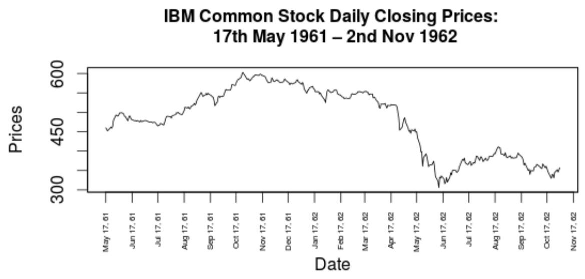
Time Series Project

Heran Song

Introduction:

This paper will provide the procedure to analyze and model Conditional Heteroscedastic Time Series Models using R. The paper can be broken down into three sections, first section will cover the process behind the selection of the ARIMA/GARCH model. Second section will be covering model diagnostics, and the last section will be on forecasting predictions using the selected model. The data used in the analysis is the IBM common stock daily closing prices May 17, 1961 to November 2, 1962, which is provided by "Time Series Data Library" (citing "Box & Jenkins (1976)"). Just by looking at plot of the data, we can tell there are some unusual signs indicated by the drastic dip of the stock around the month of May, 1962. From history, we can tell this is the "flash crash" of the stock market at 1962.

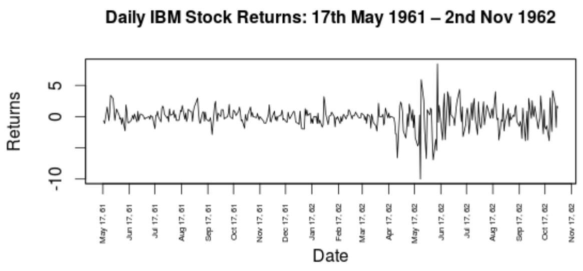
```
> library(rdatamarket);library(TSA);library(zoo);library(forecast)
> IBM<-dmseries("http://datamarket.com/data/set/2322/ibm-common-stock-closing-prices
-daily-17th-may-1961-2nd-november-1962#!ds=2322&display=line")
```



```
> plot(IBM,xlab="Date",xaxt="n",ylab="Prices",main="IBM Common Stock Daily Closing Prices:
\n 17th May 1961  2nd Nov 1962",cex.main=1)
> td<-seq(as.Date("1961/5/17"), as.Date("1962/11/17"),"months")
> axis(1, td, format(td, "%b %d, %y"), cex.axis = .5,las="2")
```

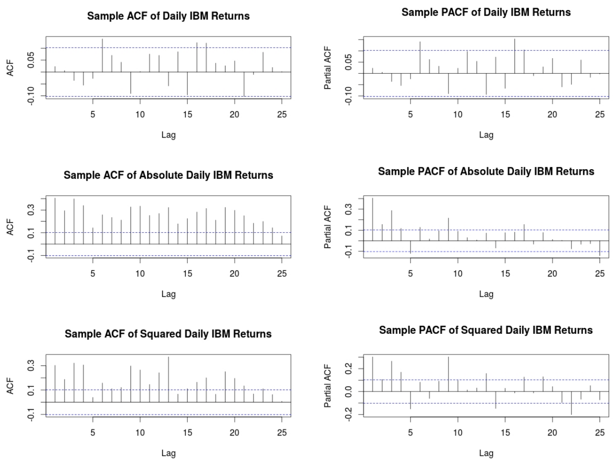
Model Selection:

The first step in selecting the model is to convert a non-stationary process to a stationary one. We can eliminate non-stationary process by taking the difference of the log prices to compute the return, and multiply by 100 so that they can be interpreted as percentage changes in the price.



