TSA Computer Lab 3

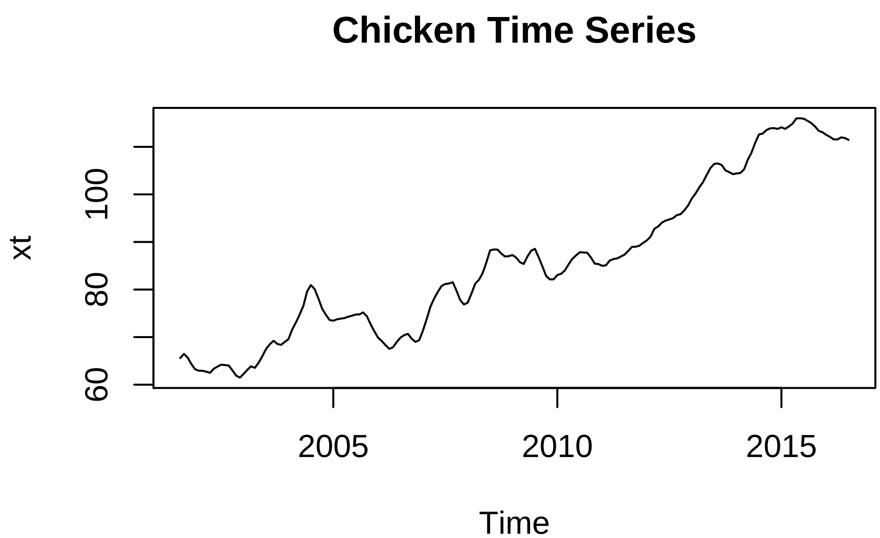
Joshua Hudson, Carles Sans

10 October 2017

Assignment 1. Spectral analysis of the chicken prices

1.

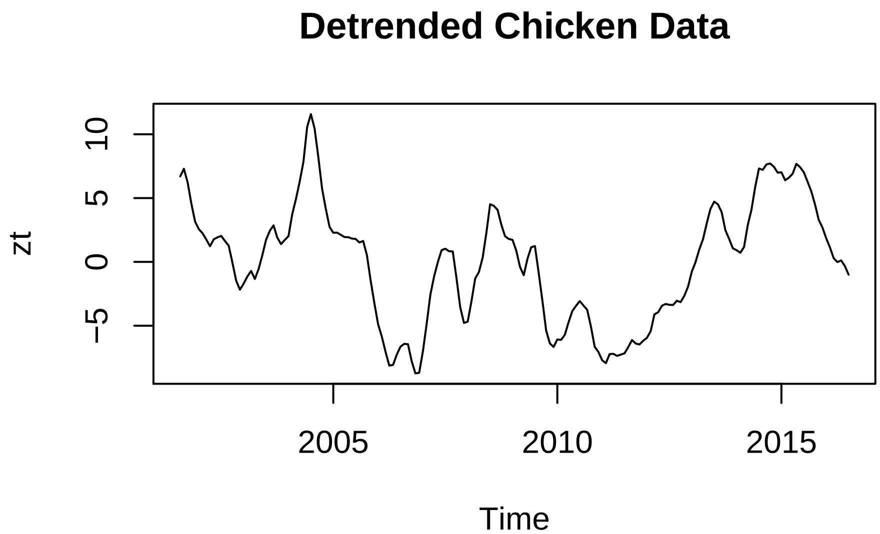
In this assignment we used the `chicken` time series data from the `astsa` package. We started by plotting the data, as shown below.



From visual inspection, the trend of the time series could be linear or possibly quadratic.

2.

The next step was to detrend the data using a linear model. We plotted the detrended daa below and named this time series z_t.



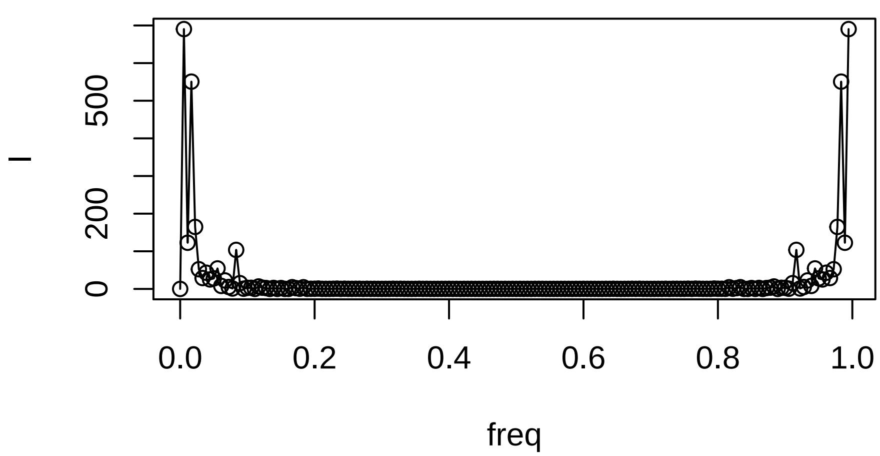
It is difficult to tell if the new time series is stationary or not by just looking at the plot. The mean seems to be around 0 and the variance seems to be finite so it could be stationary.

3.

Now we used a frequency domain approach to mode z_t. The first step was to compute the periodogram to identify the dominant frequencies. To do so , we applied the Fast Fourier Transform using the `fft()` function, transforming the output into the classic FFT form and finally computing the periodogram I using:

$$I = |d(\frac{j}{n})|^2$$

The periodogram is plotted below.



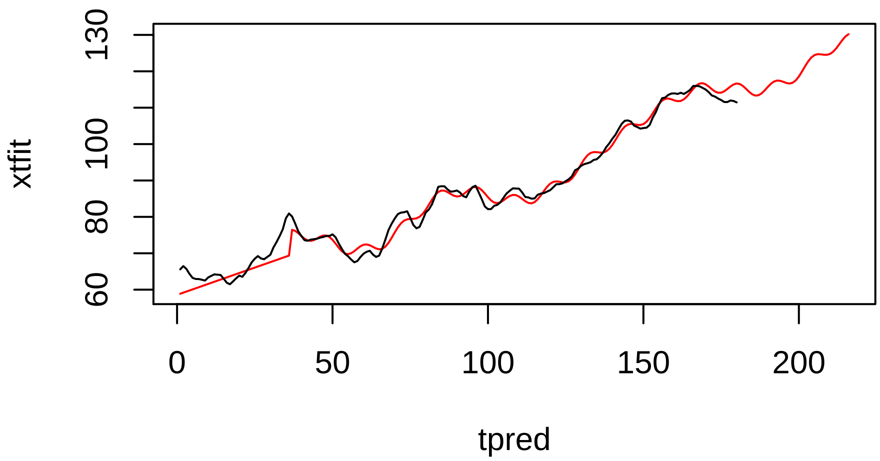
The 1st to the 4th fourier frequencies as well as the 15th appear to be dominant. For this reason, we considered a baseline of $I_0 = 100$. We computed the confidence interval for the 1st frequency (the one with the largest amplitude).

```
## [1]   187.1535 27268.8169
```

We can see that the lower bound is higher than our baseline I_0, so it seems to be appropriate.

