# 732A62 Lab 3

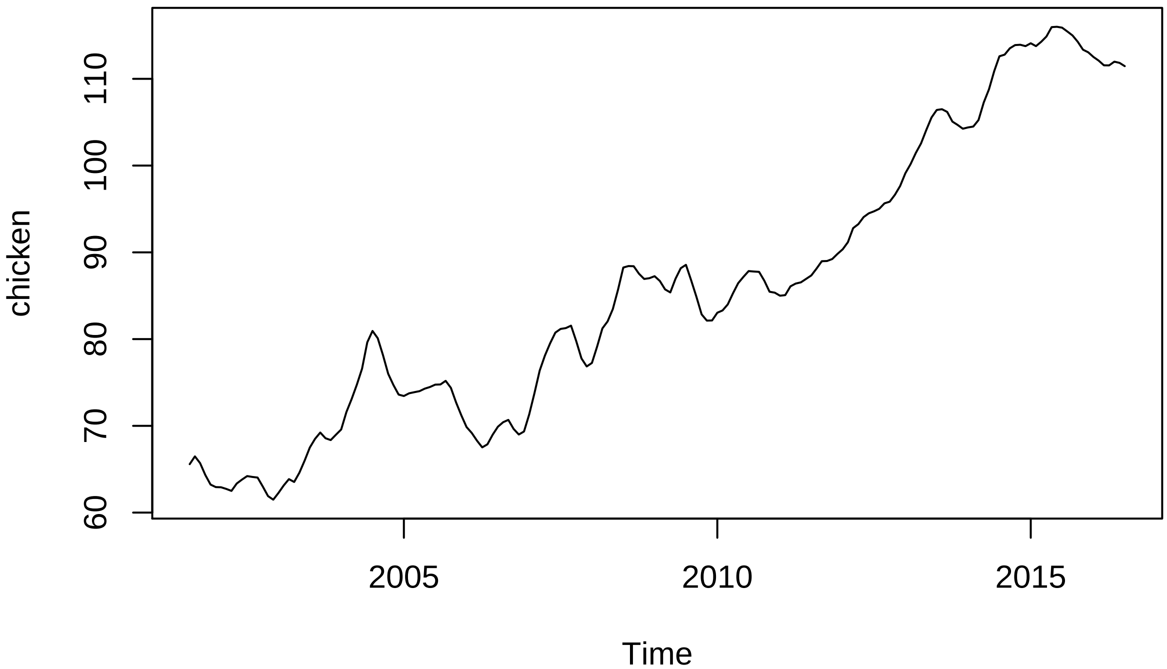
*Emil K Svensson & Rasmus Holm*

*2017-10-11*

## Assignment 1

### 1)

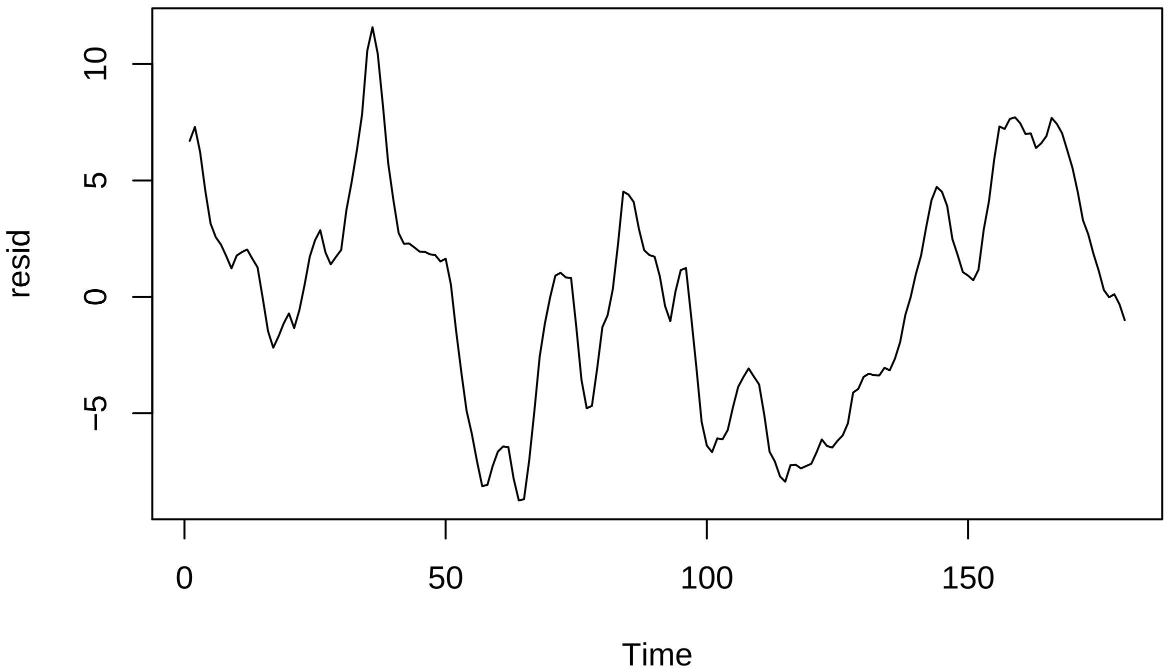
```r
plot(chicken)
```




It looks like a linear, potentially quadratic, trend.

### 2)

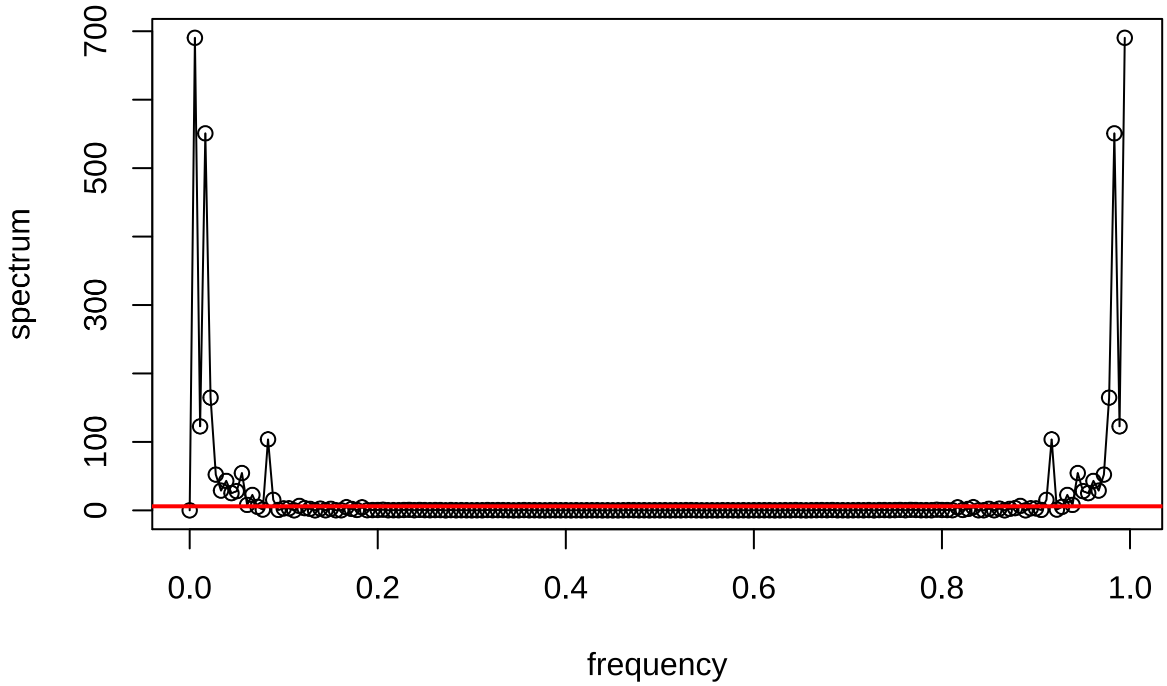
```r
lm_data <- data.frame(chicken=chicken, time=1:length(chicken))
lm_fit <- lm(chicken ~ time, lm_data)
z <- resid(lm_fit)
plot(z, type="l", ylab="resid", xlab="Time")
```




The residuals do not look stationary. The data is definitely correlated.

