**Goal:** Use time-series analysis to forecast daily closing price of various financial instruments (i.e. stocks, bonds, ETFs) up to 14 days ahead.

**Methods:** We use regression to capture the trend and seasonality of the time-series. We use ARMA on the regression residuals to capture autocorrelation. We use GARCH on the regression residuals to capture heteroskedasticity (non-constant variance). Regression and ARMA provide the point estimates, and GARCH provides standard deviation estimates, which are helpful for confidence intervals.

**Data:**

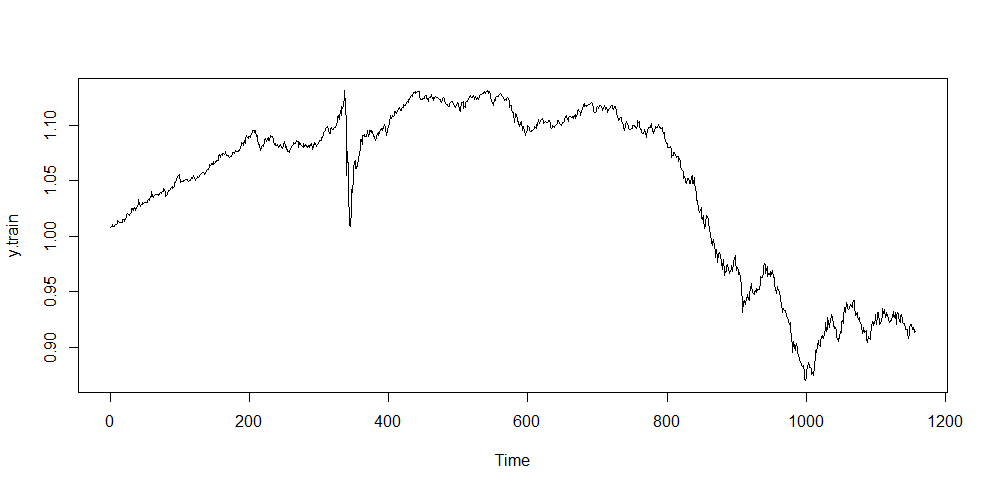
Daily close prices 2018-11-02 to 2023-11-01 of

PIMCO Active Bond Exchange-Traded Fund  
  
Vanguard Intermediate-Term Treasury Index Fd ETF (VGIT)

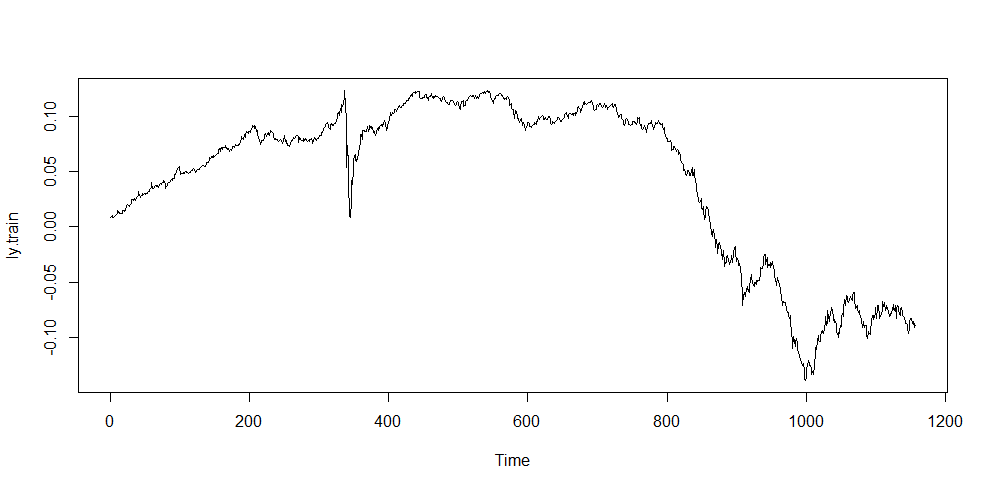
Tesla

**Fitting Process Example:**

Training time series. We see trend, non-constant variance, and potentially seasonality.



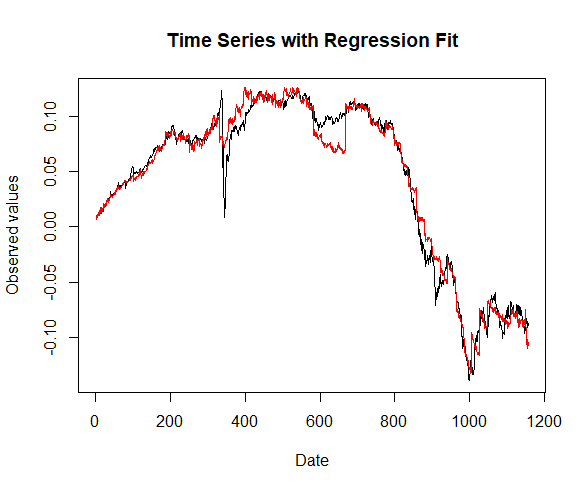
We do a log transformation to stabilize the variance a bit.



