Volatility Modeling Project Summary

Frederick Vu

This document summarizes our efforts on forecasting the volatility of six different stocks, labeled Stock a through Stock f, up to one month out from the end of the given data set. We opted to use univariate GARCH models on the cleaned, daily closing prices for each stock to model the daily log returns and to forecast volatility up to one month out.

The following table presents the annualized predicted volatilities and their 95% prediction intervals for each stock:

Stock	Predicted Volatility (%)	Lower Bound (%)	Upper Bound (%)
Stock a	192.2	121.6	272.6
Stock b	82.3	38.5	151.3
Stock c	40.7	32.7	51.2
Stock d	30.2	16.1	51.1
Stock e	36.3	24.3	47.2
Stock f	38.8	20.0	61.4

Table 1: Annualized Predicted Volatilities and Prediction Intervals for Stocks

1 Data Cleaning

We started analyzing the data simply by graphing the price history of each stock and by inspecting the summary of the imported dataframe. We found that Stocks a and d had numerous incorrectly input price values of 0 and 1, respectively, and that Stock c most likely had a stock split around day 150. We additionally found that data was missing for Stocks c and f, with the latter being observed to be rather illiquid, with large chunks of missing data.

Closer inspection of the data set confirmed that most significant errant values for Stocks a and d could be removed by setting instances of the values 0 or 1 to NaN and forward filling to remove the NaN's. The missing values for Stocks c and f were also forward filled. Moreover, we identified the three-day weekend between Day 145 to Day 149 of the data set to be the point of the two-to-one stock split and readjusted the price values before the split by scaling down by a factor of 2.

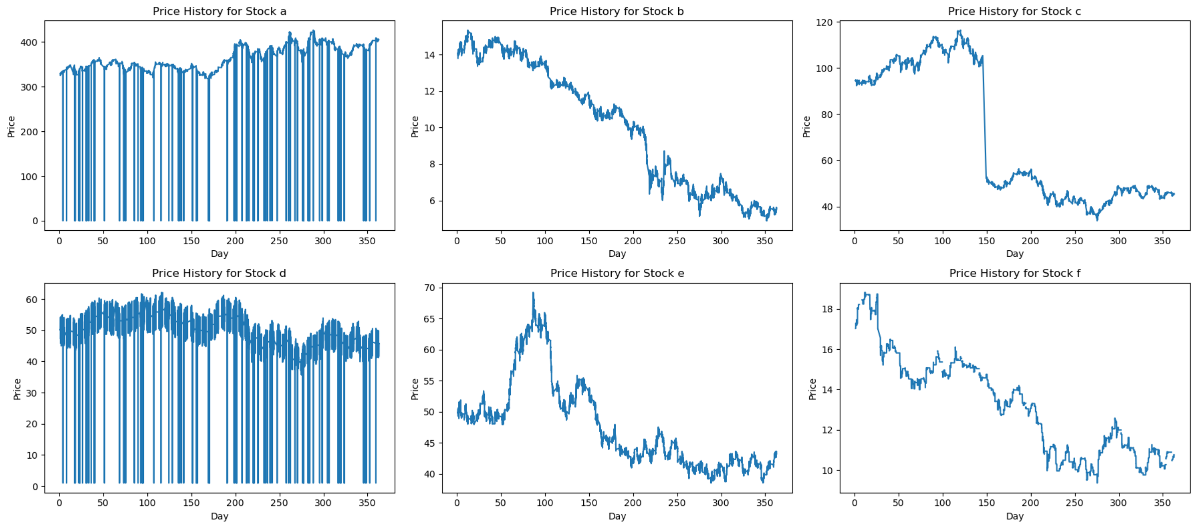

Figure 1: Raw price history of stocks

Upon closer inspection of the prices for particular days, we found Stock d also had many outlier price values that were most probably input errors, and that Stock b typically experiences signiciant high-frequency trading. We deemed that the outlier values of Stock d were errant as many of the outlier prices demonstrated greater than 5% shifts from the previous minute's value and were immediately corrected in the next minute back to a price level more in line with the prior minute's price. If these were stocks trading on the NYSE, then circuit breakers should have been activated, but price fluctuations were continued to be observed after these outlier values, indicating that the data set contains contaminated data for Stock d.

