Project 3 Github:   <https://github.com/mintzj/ST599-3>

We’ve got 2 projects to choose from this time, any opinions?

Looks like we will do project #2, since we are more interested in this topic.

~~1:  Microarray and Brain Images:~~

*The possible questions to ask are -*

*1. What are the important gene probes for the whole brain simultaneously?*

*2. What are the important gene probes for a specific part of brain?*

*3. What are the differences in important gene probes between two people?*

*4. Compare the gene probes and MRI between two people.*

**2:  Historical stock data:**

***1. Predict future stock prices from current data and check with how correct are the predictions.***

***2. Find the relation between stock prices of one product with stock prices of other products of same or past time points.***

***3. Find important predictors both in terms of time points and stock prices of other products.***

***4. Find transformations of the data for linear model assumptions to hold or perform different forms of regression.***

Links to project files: <https://quantquote.com/historical-stock-data/>

Company Abbreviations: <https://quantquote.com/docs/symbol_map_comnam.csv>

Jeff:  What does he mean by gene probes?  It’s not too bad to read the files into R, interpreting it is another matter.

Stock data seems to be interesting. May be we can make a little money out of this project.

Thanks to QE, American stock market is booming these years, I would be interesting to run a projected based on this.

Jie and Jeff will work first on visualizing and capturing the large-scale trends of the market.

Xun develop improved prediction for a small collection of core stocks.

(To make money, we really only need to predict one stock.)

Xun also liked the idea of identifying groups of stocks which may perform similarly.  Some lists of these groups exist already.  We could look for relationships between individual stocks within a group, between groups of stocks and individual stocks, or even group-group relationships.

I am working on a file to read stocks into memory and name them by their ticker (filename).  Will fix it tomorrow, version is available as read\_stocks.r

Jie:I try to plot the overall trend with combined datasets, but it’s not working.

I just find the trend plot online and that is.

We don’t need to analysis the overall dataset, its huge, We can select one company and analysis it

I will write the report at in the midnight. so put all your thoughts asap.



Hey everyone,

I've uploaded some basic aggregation functions to our git. (Sorry for all the delay; I've been dealing with some other problems simultaneously.) The resulting files are each about 13-20 Mb; too large to upload here, but the code doesn't take too long to run anyway. I'm going to continue to work towards our goal with these new input methods. Specifically, I intend to run some time series analysis on our data and see what pops out.

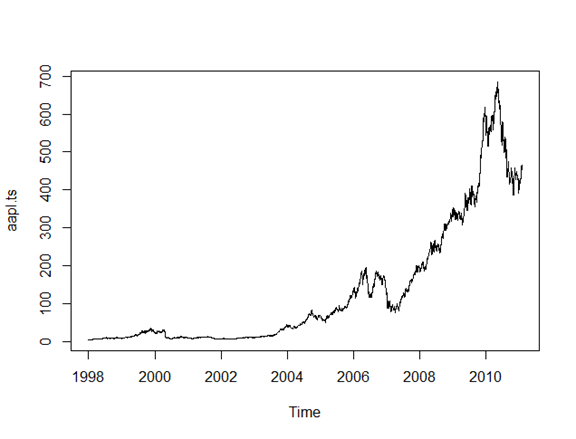
How is everyone else doing? I'm unable to see any progress on the project right now; if you have things to upload, please upload them to the git repository frequently. That way, we're all on the same page!

