to be as mechanical as possible and limit human involvement. We have certain criteria and follow them, though we make some deviations occasionally. Usually, we remove the stock that has been in the portfolio the longest when a new insider purchase appears. But we make exceptions if one stock has performed very well recently and another has done poorly. The same applies to insider purchases within the same sector. The insider portfolio always consists of only five stocks and must not become too concentrated in one industry. If there is an insider purchase in a salmon company and the insider portfolio already owns a salmon stock, we

remove the other salmon company to avoid too much exposure to a single type of stock. We deviate a bit in cases like this, but we try to stay as mechanical as possible.

Q4: How has the returns of the portfolio been since its start compared to index?

I do not have the cumulative return in my head, but we have data on the return for each year individually. I think we have beaten the index in 26 out of 30 years. I can send you this information by email later.

Q5: What do you think is the reason for this outperformance?

Well, markets run on asymmetric information: insiders know more than everyone else, and they legally exploit that edge. Insiders have much better knowledge about day-to-day operations and how the pipeline looks, than analysts and other people outside the company. Therefore, these trades can often give valuable information about how the people on the inside value the current stock price.

Q6: After managing the portfolio for many years, have you noticed any specific primary-insider transactions that is followed by greater returns?

No, I don't pay attention to that as this is a minor part of my work as a journalist. However, I have read and talked to people with more experience within this topic and they have mentioned that, for example, transactions made by CEO or CFO are followed by higher returns in the stock compared to what transactions made by board members are. This might be because they are more hands-on in the operation and know more about the outlook than the board.

Q7: The "Innsideporteføljen" has outperformed the index for 26 out of 30 years, what do you think about the regulations for primary insiders?

As said, there is asymmetrical information out there, where those inside the company know more than the rest of the market, and they make use of that — legally, of course. I haven't really got any specific thought on the regulations other than that.

Companies included in Analysis

Under 1B SEK (32 companies)

Acrinova	Actic Group	B3 Consulting Group	BE Group
Boule Diagnostics	Consilium	DORO	EWork Group
Eniro Group	Eolus Vind	HAKI Safety	Infrea
Inission	K2A	Lammhults Design Group	Micro Systemation
Moment Group	NCC	Nelly Group	Net Insight
Netel Holding	Nilörngruppen	PION	Pierce Group
ProfilGruppen	RAIL	Seafire	Senzime
Stockwik Förvaltning	Studsvik	Wall to Wall Group	Wise Group

1-5B SEK (85 companies)

AAK	ASCELIA PHARMA	AcadeMedia	Ambea	Annehem Fastigheter
Arise	BONESUPPORT HOLDING	Balco Group	Bergman & Beving	Betsson
Bulten	Byggmax Group	C-RAD	CTEK	Catella
Catena Media P.L.C	Cinclus Pharma Holding	Clas Ohlson	CoinShares International	Duni
Eastnine	Egtis Therapeutics	Elanders	Eltel	Enad Global 7
Ependion	FM Mattsson Mora Group AB	Fasadgruppen Group	Fastator	Fastighets AB Trianon
Fastighetsbolaget Emilshus	Ferronordic Machines	FormPipe Software	Gaming Innovation Group ..	Garo
Genova Property Group AB..	Green Landscaping Group	HEXPOL	Hanza Holding	Hoist Finance
Humana	Infant Bacterial Therape..	International Petroleum ..	K2A Knaust & Andersson F..	KlaraBo Sverige
Knowit	MMGR	Mangold	MedCap	Medivir
Mekonomen	Mildef Group	Momentum Group	NOTE	NP3 Fastigheter
Neobo Fastigheter	Nordisk Bergteknik	Orexo	Orrön Energy	Ovzon
Prevas	Pricer	Proact IT Group	Profoto Holding	RVRC Holding
RaySearch Laboratories	Rejlers	Rottneros	Scandi Standard	Sedana Medical
Sivers IMA Holding	Sleep Cycle	Starbreeze	Stenhus Fastigheter i No..	Svedbergs i Dalstorp
TF Bank	Tobii	Tobii Dynavox	Transtema Group	VBG GROUP
VNV Global	Vicore Pharma Holding	WBGRB	Xspray Pharma	Öresund, Investment

5-10B SEK (41 companies)

AQ Group	Alimak Group	Attendo	BTS Group	Besqab
Bilia	Boozt	Bravida Holding	Bufab	Cint Group
Cloetta	Coor Service Management ..	Episurf Medical	Fagerhult Group	Heba Fastighets
IRLAB Therapeutics	ITAB Shop Concept	Inwido	John Mattson Fastighetsf..	Karnov Group
Latour	Logistea	Lundin Mining Corporation	New Wave Group	Nivika Fastigheter
Nolato	Norion Bank	Platzer Fastigheter Hold..	Rusta	SkiStar
Sveafastigheter	Swedish Logistic Property	SynAct	Synsam	Systemair
Trianon	Truecaller	Vestum	Viaplay Group	Volati
Yubico				

10B+ SEK (96 companies)

AFRY	ASSA ABLOY	AddLife	Addtech	Alfa Laval
Alleima	Alligator Bioscience	Arjo	Atlas Copco	Atrium Ljungberg
Avanza Bank Holding	BICO Group	Beijer Ref	BillerudKorsnäs	Boliden
Bonava	Bure Equity	CATENA	Camurus	Castellum
Cibus Nordic Real Estate	Corem Property Group	Dometic Group	Dustin Group	Electrolux
Elekta	Embracer Group	Engcon	Essity	Evolution
Fabege	Fastighets AB Balder	Fastpartner	Fingerprint Cards	Getinge
Hansa Biopharma	Hemnet Group	Hennes & Mauritz AB, H & M	Hexagon	Hexatronic Group
Holmen	Hufvudstaden	Humble Group	Immunovia	Industrivärden
Indutrade	Intrum	K-Fast Holding	Kinnevik	Lagercrantz Group
Lifco	Lindab International	Lundbergföretagen AB, L E	Medicover	Mips
Mycronic	NCAB Group	NIBE Industrier	Nordea Bank	Nordnet
Nyfosa	Oncopeptides	PCELL	Pandox	Peab
Q-Linea	Ratos	SBB	SCA	SKF
SSAB	SWECO	Sagax	Sandvik	Scandic Hotels Group
Sdiptech	Securitas	Sinch	Skandinaviska Enskilda B..	Skanska
Stillfront Group	Storskogen Group	Svenska Handelsbanken	Swedbank	Swedish Orphan Biovitrum
Thule Group	Trelleborg	Troax Group	VOLCAR	Vimian Group
Vitec Software Group	Vitrolife	Volvo	Wallenstam	Wihlborgs Fastigheter
Xbrane Biopharma				

Note: Market capitalization based on median value during sample period. Companies listed alphabetically.

Python Code

Analysis of Swedish Primary Insider transactions and using the strategy.

```python
"""
Empirical analysis of Swedish insider acquisition transactions.

The script:
1. Reads insider transaction data from Excel file.
2. Applies a set of strict sample filters (transaction type, instrument, currency, etc.).
3. Cleans and validates the data (dates, prices, volumes, transaction values).
4. Links each insider transaction to a Yahoo Finance ticker.
5. Simulates a trading strategy that enters one day after the publication date
   and holds the stock for a fixed horizon (HOLDING_PERIOD_DAYS).
6. Computes buy-and-hold stock returns, insider returns (adjusted for stock splits),
   and benchmark returns using the OMX Stockholm 30 Index.
7. Estimates the market capitalization at entry using implied shares outstanding
   where available.
8. Exports the final dataset to Excel.

"""

import os
import time
from datetime import timedelta, datetime
import pandas as pd
import yfinance as yf
from openpyxl import load_workbook
from openpyxl.styles import PatternFill


# Excel file containing raw insider transaction data
EXCEL_FILE = r"C:\Users\Documents\FIN11\Transactions Swedish Primary Insiders 2016-2025.xlsm"

# Investment horizon in calendar days after our entry date
HOLDING_PERIOD_DAYS = 90

# Delay between Yahoo Finance requests (in seconds) to avoid rate limiting
DELAY_BETWEEN_REQUESTS = 0.5

# Minimum transaction value (in SEK) used to restrict the sample
MIN_TRANSACTION_VALUE_SEK = 500000 # 500K SEK minimum

# Load full dataset from the first sheet of the Excel file
df = pd.read_excel(EXCEL_FILE, sheet_name=0)
print(f"Loaded {len(df)} rows from Excel")
print(f"Columns found: {list(df.columns)}\n")

# Filter Verification
print("Applying STRICT filters...")
print(f"Starting with: {len(df)} rows\n")

# Store original categorical variables before filtering.
# These verify that the filters worked as intended afterwards.
df['Original_Nature'] = df['Nature of transaction']        # Column L
df['Original_Instrument_Type'] = df['Intrument type']      # Column M
df['Original_Currency'] = df['Currency']                   # Column T
df['Original_Trading_Venue'] = df['Trading venue']         # Column U
df['Original_Status'] = df['Status']                       # Column V
df['Original_Unit'] = df['Unit']                           # Column R

# Filter 1: Keep only acquisition transactions.
# Rationale: We want purchases (buy trades) rather than disposals or other transaction types.
print("Filter 1: Nature of transaction (Column L) = 'Acquisition'")
before = len(df)
df = df[df['Nature of transaction'] == 'Acquisition']
print(f"  Removed {before - len(df)} rows (kept {len(df)})") # Show results of filtering

# Filter 2: Restrict to ordinary shares.
# Rationale: Exclude derivatives, options, and other instruments.
print("\nFilter 2: Intrument type (Column M) = 'Share'")
before = len(df)
df = df[df['Intrument type'] == 'Share']
print(f"  Removed {before - len(df)} rows (kept {len(df)})") # Show results of filtering