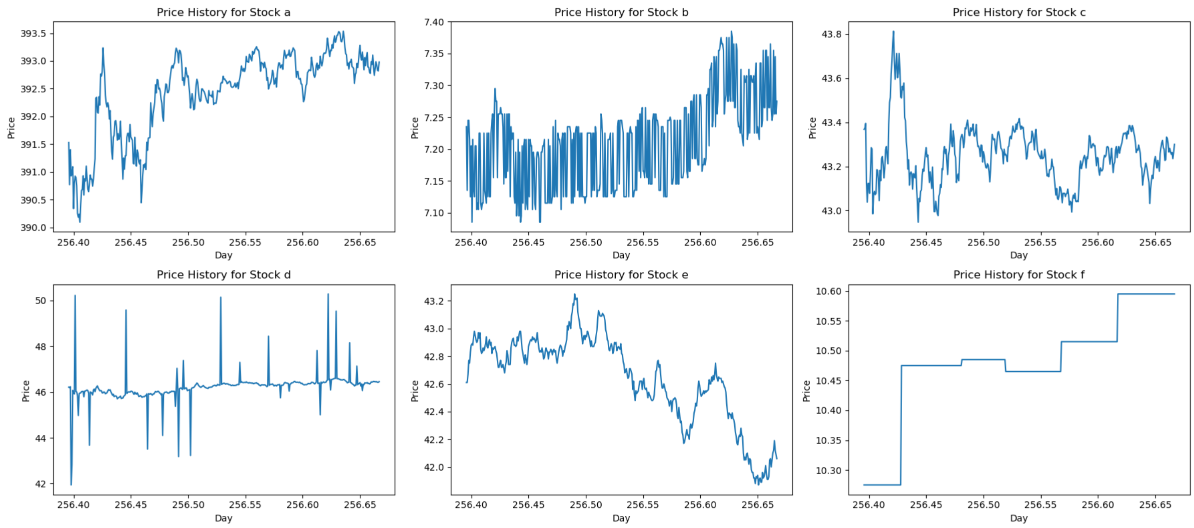

Figure 2: Price history of stocks on Day 256

We identified the errant values for Stock d using windowed interquartile range filters for each trading day (both forward and backward windows to handle the cases at the start and end of the trading day) and replaced them with forward filled values that were not filtered. We completed our data cleaning by replacing the prices of Stock b with the rolling average of the prices of the past 10 minutes for each trading day. Our reason for doing so was to remove the effects of the bid-ask bounce on the long-term volatility forecasts. The minute-by-minute clean data is plotted in 5