Hello, here is my output.            ---Yijun

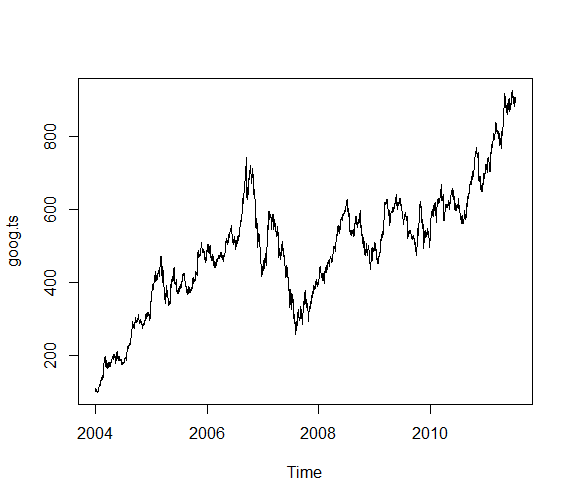
I'm analyzing the trend and predicting the price for single stock by using time series. Here I gave the results for the Apple (aapl) and the Google (goog), and I only used the closing price for the stock for the sequence of the time.

First, I transformed the data to the time series, and plot the trend.

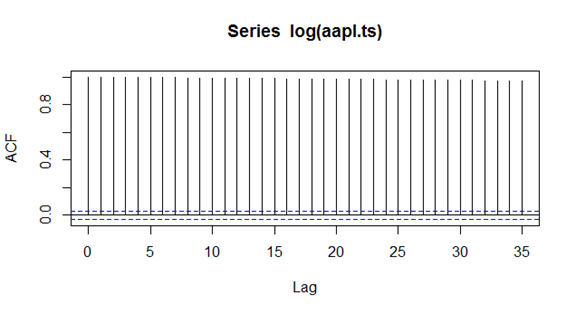
This is the time trend for Apple stock closing price from 1998 to 2013.



Here is the time trend for Google stock closing price from 2004 to 2013.



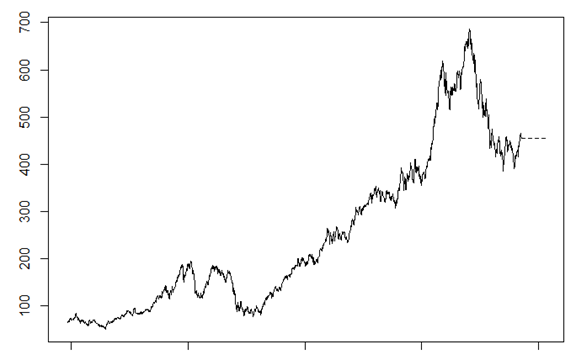
This is the ACF for Apple stock closing price. It suggests there is strong autocorrelation in the time series. Also, use the log transformation to adjust the scale.



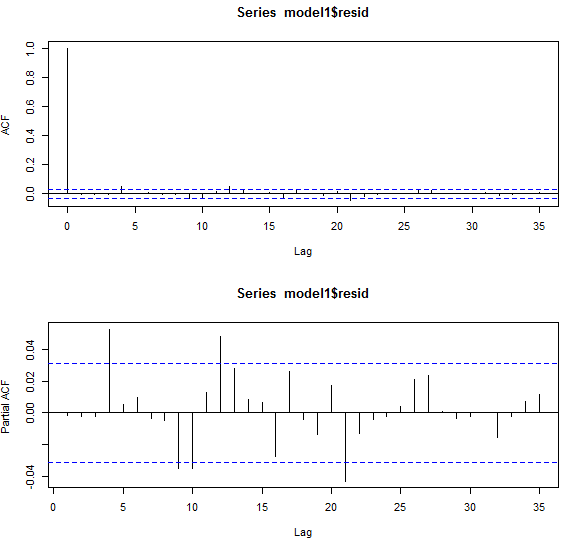
Using the ARIMA model for the time series in order to adjust for the autocorrelation, and by compare the best AIC, I chose the best model to fit the data, and predicted the future stock price, which was plotted as dashed lines on the following graph.

The predicted and the true values of the Apple stocks are compared in the table. The prediction was a bit conservative, that is lower than the true prices.

