Time Series 732A62

Lab 3

Eric Herwin
erihe068

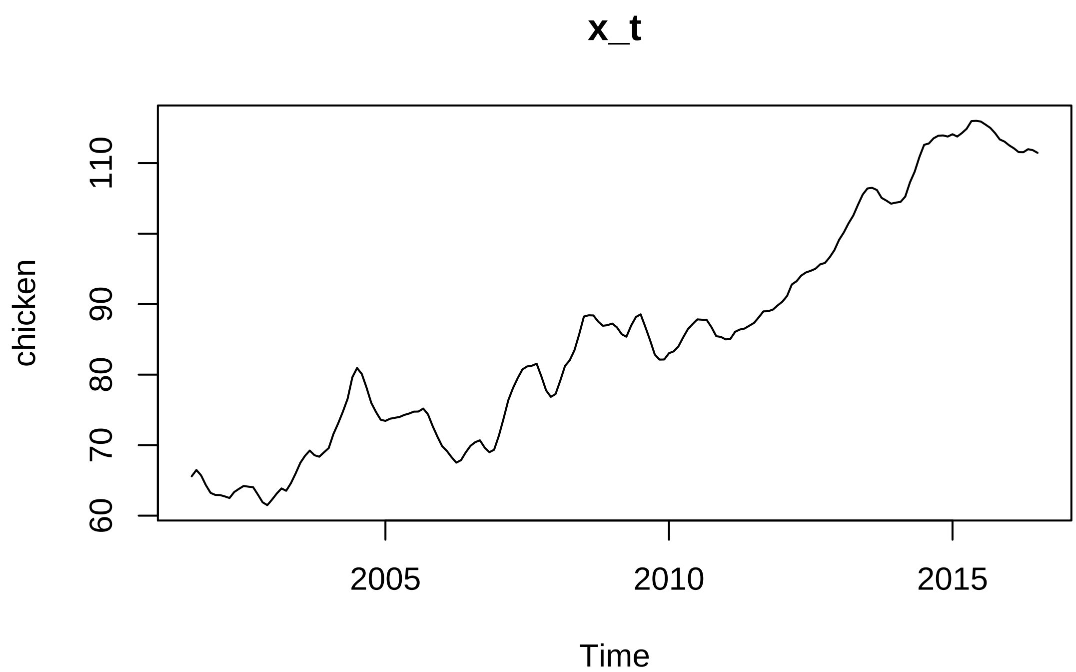
Albin Västerlund
albva223

Contents

Assignment 1. Spectral analysis of the chicken prices

1.1

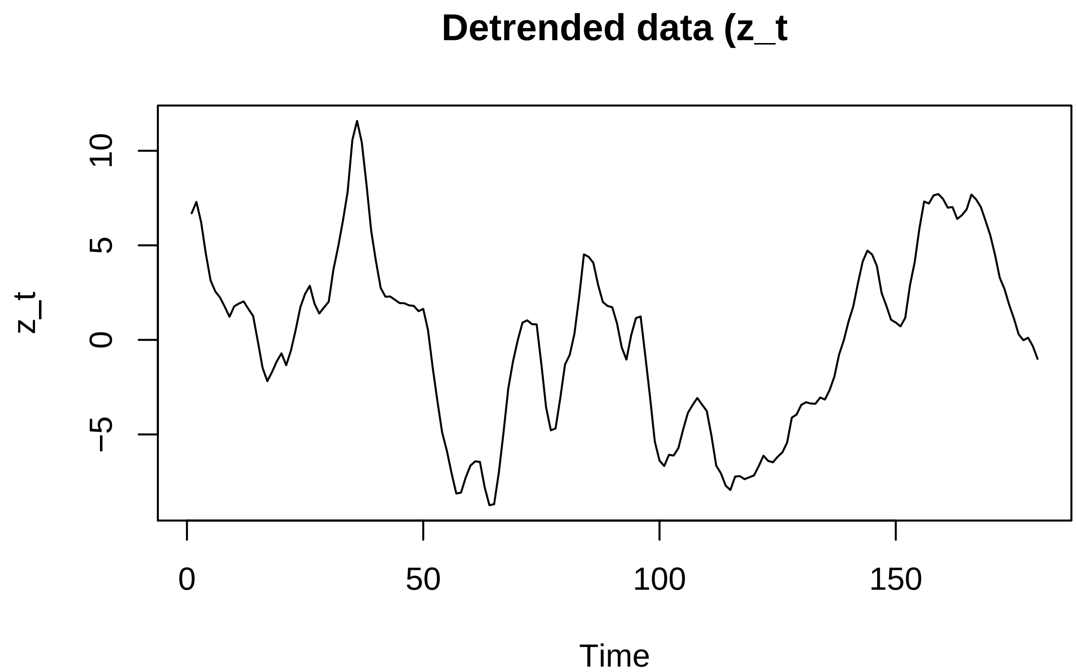
In this assignment we will plot the time series `chicken` for visual inspection. We will denote this time series as x_t.



The data have a clear upper going trend. It probably a linear trend or a exponential trend. We assume that the trend is linear.

1.2

In this assignment we will detrend x_t by applying a linear regression of x_t against the variable t that will denote the time. We will afterwards extract the residuals and observe the detrended series. Our following model-structure will look like this: $x_t = \beta_0 + \beta_1 x_t + \epsilon_t$

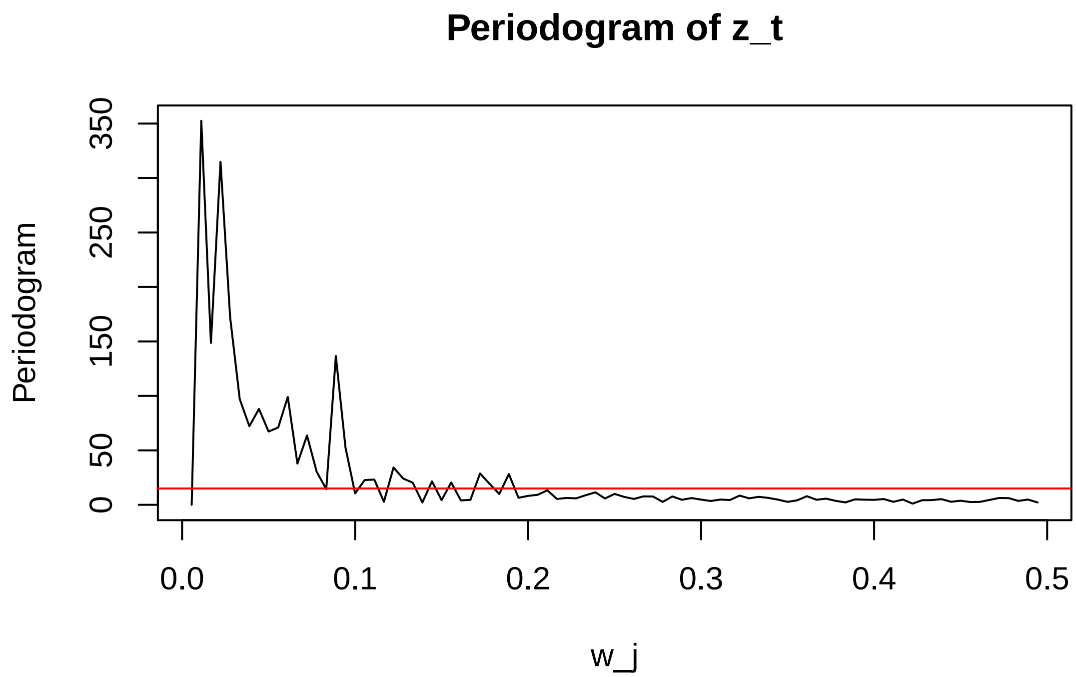


We have now detrended the data, as can be observed in the plot. Observe that the new series is denoted by z_t.

The data goes around 0 and have the same variance over the hole time series, so we assume that the time series is stationary.

1.3

In this assignment we will comute the periodogram of z_t by applying a Fast Fourier transform. The periodogram is plotted.



For the frequencies of the series, z_t, the baseline is the red line drawn in the plot. The all values below the line will be set to 0. This is only visually determined.

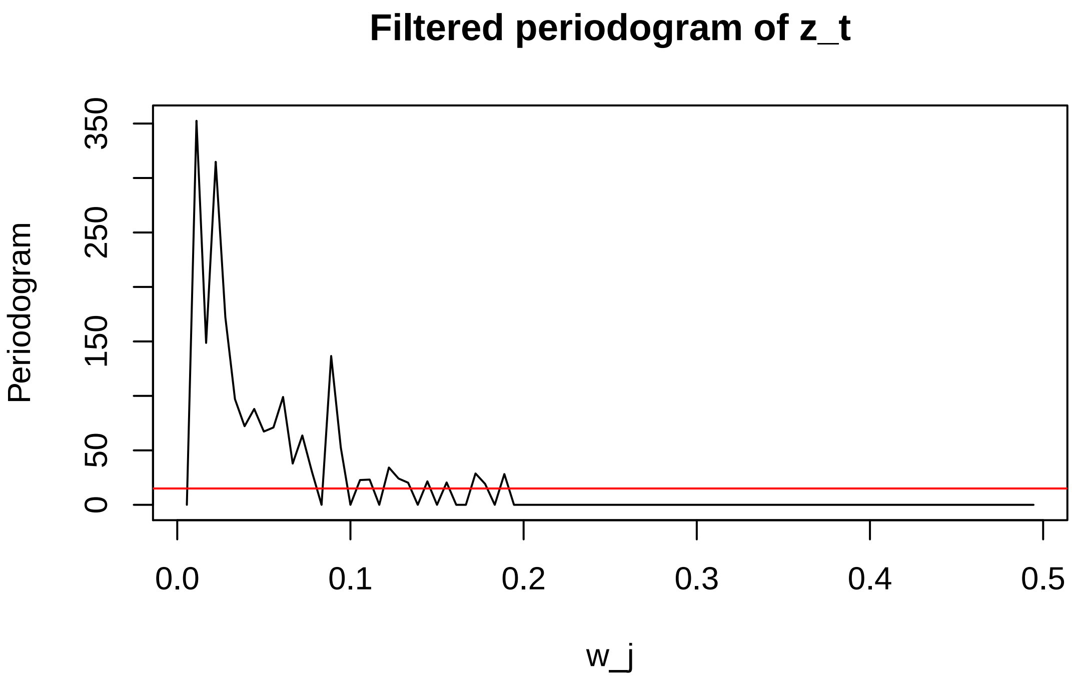
An confidence interval will be computed frequency with the highest amplitude. The interval is computed with values values from the spectral density function of z_t.

```
##       Lower       Upper
##    95.56261 13923.75424
```

We can see that the lower band of the CI does not cover the value set for I_0 and we can conclude that the frequency is significant and dominant for the highest amplitude.