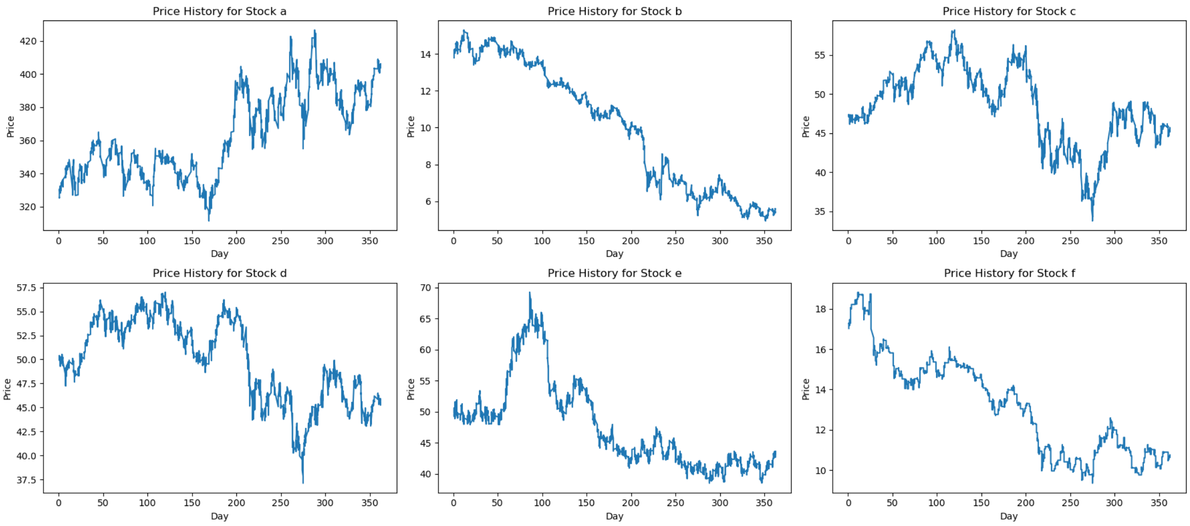

Figure 3: Cleaned price history of stocks

2 Fitting GARCH models

We were generally unfamiliar with standard (academic) practices in modeling volatility at the start of this project, and so we read [H][Ch. 21] to arrive at the decision to use GARCH models and their extensions to model volatility. These models take into account the observed phenomenon of volatility clustering, which is the tendency for (low or high) volatility to be 'sticky', and certain extensions of the standard GARCH model, such as EGARCH, or exponential GARCH, also take into account the asymmetric effects of the sign of price shocks, i.e., whether the price change is positive or negative, on the volatility.

We originally wanted to use ARMA(p,q)-EGARCH(r,o,s) models, which are models of the form

$$r_t = \mu + \sum_{i=1}^{p} \phi_i r_{t-i} + \sum_{j=1}^{q} \theta_j u_{t-j} + u_t$$

where $\{u_t\}_t$ is a serially uncorrelated, but generally dependent, white noise process which we factor as $u_t = \sigma_t \cdot e_t$ and where, under the assumption that $\{e_t\}_t$ is IID standard normal, the EGARCH(r,o,s) component of the model has the following specification

$$\ln \sigma_t^2 = \omega + \sum_{i=1}^{p} \alpha_i \left(|e_{t-i}| - \sqrt{2/\pi} \right) + \sum_{j=1}^{o} \gamma_j e_{t-j} + \sum_{k=1}^{q} \beta_k \ln \sigma_{t-k}^2,$$

where ω is another IID standard normal white noise process, and the coefficients $\phi_i, \theta_j, \alpha_i, \gamma_j$, and β_k are the model parameters. Here, the terms in the left sum give the deviation of $|e_t|$ from $\mathbb{E}(e_t) = \sqrt{2/\pi}$, and the middle sum accounts for the asymmetric effects of price shocks mentioned in the previous paragraph.

Our hopes were dashed upon realization that the standard `arch` package in Python did not have an implementation of ARMA for the conditional mean component of our log return series r_t. Additionally, we ran into numerical issues when we allowed $o > 0$. Thus, we settled for using AR(p)-EGARCH(r,0,s) models for all but Stock c, which continued to face numerical issues. For Stock c, we followed an AR(p)-GARCH(r,s) model. For the GARCH model, the specification of the conditional variance term is as follows

$$\sigma_t^2 = \omega + \sum_{i=1}^{r} \alpha_i v_{t-i}^2 + \sum_{k=1}^{s} \beta_k \sigma_{t-k}^2.$$

Of course, before we fitted the models, we verified that each of the daily log return series was stationary using the augmented Dickey-Fuller test, and we fitted preliminary AR(p) models and used the Lagrange mutiplier tests to confirm for Stocks a through d that ARCH effects were present. We provide graphs of the log returns and of rolling 21-day window realized volatility for each stock below to provide visual evidence for the stationarity and ARCH effects of the log return series.