1.4

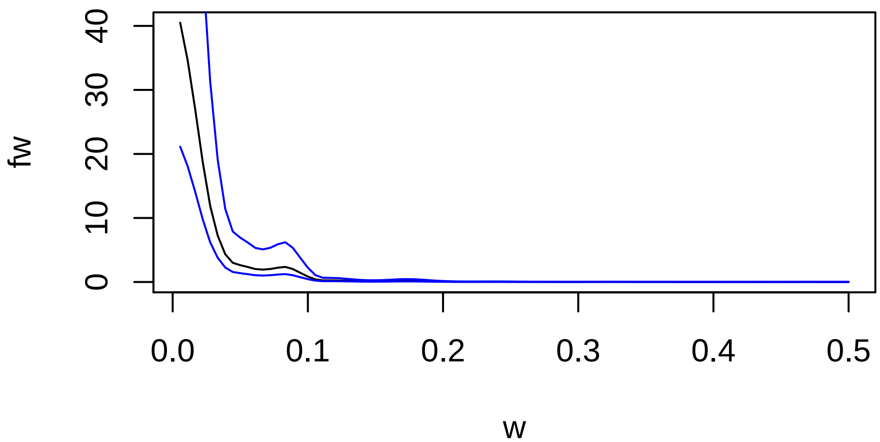
We then created a frequency model using these dominant frequencies using the Inverse Fourier Transform. We used this model to obtain filtered data and make 36-step ahead predictions for z_t. Finally we added the trend back in to get a fit for the original `chicken` data, which is plotted below.



The forecast looks reasonable as it follows the trend and the variation pattern looks similar to that of the data.

1.5

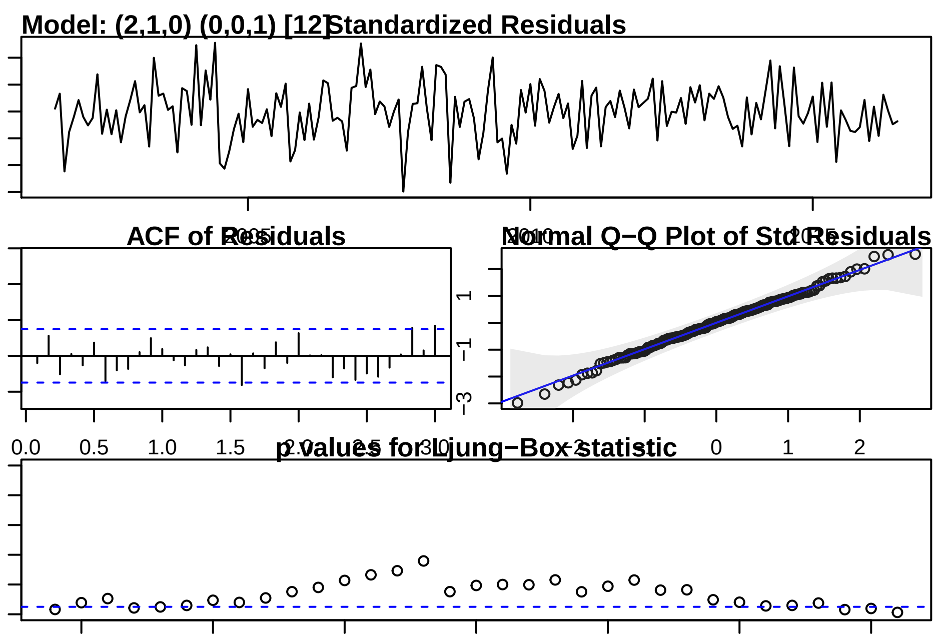
Next we looked into spectral analysis. We computed a smoothed spectrum of z_t using a non-parametric estimation with a ModifiedDaniell(2, 2) kernel. We also computed the confidence band for each frequency and plotted the results below.



From the spectrum we decided to take as a threshold frequency $w_0 = 0.1$ as as this value, the spectrum is negligible (white noise). This is the equivalent to saying that the first 15 fourier frequencies are present, while previously we found that the 5th to 14th were not.Therefore the smoothing helped identify these missing dominant frequencies.

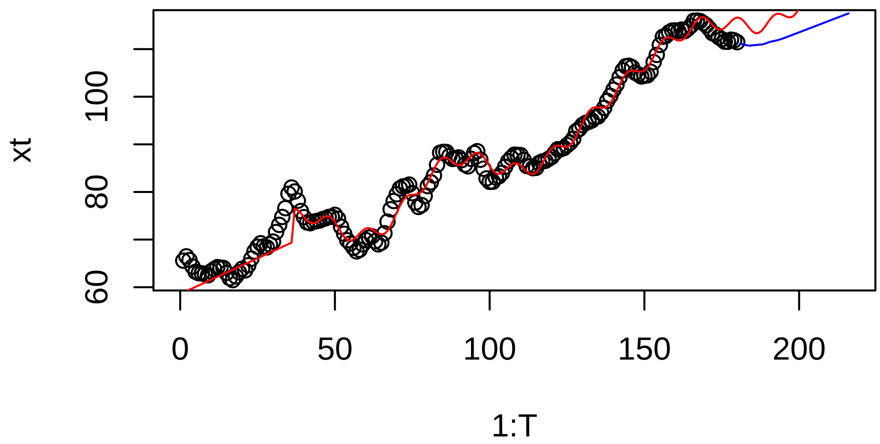
1.6

Next we fitted the $ARIMA(2,1,0) \times (0,0,1)_{12}$ to the origianl chicken data, using `sarima()` and analysed the diagnostic plots.



The model seems appropriate seeing as the ACF of residuals shows only white noise, the QQ shows normality of residuals and the LB plot p-values are outside the bands, meaning that the residual are independent.

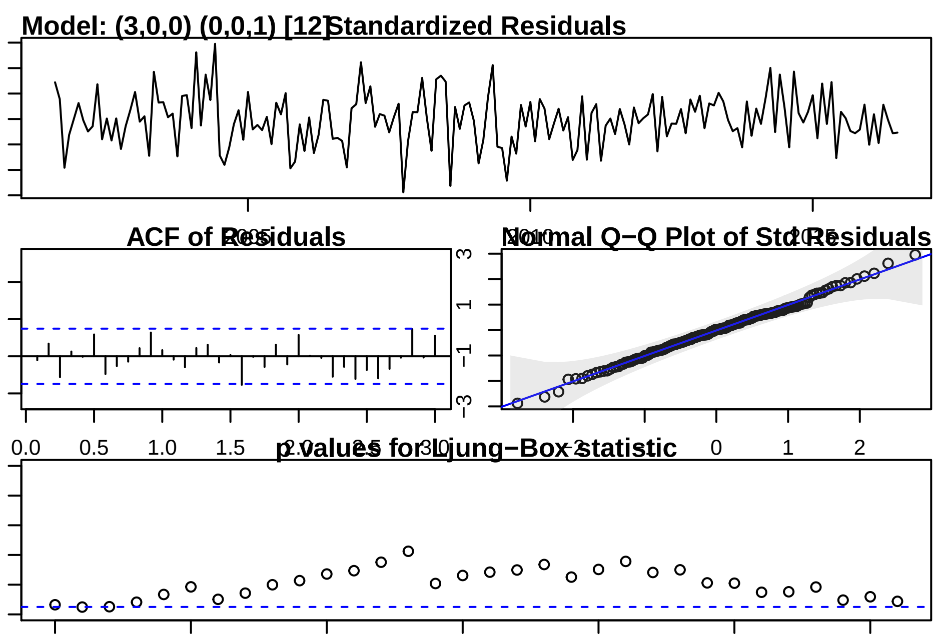
We then predicted 36 steps ahead and compared to the findings from the frequency model forecasting.