```

```
    print(f"  Removed {before - len(df)} rows (kept {len(df)})") # Show results of filtering

    # Filter 3: Restrict to SEK-denominated trades.
    # Rationale: Avoid FX conversion issues by working in a single currency.
    print("\nFilter 3: Currency (Column T) = 'SEK'")
    before = len(df)
    df = df[df['Currency'] == 'SEK']
    print(f"  Removed {before - len(df)} rows (kept {len(df)})") # Show results of filtering

    # Filter 4: Keep only "Current" status.
    # Rationale: Exclude cancelled or corrected reports to analyze valid transactions.
    print("\nFilter 4: Status (Column V) = 'Current'")
    before = len(df)
    print(f"  Status values in data: {df['Status'].value_counts().to_dict()}")
    df = df[df['Status'] == 'Current']
    print(f"  Removed {before - len(df)} rows (kept {len(df)})") # Show results of filtering

    # Filter 5: Exclude transactions reported as "Outside a trading venue".
    # Focus on trades executed in organized trading venues where prices are avaible to the outside investor.
    print("\nFilter 5: Trading venue (Column U) NOT 'Outside a trading venue'")
    before = len(df)
    df = df[df['Trading venue'] != 'Outside a trading venue']
    print(f"  Removed {before - len(df)} rows (kept {len(df)})") # Show results of filtering

    # Filter 6: Keep only transactions reported in units of quantity, not amount.
    # Remove the transactions reported as amount. Analysis dependent on quantity and price.
    print("\nFilter 6: Unit (Column R) = 'Quantity'")
    before = len(df)
    print(f"  Unit values in data: {df['Unit'].value_counts().to_dict()}")
    df = df[df['Unit'] == 'Quantity']
    print(f"  Removed {before - len(df)} rows (kept {len(df)})")


    # Data cleaning

    # Convert date and numeric fields to the appropriate data types
    df['Transaction date'] = pd.to_datetime(df['Transaction date'], errors='coerce')
    df['Publication date'] = pd.to_datetime(df['Publication date'], errors='coerce')
    df['Price'] = pd.to_numeric(df['Price'], errors='coerce')
    df['Volume'] = pd.to_numeric(df['Volume'], errors='coerce')

    # Drop observations with missing or non-positive key variables
    before = len(df)
    df = df.dropna(subset=['Publication date', 'Price', 'ISIN', 'Volume'])
    df = df[df['Price'] > 0]
    df = df[df['Volume'] > 0]
    print(f"  Removed {before - len(df)} rows with invalid data (kept {len(df)})") # Show results of filtering

    # Filter 7: Impose a minimum transaction value threshold
    # Rationale: Focus on economically meaningful insider trades.
    print(f"\nFilter 7: Transaction value >= {MIN_TRANSACTION_VALUE_SEK:,.0f} SEK")
    df['Transaction_Value_SEK'] = df['Volume'] * df['Price']
    before = len(df)
    df = df[df['Transaction_Value_SEK'] >= MIN_TRANSACTION_VALUE_SEK]
    print(f"  Removed {before - len(df)} rows (kept {len(df)})")

    df = df.reset_index(drop=True)

    print("\n" + "=" * 60)
    print(f"FINAL: {len(df)} rows ready for analysis") # Final results before analysis
    print("=" * 60 + "\n")

    # Show the esimated time from Yahoo Finance delay
    estimated_time = len(df) * DELAY_BETWEEN_REQUESTS / 60
    print(f"Estimated time with {DELAY_BETWEEN_REQUESTS}s delay: ~{estimated_time:.1f} minutes\n")

    # Market Cap calculation

    def get_market_cap_at_entry(stock, our_entry_price):

        try:
            info = stock.info

            # Priority 1: implied shares (preferred for firms with multiple share classes)
            implied_shares = info.get('impliedSharesOutstanding', None)
            if implied_shares and implied_shares > 0:
                market_cap_sek = (our_entry_price * implied_shares) / 1_000_000
                return round(market_cap_sek, 2), 'IMPLIED'

            # Priority 2: regular shares outstanding
            shares_outstanding = info.get('sharesOutstanding', None)
            if shares_outstanding and shares_outstanding > 0:
                market_cap_sek = (our_entry_price * shares_outstanding) / 1_000_000
                return round(market_cap_sek, 2), 'REGULAR'

            # If neither measure is available, we cannot compute market cap
            return None, 'FAILED'
        except Exception:
            return None, 'FAILED'


    # Main Simulation. Processing trades.

    results = []
```

```
    # Counters to document the success rate and potential sources of attrition
    stats = {
        'success': 0,
        'no_ticker': 0,
        'no_exit': 0,
        'no_index': 0,
        'skipped_future': 0,
        'error': 0,
        'mcap_implied': 0,
        'mcap_regular': 0,
        'mcap_failed': 0
    }

    for i, row in df.iterrows():
        try:
            # Key dates for each transaction
            transaction_date = row['Transaction date']
            publication_date = row['Publication date']

            # Our entry date. First trading day after the publication date
            our_entry_date = publication_date + timedelta(days=1)

            # Holding period end date (calendar days after our entry)
            exit_date = our_entry_date + timedelta(days=HOLDING_PERIOD_DAYS)

            # Security identifiers and transaction characteristics
            isin = row['ISIN']
            insider_price = row['Price']
            volume = row['Volume']
            transaction_value = row['Transaction_Value_SEK']

            # Skip observations if we do not yet have sufficient return history up to exit_date
            today = datetime.now()
            if exit_date > today:
                stats['skipped_future'] += 1
                continue


            # Delay between Yahoo Finance requests (rate limit avoidance)
            if i > 0 and DELAY_BETWEEN_REQUESTS > 0:
                time.sleep(DELAY_BETWEEN_REQUESTS)

            # FInd ticker using Yahoo Finance
            # Strategy: try ISIN, then ISIN.ST, then ISIN.SG
            # non-empty recent price history.

            ticker = None

            # Attempt 1: direct ISIN number
            try:
                test = yf.Ticker(isin)
                h = test.history(period='5d')
                if not h.empty:
                    ticker = isin
            except Exception:
                pass

            # Attempt 2: ISIN.ST
            if not ticker:
                try:
                    test_ticker = f"{isin}.ST"
                    test = yf.Ticker(test_ticker)
                    h = test.history(period='5d')
                    if not h.empty:
                        ticker = test_ticker
                except Exception:
                    pass

            # Attempt 3: ISIN.SG
            if not ticker:
                try:
                    test_ticker = f"{isin}.SG"
                    test = yf.Ticker(test_ticker)
                    h = test.history(period='5d')
                    if not h.empty:
                        ticker = test_ticker
                except Exception:
                    pass

            # If no ticker is found, we cannot proceed with this observation
            if not ticker:
                stats['no_ticker'] += 1
                continue

            # Entry Price
            # Define our entry price as the mid-point between daily high and low
            # On the first available trading day after the publish date (within a 7-day window)
            stock = yf.Ticker(ticker)
            our_entry_h = stock.history(start=our_entry_date, end=our_entry_date + timedelta(days=7))
            if our_entry_h.empty:
                stats['no_exit'] += 1
                continue

            our_entry_price = (our_entry_h['High'].iloc[0] + our_entry_h['Low'].iloc[0]) / 2
```

```
263
264         # Market Cap at entry date
265         market_cap_sek_millions, mcap_method = get_market_cap_at_entry(stock, our_entry_price)
266
267         # Update market cap statistics
268         if mcap_method == 'IMPLIED':
269             stats['mcap_implied'] += 1
270         elif mcap_method == 'REGULAR':
271             stats['mcap_regular'] += 1
272         else:
273             stats['mcap_failed'] += 1
274
275
276         # Insider price and split adjustment
277         # Use the midprice (High+Low)/2 on the transaction date to standardize comparison.
278         insider_price_adjusted = insider_price
279         split_detected = False
280         split_ratio = 1.0
281
282         try:
283             insider_h = stock.history(start=transaction_date, end=transaction_date + timedelta(days=7))
284             if not insider_h.empty:
285                 yahoo_high = insider_h['High'].iloc[0]
286                 yahoo_low = insider_h['Low'].iloc[0]
287                 yahoo_mid = (yahoo_high + yahoo_low) / 2
288
289                 # Check if insider price is within the day's trading range
290                 if yahoo_low <= insider_price <= yahoo_high:
291                     # Use midprice instead of the reported insider price
292                     insider_price_adjusted = yahoo_mid
293                 else:
294                     # Flag as possible split, but still use midprice for consistency
295                     split_detected = True
296                     insider_price_adjusted = yahoo_mid
297                     split_ratio = insider_price / yahoo_mid if yahoo_mid > 0 else 1.0
298
299         except Exception:
300             # If prices unavailable, keep original insider price
301             pass
302
303
304         # Exit Price
305         # Use the closing price on the first trading day at exit_date (within a 7-day window).
306         exit_h = stock.history(start=exit_date, end=exit_date + timedelta(days=7))
307         if exit_h.empty:
308             stats['no_exit'] += 1
309             continue
310         exit_price = exit_h['Close'].iloc[0]
311
312         # Benchmark Index (OMX Stockholm 30)
313         omx = yf.Ticker("^OMX")
314         idx_entry = omx.history(start=our_entry_date, end=our_entry_date + timedelta(days=7))
315         idx_exit = omx.history(start=exit_date, end=exit_date + timedelta(days=7))
316         if idx_entry.empty or idx_exit.empty:
317             stats['no_index'] += 1
318             continue
319
320         # Return caluclation (not dividend adjusted)
321         stock_ret = ((exit_price / our_entry_price) - 1) * 100
322         insider_ret = ((exit_price / insider_price_adjusted) - 1) * 100
323         index_ret = ((idx_exit['Close'].iloc[0] / idx_entry['Close'].iloc[0]) - 1) * 100
324         outperf = stock_ret - index_ret
325
326         # Progress update to follow progress
327         if (i + 1) % 100 == 0:
328             print(f"Processed {i + 1}/{len(df)} trades... ({stats['success']} successfully constructed)")
329
330         # Record the final observation
331         # Store verify variables to check that the filters have been applied correctly.
332         results.append({
333             'Person': row.get('Person discharging managerial responsibilities', ''),
334             'Position': row.get('Position', ''),
335             'Company': row.get('Instrument name', ''),
336             'ISIN': isin,
337             'Ticker': ticker,
338             'Transaction_Date': transaction_date,
339             'Publication_Date': publication_date,
340             'Our_Entry_Date': our_entry_date,
341             'Exit_Date': exit_date,
342             'Volume': volume,
343             'Insider_Price_Reported': insider_price,
344             'Transaction_Value_SEK': round(transaction_value, 2),
345             'Insider_Price_Adjusted': round(insider_price_adjusted, 2),
346             'Market_Cap_SEK_Millions': market_cap_sek_millions,
347             'Market_Cap_Method': mcap_method,
348             'Split_Detected': 'YES' if split_detected else 'NO',
349             'Split_Ratio': round(split_ratio, 2),
350             'Our_Entry_Price': round(our_entry_price, 2),
351             'Exit_Price': round(exit_price, 2),
352             'Our_Stock_Return_%': round(stock_ret, 2),
353             'Insider_Return_%': round(insider_ret, 2),
354             'Index_Return_%': round(index_ret, 2),
355             'Our_Outperformance_%': round(outperf, 2),
356             'VERIFY_Nature': row['Original_Nature'],
357             'VERIFY_Instrument_Type': row['Original_Instrument_Type'],
358             'VERIFY_Currency': row['Original_Currency'],
359             'VERIFY_Status': row['Original_Status'],
360             'VERIFY_Trading_Venue': row['Original_Trading_Venue'],
361             'VERIFY_Unit': row['Original_Unit']
362         })
363         stats['success'] += 1
364
```

```
        except Exception as e:
            stats['error'] += 1

    # Save to Excel and print results
    print("\n" + "=" * 60)
    print("RESULTS SUMMARY")
    print("=" * 60)
    print(f"Successfully constructed trades: {stats['success']}")
    print(f"Skipped due to insufficient future data: {stats['skipped_future']}")
    print(f"Failed - no ticker mapping: {stats['no_ticker']}")
    print(f"Failed - no exit price: {stats['no_exit']}")
    print(f"Failed - no index data: {stats['no_index']}")
    print(f"Failed - other errors: {stats['error']}")

    if results:
        df_results = pd.DataFrame(results)
        output_file = os.path.join(
            os.path.dirname(EXCEL_FILE),
            'Results_Primary_Insider_500k_365.xlsx'
        )

        # Verification of filters
        print("\n" + "=" * 60)
        print("FILTER VERIFICATION CHECK")
        print("=" * 60)

        violations = 0

        if (df_results['VERIFY_Nature'] != 'Acquisition').any():
            bad = (df_results['VERIFY_Nature'] != 'Acquisition').sum()
            print(f"Warning: {bad} results are not 'Acquisition'.")
            violations += bad

        if (df_results['VERIFY_Instrument_Type'] != 'Share').any():
            bad = (df_results['VERIFY_Instrument_Type'] != 'Share').sum()
            print(f"Warning: {bad} results are not 'Share'.")
            violations += bad

        if (df_results['VERIFY_Currency'] != 'SEK').any():
            bad = (df_results['VERIFY_Currency'] != 'SEK').sum()
            print(f"Warning: {bad} results are not 'SEK'.")
            violations += bad

        if (df_results['VERIFY_Status'] != 'Current').any():
            bad = (df_results['VERIFY_Status'] != 'Current').sum()
            print(f"Warning: {bad} results are not 'Current'.")
            violations += bad

        if (df_results['VERIFY_Trading_Venue'] == 'Outside a trading venue').any():
            bad = (df_results['VERIFY_Trading_Venue'] == 'Outside a trading venue').sum()
            print(f"Warning: {bad} results are 'Outside a trading venue'.")
            violations += bad

        if (df_results['VERIFY_Unit'] != 'Quantity').any():
            bad = (df_results['VERIFY_Unit'] != 'Quantity').sum()
            print(f"Warning: {bad} results are not 'Quantity' (likely 'Amount').")
            violations += bad

        if violations == 0:
            print("All results pass filter verification.")
        else:
            print(f"\nTotal verification violations: {violations}")
            print("Inspect the VERIFY_* columns in the output file for details.")

        # Save results to Excel
        df_results.to_excel(output_file, index=False)

        # Color-code to manually fix failed market cap calculations
        print("\nApplying color coding to Market Cap column...")
        wb = load_workbook(output_file)
        ws = wb.active

        # Identify the relevant columns by header name
        mcap_col_idx = None
        method_col_idx = None
        for idx, cell in enumerate(ws[1], 1):  # Header row
            if cell.value == 'Market_Cap_SEK_Millions':
                mcap_col_idx = idx
            if cell.value == 'Market_Cap_Method':
                method_col_idx = idx

        if mcap_col_idx and method_col_idx:
            # Light green for IMPLIED, yellow for REGULAR, light red for FAILED
            green_fill = PatternFill(start_color='90EE90', end_color='90EE90', fill_type='solid')
            yellow_fill = PatternFill(start_color='FFFF00', end_color='FFFF00', fill_type='solid')
            red_fill = PatternFill(start_color='FFB6C1', end_color='FFB6C1', fill_type='solid')

            for row_idx in range(2, ws.max_row + 1):  # Skip header
                method_cell = ws.cell(row=row_idx, column=method_col_idx)
                mcap_cell = ws.cell(row=row_idx, column=mcap_col_idx)

                if method_cell.value == 'IMPLIED':
                    mcap_cell.fill = green_fill
                elif method_cell.value == 'REGULAR':
                    mcap_cell.fill = yellow_fill
                elif method_cell.value == 'FAILED':
                    mcap_cell.fill = red_fill

        wb.save(output_file)
```

