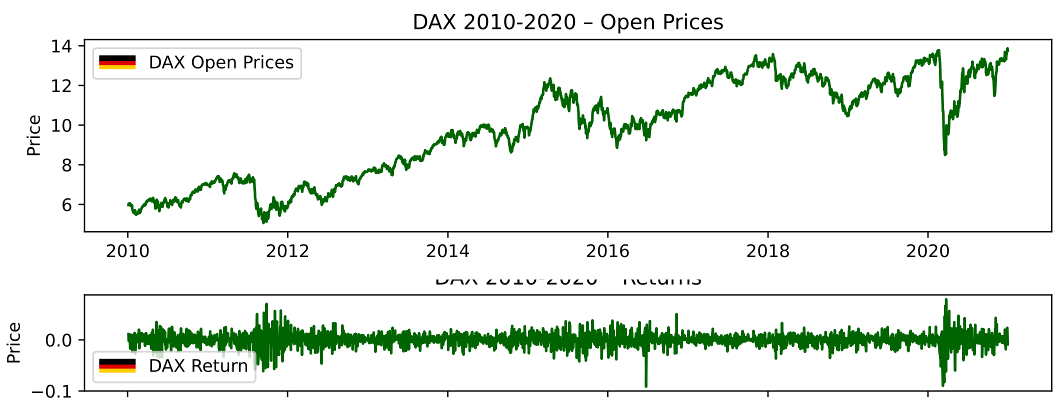
STATS 531 - Midterm Project

We look into the DAX Stock Index in Germany, and investigate how we can model the stock's opening prices, volatility, and other features. The DAX index tracks the performance of the 40 largest companies listed on the Frankfurt Stock Exchange (STOXX Ltd. 2026). They represent the diversified economy of Germany. The DAX is generally known as the German equity benchmark, surviving for over 30 years. What are the best ways to model changes in the stock's price and its returns? How do we model its volatility, and how accurately?

1 Exploratory Data Analysis

1.0.1 Introduction

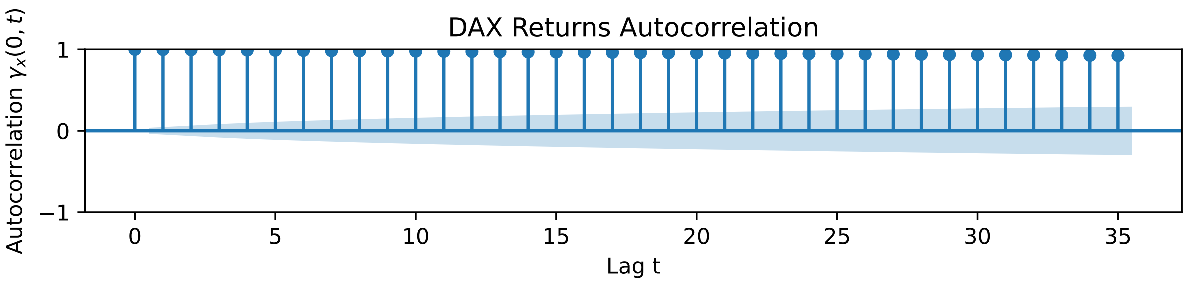
We look at the daily stock price open for the DAX index for every day from 2010 to 2020 through a time series graph. The data above for stock open shows a roughly nonstationary data. The mean is not constant, and we have a large dip in 2020, associated with covid (Caporale and Gil-Alana 2022). Then we analyze volatility for the DAX stock, $y_n = log(z_n) - log(z_{n-1})$ (Ionides 2026b).



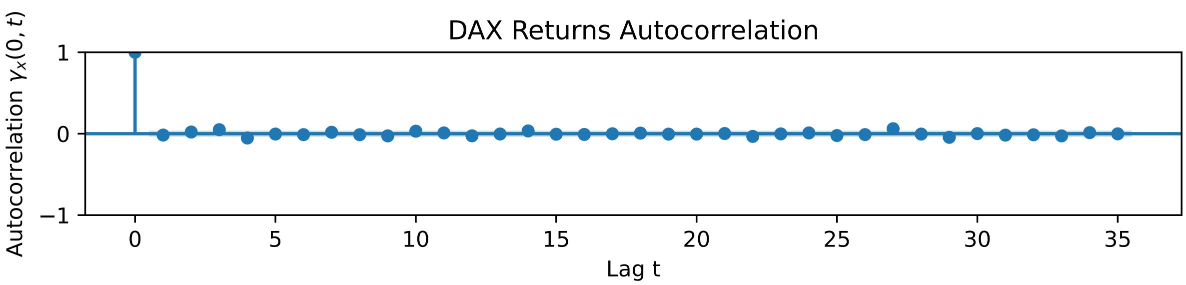
Through visual inspection, volatility is stationary with a more or less constant mean.

1.0.2 Autocorrelation

We look at the autocorrelation function below to see how much lags depend on each other. DAX opening prices are all highly correlated from the original value down until the 35th lag. Correlation then drops slowly. This tells us that a model that depends on its lags seems suitable.

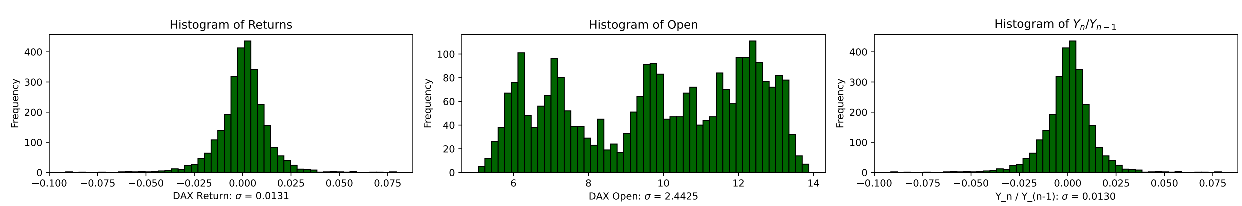


The autocorrelation graph for DAX Returns is below. The correlation of the returns are such that they could be i.i.d.. This means that the relationships between the first price return and its lags aren't as heavily correlated.