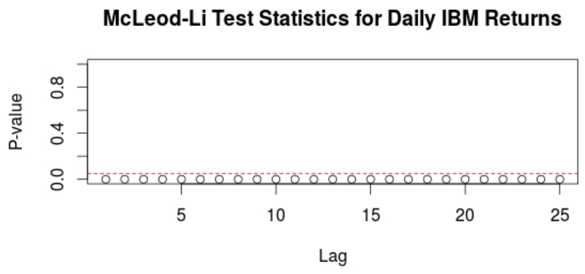
```
> r.IBM<-diff(log(IBM))*100
> plot(r.IBM,xaxt="n",xlab="Date",ylab="Returns",main="Daily IBM Stock Returns:
17th May 1961  2nd Nov 1962",cex.main=1)
> abline(h=0)
> axis(1, td, format(td, "%b %d, %y"), cex.axis = .5,las="2")

> t.test(r.IBM)

One Sample t-test

data:  r.IBM
t = -0.7444, df = 367, p-value = 0.4571
alternative hypothesis: true mean is not equal to 0
```

After transforming our data, the mean is insignificant and volatility clustering seems to appear in our data. This is indicated by the pattern of alternating quiet and volatile periods of substantial duration, which is caused by the conditional variance of the time series varying over time. Volatility clustering observed in the IBM return data gives us a hint that they may not be independently and identically distributed. To investigate further, we look at nonlinear transformations to the ACF and PACF of returns since independence is preserved by these transformations, not correlation.



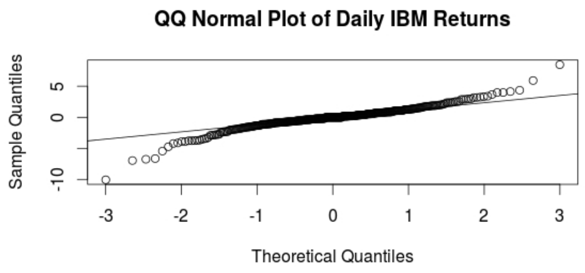
```
> IBM.ts<-as.ts(coredata(r.IBM))
> acf(IBM.ts,main="Sample ACF of Daily IBM Returns")
> pacf(IBM.ts,main="Sample PACF of Daily IBM Returns")
> acf(abs(IBM.ts),main="Sample ACF of Absolute Daily IBM Returns")
> pacf(abs(IBM.ts),main="Sample PACF of Absolute Daily IBM Returns")
> acf(IBM.ts^2,main="Sample ACF of Squared Daily IBM Returns")
> pacf(IBM.ts^2,main="Sample PACF of Squared Daily IBM Returns")
```

The plots of the absolute and squared transformations of the sample ACF and PACF of returns seems to support the argument that the observations are not independently identically distributed. To further investigate the claim of non i.i.d, we use the McLeod-Li test, which has null hypothesis of squared data are not auto-correlated.



```
> McLeod.Li.test(y=IBM.ts,main="McLeod-Li Test Statistics for Daily IBM Returns")
```

The plot show all tests are significant at the 5% level, which allows us to reject the null hypothesis for not auto-correlated.
To see if the distribution of the returns are normally distributed, we use a QQ normal plot for the visual test. To investigate further, we use the Shapiro-Wilk normality test which has the null hypothesis of normality. The Jarque-Bera test also test for normal distribution with the null hypothesis of zero kurtosis(normal distribution).



```
> qqnorm(IBM.ts,main="QQ Normal Plot of Daily IBM Returns");qqline(IBM.ts)

> shapiro.test(IBM.ts)

Shapiro-Wilk normality test

data:  IBM.ts
```

```
W = 0.9249, p-value = 1.244e-12

> kurtosis(IBM.ts)
[1] -1.263461
attr(,"method")
[1] "excess"

> jarque.bera.test(IBM.ts)

Jarque Bera Test

data:  IBM.ts
X-squared = 437.9745, df = 2, p-value < 2.2e-16
```

The QQ normal plot indicates a non-normal distribution. The result of the two tests with p-values of 1.244e-12 and 2.2e-16 respectively also points to non-normality by the rejection of the two null hypothesis.