### 3)

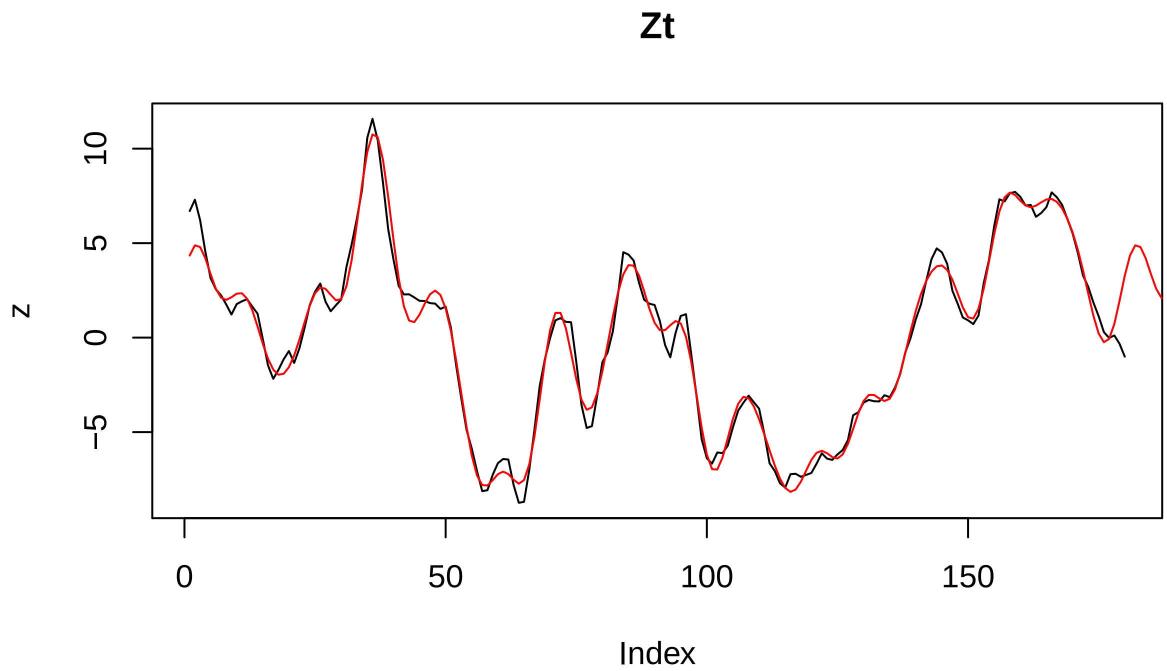
```
denom <- sqrt(length(z)) *
    exp(complex(imaginary=2 * pi * 0:(length(z) - 1) / length(z)))
density <- fft(z) / denom
periodigram <- abs(density)^2

upper <- 2 * mean(periodigram) / qchisq(0.025, 2)
lower <- 2 * mean(periodigram) / qchisq(0.975, 2)

plot(0:(length(chicken) - 1) / length(chicken), periodigram, type="o",
     xlab="frequency", ylab="spectrum")
abline(h=lower, col="red", lwd=2)
```

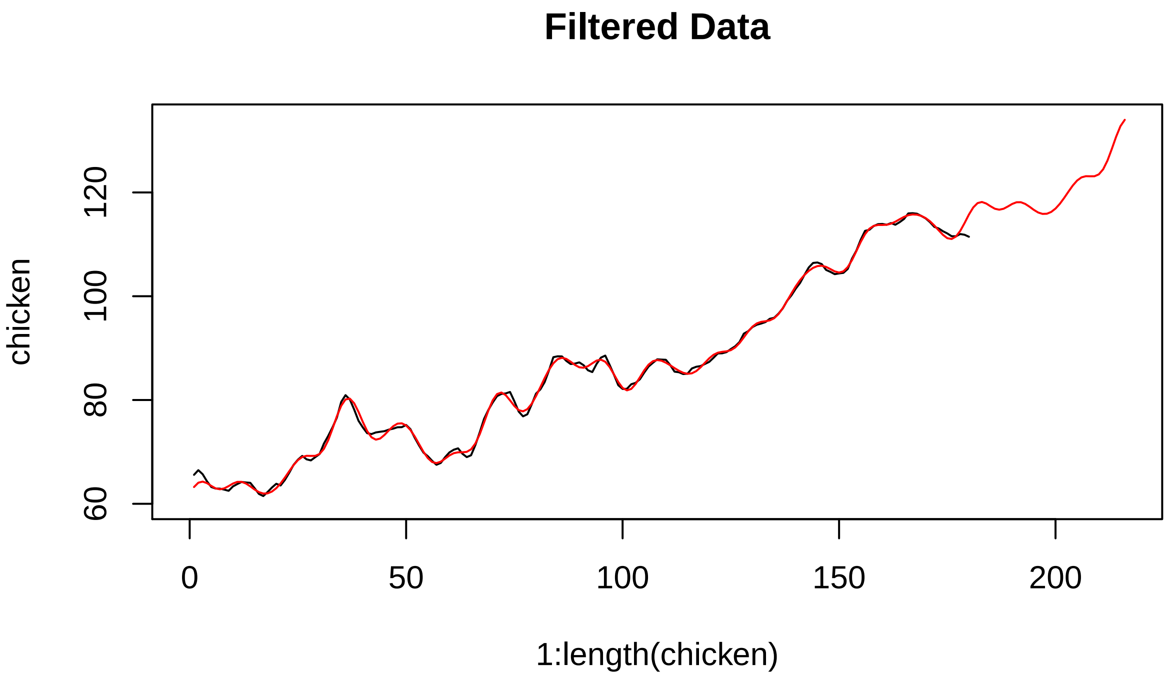



We can see that low and high frequencies are the dominant frequencies. We decided to use the mean as the baseline which sets the lower limit close to zero. This results in that most non-zero frequencies are significant.

## 4)

```
freq_density <- density
freq_density[periodigram < lower] <- 0

n <- length(z)
ts <- 1:(n + 36)

xs <- rep(0, n + 36)

for (t in ts) {
    xs[t] <- sum(freq_density * exp(complex(imaginary=2 * pi * (0:(n - 1)) / n * t))) / sqrt(n)
}

filtered_data <- predict(lm_fit, data.frame(time=1:length(xs))) + Re(xs)
```

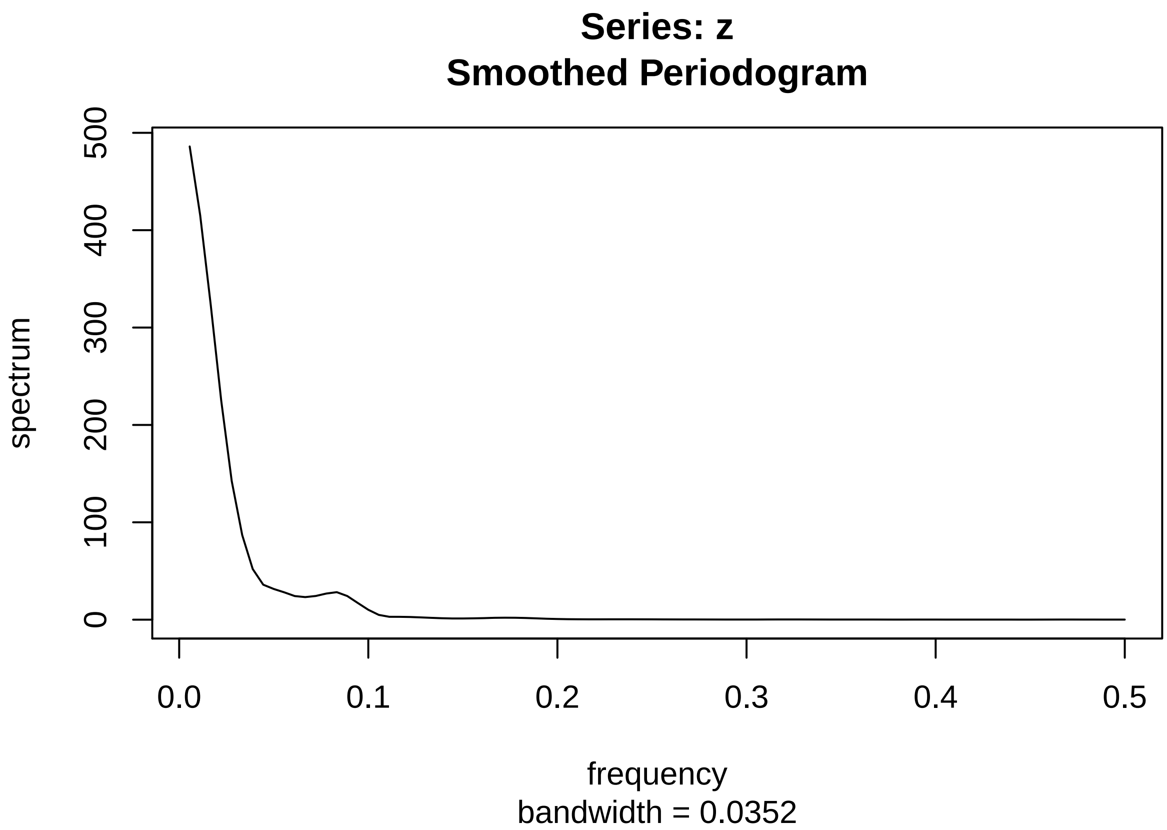



The forecast do look reasonable since it follows the general trend well.