1.0.3 Distributions

Now we show the histogram for the opening prices, returns, and volatility. These display the shapes of our data. While returns and volatility are normally distributed, opening prices are more uniformally distributed.



1.0.4 Stationary vs Non Stationary

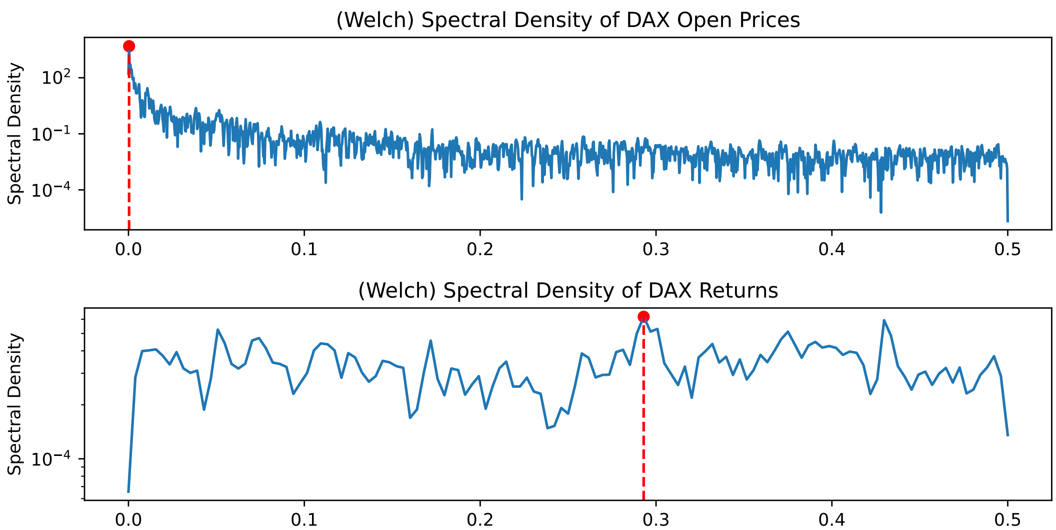
The time series graphs for DAX Opening prices look nonstationary through visual inspection, and the returns for DAX by day look stationary. We obtain empirical data to show whether the data is stationary or not through the Augmented Dickey Fuller Test. A time series model has a unit root if its first difference is stationary. For a linear time series model, this is equivalent to a $(1-B)$

factor in the AR polynomial $\phi(B)$. The Augmented Dickey Fuller Test will look for evidence of a unit root. We use the test to verify the DAX Opening prices are nonstationary, and the returns for the DAX stock index are stationary. Performing an ADF test on DAX Opens gives a p value of 0.603955, so, we fail to reject the null, so we may assume it is nonstationary. On the other hand, the DAX Return has a p value of 0 after running the test. We reject the null and there is evidence to show that the data is stationary. (See the Supplementary Details for the full results of the ADF test.)

1.0.5 Spectral Density Function

An estimator of the spectral density function determine a signal's frequency context content from a time series sample. It is measured as $\lambda(\omega) = \int_{\infty}^{-\infty} \gamma(x) e^{-2\pi i \omega x}$ (Ionides 2026b). In this section, we look at the frequencies for the DAX Stock Prices and Returns. It splits data into overlapping segments, computes modified periodograms, and averages them to reduce the variance. From the graphs, we may visually inspect the point at which the spectral density is the highest. This will be the dominant frequency, the most common rate at which an oscillation in the stock index happens.

Below, we show frequency vs spectral density function of the DAX Open and Return values.



The highest spectral densities in our graphs above are at frequencies of 0.00036 and 0.3, which indicates that these are the dominant frequencies for DAX open and return values, respectively. The periods are 2777 and 3.3 for the dominant oscillation. These make sense, as, we have roughly 2777 data, so, there is one unique non stationary pattern for open prices. Returns are stationary, and repeat roughly every 3.3 stock measurements, which is roughly 3.3 days.

2 Fitting Time Series Models

Let's train models on time series data to accurately reflect our observations on the DAX Stock Index from the exploratory data analysis, respecting stationarity.

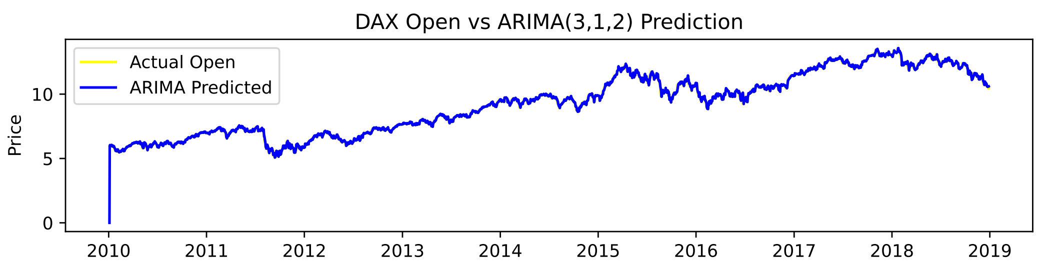
2.0.1 ARMA and ARIMA

How do we forecast DAX Stock Prices and Returns? The exploratory data analysis reveals that these values depend on lags and differencing. This information is captured in an ARMA model (Shumway and Stoffer 2000).

