STAT331 Final Report - Apple Stock Data

Kaisheng Shen, Rajat Sharma, Yongha Park

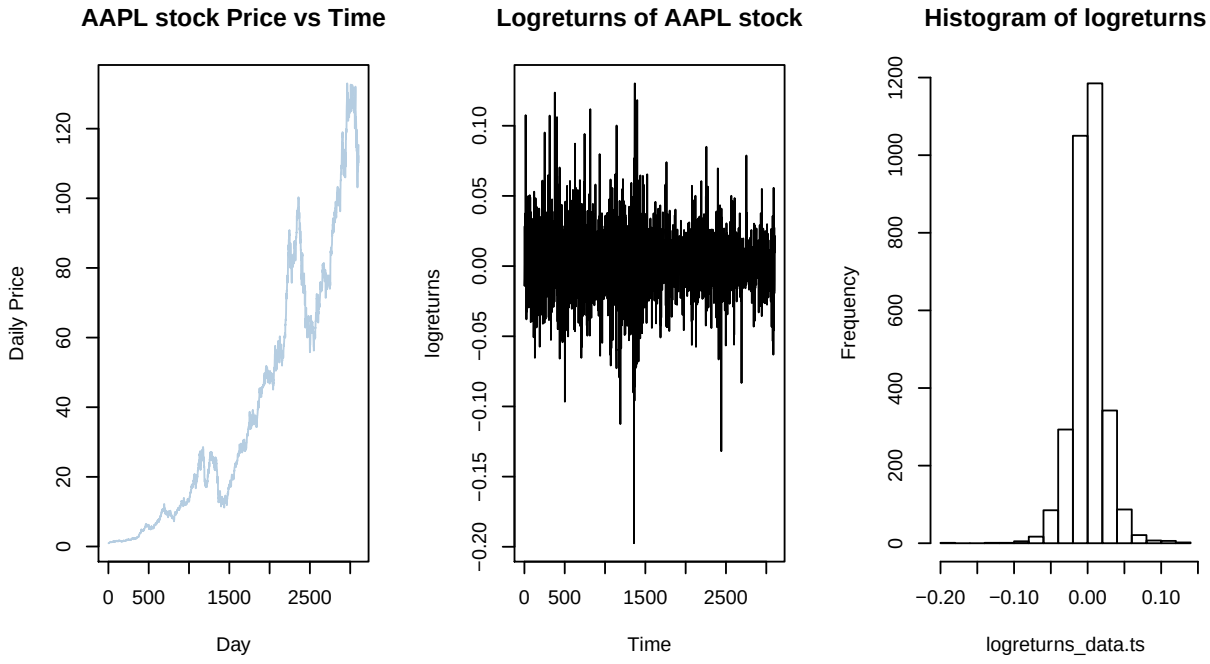
December 6, 2017

Summary

In finance, it is important to forecast volatility as this affects risk management strategies, asset allocations and bets placed on future volatility spikes. In this report, we are going to be looking at various extensions to the autoregressive model to measure volatility. In particular, we will evaluate models from the Apple Stock closing prices recorded between April 11, 2003 and September 11, 2015. In the report, we will present both autoregressive model and linear regression model and compare each model on different diagnostics. Through some analysis, it turns out that autoregressive model provides us better model fitting for predicting results in contrast to the linear model. Overall, the best model we came up with modelled the actual return rate of the Apple Stock quite closely with a small calculated error. Throughout the analysis of the data, it was concluded that it might be optimal to exclude events like the 2008 Stock market crash when coming up with a prediction model.

Taking a First look at the Data

First, let us take a look of our original data its $log(return)$ on AAPL. In finance it is popular method to take the logs of data to help protect from a Black Swan (an occurence that deviates largely from what is expected, ie 2008 financial crisis, dotcom bubble). These events are thought of to be random and extremely difficult to predict. Black Swans tend to devastate markets and create large losses for traders thus transforming using the *log* function provides a way to immunize our model from these random occurences.