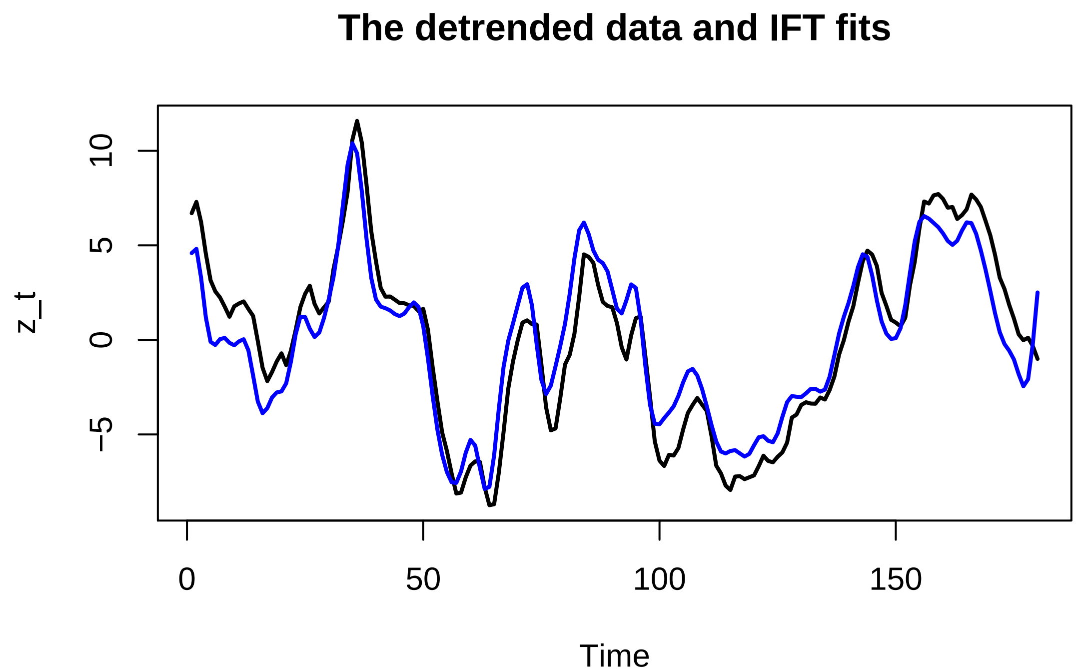
1.4

In this task we will use the same frequencies as step 3 and implement an Inverse Fourier transform (IFT) for the choosen frequencies. Then predictions of the IFT, original data and filtered data will be plotted.

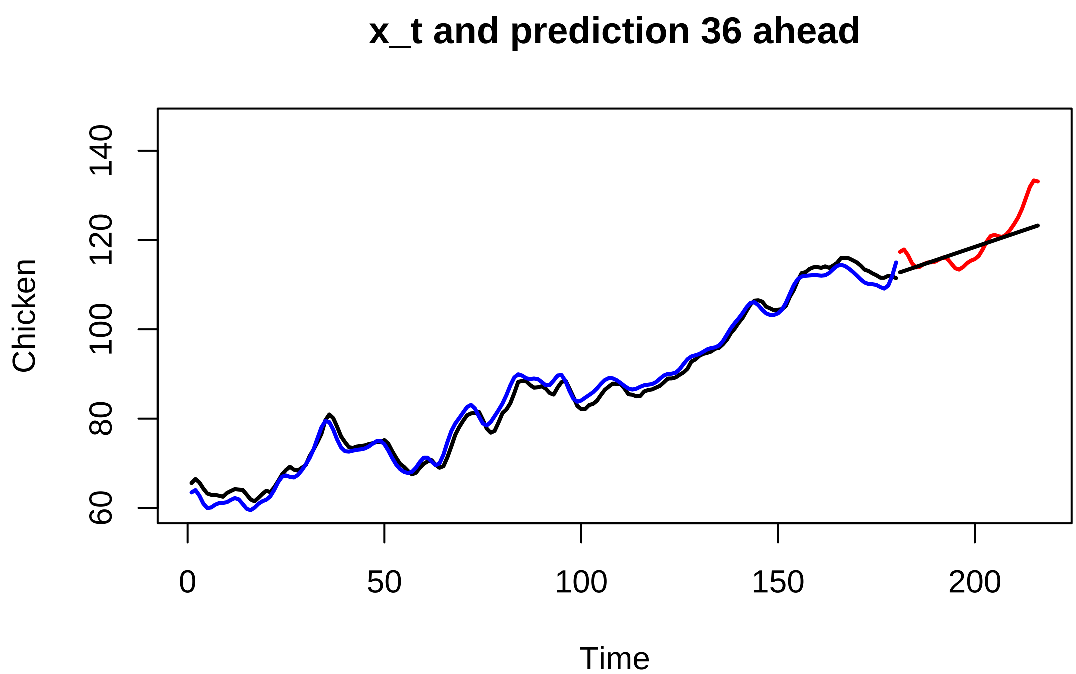


For the frequencies of the series, z_t, the baseline is the red line drawn in the plot. The all values below the line are set to 0. This is only visually determined as from the previous task. As to make a decent predicion on for the original data with a fourier transform we have to first produce a fast fourier transform on the detrended data (z_t). Then also use the same base line as in the task 3) with the filtered out frequencies. Then we perform the Inverse Fourier transform and get an smoother fit of the detrended data. We call this $\hat{e}_F$. Then we calcualte the fits for the original data by using $x_t + \hat{e}_F = \hat{x}_t$. We do the same method for obtaining a 36 step prediction. We use the linear regression obtained in step 2 and calculate a 36 step prediction and then perform the same method as obtaining $\hat{x}_t$.

The result is in the plot below where the black lines are the original data and fits from the linear regression in step 2. The blue line is $\hat{e}_F$



We can see that the detrended data (black line) and the Inverse Fourier Transform (IFT) (blue line) have a really good fit. This is an overfitted model for the detrended data where we have filtered all the frequencies below 15. We will still use the fits (blue line) as our $\hat{e}_F$ to se how it fits to the original data.

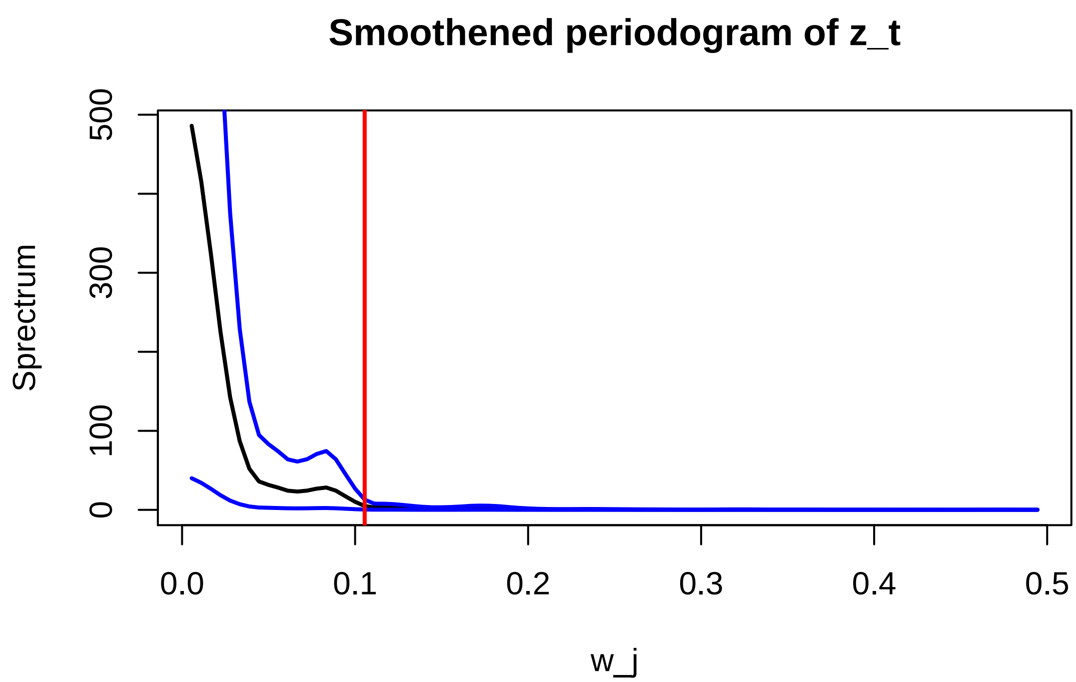


We can see that the fits from the inverse fourier transform (blue line) does fit the original data (black line) very well. We notice that there is probably a problem with overfitting due to the base line of $I_0 = 15$. The result is reasonable because the IFT should produce fits that is very similar original data. For example if we where to fit an fourier transform that is *exactly* the same fit as the original detrended data and perform this method to get the original data would obviusly get exactly the same data as the original data. If we where to raise the base line of I_0 we would obtain an even smoother fit.

All predictions made by the IFT will repeat the same pattern over and over again. This is due to that the method is based of estimate a period. So the model will not capture any anomalies that might apear in future data.