To find the ARIMA component of our model, we use the auto.arima function in the package "forecast", which returns the best ARIMA model according to either AIC, AICc or BIC.

```
> log.IBM<-as.ts(log(coredata(IBM)))
> arima.fit<-auto.arima(log.IBM)
> arima.fit
Series: log.IBM
ARIMA(0,1,0)

sigma^2 estimated as 0.0003148:  log likelihood=961.55
AIC=-1921.1   AICc=-1921.09   BIC=-1917.19
```

The function selected a ARIMA(0,1,0).
To find the order of the GARCH component, we look at the EACF plots of the Absolute and Squared values of the time series.

```
> eacf(abs(IBM.ts))
AR/MA
  0 1 2 3 4 5 6 7 8 9 10 11 12 13
0 x x x x x x x x x x x  x  x  x
1 x x x x x x o o o o o  o  x  x
2 x x x x x x o o o x o  o  x  x
3 x x x o x x x o o o o  o  o  x
4 x o x x x o x o o o o  o  o  x
5 x o x o x o x o x o o  o  o  x
6 x x x o x x x o o o o  o  o  o
7 x o x x x o x o o o o  o  o  o
> eacf(IBM.ts^2)
AR/MA
```

```
  0 1 2 3 4 5 6 7 8 9 10 11 12 13
0 x x x x o x x x x x x  x  x  o
1 x x x x o x o o x x o  o  x  x
2 x x o x x o o o x x x  o  x  x
3 x x x x x o x o o o o  o  x  x
4 x o o o x x x o o o o  o  x  x
5 x x o o x o x o x o o  o  x  o
6 x o o o x x x o x o o  o  x  x
7 x o o o x o x o x o o  o  x  o
```

The EACF plots doesn't really seem to be very helpful, so we just have to do it by trial and error by looking at the AIC and the significance of the coefficients. The first model to try is a GARCH(2,2) model.

```
> garch22.fit<-garch(x=arima.fit$res, order=c(2,2))
> summary(garch22.fit)

Call:
garch(x = arima.fit$res, order = c(2, 2), reltol = 1e-06)

Model:
GARCH(2,2)

Residuals:
    Min      1Q  Median      3Q     Max
-3.9080 -0.6353  0.0000  0.5767  3.5782

Coefficient(s):
    Estimate  Std. Error  t value Pr(>|t|)
a0 9.690e-06   4.300e-06    2.253   0.0242 *
a1 2.897e-01   6.642e-02    4.362 1.29e-05 ***
a2 3.147e-07   1.021e-01    0.000   1.0000
b1 2.662e-01   3.279e-01    0.812   0.4169
b2 4.250e-01   2.580e-01    1.647   0.0995 .
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

Only 2 out of the 5 coefficient is significant at the .05 level. Therefore, we look at a lower order model.

```
> garch11.fit<-garch(x=arima.fit$res, order=c(1,1) ,reltol=0.000001)
> summary(garch11.fit)

Call:
garch(x = arima.fit$res, order = c(1, 1), reltol = 1e-06)
```

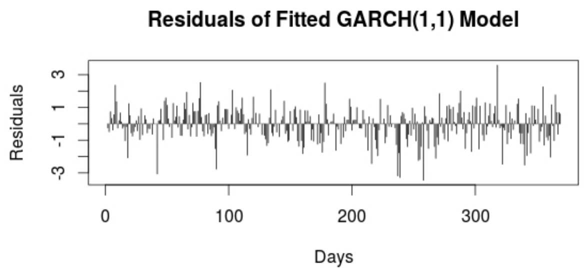
```
Model:
GARCH(1,1)

Residuals:
    Min      1Q  Median      3Q     Max
-3.4493 -0.6337  0.0000  0.5803  3.5849

Coefficient(s):
    Estimate  Std. Error  t value Pr(>|t|)
a0 8.927e-06   2.859e-06    3.123  0.00179 **
a1 2.542e-01   4.382e-02    5.801 6.58e-09 ***
b1 7.361e-01   4.072e-02   18.077  < 2e-16 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

