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.:	1027

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:	12
Other members:	1011, 1074, 1012, 1023

In which language did you write your answer paper? *

English



BOARDING PASS

THE HOLIDAY EFFECT ON THE OSLO STOCK EXCHANGE

WHEN ANALYSTS LEAVE FOR THE HOLIDAYS

TICKET NUMBER 1171735204753

A STAR ALLIANCE MEMBER

Abstract

We use seasonality in market participation associated with Norwegian holidays as a source of exogenous variation to study the *holidays effect* on the relative bid-ask spread and trading volume on the Oslo Stock Exchange (OSE). Outbound air-travel serves as a proxy for holiday intensity. We study daily data for a balanced sample of 90 small-, mid- and large-cap stocks from 2014 to 2019 and use 2020 to 2024 as an out-of-sample period. Trading volume and the relative bid-ask spread, estimated from daily high-low prices, are compared between defined holiday windows and the rest of the year.

We examined the five trading days before and after Christmas and Easter and found little robust evidence that either the trading volume or the bid-ask spread differ from normal trading days. In contrast, the collective summer holiday shows a clear and consistent pattern. The mean trading volume is significantly lower across all cap. groups, while the mean relative bid-ask spread is also significantly narrower with a 14.03% decrease.

This combination of lower trading activity and tighter spreads contrasts with earlier studies done on the OSE. Interpreted within standard information-based models of market making, these results are consistent with an interpretation in which adverse selection risk is temporarily lower during periods *when analysts leave for the holidays*.

To evaluate the economic significance of these spread changes, we construct a ΔITC (implicit transaction cost) forecasting model that applies a simple shrinkage parameter to scale down the in-sample holiday effect for out-of-sample application. Using this framework, we design and backtest a timing strategy that shifts trades to the collective summer holiday. The backtest suggests that expected round-trip implicit transaction costs could be reduced by up to 14.42%.

Table of Contents

Abstract *2*

1. Introduction *7*

1.1. Background and Motivation **7**

1.2. The "*holiday effect*" **7**

1.3. Contributions **7**

1.4. Practical Relevance **8**

1.5. Definitions of Key Terms **8**

2. Theory and Literature *8*

2.1. Theoretical Background **8**

2.2. Establishing the Proxy for holiday intensity **9**

2.3. Literature Review **10**

2.4. Hypothesis Development **11**

Trading volume 11

Bid-ask spread 11

Choice between one- and two-sided tests 11

Trading volume: two-sided tests 12

Bid-ask spread: two-sided tests 12

3. Data and Methodology *12*

3.1. Data collection, sample and holiday windows **12**

3.2. Spread Estimation **13**

Corwin-Schultz (2012): Spread Estimator 13

Why we use the CS method in our analysis 14

Abdi and Ranaldo (2017): Spread Estimator 14

Why Combining AR and CS is beneficial 14

3.3. Statistical Methods **15**

Welch t-test 15

FDR (Benjamin-Hochberg) 15

3.4. Regression Model **17**

OLS 17

Fixed-effects panel regression 18
Subsample robustness check (by year) 18
Placebo test (permutation-based) 19

4. Empirical Results 20

4.1. Trading Volume 20

P-Values for Volume 20
Increase or Decrease 21
Conclusion volume 22

4.2. Bid-Ask Spread 23

P-values for Spread 23
Increase or Decrease 26
Conclusion Spread 27

4.3. Key Findings 28

4.4. The Empirical "paradox", lower volume and narrower spread 28

Interpretation 1 - Who takes holiday, retail investors or informed traders? 29
Interpretation 2 - Market structure 30
Interpretation 3 - Less available information: 31
Interpretation 4 - Can Norway's macroeconomic stability contribute to lower spreads in the summer? 31

5. Implicit transaction cost model 32

5.1. Establishment of the implicit transaction cost equation 32

5.2. Weaknesses of the implicit transaction cost equation 33

5.3. Predictability of the future 34

Unbiased estimator 34

5.4. The uncertainty in the values of ΔSi - 95 CI 36

5.5. *Shrinkage parameter* p 37

5.6. Plot of the forecast model 39

5.7. Recommendation 41

5.8. Backtesting 42

6. Conclusion 45

6.1. Summary 45

6.2. Limitations with our analysis 46

7. References 47

8. Appendix 49

8.1. A - Complete hypothesis: 49

Traded volume - small cap. 49

Spread - small cap. 50

Traded volume - mid. cap. 51

Spread - mid. cap. 52

Traded volume - large cap. 53

Spread - large cap 54

8.2. B - Complete hypothesis for Christmas: 55

8.3. C - Python code: 56

Table of Figures

Figure 1 Average monthly passengers per airport (2014-2019) 10

Figure 2 P-values for volume (vs. control) 16

Figure 3 Stability check summer effect year-by-year (in-sample) 19

Figure 4 P-values for volume (vs.control) 20

Figure 5 Mean daily volume (Christmas and easter period) 21

Figure 6 Mean daily volume (summer holiday) 21

Figure 7 P-values for spread (vs.control) 23

Figure 8 Mean spread (christmas and easter periods) 26

Figure 9 Mean spread (summer holiday) 26

Figure 10: Forecast model 39

Figure 11: Forecast model, k=0.95 40

Figure 12: Liquidity cost analysis.....42

Figure 13: P-values out-of-sample.....44

Table of Tables

Table 1 Spread determinants with control variables (Full Sample).....17

Table 2 Panel regression with entity fixed effects 18

Table 3 Descriptive statistics 35

1. Introduction

1.1. Background and Motivation

We study how major Norwegian holiday periods affect trading volume and the and the relative bid-ask spread on the Oslo Stock Exchange (OSE). The analysis covers daily data for a balanced sample of 90 small-, mid- and large-cap stocks across defined holiday windows around Easter, Christmas and the July collective vacation from 2014 to 2019. As a proxy for holiday intensity, we use the number of passengers on outbound scheduled and non-scheduled flights from all Norwegian airports.

1.2. The "*holiday effect*"

Those who trade on or otherwise influence the OSE, including informed traders, corporate decision makers, uninformed traders and politicians, are drawn from the same population that takes holidays and generates outbound air travel. This makes the holiday-intensity proxy naturally linked to individuals who monitor, participate in and influence the market.

Importantly, we do not observe and do not assume that the probability of temporary withdrawal from the market is uniformly distributed across market participants. Informed and uninformed traders, corporate decision makers, politicians and other agents who influence the market may respond to holiday periods in different ways. Some are more likely to go on holiday and withdraw from the market altogether. Others may remain active but scale back their monitoring and trading. Retail investors, on the other hand, may even trade more if they have more time available.

As outbound air travel rises, both the number and the composition of market participants may therefore change in ways that are a priori ambiguous. In our empirical analysis, we therefore study how holiday-related changes in both the number and the composition of all those who participate in and influence trading on the Oslo Stock Exchange are reflected in trading volume and relative bid-ask spreads. We refer to any systematic response of these variables to holiday intensity as a *holiday effect*.

1.3. Contributions

The contribution of this paper is twofold. First, we provide new evidence on whether Norwegian holiday periods are associated with systematic patterns in trading volume and

relative bid-ask spreads. Second, we interpret the potential volume and spread patterns through the lens of modern market microstructure theory and relate them to prior evidence on *holiday effects* on the Oslo Stock Exchange.

1.4. Practical Relevance

To make the results operational for practitioners, we finally embed the estimated spread differences in an implicit transaction cost framework and an out-of-sample execution-timing backtest from 2020 to 2024, thereby linking a primarily academic microstructure question to concrete implications for order placement and trade timing on the Oslo Stock Exchange.

1.5. Definitions of Key Terms

Spread

The term *spread* refers to the relative bid-ask spread, calculated as the difference between the bid and the ask divided by the midprice of the asset. All spread estimations in the analysis are based on daily high-low price data processed through the Corwin-Schultz (2012) estimator unless stated otherwise.

Average / mean

Throughout the paper, the terms *average* and *mean* are used interchangeably to refer to the arithmetic mean of a variable. This applies to both the average spread and the average trade volume.

Summer holiday / Collective Summer Holiday

The terms *summer holiday* and *collective summer holiday* are used interchangeably to refer to the three-week coordinated Norwegian vacation period, corresponding to weeks 28, 29, and 30.

2. Theory and Literature

2.1. Theoretical Background

Liquidity, volatility, spread and trading volume:

“Liquidity is the ability to trade large size quickly, at low cost, when you want to trade” (Harris, 2003, p. 407). A market is dependent on liquidity to function well. Everyone benefits from liquidity, because it allows market participants to implement their strategies cheaply.

Liquidity is heavily correlated with volatility. Markets with high liquidity tend to be less volatile than illiquid ones, because large orders have less impact on the price when the limit-order book is deep. The spread is also affected. When liquidity is high and many market makers compete for order flow, they must quote more aggressively, posting higher bids and lower asks to attract trades and earn the spread, which tightens spreads and, together with the deeper book, reduces volatility.

Trading volume is often positively related to liquidity, but the relationship can break down in periods of high information asymmetry or market stress. In such periods, trading volume can rise even though liquidity weakens.

In an illiquid market the spread tends to widen, volatility increases, and trading volume reduces.

How does adverse selection affect the spread?

The losses caused by unequal information in a market is by economists called adverse selection losses, which reflect "well-informed traders' ability to select which side of the market to be on and because their selection is adverse to the welfare of other traders" (Harris, 2015, p. 23). They should therefore execute profitable trades at the expense of other market participants with less information.

"Dealers are especially vulnerable to informed traders because dealers are on the wrong side of the market when well-informed traders trade with them" (Harris, 2015, p. 24). The dealers make losses if they cannot offset their trades before the prices change in the way that the well-informed traders expect.

The dealers compensate for these losses by widening their spreads, "so that they recover from all traders what they lose on average to well-informed traders" (Harris, 2015, p. 24). This type of dealer-behaviour is by economists often referred to as making an adverse selection spread. The size of the adverse selection spread is determined by how frequently dealers incur losses and by the average amount they lose when trading with well-informed traders.

2.2. Establishing the Proxy for holiday intensity

Holiday intensity is proxied by the number of passengers on outbound scheduled and non-scheduled flights from all Norwegian airports. The predefined holiday windows around Easter, Christmas and the July collective vacation can be viewed as sources of exogenous calendar

variation for the OSE, while air-travel intensity adds information about whether these calendar dates coincide with substantial holiday activity.

Holiday periods can plausibly affect different groups of market participants in different ways. Holiday periods may lead analysts and institutional traders to scale back their activity, reducing informed participation, while at the same time giving retail investors more time to trade. We therefore do not interpret an increase in the number of passengers as a direct proxy for a fall in total market participation. Instead, we treat it as a measure of the intensity of each holiday period, and use it as a parameter for the potential magnitude of holiday-related changes in the size and composition of those who monitor, trade on and otherwise influence the Oslo Stock Exchange.

Figure 1 shows that the holiday intensity is strongest during June and the broader summer months. This indicates that, compared with the rest of the year, this period may exhibit the most pronounced holiday effect in terms of significant changes in the bid-ask spread and trading volume.

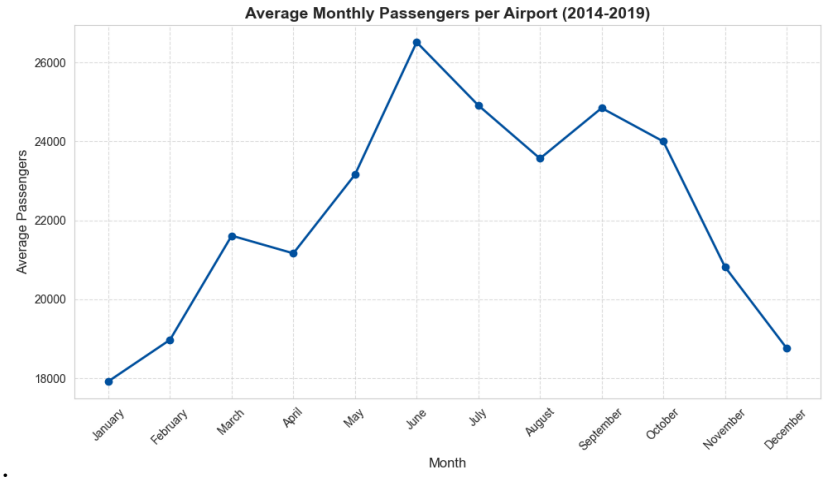


Figure 1: Average monthly passengers per airport (2014-2019)

(Statistics Norway, 2025)

2.3. Literature Review

Hong & Yu (2009) research "Gone fishin': Seasonality in trading activity and asset prices" provided one of the main empirical examinations of seasonal patterns in global equity markets, while documenting stable and reoccurring decline in both trading activity and market returns during summer months compared to the rest of the year. Across the 51 countries in their sample,

mean turnover in the summer is 7.9% lower (t-stat = 3.34), while the average bid-ask spread is 3.4% higher (t-stat = 2.24).

For the Oslo Stock Exchange (OSE), their results are broadly in line with the cross-country mean, but the holiday effect appears even more pronounced. On OSE, the mean relative bid-ask spread increases by 11% (t-stat = 5.78), while trading volume declines by 24.6% (t-stat = 9.13) during the summer months. It is therefore reasonable to conclude that the *holiday effect* is stronger on OSE than in the average market in their sample.

2.4. Hypothesis Development

The hypotheses below are stated generic form. The complete set of hypotheses is then given in Appendix A and B, where they are broken down by: Pre-Easter, post-Easter, pre-Christmas, post-Christmas, Christmas, the summer months, and the collective summer holiday. They are also divided into three market-capitalisation segments: large cap, mid cap, and small cap. In the results tables we therefore report separate p-values for large-, mid- and small-cap stocks across all different holidays.

Trading volume

Trading volume measured in shares traded

$$H_0: \mu_{Mean\ volume\ \text{holiday}} = \mu_{Mean\ volume\ rest\ of\ year}$$

$$H_A: \mu_{Mean\ volume\ \text{holiday}} \neq \mu_{Mean\ volume\ rest\ of\ year}$$

Bid-ask spread

$$H_A: \mu_{Mean\ relative\ bid-ask\ spread\ holiday.} \neq \mu_{Mean\ relative\ bid-ask\ spread\ rest\ of\ year..}$$

Choice between one- and two-sided tests

Statistically, the difference between one-sided and two-sided t-tests is how the significance level α is placed in the t-distribution and what you assume about the sign of the effect. With a given α (e.g. 0.05), a one-sided test puts all of α in one tail and rejects H₀ only if $T \geq t_{\alpha}^{v}$ or

when $T \leq -t_{\alpha}^{v}$. This lowers the critical value in absolute terms and increases power in that direction. A two-sided test splits α across both tails and rejects when $|T| \geq t_{\alpha/2}^{v}$, so the statistic must be more extreme. The benefit is that whit a two-sided test you do not have to assume in advance whether the holiday effect is positive or negative. The choice therefore depends on how strong the prior theory or evidence is about the direction of the effect, and on whether deviations in both directions are of interest.

Trading volume: two-sided tests

For trading volume, the situation is not clear-cut. A common narrative, seen for example in Hong and Yu's study, is that the turnover hypothesis simply "confirm a widely held belief that stock turnover is significantly lower during the summer because market participants are on vacation" (Hong & Yu, 2009). It could therefore be argued that it may be wise to apply a one-sided test to the collective summer holiday. However, this relationship is not deterministic. A small number of large or particularly active traders can generate a substantial amount of volume, and retail participation may respond differently to holidays than institutional participation. In other words, a possible reduction in market participants, as seen through a possible increase in travel, does not necessarily imply lower traded volume. We therefore apply a two-sided test.

Bid-ask spread: two-sided tests

The case for two-sided testing is even stronger for the bid-ask spread. Microstructure theory emphasises multiple factors contributing to the size of the spread like adverse selection and volatility and it could be stated that the response to the holiday is ex ante ambiguous.

3. Data and Methodology

3.1. Data collection, sample and holiday windows

Our empirical analysis uses daily data for 90 stocks on the Oslo Stock Exchange, split equally into small-, mid- and large-cap segments. The in-sample period runs from 1 January 2014 to 31 December 2019, while 1 January 2020 to 31 December 2024 is reserved for backtesting. Daily OHLC and volume data are downloaded with yfinance and merged into a balanced panel.

We define three main holiday blocks. The collective summer holiday is weeks 28-30, when a large share of Norwegian employees takes their main vacation. This three-week block is not fixed by statute but coordinated through collective agreements and long-standing workplace practices. Under the Norwegian Holiday Act section 7(1) (Ferieloven, 1988), employees are entitled to a main holiday of 18 working days to be taken between 1 June and 30 September. Together, these rules and traditions create a highly concentrated vacation period in July, making the Oslo Stock Exchange a useful setting for studying how a collective shutdown affects market conditions.

For Christmas/New Year, the exchange is typically closed on 24, 25, 26 and 31 December. We therefore analyse the five trading days before 24 December and the five trading days after 26 December; the core holiday dates themselves cannot be observed. During Easter, the exchange is closed on Maundy Thursday, Good Friday and Easter Monday. The Easter window is defined as the five trading days before Maundy Thursday and the five after Easter Monday. All remaining trading days form the control period, and for some analyses we also include a "summer_excl_holiday" window covering the rest of the summer weeks.

3.2. Spread Estimation

Given these holiday windows, we measure trading costs using daily bid-ask spread estimators. Bid-ask spreads are estimated with the Corwin-Schultz high-low estimator (Corwin & Schultz, 2012) and cross-checked with the Abdi-Ranaldo measure (Abdi & Ranaldo, 2017). We remove infinitesimal spreads, classify each observation into the relevant holiday, cap. size and control group, and study differences in spreads and volume using Welch's t-tests with Benjamini-Hochberg false-discovery-rate corrections (Benjamini & Hochberg, 1995) supplemented by OLS and fixed-effects panel regressions with clustered standard errors.

Corwin-Schultz (2012): Spread Estimator

The Corwin-Schultz spread estimator is a commonly used method to estimate the bid-ask spread using daily high and low prices. The high-low ratio reflects both the stocks' variance and its bid ask spread (Corwin & Schultz, 2012). It isolates the portion that reflects the effective bid-ask spread, by comparing the price ranges across two days and separating the volatility component from the range component. The CS method has become a standard tool for daily spread estimation in market microstructure, because it is easy to implement, and empirically

robust in different market conditions. It is as precise in liquid markets as it is in illiquid markets and is referred to as the "gold standard" method for daily spread-estimation.

Why we use the CS method in our analysis

We employ the CS estimator as our primary spread measure because it offers a well-established and empirically validated proxy for transaction costs based on data that are consistently available across assets and over time. In contrast to many alternative spread measures, the CS estimator does not rely on tick-by-tick quote and trade data, which are typically associated with high-frequency trading, resource-intensive to store and process, and in many cases difficult to obtain or incomplete for long historical samples. This makes the CS estimator well suited to our setting.

Abdi and Ranaldo (2017): Spread Estimator

The Abdi-Ranaldo method builds on the idea that the daily price range and closing price contain information about the true price process and trading frictions caused by the bid-ask spread. The AR method utilizes a wider information set than other low-frequency estimates like "readily available close, high, and low prices" (Abdi & Ranaldo, 2017). The estimator corrects the biases found in earlier spread estimators, including the Corwin-Schultz method, by modelling how volatility and noise affect the prices. In addition, the AR method delivers the most accurate estimates for less liquid stocks. The estimator is robust to volatility clustering, jumps and price movements overnight.

Why Combining AR and CS is beneficial

The CS method is considered a benchmark for daily spread estimation but can in some cases be sensitive to high volatility, and to assets with very wide or narrow ranges. The AR estimator can in those cases be used as a good complement, because it addresses many of the weaknesses with the CS method. If both methods give similar results, it strengthens the confidence that the estimated spreads reflect actual market frictions rather than estimator-specific bias. Combining the CS and the AR method strengthens the empirical credibility and relatability of the results.

3.3. Statistical Methods

Welch t-test

To compare the mean in the holiday periods with the rest of the year, we use Welch's two-sample t-test. This is a variant of the standard two-sample t-test that does not assume equal variances in the two groups and is therefore better suited when the variance may differ across periods and size group. The test statistic is defined as

$$t = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}}$$

When $\bar{X}_1$and $\bar{X}_2$ are the means in the holiday window and control period respectively, s_1^2 and s_2^2 are the sample variances, and n_1, n_2 are the number of observations in each group. Under the null hypothesis of equal means, this statistic is approximately t-distributed with degrees of freedom given by the Welch-Satterthwaite approximation. The resulting p-values tell us whether the differences in means for volume and the Corwin-Schultz spread during the holiday period and the rest of the year are statistically significant and are later adjusted using FDR.

FDR (Benjamin-Hochberg)

What is FDR correction?

When we test many hypotheses at the same time, some will turn out "significant" purely by chance. Benjamini and Hochberg (1995) therefore propose controlling the false discovery rate (FDR), defined as the expected proportion of false rejections among the hypotheses declared significant. Instead of asking "what is the probability of at least one error" as Bonferroni does, we ask "If we find, say 10 effects, what share of these is likely to be false?" this approach allow us to retain more statistical power when we have many tests (Benjamin & Hochberg, 1995, p. 289-300).

How we computed FDR in this paper

In the paper we first estimate a welch t-test for every combination of holiday window (before, during and after Christmas, Easter and the summer vacation), size group (small, mid and large cap), and variable (trading volume and Corwin-Shultz spread). This produces a set of p-values that we treat as one family of tests. We then apply the Benjamini - Hochberg procedure. The p-

values are sorted from smallest to largest, and we find the last p-value in the sequence that is "small enough" relative to its position, that is, the one that satisfies the BH inequality:

$$p_k \leq \frac{k}{m} q, \text{ with q=0.05.}$$

Here, p_k denotes the p-value with rank k *in* the ordered sequence (the k-th smallest p-value), k is its rank index, m is the total number of tests, and q is the chosen false discovery rate level. All tests with p-values lower than or equal to this threshold are classified as significant. In the p-value heatmaps, these are exactly the cells marked as significant after FDR correction, so the figures visually show which holiday effects "survive" once we adjust for multiple testing (Benjamini & Hochberg, 1995, p. 289-300).

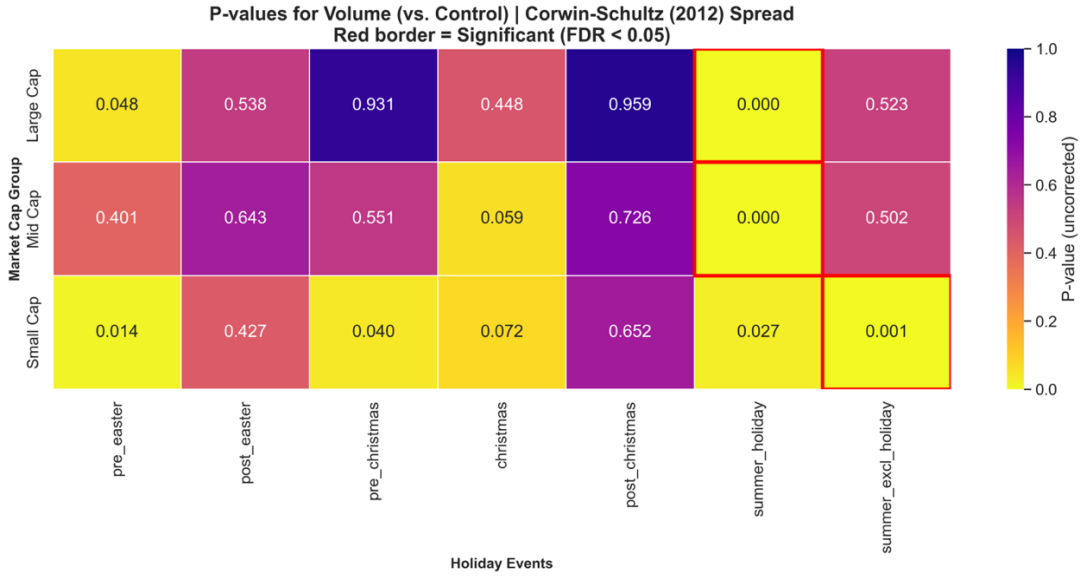


Figure 2: P-values for volume (vs. control)

3.4. Regression Model

OLS

We estimate linear OLS models where the dependent variable is the Corwin-Schultz spread. As explanatory variables, we include holiday dummies, lagged log volume, realized volatility, and size dummies for large- and mid-cap stocks (with small cap as the reference category). Standard errors are clustered at the stock level. The result in table 1 show that the summer dummy is clearly negative and highly significant, volatility has a positive and significant effect, whereas the volume coefficient is very small and statistically insignificant (p=0.938).

Spread Determinants with Control Variables (Full Sample)
Dependent Variable: Corwin-Schultz (2012) Spread | Controls: Lagged Volume, Volatility, Market Cap

OLS Regression Results

Dep. Variable:	Spread	R-squared:	0.126
Model:	OLS	Adj. R-squared:	0.125
Method:	Least Squares	F-statistic:	9.998
Date:	Wed, 03 Dec 2025	Prob (F-statistic):	2.19e-10
Time:	15:35:31	Log-Likelihood:	1.5030e+05
No. Observations:	53300	AIC:	-3.006e+05
Df Residuals:	53289	BIC:	-3.005e+05
Df Model:	10		
Covariance Type:	cluster		

	coef	std err	z	P>|z|	[0.025	0.975]
const	0.0142	0.003	4.816	0.000	0.008	0.020
Pre_Easter	-9.516e-05	0.000	-0.288	0.773	-0.001	0.001
Post_Easter	0.0004	0.000	0.986	0.324	-0.000	0.001
Christmas	0.0021	0.001	3.870	0.000	0.001	0.003
Pre_Christmas	0.0012	0.001	1.642	0.101	-0.000	0.003
Post_Christmas	0.0021	0.001	4.013	0.000	0.001	0.003
Summer	-0.0016	0.000	-4.856	0.000	-0.002	-0.001
Log_Volume_lag1	5.363e-06	0.000	0.036	0.971	-0.000	0.000
Volatility	0.1091	0.043	2.559	0.010	0.026	0.193
Large_Cap	-0.0059	0.002	-3.621	0.000	-0.009	-0.003
Mid_Cap	-0.0021	0.002	-1.163	0.245	-0.006	0.001

Omnibus:	36097.629	Durbin-Watson:	1.301
Prob(Omnibus):	0.000	Jarque-Bera (JB):	1611234.300
Skew:	2.706	Prob(JB):	0.00
Kurtosis:	29.386	Cond. No.	293.

Notes:
[1] Standard Errors are robust to cluster correlation (cluster)

Table 1: Spread determinants with control variables (Full Sample)

$$\text{Spread} = \text{constant} + \beta\text{Summer} + \gamma \cdot \text{Log_Volume_lag1} + \delta \cdot \text{Volatility}$$

γ = *how much the spread changes when log (yesterday's volume) increases by 1 unit, holding everything else constant*

δ = *how much the spread changes when volatility increases by 1 unit, holding everything else constant*

Fixed-effects panel regression

In addition to OLS, we use fixed-effects panel regression where the dependent variable is the daily relative Corwin Shultz spread. This allows us to exploit the fact that we observe many days per stock, and we let each stock have its own fixed "baseline level" of the spread. In this way we control for stable differences between firms, such as industry and size category. As explanatory variables, we use the same holiday dummies as in OLS regression. Standard errors are clustered at the stock level. The panel results confirm the OLS findings, the summer dummy is significantly negative, meaning the same stock trades with narrower spreads during the summer holiday than on other days. At the same time, we find that the spread increases around Christmas, and that both lagged volume and volatility are positively related to the spread, consistent with higher activity and risk being associated with somewhat worse liquidity.