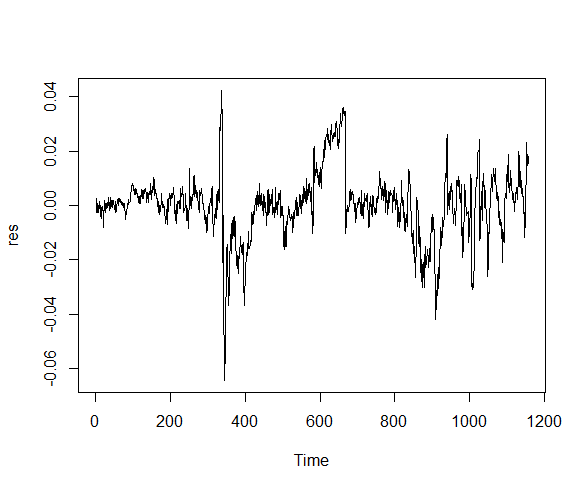
Let’s say we try a regression model. The predictor variables are time, dummy variables that indicate the month, and dummy variables that indicate the day of the month. There is no intercept since I have 12 dummy variables for month and 31 dummy variables for day of the month.

reg.model<- lm(ly ~poly(times, degree=3)\*(.-times)+0, data = X.train)

This is how the regression alone fits the training data



Here is the time-series plot of the regression residuals

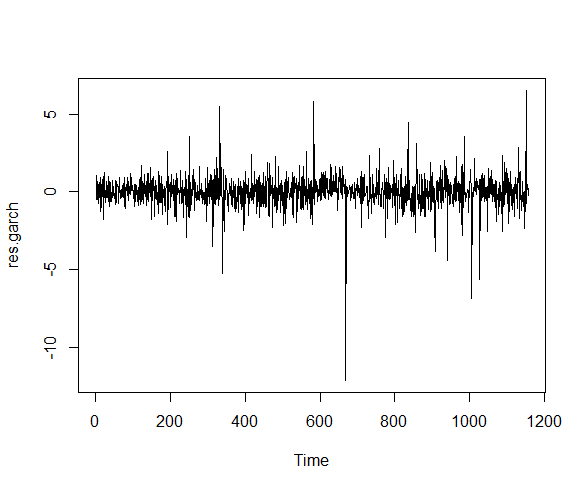


The trend seems mostly addressed, as it seems mostly constant around zero. However, there is clear autocorrelation (values staying above/below the mean for long periods of time) and non-constant variance.

We run the function auto.arima() on the residuals, which finds the best arima parameters based on various metrics like AIC and log likelihood. Let’s use these suggested parameters, which are (1,0), in this case.

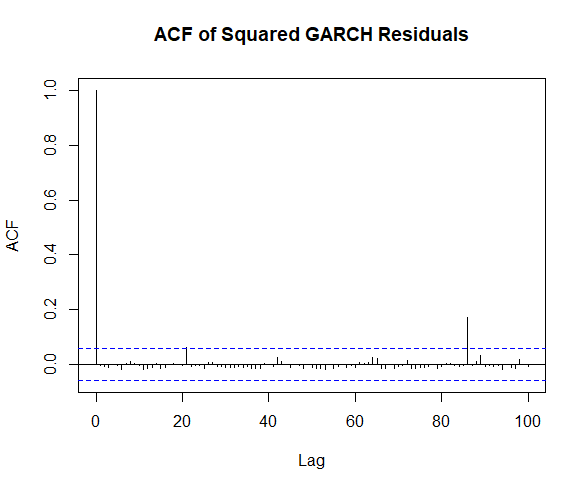
There is no auto.garch() function, so let’s just use parameters (1,1) for now.

Let’s analyze the residuals of the newly fitted ARMA-GARCH model.



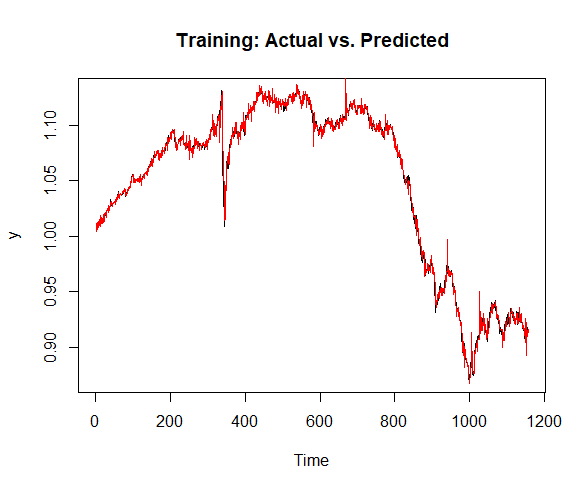
The trend and autocorrelation seem much more addressed. However, the variance still doesn’t seem to be fully non-constant. Still, everything is better than what it was; it looks like the white noise we should expect from residuals.