$$
\begin{aligned}
y_n &= \phi_1 y_{n-1} + \phi_2 y_{n-2} + \phi_3 y_{n-3} + \theta_1 \epsilon_n + \theta_2 \epsilon_{n-1} \\
y_n &= \phi_1 B y_n + \phi_2 B^2 y_n + \phi_3 B^3 y_n + \theta_1 \epsilon_n + \theta_2 B \epsilon_n \\
y_n \phi(B) &= \theta(B) \epsilon_n
\end{aligned}
$$

2.0.1.1 DAX Open Prices

Let's analyze the results of an ARIMA model on the DAX Open Prices; our visual analysis and ADF test revealed that the DAX Open Prices as a time series are nonstationary. Because of the COVID wave in 2020, for purposes of fitting a model, we use all results from Januray 1, 2010 to December 31, 2018. The best ARIMA model for DAX Open Prices is an ARIMA(3,1,2). The shape matches, and has a relatively low AIC of -3916.7 - it is trained and graphed on top of raw data below, overlapping almost perfectly. Before measuring residuals, we compare models by $AIC = 2k - 2ln(\hat{L})$, which punishes complexity and lack of goodness of fit. A lower AIC score is better.



The AIC values for the ARIMA models of different number of parameters (ex, ARIMA(1,0,1), ARIMA(2,0,2), etc.) are in the tables below.

Table 1: AIC for ARIMA(p,d,q) Model Comparison, as calculated from arma_aic_run.py

	MA Order				
AR Order	0	1	2	3	4
0	10629.7	7529.3	5005.4	3147.4	1892.3
1	-3897.5	-3895.6	-3893.6	-3893.5	-3892.2
2	-3895.6	-3894.0	-3892.1	-3891.9	-3891.6
3	-3893.6	-3892.4	-3891.1	-3898.6	-3896.7
4	-3893.6	-3891.3	-3889.2	-3896.6	-3895.1

(a) AIC for $d = 0$

	MA Order				
AR Order	0	1	2	3	4
0	-3905.2	-3903.3	-3901.3	-3901.2	-3899.9
1	-3903.3	-3901.3	-3899.3	-3899.3	-3901.8
2	-3901.3	-3899.8	-3916.6	-3906	-3903.5
3	-3901.3	-3899.3	-3916.7	-3904.1	-3903.7
4	-3899.8	-3901.5	-3903.9	-3902.4	-3915.5

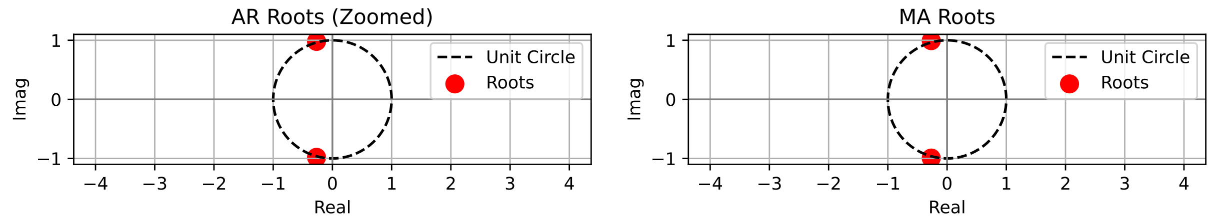
(b) AIC for $d = 1$

	MA Order				
AR Order	0	1	2	3	4
0	-2267.9	-3893.6	-3891.6	-3889.7	-3889.5
1	-2939.6	-3891.6	-3889.6	-3887.7	-3885.7
2	-3253.7	-3889.7	-3887.6	-3887.4	-3904.9
3	-3378.4	-3889.6	-3885.7	-3885.2	-3888.2
4	-3432	-3888.1	-3885.7	-3883.3	-3898.7

(c) AIC for $d = 2$

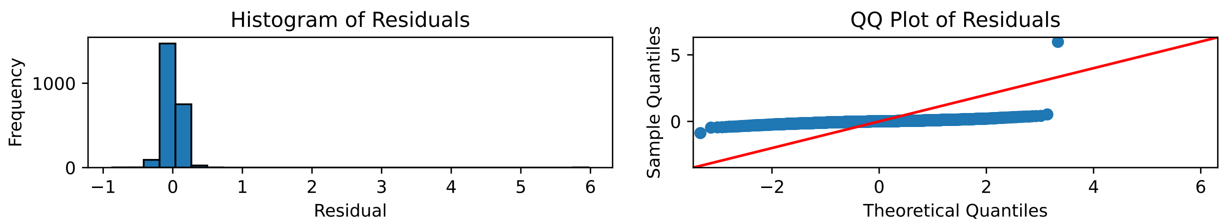
2.0.1.1.1 Model Diagnosis

We check causality and invertibility in this section. Causality is when the AR process has all roots that lie outside the unit circle (Shumway and Stoffer 2000). Invertability is when the MA process has all roots that lie outside the unit circle. We diagnose our best solution, ARIMA (3,1,2) below - it is both causal and invertible.