The prediction here follows the trend but does not incorporate the variation of the residuals. Therefore we would take the frequency model forecast instead.

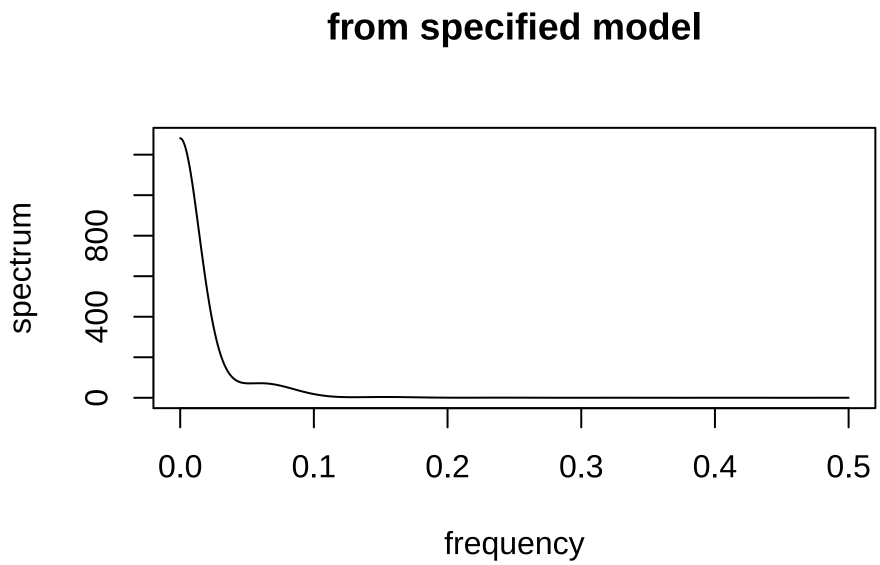
1.7

Finally, we fitted the $ARIMA(3,0,0) \times (0,0,1)_{12}$ to the detrended data, using `sarima()` and analysed the diagnostic plots.



The model seems appropriate seeing as the ACF of residuals shows only white noise, the QQ shows normality of residuals and the LB plot p-values are outside the bands, meaning that the residual are independent.

We computed the spectral density of the fitted ARIMA model and plotted below:

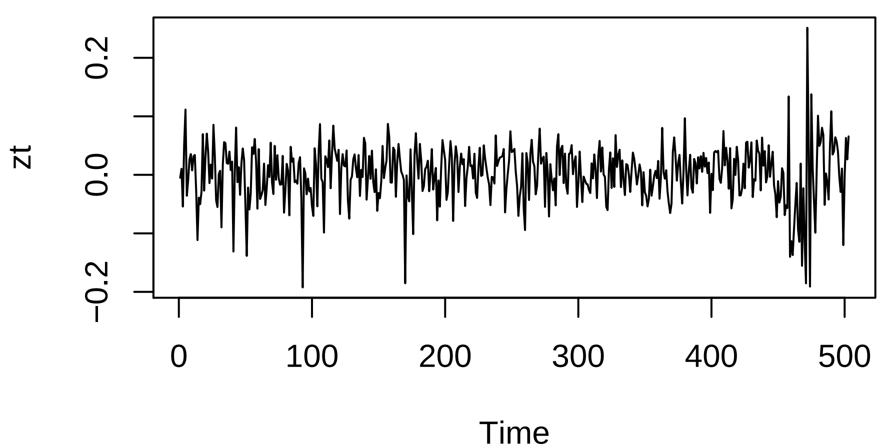


It looks similar to the spectral plot we obtained using the frequency model, tailing off close to frequency 0.1.

Assignment 2: GARCH modeling of oil prices

1.

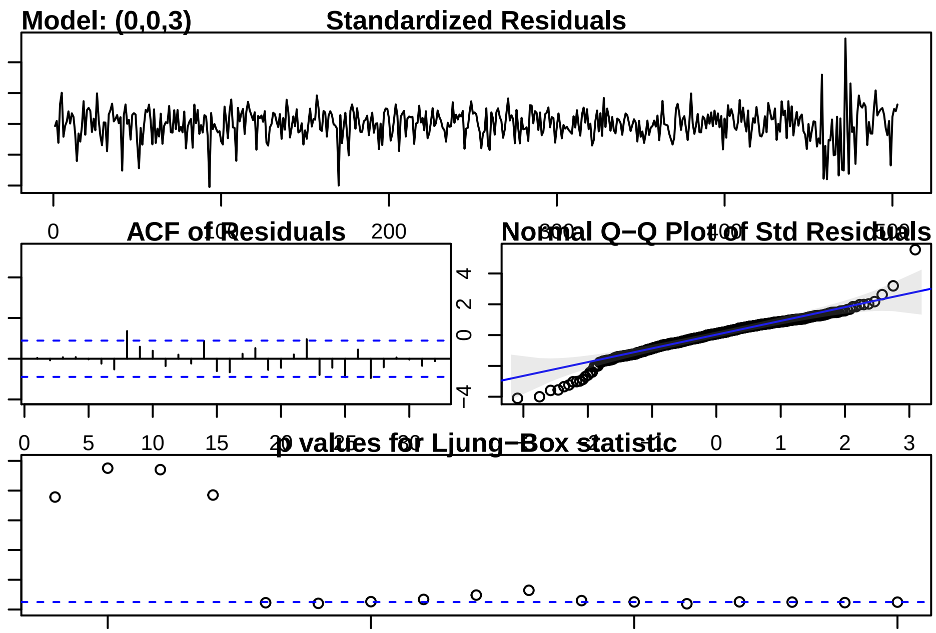
In this assignment we used the `oil` data set from the `astsa` package. We took the difference log of oil and named it x_t, and keeping observations until the 33rd week of 2009 naming it z_t. We plotted the data and looked at the empyrical ACF that can be found below in order to make our assessments.




```
## AR/MA
##   0 1 2 3 4 5 6 7 8 9 10 11 12 13
```

```
## 0 x o x o o o o x o o o  o  o  o
## 1 x o x o o o o x o o o  o  o  o
## 2 x x x o o o o x o o o  o  o  o
## 3 x x x o o o o x o o o  o  o  o
## 4 x o x o o o o x o o o  o  o  o
## 5 x x x o x o o x o o o  o  o  o
## 6 o x x o x x o x o o o  o  o  x
## 7 o x x x x x x x o x o  o  o  o
```

We suggested the ARMA(0,3) and looked at the diagnostic plots.