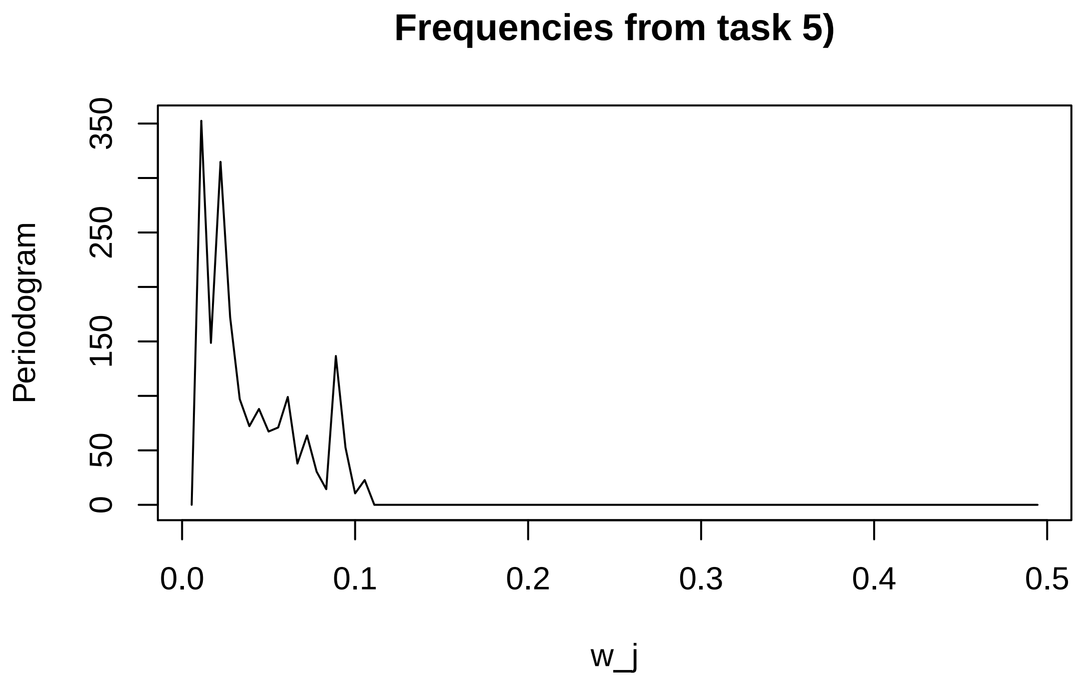
1.5

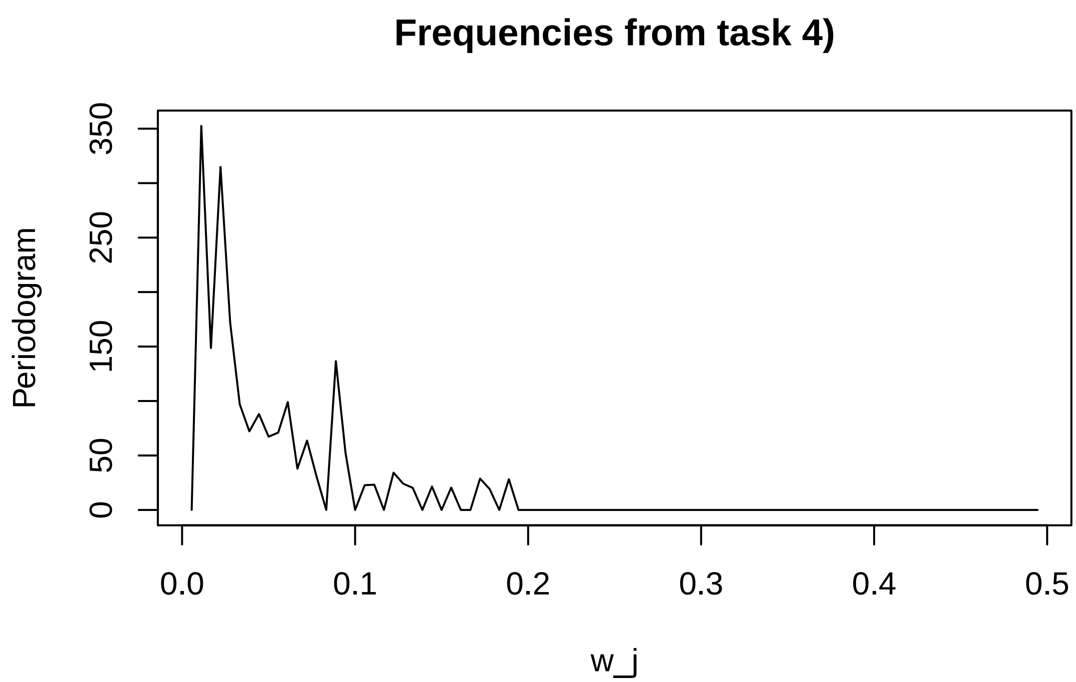
In this assignment we will comute a smoothened spectrum of z_t. This will be done by applying a Modified-Daniell(2,2). A smoothened periodogram will be comuted with confidence intervals for being able to detect dominant frequencies.



The blue lines in the graph is the lower and upper band of the CI for each frequency value. When we investigate the lower band of the CI we can see that to choose just above the noise we look at the lower bands that is larger then 1. We find that at position 19 and where $w_j = 0.1055556$ we draw a line (red line in plot). All the values that is lower then the value of where the line is drawn, are set to 0.

Next we will compare the frequencies extracted by the rule set in this task with the frequencies that were extracted at step 4.





The smoothing made it much easier to indentify frequencies. This is due to that you will be able to decide a limit for a w_j, which makes it much easier to filter out non-dominant frequencies. The frequencies obtained in task 4) are hard to see what is dominant or not.

1.6

In this task we will fit an ARIMA(2,1,0)x(0,0,1)$_{12}$ to the series x_t and make predictions 36 steps ahead. We will copare this prediction to the Fourier analysis prediction from step 4.

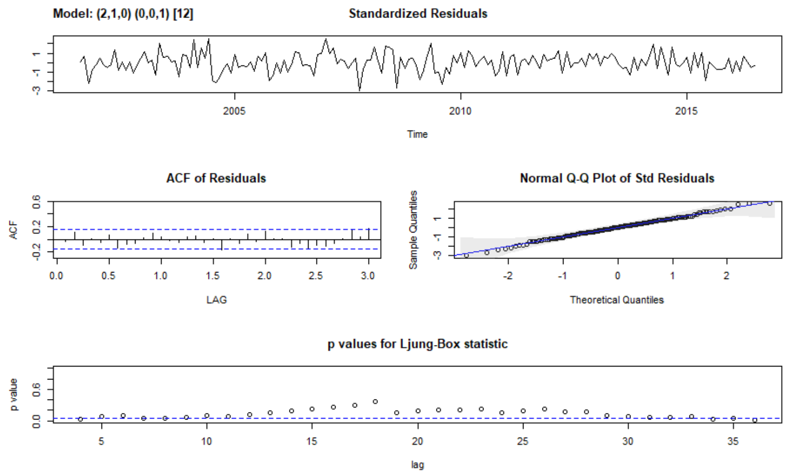
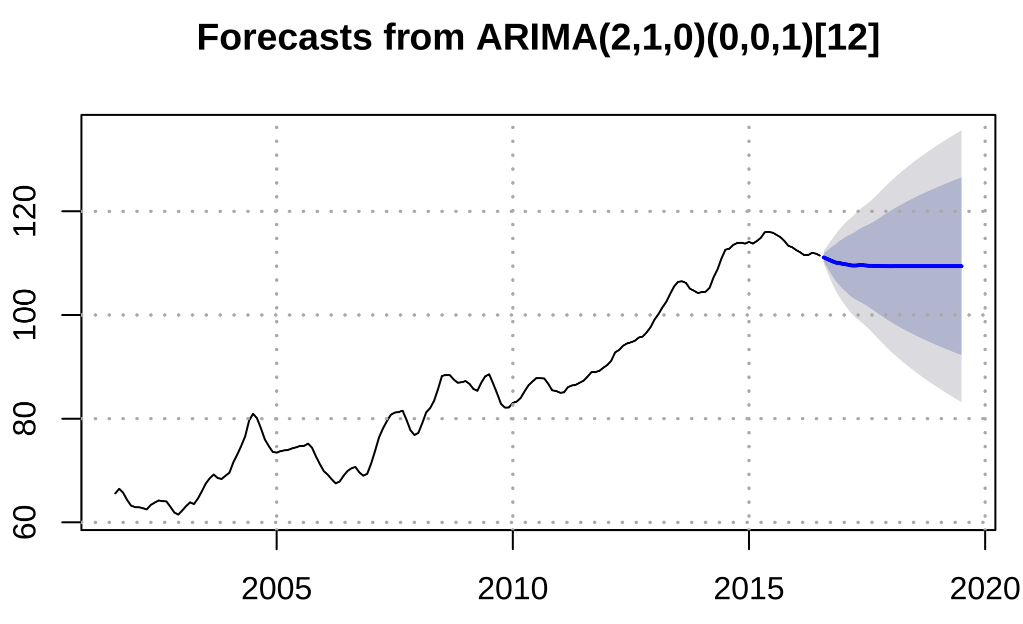
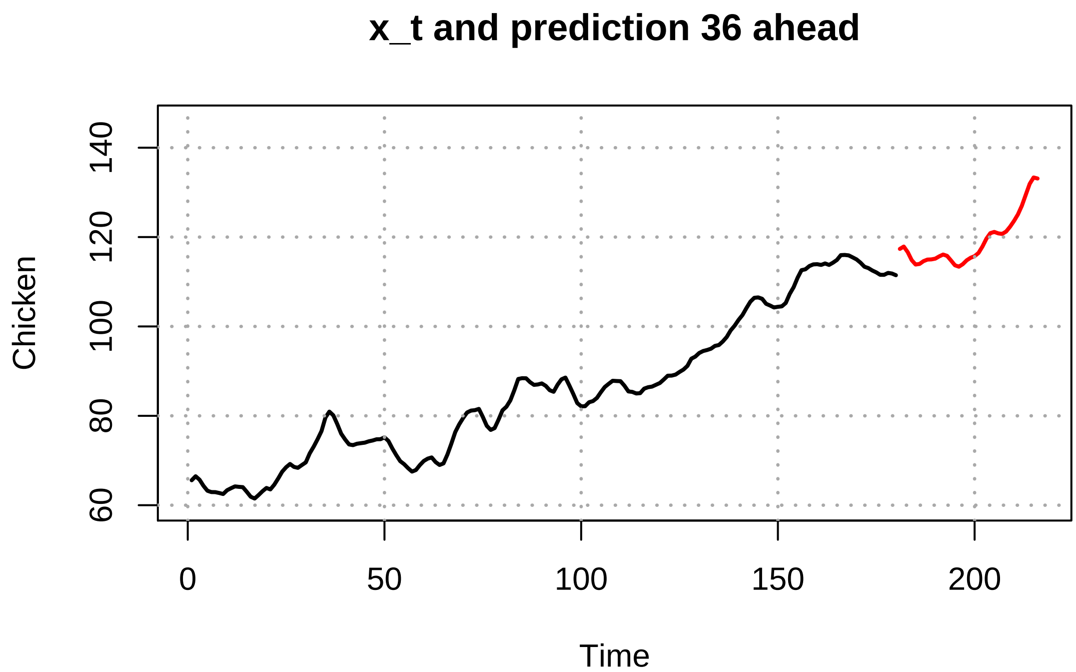