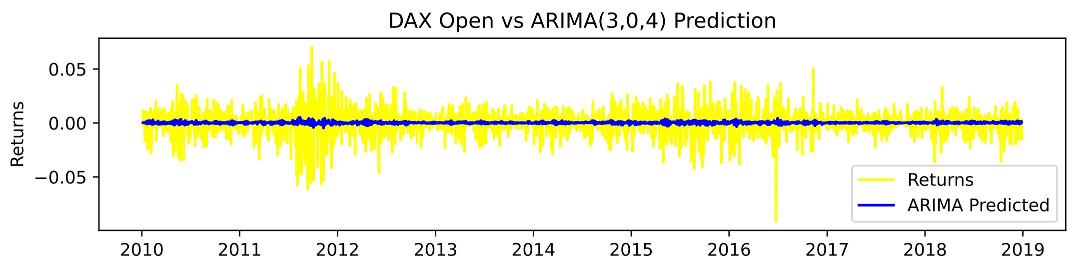
2.0.1.1.2 Residual Analysis

We look at the residuals. Both the qq plot and the histogram indicate that the residuals are normally distributed $\epsilon \approx N(0, \sigma^2)$. This indicates that the ARIMA(3,1,2) is a good fit.



2.0.1.2 DAX Returns

Let's run the same procedure for the returns. The best model is ARIMA(3, 0, 4). The empirical and visual data shows that the returns are stationary, so, having do differencing component ($d = 0$) makes sense for this data.



Now, we look at AIC for every combination of parameters in ARIMA(p,d,q) for DAX returns.

Table 2: AIC for ARIMA(p,d,q) Model Comparison, as calculated from arma_aic_run.py

AR Order	MA Order 0	1	2	3	4
0	-14028.1	-14026.4	-14024.4	-14022.6	-14024.4
1	-14026.4	-14024.4	-14022.1	-14019.3	-14025.4
2	-14024.4	-14022.1	-14020.1	-14020.4	-14025.2
3	-14022.7	-14020.6	-14018.1	-14019.7	-14029.4
4	-14024	-14022.1	-14020.1	-14017.4	-14026

(a) AIC for $d = 0$

AR Order	MA Order 0	1	2	3	4
0	-12372.7	-14012.7	-14011.1	-14009.1	-14007.2
1	-13058.6	-14011.1	-14009.6	-14005.9	-14005.6
2	-13350.5	-14008.8	-14007.2	-14007.9	-14000.3
3	-13464.5	-14006.6	-14004.7	-14006.4	-14004.5
4	-13530.4	-14007.8	-13995.3	-13982	-13995.4

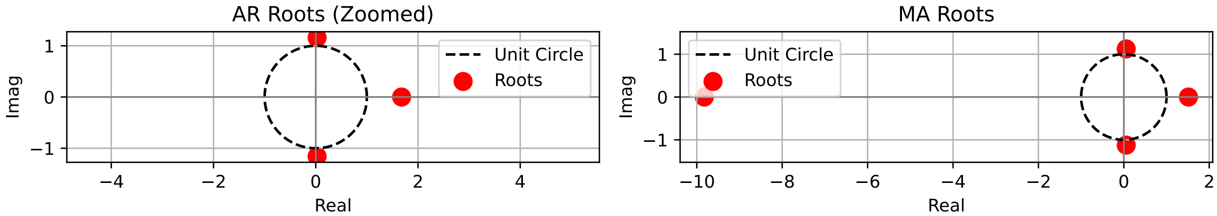
(b) AIC for $d = 1$

AR Order	MA Order 0	1	2	3	4
0	-9787.3	-12356.1	-1398	-13976.2	-13974.3
1	-11163.8	-13040.6	-13981.9	-13979.5	-13978.2
2	-11898.1	-13327.1	-13115.6	-13972.6	-13931.5
3	-12303.4	-13441.1	-13333.5	-13173.1	-13058.8
4	-12519.4	-13393.4	-13370.9	-13340.9	-13269.3

(c) AIC for $d = 2$

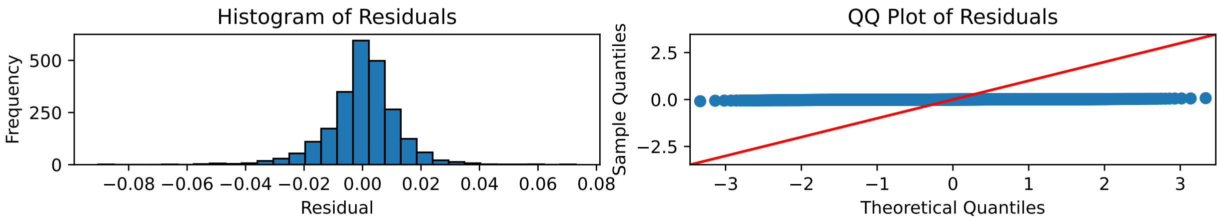
2.0.1.2.1 Model Diagnosis

We assess causality and invertability. All solutions lie outside the unit circle. Therefore, our model is causal and invertible.



2.0.1.2.2 Residual Analysis

We look at the residuals. Both the qq plot and the histogram indicate that the residuals are normally distributed $\epsilon \approx N(0, \sigma^2)$. This indicates that the ARIMA(3,0,4) is a good fit.



2.0.2 SARMA and SARIMA

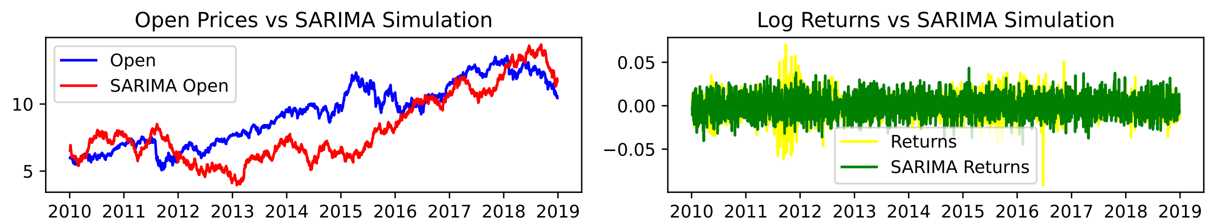
The Seasonal Autoregressive Moving Average Model (SARMA) adds a seasonal component to the ARMA and ARIMA Models. Instead of considering every lag from 1 to 35, for example, we can take lags 1, 12, and 13, repeatedly (Shumway and Stoffer 2000). This allows for monthly and yearly coefficients in our model, as one would expect with stock. We show a representation of SARIMAX(2,0,2)(1,0,0,12), and how well it performs below.