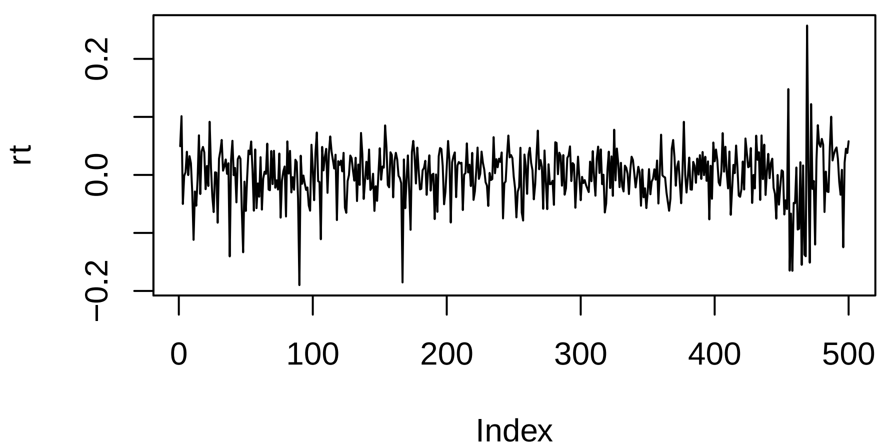

```
## $fit
##
## Call:
## stats::arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D,
##     Q), period = S), xreg = xmean, include.mean = FALSE, optim.control = list(trace = trc,
##     REPORT = 1, reltol = tol))
##
## Coefficients:
##          ma1      ma2     ma3   xmean
##       0.1617  -0.0918  0.1537  0.0020
## s.e.  0.0440   0.0435  0.0448  0.0025
##
## sigma^2 estimated as 0.002143:  log likelihood = 831.75,  aic = -1653.5
##
## $degrees_of_freedom
## [1] 499
##
## $ttable
##       Estimate     SE t.value p.value
## ma1     0.1617 0.0440  3.6721  0.0003
## ma2    -0.0918 0.0435 -2.1087  0.0355
## ma3     0.1537 0.0448  3.4319  0.0006
## xmean   0.0020 0.0025  0.7976  0.4255
```

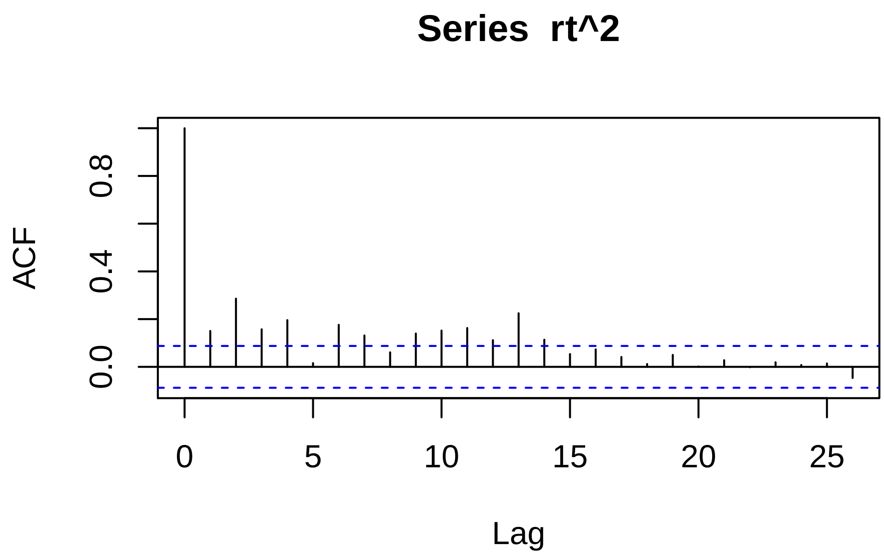
```
##
## $AIC
## [1] -5.129424
##
## $AICc
## [1] -5.125208
##
## $BIC
## [1] -6.09586
```

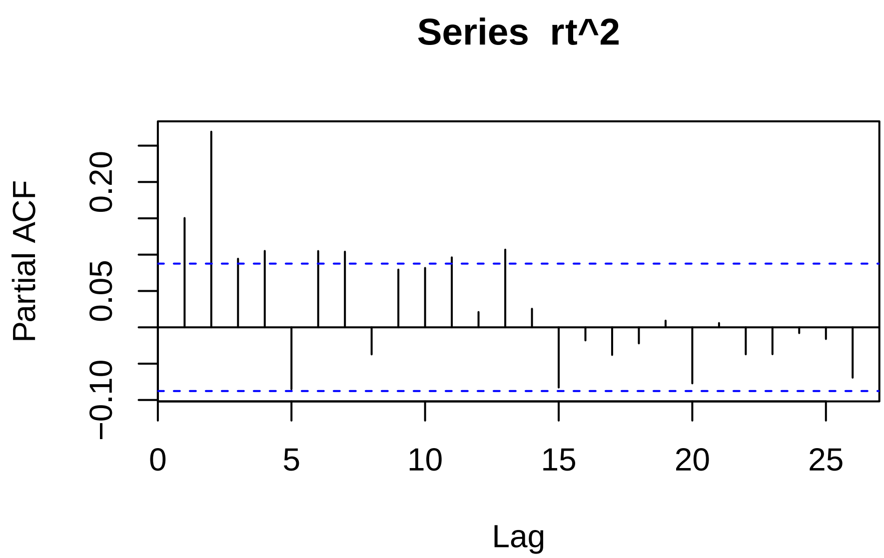
The model seems appropriate seeing as the ACF of residuals shows only white noise, the QQ shows normality of residuals and the LB plot p-values are outside the bands, meaning that the residual are independent for lags 1-4.

2.

The residuals of the model suggested above were denoted r_t. We have plotted r_t, acf and pacf of r_t^2 below:







From the plot above the non constant variance assumption seems to be satisfied. The ACF and PACF indicates that the r_t^2 can be modeled by an ARMA model of unknown orders. Hence we can only try a simple GARCH(1, 1).

3.