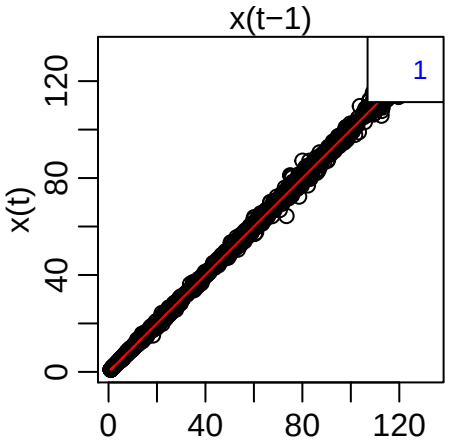


We can easily observe that APPL tends to have more days with postive return rates. The average of the histogram seems to be a positive value as well. We can also note that the return rates look normally

distributed due to the log transformation applied.

From the logreturns vs time plot, we can see that there is a large volatility spike around 1500th day; this is assumed to be the financial crisis of 2008. We can observe that after a small peroid of large low return rates, the market picks up with a period of high return rates. It is also important to note that the begining of the timeseries data (2003) had large volatility observations.

Now creating the $AAPL_i$ vs $AAPLi-1$ plot



From the plot, it is apparent that there is a linear trend between $APPL_i$ and $APPL_{i-1}$. This suggests that we can fit a linear for the relationship between the two variables.

Model Selection

First, we need to create our daily returns r_i. Next, since we are interested in modeling our *ret.AAPL_i*, we now convert our other variables to their daily returns as well.

Notice, that we need to estimate rhe i_th return using our previous information, we also need to shift our returns as known as lag. In this report we use lag 1 data in our analysis to simulate using yesterdays data.

Here is our new data set looks like:

```
##     logreturns        lag1    lag1eem      logeem      lag1spx
## 2 -0.014119836  0.02833584 0.01295085 0.011817055  0.019310221
## 3 -0.011250842 -0.01411984 0.01181706 0.011478830  0.006283663
## 4 -0.009135394 -0.01125084 0.01147883 0.007130619 -0.012311532
##        lag1vix       lag1spg       lag1bnd
## 2 -0.043057853  0.0015995525 -0.0001369332
## 3 -0.036984855  0.0143870250  0.0020975840
## 4 -0.001774623 -0.0007013185  0.0021386489
```

Now, lets try to fit our models.

For Model 1 refer to the appendix for the code and output.

We notice that for our first fitted model, it turns out that only *logeem* is Significant. We then choose the forward selection method for our selections of covariates. We try to fit more variables into our model, however we noticed that we lose significance for other variables. As we compared AIC, BIC, and R^2_{adj}, we obtain our first model (automated model).

```
## lm(formula = logreturns ~ lag1spx + logeem, data = returns_train)
```

Performing the Durbin-Watson test on our first model:

```
## [1] 0.3920539
```

We see that we get a p-value of 0.3921 which means we accept our null hypothesis of no autocorrelation. This helps solidfy the idea that our proposed model is a good fit.

Model 2:
We tried another way to fit model 2, our model was fitted by using new variables for lag 2 on $ret.AAPL$ which means we fitted our model as autoregressive model of $ret.AAPL_{i-2}$. For the output see the appendix.

```
## lm(formula = y[, 1] ~ y[, 3])
```

Performing the Durbin-Watson test on our second model:

```
## [1] 0.3169481
```

We can see that we obtain a p-value of 0.3169 indicating that we do not reject our null hypothesis of there being no autocorrelation. Thus again, this helps us solidfy that the proposed model may be apporpirate.

From the two models that we have selected above, we can see that model 1 has a better performance at prediciting the returns in contrast to model 2.