$$By_n = y_{n-1}; B^2 y_n = y_{n-2}; B^{12} y_n = y_{n-12}$$

$$y_t - \phi_1 y_{t-1} - \phi_2 y_{t-2} - \Phi_1 y_{t-12} + \Phi_1 \phi_1 y_{t-13} + \Phi_1 \phi_2 y_{t-14} = \epsilon_t + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2}$$

$$(1 - \Phi_1 B^{12})(1 - \phi_1 B - \phi_2 B^2)\, y_t = (1 + \theta_1 B + \theta_2 B^2)\, \epsilon_t$$

$$\Phi(B^{12})\,\phi(B)\,y_t = \theta(B)\,\Theta(B^{12})\,\epsilon_t$$



The SARIMA models for DAX Open prices and returns have AIC values of -3897.6 and -14029.4.

2.0.3 GARCH

The model selection above did not sufficiently rule out ARMA(0, 0) for the mean of the returns data. AIC favored ARMA(0, 0) as the model with the second best AIC, with the exception of ARMA(3, 4) discussed above which was found to have roots very close to the unit circle and thus may not be a reliable solution. Checking the acf of the squared returns data, however, as in Figure 6 shows consistent significant autocorrelation, suggesting a pattern in the variance of the data, known as volatility clustering, in which variability can, e.g. remain high then swing low and remain low. A GARCH model can often serve as a good model (Wikipedia contributors 2026). The GARCH(1, 1) is shown below. We adopt the following notation from (authors 2024):

$$r_t = \mu_t + \varepsilon_t; \qquad \varepsilon_t = \sigma_t e_t; \qquad \sigma_t^2 = \omega + \alpha\varepsilon_{t-1}^2 + \beta\sigma_{t-1}^2$$

where r_t are the returns, μ_t is the mean function, ε_t is the residual at time t, σ_t^2 is the volatility at time t, and e_t is some white noise process with variance 1 (say, $e_t \sim N(0,1)$ for simplicity). Thus, GARCH models essentially fit an ARMA model to σ_t^2, thus allowing estimates of σ_t^2. However, as we cannot observe volatility directly, we need a point of comparison for our GARCH estimate of σ_t^2. We use r_t^2 here, as that seems to be favored in the literature and among practitioners for daily data (Cipra 2020), (QuantInsti 2025). In the supplmentary material, we show that it is conditionally unbiased for σ_t^2. Smoother estimates, which are the variance of log returns over the past several days (here, we also use the past 5 days, as was suggested in (QuantInsti 2025)) are biased but have less variability.

2.0.3.1 Model Fitting

We found that GARCH(1, 1) minimized AIC across all four error distributions we attempted. We found that the Skew-t distribution had the best QQ plot of standardized residuals after model fitting, shown in Figure 2, indicating that the error terms e_t most resembled the skew-t distribution for our dataset, and Figure 5 showed that the standardized residuals did not show significant autocorrelation. Figure 3 compares the GARCH estimate of volatility to the smoothed estimate. Thus, we see that the model was generally able to adapt to the volatility, following the highs and lows, though its estimates were a more tamed version of the realized proxy. However, using the unbiased realized volatility in Figure 4 shows how large errors can dominate the dataset and skew model performance.

To better assess the performance of this model, we used a rolling window forecast as suggested in (Brownlee 2019), (QuantInsti 2025), in which we used the previous 2000 days to forecast the volatility of the next day, and repeat for each day in 2019. Comparison with the smoothed estimate is shown in Figure 1. As said above, the model responds to volatility changes but less so to the magnitude of changes. Metrics from the 2010-2018 in-sample performance and for the 2019 rolling window forecast are shown below (estimated volatility and proxies are for variance, not SD). The in-sample performance was much worse, likely as the numerous outliers in the volatility series accumulated over years and blew up the MAE and MSE.

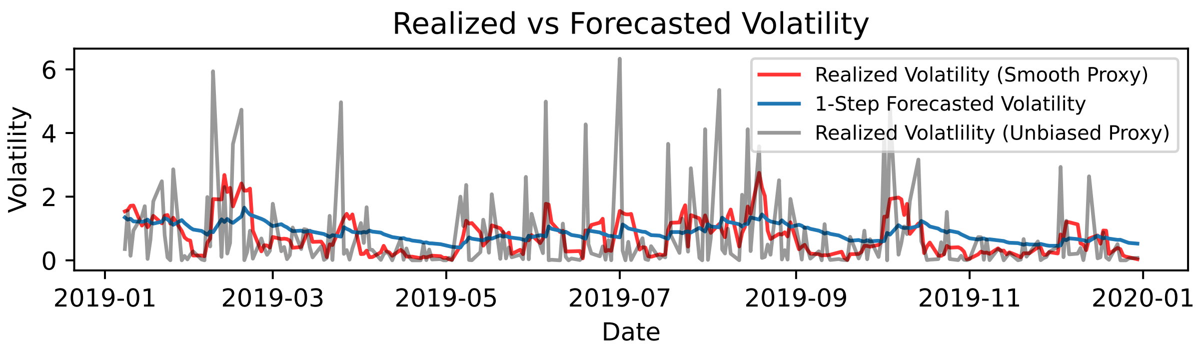


Figure 1: Comparison of Volatility to Estimated Volatility

Metric (In-Sample)	Unbiased Proxy	Smoothed Proxy	Metric (Forecast)	Unbiased Proxy	Smoothed Proxy
Mean Absolute Error	1.629	0.784	Mean Absolute Error	0.874	0.432
Mean Squared Error	12.420	2.266	Mean Squared Error	1.522	0.253