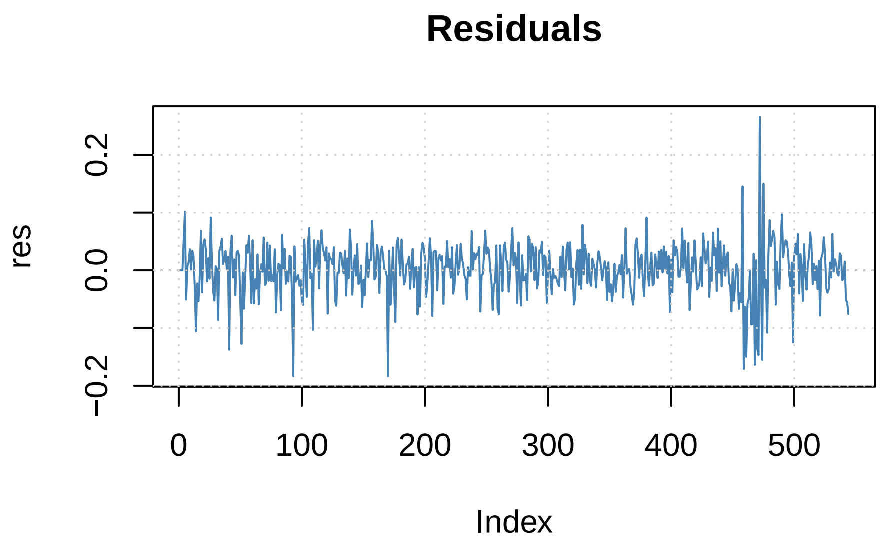
Using the information from the previous steps, we used garchfit to fit an $arma(0,3) + garch(1,1)$. For that, we have been able to see that higher coefficeints of ARMA (e.g. ma2, ma3 and the intercept coefficients) were not significan so we refitted the model taking this into account. Also, we noticed in the diagnostic plots and the tests that GARCH(2,0) was less accurate than GARCH(1,0). Thus, we took this model. The following tests and diagnostic can be seen below:

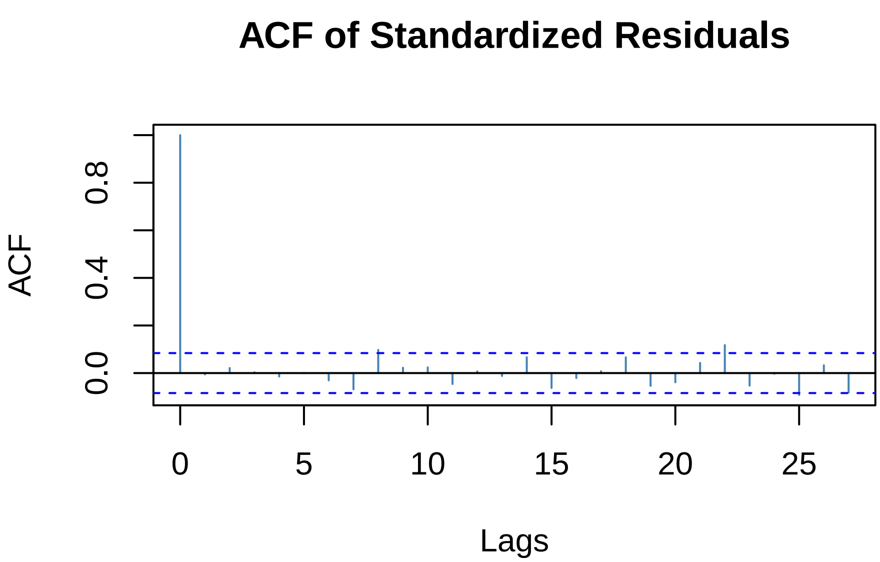
##

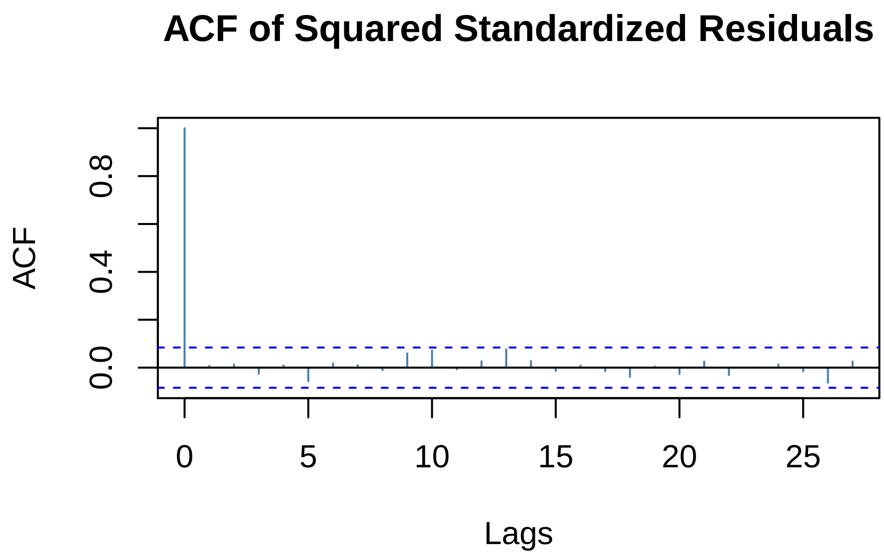
```
## Series Initialization:
##  ARMA Model:                arma
##  Formula Mean:              ~ arma(0, 3)
##  GARCH Model:               garch
##  Formula Variance:          ~ garch(1, 1)
##  ARMA Order:                0 3
##  Max ARMA Order:            3
##  GARCH Order:               1 1
##  Max GARCH Order:           1
##  Maximum Order:             3
##  Conditional Dist:          norm
##  h.start:                   4
##  llh.start:                 1
##  Length of Series:          544
##  Recursion Init:            mci
##  Series Scale:              0.04700153
##
## Parameter Initialization:
##  Initial Parameters:          $params
##  Limits of Transformations:   $U, $V
##  Which Parameters are Fixed?  $includes
##  Parameter Matrix:
##                      U           V     params includes
##     mu     -0.37951444   0.3795144  0.0000000    FALSE
##     ma1    -0.99999999   1.0000000  0.1696068     TRUE
##     ma2    -0.99999999   1.0000000 -0.0886270     TRUE
##     ma3    -0.99999999   1.0000000  0.1458111     TRUE
##     omega   0.00000100 100.0000000  0.1000000     TRUE
##     alpha1  0.00000001   1.0000000  0.1000000     TRUE
##     gamma1 -0.99999999   1.0000000  0.1000000    FALSE
##     beta1   0.00000001   1.0000000  0.8000000     TRUE
##     delta   0.00000000   2.0000000  2.0000000    FALSE
##     skew    0.10000000  10.0000000  1.0000000    FALSE
##     shape   1.00000000  10.0000000  4.0000000    FALSE
##  Index List of Parameters to be Optimized:
##    ma1    ma2    ma3  omega alpha1  beta1
##      2      3      4      5      6      8
##  Persistence:                  0.9
##
##
## --- START OF TRACE ---
## Selected Algorithm: nlminb
##
## R coded nlminb Solver:
##
##   0:     729.10402: 0.169607 -0.0886270 0.145811 0.100000 0.100000 0.800000
##   1:     728.67380: 0.170050 -0.0878899 0.143117 0.0965786 0.0950129 0.797735
##   2:     728.33108: 0.170816 -0.0867410 0.138725 0.0994908 0.0926803 0.801554
##   3:     728.13575: 0.172506 -0.0847079 0.130198 0.0947854 0.0828417 0.801106
##   4:     726.83970: 0.178489 -0.0811180 0.108034 0.0958890 0.0795052 0.816696
##   5:     726.19270: 0.186194 -0.0819615 0.0897748 0.0791091 0.0811601 0.827537
##   6:     725.66140: 0.191431 -0.0902620 0.0871735 0.0721350 0.0695638 0.850101
##   7:     725.53213: 0.183647 -0.0883820 0.0869710 0.0551502 0.0621064 0.869774
##   8:     725.40778: 0.181367 -0.0868716 0.0853227 0.0518029 0.0660425 0.878597
```