> AIC(garch22.fit)
[1] -2091.349
> AIC(garch11.fit)
[1] -2100.303

> confint(garch11.fit)
          2.5 %       97.5 %
a0 3.323876e-06 1.453053e-05
a1 1.683243e-01 3.400939e-01
b1 6.562670e-01 8.158810e-01
```

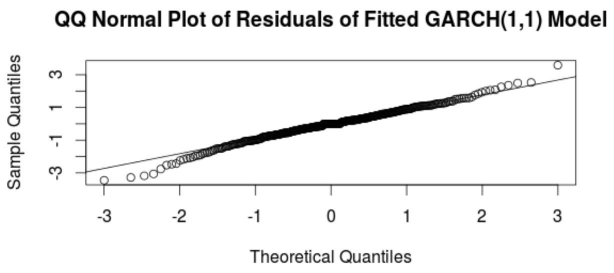
The GARCH(1,1) model seems like a better fit with both coefficients being significant and a lower AIC value comparing to the GARCH(2,2) model. The 95% confidence interval for each coefficient is also computed. The intervals are not very wide, also a sign of good fit.

Our final model seems to be a ARIMA(0,1,0)+GARCH(1,1). To check our model, we look at the residual analysis and the overall fit against the actual data.

Model Diagnostic:

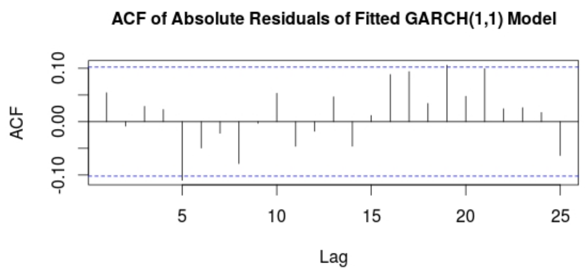
To check for normality of the residuals, we again plot the QQ normal plot, use the Shapiro-Wilk test, and the Jarque-Bera test.

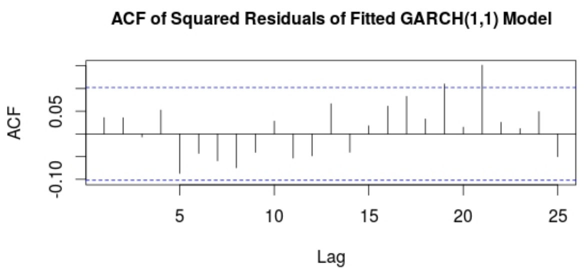


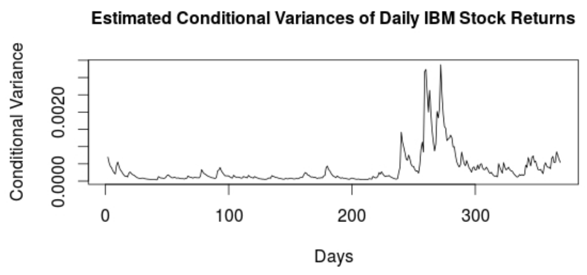


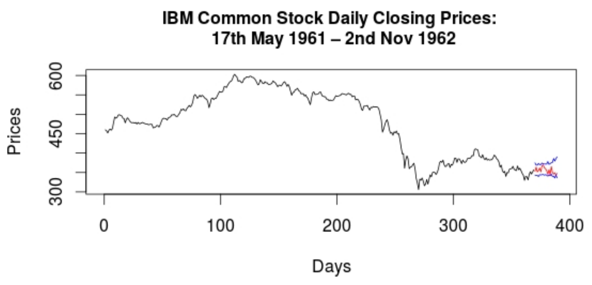
```
> plot(residuals(garch11.fit),type="h",xlab="Days",ylab="Residuals",main="Residuals
of Fitted GARCH(1,1) Model")
> qqnorm(residuals(garch11.fit),main="QQ Normal Plot of Residuals of Fitted
GARCH(1,1) Model"); qqline(residuals(garch11.fit))

> shapiro.test(residuals(garch11.fit))

Shapiro-Wilk normality test

data:  residuals(garch11.fit)
W = 0.9876, p-value = 0.00317

> jarque.bera.test(residuals(garch11.fit)[-1][-1])