3 Connection to Previous Projects

As with several other previous STATS 531 projects such as (STATS 531 2025a), (STATS 531 2025b), and (STATS 531 2024), we wanted to study financial time series data while including volatility forecasting using models such as GARCH. We also wanted to build upon (STATS 531 2025b) by forecasting volatility separately, not forecasting log returns themselves with ARMA-GARCH as in (STATS 531 2025a) or (STATS 531 2024). We wanted to overcome the heavy tail misspecification in the residual plots of (STATS 531 2025b) by using appropriate error distributions in our volatility forecasting, which we seem to have been more successful in, as their heavy tail issue was never quite resolved whereas our skew-t distribution provided a more appropriate standardized residual plot.

4 Conclusion

In conclusion, we show that modeling changes in the DAX Stock Open and Returns can be achieved through ARMA models with low AIC and mean zero residuals - these metrics indicate low complexity, goodness of fit, and low errors. We therefore solve the problem of forecasting DAX Stock prices. We expand upon it by giving insights into causality and invertibility, and we demonstrate modeling volatility with GARCH.

5 Supplementary Material

5.1 ADF Test Details

The ADF test results were as follows:

$$\begin{aligned}
&H_0 : \text{The process } \{x_t\} \text{ is nonstationary} \\
&\quad \exists\, t, h, k \text{ such that } (x_t, x_{t+1}, \dots, x_{t+k}) \overset{d}{\neq} (x_{t+h}, x_{t+1+h}, \dots, x_{t+k+h}) \\
&H_A : \text{The process } \{x_t\} \text{ is stationary} \\
&\quad \forall\, t, h, k,\ (x_t, x_{t+1}, \dots, x_{t+k}) \overset{d}{=} (x_{t+h}, x_{t+1+h}, \dots, x_{t+k+h})
\end{aligned}$$

	0	1
Series	DAX Open	DAX Returns
ADF Statistic	-1.35435	-26.565984
p-value	0.603955	0.0
Lags Used	4	3
Observations	2785	2785
Critical Value (5%)	-2.862578	-2.862578
Reject the H0?	No	Yes

5.2 Proof that $\sigma_t^2 = Var(r_t - \mu_t \mid \mathcal{F}_{t-1})$

Assume $\mu_t = 0$ for each t. This is reasonable becuase $\bar{x} \approx 0.02$ for our data and seems to have a constant zero trend line throughout. If we denote $\mathcal{F}_{t-1}$ as the σ-algebra (intuitively, information) generated by $\varepsilon_t, \sigma_{t-1}^2$ then we have

$$\begin{aligned}
Var(r_t - \mu_t \mid \mathcal{F}_{t-1}) = Var(\varepsilon_t \mid \mathcal{F}_{t-1}) &= Var(\sigma_t e_t \mid \mathcal{F}_{t-1}) \\
&= \sigma_t^2 Var(e_t \mid \mathcal{F}_{t-1}) \qquad (\sigma_t \text{ is } \mathcal{F}_{t-1} \text{ measurable}) \\
&= \sigma_t^2 Var(e_t) = \sigma_t^2 \qquad (e_t \text{ is independent of } \mathcal{F}_{t-1})
\end{aligned}$$

5.3 Proof that squared returns are conditionally unbiased

We have

$$E(r_t^2 \mid \mathcal{F}_{t-1}) = Var(r_t \mid \mathcal{F}_{t-1}) = \sigma_t^2,$$