## 5)

```r
k <- kernel("modified.daniell", c(2,2))
md_dan <- mvspec(z, kernel=k, log="no")
```



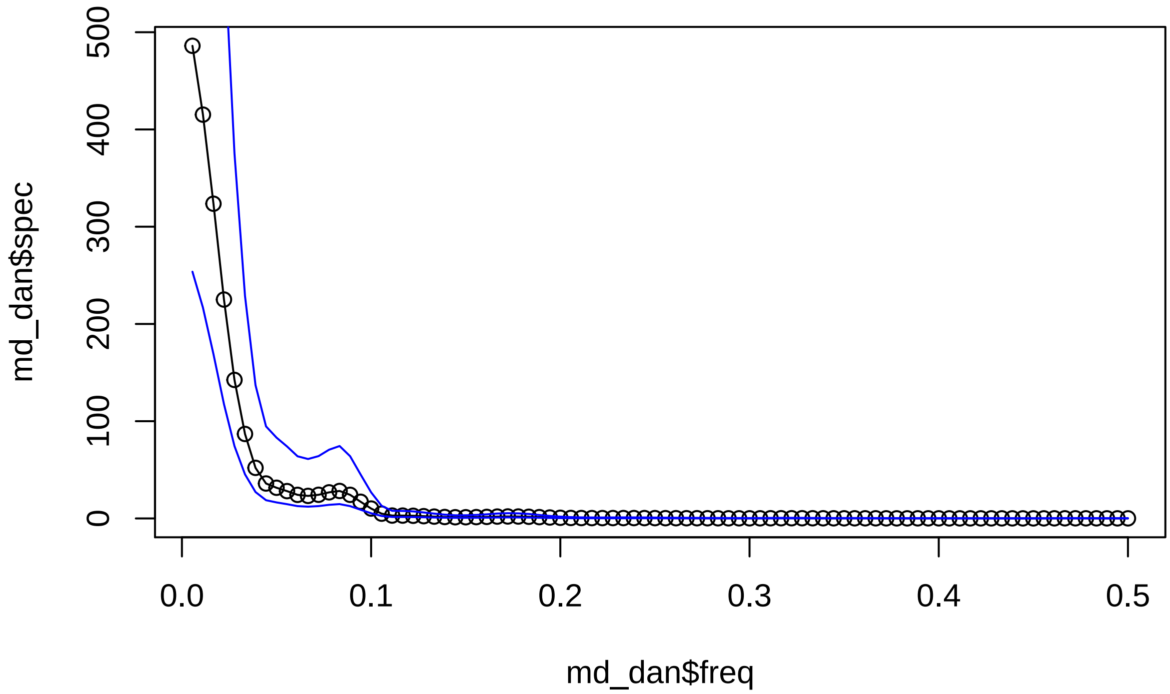

```r
Lh <- md_dan$Lh

lower1 <- 2 * Lh * md_dan$spec / qchisq(0.975,2*Lh)
upper1 <- 2 * Lh * md_dan$spec / qchisq(0.025,2*Lh)

# Comparing frequencies

freq_4 <- 0:179/180

freq_4[periodigram > lower]
```

```
##  [1] 0.005555556 0.011111111 0.016666667 0.022222222 0.027777778
##  [6] 0.033333333 0.038888889 0.044444444 0.050000000 0.055555556
## [11] 0.061111111 0.066666667 0.083333333 0.088888889 0.116666667
## [16] 0.883333333 0.911111111 0.916666667 0.933333333 0.938888889
## [21] 0.944444444 0.950000000 0.955555556 0.961111111 0.966666667
## [26] 0.972222222 0.977777778 0.983333333 0.988888889 0.994444444
```

```r
md_dan$freq[md_dan$freq < 0.1]
```

```
##  [1] 0.005555556 0.011111111 0.016666667 0.022222222 0.027777778
##  [6] 0.033333333 0.038888889 0.044444444 0.050000000 0.055555556
## [11] 0.061111111 0.066666667 0.072222222 0.077777778 0.083333333
## [16] 0.088888889 0.094444444 0.100000000
```

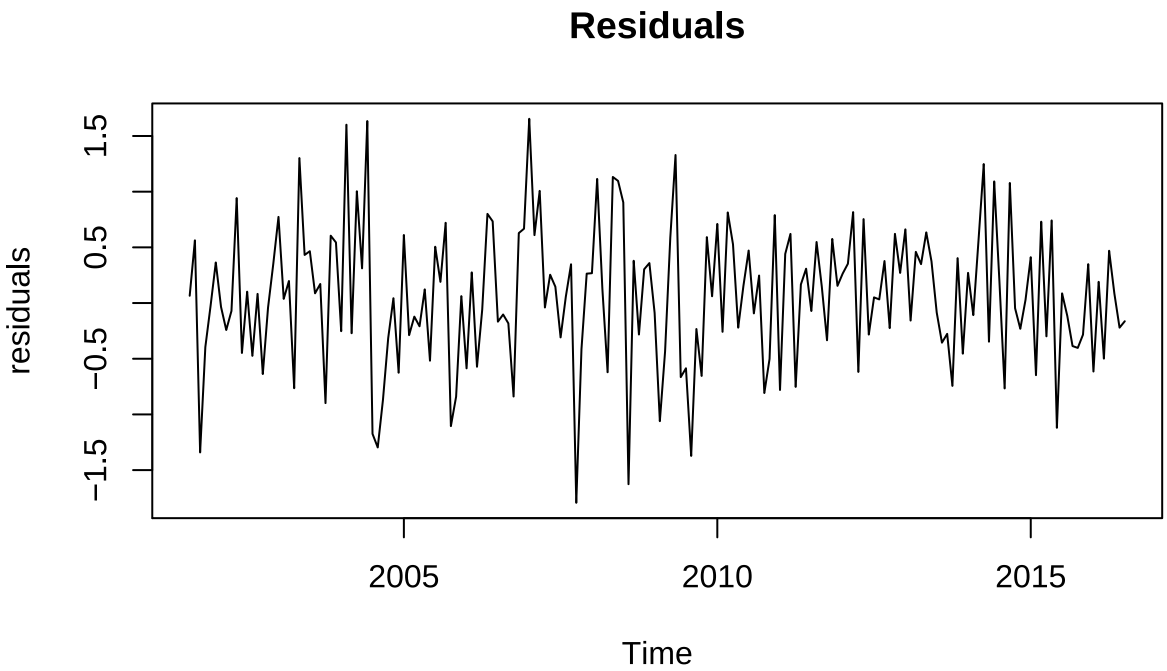
We can see that similar frequencies were found by smoothing the spectrum so the smoothing does seem to help.