Here is the ACF plot of the squared residuals, which is used to detect autocorrelation.

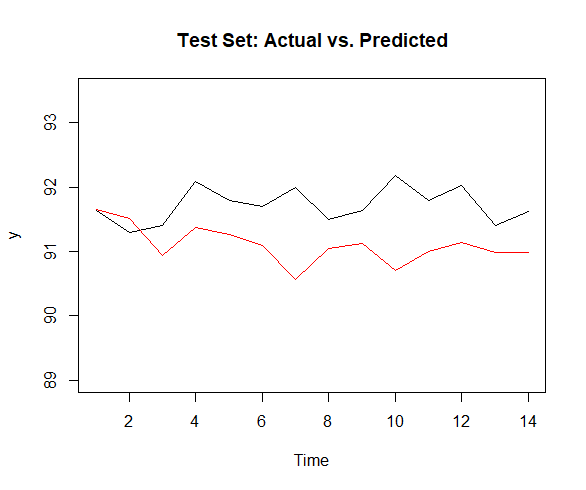


Interestingly, we see little spiked in lags that are multiples of 20. There is even a significant lag around 85-90. Weird.

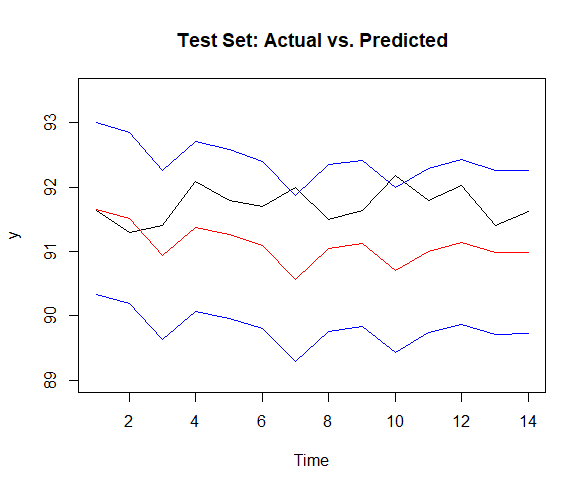
Here is the performance on the training set.



Here is the test set data with the point predictions.



And below is the 95% confidence interval with the standard deviation predicted using GARCH.



**Metrics/Evaluation:**

Before diving into metrics, I want to mention that some analyses from above bleed into evaluation. Specifically, the residual time series and ACF plots are a great way to visualize if we are missing out on autocorrelation and heteroskedasticity. There are also quantitative methods for doing this (Box test and white test, respectively). These will definitely play a role in our evaluation; however, a failure to capture these things should also be evidenced by the below metrics.

We need two sets of metrics: one for the point prediction from regression + ARMA and one for the prediction intervals.

Let’s talk about point predictions first.

For in-sample, Akaike Information Criteria (AIC) seems to be the gold standard. AIC is analogous to adjusted R2 for regression since it penalizes complexity. Unlike adjusted R2, it is only useful when comparing two models. The number in a vacuum doesn’t really mean anything.

For both in-sample(?) and out-of-sample, we will use MAE and RMSE. There are a lot of options for error functions, but these seem most common. Once again, comparative analysis is most important here.

Finally, for both in-sample (?) and out-of sample, we will use mean directional accuracy. This is a proportion of the times the model correctly predicted the direction of the time series. This has a more objective interpretation than the other metrics i.e. the closer to 100% the better.

Now, for the standard deviation predictions from GARCH.

This is tough and subject for discussion.

1. Coverage Probability:
   1. This is the proportion of actual values that fall within the prediction intervals.
2. Interval Score:
   1. Combines the width of the prediction interval with a penalty for failing to cover the actual observation. A lower interval score indicates a better model, as it implies narrower intervals with fewer failures to cover the actual value.

**Evaluation Example:**

I have two models for the bond price, as summarized below.

|  |  |  |  |
| --- | --- | --- | --- |
| **Model** | **Regression** | **ARMA** | **GARCH** |
| 1 | poly(t,3)\*dummy variables | (1,0) | (1,1) |
| 2 | poly(t,5)\*dummy variables | (1,1) | (1,1) |

The evaluation would be of the form.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Model** | **AIC (in-sample)** | **MAE** | **RMSE** | **Mean Directional Accuracy** | **Coverage Probability** | **Interval Score** |
| 1 |  |  |  |  |  |  |
| 2 |  |  |  |  |  |  |

**Challenges/Questions:**

**1) Scope – how many different time series? Different models?**

**2) Any major issues with the fitting process? Does GARCH make sense?**

**2) Do 14 day forecasts make sense?**