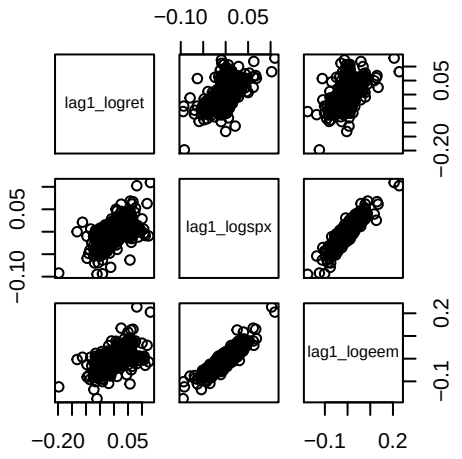
Thus we define our best fitting model as:

$$ret.AAPL_i = 0.0013199 + 0.0788710 lag1spx + 0.5550991 logeem$$

where $lag1spx = ret.SPX_{i-2}$, and $logeem = ret.EEM_{i-1}$

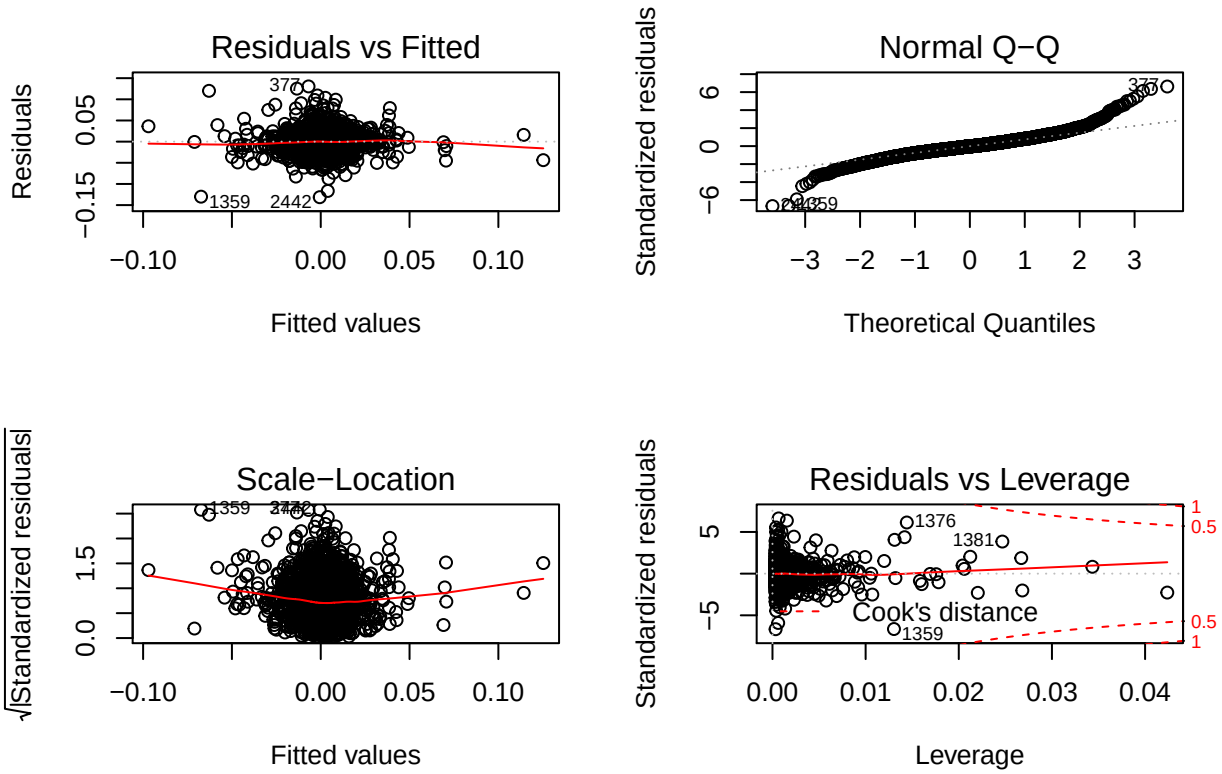
Model Diagnostics

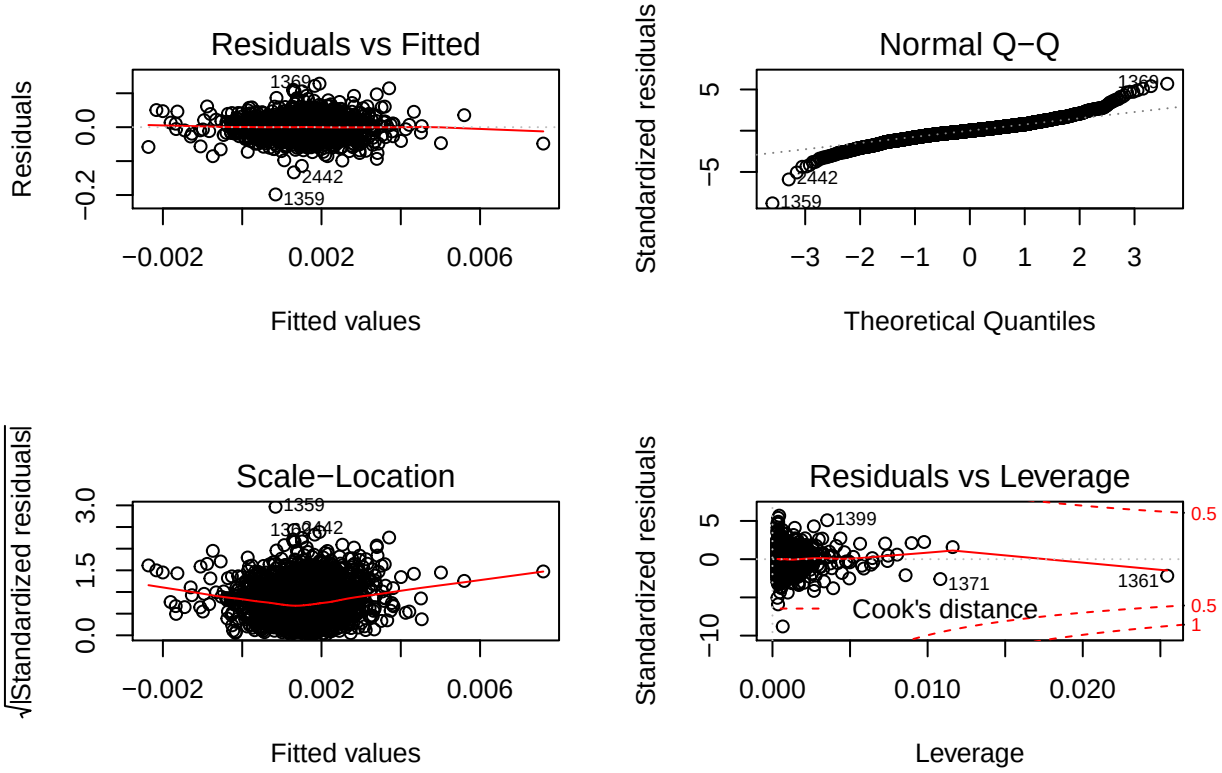