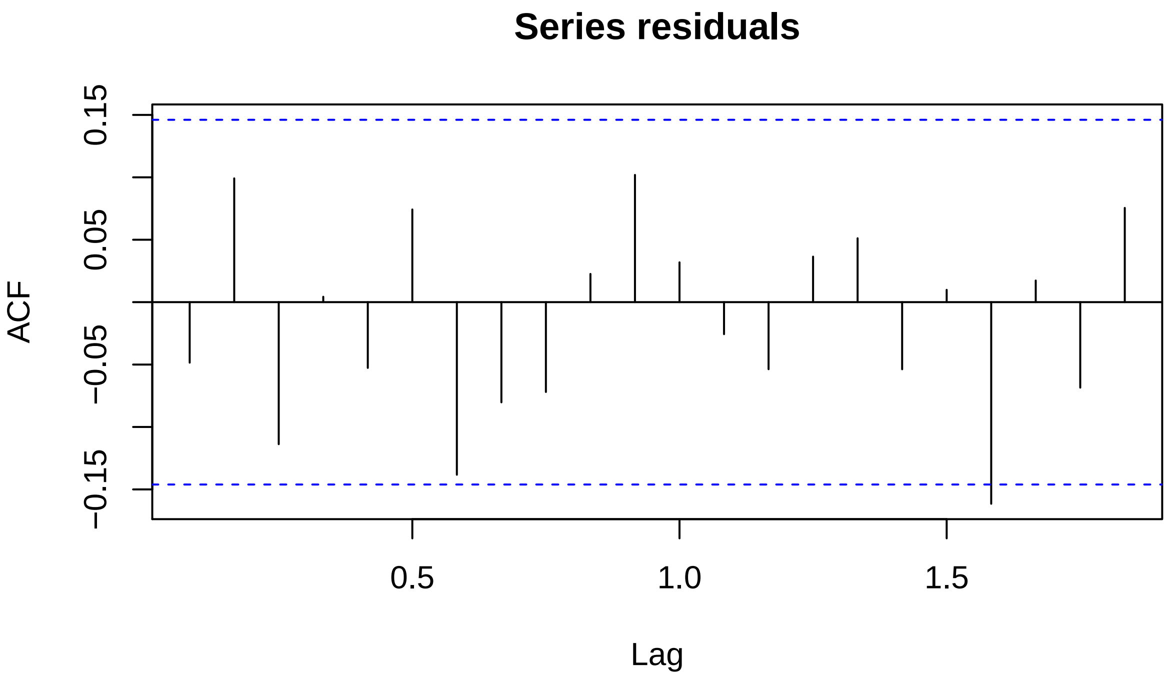



**6)**

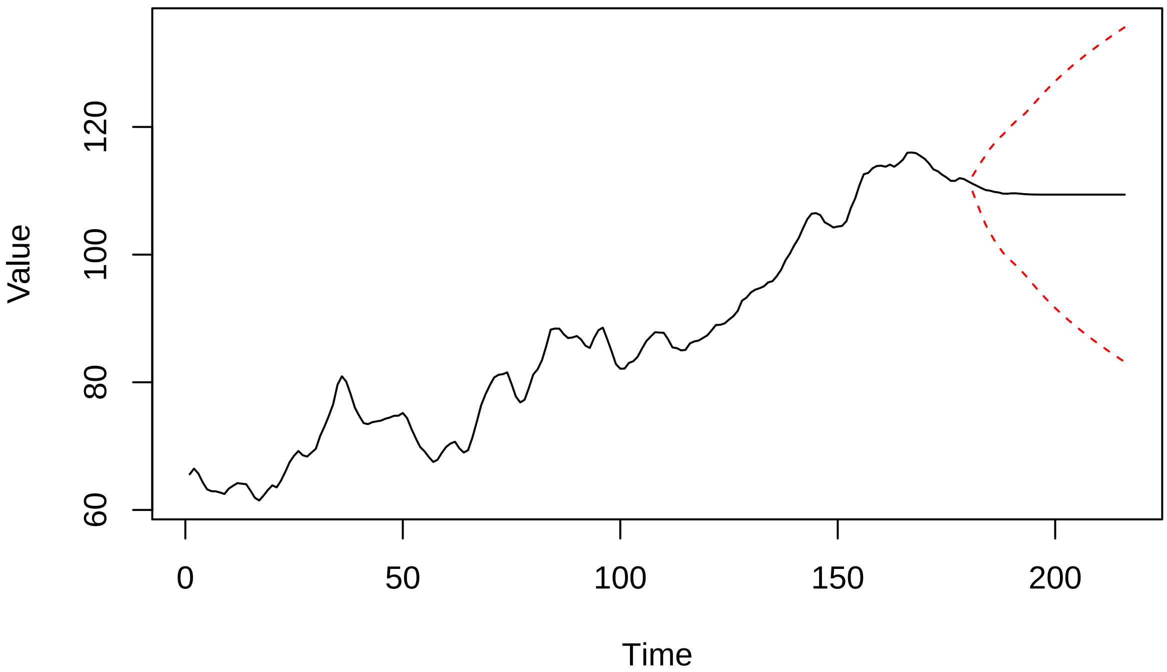
```
fit_plot <- function(model, data) {
    nahead <- 36
    pred <- predict(model, n.ahead=nahead, se.fit=TRUE)
    upper_band <- pred$pred + 1.96 * pred$se
    lower_band <- pred$pred - 1.96 * pred$se

    n <- length(data)

    plot(c(data, pred$pred), type="l",
         ylim=c(min(data), max(upper_band)), ylab="Value", xlab="Time")
    lines(n + 1:nahead, upper_band, lty=2, col="red")
    lines(n + 1:nahead, lower_band, lty=2, col="red")
}
```

```
fit <- arima(chicken, order=c(2, 1, 0), seasonal=list(order=c(0, 0, 1), period=12))
residuals <- residuals(fit)
```



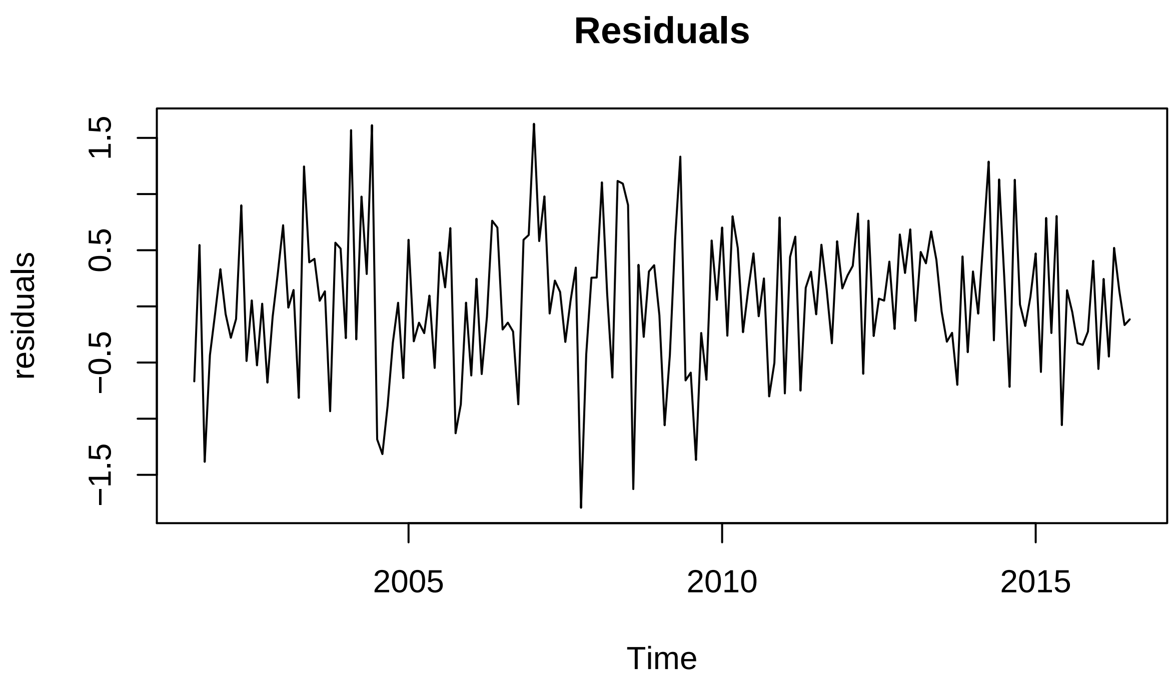

The model seem to fit the data decent with no correlation. However, the variance seem to decrease with time so it may not be completely stationary.

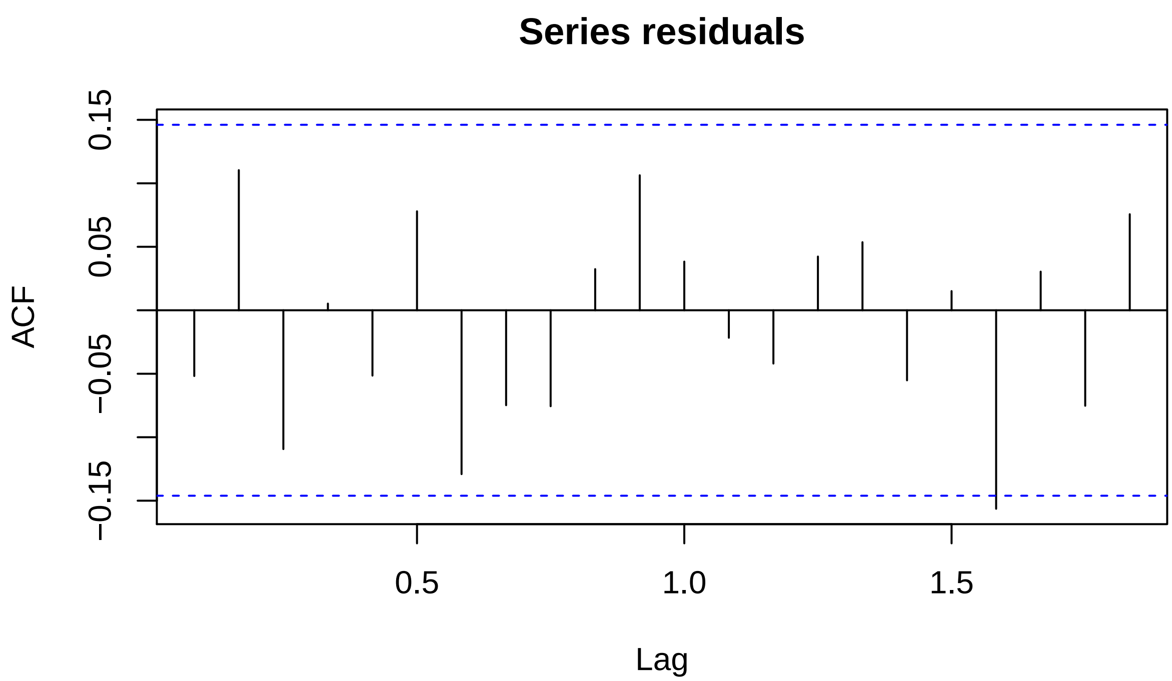



The forecast do not look very good because it does not follow the general trend. We would rather trust the forecast from 1.4.

### 7)

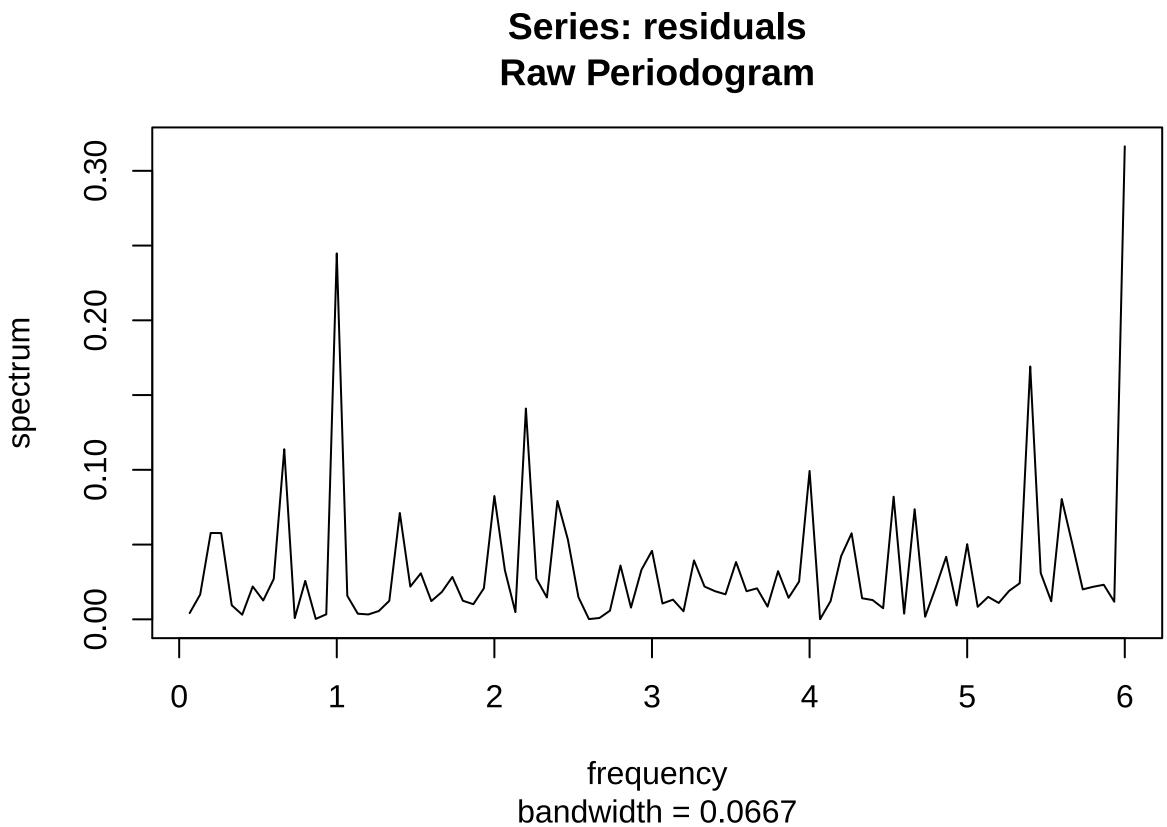
```
fit <- arima(chicken, order=c(3, 0, 0), seasonal=list(order=c(0, 0, 1), period=12))
residuals <- residuals(fit)
```