Figure 1:

We can see that from the diagnostic plots that the residuals seems to be white noise. It is also seen from the ACF that there is no autocorrelation in the residuals left. The residuals also seems normaldistributed according to the Q-Q plot. We also denote that from the plot showing the Ljung-Box statistic that for several lags, the test still indicate that autocorrelation might exist.





We can see that both of the predictions are reasonable, but as previously mentioned the IFT will replicate the pattern of the time series over and over. The ARIMA model will use the past values to predict, but will not repeat the exact the same pattern over and over. We can see that the ARIMA(2,1,0)x(0,0,1)$_{12}$ predicts a much more leveled prediction comapred to the IFT prediction that repeat its own fits for the first 36 observations but on a higher level.

In this case we choose to trust the IFT prediction due to that it still follows the trend compared to the ARIMA model.

1.7

In this task we will fit a ARIMA(3,0,0)x(0,0,1)$_{12}$ on the series z_t. Then we will compute the spectral density of the fitted ARIMA model. A plot of the spectrum for the ARIMA model will be compared to the spectrum plot in step 5.

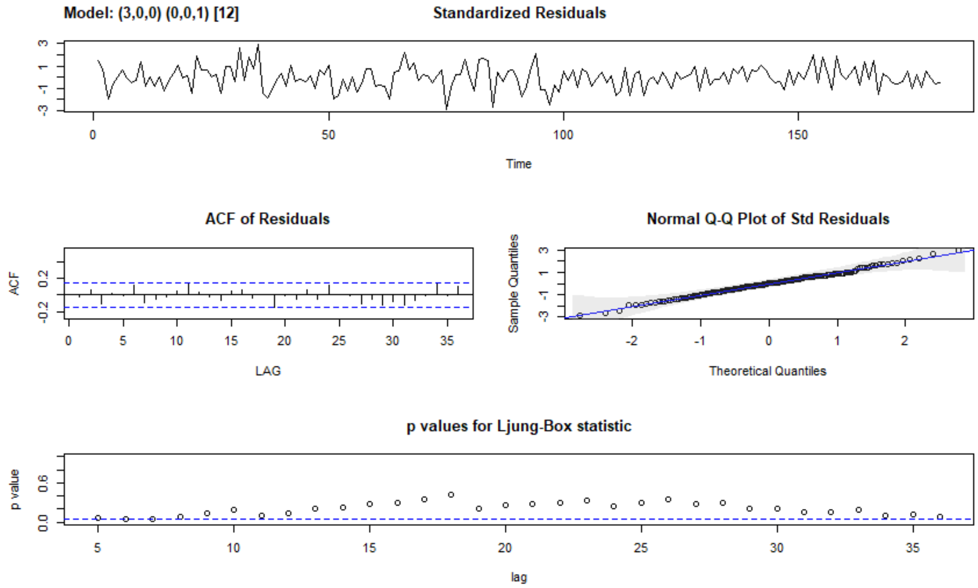
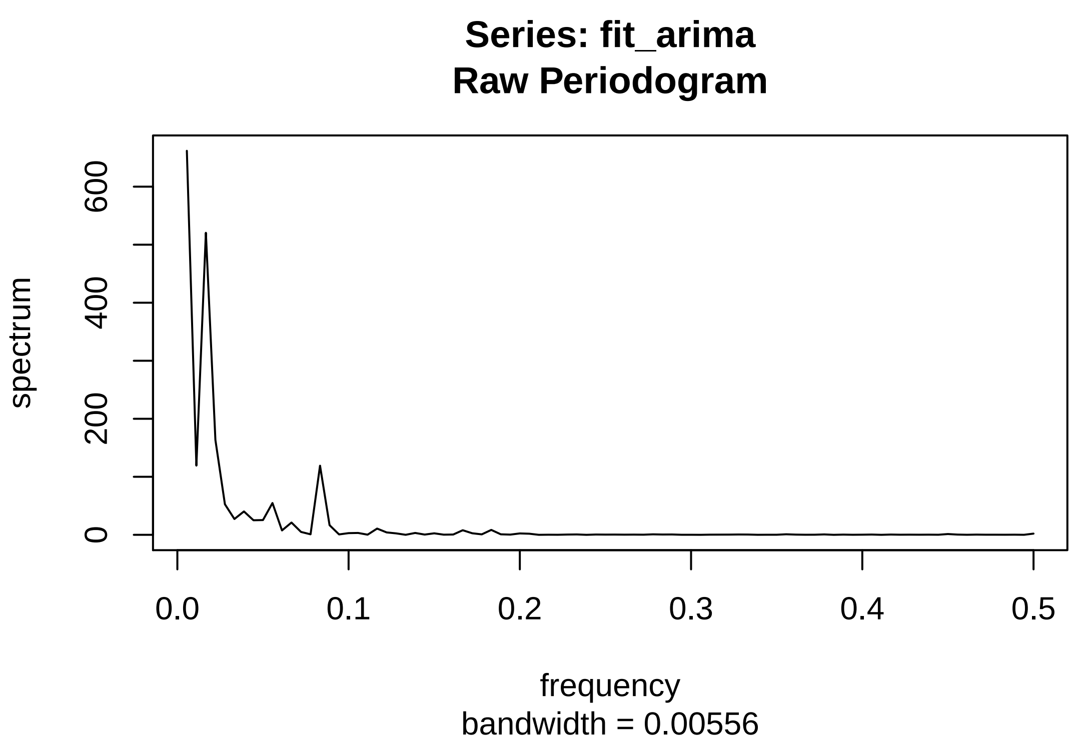
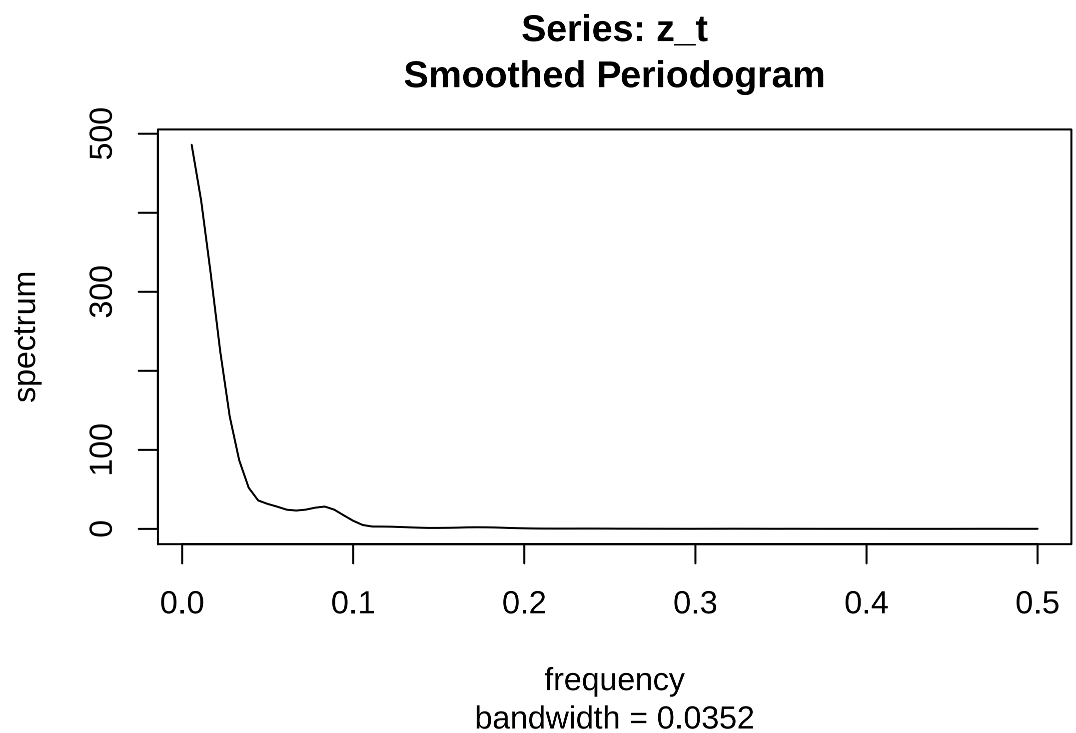


Figure 2:

We can see that from the diagnostic plots that the residuals seems to be white noise. It is also seen from the ACF that there is no autocorrelation in the residuals left. The residuals also seems normaldistributed according to the Q-Q plot. We also denote that from the plot showing the Ljung-Box statistic that for several lags, the test still indicate that autocorrelation might exist.





We can see that both periodograms have the same general pattern, but the smoothened periodogram from step 5 looks much more smoothened. We can see that in the both periodograms we would make differents conclusion on where to set the rule w_0. Beacuse if we where to decide the rule of w_0 that it is just above the noise level of the spectrum for the ARIMA, we would have a much harder time deciding that rule. This is due to that it has some very low sprectrum frequencies.