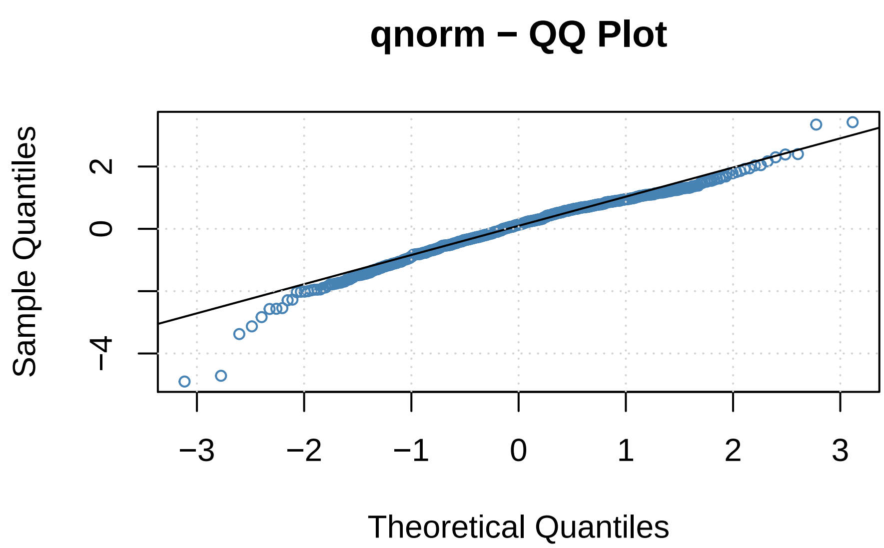
```
##   9:     725.22500: 0.189495 -0.0889897 0.0802181 0.0527852 0.0652672 0.874527
##  10:     725.22082: 0.189182 -0.0880324 0.0765452 0.0548325 0.0623685 0.876594
##  11:     725.18898: 0.188778 -0.0878984 0.0756723 0.0523601 0.0615072 0.876810
##  12:     725.15288: 0.188865 -0.0881895 0.0746384 0.0513626 0.0620952 0.879121
##  13:     725.14327: 0.189417 -0.0889152 0.0734723 0.0495989 0.0620227 0.880713
##  14:     725.13698: 0.190946 -0.0908704 0.0723892 0.0502071 0.0623878 0.880643
##  15:     725.12705: 0.191254 -0.0909577 0.0699667 0.0507222 0.0622632 0.879381
##  16:     725.12365: 0.190889 -0.0887096 0.0693746 0.0501908 0.0619009 0.880757
##  17:     725.12009: 0.192742 -0.0893413 0.0701001 0.0491982 0.0610538 0.882091
##  18:     725.11705: 0.192403 -0.0918660 0.0690477 0.0493046 0.0611150 0.882572
##  19:     725.11470: 0.191333 -0.0931703 0.0669760 0.0492203 0.0614771 0.881819
##  20:     725.11256: 0.193577 -0.0918252 0.0666468 0.0496460 0.0620044 0.881162
##  21:     725.10984: 0.194168 -0.0911172 0.0664536 0.0489680 0.0603913 0.883136
##  22:     725.10959: 0.194172 -0.0911197 0.0664274 0.0489966 0.0604715 0.883184
##  23:     725.10940: 0.194178 -0.0911239 0.0663857 0.0489156 0.0605044 0.883155
##  24:     725.10912: 0.194149 -0.0911312 0.0662406 0.0488889 0.0606045 0.883247
##  25:     725.10861: 0.194020 -0.0912504 0.0658904 0.0487808 0.0605978 0.883244
##  26:     725.10803: 0.194301 -0.0916958 0.0653575 0.0489453 0.0607480 0.883022
##  27:     725.10795: 0.193928 -0.0911566 0.0647105 0.0481176 0.0607203 0.884074
##  28:     725.10785: 0.193934 -0.0911634 0.0647045 0.0480902 0.0606913 0.884038
##  29:     725.10777: 0.193996 -0.0912244 0.0646495 0.0481917 0.0606762 0.884004
##  30:     725.10757: 0.194238 -0.0911964 0.0645275 0.0481908 0.0607262 0.883895
##  31:     725.10739: 0.194444 -0.0914936 0.0642869 0.0484514 0.0605757 0.883619
##  32:     725.10739: 0.195606 -0.0916805 0.0642347 0.0485043 0.0607301 0.883536
##  33:     725.10711: 0.195142 -0.0919391 0.0632141 0.0486613 0.0609233 0.883231
##  34:     725.10705: 0.194998 -0.0923376 0.0631520 0.0484886 0.0608423 0.883376
##  35:     725.10696: 0.194723 -0.0921617 0.0634407 0.0486082 0.0609164 0.883198
##  36:     725.10696: 0.194674 -0.0922562 0.0634375 0.0485972 0.0609153 0.883256
##  37:     725.10696: 0.194687 -0.0921410 0.0634483 0.0486148 0.0609074 0.883226
##  38:     725.10696: 0.194665 -0.0921752 0.0634482 0.0486004 0.0608996 0.883241
##  39:     725.10696: 0.194663 -0.0921924 0.0634463 0.0486010 0.0609027 0.883243
##  40:     725.10696: 0.194666 -0.0921898 0.0634462 0.0486020 0.0609035 0.883241
##
## Final Estimate of the Negative LLH:
##  LLH:  -938.214    norm LLH:  -1.724658
##           ma1           ma2           ma3         omega        alpha1
##  0.1946659900 -0.0921898180  0.0634462459  0.0001073689  0.0609035480
##         beta1
##  0.8832408335
##
## R-optimhess Difference Approximated Hessian Matrix:
##                 ma1          ma2         ma3         omega        alpha1
## ma1     -528.745239    112.51607   -58.25077 -7.162622e+03  2.163670e+01
## ma2      112.516069   -514.70261   166.42207 -1.147808e+04  1.504832e+01
## ma3      -58.250771    166.42207  -516.16837  9.129824e+04 -3.845038e+01
## omega  -7162.621530 -11478.08030 91298.24320 -6.837863e+09 -9.051769e+06
## alpha1    21.636705     15.04832   -38.45038 -9.051769e+06 -1.839067e+04
## beta1      3.839136    -50.69586   177.17067 -1.184300e+07 -1.781258e+04
##               beta1
## ma1     3.839136e+00
## ma2    -5.069586e+01
## ma3     1.771707e+02
## omega  -1.184300e+07
## alpha1 -1.781258e+04
```

```
## beta1  -2.206838e+04
## attr(,"time")
## Time difference of 0.1090009 secs
##
## --- END OF TRACE ---
##
##
## Time to Estimate Parameters:
##  Time difference of 0.4560208 secs
```










```
##
## Title:
##  GARCH Modelling
##
## Call:
##  garchFit(formula = ~arma(0, 3) + garch(1, 1), data = xt, include.mean = FALSE)
##
## Mean and Variance Equation:
##  data ~ arma(0, 3) + garch(1, 1)
## <environment: 0x000000001cc34ef0>
##  [data = xt]
##
## Conditional Distribution:
```