First, we can take a direct look at the correlation between the chosen variates in model 1:



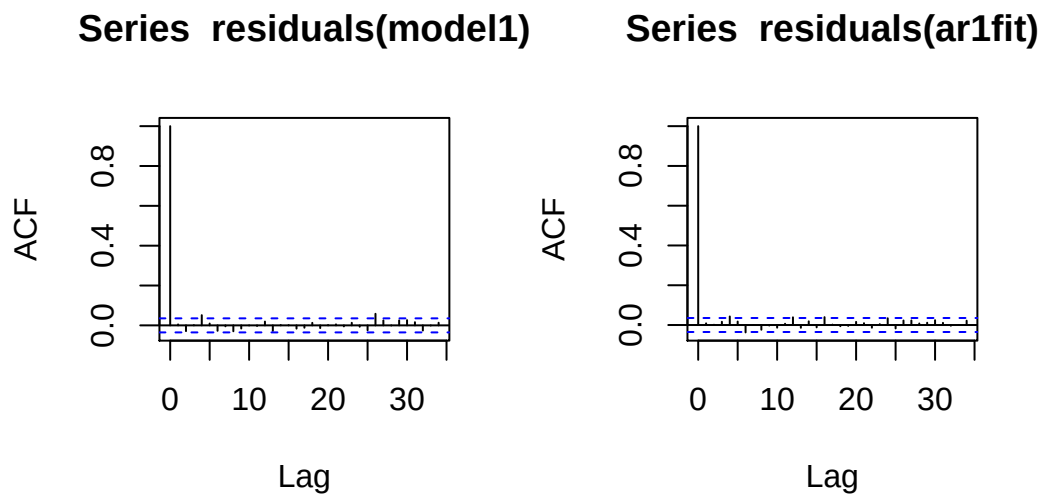
We can see clear relationships amongst the variates using model 1. There is certainly seems to be a positive correlation between all variates. When the S&P 500 goes up, both the return rate for AAPL and Emerging Markets index seem to go up.