For indentify and deciding the dominant frequencies we would choose the method of ModifiedDaniell(2,2).

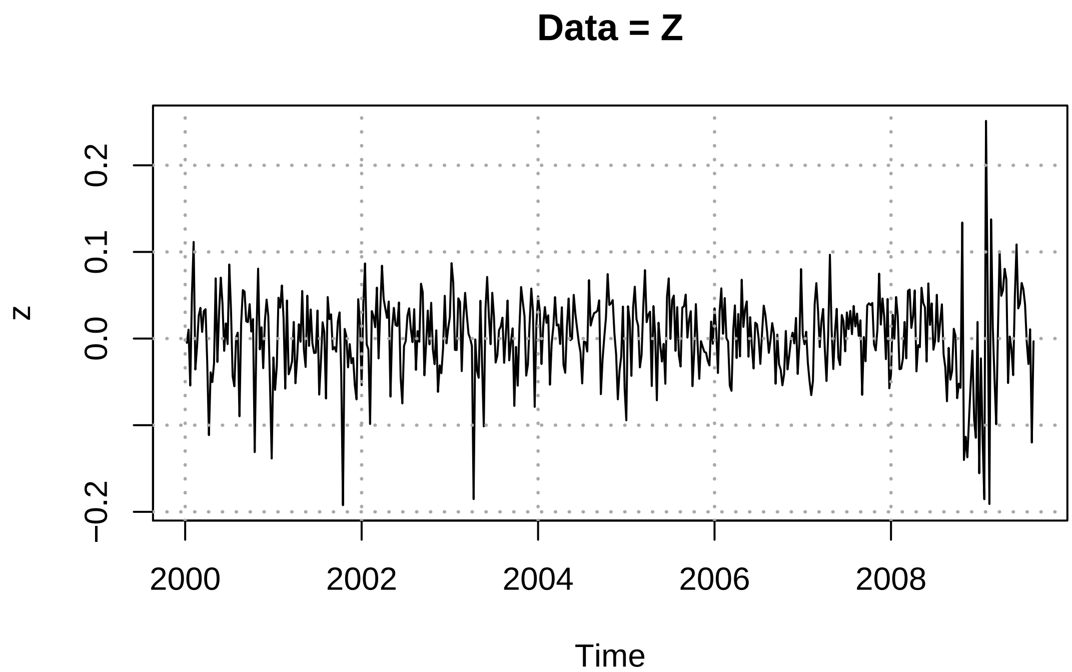
Assignment 2. GARCH modeling of the oil prices.

2.1

In this assignment we will study the oil data set in the package astsa.

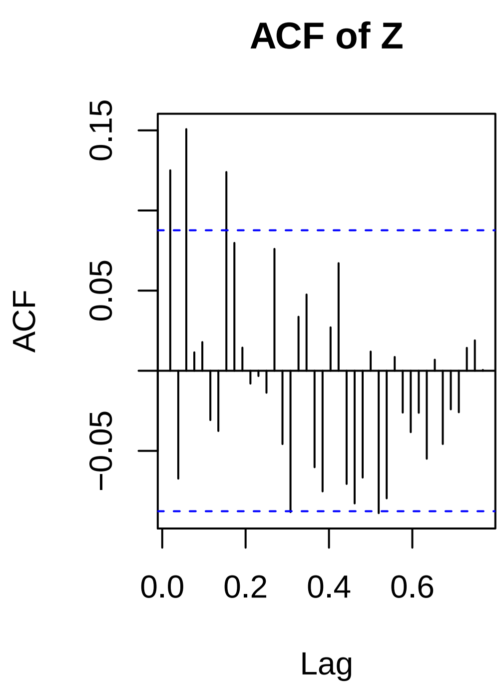
$x_t = \nabla log(oil)$

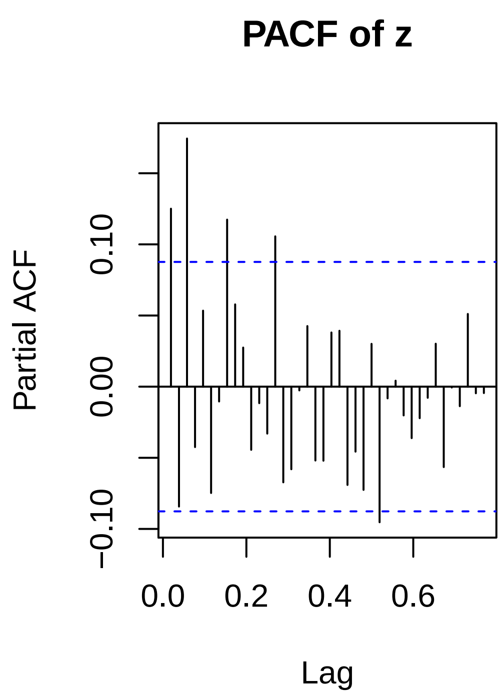
$z_t = x_t$ where t goes from 1 to 500



The time series z_t looks stationary. The only notable thing is the differ in variance in the end of the time series.

Now we want to build a ARMA model on the z_t. So we look at ACF, PACF and EACF.





```
## 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 cant se any clear spikes in both graphs, so we look at the EACF. It indicates that a propper model could be ARMA(1,1). So we build a ARMA(1,1) and look if the parameters gets significant.

```
##        Estimate     SE t.value p.value
## ar1     -0.5295 0.0908 -5.8342   0.000
## ma1      0.7136 0.0714  9.9966   0.000
## xmean    0.0017 0.0023  0.7346   0.463
```

```
##        AIC
## -5.120028
```

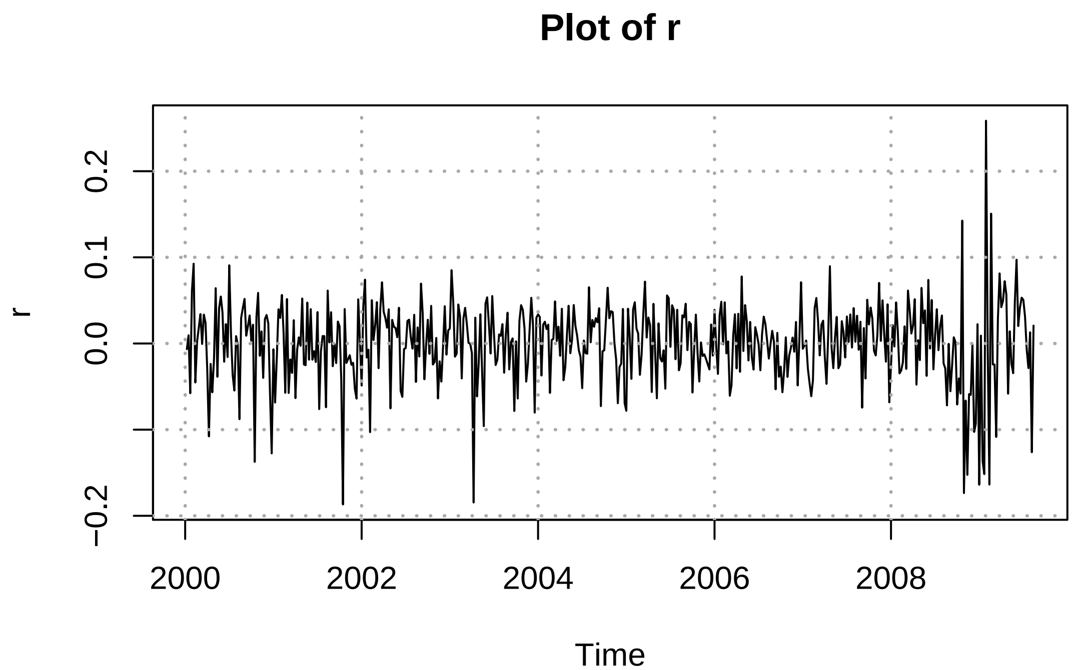
```
##       BIC
## -6.09474
```

Both the ar and ma part in the model gets significant.

2.2

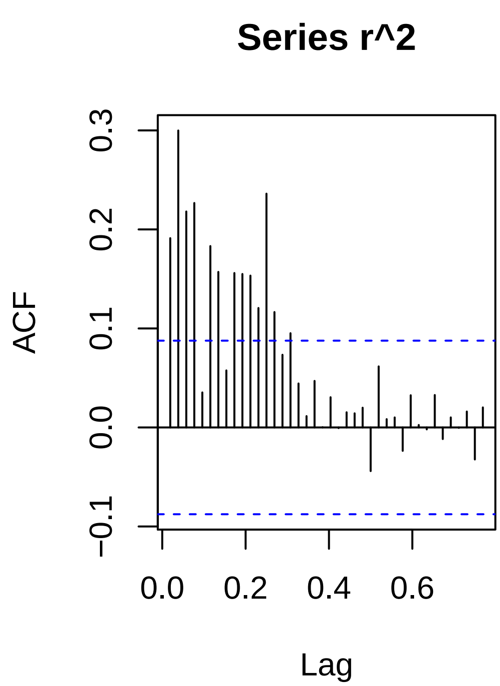
Now we want to make a residual analysis of the ARMA(1,1) model on z_t. We name the residuals as r_t

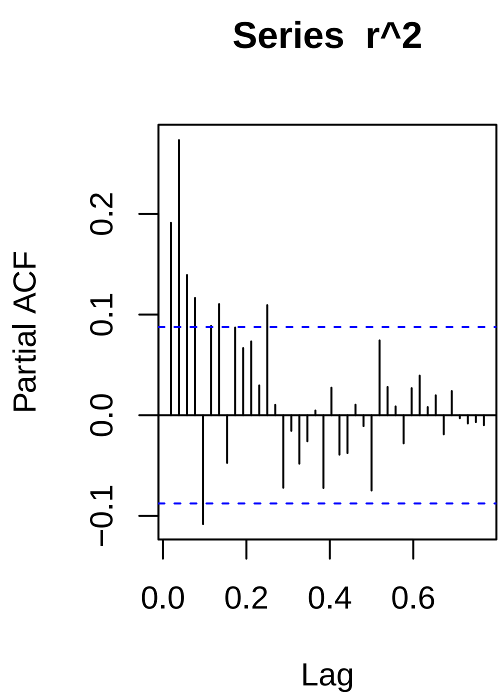
We first plot the residuals (r_t) over time.



The residuals look like white noise except the differ in variance in the end of the time series. To get a better model we will take care of this problem by fitting a Garch(p',q') to the model.

To determine p' and q' we look at ACF and PACF of Squared residuals (r^2)





Its a decreaseing pattern in both the ACF and PACF. Which means that we cant decide which p' and q' that is appropriate for ur data.

2.3

Beacuse we could not determine p' and q' in the ACF and PACF we need to try us out by fitting models. We fit the models

- Model 1=ARMA(1,1)-GARCH(1,1)
- Model 2=ARMA(1,1)-GARCH(2,2)
- Model 3=ARMA(1,1)-GARCH(1,2)
- Model 4=ARMA(1,1)-GARCH(2,1)

to se which model seems to be the best.

We start by looking if the models parameters is significant.