The residuals looks similar to those from the other fit. Uncorrelated but not stationary because of changing variance.

```
mvspec(residuals, log="no")
```

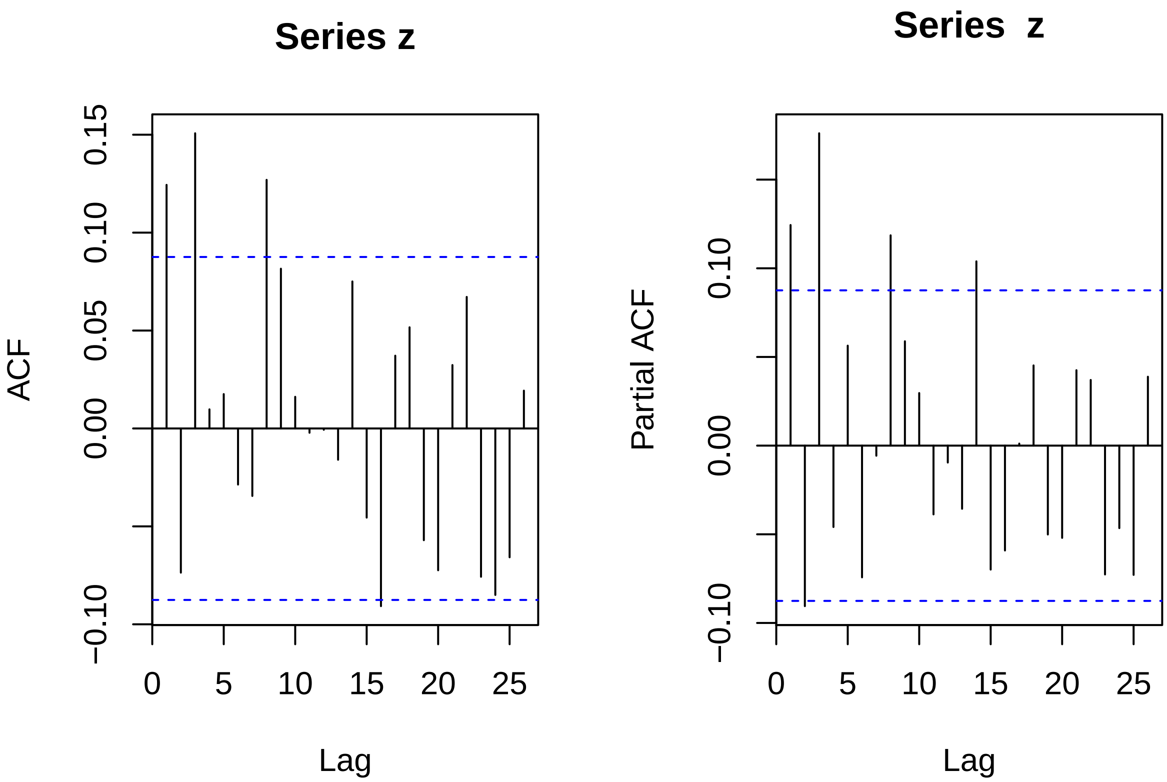



We can see that the spectrum is non-zero for a lot of frequencies and not just low ones. This indicates that the residuals are not stationary.

# Assignment 2

## 1)

```
ld_oil <-diff(log(oil))

z <-ld_oil[1:(52*9 + 33)]

old <- par(mfrow = c(1,2))
acf(z)
pacf(z)
```




```
par(old)

suggested_model <- Arima(z, order = c(3,0,0))

summary(suggested_model)
```

```
## Series: z
## ARIMA(3,0,0) with non-zero mean
##
## Coefficients:
##          ar1      ar2     ar3    mean
##        0.151  -0.1147  0.1777  0.0018
## s.e.   0.044   0.0442  0.0442  0.0026
##
## sigma^2 estimated as 0.002171:  log likelihood=827.28
## AIC=-1644.55   AICc=-1644.43   BIC=-1623.47
##
```

```
## Training set error measures:
##                        ME       RMSE        MAE  MPE MAPE      MASE
## Training set 2.381642e-05 0.04640656 0.03454024 -Inf  Inf 0.7492286
##                     ACF1
## Training set 0.008324494
```

```r
suggested_model$coef +  1.96 * sqrt(diag(suggested_model$var.coef))
```

```
##         ar1          ar2          ar3    intercept
##  0.237236987 -0.028108501  0.264357426  0.006936839
```

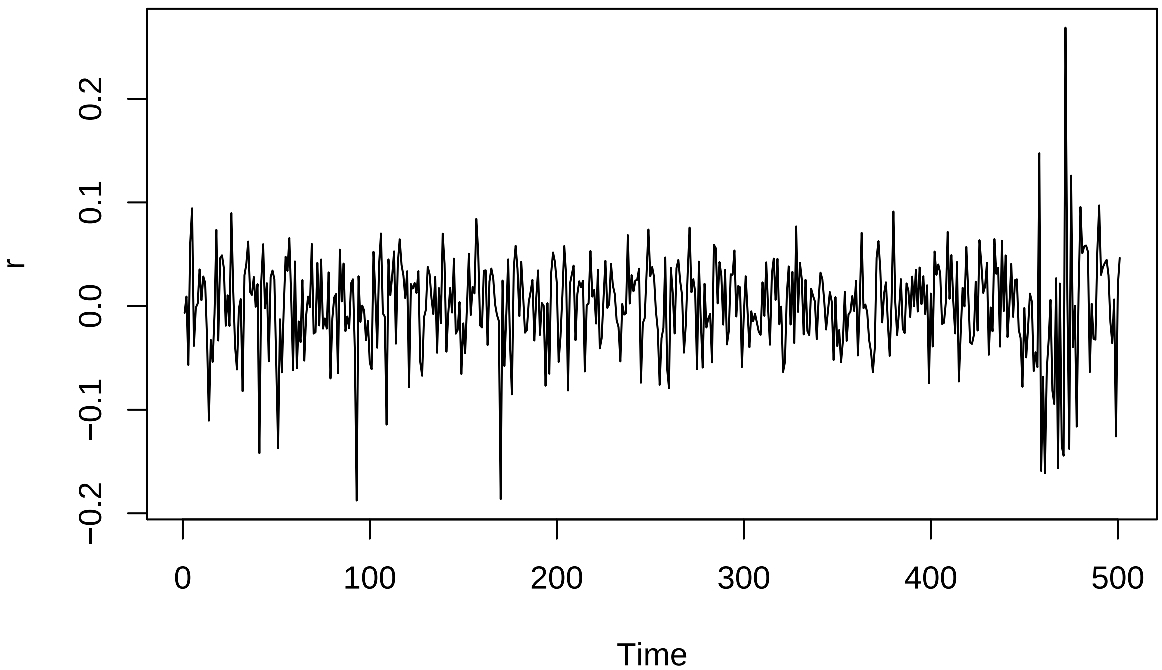
```r
suggested_model$coef -  1.96 * sqrt(diag(suggested_model$var.coef))
```

```
##         ar1          ar2          ar3    intercept
##  0.064829350 -0.201337050  0.090958540 -0.003392617
```