Buying after the dip calculation.

```python
import pandas as pd
import yfinance as yf
from datetime import timedelta, datetime
import os
import time

EXCEL_FILE = r"C:\Users\Documents\FIN11\Innsideporteføljen Return Data 1996-2025.xlsx"
HOLDING_PERIODS = [90, 180, 365]
DELAY_BETWEEN_REQUESTS = 0.5
MIN_TRANSACTION_VALUE_SEK = 500000
LOOKBACK_DAYS = 30
LOOKBACK_THRESHOLD = -20

df = pd.read_excel(EXCEL_FILE, sheet_name=0)
print(f"Loaded: {len(df)} rows")

df = df[df['Nature of transaction'] == 'Acquisition']
print(f"After Acquisition filter: {len(df)} rows")

df = df[df['Intrument type'] == 'Share']
print(f"After Share filter: {len(df)} rows")

df = df[df['Currency'] == 'SEK']
print(f"After SEK filter: {len(df)} rows")

df = df[df['Status'] == 'Current']
print(f"After Current status filter: {len(df)} rows")

df = df[df['Trading venue'] != 'Outside a trading venue']
print(f"After Trading venue filter: {len(df)} rows")

df = df[df['Unit'] == 'Quantity']
print(f"After Quantity filter: {len(df)} rows")

df['Transaction date'] = pd.to_datetime(df['Transaction date'], errors='coerce')
df['Publication date'] = pd.to_datetime(df['Publication date'], errors='coerce')
df['Price'] = pd.to_numeric(df['Price'], errors='coerce')
df['Volume'] = pd.to_numeric(df['Volume'], errors='coerce')

df = df.dropna(subset=['Publication date', 'Price', 'ISIN', 'Volume'])
df = df[df['Price'] > 0]
df = df[df['Volume'] > 0]
print(f"After cleaning NaN/invalid: {len(df)} rows")

df['Transaction_Value_SEK'] = df['Volume'] * df['Price']
df = df[df['Transaction_Value_SEK'] >= MIN_TRANSACTION_VALUE_SEK]
print(f"After min transaction value (500k SEK): {len(df)} rows")

df = df.reset_index(drop=True)
print(f"\nReady for analysis: {len(df)} trades\n")

def check_lookback_30d(ticker, entry_date, threshold_pct=-20):
    """Check if stock dropped 20%+ in 30 days before entry"""
    try:
        download_start = entry_date - timedelta(days=90)
        download_end = entry_date + timedelta(days=5)

        stock = yf.Ticker(ticker)
        hist = stock.history(start=download_start, end=download_end)
        hist.index = hist.index.tz_localize(None)

        entry_prices = hist[hist.index >= entry_date]
        if len(entry_prices) == 0:
            return False, None
        entry_price = entry_prices['Close'].iloc[0]

        target_lookback = entry_date - timedelta(days=30)
        lookback_prices = hist[hist.index <= target_lookback]
        if len(lookback_prices) == 0:
            return False, None
        lookback_price = lookback_prices['Close'].iloc[-1]

        return_pct = ((entry_price - lookback_price) / lookback_price) * 100
```

```
            return_pct = ((entry_price - lookback_price) / lookback_price) * 100
            is_down = return_pct <= threshold_pct

            return is_down, round(return_pct, 2)
        except:
            return False, None

    def get_ticker(isin):
        """Try ticker formats to find working one"""
        for ticker_format in [isin, isin + '.ST', isin + '.SG']:
            try:
                test_stock = yf.Ticker(ticker_format)
                if len(test_stock.history(period="5d")) > 0:
                    return ticker_format
            except:
                pass
        return None

    def get_market_cap_at_entry(stock, our_entry_price):
        """Calculate market cap at entry"""
        try:
            info = stock.info
            shares = info.get('sharesOutstanding', None)
            if shares and shares > 0:
                return round((shares * our_entry_price) / 1_000_000, 2), 'IMPLIED'
            mcap = info.get('marketCap', None)
            if mcap and mcap > 0:
                return round(mcap / 1_000_000, 2), 'REGULAR'
        except:
            pass
        return None, 'FAILED'

    all_results = {}

    for HOLDING_PERIOD_DAYS in HOLDING_PERIODS:
        results = []

        for i, row in df.iterrows():
            try:
                isin = row['ISIN']
                publication_date = row['Publication date']
                transaction_date = row['Transaction date']
                insider_price = row['Price']
                volume = row['Volume']
                transaction_value = row['Transaction_Value_SEK']

                our_entry_date = publication_date + timedelta(days=1)
                exit_date = our_entry_date + timedelta(days=HOLDING_PERIOD_DAYS)

                if exit_date > datetime.now():
                    continue

                ticker = get_ticker(isin)
                if not ticker:
                    continue

                is_dip, lookback_return = check_lookback_30d(ticker, our_entry_date, LOOKBACK_THRESHOLD)
                if not is_dip:
                    continue

                stock = yf.Ticker(ticker)

                entry_data = stock.history(start=our_entry_date, end=our_entry_date + timedelta(days=5))
                if len(entry_data) == 0:
                    continue

                first_day = entry_data.iloc[0]
                our_entry_price = (first_day['High'] + first_day['Low']) / 2
                market_cap, mcap_method = get_market_cap_at_entry(stock, our_entry_price)

```

Figure 1: Innsideporteføljen vs OBX 1996-2025.

```
            split_detected = False
            split_ratio = 1.0
            insider_price_adjusted = insider_price
            if abs((insider_price - our_entry_price) / our_entry_price) * 100 > 20:
                split_detected = True
                split_ratio = insider_price / our_entry_price
                insider_price_adjusted = our_entry_price

            exit_data = stock.history(start=exit_date, end=exit_date + timedelta(days=5))
            if len(exit_data) == 0:
                continue
            exit_price = exit_data['Close'].iloc[0]

            try:
                index = yf.Ticker("^OMX")
                idx_entry = index.history(start=our_entry_date, end=our_entry_date + timedelta(days=5))
                idx_exit = index.history(start=exit_date, end=exit_date + timedelta(days=5))
                if len(idx_entry) == 0 or len(idx_exit) == 0:
                    continue
            except:
                continue

            stock_ret = ((exit_price / our_entry_price) - 1) * 100
            insider_ret = ((exit_price / insider_price_adjusted) - 1) * 100
            index_ret = ((idx_exit['Close'].iloc[0] / idx_entry['Close'].iloc[0]) - 1) * 100
            outperf = stock_ret - index_ret

            results.append({
                'Person': row.get('Person discharging managerial responsibilities', ''),
                'Position': row.get('Position', ''),
                'Company': row.get('Instrument name', ''),
                'ISIN': isin,
                'Ticker': ticker,
                'Transaction_Date': transaction_date,
                'Publication_Date': publication_date,
                'Our_Entry_Date': our_entry_date,
                'Exit_Date': exit_date,
                'Volume': volume,
                'Insider_Price_Reported': insider_price,
                'Transaction_Value_SEK': round(transaction_value, 2),
                'Insider_Price_Adjusted': round(insider_price_adjusted, 2),
                'Market_Cap_SEK_Millions': market_cap,
                'Market_Cap_Method': mcap_method,
                'Split_Detected': 'YES' if split_detected else 'NO',
                'Split_Ratio': round(split_ratio, 2),
                'Our_Entry_Price': round(our_entry_price, 2),
                'Exit_Price': round(exit_price, 2),
                'Our_Stock_Return_%': round(stock_ret, 2),
                'Insider_Return_%': round(insider_ret, 2),
                'Index_Return_%': round(index_ret, 2),
                'Our_Outperformance_%': round(outperf, 2),
                'Lookback_Period_Days': 30,
                'Lookback_Return_%': lookback_return,
                'Condition_Met': '30d down -20%'})

            time.sleep(DELAY_BETWEEN_REQUESTS)
        except:
            continue

    all_results[HOLDING_PERIOD_DAYS] = results

output_file = os.path.join(os.path.dirname(EXCEL_FILE), 'Results_Lookback_Filtered.xlsx')

with pd.ExcelWriter(output_file, engine='openpyxl') as writer:
    for holding_period in HOLDING_PERIODS:
        results = all_results[holding_period]
        if results:
            df_results = pd.DataFrame(results)
            df_results = df_results.sort_values('Our_Entry_Date')
            df_results.to_excel(writer, sheet_name=f"Hold_{holding_period}d", index=False)