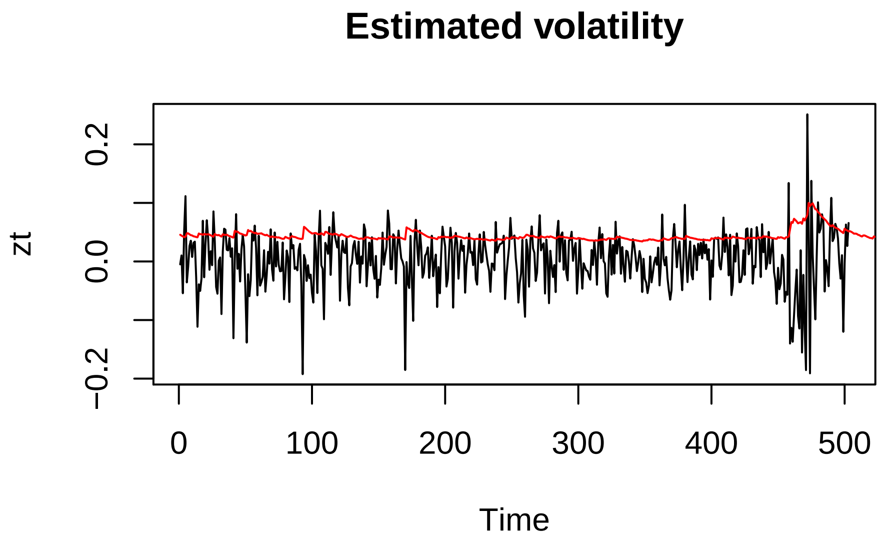
```
##  norm
##
## Coefficient(s):
##         ma1          ma2          ma3        omega       alpha1
##  0.19466599  -0.09218982   0.06344625   0.00010737   0.06090355
##       beta1
##  0.88324083
##
## Std. Errors:
##  based on Hessian
##
## Error Analysis:
##         Estimate  Std. Error  t value Pr(>|t|)
## ma1     1.947e-01   4.460e-02    4.365 1.27e-05 ***
## ma2    -9.219e-02   4.747e-02   -1.942 0.052129 .
## ma3     6.345e-02   4.701e-02    1.350 0.177107
## omega   1.074e-04   4.901e-05    2.191 0.028469 *
## alpha1  6.090e-02   1.712e-02    3.557 0.000376 ***
## beta1   8.832e-01   3.462e-02   25.511  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
##  938.214    normalized:  1.724658
##
## Description:
##  Fri Oct 20 18:15:19 2017 by user: Joshua
##
##
## Standardised Residuals Tests:
##                                 Statistic p-Value
##  Jarque-Bera Test   R    Chi^2  125.1921  0
##  Shapiro-Wilk Test  R    W      0.9716941 9.476336e-09
##  Ljung-Box Test     R    Q(10)  9.3356    0.5005775
##  Ljung-Box Test     R    Q(15)  15.29704  0.4302401
##  Ljung-Box Test     R    Q(20)  20.56608  0.4230561
##  Ljung-Box Test     R^2  Q(10)  7.420724  0.685218
##  Ljung-Box Test     R^2  Q(15)  11.62151  0.7074219
##  Ljung-Box Test     R^2  Q(20)  13.05673  0.8749369
##  LM Arch Test       R    TR^2   7.524992  0.8210629
##
## Information Criterion Statistics:
##       AIC       BIC       SIC      HQIC
## -3.427257 -3.379842 -3.427497 -3.408719
```

The model seems appropriate seeing as the ACF of the squared residuals seems to show some autocorrelation , the QQ shows does not show utterly normal. The Jarque Bera test has as null hypothesis that the residuals are normally ditributed. Since we get a p-value of 0, we can reject this hypothesis. Also, the Ljung box test shows that our residuals are dependent. For that, GARCH is good to be used.

$$\sigma_t^2 = -0.0921898 + 0.0634462 r_{t-1}^2$$

###4.

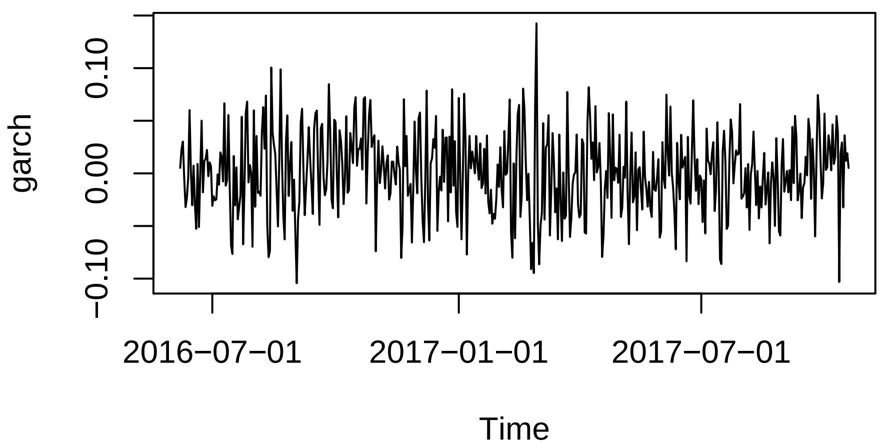
We have computed the volatility pattern alongside with z_t and plotted it below:



We cans see that it follows the variation of data increasing visibly during the biggest peaks.

5.

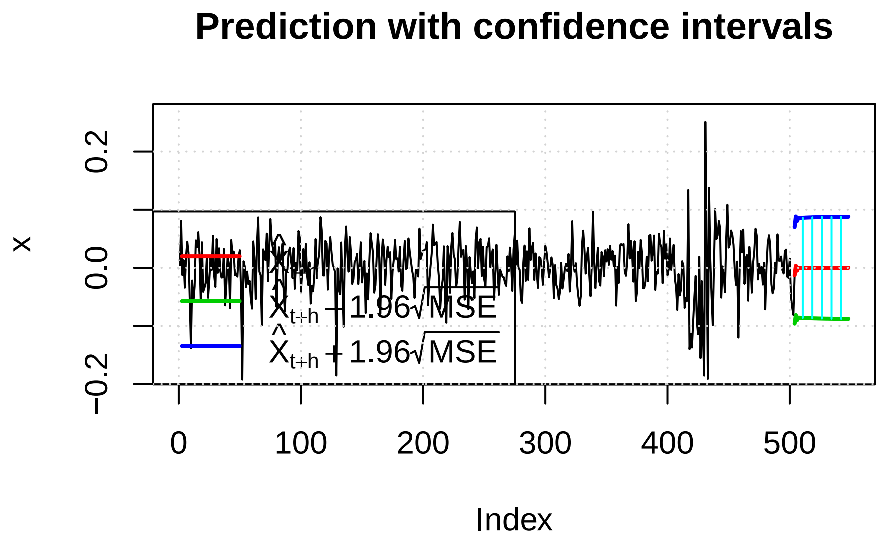
We have simulated 500 observations from the model in step 3. The resulting plots can be seen below:

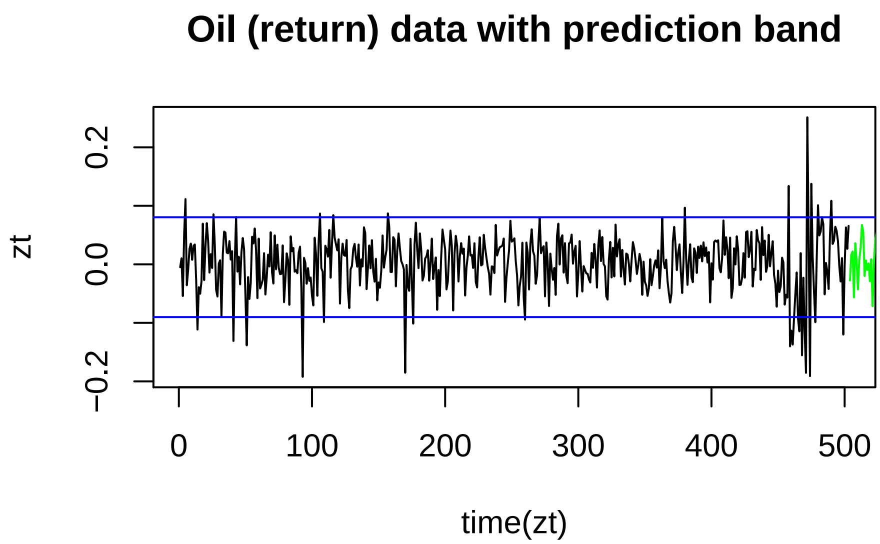


Comparing it with the plot of z_t, the patters seems to be similar, having similar variance alongside time.

6.

Finally, we computed and plotted a 45 step ahead prediction ising `predict()` as shown below. We also plotted the full data set (x_t) and overlaid the prediction band found in the forecasting on top.





We can see that the prediction bands captures all the true data.

Appendix

```
knitr::opts_chunk$set(echo = FALSE, message = FALSE, warning=FALSE, fig.width = 5, fig.asp = 0.66, fig.a
library(astsa)
data(chicken)
xt <- chicken
plot(xt, main = "Chicken Time Series")

chicken_df <- data.frame(t=1:length(xt), xt=xt)
linmod <- lm(xt~t, data=chicken_df)
zt <- as.ts(xt-linmod$fitted.values)
plot(zt, main="Detrended Chicken Data")

T <- length(zt)
n <- T
nprime <- n/2-1
dftR <- as.vector(fft(zt))
dft <- dftR/sqrt(n)/exp(complex(imaginary=2*pi*(1:n-1)/n))
I <- Mod(dft)^2
freq <- (1:n-1)/n
plot(freq, I, type = "o")

I0 <- 100
#95% CI
U <- 2*max(I)/qchisq(0.025, 2)
L <- 2*max(I)/qchisq(0.975, 2)
c(L, U)

#set negligeable frequencies to 0
dft[which(I<I0)] <- 0

#inv DFT implementation
m <- 36 #include prediction
ztfit <- vector(length=T+m)
for (t in 1:T+m) {
  ztfit[t] <- as.numeric(1/sqrt(n)*sum(dft*exp(complex(imaginary=(2*pi*(1:n-1)*t/n)))))
}

trend <- predict(linmod, newdata = data.frame(t=1:(T+m)))
xtfit <- trend+ztfit

tpred <- 1:(T+m)
plot(tpred, xtfit, type="l", col="red")
lines(1:T, xt)

k <- kernel("modified.daniell", c(2, 2))
spect_est <- mvspec(zt, kernel = k, log="no", plot=FALSE)

#CI
w<- spect_est$freq
w <- w/(2*max(w))
fw <- spect_est$spec
Lh <- spect_est$Lh
```

```r
U <- 2*Lh*fw/qchisq(0.025, df=2*Lh)
L <- 2*Lh*fw/qchisq(0.975, df=2*Lh)

plot(x=w, y=fw, type="l")
lines(x=w, y=U, col="blue")
lines(x=w, y=L, col="blue")

w0 <- 0.1 #?
par(mar=c(1,1,1,1))
sarima21 <- sarima(xt, 2, 1, 0, 0, 0, 1, 12, details=FALSE)

sarima21_pred <- sarima.for(xt, n.ahead = m,  2, 1, 0, 0, 0, 1, 12)$pred

plot(1:T, xt, xlim=c(0, T+m))
lines((T+1):(T+m),sarima21_pred, col="blue")
lines(1:(T+m), xtfit, col="red")

par(mar=c(1,1,1,1))
sarima30 <- sarima(zt, 3, 0, 0, 0, 0, 1, 12, details=FALSE)

coefs <- sarima30$ttable[, 1]
arma.spec(ar=coefs[1:3], ma = c(rep(0, 11), coefs[4]), log="no")
data(oil)
xt <- diff(log(oil))
zt <- as.ts(xt[1:(9*52+2+33)])
plot(zt)

TSA::eacf(zt)
p <- 0
q <- 3
library(tseries)
par(mar=c(1,1,1,1))
sarima(zt, p, 0, q, details=FALSE)
ma3 <- arma(zt, order = c(p, q))
rt <- ma3$residuals[4:length(ma3$residuals)]
plot(rt, type ="l")
acf(rt^2)
pacf(rt^2)



library(fGarch)
#tried arma03+grach20 showed mean/ma2/ma3 not significant
#arma01+grach10 showed better AIC and test results than arma01+garch20
garch10 <- garchFit(~arma(0, 3)+garch(1, 1), data = xt, include.mean = FALSE)
#diagnostics
plot(garch10, which=c(7, 10, 11, 13))
#tests+significance
summary(garch10)
#coeffs
coefs <- garch10@fit$coef
v <- volatility(garch10, type = "sigma")
plot(zt, main="Estimated volatility")
```

```r
lines(v, col="red")

modspec <- garchSpec(model = list(omega=coefs[4], alpha=coefs[5], beta=coefs[6], ma=coefs[1:3]))
sim <- garchSim(modspec, n=500)

plot(sim, type="l")
pred <- predict(garch10, n.ahead=45, plot=T, nx=length(zt))
U <- pred$upperInterval[3]
L <- pred$lowerInterval[3]
plot(time(zt), zt, type="l", main="Oil (return) data with prediction band")
lines(504:544, xt[504:544], col = "green")
abline(h=U, col="blue")
abline(h=L, col="blue")
```