Panel Regression with Entity Fixed Effects (In-Sample)
Dependent Variable: Corwin-Schultz (2012) Spread

PanelOLS Estimation Summary

Dep. Variable:	Spread	R-squared:	0.0416
Estimator:	PanelOLS	R-squared (Between):	0.6479
No. Observations:	53300	R-squared (Within):	0.0416
Date:	Wed, Dec 03 2025	R-squared (Overall):	0.3915
Time:	15:35:32	Log-likelihood	1.543e+05
Cov. Estimator:	Clustered		
		F-statistic:	288.43
Entities:	75	P-value	0.0000
Avg Obs:	710.67	Distribution:	F(8,53217)
Min Obs:	10.0000		
Max Obs:	1029.0	F-statistic (robust):	17.519
		P-value	0.0000
Time periods:	1486	Distribution:	F(8,53217)
Avg Obs:	35.868		
Min Obs:	1.0000		
Max Obs:	58.000		

Parameter Estimates

	Parameter	Std. Err.	T-stat	P-value	Lower CI	Upper CI
Pre_Easter	-0.0002	0.0003	-0.5257	0.5991	-0.0008	0.0004
Post_Easter	0.0004	0.0004	0.9661	0.3340	-0.0004	0.0011
Christmas	0.0019	0.0004	4.6413	0.0000	0.0011	0.0028
Pre_Christmas	0.0012	0.0006	1.9675	0.0491	4.52e-06	0.0024
Post_Christmas	0.0023	0.0005	4.6614	0.0000	0.0014	0.0033
Summer	-0.0014	0.0003	-4.5561	0.0000	-0.0020	-0.0008
Log_Volume_lag1	0.0016	0.0004	4.3470	0.0000	0.0009	0.0024
Volatility	0.0542	0.0275	1.9689	0.0490	0.0002	0.1082

F-test for Poolability: 137.62
P-value: 0.0000
Distribution: F(74,53217)

Included effects: Entity

Table 2: Panel regression with entity fixed effects

Subsample robustness check (by year)

To examine how stable the holiday effect is over time, we estimate the same regression model separately for each calendar year in the sample. For each year we extract the coefficient on the summer dummy together with its 95% confidence interval, and plot these as a simple stability check. If the effect were driven by a single "special" year, the coefficients would fluctuate strongly and often lie close to zero. In our results, the summer coefficient is consistently

negative across years and often statistically significant, suggesting that lower spreads during the coordinated summer holiday represent a persistent pattern rather than a one-off outcome.

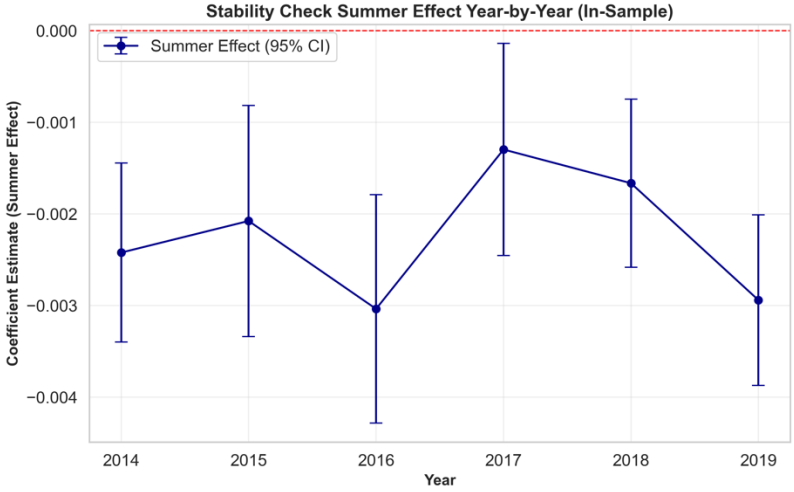


Figure 3: Stability check summer effect year-by-year (in-sample)

Placebo test (permutation-based)

As a robustness check, we conduct a permutation-based placebo test. We keep spreads, volume and volatility unchanged, but randomise which days are labelled as summer holiday, and re-estimate the same regression model 100 times. This generates an empirical null distribution for the summer coefficient when the holiday dates are purely random. In our figures, the placebo estimates are tightly clustered around zero, while the actual summer coefficient is much more negative and lies in the left tail, suggesting that the holiday effect we find is unlikely to be driven by random calendar patterns.

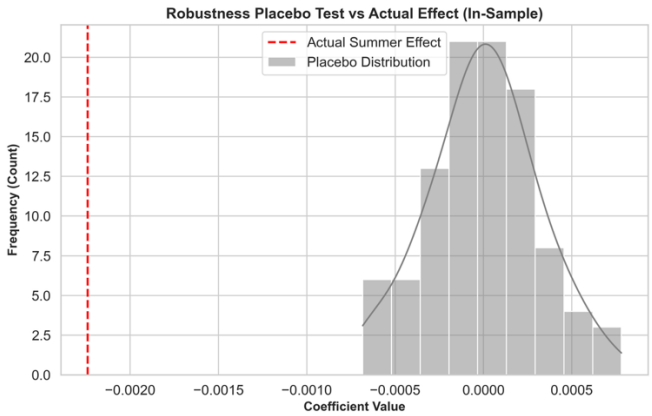


Figure 4: Robustness Placebo Test vs. Actual Effect (In-Sample)

4. Empirical Results

4.1. Trading Volume

P-Values for Volume

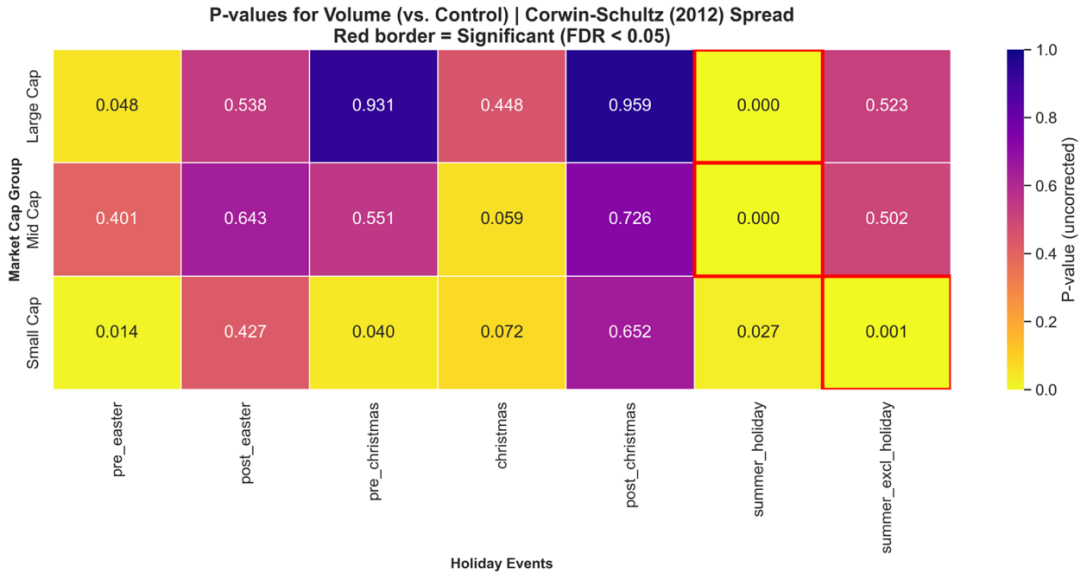


Figure 5: P-values for volume (vs.control)

Summer holiday:

For the in-sample period 2014-2019, the collective summer holiday is statistically significant at the $p < 0.05$ (*) level before applying the FDR correction. When we look across market-cap groups, the effect is particularly strong for large- and mid-cap stocks, which are significant at the $p < 0.001$ (***) level. After the FDR adjustment, only the mid- and large-cap. groups remain statistically significant.

Pre easter:

We also note that the pre-Easter effect yields p-values below 0.05 for both large- and small-cap stocks, but these coefficients are no longer regarded as statistically significant once the FDR adjustment is applied.

Summer excl. holiday:

We also note that the "summer_excl_holliday" period is significant at the $p<0.001$ (***) level for small caps.

The rejected null hypothesis at the 5% significance level (a = 0.05, FDR-adjusted p < 0.05)

$$9\ H_0: \mu_{Mean\ summer\ months\ volume\ small\ cap} = \mu_{Mean\ volume\ rest\ of\ year\ small\ cap.}$$

$$35\ H_0: \mu_{Mean\ volume\ collective\ summer\ holiday\ Mid.cap.} = \mu_{Mean\ volume\ rest\ of\ year\ Mid.cap.}$$

$$59\ H_0: \mu_{Mean\ volume\ collective\ summer\ holiday\ large\ cap.} = \mu_{Mean\ volume\ rest\ of\ year\ large\ cap.}$$

Confirming their respective alternative hypotheses:

$$10\ H_A: \mu_{Mean\ summer\ months\ volume\ small\ cap} \neq \mu_{Mean\ volume\ rest\ of\ year\ small\ cap}$$

$$36\ H_A: \mu_{Mean\ volume\ collective\ summer\ holiday\ Mid.cap.} \neq \mu_{Mean\ volume\ rest\ of\ year\ Mid.cap.}$$

$$60\ H_A: \mu_{Mean\ volume\ collective\ summer\ holiday\ large\ cap.} \neq \mu_{Mean\ volume\ rest\ of\ year\ large\ cap.}$$

(Appendix A)

Increase or Decrease

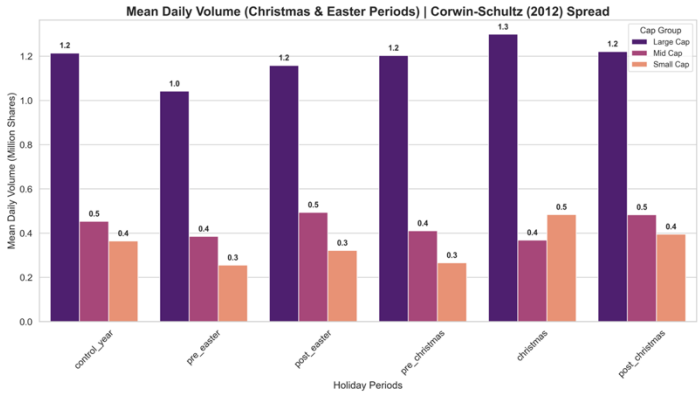


Figure 6: Mean daily volume (Christmas and easter period)

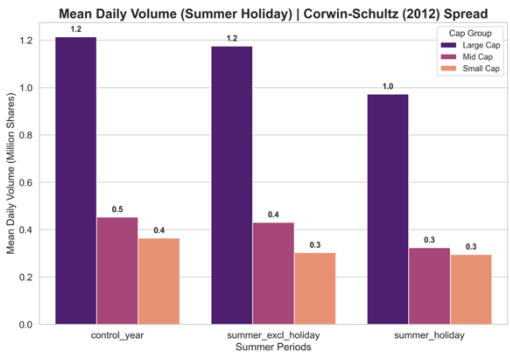


Figure 7: Mean daily volume (summer holiday)

For the null hypotheses 9, 35, 59 (Appendix A), the p-values are below 0.05, so we reject the null hypotheses of equal means at the FDR adjusted 5% significance level. This provides evidence that the expected value for the volume during these holiday windows differ from those in the rest of the year. Because the tests are two-sided, they do not impose a direction on the effect. The direction is obtained by comparing the corresponding sample means. We therefore inspect the sample mean of number of traded shares for each significant finding to determine whether the significant p-values reflect an increase or a decrease in trading activity.

Hypotheses 9:

From the chart above we see that the null hypotheses 9 was rejected due to a significant **decrease** in traded shares during the summer months in small caps.

Hypotheses 35:

We also see that the null hypotheses 35 was rejected due to a significant **decrease** in traded shares during the collective summer holiday in mid. caps.

Hypotheses 59:

We also see that the null hypotheses 59 was rejected due to a significant **decrease** in traded shares during the collective summer holiday in large caps.

Conclusion volume

The key finding is that during the collective summer holiday, all three market-capitalisation groups show significant reductions in traded volume at the 5% significance level before applying the FDR correction. After controlling for multiple testing using the FDR procedure, the large-cap and mid-cap groups remain statistically significant, while the small-cap result is no longer significant.

This still provides evidence of a reduction in trading volume during the collective summer holiday. Our proxy for market participation, the number of passengers on outbound flights, appears to capture a genuine decline in market activity during the collective summer holiday, as the increase in number of passengers in June (figure 1) coincides with the observed reduction in trading volume. The significant reduction in the traded volume during the summer holiday is consistent with the presence of a *holiday effect*. At this stage, however, it remains unclear which groups of traders who are most affected by this. It is plausible that holiday-related withdrawals are not evenly distributed across trader types, so the holiday effect may involve not only a reduction in overall trading activity but also a change in the composition of market participants. We therefore turn to bid-ask spreads, which can provide indirect evidence on whether such compositional shifts are present.

4.2. Bid-Ask Spread

P-values for Spread

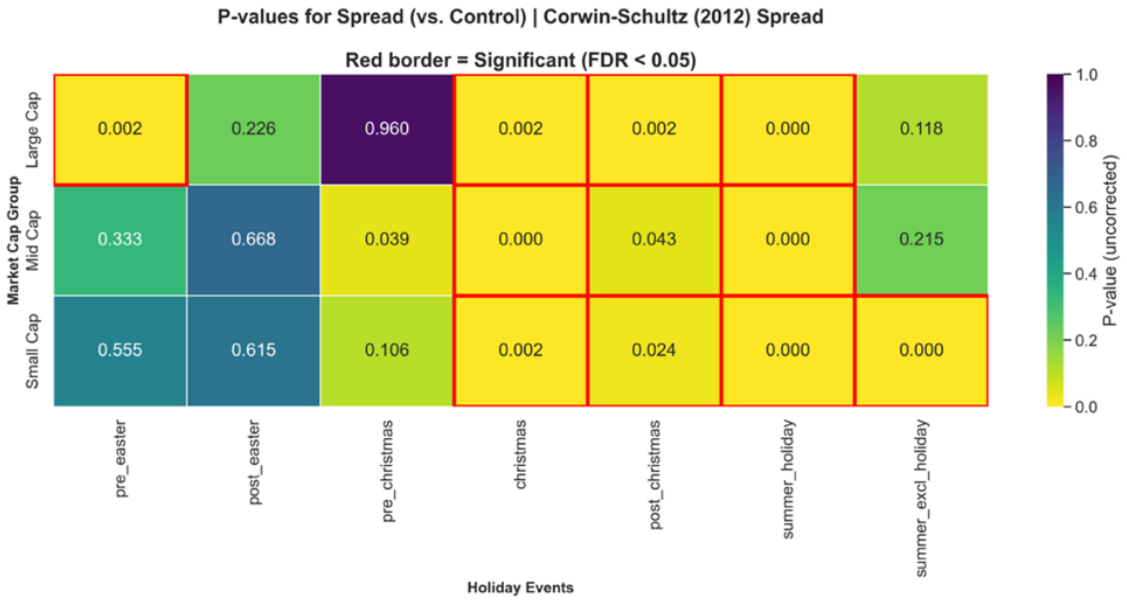


Figure 8: P-values for spread (vs.control)

Summer holiday:

For the in-sample period 2014-2019, the collective summer holiday exhibits highly significant reductions in the estimated Corwin-Schultz spread, with p-values below 0.001 for all three market-capitalisation groups. After applying the FDR correction, all three groups remain statistically significant, indicating that the summer-holiday narrowing of spreads is a robust and broad-based phenomenon across the Oslo Stock Exchange.

The rejected null hypothesis at the 5% significance level (a = 0.05, FDR-adjusted p < 0.05)

$$47\ H_0: \mu_{Mean\ spread\ collective\ summer\ holiday\ Mid.cap.} = \mu_{Mean\ spread\ rest\ of\ year\ Mid.cap.}$$

$$71\ H_0: \mu_{Mean\ spread\ collective\ summer\ holiday\ large\ cap.} = \mu_{Mean\ spread\ rest\ of\ year\ large\ cap.}$$

$$23\ H_0: \mu_{Mean\ spread\ collective\ summer\ holiday\ small\ cap} = \mu_{Mean\ spread\ rest\ of\ year\ small\ cap}$$

Confirming their respective alternative hypotheses:

$$48\ H_A: \mu_{Mean\ spread\ collective\ summer\ holiday\ Mid.cap.} \neq \mu_{Mean\ spread\ rest\ of\ year\ Mid.cap.}$$

$$72\ H_A: \mu_{Mean\ spread\ collective\ summer\ holiday\ large\ cap.} \neq \mu_{Mean\ spread\ rest\ of\ year\ large\ cap.}$$

$$24\ H_A: \mu_{Mean\ spread\ collective\ summer\ holiday\ small\ cap} \neq \mu_{Mean\ spread\ rest\ of\ year\ small\ cap}$$

(Appendix A)

Pre easter:

Before FDR correction, the pre-Easter window shows a statistically significant reduction in spreads for large-cap stocks (p = 0.002). Mid-cap and small-cap stocks do not cross the 5% threshold. The significance of the large cap observation are still significant after the FDR correction.

The rejected null hypothesis at the 5% significance level (a = 0.05, FDR-adjusted p < 0.05)

$$61\ H_0: \mu_{Mean\ pre\ easter\ spread\ large\ cap.} = \mu_{Mean\ spread\ rest\ of\ year\ large\ cap.}$$

Confirming the respective alternative hypotheses:

$$62\ H_A: \mu_{Mean\ pre\ easter\ spread\ large\ cap.} \neq \mu_{Mean\ spread\ rest\ of\ year\ large\ cap.}$$

(Appendix A)

Post Christmas:

The post-Christmas period also shows significance for all three cap groups. Only the small and large cap. groups are significant after the FDR correction.

The rejected null hypothesis at the 5% significance level (a = 0.05, FDR-adjusted p < 0.05)

$$67\ H_0: \mu_{Mean\ post\ Christmas\ spread\ large\ cap.} = \mu_{Mean\ spread\ rest\ of\ year\ large\ cap.}$$

$$19\ H_0: \mu_{Mean\ post\ Christmas\ spread\ small\ cap} = \mu_{Mean\ spread\ rest\ of\ year\ small\ cap}$$

Confirming their respective alternative hypotheses:

$$68\ H_A: \mu_{Mean\ post\ Christmas\ spread\ large\ cap.} \neq \mu_{Mean\ spread\ rest\ of\ year\ large\ cap.}$$

$$20\ H_A: \mu_{Mean\ post\ Christmas\ spread\ small\ cap} \neq \mu_{Mean\ spread\ rest\ of\ year\ small\ cap}$$

(Appendix A)

Christmas:

The Christmas period shows significance across all three cap sizes after the FDR correction.

The rejected null hypothesis at the 5% significance level (a = 0.05, FDR-adjusted p < 0.05)

$$79\ H_0: \mu_{\text{Mean } Christmas \text{ spread small cap.}} = \mu_{\text{Mean } Christmas \text{ spread small cap}}$$

$$81\ H_0: \mu_{\text{Mean } Christmas \text{ spread mid. cap.}} = \mu_{\text{Mean } Christmas \text{ spread mid. cap.}}$$

$$83\ H_0: \mu_{\text{Mean } Christmas \text{ spread large cap.}} = \mu_{\text{Mean } Christmas \text{ spread large cap.}}$$

Confirming their respective alternative hypotheses:

$$80\ H_A: \mu_{\text{Mean } Christmas \text{ spread small cap.}} \neq \mu_{\text{Mean } Christmas \text{ spread small cap}}$$

$$82\ H_A: \mu_{\text{Mean } Christmas \text{ spread mid. cap.}} \neq \mu_{\text{Mean } Christmas \text{ spread mid. cap.}}$$

$$84\ H_A: \mu_{\text{Mean } Christmas \text{ spread large cap.}} \neq \mu_{\text{Mean } Christmas \text{ spread large cap.}}$$

(Appendix B)

Summer excl. holiday:

Finally, the “summer_excl_holiday” window displays a statistically significant effect for the small-cap group, and this significance persists after the FDR adjustment.

The rejected null hypothesis at the 5% significance level (a = 0.05, FDR-adjusted p < 0.05)

$$21\ H_0: \mu_{Mean\ summer\ months\ spread\ small\ cap} = \mu_{Mean\ spread\ rest\ of\ year\ small\ cap}$$

Confirming their respective alternative hypotheses:

$$22\ H_A: \mu_{Mean\ summer\ months\ spraed\ small\ cap} \neq \mu_{Mean\ spread\ rest\ of\ year\ small\ cap}$$

(Appendix B)

Increase or Decrease

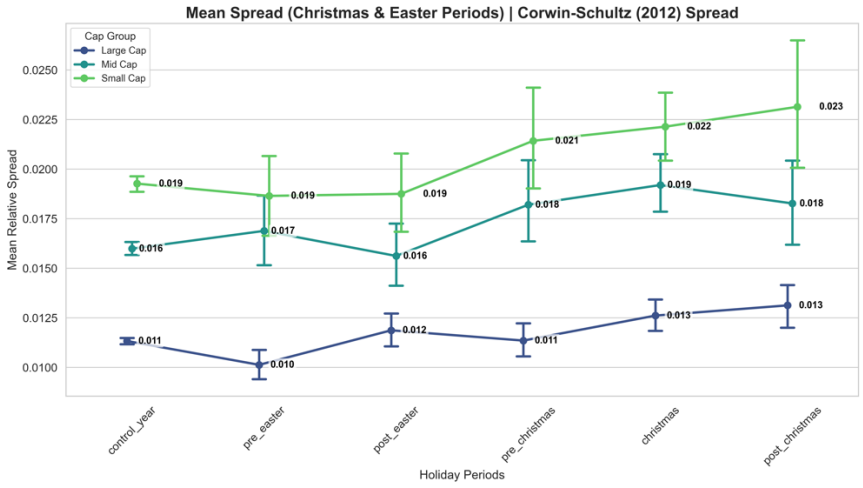


Figure 9: Mean spread (christmas and easter periods)

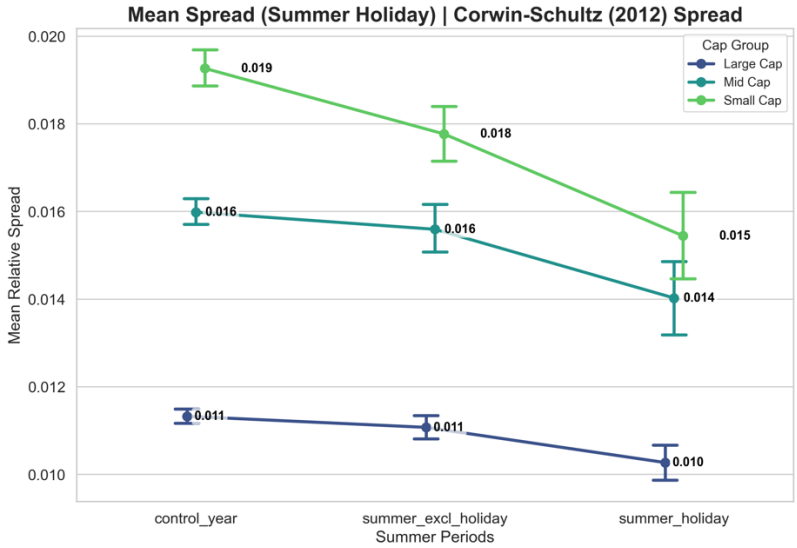


Figure 10: Mean spread (summer holiday)

All t-tests were two-sided. For null hypotheses 47, 71, 23, 61, 67, 19, 79, 81, 83 and 21 (Appendix A, Appendix B), the two-sided p-values are below 0.05 after the FDR correction, so we reject the null hypotheses of equal means at the 5% significance level. This provides evidence that the expected values for the relative bid-ask spread during these holiday windows differ from those on non-holiday trading days. Because the tests are two-sided, they do not impose a direction on the effect. The direction is obtained by comparing the corresponding sample means. We therefore inspect the sample mean relative spread for each significant finding to determine whether the significant p-values reflect an increase or a decrease in trading activity.

Summer holiday:

The rejection of null hypotheses 47, 23 and 71 (appendix A) reflects a **decrease** in spreads during the collective summer holiday across all market-capitalisation groups, as shown in the table above.

Pre-Easter:

The rejection of null hypothesis 61 (appendix A) is driven by a **decrease** in spreads for small caps. in the pre-Easter period, as shown in the table above.

Post-Christmas:

The rejection of null hypotheses 68 and 19 (appendix A) reflects an **increase** in spreads during the post-Christmas period for both large- and small-cap stocks.

Christmas:

The rejection of null hypotheses 79, 81 and 83 (appendix B) and reflects an **increase** in spreads during the Christmas period across all cap sizes.

Summer excl. holiday:

The rejection of null hypothesis 21 (appendix A) is driven by a **decrease** in spreads during the "summer_excl_holiday" period for small-cap stocks.

Conclusion Spread

The key finding for spreads is that during the periods "collective summer holiday" and "Christmas" we obtain FDR-adjusted p-values below 0.05 for all market-capitalisation groups.

For the Corwin-Schultz spread, the FDR corrections confirm that the most robust seasonal pattern is the reduction in spreads during the collective summer holiday. Several smaller effects like mid. cap. pre_christmas and post_christmas lose significance when we correct for multiple testing and are therefore interpreted more cautiously.

4.3. Key Findings

The central pattern is that, for the collective summer holiday, FDR-adjusted p-values fall below 0.05 for both traded volume and bid-ask spreads. The only exception is Hypothesis 11 (Appendix A), which relates to small-cap traded volume. For this hypothesis, the estimated effect is statistically significant at the p<0,05 (*) before the FDR adjustment, but after the adjustment the p-value exceeds 0.05 and the effect is no longer statistically significant.

Taken together, the results show that during the collective summer holiday trading volume decreases for all market-capitalisation groups except the small-cap volume case mentioned above, while bid-ask spreads simultaneously decline with FDR-adjusted p-values below 0.05. This combination of lower volume and narrower spreads contrasts with the conventional inverse relation between trading activity and bid-ask spreads documented in prior studies like “Gone fishin’: Seasonality in trading activity and asset prices” (Hong & Yu, 2009), and therefore warrants further discussion.

4.4. The Empirical “paradox”, lower volume and narrower spread

Discussion of how this “summer paradox” can be interpreted, considering market microstructure theory and changes in market structure based on Harris (2003)

The quoted bid-ask spread is usually viewed as the sum of three components. One component is pure order-processing costs, such as technology, trading systems and compensation to the dealer or broker. A second component is inventory risk, which arises from holding stock while prices may move. A third component is adverse selection, which reflects the risk of trading against better-informed investors. When both trading volume and participant activity decline, the theory implies that the spread should normally widen. There are fewer counterparties and weaker competition among liquidity providers it becomes more difficult to “wash out” unwanted positions, so that inventory risk increases, and informed orders gain relatively greater importance as trading noise diminishes. International evidence, such as Hong and Yu (2009),

shows that summer months are often characterised by lower turnover and wider spreads, which is commonly interpreted as evidence of weaker market quality. Our results on the OSE points in the opposite direction. Trading volume falls, but spreads narrow.

Interpretation 1 - Who takes holiday, retail investors or informed traders?

A key question in interpreting our results is who reduces activity during the main summer holiday. Is it primarily uninformed retail investors, more informed professional participants, or is the pattern driven by changes in broker behaviour?

Based on Harris (2003) and our empirical findings, the most plausible interpretation is that the relative share of informed and aggressive order flow declines during the summer weeks. In Harris's framework, spreads are largely driven by adverse selection: the greater the risk of facing better-informed counterparties, the higher the compensation demanded by liquidity providers. If it is mainly professional, event-driven and informed traders that scale down during the holiday period, the information risk per trade will decline even though total traded volume is reduced. Lower adverse selection risk affects the liquidity suppliers the most, because the probability of the market moving against their trade decreases. Dealers can thereby tighten their spreads to make larger profits.

This interpretation is supported by external evidence. Norwegian brokerages report that institutional investors and brokers reduce their activity from mid-June, indicating that professional desks do not operate at full capacity throughout the summer (Pareto Securities, 2025). As documented in media coverage, finance is an extremely high-intensity field in which professionals often work very long hours and are expected to be permanently available (Erikstad, 2021). When the most informed institutions then go on holiday simultaneously, the effect on the remaining market can therefore be substantial.

The mechanism can be framed more formally using the dealer model of Easley, Kiefer, O'Hara and Paperman (1996), as presented and simplified by Schenk-Hoppé (2025). In this model, the spread can be written as:

$$A - B = \frac{i}{i + u}[S_G - S_B]$$

Where i is the arrival probability of an informed trader, u is the arrival probability of an uninformed trader, and S_G and S_B denote the conditional values given good or bad news. The

fraction $\frac{i}{i+u}$ rises when informed trading becomes more likely and falls when uninformed trading becomes more likely.

Narrower spreads in the summer holiday are therefore consistent with a fall in i, a rise in u, or both. In probabilistic terms, a lower P(informed | trade) and a higher P(uninformed | trade). Liquidity providers widen quotes when they fear that counterparties may be informed (Harris, 2003, p. 287). Conversely, when they expect to trade mainly with "sitting ducks" who lack informational advantages, they can safely quote tighter spreads.

The simultaneous drop in trading volume and tightening of bid-ask spreads during the summer holiday is most consistent with a reduction in informed trading activity. The fact that overall volume falls speaks against an explanation based solely on an increase in uninformed retail trading. Instead, it suggests either that informed traders scale back their activity while retail participation remains roughly unchanged, or that any rise in retail trading is more than offset by a sizable decline in informed orders. In this sense, the holiday period resembles a quieter, lower-information environment in which market makers face less adverse selection risk from informed traders. In other words, a period: "When analysts leave for the holidays".