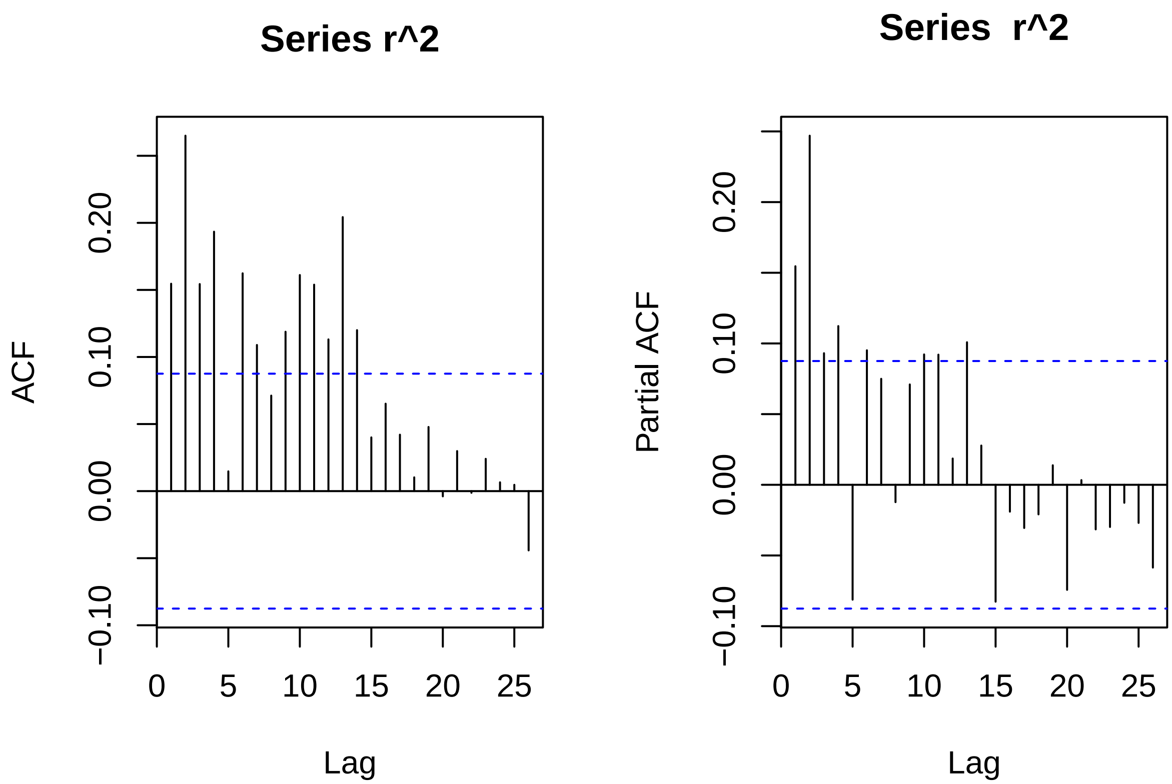
```r
r <- resid(suggested_model)
```

The ACF is dying down All coefficients of the AR3 model is significant so we move forward with it. ## 2)

```r
plot(r)
```




```r
old <- par(mfrow = c(1,2))
acf(r^2)
pacf(r^2)
```




```r
par(old)


fit1<- garchFit(~ arma(3,0) + garch(1,1) , data = ld_oil, trace = FALSE)
fit1
```

```
##
## Title:
##  GARCH Modelling
##
## Call:
##  garchFit(formula = ~arma(3, 0) + garch(1, 1), data = ld_oil,
##     trace = FALSE)
##
## Mean and Variance Equation:
##  data ~ arma(3, 0) + garch(1, 1)
## <environment: 0x7f826869a200>
##  [data = ld_oil]
##
## Conditional Distribution:
##  norm
##
## Coefficient(s):
##          mu          ar1          ar2          ar3        omega
##  0.00239398   0.17510221  -0.12421076   0.07407572   0.00011329
##      alpha1        beta1
##  0.06213760   0.87911978
##
## Std. Errors:
```

```
##  based on Hessian
##
## Error Analysis:
##         Estimate  Std. Error  t value Pr(>|t|)
## mu      0.0023940   0.0017860    1.340 0.180120
## ar1     0.1751022   0.0444931    3.935  8.3e-05 ***
## ar2    -0.1242108   0.0449940   -2.761 0.005769 **
## ar3     0.0740757   0.0457586    1.619 0.105482
## omega   0.0001133   0.0000515    2.200 0.027834 *
## alpha1  0.0621376   0.0173665    3.578 0.000346 ***
## beta1   0.8791198   0.0362308   24.264  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
##  937.9971    normalized:  1.724259
##
## Description:
##  Wed Oct 11 17:16:03 2017 by user:
```

The time series of the residuals seem to have an increasing variance in the end of the residuals.

The ACF of the squared residuals trails of and in the PACF they cuts of after 2 lags. Indicating a GARCH(p,q) An p = 2, q = 0 maybe?

## 3)

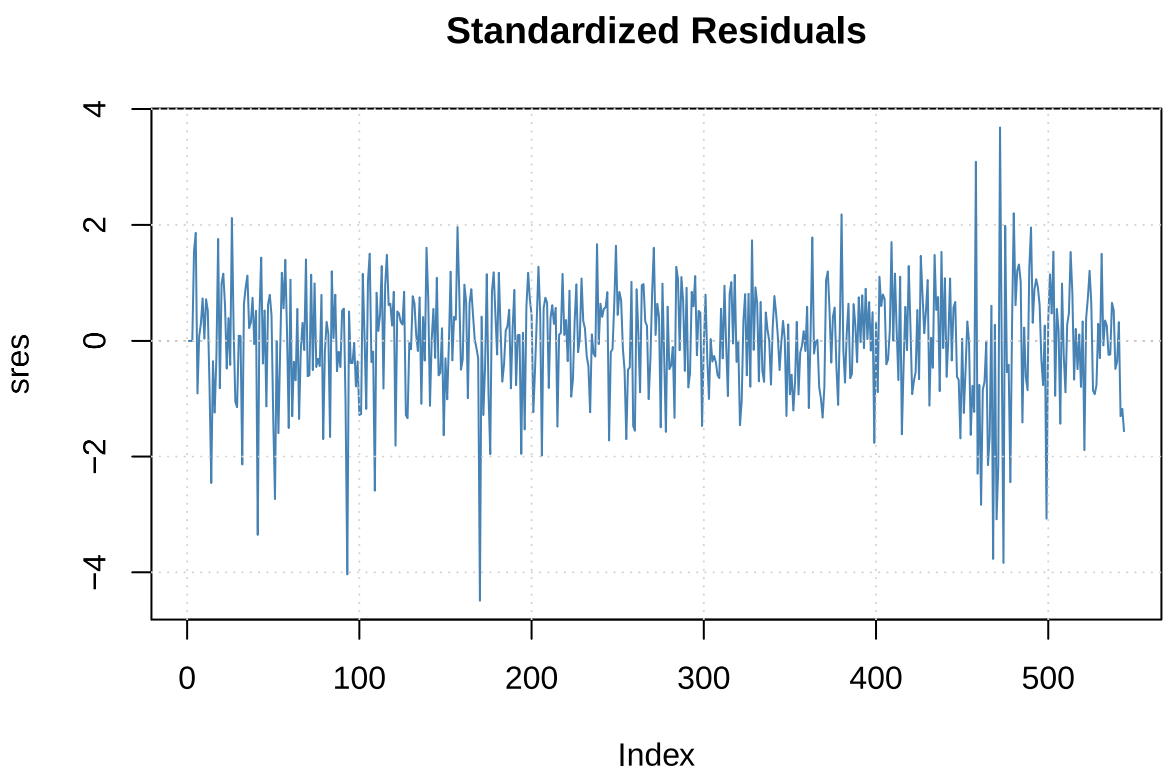
```r
fit1<- garchFit(~ arma(3,0) + garch(1,0) , data = ld_oil, trace = FALSE)
fit1
```

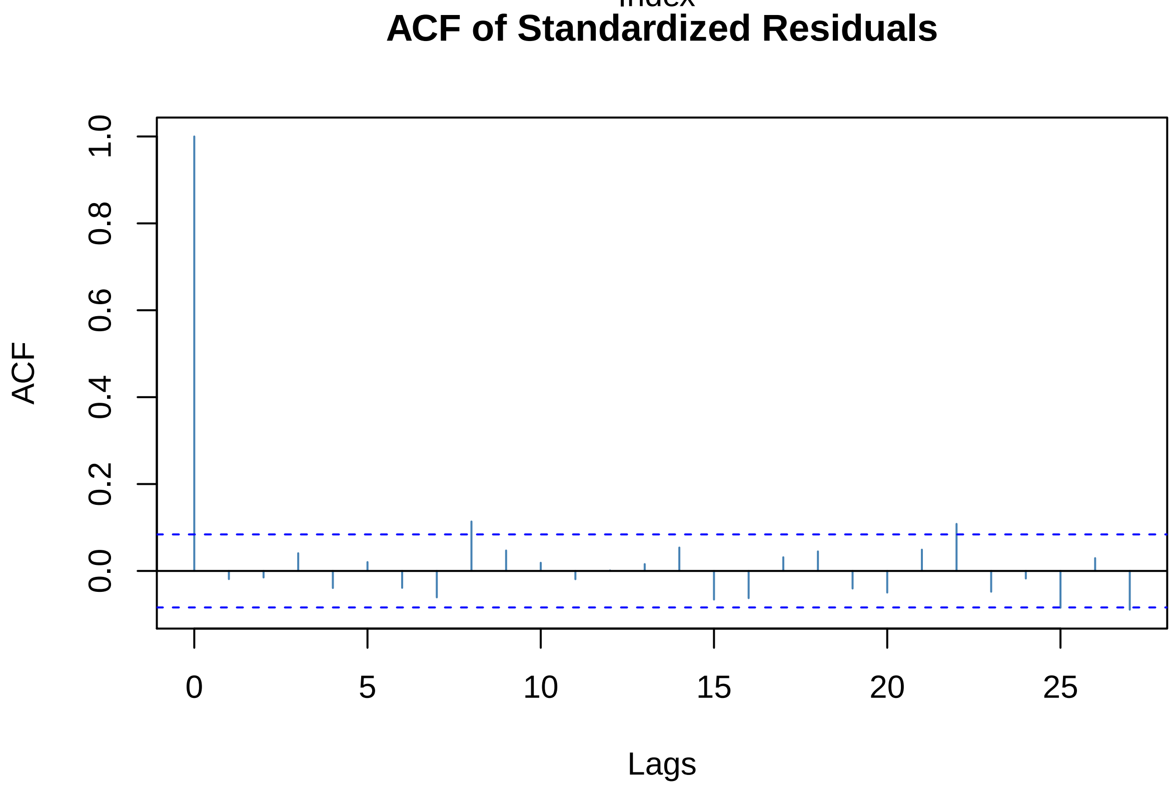
```
##
## Title:
##  GARCH Modelling
##
## Call:
##  garchFit(formula = ~arma(3, 0) + garch(1, 0), data = ld_oil,
##     trace = FALSE)
##
## Mean and Variance Equation:
##  data ~ arma(3, 0) + garch(1, 0)
## <environment: 0x7f82698e40d0>
##  [data = ld_oil]
##
## Conditional Distribution:
##  norm
##
## Coefficient(s):
##         mu         ar1         ar2         ar3       omega      alpha1
##  0.0017864   0.2225997  -0.1021282   0.0944799   0.0016814   0.1863076
##
## Std. Errors:
##  based on Hessian
##
## Error Analysis:
```

```
##         Estimate  Std. Error  t value Pr(>|t|)
## mu      0.0017864   0.0018866    0.947 0.343685
## ar1     0.2225997   0.0647443    3.438 0.000586 ***
## ar2    -0.1021282   0.0414650   -2.463 0.013778 *
## ar3     0.0944799   0.0442595    2.135 0.032787 *
## omega   0.0016814   0.0001308   12.856  < 2e-16 ***
## alpha1  0.1863076   0.0599895    3.106 0.001898 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
##  920.699    normalized:  1.692461
##
## Description:
##  Wed Oct 11 17:16:04 2017 by user:
```