```

```python
# FIGURE 1: Innsideporteføljen vs OBX

import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

def create_figure1_swapped():

    # Load data from Excel
    file_path = r'C:\Users\Documents\FIN11\Innsideporteføljen Return Data 1996-2025.xlsx'
    df = pd.read_excel(file_path)

    # Extract columns
    years = df.iloc[:, 0].astype(int).tolist()        # Column A: Year
    insider_annual = df.iloc[:, 1].tolist()           # Column B: Innsideporteføljen
    obx_annual = df.iloc[:, 2].tolist()               # Column C: OBX Index

    # Calculate cumulative returns
    insider_cumulative = [100]
    obx_cumulative = [100]

    for ret in insider_annual:
        insider_cumulative.append(insider_cumulative[-1] * (1 + ret/100))
    for ret in obx_annual:
        obx_cumulative.append(obx_cumulative[-1] * (1 + ret/100))

    # Colors
    insider_color = '#0066FF'
    obx_color = '#e74c3c'

    # Create figure
    fig, (ax1, ax2) = plt.subplots(2, 1, figsize=(14, 10))

    # Annual Returns Bar Chart
    x = np.arange(len(years))

    ax1.axhline(y=0, color='black', linewidth=0.8, zorder=2)
    ax1.set_ylabel('Annual Return (%)', fontweight='bold', fontsize=11)
    ax1.set_title('Annual Returns Comparison', fontweight='bold', fontsize=11, loc='left')
    ax1.set_xticks(x)
    ax1.set_xticklabels(years, rotation=45, ha='right', fontsize=9)
    ax1.legend(loc='upper left', fontsize=10)
    ax1.grid(axis='y', alpha=0.3, zorder=1)
    ax1.set_xlim(-0.5, len(years) - 0.5)

    # Remove top and right spines
    ax1.spines['top'].set_visible(False)
    ax1.spines['right'].set_visible(False)

    # Cumulative Performance
    years_cumulative = [years[0] - 1] + years

    ax2.semilogy(years_cumulative, insider_cumulative, color=insider_color, linewidth=2.5,
                 label='Innsideporteføljen', marker='o', markersize=5,
                 markerfacecolor='white', markeredgecolor=insider_color, markeredgewidth=1.5)
    ax2.semilogy(years_cumulative, obx_cumulative, color=obx_color, linewidth=2.5,
                 label='OBX Index', marker='o', markersize=5,
                 markerfacecolor='white', markeredgecolor=obx_color, markeredgewidth=1.5)

    ax2.set_ylabel('Cumulative Value (log scale)', fontweight='bold', fontsize=11)
    ax2.set_xlabel('Year', fontweight='bold', fontsize=11)
    ax2.set_title(f'Cumulative Performance ({years[0]}-{years[-1]})', fontweight='bold', fontsize=11, loc='left')
    ax2.legend(loc='upper left', fontsize=10)
    ax2.grid(True, alpha=0.3, which='both')
    ax2.set_xlim(years[0] - 1, years[-1])

    # Remove top and right spines
    ax2.spines['top'].set_visible(False)
    ax2.spines['right'].set_visible(False)

    # Stats box
    insider_total = insider_cumulative[-1] / 100
    obx_total = obx_cumulative[-1] / 100
    insider_avg = np.mean(insider_annual)
    obx_avg = np.mean(obx_annual)
    outperform_years = sum(1 for i, o in zip(insider_annual, obx_annual) if i > o)
    n_years = len(years)

    stats_text = (f"Innsideporteføljen: {insider_total:.0f}x ({insider_avg:.1f}% avg)\n"
                  f"OBX Index: {obx_total:.0f}x ({obx_avg:.1f}% avg)\n"
                  f"Outperformed {outperform_years}/{n_years} years")

    props = dict(boxstyle='round,pad=0.5', facecolor='white',
                 edgecolor=insider_color, linewidth=2)
    ax2.text(0.98, 0.05, stats_text, transform=ax2.transAxes, fontsize=10,
             verticalalignment='bottom', horizontalalignment='right',
             bbox=props, fontweight='bold')

    plt.tight_layout()
    plt.savefig('Figure1_final_swapped.png', dpi=300,
                bbox_inches='tight', facecolor='white')
    plt.close()
```

Figure 2, 10 and 15-19. The code is representative for all figures, creating tables with different data.

```python
import pandas as pd
import matplotlib.pyplot as plt

# File paths
df_90 = pd.read_excel(r"C:\Users\Documents\FIN11\Results_Primary_Insider_500k_90.xlsx")
df_180 = pd.read_excel(r"C:\Users\Documents\FIN11\Results_Primary_Insider_500k_180.xlsx")
df_365 = pd.read_excel(r"C:\Users\Documents\FIN11\Results_Primary_Insider_500k_365.xlsx")

def calculate_stats(df, period_name):
    n = len(df)
    strategy_return = df['Our_Stock_Return_%'].mean()
    median_return = df['Our_Stock_Return_%'].median()
    avg_index_return = df['Index_Return_%'].mean()
    mean_alpha = df['Our_Outperformance_%'].mean()
    win_rate_positive = (df['Our_Stock_Return_%'] > 0).sum() / n * 100
    win_rate_vs_index = (df['Our_Outperformance_%'] > 0).sum() / n * 100
    std_dev = df['Our_Stock_Return_%'].std()

    return {
        'Holding Period': period_name,
        'Number of Transactions': n,
        'Strategy Return': strategy_return,
        'Median Return': median_return,
        'Avg. Index Return': avg_index_return,
        'Mean Alpha': mean_alpha,
        'Win Rate (Positive)': win_rate_positive,
        'Win Rate (vs Index)': win_rate_vs_index,
        'Std. Dev.': std_dev}

# Calculate for all holding periods
stats_90 = calculate_stats(df_90, '90 days')
stats_180 = calculate_stats(df_180, '180 days')
stats_365 = calculate_stats(df_365, '365 days')

all_stats = [stats_90, stats_180, stats_365]

fig, ax = plt.subplots(figsize=(16, 2.2))
ax.axis('off')

# Headers
columns = ['Holding\nPeriod', 'Number of\nTransactions', 'Strategy\nReturn',
           'Median\nReturn', 'Avg. Index\nReturn', 'Mean\nAlpha',
           'Win Rate\n(Positive)', 'Win Rate\n(vs Index)', 'Std. Dev.\nStrategy Return']

table_data = []
for stats in all_stats:
    row = [
        stats['Holding Period'],
        f"{stats['Number of Transactions']:,}",
        f"{stats['Strategy Return']:.2f}%",
        f"{stats['Median Return']:.2f}%",
        f"{stats['Avg. Index Return']:.2f}%",
        f"{stats['Mean Alpha']:.2f}%",
        f"{stats['Win Rate (Positive)']:.1f}%",
        f"{stats['Win Rate (vs Index)']:.1f}%",
        f"{stats['Std. Dev.']:.2f}%"]
    table_data.append(row)

# Create table
table = ax.table(
    cellText=table_data,
    colLabels=columns,
    cellLoc='center',
    loc='center',
    bbox=[0, 0, 1, 1])

table.auto_set_font_size(False)
table.set_fontsize(11)
table.scale(1, 1.8)

# Colors
header_color = '#0066FF'
row_light = '#E3F2FD'
row_white = '#FFFFFF'
alpha_bg = '#C8E6C9'
alpha_positive = '#2E7D32'
alpha_negative = '#C62828'

for i in range(len(columns)):
    cell = table[(0, i)]
    cell.set_facecolor(header_color)
    cell.set_text_props(weight='bold', color='white', size=10)
    cell.set_edgecolor('black')
    cell.set_linewidth(1)
```

```
88
89  for row_idx in range(1, 4):
90      if row_idx % 2 == 1:
91          row_color = row_light
92      else:
93          row_color = row_white
94
95      for col_idx in range(len(columns)):
96          cell = table[(row_idx, col_idx)]
97          cell.set_edgecolor('black')
98          cell.set_linewidth(1)
99
100         if col_idx == 5:
101             cell.set_facecolor(alpha_bg)
102             # Get the alpha value to determine text color
103             alpha_val = all_stats[row_idx - 1]['Mean Alpha']
104             if alpha_val >= 0:
105                 cell.set_text_props(weight='bold', color=alpha_positive, size=11)
106             else:
107                 cell.set_text_props(weight='bold', color=alpha_negative, size=11)
108         else:
109             cell.set_facecolor(row_color)
110             cell.set_text_props(size=11)
111
112 plt.tight_layout(pad=0.1)
113 plt.savefig('summary_statistics_table.png', dpi=300, bbox_inches='tight',
114             facecolor='white', edgecolor='none', pad_inches=0.02)
115
116 plt.show()
```

Figure 3. Average return per trade vs OMXS30 annually 2017-2025.

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

# File paths
df_90 = pd.read_excel(r"C:\Users\Documents\FIN11\Results_Primary_Insider_500k_90.xlsx")
df_180 = pd.read_excel(r"C:\Users\Documents\FIN11\Results_Primary_Insider_500k_180.xlsx")
df_365 = pd.read_excel(r"C:\Users\Documents\FIN11\Results_Primary_Insider_500k_365.xlsx")

# Colors
COLOR_BLUE = '#0066FF'
COLOR_RED = '#e74c3c'

fig, axes = plt.subplots(3, 1, figsize=(18, 16))
fig.suptitle('AVERAGE RETURN PER TRADE VS OMX ANNUAL (2017-2025)',
             fontsize=22, fontweight='bold', y=0.995)

periods = [('90 days', df_90), ('180 days', df_180), ('365 days', df_365)]

for idx, (period_name, df) in enumerate(periods):
    ax = axes[idx]
    df_clean = df[df['Our_Stock_Return_%'].notna() & df['Index_Return_%'].notna()].copy()
    df_clean['Year'] = pd.to_datetime(df_clean['Transaction_Date']).dt.year

    yearly_stats = df_clean.groupby('Year').agg({
        'Our_Stock_Return_%': 'mean',
        'Index_Return_%': 'mean',
        'Year': 'count'
    }).rename(columns={'Year': 'Count'})

    x = yearly_stats.index
    width = 0.35
    x_pos = np.arange(len(x))

    bars1 = ax.bar(x_pos - width/2, yearly_stats['Our_Stock_Return_%'], width,
                   label='Insider Portfolio', color=COLOR_BLUE, alpha=0.85, edgecolor='black', linewidth=1.5)
    bars2 = ax.bar(x_pos + width/2, yearly_stats['Index_Return_%'], width,
                   label='OMX Index', color=COLOR_RED, alpha=0.85, edgecolor='black', linewidth=1.5)

    ax.axhline(y=0, color='black', linestyle='-', linewidth=2, zorder=0)
    ax.set_xlabel('Year', fontsize=14, fontweight='bold')
    ax.set_ylabel('Average Return (%)', fontsize=14, fontweight='bold')
    ax.set_title(f'Holding Period: {period_name} (n={len(df_clean)} transactions)',
                 fontsize=16, fontweight='bold', pad=15)
    ax.set_xticks(x_pos)
    ax.set_xticklabels(x, rotation=0, fontsize=12)
    ax.legend(loc='upper right', fontsize=13, frameon=True)
    ax.grid(True, alpha=0.3, axis='y')

    # Y axis limits
    y_max = max(yearly_stats['Our_Stock_Return_%'].max(), yearly_stats['Index_Return_%'].max())
    y_min = min(yearly_stats['Our_Stock_Return_%'].min(), yearly_stats['Index_Return_%'].min())
    y_range = y_max - y_min
    ax.set_ylim(y_min - y_range * 0.15, y_max + y_range * 0.20)