and hence

$$E(r_t^2) = E(E(r_t^2 \mid \mathcal{F}_{t-1})) = E(\sigma_t^2),$$

5.4 Supplementary GARCH plots

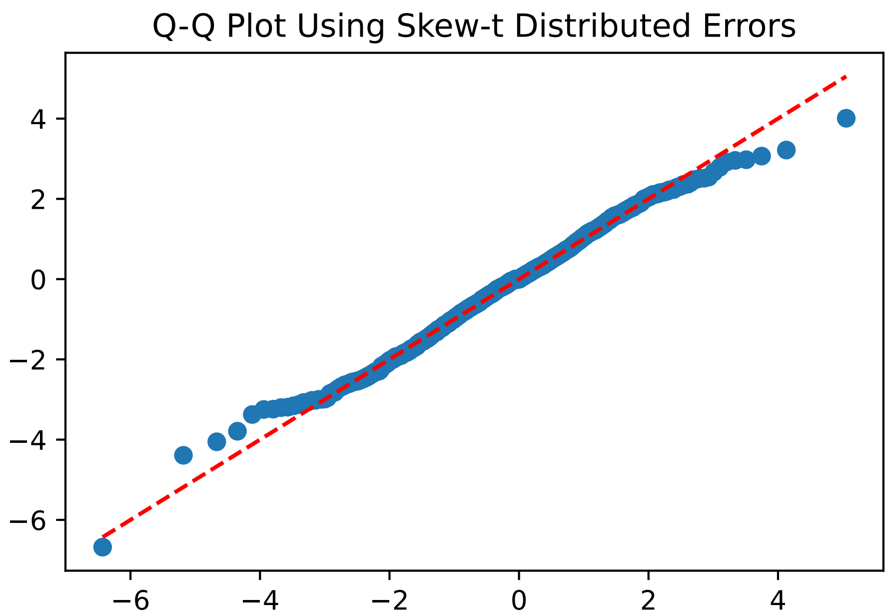


Figure 2: QQ Plot of Standardized Residuals

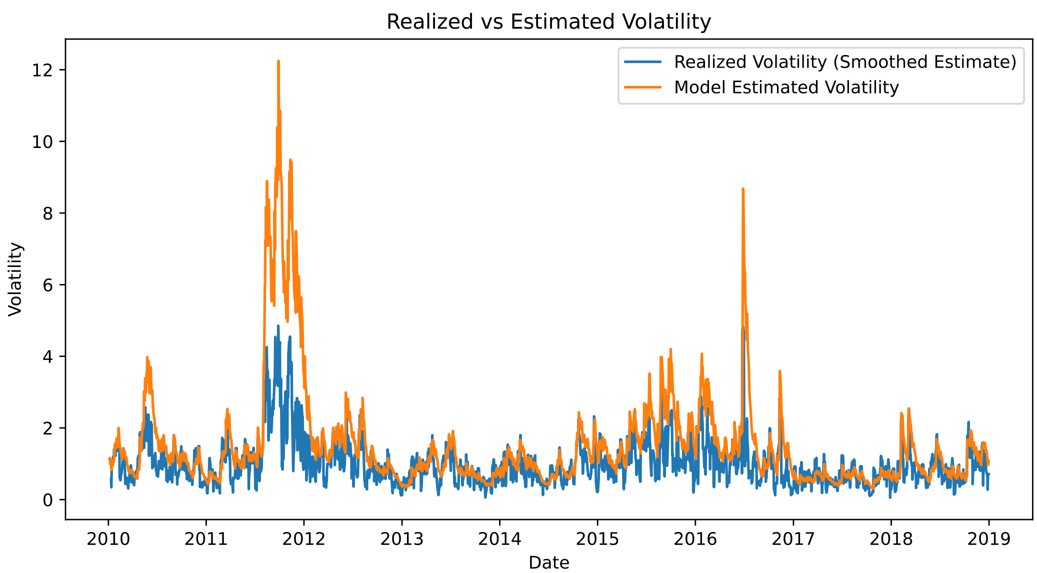


Figure 3: Comparison of Smoothed Volatility to Estimated Volatility

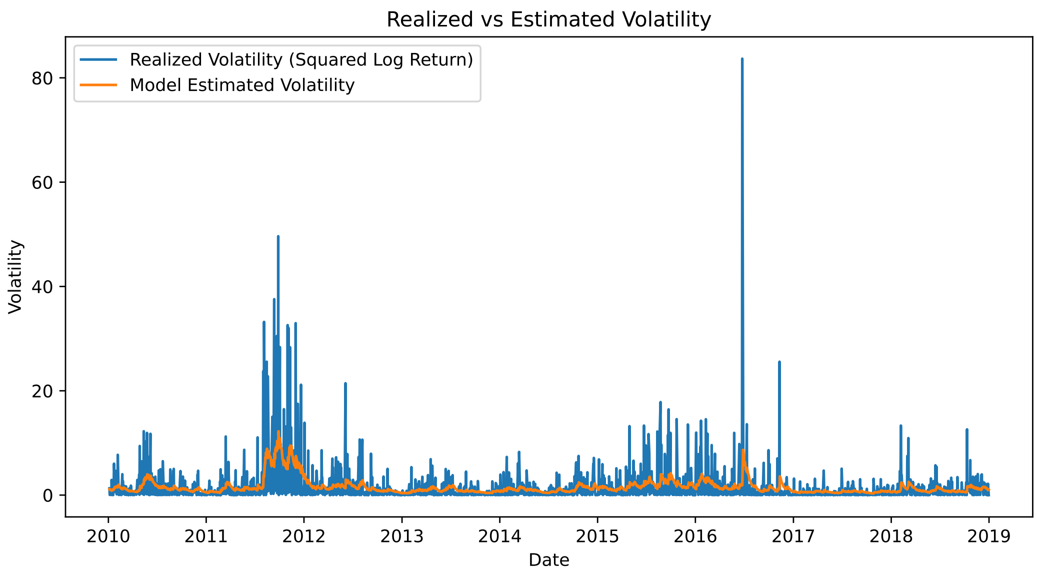


Figure 4: Comparison of Unbiased Volatility Proxy to Estimated Volatility

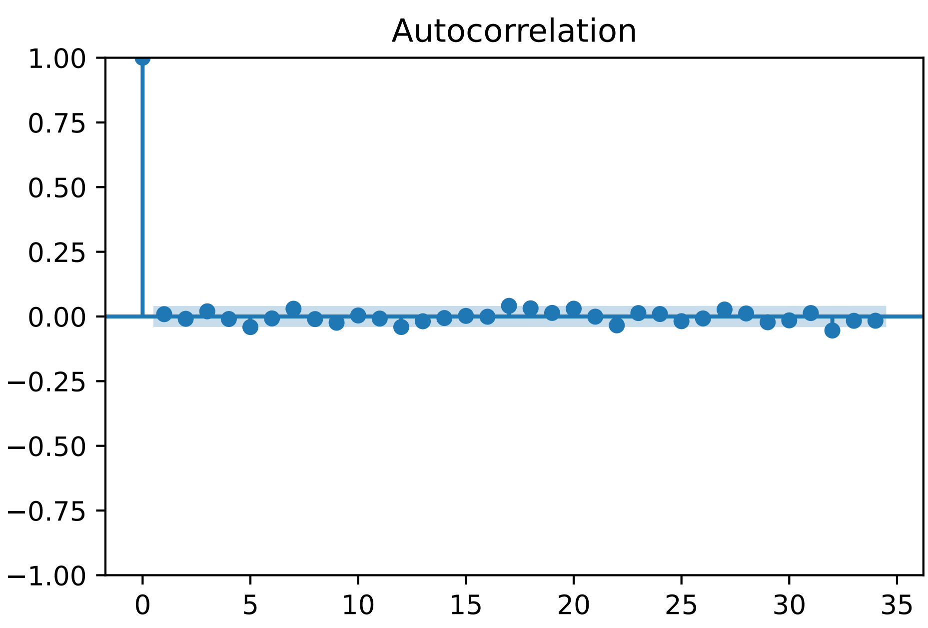


Figure 5: ACF Plot for Standardized Residuals from the GARCH model

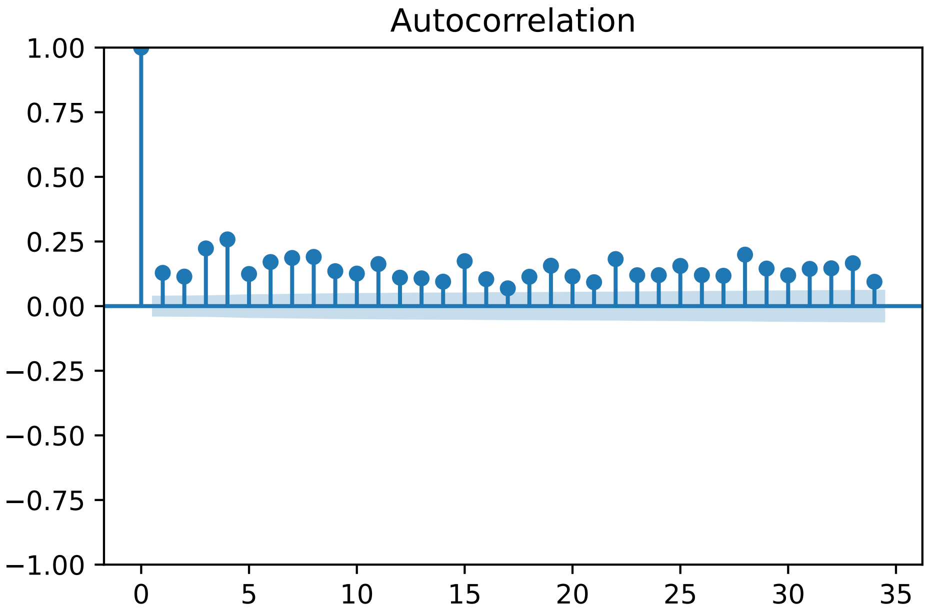


Figure 6: ACF Plot for Standardized Residuals from the GARCH model

References

authors, ARCH package. 2024. "Introduction to ARCH Models." https://arch.readthedocs.io/en/latest/univariate/introduction.html This documentation provides a general overview of the GARCH model and shows how to use the arch_model function needed to implement the code in this report.

Brownlee, Jason. 2019. "How to Backtest Machine Learning Models for Time Series Forecasting." https://machinelearningmastery.com/backtest-machine-learning-models-time-series-forecasting/.

Caporale, Guglielmo Maria, and Gil-Alana. 2022. "The Impact of the COVID-19 Pandemic on Persistence in the European Stock Markets." *Heliyon* 8 (11): e11560. https://doi.org/10.1016/j.heliyon.2022.e11560.

Cipra, Tomáš. 2020. "Volatility of Financial Time Series." In *Time Series in Economics and Finance*. Cham: Springer. https://doi.org/10.1007/978-3-030-46347-2_8.

Ionides, Edward L. 2026a. "Lecture 05: Time Series Modeling." https://ionides.github.io/531w26/05/slides.pdf.