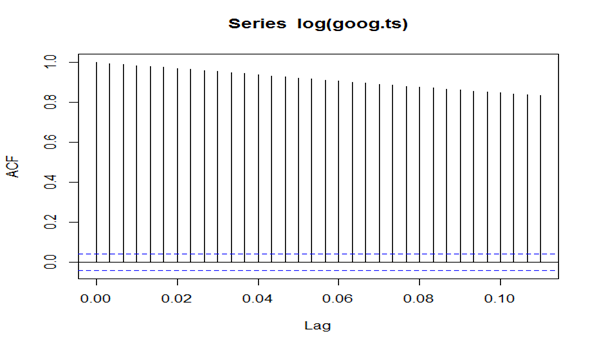
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 |
| Predicted | 454.5740 | 454.5192 | 454.5225 | 454.5220 | 454.5220 |
| True | 454.45 | 467.36 | 489.57 | 498.50 | 497.91 |



Check the ACF and PACF of the model, it shows the autocorrelation was adjusted. That is good.



This is the ACF for Google stock closing price. It suggests there is strong autocorrelation in the time series. Also, use the log transformation to adjust the scale.

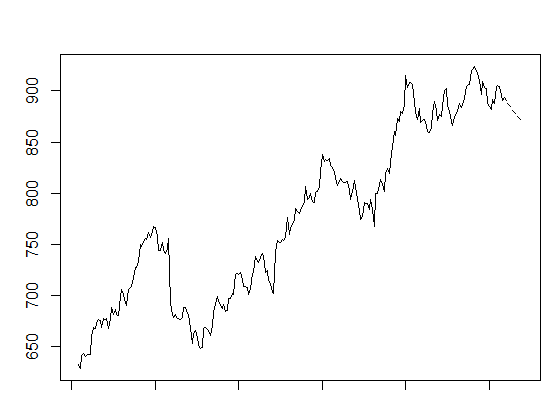


Using the ARIMA model for the time series in order to adjust for the autocorrelation, and by compare the best AIC, I chose the best model to fit the data, and predicted the future stock price, which was plotted as dashed lines on the following graph.

The predicted and the true values of the Google stocks are compared in the table. The prediction was a bit conservative, that is lower than the true prices.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 |
| Predicted | 888.0494 | 885.8044 | 883.5748 | 881.3605 | 879.1613 |
| True | 909.32 | 903.61 | 902.34 | 887.74 | 885.08 |

predicted the future stock price for Google (dashed line)



Yijun,

AWESOME job on all of this. It seems like you’ve done most of the time series work I could have done already! I might look into the possibility of GARCH-modeled errors, assuming that the current model allows for it, but other than that I think I might need to try and look at some of the other questions that have been asked. After all, your model looks to be doing its job very well (if I’m reading the model ACF right - it’s been a while!).

Terrific job again.

Here is the code I used, after reading in the file names, etc.

View(stocks$aapl)

aapl.ts <- ts(stocks$aapl$Close)

plot(aapl.ts)

acf(log(aapl.ts))

acf(aapl.ts)

plot(diff(log(aapl.ts)))

acf(diff(log(aapl.ts)))

get.best.arima <- function(aapl.ts, maxord = c(1,1,1,1,1,1)){best.aic <- 1e8

                                                           n <- length(aapl.ts)

                                                           for (p in 0:maxord[1]) for(d in 0:maxord[2]) for(q in 0:maxord[3])

                                                             for (P in 0:maxord[4]) for(D in 0:maxord[5]) for(Q in 0:maxord[6])

                                                             {

                                                               fit <- arima(aapl.ts, order = c(p,d,q),

                                                                            seas = list(order = c(P,D,Q),

                                                                                        frequency(aapl.ts)), method = "CSS")

                                                               fit.aic <- -2 \* fit$loglik + (log(n) + 1) \* length(fit$coef)

                                                               if (fit.aic < best.aic)

                                                               {best.aic <- fit.aic

                                                                best.fit <- fit

                                                                best.model <- c(p,d,q,P,D,Q)

                                                               }

                                                             }

                                                           list(best.aic, best.fit, best.model)