    # Value labels
    for bars in [bars1, bars2]:
        for bar in bars:
            height = bar.get_height()
            label_y = height + y_range * 0.02 if height > 0 else height - y_range * 0.02
            va = 'bottom' if height > 0 else 'top'
            ax.text(bar.get_x() + bar.get_width()/2., label_y, f'{height:.1f}%',
                    ha='center', va=va, fontsize=11, fontweight='bold',
                    bbox=dict(boxstyle='round,pad=0.3', facecolor='white', edgecolor='none', alpha=0.7))

    for i, (pos, count) in enumerate(zip(x_pos, yearly_stats['Count'])):
        ax.text(pos, ax.get_ylim()[1] * 0.97, f'n={count}',
                ha='center', va='top', fontsize=9, style='italic', alpha=0.6)

plt.tight_layout()
plt.savefig('annual_returns_vs_omx.png', dpi=300, bbox_inches='tight', facecolor='white')
plt.show()
```

Figure 4. Return Distribution.

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

# File paths
df_90 = pd.read_excel(r"C:\Users\Documents\FIN11\Results_Primary_Insider_500k_90.xlsx")
df_180 = pd.read_excel(r"C:\Users\Documents\FIN11\Results_Primary_Insider_500k_180.xlsx")
df_365 = pd.read_excel(r"C:\Users\Documents\FIN11\Results_Primary_Insider_500k_365.xlsx")

returns_90 = df_90['Our_Stock_Return_%'].dropna()
returns_180 = df_180['Our_Stock_Return_%'].dropna()
returns_365 = df_365['Our_Stock_Return_%'].dropna()

# Colors
color_positive = '#0066FF'
color_negative = '#e74c3c'

fig, axes = plt.subplots(1, 3, figsize=(16, 5))
datasets = [returns_90, returns_180, returns_365]
titles = ['90 days', '180 days', '365 days']

for ax, data, title in zip(axes, datasets, titles):
    data_clipped = data.clip(-100, 250)
    mean_val = data.mean()
    median_val = data.median()

    bins = np.linspace(-100, 250, 51)
    n, bin_edges, patches = ax.hist(data_clipped, bins=bins, edgecolor='white', linewidth=0.5)

    # Color bars (negative = red, positive = blue)
    for patch, left_edge in zip(patches, bin_edges[:-1]):
        if left_edge < 0:
            patch.set_facecolor(color_negative)
        else:
            patch.set_facecolor(color_positive)

    ax.axvline(mean_val, color='black', linestyle='--', linewidth=1.5,
               label=f'Mean performance: {mean_val:.1f}%')

    ax.set_xlabel('Return (%)', fontweight='bold', fontsize=10)
    ax.set_ylabel('Frequency', fontweight='bold', fontsize=10)
    ax.set_title(f'{title} (n={len(data)})', fontweight='bold', fontsize=12)
    ax.legend(loc='upper right', fontsize=9)

    # Add median as text box below legend
    ax.text(0.98, 0.82, f'Median: {median_val:.1f}%', transform=ax.transAxes,
            fontsize=9, ha='right', va='top',
            bbox=dict(boxstyle='round', facecolor='white', edgecolor='gray', alpha=0.8))

    ax.spines['top'].set_visible(False)
    ax.spines['right'].set_visible(False)

fig.suptitle('RETURN DISTRIBUTION', fontweight='bold', fontsize=14, y=1.02)
fig.text(0.5, 0.96, 'Strategy performance following insider purchases',
         ha='center', fontsize=11, fontstyle='normal')
plt.tight_layout()
plt.savefig('return_distribution.png', dpi=300, bbox_inches='tight', facecolor='white')
plt.show()
```

Figure 5 and 7. Mean Excess Return for segmented market caps / transaction sizes in different holding periods. Representative code for both figures.

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.colors import LinearSegmentedColormap

df_90 = r"C:\Users\Documents\FIN11\'Results_Primary_Insider_500k_90.xlsx"
df_180 = r"C:\Users\Documents\FIN11\'Results_Primary_Insider_500k_180.xlsx"
df_365 = r"C:\Users\Documents\FIN11\'Results_Primary_Insider_500k_365.xlsx"

df_90['Holding_Period'] = 90
df_180['Holding_Period'] = 180
df_365['Holding_Period'] = 365

def categorize_market_cap(df):
    bins = [0, 1000, 5000, 10000, float('inf')]
    labels = ['<1B', '1-5B', '5-10B', '>10B']
    df['MCap_Category'] = pd.cut(
        df['Market_Cap_SEK_Millions'],
        bins=bins,
        labels=labels)
    return df

df_90 = categorize_market_cap(df_90)
df_180 = categorize_market_cap(df_180)
df_365 = categorize_market_cap(df_365)

def calculate_cell_stats(df, holding_period):
    stats = df.groupby('MCap_Category', observed=True).agg({
        'Our_Outperformance_%': 'mean',    # Abnormal return (alpha)
        'Our_Stock_Return_%': 'mean',      # Strategy return
        'Index_Return_%': 'mean',          # Benchmark return
        'ISIN': 'count'}).rename(columns={'ISIN': 'n'}) # Sample size

    stats['Holding_Period'] = holding_period
    return stats.reset_index()

stats_90 = calculate_cell_stats(df_90, 90)
stats_180 = calculate_cell_stats(df_180, 180)
stats_365 = calculate_cell_stats(df_365, 365)

# Combine all statistics
all_stats = pd.concat([stats_90, stats_180, stats_365], ignore_index=True)

# Make data for heatmap
# Create pivot table for abnormal returns
heatmap_data = all_stats.pivot(
    index='Holding_Period',
    columns='MCap_Category',
    values='Our_Outperformance_%')

# Reorder columns and index
mcap_order = ['<1B', '1-5B', '5-10B', '>10B']
period_order = [90, 180, 365]
heatmap_data = heatmap_data.reindex(columns=mcap_order, index=period_order)

# Colors for scale
# Position calculation: (value - (-5)) / 17
colors_custom = ['#e74c3c','#f1948a','#fce4e4','#ffffff','#cce5ff','#66b3ff','#3399ff','#0066FF','#0055dd']
positions = [0.0, 0.18, 0.26, 0.29, 0.47, 0.65, 0.76, 0.88, 1.0]

# Create custom colormap
cmap_custom = LinearSegmentedColormap.from_list(
    'insider_cmap',
    list(zip(positions, colors_custom)))

fig, ax = plt.subplots(figsize=(14, 6))

im = ax.imshow(
    heatmap_data.values,
    cmap=cmap_custom,
    aspect='auto',
    vmin=-5,
    vmax=12)
```

```
    ax.set_xticks(np.arange(len(mcap_order)))
    ax.set_yticks(np.arange(len(period_order)))
    ax.set_xticklabels(mcap_order, fontsize=11, fontweight='bold')
    ax.set_yticklabels(['90 Days', '180 Days', '365 Days'], fontsize=11, fontweight='bold')

    for i in range(len(period_order) + 1):
        ax.axhline(i - 0.5, color='white', linewidth=3)
    for j in range(len(mcap_order) + 1):
        ax.axvline(j - 0.5, color='white', linewidth=3)

    for spine in ax.spines.values():
        spine.set_visible(False)

    for i, period in enumerate(period_order):
        for j, mcap in enumerate(mcap_order):
            cell_data = all_stats[
                (all_stats['Holding_Period'] == period) &
                (all_stats['MCap_Category'] == mcap)]
            if len(cell_data) > 0:
                abnormal = cell_data['Our_Outperformance_%'].values[0]
                stock_ret = cell_data['Our_Stock_Return_%'].values[0]
                index_ret = cell_data['Index_Return_%'].values[0]
                n = int(cell_data['n'].values[0])

                if abnormal > 9:
                    text_color = 'white'
                else:
                    text_color = 'black'

                abnormal_str = f"{abnormal:.1f}%"

                ax.text(j, i - 0.05, abnormal_str,
                        ha='center', va='center',
                        fontsize=18, fontweight='bold',
                        color=text_color)

                ax.text(j, i + 0.18, f"({stock_ret:.1f}% vs {index_ret:.1f}%)",
                        ha='center', va='center',
                        fontsize=10,
                        color=text_color)

                ax.text(j, i + 0.35, f"(n={n})",
                        ha='center', va='center',
                        fontsize=10,
                        color=text_color)

    # Lables and colorbar
    ax.set_xlabel('Market Capitalization (SEK)', fontsize=12, fontweight='bold')
    ax.set_ylabel('Holding Period', fontsize=12, fontweight='bold')
    ax.set_title('Mean Abnormal Return by Market Cap and Holding Period',
                 fontsize=14, fontweight='bold', pad=20)

    # Add colorbar with border
    cbar = plt.colorbar(im, ax=ax, shrink=0.8, pad=0.02)
    cbar.set_label('Mean Abnormal Return (%)', fontsize=11, fontweight='bold')
    cbar.outline.set_visible(True)  # Keep colorbar border
    cbar.outline.set_edgecolor('black')
    cbar.outline.set_linewidth(1)

    plt.tight_layout()
    plt.savefig('heatmap_final_v4.png', dpi=300, bbox_inches='tight',
                facecolor='white', edgecolor='none')
    plt.savefig('heatmap_final_v4.pdf', bbox_inches='tight',
                facecolor='white', edgecolor='none')

    pivot_abnormal = all_stats.pivot(
        index='Holding_Period',
        columns='MCap_Category',
        values='Our_Outperformance_%'
    ).reindex(columns=mcap_order, index=period_order)

    pivot_n = all_stats.pivot(
        index='Holding_Period',
        columns='MCap_Category',
        values='n'
    ).reindex(columns=mcap_order, index=period_order)

    plt.show()

```

Figure 6. Comparing win rates compared to index within the 1-5B SEK market cap.

```python
import pandas as pd
import matplotlib.pyplot as plt

# File paths
df_90 = pd.read_excel(r"C:\Users\Documents\FIN11\Results_Primary_Insider_500k_90.xlsx")
df_180 = pd.read_excel(r"C:\Users\Documents\FIN11\Results_Primary_Insider_500k_180.xlsx")
df_365 = pd.read_excel(r"C:\Users\Documents\FIN11\Results_Primary_Insider_500k_365.xlsx")

# Colors
COLOR_BLUE = '#0066FF'
COLOR_RED = '#e74c3c'

# Filtering the 1-5B segment
def filter_midcap(df):
    return df[(df['Market_Cap_SEK_Millions'] >= 1000) & (df['Market_Cap_SEK_Millions'] < 5000)]

df_90_midcap = filter_midcap(df_90)
df_180_midcap = filter_midcap(df_180)
df_365_midcap = filter_midcap(df_365)