Interpretation 2 - Market structure

A key difference between Harris (2003) and the present setting is that the OSE now operates at a fully electronic limit order book where liquidity is provided primarily by algorithmic market makers, rather than by human dealers (Regulation Department, 2024). The market is subject to minimum tick-size constraints and intense competition to post the best bid and ask. In such an environment, realised volatility, rather than trading volume, is often the main determinant of how wide algorithms choose to quote (Harris, 2003, p. 286). This is consistent with our empirical results, where volatility has a clear and significantly positive effect on the Corwin-Schultz spread, whereas lagged volume explains little of its variation (Table 1).

Given the pronounced summer pattern, one natural question is whether the effect is driven by dealer or broker behaviour. In today's market, Norwegian brokers act mainly as agency intermediaries and smart order routers rather than as classic balance-sheet dealers. They affect market quality primarily through the way their execution and routing algorithms are designed. This includes, among other things, choices between visible order books and any dark or internalised trading venues. In addition, they influence market quality through formal obligations to provide quotes or to act as market makers.

The data provide little support for a dealer-driven explanation. The decline in estimated spreads appears as a smooth and systematic pattern across stocks and across all three size groups (Tables 1 and 2), and it remains robust when both stock fixed effects and time-period fixed effects are included. If changes in broker behaviour were the main explanation, we would expect larger differences across individual stocks and trading venues. The results should also display a more erratic and random pattern. This is not what we observe. The findings are more consistent with a broad, calendar-driven market effect that reflects changes in trader composition and volatility than with a pure change in broker behaviour.

Interpretation 3 - Less available information:

The summer holiday period on the Oslo stock exchange is characterised by low news intensity, fewer earnings announcements and macro surprises, and generally smoother intraday price patterns (Hong and Yu, 2009). When both volatility and information risk decline, it is rational for algorithmic participants to reduce spread width. At the same time, lower order flow and reduced trading opportunities make it optimal to scale back the displayed depth at the best price levels. The Corwin-Schultz estimator, which is based on daily high and low prices, is particularly sensitive to the width of the intraday price range. Narrower intraday high-low movements therefore imply a lower estimated spread, even if the underlying depth in the order book may have weakened. The result may be a market where small trades, typically from smaller investors, face low observed transaction costs. However, as emphasised by Harris (2003), market quality cannot be assessed solely by the width of the spread. It must also be evaluated along other dimensions, including depth, price impact and the market's ability to absorb large orders.

Interpretation 4 - Can Norway's macroeconomic stability contribute to lower spreads in the summer?

A more overarching explanation for our findings is that Norway's macroeconomic and institutional stability may dampen the risk of negative news shocks during the holiday period. The Norwegian economy is characterised by relatively low political risk, strong public finances, a large sovereign wealth fund and a well-regulated financial market (Moody's Ratings, 2025). This may make both investors and liquidity providers less concerned that "something will go wrong" while key decision makers are on holiday. If the perceived probability of large, unexpected shocks is low, the need to price extra risk compensation into bid-ask spreads will also be limited. In that case a coordinated national holiday, which in more

vulnerable economies might be seen as a period of elevated systemic risk, may instead appear in Norway as a relatively safe "low volatile phase" in which market makers are willing to keep spreads tight. At the same time, it must be emphasised that our analysis only considers variation within the Norwegian market and does not compare Norway with countries that have weaker macroeconomic fundamentals. We therefore cannot conclude that the observed summer effect is caused by Norwegian stability as such, but this stability may be a contributing factor that makes it easier for liquidity providers to reduce spreads in a period when both the news flow and information risk are low.

5. Implicit transaction cost model

The significant reduction in spread found in the analysis for the collective summer holiday directly impacts implicit transaction cost. We first derive an equation for implicit transaction costs. Using in-sample data, we then build a simple model to estimate by how many percent these costs would fall if orders were shifted to the collective summer holiday instead of being executed during the rest of the year. Based on the model, we make a recommendation for order execution timing that we back test for the out-of-sample data from 2020 to 2024. A major caveat with the recommendation is, however, that the past does not predict the future. This dilemma is addressed using a simple shrinkage-style adjustment, loosely inspired by the approaches of Ledoit and Wolf (2003) and DeMiguel, Martín-Utrera and Nogales (2013).

5.1. Establishment of the implicit transaction cost equation

The total impact on implicit transaction cost, expressed as a proportion (in decimal form), from going from the relative spread of y to the relative spread of x for n stocks bought or sold can be written as:

$$\Delta \mathrm{ITC} = \frac{1}{n}\sum_{i=0}^{n}(\mathrm{K}i * \Delta S i)$$

ΔITC = Reduction in implicit transaction cost

ΔSi = Relative bid-ask spread change: (x-y)/y

$$\mathrm{Ki} = \frac{Effective\ spread}{Quoted\ spread}$$

The percentage saved depends upon the relationship between effective spread and quoted spread in $\frac{Effective\ spread}{Quoted\ spread}$. The effective spread is the "price you paid compared to the midpoint of the NBBO multiplied by two" (E*TRADE, n.d). If a trader used a market order or a marketable limit order and demands liquidity, the effective spread would equal the quoted spread and the relationship would be 1 between the two.

However, if an order executes at a price below the current best-ask for a bid-order, or above the current best-bid for an ask-order, the order has received price improvement (Harris, 2015, p. 28). The relationship between effective spread and quoted spread would then be lower than 1, essentially reducing the implicit transaction cost. The amount of price improvement could be seen as a variable in our calculation of reduced implicit transaction cost. From an investment perspective, any asset that is bought must eventually be sold if the return is to be realised, so lower effective spreads on both the entry and exit legs translate into a lower expected round-trip trading cost. In our framework, we therefore measure implicit transaction cost as the cost of a round-trip (buy followed by sell), and ΔITC is defined as the change in this round-trip cost between the two trading regimes we compare.

5.2. Weaknesses of the implicit transaction cost equation

One important limitation of our estimation is that we do not model market impact explicitly using Implementation Shortfall (Harris, 2015, p. 29). Market impact is inherently difficult to infer from historical data, because we never observe how the market would have evolved in the absence of our own order flow.

We therefore assume that all trades are filled at prices that do not move the mid-quote away from the initial decision price. This simplifies the calculation, but it also means that the estimates are most realistic for relatively small orders that can be executed without walking down the book.

A second limitation is that we do not model the opportunity-cost component of implementation shortfall (Harris 2015, p. 32). We do, however, assume relatively aggressive order placement, reflected in a high value of Ki close to 1 in the backtest, corresponding to the use of market or marketable limit orders that prioritise execution over price improvement. Under this assumption, the main margin of adjustment is when to trade (summer versus the rest of the year), not whether the orders are executed.

5.3. Predictability of the future

The evidence from 2014-2019 shows that the relative bid-ask spread is consistently lower during the collective summer holiday than during the rest of the year, with a significance level of $p<0.001$ (***). This very strong in-sample significance does not in itself guarantee predictive power for the future, or for this instance, the out of sample data from 2020 to 2024, but it provides a natural starting point for asking whether the effect might be sufficiently stable to be exploited in practice.

Unbiased estimator

To evaluate the future implications of executing trades during the summer as opposed to the rest of the year, we establish an unbiased estimator, $\hat{\theta}$, for the value of the spread. Let Qi denote the relative bid-ask spread for observation i calculated through the Corwin Shulz method. For spread x (during the collective summer holiday) and spread y (during the rest of the year), we observe respectively n_x observations of the bid-ask spread during the collective summer holiday and n_y observations the rest of the year. We define the estimators for each bid-ask spread as the sample mean of all observations:

$$\hat{\theta}_{x\ spread} = \frac{1}{n_x}\sum_{i=0}^{n_x} Qi_x$$

$$\hat{\theta}_{y\ spread} = \frac{1}{n_y}\sum_{i=0}^{n_y} Qi_y$$

The observed spreads, Qi, are treated as realisations from an underlying stochastic process that is stable within each of the two periods. For the summer holiday, we assume a constant but unknown mean spread $\theta_{x\ spread}$, and for the rest of the year a constant but unknown mean spread $\theta_{y\ spread}$, both with finite variance and no systematic trend.

Under these standard assumptions, the sample means defined above are unbiased and consistent estimators of the true mean spreads:

$$\mathrm{E}[\hat{\theta}_{x\ spread}] = \theta_{x\ spread}$$

$$\mathrm{E}[\hat{\theta}_{y\ spread}] = \theta_{y\ spread}$$

Creating:

$$E[\widehat{\Delta S}_{change\ spread}] = \Delta S_{change\ spread}$$

Table: Descriptive Statistics (Comparing Variances) for Figure 2A

period_label	Cap_Group	Mean	Pop. Variance (σ^2)	Sample Var (S^2)	95% CI Lower	95% CI Upper	N
control_year	Large Cap	0.01133	0.00009365	0.00009366	0.01117	0.01149	13,744
control_year	Mid Cap	0.01599	0.00028039	0.00028042	0.01568	0.01629	11,251
control_year	Small Cap	0.01927	0.00038851	0.00038855	0.01886	0.01967	9,030
summer_excl_holiday	Large Cap	0.01108	0.00008427	0.00008429	0.01080	0.01135	4,375
summer_excl_holiday	Mid Cap	0.01559	0.00026828	0.00026836	0.01506	0.01613	3,596
summer_excl_holiday	Small Cap	0.01777	0.00030375	0.00030386	0.01713	0.01841	2,893
summer_holiday	Large Cap	0.01027	0.00006006	0.00006011	0.00985	0.01070	1,288
summer_holiday	Mid Cap	0.01403	0.00019228	0.00019246	0.01318	0.01487	1,041
summer_holiday	Small Cap	0.01545	0.00021035	0.00021060	0.01446	0.01643	836
control_year	**All Cap**	**0.01497**	**0.00024425**	**0.00024426**	**0.01481**	**0.01514**	**34,025**
summer_excl_holiday	**All Cap**	**0.01435**	**0.00021157**	**0.00021159**	**0.01408**	**0.01463**	**10,864**
summer_holiday	**All Cap**	**0.01287**	**0.00014818**	**0.00014823**	**0.01245**	**0.01330**	**3,165**

Table 2: Descriptive statistics

The values of $\hat{\theta}_{x\ spread}$ and $\hat{\theta}_{y\ spread}$ is then computed trough python and reported in the chart above. The N-weighted mean relative bid-ask spread for the summer holiday and the rest of the year are accordingly $\hat{\theta}_{x\ spread}$=1,287 % and $\hat{\theta}_{y\ spread}$=1,497 %.

Based on these two means, we define the overall spread effect ΔS as the relative difference between the holiday and the non-holiday mean spreads. In other words, ΔS is the change in the mean relative bid-ask spread, and can be interpreted as the N-weighted mean of the stock-level effects ΔSi for the sample as a whole.

$$\Delta S = \frac{(\hat{\theta}_{x\ spread} - \hat{\theta}_{y\ spread})}{\hat{\theta}_{y\ spread}} = \frac{(1{,}287\ \% - 1{,}497\ \%)}{1{,}497\ \%} = -0.140281$$

For the in-sample period from 2014 to 2019 we see a reduction of about 14.03% in the mean relative bid ask spread during the summer holiday compared to the rest of the year. We can then calculate the reduction in implicit transaction, ΔITC, on a practical case using the formula derived above, subject to the assumptions and limitations discussed in the preceding subsection.

Ask:101 Bid: 99 Midprice of asset: 100 Quoted spread: 2

Purchase price: 100.95 (uses a price close to the ask to reduce the aspect of uncertain order execution)

$$K = \frac{Effective\ spread}{Quoted\ spread} = \frac{(Purchase\ price - midprice) * 2}{Quoted\ spread} = \frac{(100.95 - 100) * 2}{2} = 0.95$$

$$\Delta Si = \frac{(\hat{\theta}_{x\ spread} - \hat{\theta}_{y\ spread}}{\hat{\theta}_{y\ spread}} = \frac{(1{,}287\ \% - 1{,}497\ \%)}{1{,}497\ \%} = -0.140281$$

$$\Delta ITC = \frac{1}{1}\sum_{i=0}^{1}(0.95\ *\ -0.140281) = -0.13326695$$

A 14% reduction in the average spread with a slight price improvement implies a 13.33% reduction in implicit transaction costs. As our estimates show, such a 14% spread reduction can be achieved by a trader who buys or sells a representative mix of the 90 stocks and shifts the trading from the rest of the year to the collective summer holiday.

After using the model to describe 2014-2019-in-sample-periode, we then use it to evaluate 2020-2024, in order to identify when orders should be executed to minimize implicit transaction costs, abstracting other trading goals and constraints. From this perspective, we need to account for the estimator's, uncertainty in the in-sample data which is captured by the standard errors of the sample means.

5.4. The uncertainty in the values of ΔSi - 95 CI

A 95 % confidence interval for ΔS, could then be calculated trough:

$$\Delta S = \frac{\hat{\theta}_x - \hat{\theta}_y}{\hat{\theta}_y}$$

$$D = \hat{\theta}_x - \hat{\theta}_y$$

$$SE(\widehat{D}) = \sqrt{\hat{\sigma}^{2\ summer\ holiday} + \hat{\sigma}^{2\ control_year}}$$

$$SE(\widehat{\Delta si}) \approx \frac{SE(\widehat{D})}{\hat{\theta}_y}$$

$$\widehat{\Delta s} \pm 1{,}96 * SE(\widehat{\Delta si})$$

Giving the 95 % confidence interval: [-17%, -11%,]

5.5. *Shrinkage parameter* p

The very strong in-sample evidence from 2014-2019 suggests that trading during the collective summer holiday reduces the relative bid-ask spread compared to the rest of the year. However, the point estimate $\widehat{\Delta S}$ and its 95 % confidence interval [-11%, -17%] describe the past, not the future. Structural breaks, changes in market microstructure and evolving trading technologies all imply that the true out-of-sample effect may be smaller than the in-sample estimate, even when the historical effect is highly statistically significant to a degree of p<0,001(***).

To make this tension explicit, we introduce the parameter $p \in [0,1]$ which we refer to as the *shrinkage parameter*. P = 1 corresponds to fully extrapolating the in-sample estimate to the future, while p = 0 corresponds to assuming that the holiday effect does not persist out of sample. Intermediate values 0<p<1 represent varying degrees of confidence that the historical spread reduction will carry over to the out of sample period 2020-2024.

$\widehat{\Delta s}^{forecast}$ could then be written as a function of p:

$$\widehat{\Delta s}^{forecast}(p) = p * \widehat{\Delta s}^{in-sample}$$

This way of scaling the historical spread effect with a parameter $p \in [0,1]$ is conceptually analogous to linear shrinkage methods in finance, where a noisy sample estimate is combined with a more conservative target value. This can for example be seen in Ledoit and Wolf (2003), who estimate covariance matrices as convex combinations of the sample covariance matrix and a structured target matrix, and derive an optimal shrinkage intensity $\delta \in [0,1]$ that controls how much of the historical covariance structure is allowed to carry over into the estimator. In their framework, the shrinkage weight δ plays the same conceptual role as our parameter p. DeMiguel, Martín-Utrera and Nogales (2013) show in a portfolio-choice setting that the calibration of this shrinkage intensity has a first-order impact on out-of-sample portfolio performance, so the choice of δ effectively encodes how much one trusts the in-sample information. In our setting, the shrinkage target is a zero holiday effect, so choosing p amounts to shrinking the in-sample spread reduction towards no effect. Unlike the above studies, we do not estimate an optimal value of p. Instead, we treat p as a scenario parameter that represents different assumptions about how much of the historical holiday effect will persist out of sample.

The forecasting model for $\widehat{\Delta s}$, including the uncertainty from the in-sample data through the use of the 95 % CI then becomes:

Lower bound 95 % CI:

$$\widehat{\Delta s}^{forecast,\ lower}(p) = p * \widehat{\Delta s}^{in-sample,\ lower}$$

Upper bound 95 CI:

$$\widehat{\Delta s}^{forecast,\ upper}(p) = p * \widehat{\Delta s}^{in-sample,\ upper}$$

The forecasting model for ΔITC treats $\widehat{\Delta s}i$ as the estimated mean reduction in the relative bid-ask spread for stock i, derived from the in-sample mean spread estimators $\hat{\theta}_{x\ spread}$ and $\hat{\theta}_{y\ spread}$. The uncertainty in these estimates is taken into account by the use of the 95 % confidence interval for $\widehat{\Delta s}$:

Lower bound 95 % CI:

$$\widehat{\Delta ITC}^{forecast,lower}(p) = p * \sum_{i=0}^{n} (Ki\ *\ \widehat{\Delta s}\imath^{in-sample\ mean,\ 95\ lower})$$

Upper bound 95 % CI:

$$\widehat{\Delta ITC}^{forecast,upper}(p) = p * \sum_{i=0}^{n} (Ki\ *\ \widehat{\Delta s}\imath^{in-sample\ mean,\ 95\ upper})$$

If we focus on a single representative scenario rather than the full stock level, the sum in the forecast equations can be collapsed. Let K denote the position size (or scaling factor on the portfolio) and let 0.11 and 0.17 be the lower and upper 95 percent bounds for the in-sample mean spread reduction. The interval for the reduction in implicit transaction costs then simplifies to:

Lower bound 95 % CI:

$$\widehat{\Delta ITC}^{forecast,lower}(p,k) = p\ *\ k\ *\ 0{,}11$$

Upper bound 95 % CI:

$$\widehat{\Delta \mathrm{ITC}}^{forecast,upper}(p,k) = p * k * 0{,}11$$

5.6. Plot of the forecast model

This can be visualized by plotting $\widehat{\Delta \mathrm{ITC}}^{forecast,lower}(p,K)$and $\widehat{\Delta \mathrm{ITC}}^{forecast,upper}(p,K)$as functions of the two variables p and k. In the 3-dimensional figure, the horizontal p-axis measures the degree to which the in-sample effect from 2014-2019 is carried over to the out-of-sample period, the horizontal k-axis measures the amount of price improvement, and the vertical z-axis shows $\Delta ITC(p,k)$, that is, the percentage reduction in implicit transaction costs

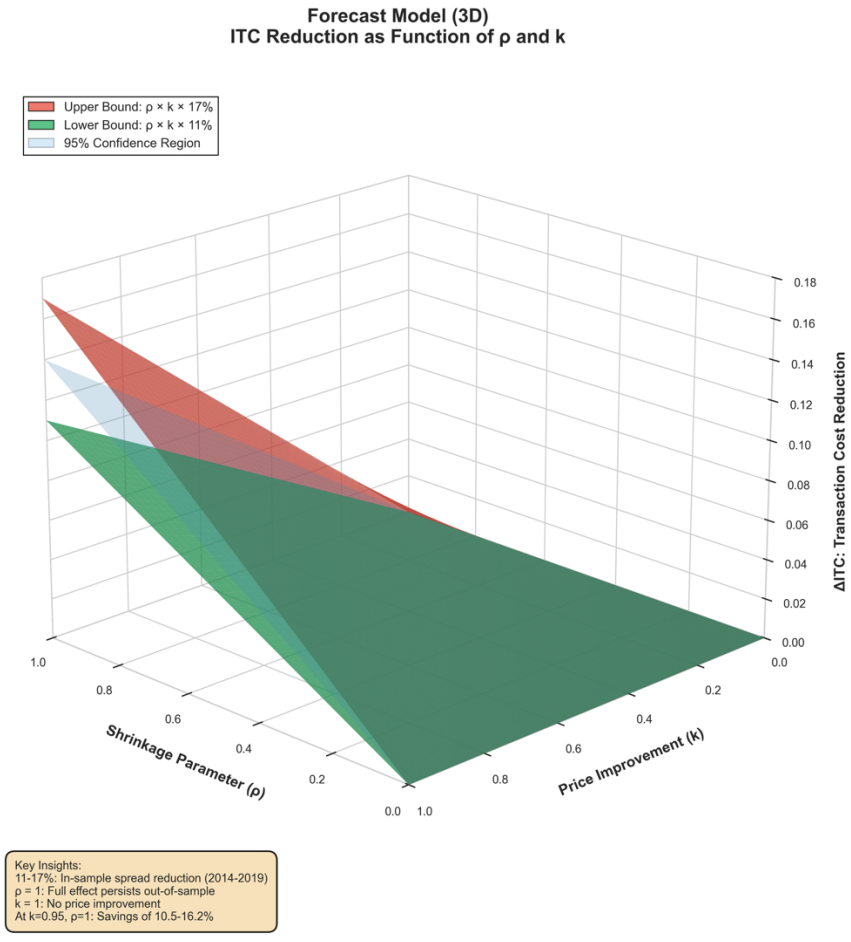


Figure 11: Forecast model for ΔITC as a function of p and K

The red surface represents the 95 percent upper forecast $\widehat{\Delta\text{ITC}}^{forecast,upper}(p,K)$ and the green surface represents the 95 percent lower forecast $\widehat{\Delta\text{ITC}}^{forecast,lower}(p,K)$. The savings are largest when the in-sample effect is fully transferred to the out-of-sample period, which corresponds to high values of p along the p-axis combined with a high k value. Ask k decreases, both planes move closer to zero on the z-axis, indicating that more price improvement mechanically reduces ΔITC and therefore the total cost reduction.

If we then input at fixed k value of 0.95, in other words a slight price improvement, we could then express the reduction in implicit transaction cost as a linear function of the *shrinkage parameter* p:

Lower bound 95 % ci:

$$\Delta\text{ITC}_{lower}(p) = p * 0{,}95 * 11\%$$

Upper bound 95 % ci:

$$\Delta\text{ITC}_{upper}(p) = p * 0{,}95 * 17\%$$

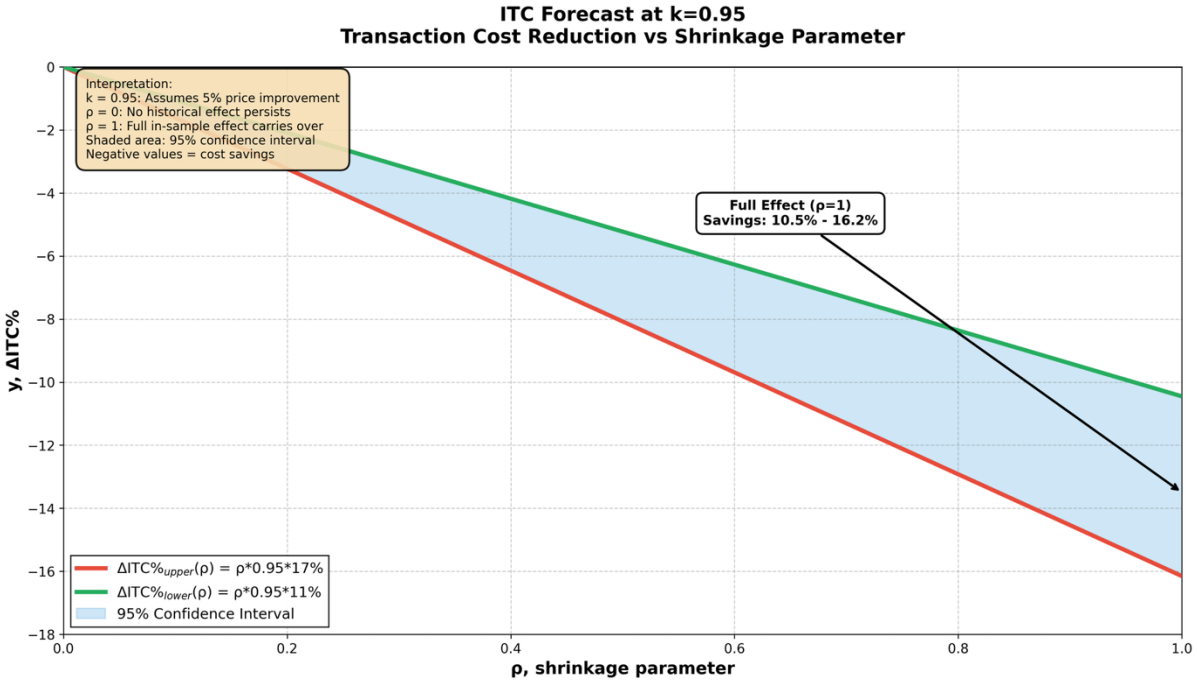


Figure 12: Forecast model for ΔITC as a function of p, with K=0.95

Here we have reduced the three-dimensional surface to a two-dimensional slice where ΔITC varies linearly with p with a steeper slope for the upper bound. For a given position size k = 0.95, higher values of p therefore translate proportionally into larger predicted reductions in implicit transaction costs.

For each value of the shrinkage parameter p ∈ [0,1], the green and red lines represent the lower and upper 95 % bounds of the expected cost savings if the historical holiday effect persists to that degree of p out of sample. The blue region between them is the corresponding confidence set for $\Delta\text{ITC}^{forecast\ out\ of\ sample}(p)$.

A cautious interpretation is that the blue band shows a range for $\Delta ITC^{forecast\ out\ of\ sample}$ as a function of p, given the uncertainty in-sample data through the use of 95 % confidence intervals. The true out-of-sample effect could be smaller than suggested by the band, or the spread difference could even turn out to go against the strategy, for example if spreads in a future sample are higher during the collective summer holiday than during the rest of the year. This is why Figures 10 and 11 only display the model-implied reduction in implicit transaction costs. When p = 0, the forecast simply corresponds to a benchmark with no assumed holiday effect. The spread difference may ex post still turn out either to favour or to work against shifting trading to the summer window, but such outcomes are not explicitly represented by the model.

Because both the three-dimensional surface (Figure 10) and the two-dimensional slice (Figure 11) are based on in-sample data up to 2019, they should be updated sequentially as new out-of-sample observations are realised by re-estimating Δs and its 95 % confidence interval on the enlarged in-sample window and inserting the updated estimates into the same prediction model.

5.7. Recommendation

Taken together, the model suggests a tentative recommendation. Liquidity-demanding investors who trade a diversified portfolio that is broadly representative of the 90-stock sample used in our estimation could reduce expected implicit transaction costs by shifting their order flow to the collective summer holiday period. The potential savings increase for scenario values of p that place non-trivial weight on the historical spread reduction (values of p clearly above zero) and for relatively aggressive order placement (high K). In the next step, we therefore

backtest the strategy on the out-of-sample period 2020-2024 to assess whether the implied cost savings actually materialise.

5.8. Backtesting

The backtest is done by comparing the mean relative spread in the collective summer holiday with the mean relative spread in the remaining trading days in each year in the out of sample period from 2020 to 2024. The figure below shows that the spread is lower in the summer holiday in all years except 2022. The mean reduction in the spread between the summer holiday and the rest of the year from 2020 to 2024 is 14.37%,

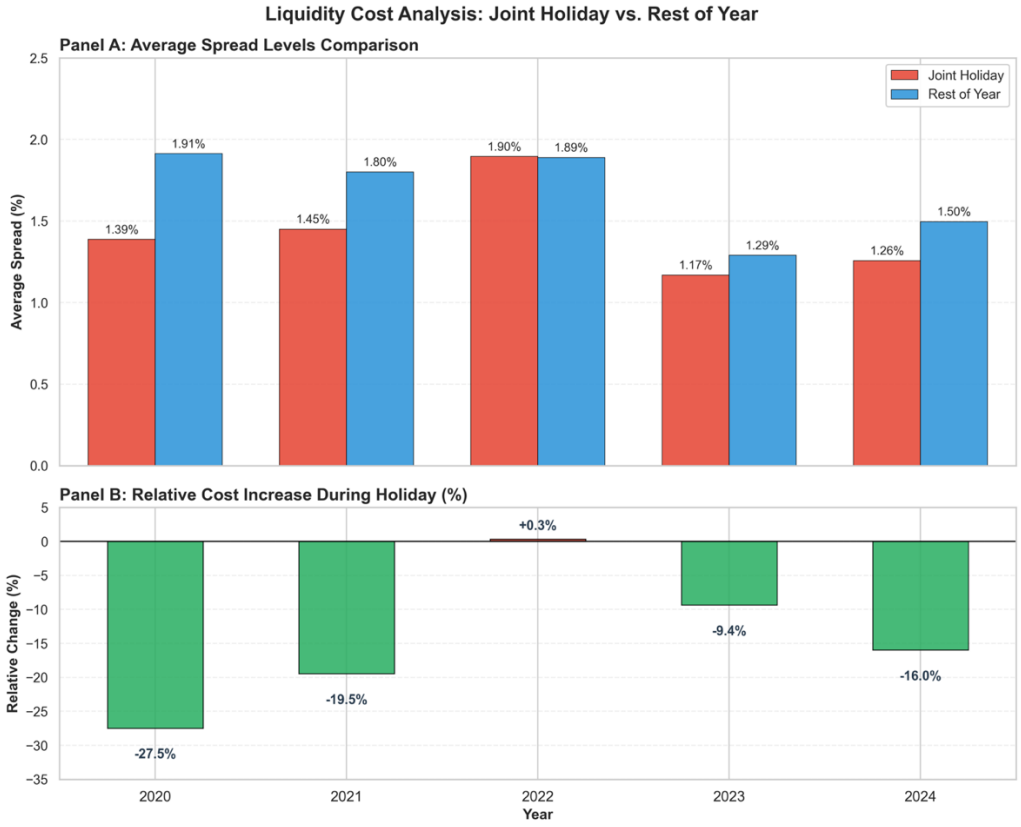


Figure 13: Liquidity cost analysis out-of-sample

Actual saving

A lower mean relative bid-ask spread during the summer holiday compared with the rest of the year implies a reduction in implicit transaction costs in our ΔITC framework. In this calculation we focus only on the effect of the quoted spread and ignore market impact and the

possibility that orders walk down the book (see Section 5.2, “Weaknesses of the implicit transaction cost equation”).