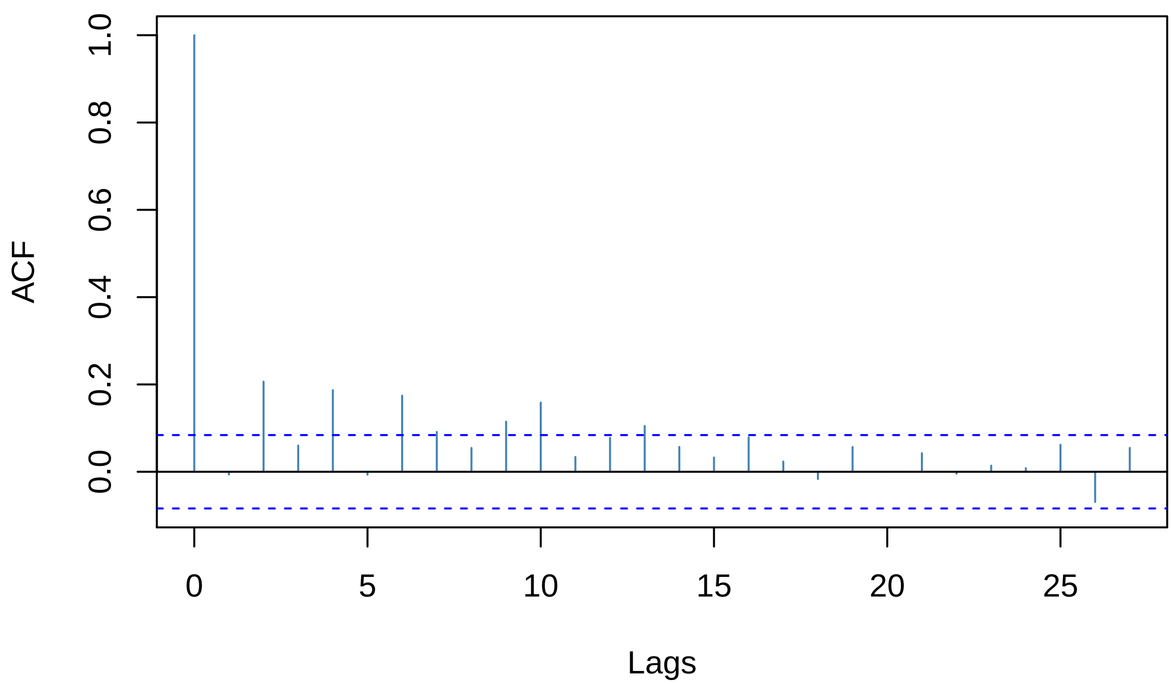
After starting with an ARMA-GARCH(3,0)-(3,0) we itteratively decrease the order of the p' since terms of the GARCH part is unsignificant. We end up with a ARMA-GARCH(3,0)-(1,0) where all parameters are significant.

```r
helper <- function(fit){
  data <- fit@residuals
# acf(scale(data)) # ACF
# acf(data^2)# ACF^2
# qqnorm(data) # QQ-plot
# qqline(data)
print(jarque.bera.test(data)) #Jarque bera-test
print(Box.test(data, lag = 1, type = c( "Ljung-Box"))) # Ljung Box -test
print(fit@fit$matcoef) # Significance.
print(fit@fit$ics) # AIC and Bic
}


plot(fit1, which = c(9,10,11,13))
```

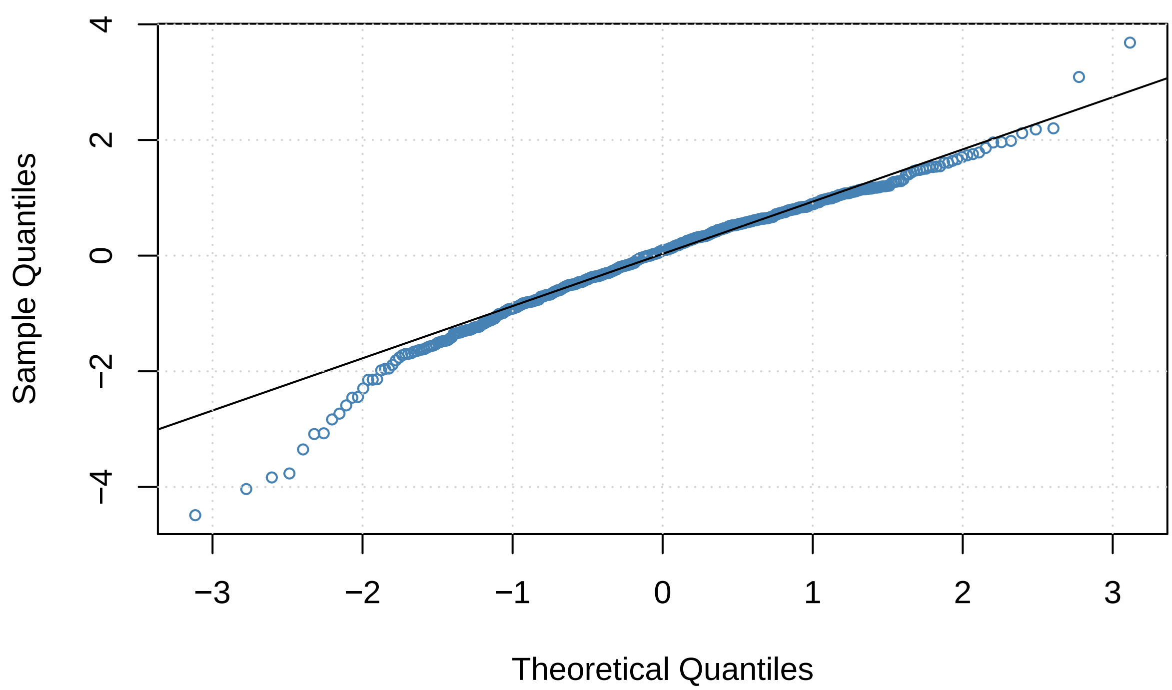

Standardized Residuals
sres
4
2
0
−2
−4
0
100
200
300
400
500
Index


ACF of Standardized Residuals
ACF
1.0
0.8
0.6
0.4
0.2
0.0
0
5
10
15
20
25
Lags

ACF of Squared Standardized Residuals




qnorm − QQ Plot




```r
helper(fit1)
```

```
##
##  Jarque Bera Test
##
```

```
## data:  data
## X-squared = 366.92, df = 2, p-value < 2.2e-16
##
##
##  Box-Ljung test
##
## data:  data
## X-squared = 1.6786, df = 1, p-value = 0.1951
##
##           Estimate  Std. Error    t value     Pr(>|t|)
## mu      0.001786427 0.001886589  0.9469085 0.3436854248
## ar1     0.222599656 0.064744251  3.4381378 0.0005857295
## ar2    -0.102128176 0.041465044 -2.4629945 0.0137782031
## ar3     0.094479933 0.044259488  2.1346820 0.0327869936
## omega   0.001681418 0.000130785 12.8563560 0.0000000000
## alpha1  0.186307604 0.059989494  3.1056705 0.0018984810
##       AIC       BIC       SIC      HQIC
## -3.362864 -3.315449 -3.363104 -3.344326
```

The GARCH(p,q) part of the model looks as follows.

$$\sigma_t^2 = 0.0016814 + 0.1863076 r_{t-1}^2$$

ACF seems to be stationary and the squared residuals dies down after around 10 lags. The QQ plot as a big tail in the left hand side of the graf but hat is the outliers of the end of the time series. Otherwise it looks like it is normaly distributed.