}

best.arima.im<- get.best.arima((log(aapl.ts)), maxord = c(2,2,2,2,2,2))

best.fit.im <- best.arima.im [[2]]

acf( resid(best.fit.im))

pacf( resid(best.fit.im))

best.arima.im

ts.plot( cbind( window(aapl.ts,start = 1985),exp(predict(best.fit.im,10)$pred) ), lty = 1:3)

exp(predict(best.fit.im,10)$pred)

model1= arima(log(aapl.ts), order = c(0,0,0),seas  =list(order = c(2,1,0),12))

AIC(model1)

# layout(c(1,1,2,2))

acf(model1$resid)

pacf(model1$resid)

plot(model1)

I’ve saved your work into an R file and uploaded it to Git!

Thank you!

Just to check: where did you find the true values for the AAPL stock? I’ve been looking around but can’t find anything on the same scale; where you listed values in the rough 400s, I can only find values around 100. (Yours seem to be accurate visually; I’m just clearly missing something!)

In the meantime, I have uploaded my work so far on regressing one stock on another. I intend to use multiple stocks tomorrow morning, and also to see how these model’s predictions turn out. (The standard error on the fits is very low; max 1.5 dollars away from the actual stock prices in generating the model!) For now, though, I am going to sleep.

Tomorrow we can discuss the report writeup.

I found the stock price on the yahoo finance, and the data we got ends Aug 2013, so the prediction price is at that time.

<http://finance.yahoo.com/q/hp?s=AAPL&a=11&b=12&c=1980&d=07&e=2&f=2013&g=d>

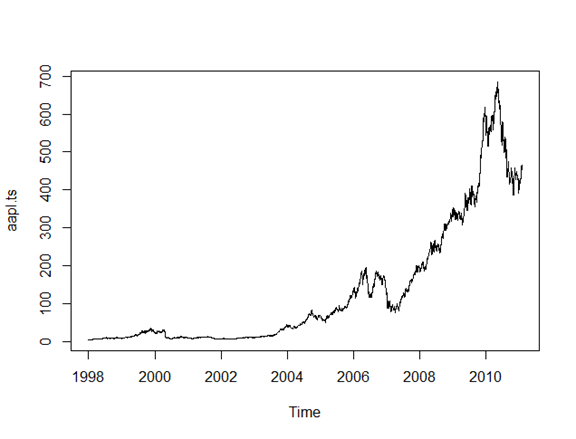
I’ve uploaded more work under arima\_forecast.R .  It’s not perfect - the residuals I’m getting doing cross-validation are fairly bad, enough that I wouldn’t want to use this as a means of prediction - but it’s at least giving some information.

I’m going to keep doing some regression work; however, in order to make sure Jie has enough to work with, I’m going to quickly write up what work I’ve done (partially edging in on Yijun’s work) to make the writeup easier.

**Project/Data Background**

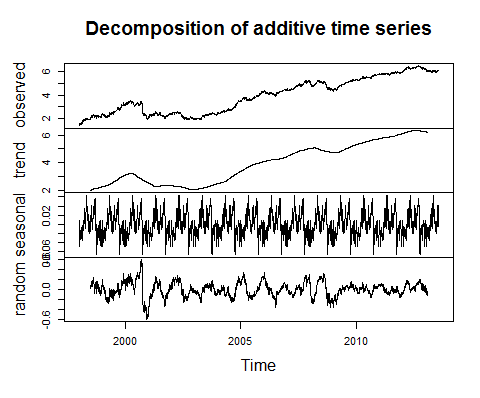
Stocks (recorded each weekday with open, close, low, and high price) were imported from the QuantQuote Historical Data repository for the 500 companies in the S&P 500 listing. A file, read\_stocks.R, was created to read in the stock data, and a further file summ\_tables.R was created to aggregate on each variable in our new list of stock data.