Jarque Bera Test

data:  residuals(garch11.fit)[-1][-1]
X-squared = 18.8394, df = 2, p-value = 8.111e-05
```

The QQ plot and both tests still suggests that the residuals are not normally distributed.
To investigate auto-correlation, we again plot the absolute and squared ACF and PACF, use the Ljung-Box test with null hypothesis of zero auto-correlation.



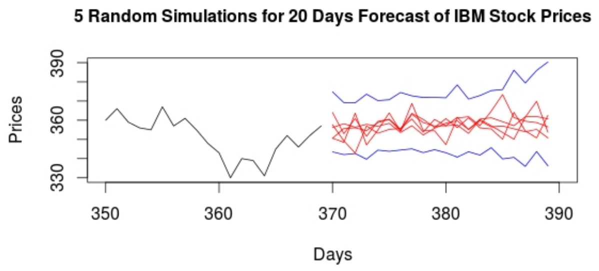






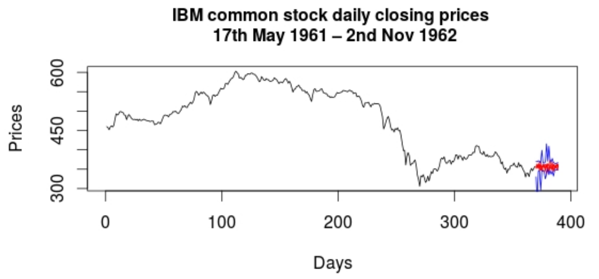
```
> acf(abs(residuals(garch11.fit)),na.action=na.omit,main="ACF of Absolute Residuals of
Fitted GARCH(1,1) Model",cex.main=1)
> pacf(abs(residuals(garch11.fit)),na.action=na.omit,main="PACF of Absolute Residuals of
Fitted GARCH(1,1) Model",cex.main=1)
> acf((residuals(garch11.fit))^2,na.action=na.omit,main="ACF of Squared Residuals of
Fitted GARCH(1,1) Model",cex.main=1)
> pacf((residuals(garch11.fit))^2,na.action=na.omit,main="PACF of Squared Residuals of
Fitted GARCH(1,1) Model",cex.main=1)

> LB.test(garch11.fit)

Box-Ljung test

data:  residuals from  garch11.fit
X-squared = 10.0963, df = 12, p-value = 0.6075
```

The plots seem to indicate zero auto-correlation. The result of the Box-Ljung test also supports the argument with a p-value of 0.6075.
To check the fit of our model, we plot the model(in red) with the actual data(in black).



```
> plot(log.IBM,xlab="Days",ylab="Prices",main="Log of IBM Stock
Daily Closing Prices")
> lines(fitted(arima.fit)+fitted(garch11.fit)[,1]^2,col="red")
```

The model seems like a good fit to the actual data.
Lets take a look at the estimated conditional variance of our model.



```
> plot((fitted(garch11.fit)[,1])^2,type="l",ylab="Conditional Variance",xlab="Days",
main="Estimated Conditional Variances of Daily IBM Stock Returns",cex.main=1)
```

The conditional variances plot successfully reflects the volatility of the time series over the entire period, and the high volatility is closely related to period where stock price dropped drastically.

Forecasting:

To test our model, we cut off the last 10 observations of our data then fit the model and compute the relative absolute error.

```
> pred1<-fitted(arima.fit)[(length(IBM)-9):length(IBM)]
> set.seed(123456)
> garch.sim1<-garch.sim(alpha=c(garch11.fit$coef[1],garch11.fit$coef[2]),
beta=garch11.fit$coef[3],n=10)
> mod.pred<-exp(pred1+garch.sim1)
> abs.error=abs(mod.pred-as.ts(coredata(IBM))[(length(IBM)-9):length(IBM)])
> rel.error=round((abs.error/as.ts(coredata(IBM))[(length(IBM)-9):length(IBM)])*100,3)
> rel.error
 [1] 0.616 5.071 4.163 0.965 3.376 3.033 3.364 1.733 2.733 0.001
```

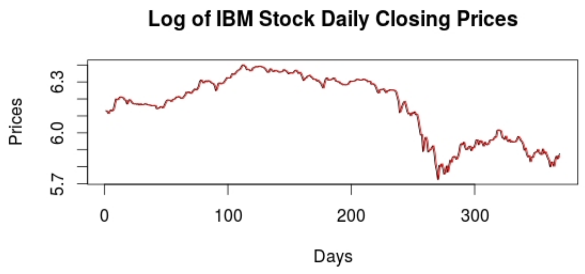
Here are the 10 values of the relative absolute value in percentages. The low values indicate the goodness of fit.

To forecast using our model, we use Monte Carlo method to compute expected value and 95% confidence intervals from 10000 random simulations of the 20 days ahead prediction.