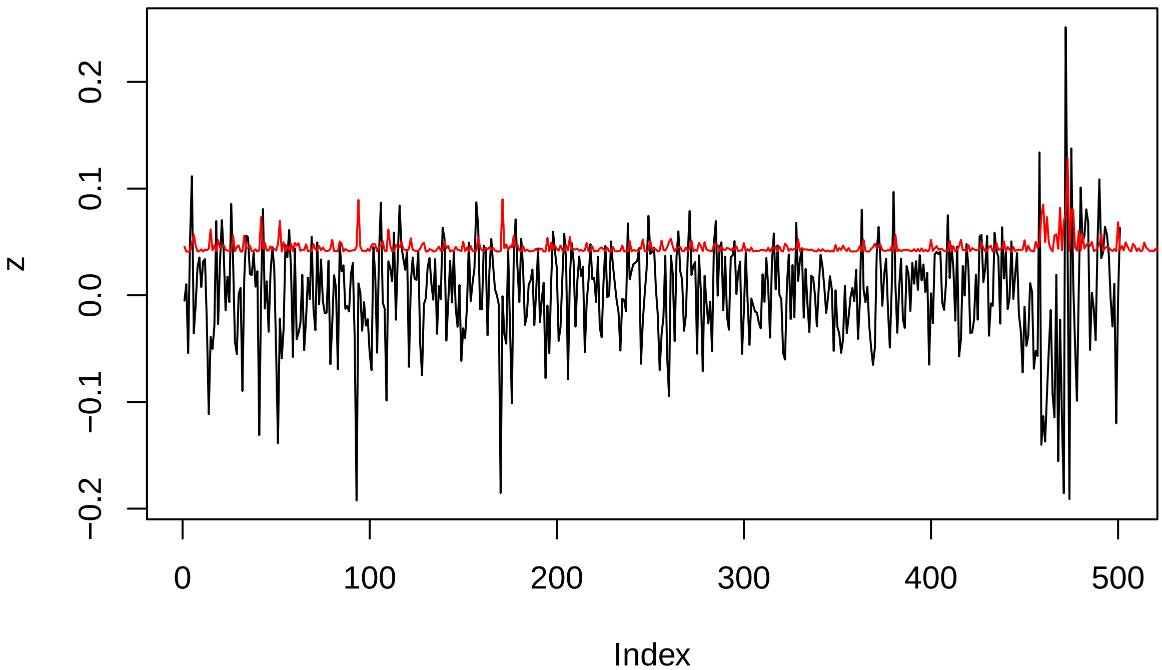
Jarque-Bera test returns a p-value of 0 and therefor we reject the null-hypothesis that the residuals are normaly distributed. This test is argued to have a tendency to give false positives with small observations, Matlab for instance approximate the p-value with an MCMC when the number of observations are below 2000.

The Box-Ljung is non-significant so we conclude the null hypothesis and say that the observations are stationary.

The AIC and BIC are measures only intressting with another model so its hard to say anything about them.

## 4)

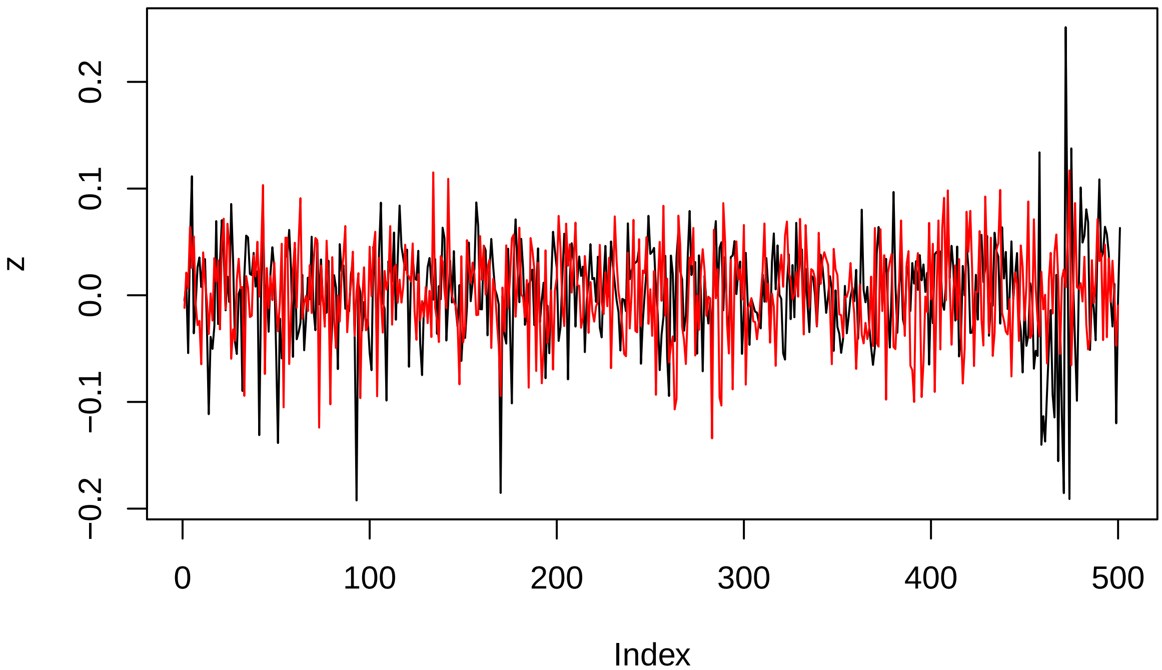
```r
plot(z, type ="l")
lines(volatility(fit1), col = "red")
```




The volatility seems to match the patterns of the observed data very well. Even matching the outliers resonably good.

### 5)

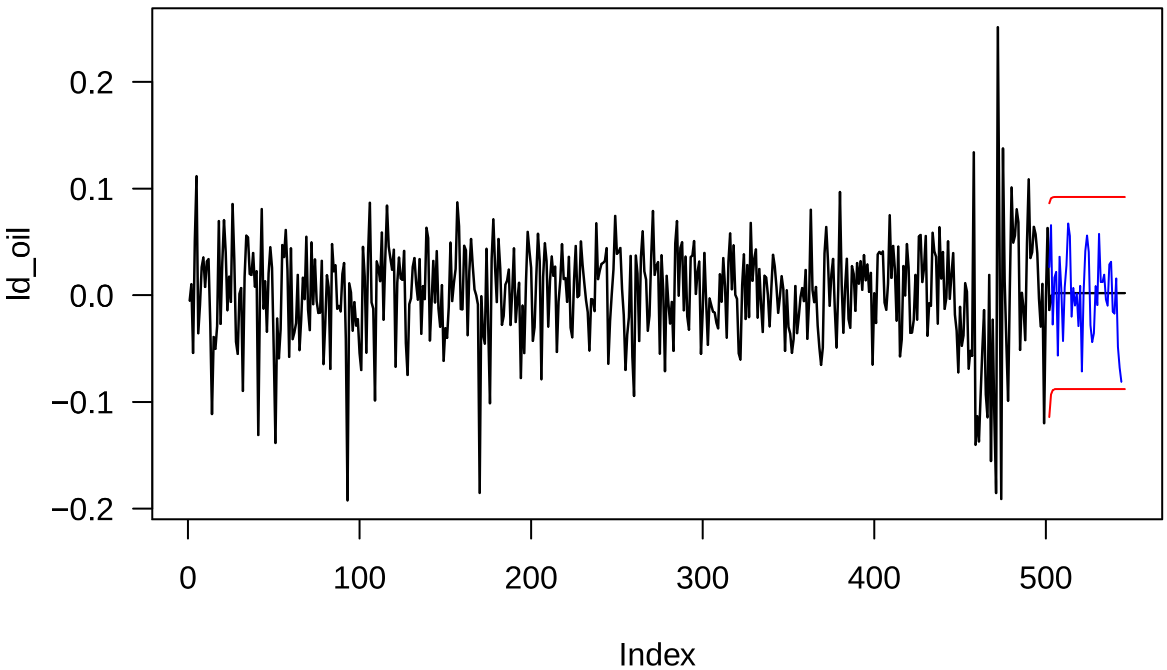
```
omega.fit1 <- 0.0016814
alpha.fit1 <- 0.1863076
sims <- garch.sim(alpha = c(omega.fit1, alpha.fit1) , n = 500)

plot(z, type ="l")
lines(sims,  col = "red")
```




The red simulated data seems to be quite similar to the observed data which is a good sign that we choosen the right model.

### 6)

```
fit2<- garchFit(~ arma(1,0) + garch(1,0) , data = ld_oil, trace = FALSE)
pred.fit1 <- predict(fit2, n.ahead = 45)

lower.fit <- pred.fit1$meanForecast - 1.96*pred.fit1$standardDeviation
upper.fit <- pred.fit1$meanForecast + 1.96*pred.fit1$standardDeviation


x_lim_band<- (length(z)+1) : (length(z)+length(lower.fit))

plot(c(z,pred.fit1$meanForecast), type = "l", col = "black", lwd =1.25, ylab ="ld_oil", las = 1)
lines(x = x_lim_band[1]:length(ld_oil),
      y = ld_oil[x_lim_band[1]:length(ld_oil)],
      col = "blue")
lines(x = x_lim_band, lower.fit, col = "red")
lines(x = x_lim_band,upper.fit, col = "red")
```




Since we have some unexplained error with a ARMA-GARCH(3,0)(1,0) we use a ARMA-GARCH(1,0)(1,0) since the only mention of this error is in on thread in Stackoverflow where the solution was contacting the maintainer of the fGarch-package. The prediction bands contain the observed balues (represented by a blue line). The mean prediction seems to be around 0.