```
garch_model1@fit$matcoef
```

```
##          Estimate  Std. Error  t value Pr(>|t|)
## mu          0.007       0.003    2.275    0.023
## ar1        -0.467       0.104   -4.474    0.000
## ma1         0.661       0.086    7.652    0.000
## omega       0.000       0.000    1.593    0.111
## alpha1      0.060       0.022    2.784    0.005
## beta1       0.895       0.042   21.543    0.000
## shape       8.101       2.399    3.376    0.001
```

```
garch_model2@fit$matcoef
```

```
##          Estimate  Std. Error  t value Pr(>|t|)
## mu          0.007       0.003    2.387    0.017
## ar1        -0.488       0.097   -5.017    0.000
## ma1         0.676       0.080    8.470    0.000
## omega       0.000       0.000    1.682    0.093
## alpha1      0.008       0.036    0.219    0.826
## alpha2      0.115       0.049    2.332    0.020
## beta1       0.103       0.174    0.588    0.557
## beta2       0.693       0.173    4.001    0.000
## shape       7.676       2.138    3.591    0.000
```

```
garch_model3@fit$matcoef
```

```
##          Estimate  Std. Error  t value Pr(>|t|)
## mu          0.007       0.003    2.270    0.023
## ar1        -0.467       0.105   -4.451    0.000
## ma1         0.661       0.087    7.623    0.000
## omega       0.000       0.000    1.310    0.190
## alpha1      0.061       0.036    1.678    0.093
## beta1       0.895       0.648    1.381    0.167
## beta2       0.000       0.598    0.000    1.000
## shape       8.101       2.414    3.356    0.001
```

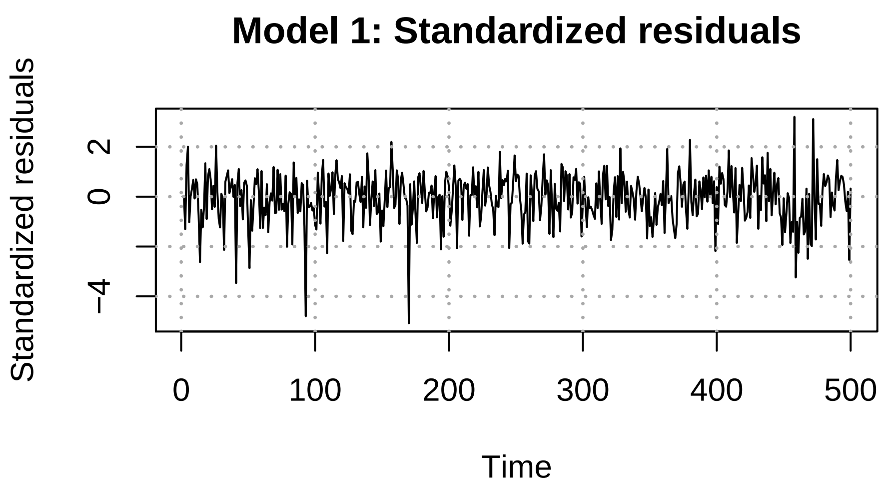
```
garch_model4@fit$matcoef
```

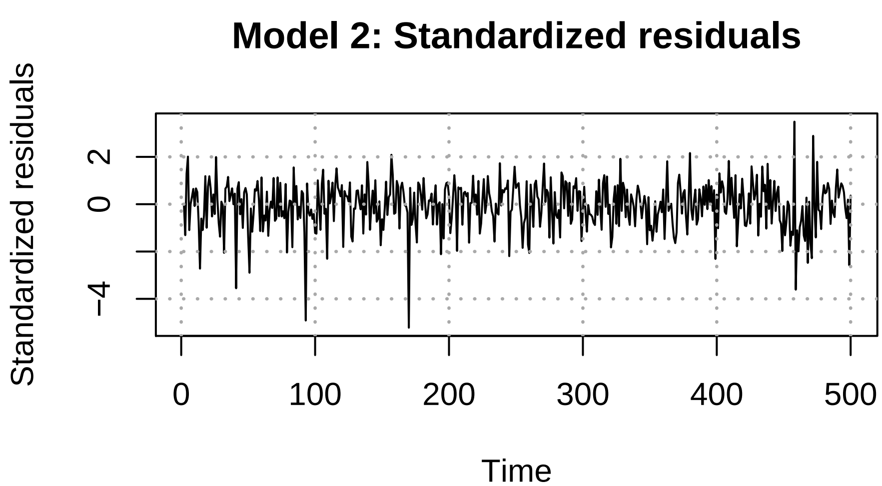
```
##          Estimate  Std. Error  t value Pr(>|t|)
## mu          0.007       0.003    2.373    0.018
```

```
## ar1        -0.489       0.103   -4.751    0.000
## ma1         0.673       0.084    7.992    0.000
## omega       0.000       0.000    1.619    0.105
## alpha1      0.007       0.042    0.176    0.860
## alpha2      0.068       0.056    1.214    0.225
## beta1       0.868       0.054   16.181    0.000
## shape       7.622       2.149    3.547    0.000
```

All the models have significant ar and ma parts. Then we can look at the alpha and beta parts in the models. Model 4 have alpha2 un-significant and model 3 have beta2 un-significant. Model 2 have beta1 and alpha1 un-significant but beta2 and alpha2 significant which indicates that model 2 could be a appropriate model.

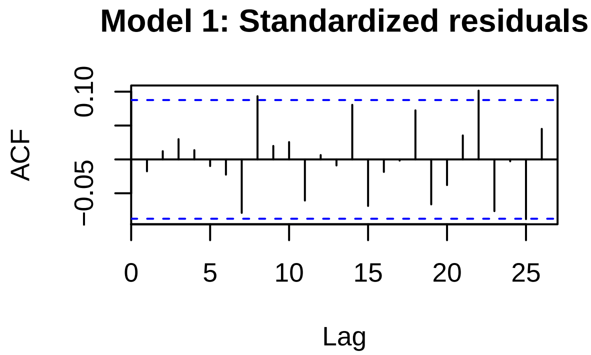
So now will we decide if model 1 or model 2 is the best model for ur data by looking at standardized residuals.

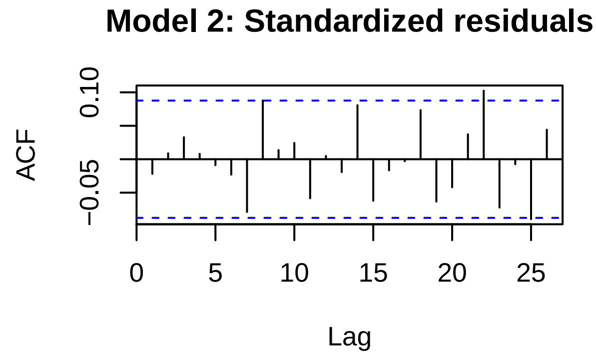


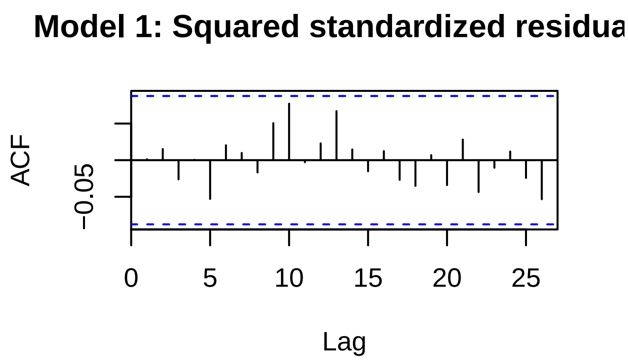


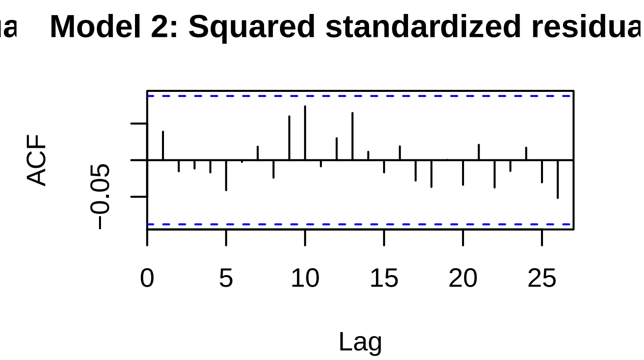
The standardized residuals for model 1 and model 2 looks very simular. It seems like the variance is constant over time now in both models.

We also want to look at the ACF of the standardized residuals and the squared standardized residuals to se if we got autocorrelation and if we have modeling the differ in variance on a good way.



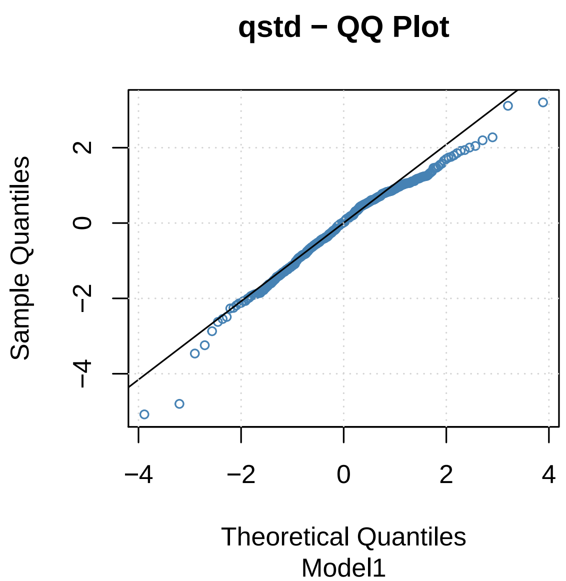


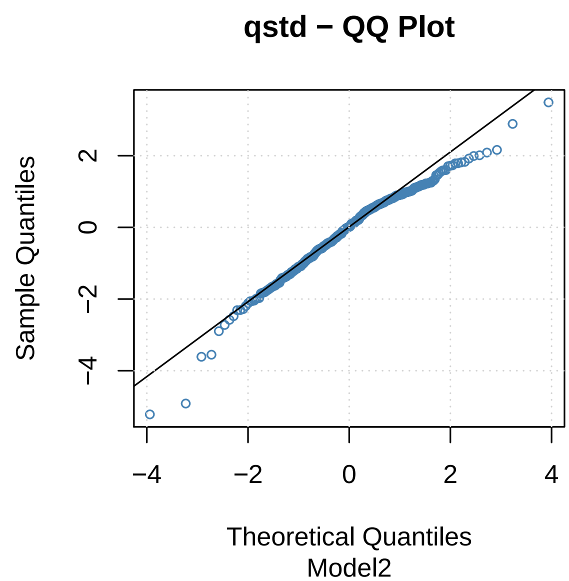




The models ACF of standardized residuals and the squared standardized residuals look very simular. We dont have any spikes in either the ACF of standardized residuals or the ACF of squared standardize residuals. Which means that we dont have any autocorrelation and the variance seems to be constant over time.

A requirement for the ARMA(p,q)-GARCH(p',q') model is that the residuals should be normally distributed. We look in a Q-Q plot for analyze the requirement.





The graphs is very simular. The tails of the the data dont follows the line in the graphs, witch indicates that the residuals not is normally distributed. We can check the normally distributed requirement by the test Jarque-Bera Test.

Here comes the symmary of model 1 and model 2.

Reminder of which models we are talkning about.

- Model 1=ARMA(1,1)-GARCH(1,1)
- Model 2=ARMA(1,1)-GARCH(2,2)