———. 2026b. *Modeling and Analysis of Time Series Data*. Department of Statistics, University of Michigan. https://ionides.github.io/531w26/01/slides.pdf.

OpenAI. 2026. "ChatGPT (GPT-5.2)." https://chat.openai.com/ ChatGPT helped in formatting plots tables, and code output, and general debugging help for quarto. ChatGPT helped adapt code from QuantInsti to create a rolling window volatility forecast. ChatGPT helped in creating a QQ plot for non-normal distributions.

QuantInsti. 2025. “GARCH Vs. GJR-GARCH Models in Python for Volatility Forecasting.” https://blog.quantinsti.com/garch-gjr-garch-volatility-forecasting-python/ This gave a tutorial of how to use GARCH models in python, including how to do a rolling window forecast.

Shumway, Robert H., and David S. Stoffer. 2000. *Time Series Analysis and Its Applications*. Springer.

STATS 531. 2024. “Midterm Project 14 (Blinded).” https://ionides.github.io/531w24/midterm_project/project14/blinded.html.

———. 2025a. “Midterm Project 07 (Blinded).” https://ionides.github.io/531w25/midterm_project/project07/blinded.html.

———. 2025b. “Midterm Project 12 (Blinded).” https://ionides.github.io/531w25/midterm_project/project12/blinded.html.

———. 2025c. “Midterm Project 13 (Blinded).” https://ionides.github.io/531w25/midterm_project/project13/blinded.html.

STOXX Ltd. 2026. “DAX® Index.” https://www.stoxx.com/index/dax/.

Wikipedia contributors. 2026. “Autoregressive Conditional Heteroskedasticity — Wikipedia, the Free Encyclopedia.” https://en.wikipedia.org/w/index.php?title=Autoregressive_conditional_heteroskedasticity&oldid=1331240335.