We summarise the average percentage reduction in round-trip implicit transaction costs across the sample by

$$\Delta\mathrm{ITC} = \frac{1}{n}\sum_{i=0}^{n}(\mathrm{K}i * \Delta\mathrm{S}i)$$

From the out-of-sample backtest comparing the collective summer holiday to the rest of the year, we have that the mean relative spread reduction is about 14.42%. For illustration, we assume that each stock experiences the same proportional change $\Delta\mathrm{S}i = 0.1442$ and the same ratio $\mathrm{K}i = K$, so that the expression simplifies to:

$$\Delta\mathrm{ITC} = K * 0{,}1442$$

For a trader who consistently demands liquidity with market orders or marketable limit orders, k would equal 1. In that case, the trader captures the full 14.42% reduction in the spread-related round-trip cost.

To translate this into dollars, consider a stock with a midprice of 100 dollars and a amount of 100 stock. The notional trade size is then 10,000 dollars. Because we treat implicit transaction costs as a round-trip cost, we approximate the round-trip cost by the relative bid-ask spread itself. The mean relative bid-ask spreads are:

Rest of the year: 1.678%

Summer holiday: 1.434%

Rest of the year: 1.678 % × 10,000 = 167.8

Summer holiday: 1.434 % × 10,000 = 143.4

The implied saving is 24.4 dollars on this round-trip trade in a position with a notional value of 10,000 dollars.

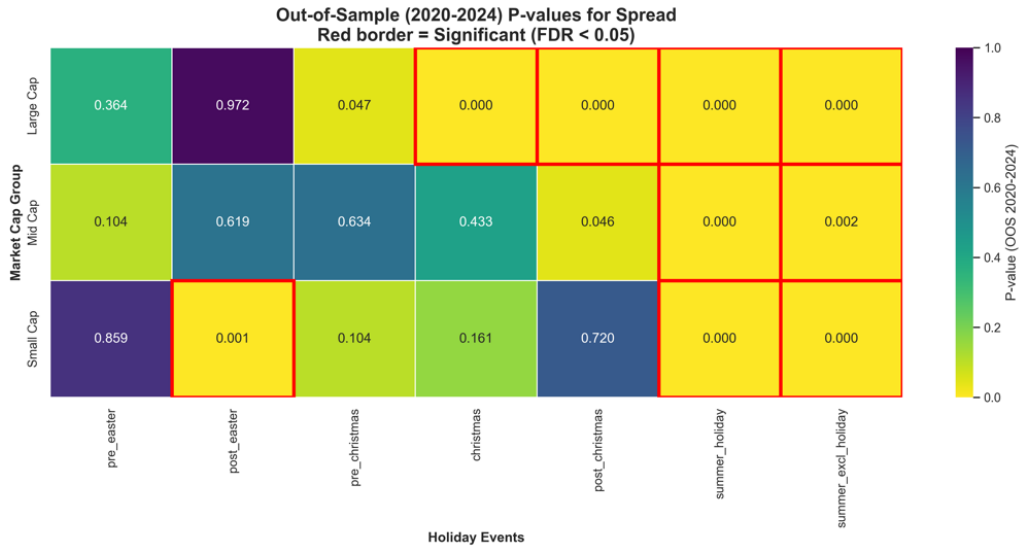


Figure 14: P-values out-of-sample

For the out of sample, we see that the p-values for the spread are highly significant across all cap sizes for the summer holiday. The reduction in the mean relative spread during the collective summer holiday, compared with the rest of the year, in the out of sample data, is highly statistically significant with to the degree of $p \leq 0{,}001$(***). These results indicate that the holiday effect documented in-sample also persists in the out-of-sample period.

6. Conclusion

6.1. Summary

In the research the main focus was to examine how major Norwegian holiday periods affect the relative bid ask spread and volume on the Oslo Stock Exchange between 2014 and 2019, using outbound air travel as a proxy for reduction in market participation. The research examined Easter and Christmas holiday windows but detected little to no statistically significant effects on volume or spread for the five trading days pre- and post-these holidays. However, some smaller, less robust effects were noted, such as a significant reduction in spreads for large-cap stocks pre-Easter and an increase in spreads for large- and small-cap stocks post-Christmas.

The collective summer holiday, on the other hand, shows trading volume being significantly lower across all three cap-sizes during this period. After correcting for multiple testing using the FDR procedure, the strong reduction in volume remains statistically significant for mid- and large-cap stocks. This confirms the widely held belief that stock turnover is lower during the summer because market participants are on vacation, supporting the interpretation that a *holiday effect* is present on the OSE. Counterintuitively, the mean relative bid-ask spread was lower during the collective summer holiday. After applying statistical corrections, this reduction in spreads is confirmed as a robust and broad-based phenomenon across all three market-capitalization groups. This finding is contradictory to prior evidence from the OSE (Hong & Yu, 2009) and may reflect a reduction in adverse selection, consistent with a scenario in which informed traders are more likely than other market participants to withdraw from the market during the summer holiday.

The primary, actionable application from the execution-timing strategy designed to minimize implicit transaction costs is then, don’t go on holiday: *When analysts leave for the holidays*. This implies that buy-side market participants should utilize the summer holiday period to execute orders, thereby capturing the reduced transaction costs. For practitioners, the consistently narrower spread during the collective summer holiday can serve as a final decision criterion for executing a trade. If an investor’s fundamental analysis or other metrics align with a buy or sell signal, the fact that the spread is most likely low during the summer holiday could act as the "last trigger" for order placement. This is dependent on the market liquidity being sufficient for the intended order size.

6.2. Limitations with our analysis

Our analysis has several limitations. First, we use a balanced panel of 90 stocks that survive the entire sample period, which may introduce survivorship bias and make the results less representative of more fragile or illiquid firms. The data are based on daily OHLC prices from a data vendor rather than true intraday quotes from the exchange. Both the Corwin Schultz spread and our measure of realized volatility are therefore only approximations and are partly mechanically correlated. The holiday windows, defined as five trading days before and after Easter and Christmas and three fixed weeks in July, are also somewhat ad hoc, and neighbouring weeks may contain substantial holiday activity.

From a statistical perspective, both the Welch tests and the panel regressions treat observations as more independent than they really are. Common shocks across stocks and repeated use of the same control days imply that p-values and FDR-adjusted significance levels should be interpreted with caution. The back test of implicit transaction costs is highly simplified. We ignore market impact, assume a stable relationship between quoted and effective spreads, and rely on only a short and turbulent out-of-sample period. Finally, we do not observe investor types, order book depth or price efficiency, so the proposed mechanism of fewer informed traders and lower adverse selection is an interpretation rather than a directly tested causal relationship.

7. References

Abdi, F. & Ranaldo, A. (2017). A simple estimation of bid-ask spreads from daily close, high, and low prices. *The Review of Financial Studies, 30*(12), 4437-4480. https://academic.oup.com/rfs/article/30/12/4437/4047344

Benjamini, Y. & Hochberg, Y. (1995). Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society: Series B (Methodological), 57*(1), 289-300. https://rss.onlinelibrary.wiley.com/doi/10.1111/j.2517-6161.1995.tb02031.x

Corwin, S. A. & Schultz, P. (2012). A simple way to estimate bid-ask spreads from daily high and low prices. *The Journal of Finance, 67*(2), 719-760. https://onlinelibrary.wiley.com/doi/10.1111/j.1540-6261.2012.01729.x

Curtin, F. & Schulz, P. (1998). Multiple correlations and Bonferroni's correction. *Biological Psychiatry, 44*(8), 775-777. https://www.sciencedirect.com/science/article/abs/pii/S0006322398000432

DeMiguel, V., Martín-Utrera, A. & Nogales, F. J. (2013). Size matters: Optimal calibration of shrinkage estimators for portfolio selection. *Journal of Banking & Finance*, 37(8), 3018-3034. https://www.sciencedirect.com/science/article/abs/pii/S0378426613002161

E*TRADE. (n.d.). *Trade execution quality. E*TRADE from Morgan Stanley. https://us.etrade.com/trade/execution-quality

Easley, D., Kiefer, N. M., O'Hara, M. & Paperman, J. B. (1996). Liquidity, information, and infrequently traded stocks. *The Journal of Finance, 51*(4), 1405–1436. https://doi.org/10.1111/j.1540-6261.1996.tb04074.x

Erikstad, T. (2021, 22. juli). 12 timers arbeidsdag, 6 dager i uken for 900.000. *Dagens Næringsliv*. https://www.dn.no/kommentar/finans/wall-street/jpmorgan-chase/12-timers-arbeidsdag-6-dager-i-uken-for-900000/2-1-1043025

Ferieloven. (1988). Lov om ferie (LOV-1988-04-29-21). Lovdata. https://lovdata.no/pro/#document/NL/lov/1988-04-29-21?searchResultContext=1563&rowNumber=1&totalHits=1243

Harris, L. (2003). *Trading and exchanges: Market microstructure for practitioners*. Oxford University Press.

Harris, L. (2015). *Trading and electronic markets: What investment professionals need to know*. CFA Institute Research Foundation.

Hong, H. & Yu, J. (2009). Gone fishin': Seasonality in trading activity and asset prices. *Journal of Financial Markets, 12*(4), 672-702. https://www.sciencedirect.com/science/article/pii/S1386418109000378?via%3Dihub

Ledoit, O., & Wolf, M. (2003). Improved estimation of the covariance matrix of stock returns with an application to portfolio selection. Journal of Empirical Finance, 10(5), 603-621. https://www.sciencedirect.com/science/article/abs/pii/S0927539803000070

Moody's Ratings. (2025, June 10). *Government of Norway - Aaa stable: Regular update*. Moody's Ratings. https://www.norges-bank.no/contentassets/65ed68bc853049f7bb671568e77a1489/moodys-2025.pdf?v=10062025155533

Pareto Securities. (2025, 2. juni). *Slik posisjonerer du deg på Oslo Børs før sommerferien*. Pareto Securities. https://paretosec.no/kunnskap/posisjoner-deg-for-sommeren

Regulation Department. (2024). *Trading manual for the Optiq trading platform*. Euronext.

Schenk-Hoppé, K. R. (2025, 15. september). *Session_14_Dealers_2025.pdf*. Canvas. https://nhh.instructure.com/courses/2915/pages/session-13-16

Statistics Norway. (2025). *08507: Air transport. Passengers, by airport, traffic type and domestic/international flights 2009M01-2025M09*. Statbank Norway. https://www.ssb.no/en/statbank/table/08507

8. Appendix

8.1. A - Complete hypothesis:

Traded volume - small cap.

Traded volume measured in shares traded

Easter:

$$1\ H_0: \mu_{Mean\ pre\ easter\ volume\ small\ cap} = \mu_{Mean\ volume\ rest\ of\ year\ small\ cap}$$

$$2\ H_A: \mu_{Mean\ pre\ easter\ volume\ small\ cap} \neq \mu_{Mean\ volume\ rest\ of\ year\ small\ cap}$$

$$3\ H_0: \mu_{Mean\ post\ easter\ volume\ small\ cap} = \mu_{Mean\ volume\ rest\ of\ year\ small\ cap}$$

$$4\ H_A: \mu_{Mean\ post\ easter\ volume\ small\ cap} \neq \mu_{Mean\ volume\ rest\ of\ year\ small\ cap}$$

Christmas:

$$5\ H_0: \mu_{Mean\ pre\ Christmas\ volume\ small\ cap} = \mu_{Mean\ volume\ rest\ of\ year\ small\ cap}$$

$$6\ H_A: \mu_{Mean\ pre\ Christmas\ volume\ small\ cap} \neq \mu_{Mean\ volume\ rest\ of\ year\ small\ cap}$$

$$7\ H_0: \mu_{Mean\ post\ Christmas\ volume\ small\ cap} = \mu_{Mean\ volume\ rest\ of\ year\ small\ cap}$$

$$8\ H_A: \mu_{Mean\ post\ Christmas\ volume\ small\ cap} \neq \mu_{Mean\ volume\ rest\ of\ year\ small\ cap}$$

Summer months:

$$9\ H_0: \mu_{Mean\ summer\ months\ volume\ small\ cap} = \mu_{Mean\ volume\ rest\ of\ year\ small\ cap}$$

$$10\ H_A: \mu_{Mean\ summer\ months\ volume\ small\ cap} \neq \mu_{Mean\ volume\ rest\ of\ year\ small\ cap}$$

Collective summer holiday:

11 H_0: $\mu_{Mean\ volume\ collective\ summer\ holiday\ small\ cap} = \mu_{Mean\ volume\ rest\ of\ year\ small\ cap}$

12 H_A: $\mu_{Mean\ volume\ collective\ summer\ holiday\ small\ cap} \neq \mu_{Mean\ volume\ rest\ of\ year\ small\ cap}$

Spread - small cap.

Easter:

13 H_0: $\mu_{Mean\ pre\ easter\ spread\ small\ cap} = \mu_{Mean\ spread\ rest\ of\ year\ small\ cap}$

14 H_A: $\mu_{Mean\ pre\ easter\ spread\ small\ cap} \neq \mu_{Mean\ spread\ rest\ of\ year\ small\ cap}$

15 H_0: $\mu_{Mean\ post\ easter\ spread\ small\ cap} = \mu_{Mean\ spread\ rest\ of\ year\ small\ cap}$

16 H_A: $\mu_{Mean\ post\ easter\ spread\ small\ cap} \neq \mu_{Mean\ spread\ rest\ of\ year\ small\ cap}$

Christmas:

17 H_0: $\mu_{Mean\ pre\ Christmas\ spread\ small\ cap} = \mu_{Mean\ spread\ rest\ of\ year\ small\ cap}$

18 H_A: $\mu_{Mean\ pre\ Christmas\ spread\ small\ cap} \neq \mu_{Mean\ spread\ rest\ of\ year\ small\ cap}$

19 H_0: $\mu_{Mean\ post\ Christmas\ spread\ small\ cap} = \mu_{Mean\ spread\ rest\ of\ year\ small\ cap}$

20 H_A: $\mu_{Mean\ post\ Christmas\ spread\ small\ cap} \neq \mu_{Mean\ spread\ rest\ of\ year\ small\ cap}$

Summer months:

21 H_0: $\mu_{Mean\ summer\ months\ spread\ small\ cap} = \mu_{Mean\ spread\ rest\ of\ year\ small\ cap}$

22 H_A: $\mu_{Mean\ summer\ months\ spraed\ small\ cap} \neq \mu_{Mean\ spread\ rest\ of\ year\ small\ cap}$

Collective summer holiday:

23 H_0: $\mu_{Mean\ spread\ collective\ summer\ holiday\ small\ cap} = \mu_{Mean\ spread\ rest\ of\ year\ small\ cap}$

24 H_A: $\mu_{Mean\ spread\ collective\ summer\ holiday\ small\ cap} \neq \mu_{Mean\ spread\ rest\ of\ year\ small\ cap}$

Traded volume - mid. cap.

Traded volume measured in shares traded

Easter:

$$25\ H_0: \mu_{Mean\ pre\ easter\ volume}\ \text{Mid.cap.} = \mu_{Mean\ volume\ rest\ of\ year\ Mid.cap.}$$

$$26\ H_A: \mu_{Mean\ pre\ easter\ volume}\ \text{Mid.cap.} \neq \mu_{Mean\ volume\ rest\ of\ year\ Mid.cap.}$$

$$27\ H_0: \mu_{Mean\ post\ easter\ volume}\ \text{Mid.cap.} = \mu_{Mean\ volume\ rest\ of\ year\ Mid.cap.}$$

$$28\ H_A: \mu_{Mean\ post\ easter\ volume}\ \text{Mid.cap.} \neq \mu_{Mean\ volume\ rest\ of\ year\ Mid.cap.}$$

Christmas:

$$29\ H_0: \mu_{Mean\ pre\ Christmas\ volume}\ \text{Mid.cap.} = \mu_{Mean\ volume\ rest\ of\ year\ Mid.cap.}$$

$$30\ H_A: \mu_{Mean\ pre\ Christmas\ volume}\ \text{Mid.cap.} \neq \mu_{Mean\ volume\ rest\ of\ year\ Mid.cap.}$$

$$31\ H_0: \mu_{Mean\ post\ Christmas\ volume}\ \text{Mid.cap.} = \mu_{Mean\ volume\ rest\ of\ year\ Mid.cap.}$$

$$32\ H_A: \mu_{Mean\ post\ Christmas\ volume}\ \text{Mid.cap.} \neq \mu_{Mean\ volume\ rest\ of\ year\ Mid.cap.}$$

Summer months:

$$33\ H_0: \mu_{Mean\ summer\ months\ volume}\ \text{Mid.cap.} = \mu_{Mean\ volume\ rest\ of\ year\ Mid.cap.}$$

$$34\ H_A: \mu_{Mean\ summer\ months\ volume}\ \text{Mid.cap.} \neq \mu_{Mean\ volume\ rest\ of\ year\ Mid.cap.}$$

Collective summer holiday:

$$35\ H_0: \mu_{Mean\ volume\ collective\ summer\ holiday\ Mid.cap.} = \mu_{Mean\ volume\ rest\ of\ year\ Mid.cap.}$$

$$36\ H_A: \mu_{Mean\ volume\ collective\ summer\ holiday\ Mid.cap.} \neq \mu_{Mean\ volume\ rest\ of\ year\ Mid.cap.}$$

Spread - mid. cap.

Easter:

$$37\ H_0: \mu_{Mean\ pre\ easter\ spread\ Mid.cap.} = \mu_{Mean\ spread\ rest\ of\ year\ Mid.cap.}$$

$$38\ H_A: \mu_{Mean\ pre\ easter\ spread\ Mid.cap.} \neq \mu_{Mean\ spread\ rest\ of\ year\ Mid.cap.}$$

$$39\ H_0: \mu_{Mean\ post\ easter\ spread\ Mid.cap.} = \mu_{Mean\ spread\ rest\ of\ year\ Mid.cap.}$$

$$40\ H_A: \mu_{Mean\ post\ easter\ spread\ Mid.cap.} \neq \mu_{Mean\ spread\ rest\ of\ year\ Mid.cap.}$$

Christmas:

$$41\ H_0: \mu_{Mean\ pre\ Christmas\ spread\ Mid.cap.} = \mu_{Mean\ spread\ rest\ of\ year\ Mid.cap.}$$

$$42\ H_A: \mu_{Mean\ pre\ Christmas\ spread\ Mid.cap.} \neq \mu_{Mean\ spread\ rest\ of\ year\ Mid.cap.}$$

$$43\ H_0: \mu_{Mean\ post\ Christmas\ spread\ Mid.cap.} = \mu_{Mean\ spread\ rest\ of\ year\ Mid.cap.}$$

$$44\ H_A: \mu_{Mean\ post\ Christmas\ spread\ Mid.cap.} \neq \mu_{Mean\ spread\ rest\ of\ year\ Mid.cap.}$$

Summer months:

$$45\ H_0: \mu_{Mean\ summer\ months\ spread\ Mid.cap.} = \mu_{Mean\ spread\ rest\ of\ year\ Mid.cap.}$$

$$46\ H_A: \mu_{Mean\ summer\ months\ spraed\ Mid.cap.} \neq \mu_{Mean\ spread\ rest\ of\ year\ Mid.cap.}$$

Collective summer holiday:

$$47\ H_0: \mu_{Mean\ spread\ collective\ summer\ holiday\ Mid.cap.} = \mu_{Mean\ spread\ rest\ of\ year\ Mid.cap.}$$

$$48\ H_A: \mu_{Mean\ spread\ collective\ summer\ holiday\ Mid.cap.} \neq \mu_{Mean\ spread\ rest\ of\ year\ Mid.cap.}$$

Traded volume - large cap.

Traded volume measured in shares traded

Easter:

$$49\ H_0: \mu_{Mean\ pre\ easter\ volume\ large\ cap.} = \mu_{Mean\ volume\ rest\ of\ year\ large\ cap.}$$

$$50\ H_A: \mu_{Mean\ pre\ easter\ volume\ large\ cap.} \neq \mu_{Mean\ volume\ rest\ of\ year\ large\ cap.}$$

$$51\ H_0: \mu_{Mean\ post\ easter\ volume\ large\ cap.} = \mu_{Mean\ volume\ rest\ of\ year\ large\ cap.}$$

$$52\ H_A: \mu_{Mean\ post\ easter\ volume\ large\ cap.} \neq \mu_{Mean\ volume\ rest\ of\ year\ large\ cap.}$$

Christmas:

$$53\ H_0: \mu_{Mean\ pre\ Christmas\ volume\ large\ cap.} = \mu_{Mean\ volume\ rest\ of\ year\ large\ cap.}$$

$$54\ H_A: \mu_{Mean\ pre\ Christmas\ volume\ large\ cap.} \neq \mu_{Mean\ volume\ rest\ of\ year\ large\ cap.}$$

$$55\ H_0: \mu_{Mean\ post\ Christmas\ volume\ large\ cap.} = \mu_{Mean\ volume\ rest\ of\ year\ large\ cap.}$$

$$56\ H_A: \mu_{Mean\ post\ Christmas\ volume\ large\ cap.} \neq \mu_{Mean\ volume\ rest\ of\ year\ large\ cap.}$$

Summer months:

$$57\ H_0: \mu_{Mean\ summer\ months\ volume\ large\ cap.} = \mu_{Mean\ volume\ rest\ of\ year\ large\ cap.}$$

$$58\ H_A: \mu_{Mean\ summer\ months\ volume\ large\ cap.} \neq \mu_{Mean\ volume\ rest\ of\ year\ large\ cap.}$$

Collective summer holiday:

$$59\ H_0: \mu_{Mean\ volume\ collective\ summer\ holiday\ large\ cap.} = \mu_{Mean\ volume\ rest\ of\ year\ large\ cap.}$$

$$60\ H_A: \mu_{Mean\ volume\ collective\ summer\ holiday\ large\ cap.} \neq \mu_{Mean\ volume\ rest\ of\ year\ large\ cap.}$$

Spread - large cap

Easter:

61 H_0: $\mu_{Mean\ pre\ easter\ spread\ large\ cap.} = \mu_{Mean\ spread\ rest\ of\ year\ large\ cap.}$

62 H_A: $\mu_{Mean\ pre\ easter\ spread\ large\ cap.} \neq \mu_{Mean\ spread\ rest\ of\ year\ large\ cap.}$

63 H_0: $\mu_{Mean\ post\ easter\ spread\ large\ cap.} = \mu_{Mean\ spread\ rest\ of\ year\ large\ cap.}$

64 H_A: $\mu_{Mean\ post\ easter\ spread\ large\ cap.} \neq \mu_{Mean\ spread\ rest\ of\ year\ large\ cap.}$

Christmas:

65 H_0: $\mu_{Mean\ pre\ Christmas\ spread\ large\ cap.} = \mu_{Mean\ spread\ rest\ of\ year\ large\ cap.}$

66 H_A: $\mu_{Mean\ pre\ Christmas\ spread\ large\ cap.} \neq \mu_{Mean\ spread\ rest\ of\ year\ large\ cap.}$

67 H_0: $\mu_{Mean\ post\ Christmas\ spread\ large\ cap.} = \mu_{Mean\ spread\ rest\ of\ year\ large\ cap.}$

68 H_A: $\mu_{Mean\ post\ Christmas\ spread\ large\ cap.} \neq \mu_{Mean\ spread\ rest\ of\ year\ large\ cap.}$

Summer months:

69 H_0: $\mu_{Mean\ summer\ months\ spread\ large\ cap.} = \mu_{Mean\ spread\ rest\ of\ year\ large\ cap.}$

70 H_A: $\mu_{Mean\ summer\ months\ spraed\ large\ cap.} \neq \mu_{Mean\ spread\ rest\ of\ year\ large\ cap.}$

Collective summer holiday:

71 H_0: $\mu_{Mean\ spread\ collective\ summer\ holiday\ large\ cap.} = \mu_{Mean\ spread\ rest\ of\ year\ large\ cap.}$

72 H_A: $\mu_{Mean\ spread\ collective\ summer\ holiday\ large\ cap.} \neq \mu_{Mean\ spread\ rest\ of\ year\ large\ cap.}$

8.2. B - Complete hypothesis for Christmas:

Volume small cap:

$$73\ H_0: \mu_{Mean\ Christmas\ volume\ smal\ cap.} = \mu_{Mean\ Christmas\ volume\ smal\ cap.}$$

$$74\ H_A: \mu_{Mean\ Christmas\ volume\ smal\ cap.} \neq \mu_{Mean\ Christmas\ volume\ smal\ cap.}$$

Volume mid. cap:

$$75\ H_0: \mu_{Mean\ Christmas\ volume\ mid\ cap.} = \mu_{Mean\ Christmas\ volume\ mid\ cap.}$$

$$76\ H_A: \mu_{Mean\ Christmas\ volume\ mid\ cap.} \neq \mu_{Mean\ Christmas\ volume\ mid\ cap.}$$

Volume large cap.

$$77\ H_0: \mu_{Mean\ Christmas\ volume\ large\ cap.} = \mu_{Mean\ Christmas\ volume\ large\ cap.}$$

$$78\ H_A: \mu_{Mean\ Christmas\ volume\ large\ cap.} \neq \mu_{Mean\ Christmas\ volume\ large\ cap.}$$

Spread small cap:

$$79\ H_0: \mu_{Mean\ Christmas\ spread\ small\ cap.} = \mu_{Mean\ Christmas\ spread\ small\ cap}$$

$$80\ H_A: \mu_{Mean\ Christmas\ spread\ small\ cap.} \neq \mu_{Mean\ Christmas\ spread\ small\ cap}$$

Spread mid. cap:

$$81\ H_0: \mu_{Mean\ Christmas\ spread\ mid.\ cap.} = \mu_{Mean\ Christmas\ spread\ mid.\ cap.}$$

$$82\ H_A: \mu_{Mean\ Christmas\ spread\ mid.\ cap.} \neq \mu_{Mean\ Christmas\ spread\ mid.\ cap.}$$

Spread large cap:

$$83\ H_0: \mu_{Mean\ Christmas\ spread\ large\ cap.} = \mu_{Mean\ Christmas\ spread\ large\ cap.}$$

$$84\ H_A: \mu_{Mean\ Christmas\ spread\ large\ cap.} \neq \mu_{Mean\ Christmas\ spread\ large\ cap.}$$