```
##
## Title:
##  GARCH Modelling
##
## Call:
##  garchFit(formula = ~arma(1, 1) + garch(1, 1), data = z, cond.dist = "std")
##
## Mean and Variance Equation:
##  data ~ arma(1, 1) + garch(1, 1)
## <environment: 0x00000000228128d8>
##  [data = z]
##
## Conditional Distribution:
##  std
##
## Coefficient(s):
##          mu          ar1          ma1        omega       alpha1
##  6.9644e-03  -4.6742e-01   6.6087e-01   8.7771e-05   6.0400e-02
##       beta1        shape
##  8.9513e-01   8.1011e+00
##
## Std. Errors:
##  based on Hessian
##
## Error Analysis:
```

```
##         Estimate  Std. Error  t value Pr(>|t|)
## mu         0.007       0.003    2.275    0.023 *
## ar1       -0.467       0.104   -4.474   <2e-16 ***
## ma1        0.661       0.086    7.652   <2e-16 ***
## omega      0.000       0.000    1.593    0.111
## alpha1     0.060       0.022    2.784    0.005 **
## beta1      0.895       0.042   21.543   <2e-16 ***
## shape      8.101       2.399    3.376    0.001 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
##  867.0019    normalized:  1.734004
##
## Description:
##  Tue Oct 17 22:58:54 2017 by user: Eric
##
##
## Standardised Residuals Tests:
##                                 Statistic p-Value
##  Jarque-Bera Test   R    Chi^2  123.9226  0
##  Shapiro-Wilk Test  R    W      0.9702826 1.562755e-08
##  Ljung-Box Test     R    Q(10)  9.251306  0.5084315
##  Ljung-Box Test     R    Q(15)  17.01496  0.317971
##  Ljung-Box Test     R    Q(20)  22.98842  0.2893635
##  Ljung-Box Test     R^2  Q(10)  6.654957  0.757568
##  Ljung-Box Test     R^2  Q(15)  9.475563  0.8513694
##  Ljung-Box Test     R^2  Q(20)  11.20706  0.9406749
##  LM Arch Test       R    TR^2   6.614136  0.8820251
##
## Information Criterion Statistics:
##       AIC       BIC       SIC      HQIC
## -3.440007 -3.381003 -3.440392 -3.416854
```

```
##
## Title:
##  GARCH Modelling
##
## Call:
##  garchFit(formula = ~arma(1, 1) + garch(2, 2), data = z, cond.dist = "std")
##
## Mean and Variance Equation:
##  data ~ arma(1, 1) + garch(2, 2)
## <environment: 0x0000000021eb2100>
##  [data = z]
##
## Conditional Distribution:
##  std
##
## Coefficient(s):
##          mu          ar1          ma1        omega       alpha1
##  0.00726133  -0.48838637   0.67609178   0.00016618   0.00791226
##      alpha2        beta1        beta2        shape
##  0.11505110   0.10252320   0.69259556   7.67647148
```