# Calculating win rates
def calc_win_rate(df):
    wins = (df['Our_Outperformance_%'] > 0).sum()
    total = len(df)
    losses = total - wins
    return wins, losses, total

fig, axes = plt.subplots(1, 3, figsize=(14, 5))
fig.suptitle('Win Rate by Holding Period (1-5B SEK Market Cap)', fontweight='bold', fontsize=16, y=0.98)

datasets = [
    (df_90_midcap, '90 Days'),
    (df_180_midcap, '180 Days'),
    (df_365_midcap, '365 Days')]

for ax, (df, period) in zip(axes, datasets):
    wins, losses, total = calc_win_rate(df)
    win_rate = wins / total * 100

    sizes = [wins, losses]
    colors = [COLOR_BLUE, COLOR_RED]

    ax.pie(sizes, colors=colors, startangle=90,
           wedgeprops=dict(width=0.4, edgecolor='white', linewidth=2))

    ax.text(0, 0.05, f'{win_rate:.1f}%', ha='center', va='center',
            fontsize=24, fontweight='bold', color=COLOR_BLUE)
    ax.text(0, -0.15, 'Win Rate', ha='center', va='center',
            fontsize=10, color=COLOR_BLUE)

    ax.set_title(f'{period}', fontweight='bold', fontsize=13, pad=10)

    ax.text(0, -0.65, f'{wins} wins | {losses} losses', ha='center', va='center',
            fontsize=9, color='gray')
    ax.text(0, -0.78, 'Outperformed | Underperformed', ha='center', va='center',
            fontsize=8, color='lightgray', style='italic')

plt.tight_layout()
plt.savefig('winrate_figure.png', dpi=300, bbox_inches='tight', facecolor='white')
plt.show()
```

Figure 8. Nested heatmap for market cap and transaction value.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns


FILE_90 = r"C:\Users\Documents\FIN11\Results_Primary_Insider_500k_90.xlsx"
FILE_180 = r"C:\Users\Documents\FIN11\Results_Primary_Insider_500k_180.xlsx"
FILE_365 = r"C:\Users\Documents\FIN11\Results_Primary_Insider_500k_365.xlsx"

MIN_OBSERVATIONS = 75

df_90 = pd.read_excel(FILE_90)
df_180 = pd.read_excel(FILE_180)
df_365 = pd.read_excel(FILE_365)

def segment_data(df):
    df = df.copy()

    # Market cap
    df['Market_Cap_Billions'] = df['Market_Cap_SEK_Millions'] / 1000
    cap_conditions = [
        df['Market_Cap_Billions'] < 1,
        (df['Market_Cap_Billions'] >= 1) & (df['Market_Cap_Billions'] < 5),
        (df['Market_Cap_Billions'] >= 5) & (df['Market_Cap_Billions'] < 10),
        df['Market_Cap_Billions'] >= 10]
    cap_choices = ['<1B', '1-5B', '5-10B', '>10B']
    df['Market_Cap_Segment'] = np.select(cap_conditions, cap_choices, default='Unknown')

    # Transaction value
    df['Transaction_Value_MSEK'] = df['Transaction_Value_SEK'] / 1_000_000
    val_conditions = [
        df['Transaction_Value_MSEK'] < 1,
        (df['Transaction_Value_MSEK'] >= 1) & (df['Transaction_Value_MSEK'] < 5),
        (df['Transaction_Value_MSEK'] >= 5) & (df['Transaction_Value_MSEK'] < 10),
        df['Transaction_Value_MSEK'] >= 10]
    val_choices = ['0.5-1M', '1-5M', '5-10M', '>10M']
    df['Transaction_Segment'] = np.select(val_conditions, val_choices, default='Unknown')

    return df
def calculate_nested_stats(df, period_name):
    results = []

    cap_order = ['<1B', '1-5B', '5-10B', '>10B']
    val_order = ['0.5-1M', '1-5M', '5-10M', '>10M']

    for cap in cap_order:
        for val in val_order:
            subset = df[(df['Market_Cap_Segment'] == cap) &
                        (df['Transaction_Segment'] == val)]

            if len(subset) > 0:
                results.append({
                    'Period': period_name,
                    'Market_Cap': cap,
                    'Transaction_Value': val,
                    'Mean_Alpha': subset['Our_Outperformance_%'].mean(),
                    'N': len(subset)
                })

    return pd.DataFrame(results)

df_90 = segment_data(df_90)
df_180 = segment_data(df_180)
df_365 = segment_data(df_365)

stats_90 = calculate_nested_stats(df_90, '90d')
stats_180 = calculate_nested_stats(df_180, '180d')
stats_365 = calculate_nested_stats(df_365, '365d')

all_stats = pd.concat([stats_90, stats_180, stats_365], ignore_index=True)

fig, axes = plt.subplots(1, 3, figsize=(18, 8))

periods = ['90d', '180d', '365d']
cap_order = ['<1B', '1-5B', '5-10B', '>10B']
val_order = ['0.5-1M', '1-5M', '5-10M', '>10M']

# Find global min/max for consistent color scale
all_alphas = all_stats[all_stats['N'] >= MIN_OBSERVATIONS]['Mean_Alpha']
vmin = all_alphas.min()
vmax = all_alphas.max()
abs_max = max(abs(vmin), abs(vmax))
```

```python
abs_max = max(abs(vmin), abs(vmax))

for idx, period in enumerate(periods):
    ax = axes[idx]

    period_data = all_stats[all_stats['Period'] == period]

    pivot_alpha = period_data.pivot(index='Transaction_Value',
                                    columns='Market_Cap',
                                    values='Mean_Alpha')
    pivot_n = period_data.pivot(index='Transaction_Value',
                                columns='Market_Cap',
                                values='N')

    pivot_alpha = pivot_alpha.reindex(index=val_order, columns=cap_order)
    pivot_n = pivot_n.reindex(index=val_order, columns=cap_order)

    mask = pivot_n < MIN_OBSERVATIONS

    sns.heatmap(pivot_alpha,
                ax=ax,
                mask=mask,
                cmap='RdYlGn',
                center=0,
                vmin=-abs_max,
                vmax=abs_max,
                annot=False,
                cbar=idx == 2,
                cbar_kws={'label': 'Mean Alpha (%)'} if idx == 2 else {})

    for i, val in enumerate(val_order):
        for j, cap in enumerate(cap_order):
            if not mask.iloc[i, j]:
                alpha = pivot_alpha.iloc[i, j]
                n = int(pivot_n.iloc[i, j])

                text_color = 'white' if abs(alpha) > abs_max * 0.6 else 'black'

                ax.text(j + 0.5, i + 0.5,
                        f'{alpha:.1f}%\nn={n}',
                        ha='center', va='center',
                        fontsize=10, fontweight='bold',
                        color=text_color)

    ax.set_title(f'{period} Holding Period', fontsize=14, fontweight='bold')
    ax.set_xlabel('Market Cap (SEK)', fontsize=12)
    ax.set_ylabel('Transaction Value (SEK)' if idx == 0 else '', fontsize=12)
    ax.set_xticklabels(cap_order, rotation=0)
    ax.set_yticklabels(val_order, rotation=0)

plt.suptitle('Abnormal Returns by Market Cap and Transaction Value\n(cells with n < 75 hidden)',
             fontsize=16, fontweight='bold', y=1.02)

plt.tight_layout()
plt.savefig('nested_heatmap.png', dpi=300, bbox_inches='tight', facecolor='white')
plt.show()
```

Figure 9. Scatter plot comparing the return across transaction sizes.

```
    import pandas as pd
    import numpy as np
    import matplotlib.pyplot as plt
    from matplotlib.colors import LinearSegmentedColormap, TwoSlopeNorm
    from scipy import stats

    FILE_90 = r"C:\Users\Documents\FIN11\Results_Primary_Insider_500k_90.xlsx"
    FILE_180 = r"C:\Users\Documents\FIN11\Results_Primary_Insider_500k_180.xlsx"
    FILE_365 = r"C:\Users\Documents\FIN11\Results_Primary_Insider_500k_365.xlsx"

    COLOR_BLUE = '#0066FF'
    COLOR_RED = '#e74c3c'

    RANGES = [
        ('500k - 1M SEK', 500000, 1000000),
        ('1M - 5M SEK', 1000000, 5000000),
        ('5M - 10M SEK', 5000000, 10000000),
        ('10M+ SEK', 10_000_000, float('inf'))]

    def create_scatter_plot(df, holding_period, output_filename):
        """Create scatter plot for a specific holding period."""

        # Colormap (red, white blue)
        colors_gradient = [COLOR_RED, '#f5b7b1', '#fef9e7', '#aed6f1', COLOR_BLUE]
        cmap = LinearSegmentedColormap.from_list('custom', colors_gradient)

        # Filter for market cap
        df_filtered = df[(df['Market_Cap_SEK_Millions'] >= 1000) &
                         (df['Market_Cap_SEK_Millions'] <= 5000)].copy()

        fig, axes = plt.subplots(2, 2, figsize=(14, 12))
        axes = axes.flatten()
        fig.suptitle(f'Strategy vs Index Return by Transaction Size\n(1-5B SEK Market Cap, {holding_period}-Day Holding Period)',
                     fontsize=16, fontweight='bold', y=1.02)

        for idx, (ax, (range_name, low, high)) in enumerate(zip(axes, RANGES)):
            df_range = df_filtered[(df_filtered['Transaction_Value_SEK'] >= low) &
                                   (df_filtered['Transaction_Value_SEK'] < high)].copy()

            if len(df_range) < 5:
                ax.text(0.5, 0.5, 'N < 5', ha='center', va='center', fontsize=14)
                ax.set_title(f'{range_name}', fontsize=13, fontweight='bold')
                ax.set_xlim(-150, 150)
                ax.set_ylim(-150, 150)
                continue

            x = df_range['Index_Return_%'].values
            y = df_range['Our_Stock_Return_%'].values
            outperf = df_range['Our_Outperformance_%'].values

            ax.set_xlim(-150, 150)
            ax.set_ylim(-150, 150)

            ax.plot([-150, 150], [-150, 150], '--', color='gray', linewidth=1.5, alpha=0.7,
                    label='Equal Performance')

            # Best fit line (min to max)
            if len(x) > 2:
                slope, intercept, r_value, p_value, std_err = stats.linregress(x, y)
                x_fit = np.linspace(x.min(), x.max(), 100)
                y_fit = slope * x_fit + intercept
                ax.plot(x_fit, y_fit, '-', color=COLOR_BLUE, linewidth=2.5,
                        label=f'Best Fit (slope={slope:.2f})')

            mean_alpha = outperf.mean()
            win_rate = (outperf > 0).mean() * 100
            corr = np.corrcoef(x, y)[0, 1]

            stats_text = f'N = {len(df_range)}\nMean α = {mean_alpha:.1f}%\nWin Rate = {win_rate:.1f}%\nCorr = {corr:.2f}'
            props = dict(boxstyle='round,pad=0.4', facecolor='white', edgecolor=COLOR_BLUE,
                         linewidth=1.5, alpha=0.9)
            ax.text(0.03, 0.97, stats_text, transform=ax.transAxes, fontsize=9, fontweight='bold',
                    verticalalignment='top', bbox=props)

            ax.set_xlabel('Index Return (%)', fontsize=11, fontweight='bold')
            ax.set_ylabel('Strategy Return (%)', fontsize=11, fontweight='bold')
            ax.set_title(f'{range_name}', fontsize=13, fontweight='bold', pad=10)
            ax.axhline(y=0, color='gray', linestyle='-', linewidth=0.5, alpha=0.5)
            ax.axvline(x=0, color='gray', linestyle='-', linewidth=0.5, alpha=0.5)
            ax.grid(True, alpha=0.3, linestyle='--')
            ax.spines['top'].set_visible(False)
            ax.spines['right'].set_visible(False)
            ax.legend(loc='lower right', fontsize=8)

        cbar_ax = fig.add_axes([0.92, 0.15, 0.02, 0.7])
        sm = plt.cm.ScalarMappable(cmap=cmap, norm=TwoSlopeNorm(vmin=-50, vcenter=0, vmax=50))
        sm.set_array([])
        cbar = fig.colorbar(sm, cax=cbar_ax)
        cbar.set_label('Excess Return (%)', fontsize=11, fontweight='bold')

        plt.tight_layout(rect=[0, 0, 0.9, 0.96])
        plt.savefig(output_filename, dpi=300, bbox_inches='tight', facecolor='white')
        plt.show()

    # 90-day period
    df_90 = pd.read_excel(FILE_90)
    create_scatter_plot(df_90, 90, 'scatter_1_5B_90days.png')
```