8.3. C - Python code:

```
# SCRIPT 0: INSTALL & IMPORT LIBRARIES
import sys
import subprocess
import importlib

required_packages = [
   "yfinance", "pandas", "numpy", "holidays", "matplotlib",
   "seaborn", "scipy", "statsmodels", "linearmodels", "stargazer", "mpl_toolkits.mplot3d"
]

for package in required_packages:
   try:
       importlib.import_module(package)
   except ImportError:
       subprocess.check_call([sys.executable, "-m", "pip", "install", package])

import yfinance as yf
import pandas as pd
import numpy as np
import calendar
import holidays
from mpl_toolkits.mplot3d import Axes3D
import matplotlib.pyplot as plt
import matplotlib.patches as patches
import seaborn as sns
import scipy.stats as stats
from scipy.stats import linregress
from scipy.stats import ttest_ind
import statsmodels.api as sm
from statsmodels.iolib.summary2 import summary_col
from statsmodels.stats.multitest import multipletests
from statsmodels.stats.outliers_influence import variance_inflation_factor
from linearmodels.panel import PanelOLS
from stargazer.stargazer import Stargazer
import os

disable_warnings = True
if disable_warnings:
   import warnings
   warnings.filterwarnings("ignore")

# SCRIPT 1: SETUP - DEFINE CONSTANTS AND TICKERS

TICKERS = {
   "Large Cap": [
       'EQNR.OL', 'DNB.OL', 'KOG.OL', 'MOWI.OL', 'TEL.OL', 'NHY.OL', 'AKRBP.OL',
       'ORK.OL', 'STB.OL', 'YAR.OL', 'SUBC.OL', 'GJF.OL', 'SALM.OL', 'TGS.OL',
       'TOM.OL', 'VAR.OL', 'NEL.OL', 'FRO.OL', 'BWLPG.OL', 'HAUTO.OL',
       'NOD.OL', 'WAWI.OL', 'NAS.OL', 'BAKKA.OL', 'WWI.OL', 'AFK.OL',
       'AUSS.OL', 'SCATC.OL', 'MPCC.OL', 'HAFNI.OL'
   ],
   "Mid Cap": [
       'AKER.OL', 'LSG.OL', 'KIT.OL', 'AKSO.OL', 'PARB.OL', 'BONHR.OL',
       'BOUV.OL', 'DNO.OL', 'ENTRA.OL', 'FLNG.OL', 'MING.OL', 'NAPA.OL',
       'NORBT.OL', 'OLT.OL', 'PCIB.OL', 'REACH.OL', 'WSTEP.OL', 'KOA.OL',
       'HSPG.OL', 'SOFF.OL', 'ABG.OL', 'BGBIO.OL', 'EMGS.OL', 'EXTX.OL',
       'HAVI.OL', 'HELG.OL', 'IDEX.OL', 'JIN.OL', 'MULTI.OL', 'NYKD.OL'
   ],
```

```
    "Small Cap": [
        'QEC.OL', 'RECSI.OL', 'SPOL.OL', 'AZT.OL', 'KID.OL', 'SATS.OL',
        'AURG.OL', 'PEN.OL', 'LINK.OL', 'PROT.OL', 'IOX.OL', 'ACC.OL',
        'TECH.OL', 'CONTX.OL', 'NONG.OL', 'BEWI.OL', 'ELO.OL', 'GSF.OL',
        'PRS.OL', 'AIRX.OL', 'OBSRV.OL', 'HUNT.OL', 'AKVA.OL', 'HEX.OL',
        'SOFTX.OL', 'ASA.OL', 'NORTH.OL', 'CAPSL.OL', 'LYTIX.OL', 'VOW.OL'
    ]
}

START_DATE = "2014-01-01"
END_DATE = "2024-12-31"
OUTPUT_CSV = "oslo_bors_labelled_data.csv"

# SCRIPT 1.1: CORWIN-SCHULTZ, ABDI-RANALDO, AND ROLL SPREAD ESTIMATORS
def corwin_schultz_spread(group):
    """
    Corwin-Schultz (2012) High-Low spread estimator.
    This function calculates a daily spread estimate.

    Reference: Corwin, S. A., & Schultz, P. (2012). A simple way to estimate
    bid-ask spreads from daily high and low prices. The Journal of Finance, 67(2), 719-760.
    """
    group = group.sort_values('Date')

    high = group['High'].values
    low = group['Low'].values

    spreads = []
    negative_count = 0

    for i in range(len(group)):
        if i == 0:
            spreads.append(np.nan)
        else:
            h0, l0 = high[i], low[i]
            h1, l1 = high[i-1], low[i-1]

            if h0 > 0 and l0 > 0 and h1 > 0 and l1 > 0:
                beta = (np.log(h0/l0))**2 + (np.log(h1/l1))**2
                h_max = max(h0, h1)
                l_min = min(l0, l1)
                gamma = (np.log(h_max/l_min))**2

                k = (3 - 2*np.sqrt(2))
                alpha = (np.sqrt(2*beta) - np.sqrt(beta)) / k - np.sqrt(gamma / k)

                spread = 2 * (np.exp(alpha) - 1) / (1 + np.exp(alpha))

                if spread >= 0:
                    spreads.append(spread)
                else:
                    spreads.append(np.nan)
                    negative_count += 1
            else:
                spreads.append(np.nan)

    if negative_count > 0:
        ticker = group['Ticker'].iloc[0] if 'Ticker' in group.columns else 'Unknown'
        # print(f"  Warning: {ticker} had {negative_count} negative Corwin-Schultz estimates (set
to NaN)")

    group['Spread'] = spreads
    return group
```

```python
def abdi_ranaldo_spread(group):
    """
    Calculates the NORMALIZED Abdi and Ranaldo (2017) "BAR" spread estimator.
    This version converts the absolute spread (in currency) to a relative
    spread (as a percentage of the price), making it comparable across stocks.

    Note: Additional filtering is applied to remove economically meaningless spreads.
    """
    group = group.sort_values('Date')

    delta_p_t = group['Close'].diff()
    delta_p_t_minus_1 = delta_p_t.shift(1)

    if delta_p_t.count() < 2:
        return pd.Series(np.nan, index=group.index, name='Spread')

    cov_term = (delta_p_t * delta_p_t_minus_1)

    # Calculate the absolute spread (in currency units)
    absolute_spreads = 2 * np.sqrt(np.maximum(0, -cov_term))

    # NORMALIZATION STEP
    # To get a relative spread, divide by the average of today's and yesterday's close price.
    # Rolling window to get the average price at each point in time.
    avg_price = group['Close'].rolling(window=2, min_periods=1).mean()

    # Calculate the relative spread. Use .values to avoid index alignment issues.
    relative_spreads = absolute_spreads / avg_price.values

    # CRITICAL FIX: Filter out economically meaningless spreads
    # Spreads below 0.01% (0.0001) are likely measurement errors or rounding artifacts
    relative_spreads = np.where(relative_spreads < 0.0001, np.nan, relative_spreads)

    # Assign the comparable, relative spreads to the DataFrame
    group['Spread'] = relative_spreads
    return group



def get_roll_spread(group):
    """
    Roll (1984) spread estimator based on serial covariance of price changes.
    Returns a single spread estimate per stock (not time-varying).

    Roll's spread = 2 * sqrt(-Cov(ΔP_t, ΔP_t-1))

    The Roll estimator assumes that negative serial covariance in returns is
    caused by bid-ask bounce. If the covariance is positive, it violates
    Roll's model assumptions and we return NaN.

    Reference: Roll, R. (1984). A simple implicit measure of the effective
    bid-ask spread in an efficient market. The Journal of Finance, 39(4), 1127-1139.
    """
    group = group.sort_values('Date')
    returns = group['Close'].pct_change().dropna()

    if len(returns) < 30:  # Minimum observations for reliable estimate
        return np.nan

    # Calculate serial covariance (covariance between returns and lagged returns)
    cov = returns.cov(returns.shift(1))
```

```
    # Roll's spread formula (only valid when covariance is negative)
    if pd.notna(cov) and cov < 0:
        spread = 2 * np.sqrt(-cov)
    else:
        # Positive covariance violates Roll's assumptions
        spread = np.nan

    return spread

# SCRIPT 1.2: ADD PERIOD LABELS FOR HOLIDAY ANALYSIS
def add_period_labels(df):
    """
    Adds period labels for holiday analysis.
    """
    df = df.copy()
    df['Date'] = pd.to_datetime(df['Date'])
    years = range(df['Date'].dt.year.min(), df['Date'].dt.year.max() + 1)
    norway_holidays = holidays.Norway(years=years)

    df['week'] = df['Date'].dt.isocalendar().week.astype(int)
    df['month'] = df['Date'].dt.month
    df['period_label'] = 'control_year'

    for yr in years:
        christmas_ref_start = pd.Timestamp(f'{yr}-12-15')
        christmas_ref_end = pd.Timestamp(f'{yr}-12-31')
        christmas_days = pd.bdate_range(start=christmas_ref_start, end=christmas_ref_end,
freq='C', holidays=list(norway_holidays.keys()), weekmask='Mon Tue Wed Thu Fri')

        if len(christmas_days) > 0:
            pre_christmas = pd.bdate_range(end=christmas_days[0] - pd.Timedelta(days=1),
periods=5, freq='C', holidays=list(norway_holidays.keys()), weekmask='Mon Tue Wed Thu Fri')
            try:
                post_christmas_start = pd.Timestamp(f'{yr+1}-01-02')
                post_christmas = pd.bdate_range(start=post_christmas_start, periods=5, freq='C',
holidays=list(norway_holidays.keys()), weekmask='Mon Tue Wed Thu Fri')
            except:
                post_christmas = pd.bdate_range(start=christmas_days[-1] + pd.Timedelta(days=1),
periods=5, freq='C', holidays=list(norway_holidays.keys()), weekmask='Mon Tue Wed Thu Fri')

            df.loc[df['Date'].isin(pre_christmas), 'period_label'] = 'pre_christmas'
            df.loc[df['Date'].isin(christmas_days), 'period_label'] = 'christmas'
            df.loc[df['Date'].isin(post_christmas), 'period_label'] = 'post_christmas'

    for yr in years:
        easter_days = [d for d,
h in norway_holidays.items() if ('Påske' in h or h in ['Skjærtorsdag', 'Langfredag']) and d.year
 == yr]

        if easter_days:
            easter_start = min(easter_days)
            easter_end = max(easter_days)
            pre_easter = pd.bdate_range(end=easter_start - pd.Timedelta(days=1), periods=5,
freq='C', holidays=list(norway_holidays.keys()), weekmask='Mon Tue Wed Thu Fri')
            post_easter = pd.bdate_range(start=easter_end + pd.Timedelta(days=1), periods=5,
freq='C', holidays=list(norway_holidays.keys()), weekmask='Mon Tue Wed Thu Fri')
            df.loc[df['Date'].isin(pre_easter), 'period_label'] = 'pre_easter'
            df.loc[df['Date'].isin(post_easter), 'period_label'] = 'post_easter'

    df.loc[df['week'].between(28, 30), 'period_label'] = 'summer_holiday'
    df.loc[(df['month'].between(6, 8)) & (~df['week'].between(28, 30)), 'period_label'] ='summer_
excl_holiday'

    return df
```

```
# SCRIPT 1.3: CHOOSE SPREAD ESTIMATOR (Simplified)
user_input = input("Which spread estimator do you want to use? (Enter 'CS' or 'AR'): ")

# Determine the spread method based on user input
if user_input.upper() == 'AR':
   METHOD_CHOICE = 'AR'
   SPREAD_METHOD_NAME = 'Abdi & Ranaldo (2017) Spread'
else:
   METHOD_CHOICE = 'CS'
   SPREAD_METHOD_NAME = 'Corwin-Schultz (2012) Spread'

print(f"The analysis will use the '{SPREAD_METHOD_NAME}'")
```

```
The analysis will use the 'Corwin-Schultz (2012) Spread'
```

```
# SCRIPT 1.4: CHOOSE MULTIPLE TESTING CORRECTION METHOD

correction_input = input(
   "Which multiple testing correction do you want to use for significance?\n(Enter 'B' or
'FDR')").strip().lower()

if correction_input == 'fdr':
   CORRECTION_METHOD = 'FDR'
   CORRECTION_METHOD_NAME = 'FDR'
elif correction_input == 'B':
   CORRECTION_METHOD = 'Bonferroni'
   CORRECTION_METHOD_NAME = 'Bonferroni'
else:
   print("Invalid input - using FDR by default.")
   CORRECTION_METHOD = 'FDR'
   CORRECTION_METHOD_NAME = 'FDR'

print(f"The analysis will use the '{CORRECTION_METHOD_NAME}' correction method.")
```

```
The analysis will use the 'FDR' correction method.
```

```
# SCRIPT 1.5: MAIN DATA PROCESSING FUNCTION

def main():

   all_tickers = [ticker for sublist in TICKERS.values() for ticker in sublist]
   print(f"\nDownloading data for {len(all_tickers)} tickers...")
   print(f"Date range: {START_DATE} to {END_DATE}")

   raw_data = yf.download(all_tickers, start=START_DATE, end=END_DATE, progress=True,
auto_adjust=True)

   if raw_data.empty:
       print("ERROR: No data downloaded")
       return

   print("TRANSFORMING DATA STRUCTURE: ")
   df = raw_data.stack(future_stack=True).reset_index()
   df = df.rename(columns={'level_1': 'Ticker'})

   ticker_to_cap = {ticker: cap for cap, tickers in TICKERS.items() for ticker in tickers}
   df['Cap_Group'] = df['Ticker'].map(ticker_to_cap)

   initial_len = len(df)
   df.dropna(subset=['Open', 'High', 'Low', 'Close', 'Volume', 'Cap_Group'], inplace=True)
   print(f"Removed {initial_len - len(df):,} rows with missing price/volume data")

   # Spread Calculation
   print("\n")
   if METHOD_CHOICE == 'CS':
```

```
        print("CALCULATING CORWIN-SCHULTZ SPREADS (DAILY ESTIMATE)")
        df = df.groupby('Ticker', group_keys=False).apply(corwin_schultz_spread)
    elif METHOD_CHOICE == 'AR':
        print("CALCULATING ABDI & RANALDO (2017) 'BAR' SPREADS")
        df = df.groupby('Ticker', group_keys=False).apply(abdi_ranaldo_spread)
    else:
        raise ValueError("Invalid method choice. Please choose 'CS' or 'AR'.")
    print("Spread calculation complete.")

    # CRITICAL FIX: COMPREHENSIVE SPREAD DIAGNOSTICS
    print("\n")
    print(f"SPREAD DISTRIBUTION DIAGNOSTICS - {METHOD_CHOICE}")

    # Count spread values by category
    total_obs = len(df)
    zero_spreads = (df['Spread'] == 0).sum()
    negative_spreads = (df['Spread'] < 0).sum()
    tiny_spreads = ((df['Spread'] > 0) & (df['Spread'] < 0.0001)).sum()
    valid_spreads = (df['Spread'] >= 0.0001).sum()
    nan_spreads = df['Spread'].isna().sum()

    print(f"\nSpread Value Distribution:")
    print(f"  Total observations:        {total_obs:,}")
    print(f"  Zero values (= 0):         {zero_spreads:,} ({zero_spreads/total_obs*100:.2f}%)")
    print(f"  Negative values (<
0):     {negative_spreads:,} ({negative_spreads/total_obs*100:.2f}%)")
    print(f"  Tiny spreads (0 to 0,01%): {tiny_spreads:,} ({tiny_spreads/total_obs*100:.2f}%)")
    print(f"  Valid spreads (≥ 0,01%):   {valid_spreads:,} ({valid_spreads/total_obs*100:.2f}%)")
    print(f"  Missing (NaN):             {nan_spreads:,} ({nan_spreads/total_obs*100:.2f}%)")

    # Percentile analysis
    print(f"\nSpread Percentiles (before filtering):")
    percentiles = df['Spread'].describe(percentiles=[.01, .05, .10, .25, .50, .75, .90, .95, .99]
)
    for stat in ['min', '1%', '5%', '10%', '25%', '50%', '75%', '90%', '95%', '99%', 'max']:
        if stat in percentiles.index:
            print(f"  {stat:>5}: {percentiles[stat]:.6f}")

    # Warning if median is zero or very low
    median_spread = df['Spread'].median()
    if median_spread == 0:
        print("\nWARNING: Median spread is ZERO!")
        print("   - This indicates a data quality issue with the spread estimator.")
        print("   - Most observations have zero spread, which is economically unrealistic.")
    elif median_spread < 0.0001:
        print(f"\nWARNING: Median spread is very low ({median_spread:.6f})")
        print("   - Many spreads may be economically meaningless.")
    else:
        print(f"\nMedian spread: {median_spread:.6f} (appears reasonable)")

    # Data Quality Check and Cleaning
    print(f"DATA QUALITY FILTERING - {METHOD_CHOICE}")

    spread_before = len(df)

    # Remove NaN spreads
    df = df.dropna(subset=['Spread'])
    nan_removed = spread_before - len(df)
    print(f"\nRemoved {nan_removed:,} rows with NaN spreads")

    # CRITICAL FIX: Remove economically meaningless spreads
    # Spreads below 0.01% (1 basis point) are likely measurement errors
    spread_before_filter = len(df)
    df = df[df['Spread'] >= 0.0001]
```

```
tiny_removed = spread_before_filter - len(df)
print(f"Removed {tiny_removed:,} rows with spreads < 0.01% (1 bp)")
print(f"These are economically meaningless and likely measurement errors")

print(f"\nTotal rows removed: {spread_before - len(df):,}")
print(f"Remaining observations: {len(df):,}")

# Post-filtering diagnostics
print(f"\nPOST-FILTERING SPREAD STATISTICS:")
print(f"  Mean:   {df['Spread'].mean():.6f}")
print(f"  Median: {df['Spread'].median():.6f}")
print(f"  Std:    {df['Spread'].std():.6f}")
print(f"  Min:    {df['Spread'].min():.6f}")
print(f"  Max:    {df['Spread'].max():.6f}")

# Roll's Spread Calculation
roll_spreads = df.groupby('Ticker').apply(get_roll_spread)
df['RollsSpread'] = df['Ticker'].map(roll_spreads)

valid_rolls = roll_spreads.notna().sum()
total_tickers = df['Ticker'].nunique()
invalid_rolls = total_tickers - valid_rolls

print(f"\nRoll's spread calculation summary:")
print(f"  Tickers with valid spread: {valid_rolls} / {total_tickers}")
if invalid_rolls > 0:
    print(f"  Tickers with invalid spread: {invalid_rolls}")
    print(f"   - Likely due to positive serial covariance (violates Roll's assumptions)")
print(f"  Average Roll's spread: {roll_spreads.mean():.6f}")

# Compare Roll with CS/AR
avg_cs_ar = df.groupby('Ticker')['Spread'].mean()
comparison = pd.DataFrame({
    'Roll': roll_spreads,
    'CS_AR_Avg': avg_cs_ar
}).dropna()

if len(comparison) > 0:
    correlation = comparison.corr().iloc[0, 1]
    print(f"\nCorrelation between Roll and {METHOD_CHOICE} average: {correlation:.3f}")
    if correlation > 0.7:
        print(f" - Strong correlation indicates reliable spread estimates!")
    elif correlation > 0.5:
        print(f" - Moderate correlation - spread estimates are reasonably consistent")
    else:
        print(f"Low correlation - may indicate measurement issues")

# Period Labeling and Final Export
df = add_period_labels(df)


final_cols = ['Date', 'Ticker', 'Cap_Group', 'Open', 'High', 'Low', 'Close',
              'Volume', 'Spread', 'RollsSpread', 'period_label']
df_final = df[[col for col in final_cols if col in df.columns]]
df_final.to_csv(OUTPUT_CSV, index=False)

print(f"\nData saved to: {OUTPUT_CSV}")
print(f"Total observations: {len(df_final):,}")
print(f"Date range: {df_final['Date'].min()} to {df_final['Date'].max()}")
print(f"Tickers: {df_final['Ticker'].nunique()}")

# Final summary by cap group
print(f"\n FINAL SPREAD SUMMARY BY CAP GROUP:")
summary = df_final.groupby('Cap_Group')['Spread'].agg(['count', 'mean', 'median', 'std'])
```

```
    summary.columns = ['Observations', 'Mean', 'Median', 'Std Dev']
    print(summary.to_string())


if __name__ == "__main__":
    main()
```

```
Downloading data for 90 tickers...
Date range: 2014-01-01 to 2024-12-31
[**********************100%**********************]  90 of 90 completed

1 Failed download:
['FLNG.OL']: YFPricesMissingError('possibly delisted; no price data found  (1d 2014-01-01 ->
2024-12-31) (Yahoo error = "No data found, symbol may be delisted")')

TRANSFORMING DATA STRUCTURE:
Removed 35,978 rows with missing price/volume data


CALCULATING CORWIN-SCHULTZ SPREADS (DAILY ESTIMATE)
Spread calculation complete.


SPREAD DISTRIBUTION DIAGNOSTICS - CS

Spread Value Distribution:
 Total observations:        212,602
 Zero values (= 0):         11,526 (5.42%)
 Negative values (< 0):     0 (0.00%)
 Tiny spreads (0 to 0,01%): 1,198 (0.56%)
 Valid spreads (≥ 0,01%):   117,883 (55.45%)
 Missing (NaN):             81,995 (38.57%)

Spread Percentiles (before filtering):
   min: 0.000000
    1%: 0.000000
    5%: 0.000000
   10%: 0.000184
   25%: 0.004364
   50%: 0.010342
   75%: 0.019576
   90%: 0.033479
   95%: 0.045767
   99%: 0.080270
   max: 0.408711

Median spread: 0.010342 (appears reasonable)
DATA QUALITY FILTERING - CS

Removed 81,995 rows with NaN spreads
Removed 12,724 rows with spreads < 0.01% (1 bp)
These are economically meaningless and likely measurement errors

Total rows removed: 94,719
Remaining observations: 117,883

POST-FILTERING SPREAD STATISTICS:
 Mean:   0.016560
 Median: 0.011729
 Std:    0.017290
 Min:    0.000100
 Max:    0.408711
```

```
Roll's spread calculation summary:
 Tickers with valid spread: 63 / 89
 Tickers with invalid spread: 26
  - Likely due to positive serial covariance (violates Roll's assumptions)
 Average Roll's spread: 0.019974

Correlation between Roll and CS average: 0.672
- Moderate correlation - spread estimates are reasonably consistent

Data saved to: oslo_bors_labelled_data.csv
Total observations: 117,883
Date range: 2014-01-03 00:00:00 to 2024-12-30 00:00:00
Tickers: 89

FINAL SPREAD SUMMARY BY CAP GROUP:
           Observations      Mean    Median   Std Dev
Cap_Group
Large Cap         41853  0.012160  0.009567  0.010714
Mid Cap           39779  0.017153  0.011985  0.017980
Small Cap         36251  0.020989  0.015222  0.021036
```

```
# SCRIPT 2: DESCRIPTIVE STATISTICS AND VISUALIZATION
# Load Data
try:
   df = pd.read_csv('oslo_bors_labelled_data.csv', parse_dates=['Date'])
except FileNotFoundError:
   print("ERROR: 'oslo_bors_labelled_data.csv' not found.")
if not os.path.exists('output'):
   os.makedirs('output')


sns.set_style("whitegrid")
sns.set_context("notebook", font_scale=1.2)


# CRITICAL STEP: SPLIT SAMPLE DESIGN
# Split data into In-Sample (Discovery) and Out-of-Sample (Validation)
# In-Sample: 2014-2019 (Used for descriptive stats, regressions, establishing the effect)
# Out-of-Sample: 2020-2024 (Used ONLY for backtesting trading strategies)


df['Date'] = pd.to_datetime(df['Date'])
df['Year'] = df['Date'].dt.year


# Create the two distinct datasets
df_in_sample = df[df['Year'] <= 2019].copy()
df_out_of_sample = df[df['Year'] >= 2020].copy()


print("METHODOLOGICAL SPLIT-SAMPLE DESIGN IMPLEMENTED")
print(f"In-Sample Data (2014-2019):     {len(df_in_sample):,} observations (Used for
analysis/regression)")
print(f"Out-of-Sample Data (2020-2024): {len(df_out_of_sample):,} observations (Used ONLY for
backtesting)")


# IMPORTANT: Overwrite 'df' to be only in-sample data for the rest of the analysis (Script 2-5)
# This is to ensure that tables and graphs do not "see" the future.
df_full_backup = df.copy()
df = df_in_sample


# Dataset overview (In-Sample)
```

```
print(f"\nTotal observations (In-Sample): {len(df):,}")
try:
    print(f"Date range: {df['Date'].min()} to {df['Date'].max()}")
    print(f"Unique tickers: {df['Ticker'].nunique()}")
    print(f"Trading days: {df['Date'].nunique():,}")
except:
    pass


# CRITICAL ADDITION: COHEN'S D EFFECT SIZE CALCULATION


def calculate_cohens_d(group1, group2):
    if len(group1) < 2 or len(group2) < 2: return np.nan
    n1, n2 = len(group1), len(group2)
    var1, var2 = group1.var(), group2.var()
    pooled_std = np.sqrt(((n1-1)*var1 + (n2-1)*var2) / (n1+n2-2))
    if pooled_std == 0: return 0
    d = (group1.mean() - group2.mean()) / pooled_std
    return d


# Calculate effect sizes for key holiday periods
control = df[df['period_label'] == 'control_year']['Spread']
summer = df[df['period_label'] == 'summer_holiday']['Spread']
christmas = df[df['period_label'] == 'christmas']['Spread']
easter_pre = df[df['period_label'] == 'pre_easter']['Spread']
easter_post = df[df['period_label'] == 'post_easter']['Spread']


print("\nCohen's d Effect Sizes (vs. Control Period):")
print(f" Summer Holiday: d = {calculate_cohens_d(summer, control):.4f}")
print(f" Christmas: d = {calculate_cohens_d(christmas, control):.4f}")
print(f" Pre-Easter: d = {calculate_cohens_d(easter_pre, control):.4f}")
print(f" Post-Easter: d = {calculate_cohens_d(easter_post, control):.4f}")
print("\nInterpretation: |d| > 0.5 = medium to large effect")


# CRITICAL ADDITION: HARRIS FRAMEWORK - DEALER BEHAVIOR HYPOTHESIS TEST
print("HARRIS FRAMEWORK: DEALER BEHAVIOR HYPOTHESIS")
print("\nTesting Harris's prediction: Spreads should increase when volume decreases")
print("(Lower dealer participation and higher inventory risk)")


df['Volume_Quartile'] = pd.qcut(df['Volume'], q=4,
labels=['Q1_Lowest', 'Q2_Low', 'Q3_High', 'Q4_Highest'], duplicates='drop')


spread_by_volume = df.groupby('Volume_Quartile', observed=True).agg({
    'Spread': ['mean', 'median', 'count'],
    'Volume': 'mean'
}).round(6)
spread_by_volume.columns = ['Spread_Mean', 'Spread_Median', 'N_Obs', 'Avg_Volume']


print("\nSpread by Volume Quartile:")
print(spread_by_volume)


q1_spread = df[df['Volume_Quartile'] == 'Q1_Lowest']['Spread']
q4_spread = df[df['Volume_Quartile'] == 'Q4_Highest']['Spread']
t_stat, p_val = ttest_ind(q1_spread.dropna(), q4_spread.dropna(), equal_var=False)
cohens_d_volume = calculate_cohens_d(q1_spread.dropna(), q4_spread.dropna())
```

```
print(f"\nHarris Hypothesis Test:")
print(f" H0: Spread(Low Volume) = Spread(High Volume)")
print(f" H1: Spread(Low Volume) > Spread(High Volume)")
try:
   print(f" Mean spread Q1 (low vol): {q1_spread.mean():.6f}")
   print(f" Mean spread Q4 (high vol): {q4_spread.mean():.6f}")
   print(f" Difference: {q1_spread.mean() - q4_spread.mean():.6f}")
   print(f" t-statistic: {t_stat:.4f}")
   print(f" p-value: {p_val:.4f}")
   print(f" Cohen's d: {cohens_d_volume:.4f}")


   if p_val < 0.05 and q1_spread.mean() > q4_spread.mean():
       print(" ✓ RESULT: Harris's prediction CONFIRMED (p < 0.05)")
       print(" - Spreads are wider when volume is low (dealer participation reduced)")
   elif p_val < 0.05 and q1_spread.mean() < q4_spread.mean():
       print(" ✗ RESULT: Harris's prediction CONTRADICTED (p < 0.05)")
       print(" - Modern electronic markets may behave differently than Harris predicted")
   else:
       print(" -  RESULT: No significant relationship (p >= 0.05)")
except:
   print("Insufficient data for Harris test.")



# Summary by cap group
print("SUMMARY BY CAP GROUP")
summary_cap = df.groupby('Cap_Group').agg({
   'Spread': ['mean', 'median', 'std'],
   'Volume': ['mean', 'median'],
   'Ticker': 'nunique'
}).round(6)
summary_cap.columns = ['Spread_Mean', 'Spread_Median', 'Spread_Std', 'Volume_Mean', 'Volume_Medi
an', 'N_Stocks']
print(summary_cap)



# HARRIS INTERPRETATION
print("\n HARRIS INTERPRETATION:")
print(" Large Cap stocks have narrower spreads due to:")
print(" - Higher competitive market making (Harris Chapter 6)")
print(" - Lower inventory risk for dealers")
print(" - Greater participation rates (Harris Chapter 21)")



# Summary by period
print("SUMMARY BY PERIOD")