Now we begin to use our models to come up with a series of plots to enrich our analysis. We want to model both the residuals and the standardized residuals. Using $plot(model)$ gives us 4 different plots to analyze.





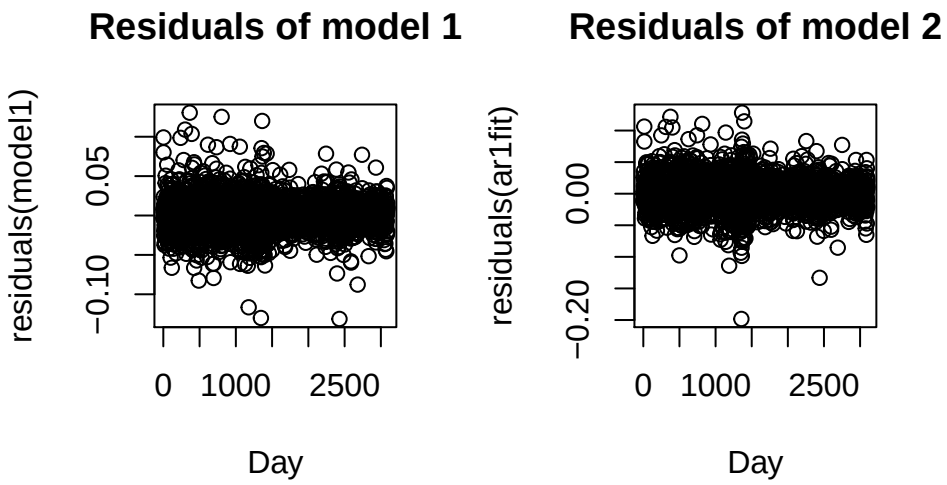
Now we use the autocorrelation function esitmate on our models:



We can clearly see from both plots that there is not a significant amount autocorrelation thus validating our models further. It also seems like our first model has less frequent autocorrelation spikes compared to model 2.

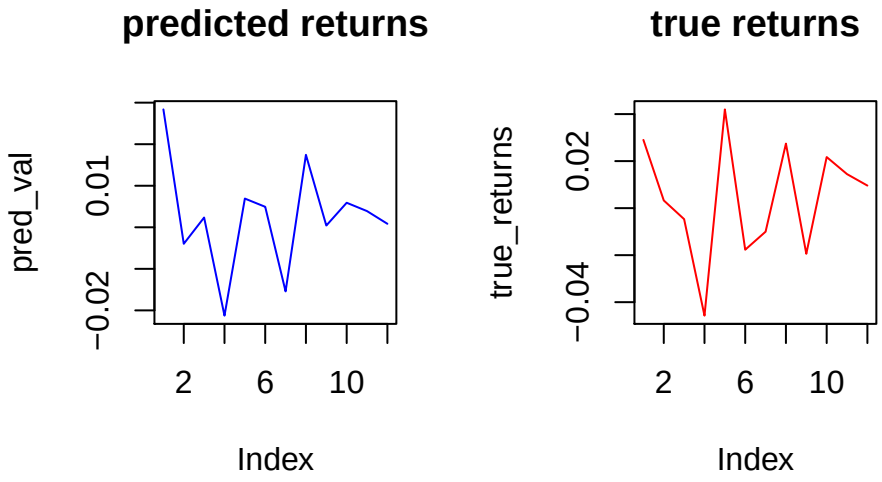
Clearly, model 1 has better performance than model 2. Looking at Fitted vs Residuals plot, the data in model 1 is closer to zero and more dispersed out. Looking at QQ plot, it is hard to tell since both plots have heavy tails on both sides. However we can clearly see that, model 1 fits more better than model 2 from our plots.