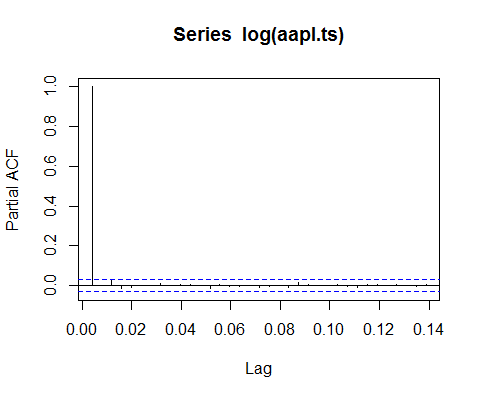
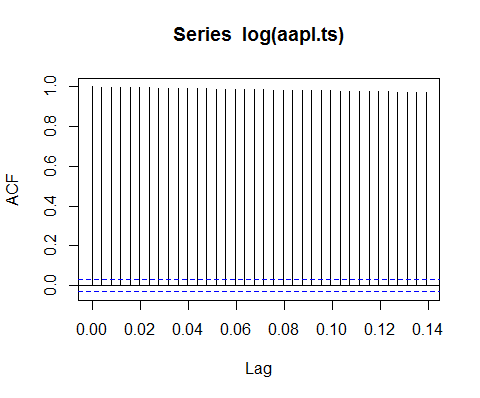
(MORE BACKGROUND DESCRIPTION HERE? DESCRIBE THE DATA SOME)

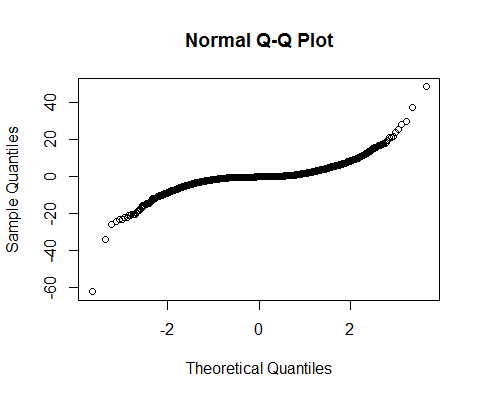


**Findings**

Thanks to the daily nature of the stock market, a time series approach was intuitive. We chose to narrow our focus to one stock at a time initially, beginning with AAPL - Apple Inc. - for a few useful reasons: AAPL contains no missing data (being present from the first observed day to the most recent), and is well-behaved pricewise while still varying from day to day.

Thanks to rapid scaling in stock price, a log transformation may be appropriate here. We transform the series and observe its decomposition (seen left). A very weak seasonal component can be observed, with somewhat-poorly behaved random component; however, these are still potentially appropriate for time series residuals. From here, we observe the auto-correlation function (ACF) and partial auto-correlation function (PACF):



These are immediately consistent with what we would expect of an AR(1) time series with *large*  coefficient phi (approaching 1). Our series does still fall within the realm of stationarity, however, and thus we may fit a model of AR(1) errors. While the ACF and PACF for this mdoel function as we would hope, the residuals are not entirely well-behaved; our normal quantile-quantile is fairly heavy-tailed, indicating that we are more likely to have heavy outliers than we’d like.

From here, we attempted prediction of future AAPL stock prices in two methods: prediction solely off of our ARIMA model, and Holt-Winters exponential smoothing. Of the two, ARIMA model prediction proved to be significantly more accurate:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 |
| Predicted | 888.0494 | 885.8044 | 883.5748 | 881.3605 | 879.1613 |
| True | 909.32 | 903.61 | 902.34 | 887.74 | 885.08 |

**Obstacles/Solutions**

The data was presented in such a way that input and analysis was significantly easier than in previous projects. Two of the notable issues we ran into fell under two categories: NA prevalence and the ts() time series format.