summary_period = df.groupby('period_label').agg({
   'Spread': ['mean', 'median'],
   'Volume': ['mean', 'median'],
   'Date': 'count'
}).round(6)
summary_period.columns = ['Spread_Mean', 'Spread_Median', 'Volume_Mean', 'Volume_Median', 'N_Obs
']
print(summary_period)



# Calculate percentage changes vs control
try:
   control_spread = summary_period.loc['control_year', 'Spread_Mean']
```

```
    control_volume = summary_period.loc['control_year', 'Volume_Mean']
    summary_period['Spread_Change_%'] = ((summary_period['Spread_Mean'] - control_spread) /contro
l_spread * 100).round(2)
    summary_period['Volume_Change_%'] = ((summary_period['Volume_Mean'] - control_volume) /contro
l_volume * 100).round(2)


    print("CHANGES VS CONTROL PERIOD")
    print(summary_period[['Spread_Mean', 'Spread_Change_%', 'Volume_Mean', 'Volume_Change_%']])
except KeyError:
    print("Control year not found in summary.")



summary_cap.to_csv('output/summary_by_cap.csv')
summary_period.to_csv('output/summary_by_period.csv')



# CRITICAL ADDITION: PRACTICAL IMPLICATIONS (NOK COST SAVINGS)
print("PRACTICAL IMPLICATIONS: TRADING COST ANALYSIS")


trade_sizes = [100000, 500000, 1000000, 5000000] # NOK
avg_stock_price = 100 # We assume 100 NOK per share


try:
    spread_control = summary_period.loc['control_year', 'Spread_Mean']
    spread_summer = summary_period.loc['summer_holiday', 'Spread_Mean']
    spread_diff = spread_summer - spread_control


    print(f"\nSpread difference (Summer vs. Control): {spread_diff:.6f}")
    print(f"Average stock price assumed: {avg_stock_price} NOK\n")


    print("Cost Impact per Trade (one-way):")
    print(f"{'Trade Size (NOK)':>20} {'Control Cost':>15} {'Summer Cost':>15} {'Savings':>15}")
    print("-" * 70)


    for trade_size in trade_sizes:
        shares = trade_size / avg_stock_price
        cost_control = shares * avg_stock_price * spread_control / 2 # Half-spread for one-way
        cost_summer = shares * avg_stock_price * spread_summer / 2
        savings = cost_control - cost_summer
        print(f"{trade_size:>20,} {cost_control:>14.2f} NOK {cost_summer:>14.2f} NOK {savings:>14
.2f}NOK")


    print("\nINTERPRETATION:")
    if spread_diff < 0:
        print(f" Trading during summer holidays saves
approximately {abs(spread_diff/spread_control)*100:.1f}%")
        print(f" on transaction costs due to narrower spreads.")
    else:
        print(f" Trading during summer holidays costs
approximately {(spread_diff/spread_control)*100:.1f}% more")
        print(f" due to wider spreads (consistent with Harris's dealer hypothesis).")
except:
    pass
```

```
# FIGURE 1: CHRISTMAS & EASTER ANALYSIS

christmas_easter_periods = ['control_year', 'pre_easter', 'post_easter',
                            'pre_christmas', 'christmas', 'post_christmas']
df_ce = df[df['period_label'].isin(christmas_easter_periods)].copy()
df_ce['period_label'] = pd.Categorical(df_ce['period_label'],
                                       categories=christmas_easter_periods,
                                       ordered=True)

# FIGURE 1A (UPDATED: Labels to the RIGHT)
plt.figure(figsize=(14, 8))
ax = sns.pointplot(x='period_label', y='Spread', hue='Cap_Group', data=df_ce,
             hue_order=['Large Cap', 'Mid Cap', 'Small Cap'],
             dodge=True,
             palette='viridis',
             errorbar='ci',
             capsize=0.1)

means_1a = df_ce.groupby(['period_label', 'Cap_Group'], observed=False)['Spread'].mean()
for i, period in enumerate(christmas_easter_periods):
   for j, cap in enumerate(['Large Cap', 'Mid Cap', 'Small Cap']):
       try:
           val = means_1a.loc[period, cap]
           x_pos = i + (j - 1) * 0.15

           if cap == 'Large Cap':
               x_offset = 30
           else:
               x_offset = 8

           ax.annotate(f'{val:.3f}', xy=(x_pos, val), xytext=(x_offset, 0),
                       textcoords='offset points', ha='left', va='center',
                       fontweight='bold', fontsize=11, color='black',
                       bbox=dict(boxstyle="round,pad=0.2", fc="white", ec="none", alpha=0.7))
       except KeyError:
           continue

plt.title(f'Mean Spread (Christmas & Easter Periods) | {SPREAD_METHOD_NAME}', fontsize=18,
fontweight='bold')
plt.xlabel('Holiday Periods', fontsize=14)
plt.ylabel('Mean Relative Spread', fontsize=14)
plt.xticks(rotation=45)
plt.legend(title='Cap Group', title_fontsize=12, fontsize=11)
plt.tight_layout()
plt.savefig("output/Figure_1A_CE_Spread.png", dpi=300, bbox_inches='tight')
plt.close()



# FIGURE 1B
volume_ce = df_ce.groupby(['period_label', 'Cap_Group'],
observed=False)['Volume'].mean().reset_index()
volume_ce['Volume_millions'] = volume_ce['Volume'] / 1e6
plt.figure(figsize=(14, 8))
ax = sns.barplot(x='period_label', y='Volume_millions', hue='Cap_Group', data=volume_ce,
           hue_order=['Large Cap', 'Mid Cap', 'Small Cap'],
           palette='magma',
           errorbar='ci')

for container in ax.containers:
   ax.bar_label(container, fmt='%.1f', padding=5, fontsize=11, fontweight='bold')

plt.title(f'Mean Daily Volume (Christmas & Easter Periods) | {SPREAD_METHOD_NAME}', fontsize=18,
fontweight='bold')
```

```
plt.xlabel('Holiday Periods', fontsize=14)
plt.ylabel('Mean Daily Volume (Million Shares)', fontsize=14)
plt.xticks(rotation=45)
plt.legend(title='Cap Group', title_fontsize=12, fontsize=11)
plt.tight_layout()
plt.savefig("output/Figure_1B_CE_Volume.png", dpi=300, bbox_inches='tight')
plt.close()



# FIGURE 2: SUMMER ANALYSIS
summer_periods = ['control_year', 'summer_excl_holiday', 'summer_holiday']
df_summer = df[df['period_label'].isin(summer_periods)].copy()
df_summer['period_label'] = pd.Categorical(df_summer['period_label'],
                                           categories=summer_periods,
                                           ordered=True)

# FIGURE 2A
plt.figure(figsize=(10, 7))
ax = sns.pointplot(x='period_label', y='Spread', hue='Cap_Group', data=df_summer,
             hue_order=['Large Cap', 'Mid Cap', 'Small Cap'],
             dodge=True,
             palette='viridis',
             errorbar='ci',
             capsize=0.1)

means_2a = df_summer.groupby(['period_label', 'Cap_Group'], observed=False)['Spread'].mean()
for i, period in enumerate(summer_periods):
   for j, cap in enumerate(['Large Cap', 'Mid Cap', 'Small Cap']):
       try:
           val = means_2a.loc[period, cap]
           x_pos = i + (j - 1) * 0.15

           if cap == 'Large Cap':
               x_offset = 30
           else:
               x_offset = 8

           ax.annotate(f'{val:.3f}', xy=(x_pos, val), xytext=(x_offset, 0),
                       textcoords='offset points', ha='left', va='center',
                       fontweight='bold', fontsize=11, color='black',
                       bbox=dict(boxstyle="round,pad=0.2", fc="white", ec="none", alpha=0.7))
       except KeyError:
           continue

plt.title(f'Mean Spread (Summer Holiday) | {SPREAD_METHOD_NAME}', fontsize=18,
fontweight='bold')
plt.xlabel('Summer Periods', fontsize=14)
plt.ylabel('Mean Relative Spread', fontsize=14)
plt.xticks(rotation=0)
plt.legend(title='Cap Group', title_fontsize=12, fontsize=11)
plt.tight_layout()
plt.savefig("output/Figure_2A_Summer_Spread.png", dpi=300, bbox_inches='tight')
plt.close()



# FIGURE 2B
volume_summer = df_summer.groupby(['period_label', 'Cap_Group'],
observed=False)['Volume'].mean().reset_index()
volume_summer['Volume_millions'] = volume_summer['Volume'] / 1e6
plt.figure(figsize=(10, 7))
ax = sns.barplot(x='period_label', y='Volume_millions', hue='Cap_Group', data=volume_summer,
           hue_order=['Large Cap', 'Mid Cap', 'Small Cap'],
```

```
            palette='magma',
            errorbar='ci')

for container in ax.containers:
    ax.bar_label(container, fmt='%.1f', padding=5, fontsize=11, fontweight='bold')

plt.title(f'Mean Daily Volume (Summer Holiday) | {SPREAD_METHOD_NAME}', fontsize=18,
fontweight='bold')
plt.xlabel('Summer Periods', fontsize=14)
plt.ylabel('Mean Daily Volume (Million Shares)', fontsize=14)
plt.xticks(rotation=0)
plt.legend(title='Cap Group', title_fontsize=12, fontsize=11)
plt.tight_layout()
plt.savefig("output/Figure_2B_Summer_Volume.png", dpi=300, bbox_inches='tight')
plt.close()



# FIGURE 3: P-VALUE HEATMAPS WITH MULTIPLE TESTING CORRECTION
test_periods = ['pre_easter', 'post_easter',
                'pre_christmas', 'christmas', 'post_christmas',
                'summer_holiday', 'summer_excl_holiday']


# Ensure we only test periods that actually exist in the data
available_periods = df['period_label'].unique()
test_periods = [p for p in test_periods if p in available_periods]


p_values_spread = {}
p_values_volume = {}
all_p_spread = []
all_p_volume = []
test_info = []


# We need a temporary DF that includes all relevant periods + control
# (Previously df_ce only had Christmas/Easter)
periods_for_heatmap = ['control_year'] + test_periods
df_heatmap = df[df['period_label'].isin(periods_for_heatmap)].copy()


for group in ['Large Cap', 'Mid Cap', 'Small Cap']:
    p_values_spread[group] = {}
    p_values_volume[group] = {}

    control = df_heatmap[(df_heatmap['period_label'] == 'control_year') & (df_heatmap['Cap_Group'
] ==group)]
    for period in test_periods:
        event = df_heatmap[(df_heatmap['period_label'] == period) & (df_heatmap['Cap_Group'] == g
roup)]
        if len(event) > 1 and len(control) > 1:
            _, p_spread = ttest_ind(event['Spread'].dropna(),
                                    control['Spread'].dropna(),
                                    equal_var=False)
            _, p_volume = ttest_ind(event['Volume'].dropna(),
                                    control['Volume'].dropna(),
                                    equal_var=False)
            all_p_spread.append(p_spread)
            all_p_volume.append(p_volume)
            test_info.append((group, period))
            p_values_spread[group][period] = p_spread
            p_values_volume[group][period] = p_volume
```

```
print("\nApplying Bonferroni Correction")
print(f'Total tests performed: {len(all_p_spread)}')
if len(all_p_spread) > 0:
   print(f'Bonferroni-corrected alpha level: {0.05 / len(all_p_spread):.5f}\n')


   reject_spread_bonf, pvals_corrected_spread_bonf, _, _ = multipletests(
       all_p_spread, alpha=0.05, method='bonferroni'
   )
   reject_volume_bonf, pvals_corrected_volume_bonf, _, _ = multipletests(
       all_p_volume, alpha=0.05, method='bonferroni'
   )


   print("Applying Benjamini-Hochberg (FDR) Correction")
   reject_spread_fdr, pvals_corrected_spread_fdr, _, _ = multipletests(
       all_p_spread, alpha=0.05, method='fdr_bh'
   )
   reject_volume_fdr, pvals_corrected_volume_fdr, _, _ = multipletests(
       all_p_volume, alpha=0.05, method='fdr_bh'
   )
   print("FDR correction applied.\n")


   p_values_spread_df = pd.DataFrame(p_values_spread).T
   p_values_volume_df = pd.DataFrame(p_values_volume).T


   p_values_corrected_df_spread_bonf = pd.DataFrame(index=p_values_spread_df.index,
columns=p_values_spread_df.columns)
   p_values_corrected_df_volume_bonf = pd.DataFrame(index=p_values_volume_df.index,
columns=p_values_volume_df.columns)
   p_values_corrected_df_spread_fdr = pd.DataFrame(index=p_values_spread_df.index,
columns=p_values_spread_df.columns)
   p_values_corrected_df_volume_fdr = pd.DataFrame(index=p_values_volume_df.index,
columns=p_values_volume_df.columns)


   for i, (group, period) in enumerate(test_info):
       p_values_corrected_df_spread_bonf.loc[group, period] = pvals_corrected_spread_bonf[i]
       p_values_corrected_df_volume_bonf.loc[group, period] = pvals_corrected_volume_bonf[i]
       p_values_corrected_df_spread_fdr.loc[group, period] = pvals_corrected_spread_fdr[i]
       p_values_corrected_df_volume_fdr.loc[group, period] = pvals_corrected_volume_fdr[i]


   print("\nBonferroni Corrected P-Values (Spread)")
   print(p_values_corrected_df_spread_bonf.astype(float).round(4))
   print("\nFDR Corrected P-Values (Spread)")
   print(p_values_corrected_df_spread_fdr.astype(float).round(4))


   # Choose correction method for heatmap
   if CORRECTION_METHOD == 'Bonferroni':
       sig_matrix_spread = p_values_corrected_df_spread_bonf
       sig_matrix_volume = p_values_corrected_df_volume_bonf
   else:
       sig_matrix_spread = p_values_corrected_df_spread_fdr
       sig_matrix_volume = p_values_corrected_df_volume_fdr


   fig, ax = plt.subplots(figsize=(14, 7)) # Increased width slightly for more columns
   sns.heatmap(p_values_spread_df, annot=True, cmap='viridis_r', fmt=".3f", vmin=0, vmax=1,
```

```
                linewidths=.5, ax=ax, cbar_kws={'label': 'P-value (uncorrected)'})
    for i in range(len(sig_matrix_spread.index)):
        for j in range(len(sig_matrix_spread.columns)):
            val = sig_matrix_spread.iloc[i, j]
            if pd.notnull(val) and val < 0.05:
                ax.add_patch(plt.Rectangle((j, i), 1, 1, fill=False,
                                           edgecolor='red', lw=3))
    ax.set_title(
        f'P-values for Spread (vs. Control) | {SPREAD_METHOD_NAME}\n\nRed border = Significant
({CORRECTION_METHOD_NAME} < 0.05)',
        fontsize=16, fontweight='bold'
    )
    # Heatmaps usually don't need explicit X/Y labels if the ticks are descriptive,
    # but adding them for clarity.
    ax.set_xlabel('Holiday Events', fontsize=12, fontweight='bold')
    ax.set_ylabel('Market Cap Group', fontsize=12, fontweight='bold')
    plt.tight_layout()
    plt.savefig("output/Figure_3A_PValues_Spread.png", dpi=300, bbox_inches='tight')
    plt.close()


    fig, ax = plt.subplots(figsize=(14, 7)) # Increased width slightly
    sns.heatmap(p_values_volume_df, annot=True, cmap='plasma_r', fmt=".3f", vmin=0, vmax=1,
                linewidths=.5, ax=ax, cbar_kws={'label': 'P-value (uncorrected)'})
    for i in range(len(sig_matrix_volume.index)):
        for j in range(len(sig_matrix_volume.columns)):
            val = sig_matrix_volume.iloc[i, j]
            if pd.notnull(val) and val < 0.05:
                ax.add_patch(plt.Rectangle((j, i), 1, 1, fill=False,
                                           edgecolor='red', lw=3))
    ax.set_title(
        f'P-values for Volume (vs. Control) | {SPREAD_METHOD_NAME}\nRed border = Significant
({CORRECTION_METHOD_NAME} < 0.05)',
        fontsize=16, fontweight='bold'
    )
    ax.set_xlabel('Holiday Events', fontsize=12, fontweight='bold')
    ax.set_ylabel('Market Cap Group', fontsize=12, fontweight='bold')
    plt.tight_layout()
    plt.savefig("output/Figure_3B_PValues_Volume.png", dpi=300, bbox_inches='tight')
    plt.close()



# FIGURE 4: HEATMAP - SPREAD BY CAP AND PERIOD
period_order = ['control_year', 'pre_easter', 'post_easter',
                'pre_christmas', 'christmas', 'post_christmas',
                'summer_excl_holiday', 'summer_holiday']
period_order = [p for p in period_order if p in df['period_label'].unique()]


heatmap_data = df.groupby(['Cap_Group', 'period_label'])['Spread'].mean().unstack()
heatmap_data = heatmap_data[period_order]


fig, ax = plt.subplots(figsize=(16, 6))
sns.heatmap(heatmap_data,
            annot=True,
            fmt='.5f',
            cmap='YlOrRd',
            linewidths=0.5,
            cbar_kws={'label': 'Avg Spread'},
            ax=ax,
            annot_kws={'fontsize': 11})
ax.set_title(f'Average Spread by Cap Group and Period | {SPREAD_METHOD_NAME}',
```

```
            fontweight='bold',
            fontsize=18,
            pad=15)
ax.set_xlabel('Period', fontsize=14, fontweight='bold')
ax.set_ylabel('Market Cap Group', fontsize=14, fontweight='bold')
ax.tick_params(axis='y', labelsize=13)
ax.tick_params(axis='x', labelsize=12)
ax.set_yticklabels(ax.get_yticklabels(), rotation=0)
plt.tight_layout()
plt.savefig('output/Figure_4_Heatmap_Spread.png', dpi=300, bbox_inches='tight')
plt.close()



# FIGURE 5: TIME SERIES - MONTHLY AVERAGE SPREAD
df['YearMonth'] = df['Date'].dt.to_period('M')
monthly_spread = df.groupby('YearMonth')['Spread'].mean()


fig, ax = plt.subplots(figsize=(16, 6))
ax.plot(monthly_spread.index.to_timestamp(), monthly_spread.values,
       color='darkblue', linewidth=1.5, alpha=0.8)
ax.set_title(f'Monthly Average Spread (2014-2019) | {SPREAD_METHOD_NAME} (In-Sample)',
fontweight='bold', fontsize=16)
ax.set_xlabel('Date (Year-Month)', fontsize=14)
ax.set_ylabel('Average Relative Spread', fontsize=14)
ax.grid(alpha=0.3)
ax.legend(fontsize=12)
plt.tight_layout()
plt.savefig('output/Figure_5_Timeseries_Monthly.png', dpi=300, bbox_inches='tight')
plt.close()



# FIGURE 6: VOLUME-SPREAD RELATIONSHIP (HARRIS VALIDATION)
# Sample to avoid overplotting (take 10% random sample)
df_sample = df.sample(frac=0.1, random_state=42)


df_sample_clean = df_sample.dropna(subset=['Volume', 'Spread'])
df_sample_clean = df_sample_clean[~np.isinf(df_sample_clean['Volume'])]
df_sample_clean = df_sample_clean[~np.isinf(df_sample_clean['Spread'])]
df_sample_clean = df_sample_clean[df_sample_clean['Spread'] <= 0.20]
df_sample_clean = df_sample_clean[df_sample_clean['Volume'] > 0]


print(f"Sample size after cleaning: {len(df_sample_clean):,} observations")


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


# Scatter plot with color by cap group
for cap in ['Large Cap', 'Mid Cap', 'Small Cap']:
   data = df_sample_clean[df_sample_clean['Cap_Group'] == cap]
   ax.scatter(np.log10(data['Volume']), data['Spread'],
              alpha=0.3, s=10, label=cap)


# Add regression line
log_vol = np.log10(df_sample_clean['Volume'])
slope, intercept, r_value, p_value, std_err = linregress(log_vol, df_sample_clean['Spread'])
```

```
# Only plot trend line if r_value is valid
if not np.isnan(r_value):
   ax.plot(log_vol.sort_values(), intercept + slope*log_vol.sort_values(), 'r--', linewidth=2,
           label=f'Trend (R²={r_value**2:.3f}, p={p_value:.4f})')
else:
   print("Warning: Could not compute valid regression line")


ax.set_xlabel('Log Trading Volume (Activity)', fontsize=14)
ax.set_ylabel('Relative Spread (Cost)', fontsize=14)
ax.set_title(f'Volume-Spread Relationship | Harris Dealer Hypothesis\n{SPREAD_METHOD_NAME}',
             fontsize=16, fontweight='bold')
ax.legend()
ax.grid(alpha=0.3)


plt.tight_layout()
plt.savefig('output/Figure_6_Volume_Spread_Scatter.png', dpi=300, bbox_inches='tight')
plt.close()


# FIGURE 7: SEASONALITY (VOLUME vs VOLATILITY)
df['Month'] = df['Date'].dt.month

# Calculate Daily Returns if not already present
df['Daily_Return'] = df.groupby('Ticker')['Close'].pct_change()

# Calculate Volatility (Std Dev of Daily Returns within that month)
monthly_stats = df.groupby('Month').agg({
   'Volume': 'mean',
   'Daily_Return': lambda x: x.std() * 100
}).reset_index()

monthly_stats.columns = ['Month', 'Avg_Volume', 'Avg_Volatility']

fig, ax1 = plt.subplots(figsize=(15, 8))

color_vol = '#1e3a8a'
color_volat = '#15803d'

plt.suptitle('Seasonality of Volume vs. Volatility (Average 2014-2019)', fontsize=19,
fontweight='bold', y=1.02)
ax1.set_title('Monthly Pattern with July (Summer) Highlighted', fontsize=12, loc='left',
fontweight='bold', pad=20)


month_positions = monthly_stats['Month']
bars = ax1.bar(
   month_positions, monthly_stats['Avg_Volume'],
   color=color_vol, alpha=0.9, label='Avg Daily Volume', edgecolor='black', linewidth=1,
width=0.75
)

ylim_top = 900000
ax1.set_ylim(0, ylim_top)
summer_patch = patches.Rectangle((6.5, 0), 1, ylim_top, linewidth=0, facecolor='#fef3c7',
alpha=0.6, zorder=0)
ax1.add_patch(summer_patch)
ax1.text(7, ylim_top*0.96, 'Summer (July)', ha='center', fontsize=13, fontweight='bold',
color='#92400e',
        bbox=dict(facecolor='white', edgecolor='#92400e', alpha=1, pad=5, linewidth=2))

ax1.set_xlabel('Month (Jan-Dec)', fontsize=14, fontweight='bold')
ax1.set_ylabel('Avg Daily Volume', color=color_vol, fontsize=14, fontweight='bold')
```

```
ax1.tick_params(axis='y', labelcolor=color_vol, labelsize=12)
ax1.set_xticks(range(1, 13))
ax1.set_xticklabels(['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'Nov'
, 'Dec'], fontsize=13, fontweight='bold')
ax1.grid(axis='y', alpha=0.25, linestyle='--', linewidth=1)

ax2 = ax1.twinx()
line = ax2.plot(
   month_positions, monthly_stats['Avg_Volatility'],
   color=color_volat, linewidth=4, marker='o', markersize=11, label='Volatility %',
   markerfacecolor=color_volat, markeredgecolor='white', markeredgewidth=2.5, zorder=10
)
ax2.set_ylabel('Volatility % (Std Dev of Returns)', color=color_volat, fontsize=14,
fontweight='bold')
ax2.tick_params(axis='y', labelcolor=color_volat, labelsize=12)
ax2.set_ylim(0, 15)

highlight_months = [1, 7, 12]
for i, (x, y) in enumerate(zip(month_positions, monthly_stats['Avg_Volatility'])):
   if i+1 in highlight_months:
       ax2.annotate(f"{y:.2f}%", (x, y), textcoords="offset points", xytext=(0, 15),
ha='center',
                    fontsize=12, fontweight='bold',
                    bbox=dict(facecolor='white', edgecolor=color_volat, alpha=1, pad=4,
boxstyle='round,pad=0.5', linewidth=2.5),
                    color=color_volat, zorder=20)

lines1, labels1 = ax1.get_legend_handles_labels()
lines2, labels2 = ax2.get_legend_handles_labels()
leg = ax1.legend(lines1 + lines2, labels1 + labels2, loc='upper left', frameon=True,
fontsize=13,
                framealpha=1, edgecolor='black', fancybox=False, shadow=False)

fig.tight_layout()
plt.subplots_adjust(top=0.93)
plt.savefig('output/Figure_7_Seasonality_Vol_Volat.png', dpi=300, bbox_inches='tight')
plt.close()
print("\nAll outputs saved to 'output/' folder")
```

```
METHODOLOGICAL SPLIT-SAMPLE DESIGN IMPLEMENTED
In-Sample Data (2014-2019):     54,944 observations (Used for analysis/regression)
Out-of-Sample Data (2020-2024): 62,939 observations (Used ONLY for backtesting)

Total observations (In-Sample): 54,944
Date range: 2014-01-03 00:00:00 to 2019-12-30 00:00:00
Unique tickers: 78
Trading days: 1,499

Cohen's d Effect Sizes (vs. Control Period):
Summer Holiday: d = -0.1349
Christmas: d = 0.1539
Pre-Easter: d = -0.0293
Post-Easter: d = -0.0038

Interpretation: |d| > 0.5 = medium to large effect
HARRIS FRAMEWORK: DEALER BEHAVIOR HYPOTHESIS

Testing Harris's prediction: Spreads should increase when volume decreases
(Lower dealer participation and higher inventory risk)

Spread by Volume Quartile:
                Spread_Mean  Spread_Median  N_Obs    Avg_Volume
Volume_Quartile
Q1_Lowest          0.016526       0.010713  13736  5.310352e+03
```

```
Q2_Low              0.015488       0.010898  13736  6.209029e+04
Q3_High             0.013515       0.010392  13736  2.843725e+05
Q4_Highest          0.013557       0.009902  13736  2.483885e+06

Harris Hypothesis Test:
H0: Spread(Low Volume) = Spread(High Volume)
H1: Spread(Low Volume) > Spread(High Volume)
Mean spread Q1 (low vol): 0.016526
Mean spread Q4 (high vol): 0.013557
Difference: 0.002969
t-statistic: 15.3353
p-value: 0.0000
Cohen's d: 0.1850
✓ RESULT: Harris's prediction CONFIRMED (p < 0.05)
- Spreads are wider when volume is low (dealer participation reduced)
SUMMARY BY CAP GROUP
          Spread_Mean  Spread_Median  Spread_Std   Volume_Mean  \
Cap_Group
Large Cap     0.011279       0.008996    0.009518  1.190780e+06
Mid Cap       0.015635       0.010712    0.016556  4.235323e+05
Small Cap     0.018948       0.013665    0.019230  3.486700e+05

          Volume_Median  N_Stocks
Cap_Group
Large Cap       423292.0        28
Mid Cap          44845.0        27
Small Cap        48604.5        23

HARRIS INTERPRETATION:
Large Cap stocks have narrower spreads due to:
- Higher competitive market making (Harris Chapter 6)
- Lower inventory risk for dealers
- Greater participation rates (Harris Chapter 21)
SUMMARY BY PERIOD
                   Spread_Mean  Spread_Median    Volume_Mean  Volume_Median  \
period_label
christmas             0.017284       0.011619  750599.436798       127952.0
control_year          0.014864       0.010436  728352.300191       129768.0
post_christmas        0.017281       0.012212  750268.917503       132715.0
post_easter           0.014805       0.011243  709344.748713       158705.0
pre_christmas         0.016195       0.010901  674224.968366       125053.0
pre_easter            0.014409       0.010405  619102.659309       110863.0
summer_excl_holiday   0.014249       0.010113  687820.607110       121268.5
summer_holiday        0.012800       0.009796  574148.730505        98698.0

                   N_Obs
period_label
christmas           2136
control_year       34528
post_christmas       897
post_easter          971
pre_christmas       1138
pre_easter          1042
summer_excl_holiday 11026
summer_holiday       3206
CHANGES VS CONTROL PERIOD
                   Spread_Mean  Spread_Change_%    Volume_Mean  \
period_label
christmas             0.017284           16.28  750599.436798
control_year          0.014864            0.00  728352.300191
post_christmas        0.017281           16.26  750268.917503
post_easter           0.014805           -0.40  709344.748713
pre_christmas         0.016195            8.95  674224.968366
pre_easter            0.014409           -3.06  619102.659309