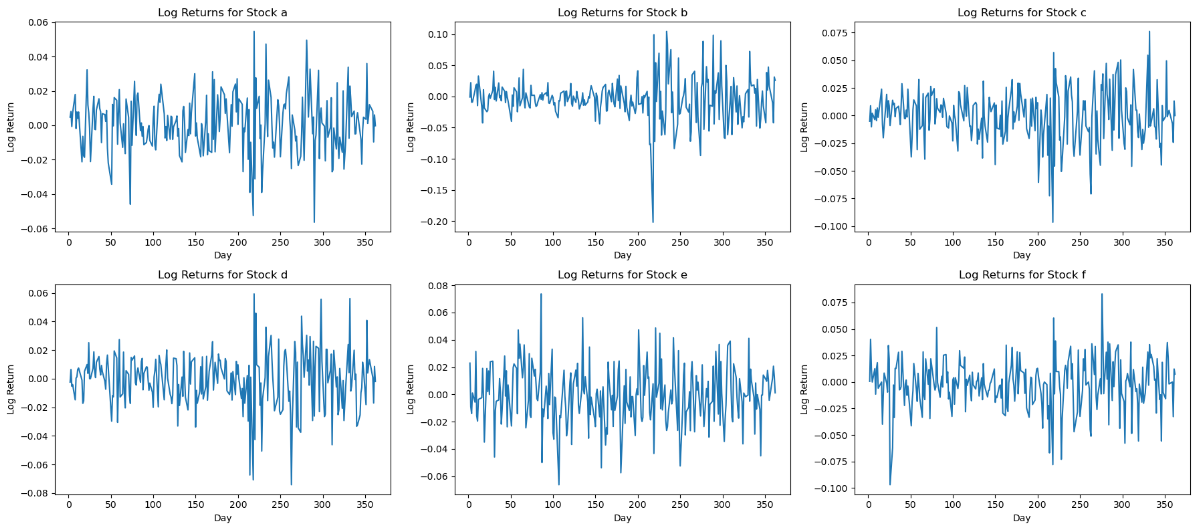

Figure 4: Log returns of stocks

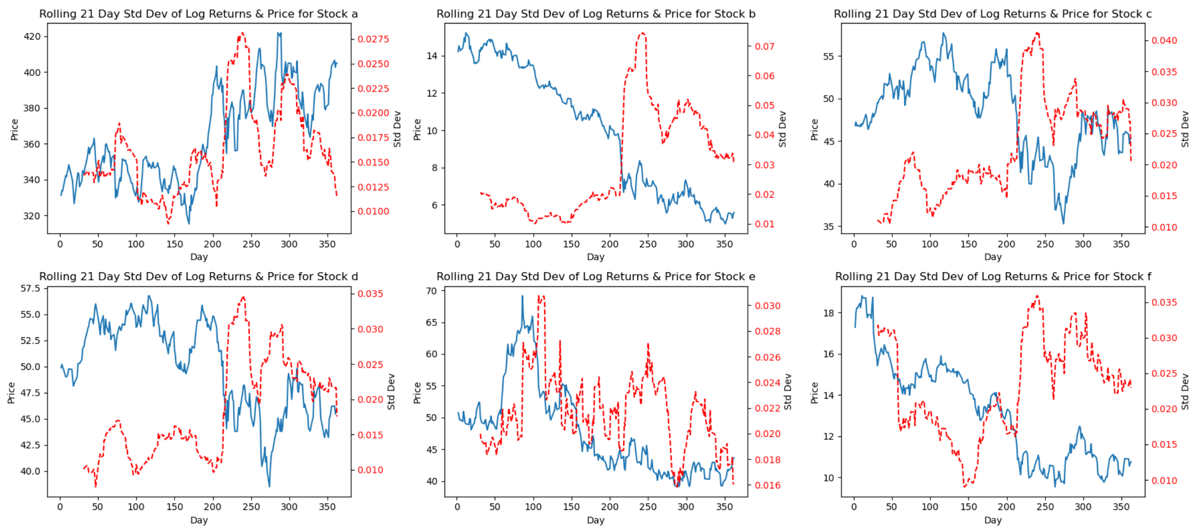

Figure 5: Rolling 21-day realized volatility of stocks

To determine the hyperparameters p, r, and s, we performed a simple grid search and compared models using the Akaike Information Criterion (AIC). For each of the stocks, we let the p parameter range over values $[0, 6]$, the r parameter over $[1, 4]$, and the s parameter over $[0, 3]$. We note that Stocks a, c, d, and f were found to be best fit with $p = 0$, while the best models for Stock b and e had $p = 2$ and $p = 1$, respectively. The precise parameter estimates are given in the accompanying `.ipynb` file.

Finally, to obtain the prediction intervals, we simply simulated each fitted model 1,000 times with a horizon of 21 days (the average number of trading days in a month) and determined the 2.5% and 97.5% percentile cutoffs.

References

[H] Hamilton, J. “Time Series Analysis”, Princeton University Press, 1994.