The prevalence of NAs was fairly sizeable in the data. The majority of the 500 stocks do not have a full 17 years of data; as such, everything before the first recorded date and everything after the last recorded date are imputed as NA. Additionally, a number of the stocks (between 5-10%) have some missing data at seemingly random dates in the middle. For simple ARIMA analysis, this does not provide much of an issue; however, when regressing one stock on another it can be a problem. In these cases, imputation of 0 for NA works appropriately, as in this way the covariate stock does not affect the response stock in case of NAs.

Additionally, working with the ts() format forced a few adjustments in our code thanks to its unique setup. Most notably, a frequency was necessary to determine for the stocks in order to determine seasonality. After some discussion, a frequency of 252 recorded prices per year was chosen; this was determined by looking at several stocks and figuring out the rough number of prices per year.

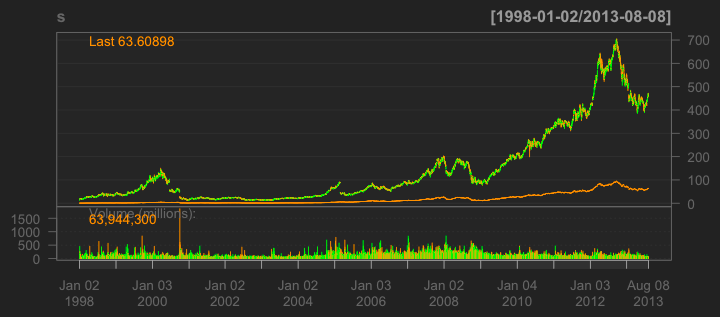
**Additional Questions**

We worked towards regressing AAPL’s closing price on the past closing prices of other stocks; however, we were unable to achieve a meaningful improvement using this method. (This is likely due to programming error, as we in fact were given significantly *less* accurate predictions by doing this.) In the future, finding some way to regress on a past number of stock closing prices - say, the past week - would give a lightweight and potentially useful tool for estimating stock price.

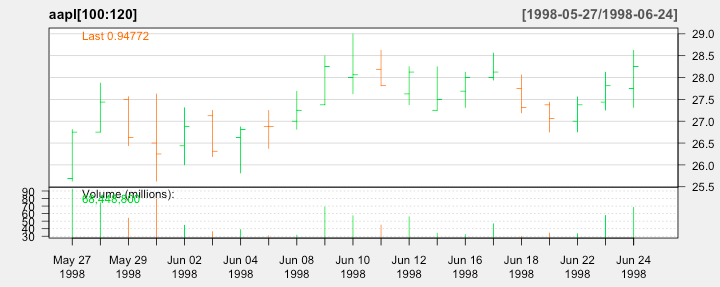
An additional, potentially interesting method of error analysis would be GARCH-modeled errors (General Autoregressive Conditionally Heteroskedastic). This method is often used in econometric analysis, and works in situations where sudden and abrupt shifts in value lead to larger variance temporarily (such as the stock market). However, given the relative accuracy of our prediction and the decent fit of a model as simple as AR(1), we put off this method of analysis for potential future work.

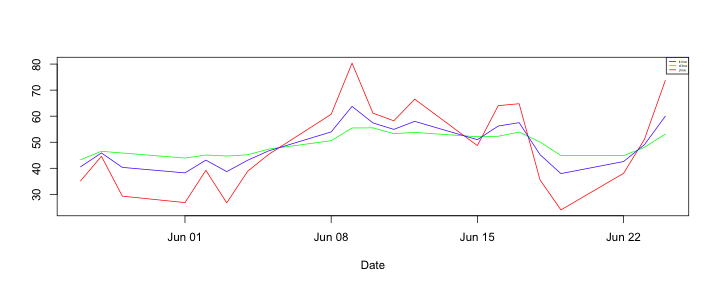
There - I think that’s a decent amount to start with on the work that’s been done. Yijun, if you have anything you’d like to add for the writeup, feel free to let us know here and we can get it in.

I drew a couple candle plots using R which will look professional, still working on decision function



I then introduced kdj lines as a predictor of stock price increase or decrease (Buy-in signal)





I am now working on the decision function, it seems not very accurate