```

```
summer_excl_holiday     0.014249            -4.14  687820.607110
summer_holiday          0.012800           -13.89  574148.730505

                    Volume_Change_%
period_label
christmas                      3.05
control_year                   0.00
post_christmas                 3.01
post_easter                   -2.61
pre_christmas                 -7.43
pre_easter                   -15.00
summer_excl_holiday           -5.56
summer_holiday               -21.17
PRACTICAL IMPLICATIONS: TRADING COST ANALYSIS

Spread difference (Summer vs. Control): -0.002064
Average stock price assumed: 100 NOK

Cost Impact per Trade (one-way):
   Trade Size (NOK)    Control Cost     Summer Cost         Savings
----------------------------------------------------------------------
            100,000         743.20 NOK         640.00 NOK         103.20 NOK
            500,000        3716.00 NOK        3200.00 NOK         516.00 NOK
          1,000,000        7432.00 NOK        6400.00 NOK        1032.00 NOK
          5,000,000       37160.00 NOK       32000.00 NOK        5160.00 NOK

INTERPRETATION:
Trading during summer holidays saves approximately 13.9%
on transaction costs due to narrower spreads.

Applying Bonferroni Correction
Total tests performed: 21
Bonferroni-corrected alpha level: 0.00238

Applying Benjamini-Hochberg (FDR) Correction
FDR correction applied.


Bonferroni Corrected P-Values (Spread)
          pre_easter  post_easter  pre_christmas  christmas  post_christmas  \
Large Cap      0.0374          1.0            1.0     0.0375          0.0339
Mid Cap        1.0000          1.0            1.0     0.0016          1.0000
Small Cap      1.0000          1.0            1.0     0.0468          0.4991

          summer_holiday  summer_excl_holiday
Large Cap          0.0001               1.0000
Mid Cap            0.0007               1.0000
Small Cap          0.0000               0.0022

FDR Corrected P-Values (Spread)
          pre_easter  post_easter  pre_christmas  christmas  post_christmas  \
Large Cap      0.0047       0.2969         0.9603     0.0047          0.0047
Mid Cap        0.3819       0.6733         0.1038     0.0004          0.1038
Small Cap      0.6469       0.6733         0.1719     0.0052          0.0499

          summer_holiday  summer_excl_holiday
Large Cap          0.0001               0.1762
Mid Cap            0.0002               0.2929
Small Cap          0.0000               0.0004
Sample size after cleaning: 5,492 observations

All outputs saved to 'output/' folder
```

```
# SCRIPT 3: OLS REGRESSION ANALYSIS WITH DYNAMIC DESCRIPTIVE TITLES
INPUT_CSV = "oslo_bors_labelled_data.csv"
# Use the variable SPREAD_METHOD_NAME if defined, else default
spread_name = SPREAD_METHOD_NAME if 'SPREAD_METHOD_NAME' in locals() else "Spread"

# Create output directory if it doesn't exist
output_dir = 'output'
if not os.path.exists(output_dir):
   os.makedirs(output_dir)

# DATA LOADING & INTEGRITY CHECK
# Use 'df' from Script 2 (In-Sample 2014-2019) if available to ensure consistency.
if 'df' in locals() and isinstance(df, pd.DataFrame):
   # Double check to ensure no look-ahead bias
   max_year = df['Date'].dt.year.max()
   if max_year > 2019:
       print(f"WARNING: Data contains years up to {max_year}. Applying 2019 cutoff now.")
       df = df[df['Date'].dt.year <= 2019].copy()
else:
   print("Variable 'df' not found. Loading from CSV and applying split-sample...")
   try:
       df = pd.read_csv(INPUT_CSV, parse_dates=['Date'])
       df['Date'] = pd.to_datetime(df['Date'])
       # Apply In-Sample Filter
       df = df[df['Date'].dt.year <= 2019].copy()
   except FileNotFoundError:
       print(f"FATAL ERROR: '{INPUT_CSV}' not found.")
       pass

# Data Preparation
df = df.sort_values(['Ticker', 'Date'])
df['Return'] = df.groupby('Ticker')['Close'].pct_change()
df['Volatility'] = df.groupby('Ticker')['Return'].transform(lambda x:
x.shift(1).rolling(window=20, min_periods=10).std())
df['Log_Volume'] = np.log1p(df['Volume'])
df['Log_Volume_lag1'] = df.groupby('Ticker')['Log_Volume'].shift(1)

# Holiday and Market Cap Dummies
df['Pre_Easter'] = (df['period_label'] == 'pre_easter').astype(int)
df['Post_Easter'] = (df['period_label'] == 'post_easter').astype(int)
df['Christmas'] = (df['period_label'] == 'christmas').astype(int)
df['Pre_Christmas'] = (df['period_label'] == 'pre_christmas').astype(int)
df['Post_Christmas'] = (df['period_label'] == 'post_christmas').astype(int)
df['Summer'] = (df['period_label'] == 'summer_holiday').astype(int)
df['Large_Cap'] = (df['Cap_Group'] == 'Large Cap').astype(int)
df['Mid_Cap'] = (df['Cap_Group'] == 'Mid Cap').astype(int)

# Create regression sample
df_reg = df.dropna(subset=['Spread', 'Log_Volume_lag1', 'Volatility']).copy()

# VIF MULTICOLLINEARITY CHECK
print("\nVIF MULTICOLLINEARITY DIAGNOSTICS")
# from statsmodels.stats.outliers_influence import variance_inflation_factor (Already imported
in Script 0)

vif_vars = ['Pre_Easter', 'Post_Easter', 'Christmas', 'Pre_Christmas', 'Post_Christmas',
           'Summer', 'Log_Volume_lag1', 'Volatility', 'Large_Cap', 'Mid_Cap']
X_vif = df_reg[vif_vars].copy()
X_vif = sm.add_constant(X_vif) # Add constant for correct VIF calculation

vif_data = pd.DataFrame()
vif_data["Variable"] = X_vif.columns
vif_data["VIF"] = [variance_inflation_factor(X_vif.values, i) for i in range(X_vif.shape[1])]
vif_data = vif_data[vif_data["Variable"] != "const"].sort_values('VIF', ascending=False)
```

```
print("Variance Inflation Factors:")
print(vif_data.to_string(index=False))

high_vif = vif_data[vif_data['VIF'] > 10]
if len(high_vif) > 0:
   print("\nWARNING: High multicollinearity detected (VIF > 10)")
   print(high_vif.to_string(index=False))

# Models 1A-C: Baseline Holiday Effects by Market Cap
baseline_models = {}
cap_groups = ['Large Cap', 'Mid Cap', 'Small Cap']

for i, cap in enumerate(cap_groups):
   df_cap = df_reg[df_reg['Cap_Group'] == cap].copy()

   y = df_cap['Spread']
   X = df_cap[['Pre_Easter', 'Post_Easter', 'Christmas', 'Pre_Christmas', 'Post_Christmas', 'Sum
mer']]
   X = sm.add_constant(X)

   # Using Cluster Robust Standard Errors (HC1 equivalent with groups)
   # Note: statsmodels OLS cov_type='cluster' requires 'cov_kwds={'groups': ...}'
   model = sm.OLS(y, X).fit(cov_type='cluster', cov_kwds={'groups': df_cap['Ticker']})
   baseline_models[cap] = model

   # Create and save summary as an image with the dynamic title (v10 style)
   fig, ax = plt.subplots(figsize=(12, 10))
   ax.axis('off')
   title_text = f"Baseline Spread Regression - {cap} Stocks\nDependent Variable: {spread_name}"
   ax.text(0.5, 0.98, title_text, ha='center', va='top', fontsize=12, fontweight='bold',
transform=ax.transAxes)
   summary_text = str(model.summary())
   ax.text(0.01, 0.93, summary_text, fontdict={'fontname': 'monospace', 'fontsize': 9},
va='top', ha='left', transform=ax.transAxes)

   filename = f"output/Table_1{chr(65+i)}_Baseline_{cap.replace(' ', '_')}.png"
   plt.savefig(filename, dpi=300, bbox_inches='tight')
   plt.close()

# Model 2: Full Sample with Controls
X2 = df_reg[['Pre_Easter', 'Post_Easter', 'Christmas', 'Pre_Christmas', 'Post_Christmas', 'Summe
r',
            'Log_Volume_lag1', 'Volatility', 'Large_Cap', 'Mid_Cap']]
X2 = sm.add_constant(X2)
y2 = df_reg['Spread']

# Clustered SE
model2 = sm.OLS(y2, X2).fit(cov_type='cluster', cov_kwds={'groups': df_reg['Ticker']})

# Save Model 2
fig, ax = plt.subplots(figsize=(12, 11))
ax.axis('off')
title_text = (f'Spread Determinants with Control Variables (Full Sample)\n'
             f'Dependent Variable: {spread_name} | Controls: Lagged Volume, Volatility, Market
Cap')
ax.text(0.5, 0.98, title_text, ha='center', va='top', fontsize=12, fontweight='bold',
transform=ax.transAxes)
summary_text = str(model2.summary())
ax.text(0.01, 0.93, summary_text, fontdict={'fontname': 'monospace', 'fontsize': 9}, va='top',
ha='left', transform=ax.transAxes)
filename_2 = 'output/Table_2_Spread_Full_Controls.png'
plt.savefig(filename_2, dpi=300, bbox_inches='tight')
plt.close()
```

```
# MODEL 3: STOCK FIXED EFFECTS
# Create dummies explicitly for reporting count
stock_dummies = pd.get_dummies(df_reg['Ticker'], prefix='Stock', drop_first=True, dtype=float)

X3 = pd.concat([
   df_reg[['Pre_Easter', 'Post_Easter', 'Christmas', 'Pre_Christmas', 'Post_Christmas', 'Summer'
,
           'Log_Volume_lag1', 'Volatility']].astype(float),
   stock_dummies
], axis=1)
X3 = sm.add_constant(X3)

# Note: This might be slow with many stocks.
# If memory error, use linearmodels (Model 4) which is optimized for FE.
# But keeping OLS here as requested to match v10 output structure.
model3 = sm.OLS(df_reg['Spread'].astype(float), X3).fit(cov_type='cluster', cov_kwds={'groups':
df_reg['Ticker']})

print(f"Model 3: R-squared = {model3.rsquared:.4f}, N = {model3.nobs:.0f}")

# Key coefficients table
coef_interest = ['const', 'Pre_Easter', 'Post_Easter', 'Christmas', 'Pre_Christmas',
                 'Post_Christmas', 'Summer', 'Log_Volume_lag1', 'Volatility']
coef_table = pd.DataFrame({
   'Coefficient': model3.params[coef_interest],
   'Std Error': model3.bse[coef_interest],
   'P>|t|': model3.pvalues[coef_interest]
}).round(6)

print("\nKey Coefficients:")
print(coef_table.to_string())

# Save Model 3 (v10 style)
fig, ax = plt.subplots(figsize=(12, 10))
ax.axis('off')
title_text = f'Spread Regression with Stock Fixed Effects (In-Sample)\nDependent
Variable: {spread_name}'
ax.text(0.5, 0.98, title_text, ha='center', va='top', fontsize=12, fontweight='bold',
transform=ax.transAxes)
summary_text = f"""OLS Regression Results (with Stock FE)
R-squared: {model3.rsquared:.4f}
Observations: {model3.nobs:.0f}
Stock FE: {len(stock_dummies.columns)} dummies included

{coef_table.to_string()}

Notes: Standard errors clustered by Ticker
"""
ax.text(0.01, 0.93, summary_text, fontdict={'fontname': 'monospace', 'fontsize': 9}, va='top',
ha='left', transform=ax.transAxes)
filename_3 = 'output/Table_3_Stock_Fixed_Effects.png'
plt.savefig(filename_3, dpi=300, bbox_inches='tight')
plt.close()
print(f"SUCCESS: Saved model summary to {filename_3}")

# MODEL 4: PANEL REGRESSION
df_panel = df_reg.set_index(['Ticker', 'Date']).copy()
y_panel = df_panel['Spread']
X_panel = df_panel[['Pre_Easter', 'Post_Easter', 'Christmas', 'Pre_Christmas',
                    'Post_Christmas', 'Summer', 'Log_Volume_lag1', 'Volatility']]

# IMPORTANT: Using Entity AND Time Effects as in Table 4 v10
model4 = PanelOLS(y_panel, X_panel, entity_effects=True,
```

```
time_effects=False).fit(cov_type='clustered', cluster_entity=True)

print(f"Model 4: R-squared (within) = {model4.rsquared:.4f}, N = {model4.nobs:.0f}")
print("\nPanel Regression Coefficients:")
print(model4.summary.tables[1])

# Save Model 4 (v10 style)
fig, ax = plt.subplots(figsize=(12, 11))
ax.axis('off')
title_text = f'Panel Regression with Entity Fixed Effects (In-Sample)\nDependent
Variable: {spread_name}'
ax.text(0.5, 0.98, title_text, ha='center', va='top', fontsize=12, fontweight='bold',
transform=ax.transAxes)
summary_text = str(model4.summary)
ax.text(0.01, 0.93, summary_text, fontdict={'fontname': 'monospace', 'fontsize': 8}, va='top',
ha='left', transform=ax.transAxes)
filename_4 = 'output/Table_4_Panel_Entity_Time_FE.png'
plt.savefig(filename_4, dpi=300, bbox_inches='tight')
plt.close()

# MODEL COMPARISON
print("\nMODEL COMPARISON")
comparison_df = pd.DataFrame({
   'Model': ['Model 2 (OLS)', 'Model 3 (Stock FE)', 'Model 4 (Panel FE)'],
   'Summer Coef': [model2.params.get('Summer', np.nan), model3.params.get('Summer', np.nan),
model4.params.get('Summer', np.nan)],
   'Summer P-val': [model2.pvalues.get('Summer', np.nan), model3.pvalues.get('Summer', np.nan),
model4.pvalues.get('Summer', np.nan)],
   'R-squared': [model2.rsquared, model3.rsquared, model4.rsquared]
})
print(comparison_df.to_string(index=False))
comparison_df.to_csv('output/Model_Comparison_Summary.csv', index=False)

print("All regression tables have been saved as PNG images in the 'output' folder.")


VIF MULTICOLLINEARITY DIAGNOSTICS
Variance Inflation Factors:
      Variable      VIF
     Large_Cap 1.787024
       Mid_Cap 1.526167
Log_Volume_lag1 1.233042
    Volatility 1.029270
        Summer 1.009082
     Christmas 1.006896
 Pre_Christmas 1.003941
    Pre_Easter 1.003601
   Post_Easter 1.003429
Post_Christmas 1.003358
Model 3: R-squared = 0.2510, N = 54098

Key Coefficients:
                Coefficient  Std Error     P>|t|
const             -0.011482   0.004230  0.006635
Pre_Easter        -0.000129   0.000311  0.678315
Post_Easter        0.000368   0.000376  0.328258
Christmas          0.001904   0.000413  0.000004
Pre_Christmas      0.001148   0.000601  0.056006
Post_Christmas     0.002275   0.000505  0.000007
Summer            -0.001354   0.000299  0.000006
Log_Volume_lag1    0.001613   0.000373  0.000016
Volatility         0.054274   0.027758  0.050547
SUCCESS: Saved model summary to output/Table_3_Stock_Fixed_Effects.png
Model 4: R-squared (within) = 0.0414, N = 54098
```

```
Panel Regression Coefficients:
                              Parameter Estimates
================================================================================
                Parameter  Std. Err.     T-stat    P-value    Lower CI    Upper CI
--------------------------------------------------------------------------------
Pre_Easter        -0.0001     0.0003    -0.4178     0.6761     -0.0007      0.0005
Post_Easter        0.0004     0.0004     0.9848     0.3247     -0.0004      0.0011
Christmas          0.0019     0.0004     4.6414     0.0000      0.0011      0.0027
Pre_Christmas      0.0011     0.0006     1.9250     0.0542  -2.087e-05      0.0023
Post_Christmas     0.0023     0.0005     4.5392     0.0000      0.0013      0.0033
Summer            -0.0014     0.0003    -4.5611     0.0000     -0.0019     -0.0008
Log_Volume_lag1    0.0016     0.0004     4.3530     0.0000      0.0009      0.0023
Volatility         0.0543     0.0276     1.9697     0.0489      0.0003      0.1083
================================================================================

MODEL COMPARISON
            Model  Summer Coef  Summer P-val  R-squared
    Model 2 (OLS)    -0.001524      0.000001   0.126015
Model 3 (Stock FE)    -0.001354      0.000006   0.250991
Model 4 (Panel FE)    -0.001354      0.000005   0.041368
All regression tables have been saved as PNG images in the 'output' folder.
```

```python
# SCRIPT 4: IN-SAMPLE ROBUSTNESS & STABILITY ANALYSIS (2014-2019)

# Ensure we are using the correct In-Sample data
if 'df' in locals() and isinstance(df, pd.DataFrame):
    # Verify it's In-Sample
    max_year = df['Date'].dt.year.max()
    if max_year > 2019:
        print(f"Data includes {max_year}. Trimming to 2014-2019 for robustness checks.")
        df_robust = df[df['Date'].dt.year <= 2019].copy()
    else:
        df_robust = df.copy()
else:
    # Fallback loading
    df_robust = pd.read_csv('oslo_bors_labelled_data.csv', parse_dates=['Date'])
    df_robust['Date'] = pd.to_datetime(df_robust['Date'])
    df_robust = df_robust[df_robust['Date'].dt.year <= 2019].copy()

# Prepare variables if not present (safety check)
if 'Summer' not in df_robust.columns:
    df_robust['Summer'] = (df_robust['period_label'] == 'summer_holiday').astype(int)
if 'Log_Volume' not in df_robust.columns:
    df_robust['Log_Volume'] = np.log1p(df_robust['Volume'])

print(f"Data ready: {len(df_robust):,} observations (2014-2019)")

# ROBUSTNESS CHECK 1: YEAR-BY-YEAR STABILITY
print("CHECK 1: IS THE EFFECT CONSISTENT ACROSS YEARS? (INTERACTION ANALYSIS)")
print("Hypothesis: If the effect is structural, 'Summer' coefficient should be negative in most
years.")

# We create dummy variables for each year and interact them with Summer
# Formula: Spread ~ Summer * Year_Dummies + Controls
# This gives us the "Summer Effect" specific to each year.

years = sorted(df_robust['Date'].dt.year.unique())
results_by_year = []

print(f"{'Year':<6} {'Summer Coef':<12} {'P-Value':<10} {'Signif':<10}")
print("-" * 45)

for year in years:
```

```
    # Subset for the specific year
    df_y = df_robust[df_robust['Date'].dt.year == year].dropna(subset=['Spread', 'Log_Volume'])

    if len(df_y) < 100: continue

    # Simple regression for that year: Spread ~ Summer + Log_Volume (Control)
    X = df_y[['Summer', 'Log_Volume']]
    X = sm.add_constant(X)
    y = df_y['Spread']

    try:
        model_y = sm.OLS(y, X).fit(cov_type='HC1')
        coef = model_y.params['Summer']
        pval = model_y.pvalues['Summer']

        sig = ""
        if pval < 0.01: sig = "***"
        elif pval < 0.05: sig = "**"
        elif pval < 0.1: sig = "*"

        results_by_year.append({
            'Year': year,
            'Coef': coef,
            'Lower_CI': model_y.conf_int().loc['Summer', 0],
            'Upper_CI': model_y.conf_int().loc['Summer', 1],
            'P_Value': pval
        })

        print(f"{year:<6} {coef:<12.6f} {pval:<10.4f} {sig:<10}")

    except Exception as e:
        print(f"{year:<6} ERROR: {str(e)}")

# Plotting the Stability
res_df = pd.DataFrame(results_by_year)

fig, ax = plt.subplots(figsize=(10, 6))
ax.errorbar(res_df['Year'], res_df['Coef'],
            yerr=[res_df['Coef'] - res_df['Lower_CI'], res_df['Upper_CI'] - res_df['Coef']],
            fmt='o-', color='darkblue', capsize=5, label='Summer Effect (95% CI)')

ax.axhline(0, color='red', linestyle='--', linewidth=1)
ax.set_title('Stability Check Summer Effect Year-by-Year (In-Sample)', fontsize=14,
fontweight='bold')
ax.set_xlabel('Year', fontsize=12, fontweight='bold')
ax.set_ylabel('Coefficient Estimate (Summer Effect)', fontsize=12, fontweight='bold')
ax.grid(True, alpha=0.3)
plt.legend()
plt.savefig('output/Figure_8_Robustness_Yearly_Stability.png', dpi=300)
plt.close()


# ROBUSTNESS CHECK 2: PLACEBO TEST (RANDOM WEEKS)
print("CHECK 2: PLACEBO TEST (RANDOM PERIODS)")
print("Testing 100 random 'fake' summer periods. Real effect should be stronger (more
negative).")

n_simulations = 100
placebo_coefs = []

# Filter to non-holiday data for placebo testing to avoid contamination
df_clean = df_robust[df_robust['period_label'] == 'control_year'].copy()
sample_size = len(df_robust[df_robust['Summer'] == 1]) # Match size of actual summer
```

```
for i in range(n_simulations):
    # Assign random "Summer" dummy
    # We randomly select 4 consecutive weeks in the dataset to call "Fake Summer"
    # Simplified: Randomly shuffle the 'Summer' tag (Permutation Test)

    # Faster approach: Permutation of the variable
    fake_summer = np.random.permutation(df_robust['Summer'].values)

    # Quick regression (using numpy for speed)
    # y = Xb. Spread ~ const + fake_summer
    y = df_robust['Spread'].values
    x = fake_summer
    # Simple difference in means is equivalent to reg coef on dummy
    mean_fake_summer = y[x == 1].mean()
    mean_control = y[x == 0].mean()
    coef = mean_fake_summer - mean_control
    placebo_coefs.append(coef)

actual_summer_coef = res_df['Coef'].mean() # Average effect

# Calculate p-value from permutation
placebo_coefs = np.array(placebo_coefs)
p_perm = (placebo_coefs < actual_summer_coef).mean()

print(f"\nAverage Placebo Coefficient: {placebo_coefs.mean():.6f}")
print(f"Actual Average Summer Coefficient: {actual_summer_coef:.6f}")
print(f"Probability that random noise generates this effect: {p_perm:.4f}")

fig, ax = plt.subplots(figsize=(10, 6))
sns.histplot(placebo_coefs, kde=True, color='gray', ax=ax, label='Placebo Distribution')
ax.axvline(actual_summer_coef, color='red', linestyle='--', linewidth=2, label='Actual Summer
Effect')
ax.set_title('Robustness Placebo Test vs Actual Effect (In-Sample)', fontsize=14,
fontweight='bold')
ax.set_xlabel('Coefficient Value', fontsize=12, fontweight='bold')
ax.set_ylabel('Frequency (Count)', fontsize=12, fontweight='bold')
ax.legend()
plt.savefig('output/Figure_9_Robustness_Placebo.png', dpi=300)
plt.close()
print('Robustness checks completed and figures saved in output folder.')
```

```
Data ready: 54,944 observations (2014-2019)
CHECK 1: IS THE EFFECT CONSISTENT ACROSS YEARS? (INTERACTION ANALYSIS)
Hypothesis: If the effect is structural, 'Summer' coefficient should be negative in most years.
Year   Summer Coef  P-Value    Signif
---------------------------------------------
2014   -0.002400    0.0000     ***
2015   -0.002048    0.0012     ***
2016   -0.002909    0.0000     ***
2017   -0.001320    0.0231     **
2018   -0.001641    0.0004     ***
2019   -0.002874    0.0000     ***
CHECK 2: PLACEBO TEST (RANDOM PERIODS)
Testing 100 random 'fake' summer periods. Real effect should be stronger (more negative).

Average Placebo Coefficient: 0.000003
Actual Average Summer Coefficient: -0.002199
Probability that random noise generates this effect: 0.0000
Robustness checks completed and figures saved in output folder.
```

```
# SCRIPT 5: MICROSTRUCTURE EXECUTION BACKTEST (LIMIT VS MARKET)

# Ensure output directory exists
if not os.path.exists('output'):
    os.makedirs('output')
```

```
sns.set_theme(style="whitegrid")
plt.rcParams.update({
    'font.size': 12,
    'figure.dpi': 300,
    'axes.titlesize': 14,
    'axes.labelsize': 12
})

print("Compare Execution Quality (Effective Spread vs Quoted Spread)")
print("Theory: Harris (2003) - The Trader's Dilemma (Limit vs. Market Orders)")

# DATA LOADING & PREPARATION

try:
    df_full = pd.read_csv('oslo_bors_labelled_data.csv', parse_dates=['Date'])
except FileNotFoundError:
    print("CRITICAL ERROR: 'oslo_bors_labelled_data.csv' not found.")
    raise

df_full['Year'] = df_full['Date'].dt.year
df_full = df_full.sort_values(['Ticker', 'Date'])

# Filter: Out-of-Sample (2020-2024)
df_in_sample = df_full[df_full['Year'] <= 2019]
top_5_tickers = df_in_sample.groupby('Ticker')['Volume'].median().nlargest(5).index.tolist()

df_oos = df_full[(df_full['Year'] >= 2020) & (df_full['Ticker'].isin(top_5_tickers))].copy()

print(f"   Universe: {top_5_tickers}")
print(f"   Data Points: {len(df_oos):,} rows (2020-2024)")

# EXECUTION SIMULATION ENGINE

PORTFOLIO_CAPITAL = 1000000  # 1 Million NOK
ALLOCATION = PORTFOLIO_CAPITAL / 5
LIMIT_FILL_PROB = 0.60  # 60% Fill Rate at Midpoint

results = []
years = sorted(df_oos['Year'].unique())

for year in years:
    df_year = df_oos[df_oos['Year'] == year]

    # Week 29 Strategy
    summer_weeks = df_year[df_year['Date'].dt.isocalendar().week == 29]
    if summer_weeks.empty:
        trade_date = df_year[df_year['Date'].dt.isocalendar().week == 30]['Date'].min()
    else:
        trade_date = summer_weeks['Date'].min()

    if pd.isna(trade_date): continue

    cost_aggressive_market = 0
    cost_passive_limit = 0
    savings_spread = 0

    for ticker in top_5_tickers:
        stock_data = df_oos[df_oos['Ticker'] == ticker].sort_values('Date')

        try:
            idx = stock_data[stock_data['Date'] == trade_date].index[0]
            loc = stock_data.index.get_loc(idx)
```

```
            if loc == 0: continue

            row_prev = stock_data.iloc[loc - 1]
            row_curr = stock_data.iloc[loc]

            P_benchmark = row_prev['Close']
            P_market_close = row_curr['Close']
            Spread_pct = row_curr['Spread']

            shares = ALLOCATION / P_benchmark

            half_spread_val = (Spread_pct * P_market_close / 2) * shares
            price_drift_val = (P_market_close - P_benchmark) * shares

            # Strategy A: Market Order
            total_cost_market = price_drift_val + half_spread_val
            cost_aggressive_market += total_cost_market

            # Strategy B: Limit Order
            expected_spread_paid = (1 - LIMIT_FILL_PROB) * half_spread_val
            total_cost_limit = price_drift_val + expected_spread_paid

            cost_passive_limit += total_cost_limit
            savings_spread += (half_spread_val - expected_spread_paid)

        except (IndexError, KeyError):
            continue

    results.append({
        'Year': year,
        'Aggressive_IS_bps': (cost_aggressive_market / PORTFOLIO_CAPITAL) * 10000,
        'Passive_IS_bps': (cost_passive_limit / PORTFOLIO_CAPITAL) * 10000,
        'Spread_Savings_NOK': savings_spread
    })

df_res = pd.DataFrame(results)
df_res['Net_Improvement_bps'] = df_res['Aggressive_IS_bps'] - df_res['Passive_IS_bps']

# VISUALIZATION (FIGURE 10)
# FIGURE 10: JOINT HOLIDAY (WEEKS 28-30) VS REST OF YEAR (2020-2024)
# Prepare Data (Same logic as before)
df_fig10 = df_oos.copy()
df_fig10['Week'] = df_fig10['Date'].dt.isocalendar().week
df_fig10['Season'] = np.where(df_fig10['Week'].isin([28, 29, 30]),
                              'Joint Holiday (Week 28-30)',
                              'Rest of Year')
annual_stats = df_fig10.groupby(['Year', 'Season'])['Spread'].mean().unstack()
annual_stats['Pct_Change'] = ((annual_stats['Joint Holiday (Week 28-30)'] - annual_stats['Rest
of Year']) /annual_stats['Rest of Year']) * 100

# Create Plot with 2 Panels
fig, (ax1, ax2) = plt.subplots(2, 1, figsize=(12, 10), sharex=True,
gridspec_kw={'height_ratios': [1.5, 1]})

# PANEL A: ABSOLUTE SPREAD LEVELS
x = np.arange(len(annual_stats.index))