Figure 11. Annual performance buying the dip.

```
import pandas as pd
import matplotlib.pyplot as plt
from matplotlib.patches import Patch

# Load data
file_path = r"C:\Users\Documents\FIN11\Results_Lookback_Filtered.xlsx"
df = pd.read_excel(file_path, sheet_name='Hold_365d')

# Filter for 30-day lookback with -20% condition
df_dip = df[df['Condition_Met'] == '30d down -20%'].copy()
df_dip['Year'] = pd.to_datetime(df_dip['Publication_Date']).dt.year

# Calculate yearly
yearly_stats = df_dip.groupby('Year').agg({
    'Our_Outperformance_%': ['mean', 'count']
}).reset_index()
yearly_stats.columns = ['Year', 'Alpha', 'N']

fig, ax1 = plt.subplots(figsize=(12, 8))

years = yearly_stats['Year'].tolist()
alpha_values = yearly_stats['Alpha'].tolist()
n_values = yearly_stats['N'].tolist()

x_pos = range(len(years))

colors = ['#0066FF' if a > 0 else '#e74c3c' for a in alpha_values]
bars = ax1.bar(x_pos, alpha_values, color=colors, edgecolor='none', alpha=0.9, width=0.5)

ax1.axhline(y=0, color='black', linewidth=1.2)
ax1.set_ylabel('Mean Abnormal Return (%)', fontweight='bold', fontsize=12)

for i, (bar, alpha, year) in enumerate(zip(bars, alpha_values, years)):
    height = bar.get_height()

    if height >= 0:
        y_pos = height + 2
        va = 'bottom'
    else:
        y_pos = height - 2
        va = 'top'

    ax1.annotate(f'{alpha:.1f}%',
                 xy=(i, y_pos),
                 ha='center', va=va,
                 fontsize=11, fontweight='bold', color='black')

ax1.set_ylim(-70, 70)
ax1.set_yticks([-60, -40, -20, 0, 20, 40, 60])

ax1.set_xticks(x_pos)
ax1.set_xticklabels([])

for i, (year, n) in enumerate(zip(years, n_values)):
    ax1.text(i, -77, f'{year}', ha='center', va='top', fontsize=12, fontweight='bold')
    ax1.text(i, -82, f'n={n}', ha='center', va='top', fontsize=9)

ax1.grid(axis='y', alpha=0.3, linestyle='--')
ax1.spines['top'].set_visible(False)
ax1.spines['right'].set_visible(False)

legend_elements = [
    Patch(facecolor='#0066FF', edgecolor='none', label='Positive Alpha'),
    Patch(facecolor='#e74c3c', edgecolor='none', label='Negative Alpha'),]
ax1.legend(handles=legend_elements, loc='lower left', fontsize=10)

fig.suptitle('Buy The Dip: Performance by Year', fontweight='bold', fontsize=14, y=0.96)
fig.text(0.5, 0.90, '(30-day lookback, -20% trigger, 365-day holding period)',
         ha='center', fontsize=11, color='black')
fig.text(0.5, 0.86, 'Trades are grouped by the year of purchase',
         ha='center', fontsize=10, color='gray', style='italic')

plt.tight_layout()
plt.subplots_adjust(top=0.87, bottom=0.15)
plt.savefig('buy_the_dip.png', dpi=300, bbox_inches='tight', facecolor='white')
plt.show()
```

Figure 12. CEO + CFO vs All Others. Mean excess returns.

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

FILE_90 = r"C:\Users\Documents\FIN11\Results_Primary_Insider_500k_90.xlsx"
FILE_180 = r"C:\Users\Documents\FIN11\Results_Primary_Insider_500k_180.xlsx"
FILE_365 = r"C:\Users\Documents\FIN11\Results_Primary_Insider_500k_365.xlsx"

COLOR_INSIDER = '#0066FF'
COLOR_BENCHMARK = '#e74c3c'

plt.rcParams['font.family'] = 'sans-serif'
plt.rcParams['font.sans-serif'] = ['Arial', 'DejaVu Sans']
plt.rcParams['font.size'] = 13
plt.rcParams['font.weight'] = 'bold'

df_90 = pd.read_excel(FILE_90)
df_180 = pd.read_excel(FILE_180)
df_365 = pd.read_excel(FILE_365)

df_90['Holding_Period'] = 90
df_180['Holding_Period'] = 180
df_365['Holding_Period'] = 365

df_all = pd.concat([df_90, df_180, df_365], ignore_index=True)

def categorize_position(position):
    """Classify insider positions into CEO, CFO, or Other."""
    if pd.isna(position):
        return 'Other'
    position_upper = str(position).upper()
    if 'CEO' in position_upper or 'VD' in position_upper or 'VERKSTÄLLANDE DIREKTÖR' in position_upper:
        return 'CEO'
    elif 'CFO' in position_upper or 'EKONOMIDIREKTØR' in position_upper or 'FINANCE' in position_upper:
        return 'CFO'
    else:
        return 'Other'

df_all['Position_Category'] = df_all['Position'].apply(categorize_position)
df_all['Position_Group'] = df_all['Position_Category'].apply(
    lambda x: 'CEO + CFO' if x in ['CEO', 'CFO'] else 'All Others')

df_plot = df_all[df_all['Position_Group'].isin(['CEO + CFO', 'All Others'])].copy()

fig, ax = plt.subplots(1, 1, figsize=(16, 10))

avg_returns = df_plot.groupby(['Position_Group', 'Holding_Period'])['Our_Outperformance_%'].mean().reset_index()
pivot_data = avg_returns.pivot(index='Holding_Period', columns='Position_Group', values='Our_Outperformance_%')
pivot_data = pivot_data[['CEO + CFO', 'All Others']]

x = np.arange(len(pivot_data.index))
width = 0.35

bars1 = ax.bar(x - width/2, pivot_data['CEO + CFO'].values, width,
               label='CEO + CFO', color=COLOR_INSIDER,
               edgecolor='black', linewidth=2.5, alpha=0.9, zorder=3)
bars2 = ax.bar(x + width/2, pivot_data['All Others'].values, width,
               label='All Others', color=COLOR_BENCHMARK,
               edgecolor='black', linewidth=2.5, alpha=0.9, zorder=3)

for bars in [bars1, bars2]:
    for bar in bars:
        height = bar.get_height()
        y_pos = height + (0.35 if height > 0 else -0.35)
        va = 'bottom' if height > 0 else 'top'
        ax.text(bar.get_x() + bar.get_width()/2., y_pos,
                f'{height:.2f}%',
                ha='center', va=va,
                fontsize=16, fontweight='bold', zorder=4)

ax.axhline(y=0, color='black', linestyle='-', linewidth=3, alpha=0.8, zorder=2)
ax.set_xlabel('Holding Period (Days)', fontsize=17, fontweight='bold', labelpad=15)
ax.set_ylabel('Mean Excess Return (%)', fontsize=17, fontweight='bold', labelpad=15)
ax.set_title('CEO + CFO vs All Others: Mean Excess Returns\nDo Top Executives Outperform?',
             fontsize=20, fontweight='bold', pad=30)
ax.set_xticks(x)
ax.set_xticklabels(['90 Days', '180 Days', '365 Days'], fontsize=16, fontweight='bold')
ax.tick_params(axis='y', labelsize=15)
ax.legend(fontsize=16, loc='upper left', frameon=True, shadow=True, framealpha=0.95,
          edgecolor='black', fancybox=True).get_frame().set_linewidth(2)
ax.grid(True, alpha=0.25, linestyle='--', zorder=1)
ax.set_axisbelow(True)

y_min = min(pivot_data.min().min(), -2) - 1
y_max = max(pivot_data.max().max(), 3) + 1
ax.set_ylim([y_min, y_max])
ax.set_facecolor('#f8f9fa')

plt.tight_layout()
plt.savefig('mean_excess_returns_ceo_cfo.png', dpi=300, bbox_inches='tight', facecolor='white')
plt.show()
```

Figure 13. Distribution of excess returns. CEO + CFO vs All Others.

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.lines import Line2D

FILE_90 = r"C:\Users\Documents\FIN11\Results_Primary_Insider_500k_90.xlsx"
FILE_180 = r"C:\Users\Documents\FIN11\Results_Primary_Insider_500k_180.xlsx"
FILE_365 = r"C:\Users\Documents\FIN11\Results_Primary_Insider_500k_365.xlsx"

COLOR_INSIDER = '#0066FF'
COLOR_BENCHMARK = '#e74c3c'

plt.rcParams['font.size'] = 12
plt.rcParams['font.weight'] = 'bold'

df_90 = pd.read_excel(FILE_90)
df_180 = pd.read_excel(FILE_180)
df_365 = pd.read_excel(FILE_365)

df_90['Holding_Period'] = 90
df_180['Holding_Period'] = 180
df_365['Holding_Period'] = 365

df_all = pd.concat([df_90, df_180, df_365], ignore_index=True)

def categorize_position(position):
    """Classify insider positions into CEO, CFO, or Other."""
    if pd.isna(position):
        return 'Other'
    position_upper = str(position).upper()
    if 'CEO' in position_upper or 'VD' in position_upper or 'VERKSTÄLLANDE DIREKTÖR' in position_upper:
        return 'CEO'
    elif 'CFO' in position_upper or 'EKONOMIDIREKTØR' in position_upper or 'FINANCE' in position_upper:
        return 'CFO'
    else:
        return 'Other'

df_all['Position_Category'] = df_all['Position'].apply(categorize_position)
df_all['Position_Group'] = df_all['Position_Category'].apply(
    lambda x: 'CEO + CFO' if x in ['CEO', 'CFO'] else 'All Others')

df_plot = df_all[df_all['Position_Group'].isin(['CEO + CFO', 'All Others'])].copy()

fig, axes = plt.subplots(1, 3, figsize=(20, 8))

holding_periods = [90, 180, 365]
period_names = ['90 Days', '180 Days', '365 Days']

for idx, (period, period_name) in enumerate(zip(holding_periods, period_names)):
    ax = axes[idx]
    data_period = df_plot[df_plot['Holding_Period'] == period].copy()

    ceo_cfo_data = data_period[data_period['Position_Group'] == 'CEO + CFO']['Our_Outperformance_%'].values
    others_data = data_period[data_period['Position_Group'] == 'All Others']['Our_Outperformance_%'].values

    bp = ax.boxplot([ceo_cfo_data, others_data],
                    positions=[0, 1],
                    widths=0.6,
                    patch_artist=True,
                    showmeans=True,
                    meanprops=dict(marker='D', markerfacecolor='white', markeredgecolor='black',
                                   markersize=10, linewidth=2.5),
                    medianprops=dict(color='black', linewidth=3.5),
                    boxprops=dict(linewidth=2.5),
                    whiskerprops=dict(linewidth=2.5),
                    capprops=dict(linewidth=2.5),
                    flierprops=dict(marker='o', alpha=0.4, markersize=5))

    bp['boxes'][0].set_facecolor(COLOR_INSIDER)
    bp['boxes'][0].set_alpha(0.8)
    bp['boxes'][0].set_edgecolor('black')
    bp['boxes'][1].set_facecolor(COLOR_BENCHMARK)
    bp['boxes'][1].set_alpha(0.8)
    bp['boxes'][1].set_edgecolor('black')

    ax.axhline(y=0, color='black', linestyle='--', linewidth=2.5, alpha=0.7, zorder=1)

    ax.set_xticks([0, 1])
    ax.set_xticklabels(['CEO + CFO', 'All Others'], fontsize=14, fontweight='bold')
    ax.set_ylabel('Excess Return (%)', fontsize=14, fontweight='bold', labelpad=10)
    ax.set_title(f'{period_name}', fontsize=16, fontweight='bold', pad=15)
    ax.grid(True, alpha=0.3, axis='y', linestyle='--', linewidth=1.5, zorder=0)
    ax.set_axisbelow(True)

    all_data = np.concatenate([ceo_cfo_data, others_data])
    y_min = np.percentile(all_data, 0.5) - 20
    y_max = np.percentile(all_data, 99.5) + 20
    ax.set_ylim([y_min, y_max])
    ax.set_facecolor('#f8f9fa')

legend_elements = [
    Line2D([0], [0], color='black', linewidth=3.5, label='Median (Middle Value)'),
    Line2D([0], [0], marker='D', color='w', markerfacecolor='white',
           markeredgecolor='black', markersize=10, linewidth=0, label='Mean (Average)'),
    Line2D([0], [0], color='black', linestyle='--', linewidth=2.5, label='Zero = Break-Even')]

legend = fig.legend(handles=legend_elements, loc='upper center', bbox_to_anchor=(0.5, 1.00),
                    ncol=3, fontsize=13, frameon=True, shadow=True, framealpha=0.98,
                    edgecolor='black', fancybox=True)
legend.get_frame().set_linewidth(2)

fig.text(0.5, 1.08, 'Distribution of Excess Returns: CEO + CFO vs All Others',
         ha='center', fontsize=20, fontweight='bold', transform=fig.transFigure)

fig.text(0.5, 1.04, 'Box Shows Where 50% of Trades Fall | Wider Box = More Variation',
         ha='center', fontsize=14, style='italic', transform=fig.transFigure,
         color='#333', weight='bold')

plt.tight_layout(rect=[0, 0, 1, 0.94])
plt.savefig('distribution_excess_returns_ceo_cfo.png', dpi=300, bbox_inches='tight', facecolor='white')
plt.show()
```