```
> pred<-predict(arima.fit,n.ahead=20)$pred
> iterations=10000;steps=20
> y=matrix(nrow=iterations,ncol=steps)
> for(i in 1:iterations){
+   set.seed(sample(1:99999,1))
+   y[i,]=exp(pred+garch.sim(alpha=c(garch11.fit$coef[1],garch11.fit$coef[2]),
beta=garch11.fit$coef[3],n=steps))
+ }
> means=colMeans(y)
> conf.int<-apply(y,2,function(x){quantile(x,c(.025,.975))})
> conf.int.ts<-ts(t(conf.int),start=370)
> round(cbind("Lower Bound"=conf.int.ts[,1],"Mean Predictions"=means,
"Upper Bound"=conf.int.ts[,2]),2)
Time Series:
Start = 370
End = 389
Frequency = 1
    Lower Bound Mean Predictions Upper Bound
370      343.52           356.99      374.80
371      342.08           356.58      369.22
372      342.65           356.73      369.09
373      339.67           356.56      373.58
374      344.44           357.07      370.23
375      343.80           357.18      370.69
376      344.45           357.36      374.46
377      345.08           358.24      372.73
```

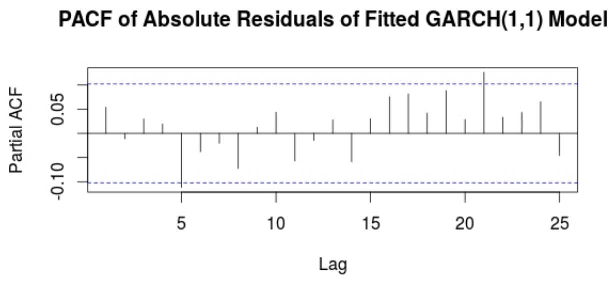
378	343.12	356.16	371.84
379	344.65	356.30	371.89
380	343.07	357.13	371.67
381	340.74	357.67	378.43
382	343.58	358.71	371.02
383	341.81	355.93	372.86
384	345.70	357.68	375.45
385	339.82	357.08	376.01
386	340.64	358.25	386.11
387	335.93	357.42	379.39
388	343.68	359.21	385.93
389	336.20	357.78	390.34

Here is 1 random simulations(in red) for the 20 days ahead prediction of our model with a 95% confidence interval(in blue) plotted with the over all data.



```
> plot(as.ts(coredata(IBM)),xlim=c(0,390),xlab="Days",ylab="Prices",main="IBM
Common Stock Daily Closing Prices: \n 17th May 1961  2nd Nov 1962",cex.main=1)
> set.seed(sample(1:99999,1))
> garch.sim1<-garch.sim(alpha=c(garch11.fit$coef[1],garch11.fit$coef[2]),
beta=garch11.fit$coef[3],n=20)
> mod.pred<-exp(pred+garch.sim1)
> lines(ts(mod.pred,start=370),col="red")
> lines(conf.int.ts[,1],col="blue");lines(conf.int.ts[,2],col="blue")
```

Here is a closer look at 5 random simulations.



```
> plot(window(as.ts(coredata(IBM)),start=350),xlim=c(350,390),ylim=c(330,390),xlab="Days"
,ylab="Prices",main="5 Random Simulations for 20 Days Forecast of IBM Stock Prices",
cex.main=1)
> lines(conf.int.ts[,1],col="blue");lines(conf.int.ts[,2],col="blue")
> for(n in 1:5){
+ set.seed(sample(1:99999,1))
+ garch.sim1<-garch.sim(alpha=c(garch11.fit$coef[1],garch11.fit$coef[2]),
beta=garch11.fit$coef[3],n=20)
+ mod.pred<-exp(pred+garch.sim1)
+ lines(ts(mod.pred,start=370),col="red")
+ }
```

Conclusion:

Our final model seems to fit well with the actual data and the GARCH component of our model seems to fully incorporate the volatility of our time series. The forecasting power of our model does not seem very powerful because of our wide confidence interval.