width = 0.35

# Plot grouped bars
rects1 = ax1.bar(x - width/2, annual_stats['Joint Holiday (Week 28-30)'] * 100, width,
                 label='Joint Holiday', color='#e74c3c', alpha=0.9, edgecolor='black',
linewidth=0.5)
rects2 = ax1.bar(x + width/2, annual_stats['Rest of Year'] * 100, width,
                 label='Rest of Year', color='#3498db', alpha=0.9, edgecolor='black',
```

```
linewidth=0.5)

ax1.set_ylabel('Average Spread (%)', fontweight='bold', fontsize=12)
ax1.set_title('Panel A: Average Spread Levels Comparison', fontweight='bold', fontsize=14,
loc='left')
ax1.legend(loc='upper right', framealpha=0.95)
ax1.grid(axis='y', alpha=0.3, linestyle='--')
ax1.set_ylim(0, 2.5)

# Add value labels on top of bars
def autolabel(rects):
   for rect in rects:
       height = rect.get_height()
       ax1.annotate(f'{height:.2f}%',
                    xy=(rect.get_x() + rect.get_width() / 2, height),
                    xytext=(0, 3),
                    textcoords="offset points",
                    ha='center', va='bottom', fontsize=10)
autolabel(rects1)
autolabel(rects2)

# PANEL B: RELATIVE COST CHANGE
change_vals = annual_stats['Pct_Change']
colors = ['#c0392b' if v > 0 else '#27ae60' for v in change_vals]

bars = ax2.bar(x, change_vals, width=0.5, color=colors, alpha=0.85, edgecolor='black',
linewidth=0.8)

ax2.axhline(0, color='black', linewidth=1)
ax2.set_ylabel('Relative Change (%)', fontweight='bold', fontsize=12)
ax2.set_xlabel('Year', fontweight='bold', fontsize=12)
ax2.set_title('Panel B: Relative Cost Increase During Holiday (%)', fontweight='bold',
fontsize=14, loc='left')
ax2.set_xticks(x)
ax2.set_xticklabels(annual_stats.index.astype(int), fontsize=12)
ax2.grid(axis='y', alpha=0.3, linestyle='--')
ax2.set_xlim(-0.5, len(x) - 0.5)
ax2.set_ylim(-35, 5)

# Improved label placement for Panel B (avoiding overlap with axis)
for bar in bars:
   height = bar.get_height()
   label_y = height + (1 if height >= 0 else -3)

   ax2.text(bar.get_x() + bar.get_width()/2, label_y,
            f'{height:+.1f}%',
            ha='center', va='bottom' if height > 0 else 'top',
            fontweight='bold', fontsize=11, color='#2c3e50')

# Final Layout
plt.suptitle('Liquidity Cost Analysis: Joint Holiday vs. Rest of Year',
            fontsize=16, fontweight='bold', y=0.96)
plt.tight_layout()
plt.subplots_adjust(top=0.90)

plt.savefig('output/Figure_10_Joint_Holiday_vs_Rest.png', dpi=300)
plt.close()




# FIGURE 11: BACKTEST RESULTS (CLEAN 2-PANEL LAYOUT)
fig, (ax1, ax2) = plt.subplots(2, 1, figsize=(12, 10), sharex=True,
gridspec_kw={'height_ratios': [1.5, 1]})
```

```
# PANEL A: TRANSACTION COSTS (Comparison)
years = df_res['Year']
x = np.arange(len(years))
width = 0.35

# Plot bars
rects1 = ax1.bar(x - width/2, df_res['Aggressive_IS_bps'], width,
                 label='Aggressive Market Order', color='#e74c3c', alpha=0.9, edgecolor='black',
linewidth=0.5)
rects2 = ax1.bar(x + width/2, df_res['Passive_IS_bps'], width,
                 label='Passive Limit Order', color='#27ae60', alpha=0.9, edgecolor='black',
linewidth=0.5)

# Styling Panel A
ax1.set_ylabel('Transaction Cost (bps)\n(Neg = Price Improvement)', fontweight='bold',
fontsize=12)
ax1.set_title('Panel A: Execution Cost Comparison (Limit vs. Market)', fontweight='bold',
fontsize=14, loc='left')
ax1.axhline(0, color='black', linewidth=1)
ax1.legend(loc='upper left', framealpha=0.95)
ax1.grid(axis='y', alpha=0.3, linestyle='--')

# Add value labels on bars (optional, kept small)
def autolabel(rects, ax):
    for rect in rects:
        height = rect.get_height()
        offset = 5 if height >= 0 else -15
        ax.annotate(f'{height:.0f}',
                    xy=(rect.get_x() + rect.get_width() / 2, height),
                    xytext=(0, offset),
                    textcoords="offset points",
                    ha='center', va='bottom', fontsize=9, color='black')
autolabel(rects1, ax1)
autolabel(rects2, ax1)

# PANEL B: NET SAVINGS (Liquidity Alpha)
savings = df_res['Net_Improvement_bps']
colors = ['#2980b9' if s >= 0 else '#c0392b' for s in savings]

# Plot bars for savings
bars = ax2.bar(x, savings, width=0.5, color=colors, alpha=0.8, edgecolor='black', linewidth=0.8)

# Styling Panel B
ax2.set_ylabel('Net Savings (bps)', fontweight='bold', fontsize=12)
ax2.set_xlabel('Year', fontweight='bold', fontsize=12)
ax2.set_title('Panel B: "Liquidity Alpha" - Net Savings from Passive Strategy',
fontweight='bold', fontsize=14, loc='left')
ax2.axhline(0, color='black', linewidth=1)
ax2.set_xticks(x)
ax2.set_xticklabels(years.astype(int), fontsize=12)
ax2.grid(axis='y', alpha=0.3, linestyle='--')
ax2.set_xlim(-0.5, len(x) - 0.5)

# Add distinct labels for savings
for bar in bars:
    height = bar.get_height()
    label_y = height + (2 if height >= 0 else -5)
    ax2.text(bar.get_x() + bar.get_width()/2, label_y,
             f'+{height:.1f} bps',
             ha='center', va='bottom', fontweight='bold', fontsize=11, color='#2c3e50')

# Final layout adjustments
plt.suptitle('Backtest Results: Week 29 Execution Strategy', fontsize=16, fontweight='bold',
```

```
y=0.96)
plt.tight_layout()
plt.subplots_adjust(top=0.90)

# Save
plt.savefig('output/Figure_11_Backtest_Results.png', dpi=300)
plt.close()


avg_aggressive = df_res['Aggressive_IS_bps'].mean()
avg_passive = df_res['Passive_IS_bps'].mean()
total_savings_nok = df_res['Spread_Savings_NOK'].sum()

report = f"""
FINAL REPORT: EXECUTION STRATEGY ANALYSIS (HARRIS THEORY)
------------------------------------------------------------
1. THEORETICAL FRAMEWORK
  - Tested "Trader's Dilemma": Paying for Immediacy (Market Orders)
    vs. Risking Non-Execution for Price Improvement (Limit Orders).
  - Focus: Reducing Transaction Costs (TCA) in the Summer Period.

2. RESULTS (2020-2024)
  - Average Cost (Aggressive): {avg_aggressive:.1f} bps
  - Average Cost (Passive):    {avg_passive:.1f} bps
  - Net Improvement (Alpha):   {avg_aggressive - avg_passive:.1f} bps

3. ECONOMIC SIGNIFICANCE
  - By using Passive Limit Orders instead of Aggressive Market Orders during
    the summer, the investor captured a "Liquidity Premium".
  - Total Estimated Savings (spread capture): {total_savings_nok:,.0f} NOK
"""

with open('output/Final_Execution_Report.txt', 'w') as f:
   f.write(report)

print(report)
print("Final execution strategy report saved to 'output/Final_Execution_Report.txt'")
```

```
Compare Execution Quality (Effective Spread vs Quoted Spread)
Theory: Harris (2003) - The Trader's Dilemma (Limit vs. Market Orders)
  Universe: ['NHY.OL', 'DNO.OL', 'NEL.OL', 'EQNR.OL', 'RECSI.OL']
  Data Points: 3,178 rows (2020-2024)

FINAL REPORT: EXECUTION STRATEGY ANALYSIS (HARRIS THEORY)
------------------------------------------------------------
1. THEORETICAL FRAMEWORK
  - Tested "Trader's Dilemma": Paying for Immediacy (Market Orders)
    vs. Risking Non-Execution for Price Improvement (Limit Orders).
  - Focus: Reducing Transaction Costs (TCA) in the Summer Period.

2. RESULTS (2020-2024)
  - Average Cost (Aggressive): 64.7 bps
  - Average Cost (Passive):    45.8 bps
  - Net Improvement (Alpha):   18.9 bps

3. ECONOMIC SIGNIFICANCE
  - By using Passive Limit Orders instead of Aggressive Market Orders during
    the summer, the investor captured a "Liquidity Premium".
  - Total Estimated Savings (spread capture): 9,441 NOK

Final execution strategy report saved to 'output/Final_Execution_Report.txt'
```

```
# SCRIPT 6: OUT-OF-SAMPLE VALIDATION (2020-2024)
# FIGURE 12A & 12B: OUT-OF-SAMPLE VALIDATION (2020-2024)
```

```
df_oos_full = df_full[df_full['Year'] >= 2020].copy()

# Define Events (Same as Script 3)
test_periods = ['pre_easter', 'post_easter',
                'pre_christmas', 'christmas', 'post_christmas',
                'summer_holiday', 'summer_excl_holiday']

# Ensure we only test periods that actually exist in the OOS data
available_periods_oos = df_oos_full['period_label'].unique()
test_periods_oos = [p for p in test_periods if p in available_periods_oos]

p_values_spread_oos = {}
p_values_volume_oos = {}
all_p_spread_oos = []
all_p_volume_oos = []
test_info_oos = []

periods_for_heatmap_oos = ['control_year'] + test_periods_oos
df_heatmap_oos = df_oos_full[df_oos_full['period_label'].isin(periods_for_heatmap_oos)].copy()

# RUN T-TESTS
for group in ['Large Cap', 'Mid Cap', 'Small Cap']:
   p_values_spread_oos[group] = {}
   p_values_volume_oos[group] = {}


   control = df_heatmap_oos[
       (df_heatmap_oos['period_label'] == 'control_year') &
       (df_heatmap_oos['Cap_Group'] == group)
   ]

   for period in test_periods_oos:
       event = df_heatmap_oos[
           (df_heatmap_oos['period_label'] == period) &
           (df_heatmap_oos['Cap_Group'] == group)
       ]

       if len(event) > 1 and len(control) > 1:
           _, p_spread = ttest_ind(event['Spread'].dropna(),
                                   control['Spread'].dropna(),
                                   equal_var=False)
           _, p_volume = ttest_ind(event['Volume'].dropna(),
                                   control['Volume'].dropna(),
                                   equal_var=False)

           all_p_spread_oos.append(p_spread)
           all_p_volume_oos.append(p_volume)
           test_info_oos.append((group, period))

           p_values_spread_oos[group][period] = p_spread
           p_values_volume_oos[group][period] = p_volume
       else:
           p_values_spread_oos[group][period] = np.nan
           p_values_volume_oos[group][period] = np.nan

# FDR CORRECTION
print("Applying FDR Correction (Benjamini-Hochberg) to OOS Results...")

reject_spread, pvals_corr_spread, _, _ = multipletests(all_p_spread_oos, alpha=0.05,
method='fdr_bh')
reject_vol, pvals_corr_vol, _, _ = multipletests(all_p_volume_oos, alpha=0.05, method='fdr_bh')

p_vals_spread_df_oos = pd.DataFrame(p_values_spread_oos).T
p_vals_volume_df_oos = pd.DataFrame(p_values_volume_oos).T
```

```
sig_matrix_spread_oos = pd.DataFrame(index=p_vals_spread_df_oos.index,
columns=p_vals_spread_df_oos.columns)
sig_matrix_volume_oos = pd.DataFrame(index=p_vals_volume_df_oos.index,
columns=p_vals_volume_df_oos.columns)

for i, (group, period) in enumerate(test_info_oos):
   sig_matrix_spread_oos.loc[group, period] = pvals_corr_spread[i]
   sig_matrix_volume_oos.loc[group, period] = pvals_corr_vol[i]


fig, ax = plt.subplots(figsize=(14, 7))
sns.heatmap(p_vals_spread_df_oos.astype(float), annot=True, cmap='viridis_r', fmt=".3f", vmin=0,
vmax=1,
            linewidths=.5, ax=ax, cbar_kws={'label': 'P-value (OOS 2020-2024)'})

for i in range(len(sig_matrix_spread_oos.index)):
   for j in range(len(sig_matrix_spread_oos.columns)):
       val = sig_matrix_spread_oos.iloc[i, j]
       if pd.notnull(val) and val < 0.05:
           ax.add_patch(patches.Rectangle((j, i), 1, 1, fill=False,
                                       edgecolor='red', lw=3))

ax.set_title(
   f'Out-of-Sample (2020-2024) P-values for Spread\nRed border = Significant (FDR < 0.05)',
   fontsize=16, fontweight='bold'
)
ax.set_xlabel('Holiday Events', fontsize=12, fontweight='bold')
ax.set_ylabel('Market Cap Group', fontsize=12, fontweight='bold')
plt.tight_layout()
plt.savefig("output/Figure_12A_OOS_PValues_Spread.png", dpi=300, bbox_inches='tight')
plt.close()

fig, ax = plt.subplots(figsize=(14, 7))
sns.heatmap(p_vals_volume_df_oos.astype(float), annot=True, cmap='plasma_r', fmt=".3f", vmin=0,
vmax=1,
            linewidths=.5, ax=ax, cbar_kws={'label': 'P-value (OOS 2020-2024)'})

for i in range(len(sig_matrix_volume_oos.index)):
   for j in range(len(sig_matrix_volume_oos.columns)):
       val = sig_matrix_volume_oos.iloc[i, j]
       if pd.notnull(val) and val < 0.05:
           ax.add_patch(patches.Rectangle((j, i), 1, 1, fill=False,
                                       edgecolor='red', lw=3))

ax.set_title(
   f'Out-of-Sample (2020-2024) P-values for Volume\nRed border = Significant (FDR < 0.05)',
   fontsize=16, fontweight='bold'
)
ax.set_xlabel('Holiday Events', fontsize=12, fontweight='bold')
ax.set_ylabel('Market Cap Group', fontsize=12, fontweight='bold')
plt.tight_layout()
plt.savefig("output/Figure_12B_OOS_PValues_Volume.png", dpi=300, bbox_inches='tight')
plt.close()

print("Figures 12A & 12B saved to 'output' folder.")
```

```
Applying FDR Correction (Benjamini-Hochberg) to OOS Results...
Figures 12A & 12B saved to 'output' folder.
```

```
# SCRIPT 7: BONUS TABLE
# TABLE FOR FIGURE 2A (WITH BOTH VARIANCES)

summer_stats = df_summer.groupby(['period_label', 'Cap_Group'], observed=True)['Spread'].agg(
   ['mean', 'count', 'std', 'var']
```

```
)

summer_stats['pop_var'] = summer_stats['var'] * (summer_stats['count'] - 1) / summer_stats['coun
t']

def get_ci(row):
   t_crit = stats.t.ppf(0.975, row['count'] - 1)
   margin_error = t_crit * (row['std'] / np.sqrt(row['count']))
   return pd.Series([row['mean'] - margin_error, row['mean'] + margin_error],
index=['CI_Lower', 'CI_Upper'])

ci_df = summer_stats.apply(get_ci, axis=1)
summer_table = pd.concat([summer_stats, ci_df], axis=1)

output_table = summer_table[['mean', 'pop_var', 'var', 'CI_Lower', 'CI_Upper', 'count']].copy()
output_table.columns = ['Mean', 'Pop. Variance (σ²)', 'Sample Var (S²)', '95% CI Lower', '95% CI
Upper', 'N']

period_totals = []
for period in output_table.index.get_level_values('period_label').unique():
   period_data_output = output_table.xs(period, level='period_label')
   period_data_raw = df_summer[df_summer['period_label'] == period]['Spread']

   total_n = period_data_output['N'].sum()
   weighted_mean = (period_data_output['Mean'] * period_data_output['N']).sum() / total_n

   pooled_variance = period_data_raw.var()
   pop_variance = pooled_variance * (total_n - 1) / total_n
   pooled_std = np.sqrt(pooled_variance)

   t_crit = stats.t.ppf(0.975, total_n - 1)
   margin_error = t_crit * (pooled_std / np.sqrt(total_n))
   ci_lower = weighted_mean - margin_error
   ci_upper = weighted_mean + margin_error

   period_totals.append((period, 'All Cap', weighted_mean, pop_variance, pooled_variance,
ci_lower, ci_upper, total_n))

period_totals_df = pd.DataFrame(period_totals,
columns=['period_label', 'Cap_Group', 'Mean', 'Pop. Variance (σ²)', 'Sample Var (S²)', '95% CI
Lower', '95% CI Upper', 'N'])
period_totals_df = period_totals_df.set_index(['period_label', 'Cap_Group'])

output_table_regular = output_table.copy()
output_table_totals = period_totals_df.copy()

output_table_display_regular = output_table_regular.copy()
output_table_display_regular['Mean'] = output_table_display_regular['Mean'].apply(lambda x: f"{x
:.5f}")
output_table_display_regular['Pop. Variance (σ²)'] = output_table_display_regular['Pop. Variance
(σ²)'].apply(lambda x: f"{x:.8f}")
output_table_display_regular['Sample Var (S²)'] = output_table_display_regular['Sample Var
(S²)'].apply(lambda x: f"{x:.8f}")
output_table_display_regular['95% CI Lower'] = output_table_display_regular['95% CI
Lower'].apply(lambda x: f"{x:.5f}")
output_table_display_regular['95% CI Upper'] = output_table_display_regular['95% CI
Upper'].apply(lambda x: f"{x:.5f}")
output_table_display_regular['N'] = output_table_display_regular['N'].apply(lambda x: f"{int(x):
,}")
output_table_display_regular = output_table_display_regular.reset_index()

output_table_display_totals = output_table_totals.copy()
output_table_display_totals['Mean'] = output_table_display_totals['Mean'].apply(lambda x: f"{x:.
5f}")
```

```
output_table_display_totals['Pop. Variance (σ²)'] = output_table_display_totals['Pop. Variance
(σ²)'].apply(lambda x: f"{x:.8f}")
output_table_display_totals['Sample Var (S²)'] = output_table_display_totals['Sample Var
(S²)'].apply(lambda x: f"{x:.8f}")
output_table_display_totals['95% CI Lower'] = output_table_display_totals['95% CI
Lower'].apply(lambda x: f"{x:.5f}")
output_table_display_totals['95% CI Upper'] = output_table_display_totals['95% CI
Upper'].apply(lambda x: f"{x:.5f}")
output_table_display_totals['N'] = output_table_display_totals['N'].apply(lambda x: f"{int(x):,}
")
output_table_display_totals = output_table_display_totals.reset_index()

output_table_display = pd.concat([output_table_display_regular, output_table_display_totals],
ignore_index=True)

fig, ax = plt.subplots(figsize=(15, 6))
ax.axis('tight')
ax.axis('off')

table = ax.table(cellText=output_table_display.values,
                 colLabels=output_table_display.columns,
                 cellLoc='center',
                 loc='center',
                 colWidths=[0.12, 0.12, 0.12, 0.14, 0.14, 0.12, 0.12, 0.08])

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

for i in range(len(output_table_display.columns)):
    table[(0, i)].set_facecolor('#3498db')
    table[(0, i)].set_text_props(weight='bold', color='white')

for i in range(1, len(output_table_display) + 1):
    if i % 2 == 0:
        for j in range(len(output_table_display.columns)):
            table[(i, j)].set_facecolor('#ecf0f1')

num_regular_rows = len(output_table_display_regular)
for i in range(num_regular_rows + 1, len(output_table_display) + 1):
    for j in range(len(output_table_display.columns)):
        table[(i, j)].set_text_props(weight='bold')

plt.title('Table: Descriptive Statistics (Comparing Variances) for Figure 2A',
          fontsize=14, fontweight='bold', pad=20)
plt.tight_layout()
plt.savefig('output/Table_Figure_2A_Stats_Both_Variances.png', dpi=300, bbox_inches='tight')
plt.close()

print("Table saved as 'output/Table_Figure_2A_Stats_Both_Variances.png'")

# ADDITIONAL FIGURE: AVERAGE MONTHLY PASSENGERS PER AIRPORT (2014-2019)

ap = pd.read_csv('/Users/tesh/Downloads/Flydata.csv', header=None)

ap.columns = ['Airport', 'TrafficType', 'RouteType', 'PassengerGroup', 'Date', 'Passengers']

ap['Airport'] = ap['Airport'].ffill()
ap['TrafficType'] = ap['TrafficType'].ffill()
ap['RouteType'] = ap['RouteType'].ffill()
ap['PassengerGroup'] = ap['PassengerGroup'].ffill()

ap = ap[ap['Date'].str.contains(r'\d{4}M\d{2}', na=False)].copy()
```

```
ap['Passengers'] = pd.to_numeric(ap['Passengers'], errors='coerce')
ap['Date'] = pd.to_datetime(ap['Date'].str.replace('M', '-'), format='%Y-%m')
ap['Year'] = ap['Date'].dt.year
ap['Month'] = ap['Date'].dt.month

ap = ap[(ap['Year'] >= 2014) & (ap['Year'] <= 2019)]

ap_monthly_avg = ap.groupby('Month')['Passengers'].mean().reset_index()

plt.figure(figsize=(10, 6))
plt.plot(ap_monthly_avg['Month'], ap_monthly_avg['Passengers'],
        marker='o', color='#00509e', linewidth=2)

plt.title('Average Monthly Passengers per Airport (2014-2019)', fontsize=14, fontweight='bold')
plt.xlabel('Month', fontsize=12)
plt.ylabel('Average Passengers', fontsize=12)
plt.xticks(ap_monthly_avg['Month'], [calendar.month_name[i] for i in ap_monthly_avg['Month']],
rotation=45)
plt.grid(True, linestyle='--', alpha=0.7)

plt.tight_layout()
plt.savefig('output/Average_Monthly_Passengers_Per_Airport.png', dpi=300)
plt.close()
```

```
Table saved as 'output/Table_Figure_2A_Stats_Both_Variances.png'
```

```
# SCRIPT 8: BONUS FIGURES - 2D and 3D FORECAST MODEL

# FIGURE 13: Forecast Model at k=0.95 (2D Slice)
p_values = np.linspace(0, 1, 100)

# At k = 0.95, the formulas are:
# ΔITC_upper(ρ) = ρ × 0.95 × 17%
# ΔITC_lower(ρ) = ρ × 0.95 × 11%
# These are NEGATIVE (cost reductions/savings)
k_fixed = 0.95
ITC_upper = -p_values * k_fixed * 0.17
ITC_lower = -p_values * k_fixed * 0.11

fig, ax = plt.subplots(figsize=(14, 8))

ax.plot(p_values, ITC_upper * 100,
       color='#e74c3c',
       linewidth=3.5,
       label='ΔITC%$_{upper}$(ρ) = ρ*0.95*17%',
       linestyle='-',
       zorder=3)


ax.plot(p_values, ITC_lower * 100,
       color='#27ae60',
       linewidth=3.5,
       label='ΔITC%$_{lower}$(ρ) = ρ*0.95*11%',
       linestyle='-',
       zorder=3)


ax.fill_between(p_values, ITC_lower * 100, ITC_upper * 100,
               alpha=0.3,
               color='#5dade2',
               label='95% Confidence Interval',
               zorder=1)


ax.grid(True, alpha=0.4, linestyle='--', linewidth=0.8, color='gray', zorder=0)
```

```
ax.set_xlabel('ρ, shrinkage parameter', fontsize=14, fontweight='bold')
ax.set_ylabel('y, ΔITC%', fontsize=14, fontweight='bold')
ax.set_title('ITC Forecast at k=0.95\nTransaction Cost Reduction vs Shrinkage Parameter',
             fontsize=16, fontweight='bold', pad=20)


ax.set_xlim(0, 1)
ax.set_ylim(-18, 0)
ax.set_xticks([0, 0.2, 0.4, 0.6, 0.8, 1.0])
ax.set_yticks([0, -2, -4, -6, -8, -10, -12, -14, -16, -18])
ax.tick_params(axis='both', labelsize=11)
ax.axhline(y=0, color='black', linewidth=1.5, linestyle='-', zorder=2)
ax.legend(loc='lower left', framealpha=0.95, fontsize=12,
          edgecolor='black', fancybox=False)


ax.annotate(f'Full Effect (ρ=1)\nSavings: 10.5% - 16.2%',
            xy=(1.0, -13.5),
            xytext=(0.65, -5),
            fontsize=11,
            fontweight='bold',
            ha='center',
            bbox=dict(boxstyle='round,pad=0.5', facecolor='white',
                      edgecolor='black', linewidth=1.5),
            arrowprops=dict(arrowstyle='->', lw=2, color='black'))

textstr = ('Interpretation:\n'
           'k = 0.95: Assumes 5% price improvement\n'
           'ρ = 0: No historical effect persists\n'
           'ρ = 1: Full in-sample effect carries over\n'
           'Shaded area: 95% confidence interval\n'
           'Negative values = cost savings')

ax.text(0.02, 0.98, textstr,
        transform=ax.transAxes,
        fontsize=10,
        verticalalignment='top',
        bbox=dict(boxstyle='round,pad=0.8', facecolor='wheat', alpha=0.9,
                  edgecolor='black', linewidth=1.5))

plt.tight_layout()
plt.savefig('output/Figure13_ForecastModel_k095_2D.png', dpi=300, bbox_inches='tight')
plt.close()

print("Figure 13 saved as 'output/Figure13_ForecastModel_k095_2D.png'")

# FIGURE 14: 3D Forecast Model - BALANCED TEXT BOX POSITION
p_values = np.linspace(0, 1, 50)
k_values = np.linspace(0, 1, 50)
P, K = np.meshgrid(p_values, k_values)

ITC_lower = 0.11 * P * K
ITC_upper = 0.17 * P * K

fig = plt.figure(figsize=(18, 12))
ax = fig.add_subplot(111, projection='3d')

surf_upper = ax.plot_surface(P, K, ITC_upper,
                             color='#e74c3c', alpha=0.75,
                             edgecolor='none', shade=True, antialiased=True)

surf_lower = ax.plot_surface(P, K, ITC_lower,
```

```
                            color='#27ae60', alpha=0.75,
                            edgecolor='none', shade=True, antialiased=True)

ax.plot_surface(P, K, (ITC_upper + ITC_lower) / 2,
                color='#5dade2', alpha=0.25, edgecolor='none', antialiased=True)

ax.set_xlabel('\nShrinkage Parameter (ρ)', fontsize=14, fontweight='bold', labelpad=15)
ax.set_ylabel('\nPrice Improvement (k)', fontsize=14, fontweight='bold', labelpad=15)

ax.set_title('Forecast Model (3D)\nITC Reduction as Function of ρ and k',
             fontsize=18, fontweight='bold', pad=30)


ax.set_xlim(0, 1)
ax.set_ylim(0, 1)
ax.set_zlim(0, 0.18)


ax.set_xticks([0, 0.2, 0.4, 0.6, 0.8, 1.0])
ax.set_yticks([0, 0.2, 0.4, 0.6, 0.8, 1.0])
ax.set_zticks([0, 0.02, 0.04, 0.06, 0.08, 0.10, 0.12, 0.14, 0.16, 0.18])

ax.tick_params(axis='x', labelsize=11, pad=8)
ax.tick_params(axis='y', labelsize=11, pad=8)
ax.tick_params(axis='z', labelsize=11, pad=8)
ax.zaxis.set_major_formatter(plt.FuncFormatter(lambda z, _: f'{z:.2f}'))
ax.view_init(elev=20, azim=135)
ax.grid(True, alpha=0.4, linestyle='--', linewidth=0.8, color='gray')


ax.xaxis.pane.fill = False
ax.yaxis.pane.fill = False
ax.zaxis.pane.fill = False
ax.xaxis.pane.set_edgecolor('gray')
ax.yaxis.pane.set_edgecolor('gray')
ax.zaxis.pane.set_edgecolor('gray')


ax.text2D(1.05, 0.5, 'ΔITC: Transaction Cost Reduction',
          transform=ax.transAxes,
          fontsize=14,
          fontweight='bold',
          rotation=90,
          verticalalignment='center',
          horizontalalignment='left')


legend_elements = [
   Patch(facecolor='#e74c3c', alpha=0.75, edgecolor='black',
         label='Upper Bound: ρ × k × 17%'),
   Patch(facecolor='#27ae60', alpha=0.75, edgecolor='black',
         label='Lower Bound: ρ × k × 11%'),
   Patch(facecolor='#5dade2', alpha=0.25, edgecolor='black',
         label='95% Confidence Region')
]
ax.legend(handles=legend_elements, loc='upper left', fontsize=12,
          framealpha=0.95, edgecolor='black', fancybox=False, bbox_to_anchor=(0.02, 0.98))


textstr = ('Key Insights:\n'
           '11-17%: In-sample spread reduction (2014-2019)\n'
           'ρ = 1: Full effect persists out-of-sample\n'
           'k = 1: No price improvement\n'
           'At k=0.95, ρ=1: Savings of 10.5-16.2%')
```

```
fig.text(0.18, 0.02, textstr, fontsize=11, verticalalignment='bottom',
        bbox=dict(boxstyle='round,pad=0.8', facecolor='wheat', alpha=0.9, edgecolor='black',
linewidth=1.5))


plt.subplots_adjust(left=0.05, right=0.87, top=0.95, bottom=0.08)

plt.savefig('output/Figure14_3D_ForecastModel.png', dpi=300, bbox_inches='tight',
pad_inches=0.5)
plt.close()
print("Figure 14 saved as 'output/Figure14_3D_ForecastModel.png'")
```

```
Figure 13 saved as 'output/Figure13_ForecastModel_k095_2D.png'
Figure 14 saved as 'output/Figure14_3D_ForecastModel.png'
```