Figure 14. T-test and plot showing the results.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from scipy import stats
import seaborn as sns

FILE_90 = r"C:\Users\Documents\FIN11\Results_Primary_Insider_500k_90.xlsx"
FILE_180 = r"C:\Users\Documents\FIN11\Results_Primary_Insider_500k_180.xlsx"
FILE_365 = r"C:\Users\Documents\FIN11\Results_Primary_Insider_500k_365.xlsx"

sns.set_style("whitegrid")
plt.rcParams['font.size'] = 10
plt.rcParams['figure.facecolor'] = 'white'

df_90 = pd.read_excel(FILE_90)
df_180 = pd.read_excel(FILE_180)
df_365 = pd.read_excel(FILE_365)


# T-test
def one_sample_t_test_analysis(data, period_name):
    """
    Perform one-sample t-test and create visualization.
    """
    returns = data['Our_Outperformance_%'].dropna().values

    # Calculate statistics
    n = len(returns)
    mean_return = np.mean(returns)
    median_return = np.median(returns)
    std_return = np.std(returns, ddof=1)
    se_return = std_return / np.sqrt(n)

    # One-sample t-test (one-tailed)
    t_statistic, p_value_two_tailed = stats.ttest_1samp(returns, 0)
    p_value_one_tailed = p_value_two_tailed / 2 if t_statistic > 0 else 1 - p_value_two_tailed / 2

    # 95% Confidence interval
    ci_95 = stats.t.interval(0.95, n-1, loc=mean_return, scale=se_return)

    # Effect size (Cohen's d)
    cohens_d = mean_return / std_return

    # Create figure
    fig = plt.figure(figsize=(16, 7))
    gs = fig.add_gridspec(1, 2, width_ratios=[1.2, 1], wspace=0.25)
    ax1 = fig.add_subplot(gs[0])
    ax2 = fig.add_subplot(gs[1])

    # LEFT: Distribution histogram
    ax1.hist(returns, bins=50, color='steelblue', alpha=0.7, edgecolor='black', density=True)

    # Normal distribution overlay
    x = np.linspace(returns.min(), returns.max(), 100)
    normal_dist = stats.norm.pdf(x, mean_return, std_return)
    ax1.plot(x, normal_dist, 'r-', linewidth=2, label='Normal Distribution')

    # Reference lines
    ax1.axvline(0, color='black', linestyle='--', linewidth=2, label='H₀: μ = 0', alpha=0.7)
    ax1.axvline(mean_return, color='green', linestyle='-', linewidth=2.5, label=f'Mean = {mean_return:.2f}%')
    ax1.axvline(median_return, color='orange', linestyle='-', linewidth=1.5, label=f'Median = {median_return:.2f}%')
    ax1.axvspan(ci_95[0], ci_95[1], alpha=0.2, color='green', label='95% CI')

    ax1.set_xlabel('Abnormal Return (%)', fontsize=13, fontweight='bold')
    ax1.set_ylabel('Density', fontsize=13, fontweight='bold')
    ax1.set_title(f'Distribution of Abnormal Returns - {period_name}', fontsize=15, fontweight='bold', pad=15)
    ax1.legend(loc='upper right', fontsize=10)
    ax1.grid(alpha=0.3)

    # RIGHT: Statistics summary
    ax2.axis('off')

    stats_text = f"""
ONE-SAMPLE T-TEST RESULTS
{'='*45}

Hypothesis Testing:
  H₀: μ ≤ 0  (No abnormal return)
  H₁: μ > 0  (Positive abnormal return)

{'='*45}
DESCRIPTIVE STATISTICS
{'='*45}
  Sample Size (n)          {n:>14,}
  Mean Return              {mean_return:>14.2f}%
  Median Return            {median_return:>14.2f}%
  Std. Deviation           {std_return:>14.2f}%
  Std. Error               {se_return:>14.4f}%

{'='*45}
TEST STATISTICS
{'='*45}
  t-statistic              {t_statistic:>14.4f}
  p-value (one-tailed)     {p_value_one_tailed:>14.6f}

  95% Confidence Interval:
    [{ci_95[0]:.2f}%, {ci_95[1]:.2f}%]
```

```
    {'='*45}
    EFFECT SIZE
    {'='*45}
      Cohen's d               {cohens_d:>14.4f}
    """

        ax2.text(0.05, 0.98, stats_text, transform=ax2.transAxes, fontsize=11,
                 verticalalignment='top', fontfamily='monospace',
                 bbox=dict(boxstyle='round', facecolor='wheat', alpha=0.25, pad=1))

        ax2.text(0.5, 0.98, 'Statistical Test Results', transform=ax2.transAxes,
                 fontsize=15, fontweight='bold', ha='center', va='top')

        plt.tight_layout()

        return fig, {
            'n': n, 'mean': mean_return, 'median': median_return,
            'std': std_return, 'se': se_return, 't_stat': t_statistic,
            'p_value': p_value_one_tailed, 'ci_95': ci_95, 'cohens_d': cohens_d,
            'significant': p_value_one_tailed < 0.05}


    # Analysis
    periods = [
        (df_90, '90-Day Holding Period', 90),
        (df_180, '180-Day Holding Period', 180),
        (df_365, '365-Day Holding Period', 365)]

    results_summary = []

    for df, period_name, days in periods:
        fig, results = one_sample_t_test_analysis(df, period_name)
        plt.savefig(f't_test_analysis_{days}_days.png', dpi=300, bbox_inches='tight')
        results_summary.append({'Period': period_name, **results})
        plt.close()


    # SUMMARY COMPARISON PLOT

    fig, axes = plt.subplots(2, 2, figsize=(14, 10))

    # Plot 1: Mean returns with CI
    ax = axes[0, 0]
    periods_short = ['90d', '180d', '365d']
    means = [r['mean'] for r in results_summary]
    ci_lower = [r['ci_95'][0] for r in results_summary]
    ci_upper = [r['ci_95'][1] for r in results_summary]
    errors_lower = [means[i] - ci_lower[i] for i in range(3)]
    errors_upper = [ci_upper[i] - means[i] for i in range(3)]

    colors = ['green' if r['significant'] else 'red' for r in results_summary]
    bars = ax.bar(periods_short, means, color=colors, alpha=0.6, edgecolor='black')
    ax.errorbar(periods_short, means, yerr=[errors_lower, errors_upper],
                fmt='none', color='black', capsize=5, capthick=2)
    ax.axhline(0, color='black', linestyle='--', linewidth=1)
    ax.set_ylabel('Mean Abnormal Return (%)', fontweight='bold')
    ax.set_title('Mean Abnormal Returns with 95% CI', fontweight='bold')
    ax.grid(axis='y', alpha=0.3)

    for bar, mean, sig in zip(bars, means, [r['significant'] for r in results_summary]):
        label = f"{mean:.2f}%" + (" *" if sig else "")
        ax.text(bar.get_x() + bar.get_width()/2, bar.get_height() + 0.5,
                label, ha='center', va='bottom', fontweight='bold')

    # Plot 2: T-statistics
    ax = axes[0, 1]
    t_stats = [r['t_stat'] for r in results_summary]
    bars = ax.bar(periods_short, t_stats, color='steelblue', alpha=0.7, edgecolor='black')
    ax.axhline(1.96, color='orange', linestyle='--', linewidth=2, label='Critical value (α=0.05)')
    ax.axhline(2.576, color='red', linestyle='--', linewidth=2, label='Critical value (α=0.01)')
    ax.set_ylabel('t-statistic', fontweight='bold')
    ax.set_title('T-Statistics by Holding Period', fontweight='bold')
    ax.legend()
    ax.grid(axis='y', alpha=0.3)

    for bar, t_stat in zip(bars, t_stats):
        ax.text(bar.get_x() + bar.get_width()/2, bar.get_height() + 0.1,
                f"{t_stat:.2f}", ha='center', va='bottom', fontweight='bold')

    # Plot 3: P-values
    ax = axes[1, 0]
    p_values = [r['p_value'] for r in results_summary]
    bars = ax.bar(periods_short, p_values, color='coral', alpha=0.7, edgecolor='black')
    ax.axhline(0.05, color='red', linestyle='--', linewidth=2, label='α = 0.05')
    ax.axhline(0.01, color='darkred', linestyle='--', linewidth=2, label='α = 0.01')
    ax.set_ylabel('p-value (one-tailed)', fontweight='bold')
    ax.set_title('P-Values by Holding Period', fontweight='bold')
    ax.legend()
    ax.grid(axis='y', alpha=0.3)

    for bar, p_val in zip(bars, p_values):
        ax.text(bar.get_x() + bar.get_width()/2, bar.get_height() + 0.001,
                f"{p_val:.4f}", ha='center', va='bottom', fontweight='bold', fontsize=9)

    # Plot 4: Summary table
    ax = axes[1, 1]
    ax.axis('off')
```

```
    summary_data = []
    for r in results_summary:
        period = r['Period'].replace('-Day Holding Period', 'd')
        sig = '✓' if r['significant'] else '✗'
        summary_data.append([period, f"{r['n']:,}", f"{r['mean']:.2f}%",
                             f"{r['std']:.2f}%", f"{r['t_stat']:.2f}",
                             f"{r['p_value']:.4f}", sig])

    table = ax.table(cellText=summary_data,
                     colLabels=['Period', 'N', 'Mean', 'Std', 't-stat', 'p-value', 'Sig'],
                     cellLoc='center', loc='center', bbox=[0, 0.2, 1, 0.6])

    table.auto_set_font_size(False)
    table.set_fontsize(10)
    table.scale(1, 2)

    for i in range(7):
        table[(0, i)].set_facecolor('#4CAF50')
        table[(0, i)].set_text_props(weight='bold', color='white')

    for i in range(1, 4):
        sig_cell = table[(i, 6)]
        sig_cell.set_facecolor('#90EE90' if summary_data[i-1][6] == '✓' else '#FFB6C1')

    ax.set_title('Summary Statistics', fontweight='bold', fontsize=12, pad=20)

    plt.suptitle('ONE-SAMPLE T-TEST: ABNORMAL RETURNS > 0?\nStatistical Analysis Summary',
                 fontsize=14, fontweight='bold', y=0.98)

    plt.tight_layout()
    plt.savefig('t_test_summary_comparison.png', dpi=300, bbox_inches='tight')
    plt.show()
```