We can also clearly see "Black Swan" days on some of the plots as the points are labeled with the day. We can see that returns on days around 1300 tend to be more deviated away from the mean. From the plot, we can also see days around 1300-1500 tend to have a very large cook's distance suggesting that deleting these data points may result in a better fit for our model.



Predicting Values using our Model

Finally, we take a look at how well our model actually performs in contrast to the actual returns of the Apple stock data. Refer to the appendix to see how we come up with the values to come up with the prediction. We want to first plot the actual returns of the Apple data and then plot the returns that our predicted model came up with using the apporpirate variables then plot them against eachother.



Clearly, we can see that our prediction model captures some features of the actual return, however it did not predict it perfectly by any means. Our model seems to capture the long term trend of the actual returns suggesting that this model may be useful for investments over periods of weeks to months rather than higher freqeuncy trading. The model captures the general trend but does not capture extremes very well (ie looking

at the increase from index 4-5 on predicted vs actual returns shos that the model vastly underestimates the true return).

Now we shall calculate the errors. Refer to appendix for the calculation.

```
## [1] 0.003404858
```

We observe that our error is quite small which is a good indicator for our model's fit. With that being said, it does not necessarily mean that our model is perfect at predicting returns.

Discussion

Since, we did our analysis by converting all variables to returns first, From model 1, we can see that covariate SPX had the most impact on our model. Another variable called $AAPL$ also had a large influence on our model as well, but we drop it since it has the tendency to make the other values less significant. Therefore we concluded that the most important variables that influence our model the most are the returns of SPX and EEM variables. These two covariates had high p-values which makes sense because Apple is listed under the companies in the S&P 500 plus emerging markets (like China) performing well means more consumption of Apple products in these places.

As we disscuss in the beginning of our report, due to financial crisis in 2008, we might want to drop those days when we fit our model as they greatly influence our results. As observed in our plots, the 2008 financial crisis created many outliers in our data (specifically the residual plots) which were easily observable.

Since we are using autoregressive models in our analysis, we might want to try some time series approach rather than using linear models as discussed previously in the analysis. Futher imporvement would try to fit our model in ARCH or GARCH models. These models are perferred to financial time series data, since these models foucs on more volatility, rather than autoregressive models.

Appendix

RCode used in the report according to the order it appears.

```
#Code for timeseries plot and logreturns

layout(matrix(c(1,2),1,2))
ts.plot(data$AAPL, gpars=list(ylab="Daily Price", xlab="Day"))

data.ts = ts(data$AAPL)
logreturns_data.ts <- log(lag(data.ts)/data.ts)
plot(logreturns_data.ts, ylab = "logreturns")
```

```
#Create plot for APPL_i vs APPL_i-1

x = data$AAPL
x = ts(x)
lag1.plot(x,1)
```

```
#Get our log returns for APPL_i-1

logreturns = diff(log(data$AAPL), lag=1)
```

```
#Create a dataframe of all important varaibles with their logs taken

logeem = diff(log(data$EEM), lag=1)
logspx = diff(log(data$SPX), lag=1)
logvix = diff(log(data$VIX), lag=1)
logspg = diff(log(data$SPGSCITR), lag=1)
logbnd = diff(log(data$BNDGLB), lag=1)

returns = as.data.frame(logreturns)
```

```
# Create i-1 data for all covariates and add them to the returns data frame


lag1 = function (x) c(NA, x[1:(length(x)-1)])
lag1_logret = lag1(returns$logreturns)
lag1_logeem = lag1(logeem)
lag1_logspx = lag1(logspx)
lag1_logvix = lag1(logvix)
lag1_logspg = lag1(logspg)
lag1_logbnd = lag1(logbnd)
returns$lag1 = lag1_logret
returns$lag1eem = lag1_logeem
returns$logeem = logeem
returns$lag1spx = lag1_logspx
returns$lag1vix = lag1_logvix
returns$lag1spg = lag1_logspg
returns$lag1bnd = lag1_logbnd

returns = returns[-1,]
```

```
# Fit our first linear model initially using all covariates

model1 = lm(logreturns~., data=returns)
```

```r
summary(model1)
```

```
##
## Call:
## lm(formula = logreturns ~ ., data = returns)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.199022 -0.011357 -0.000146  0.011721  0.128641
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.0015490  0.0004047   3.827 0.000132 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02255 on 3102 degrees of freedom
```

```r
# Durbin-Watson Test on the first model

dwtest(model1)
```

```r
# Obtaining the second model by using a lag of 2

data.ts = ts(data$AAPL)
logreturns_data.ts <- log(lag(data.ts)/data.ts)
#lag1.plot(logreturns_data.ts,1)
#acf(logreturns_data.ts)
xlag1=lag(logreturns_data.ts,-1)
xlag2=lag(xlag1,-1)
xlag3=lag(xlag2,-1)
lag1eem = lag(lag1_logeem, -1)
lag2eem = lag(lag1eem, -1)
y=cbind(logreturns_data.ts,xlag1,xlag2,xlag3, lag1eem)
ar1fit=lm(y[,1]~y[,3])
summary(ar1fit)
```

```
##
## Call:
## lm(formula = y[, 1] ~ y[, 3])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.19831 -0.01122 -0.00011  0.01165  0.12824
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.0015922  0.0004057   3.925 8.87e-05 ***
## y[, 3]      -0.0303913  0.0179510  -1.693   0.0906 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02254 on 3099 degrees of freedom
##   (5 observations deleted due to missingness)
## Multiple R-squared:  0.0009241,  Adjusted R-squared:  0.0006017
```

```
## F-statistic: 2.866 on 1 and 3099 DF,  p-value: 0.09055
##
##  Durbin-Watson test
##
## data:  ar1fit
## DW = 1.9829, p-value = 0.3169
## alternative hypothesis: true autocorrelation is greater than 0
```

```
# Plots of our models (residual plots, qqplot)

layout(matrix(c(1,2,3,4),2,2))
plot(model1)
layout(matrix(c(1,2,3,4),2,2))
plot(ar1fit)
```

```
#Scatterplot Matrix of Model 1 covaraites

pairs(lag1_logret~lag1_logspx+lag1_logeem)
```

```
# Plots of residuals for model2 transformed

plot(residuals(ar1fit))
plot(residuals(ar1fit_z))
```

```
#Creating crossvaladation for linear model

data_new$logreturns = zlag1
cv.lm(df = data_new, form.lm = lm(data_new$logreturns~data_new$EEM), m=3)
```

```
#Plots of the predicted returns and true returns

par(mfrow=c(1,2))
pred_val = predict(model1, newdata = returns_test) #predicted  returns
true_returns = returns_test$logreturns             #true returns
plot(pred_val, type = "l", col = "blue", main = "predicted returns")
plot(true_returns, type = "l", col = "red", main = "true returns")
```

```
# Calculate the sum of squared errors of prediction of returns

SSE = sum((pred_val-true_returns)^2)
SSE
```

References

[1] Julian J. Faraway, *Regression and Anova using R*, (University of Bath, 2002).

[2] *Multiple Linear Regression*, available at `https://www.statmethods.net/stats/regression.html`.

[3] *Timeseries*, available at `https://www.statmethods.net/advstats/timeseries.html`.

[4] *Lag function*, available at `https://stats.stackexchange.com/questions/92498/forecasting-time-series-regression-in-r-using-lm`.

[5] *Autoregressive model*, available at `https://onlinecourses.science.psu.edu/stat501/node/358`.