```
##
## Std. Errors:
##  based on Hessian
##
## Error Analysis:
##         Estimate  Std. Error  t value Pr(>|t|)
## mu         0.007       0.003    2.387    0.017 *
## ar1       -0.488       0.097   -5.017   <2e-16 ***
## ma1        0.676       0.080    8.470   <2e-16 ***
## omega      0.000       0.000    1.682    0.093 .
## alpha1     0.008       0.036    0.219    0.826
## alpha2     0.115       0.049    2.332    0.020 *
## beta1      0.103       0.174    0.588    0.557
## beta2      0.693       0.173    4.001   <2e-16 ***
## shape      7.676       2.138    3.591   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
##  868.7763    normalized:  1.737553
##
## Description:
##  Tue Oct 17 22:58:55 2017 by user: Eric
##
##
## Standardised Residuals Tests:
##                                 Statistic p-Value
##  Jarque-Bera Test   R    Chi^2  160.9756  0
##  Shapiro-Wilk Test  R    W      0.9668412 3.362419e-09
##  Ljung-Box Test     R    Q(10)  8.680334  0.5626857
##  Ljung-Box Test     R    Q(15)  15.99853  0.3821497
##  Ljung-Box Test     R    Q(20)  21.97107  0.3420825
##  Ljung-Box Test     R^2  Q(10)  7.020319  0.7235254
##  Ljung-Box Test     R^2  Q(15)  9.875054  0.8275249
##  Ljung-Box Test     R^2  Q(20)  11.75659  0.9241993
##  LM Arch Test       R    TR^2   6.970754  0.8595402
##
## Information Criterion Statistics:
##       AIC       BIC       SIC      HQIC
## -3.439105 -3.363242 -3.439738 -3.409337
```

Jarque-Bera test is significant in both models witch indicates that the residuals is not normally distributed. But as we saw in the Q-Q we think we think that it is the tails that not follows a normal distribution.

Ljung-Box Test for the Standardize residuals and for the Squared standardize residuals at the lag 10, 15 and 20 is un-significant. Which indicates on that the residuals is not autocorrelated and that we have has made a good modling for the defference in variance.

If we compare the AIC and BIC for the model we get lowest values for model 2 (AIC=-3.39 and BIC=-3.32). But model 1 have very simular values of AIC and BIC.

Model 1 and model 2 seems to be very simmilar. In the graphs we can't se any difference and the AIC and BIC is very simular. Because model 1 is less complex we choose model 1.

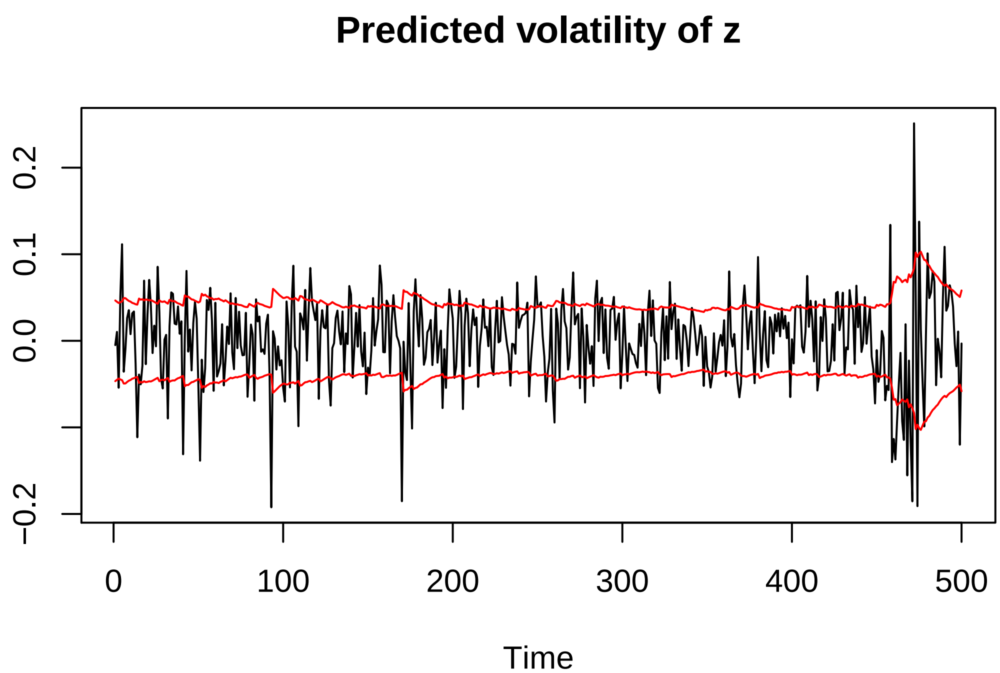
- Model 1=ARMA(1,1)-GARCH(1,1)

$r_t = \sigma_t \varepsilon_t$

$\hat{\sigma}_t^2 = \hat{\alpha}_0 + \hat{\alpha}_1 r_{t-1}^2 + \hat{\beta}_1 \hat{\sigma}_t^2 = 0.00001 + 0.060 r_{t-1}^2 + 0.895 \hat{\sigma}_t^2$

2.4

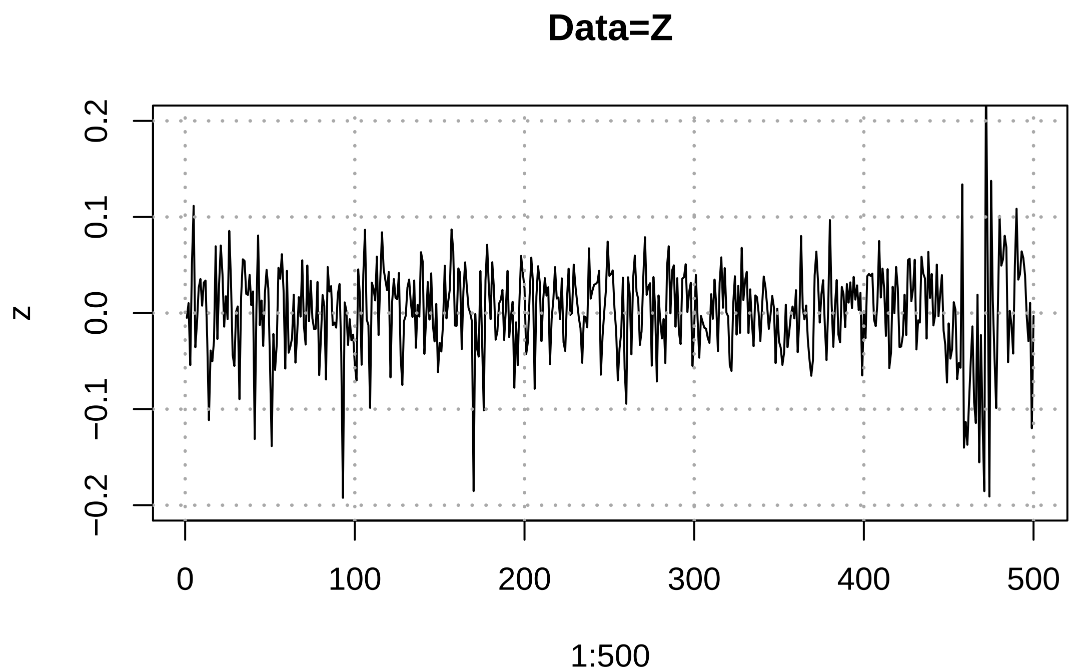
In this assignment we will present z_t and the predicted volatility in one plot. We will look if the predicted volatility pattern seem to follow the variation of the data.

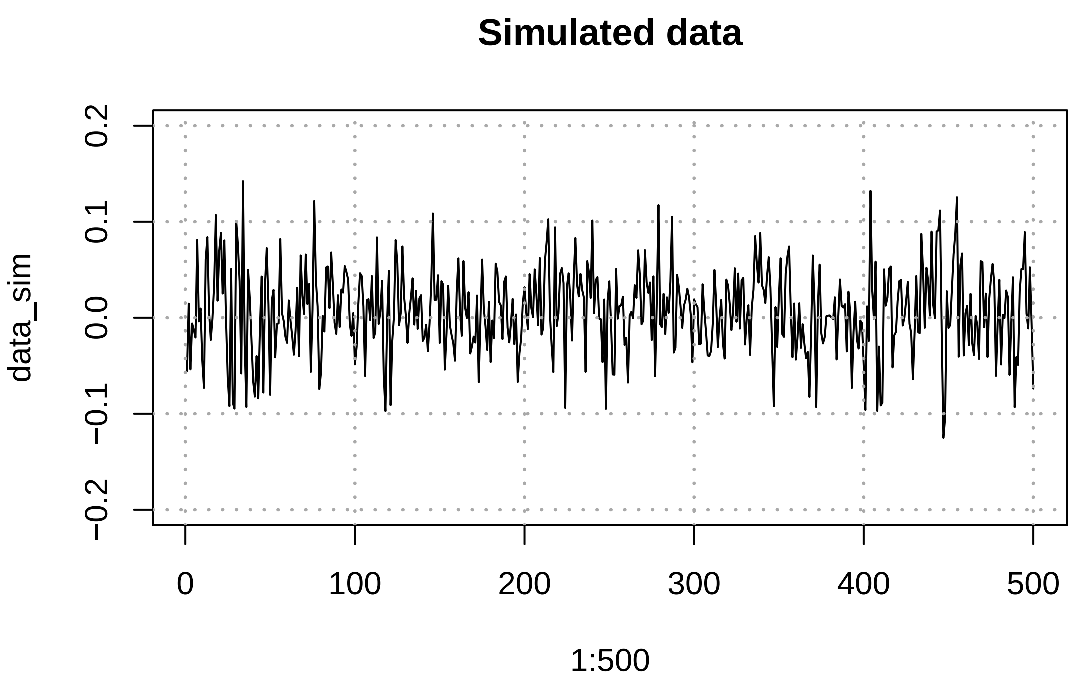


The graph shows the z and the predicted volatility. The predicted volatility seems to follow the variation of the data. In the end of the time series the variance of the data is higher then the rest of the data, witch the predicted volatility allso says.

2.5

Now we will simulate 500 observations with the same parameters from the chosen model ARMA(1,1)-GARCH(1,1).



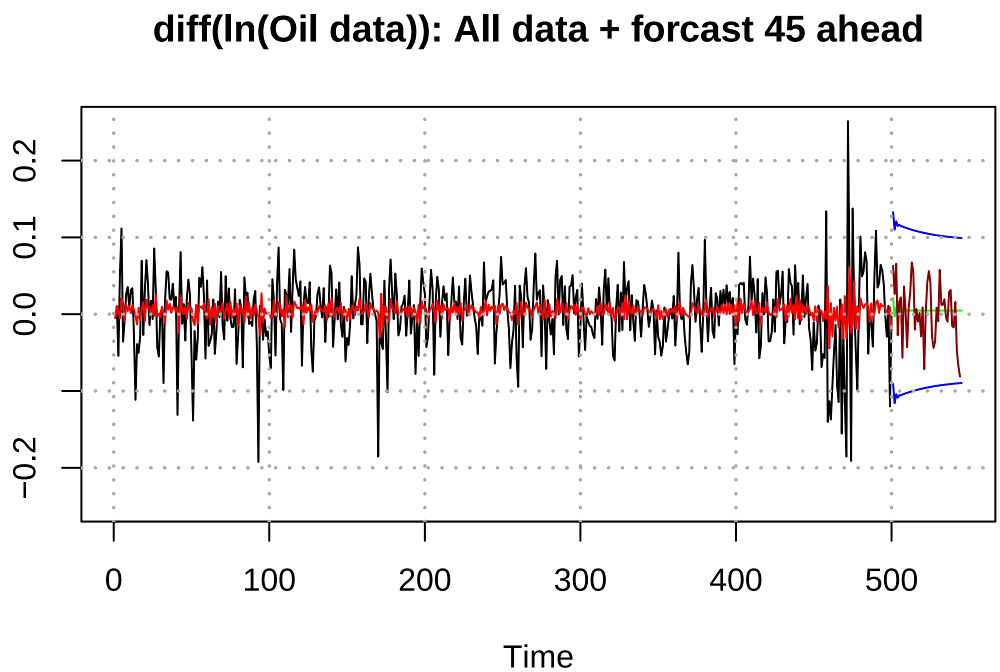


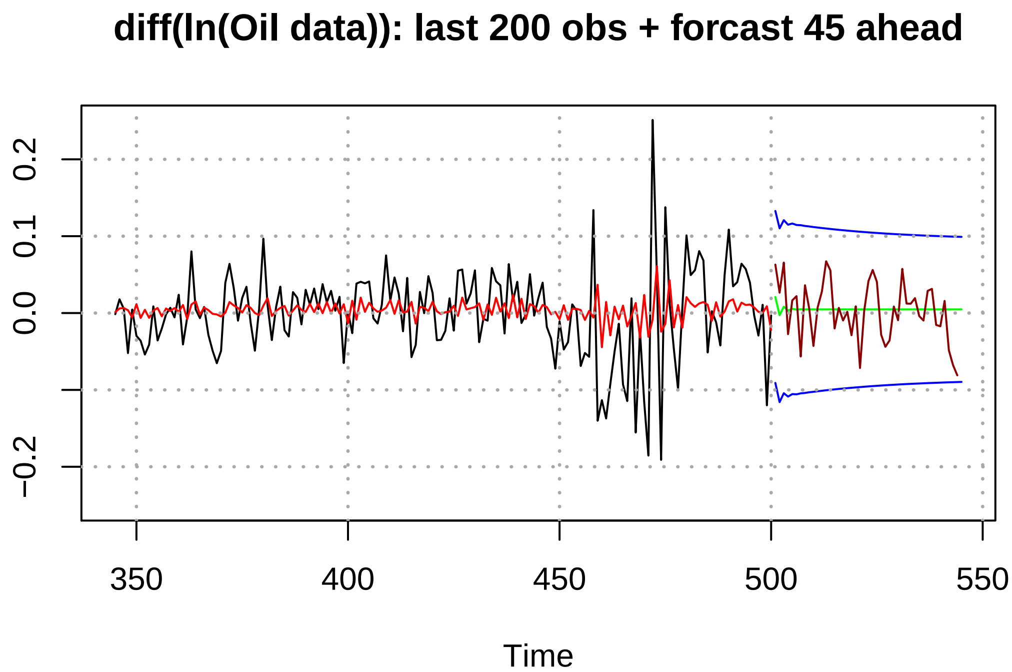
The graphs are simular in several things. They have simular variance, both moving around zero and jumping down and up equal. This indicates that we have made a ok model.

2.6

In this assignment we will compute and plot 45 step ahead predictions from the obtained ARMA(1,1)-GARCH(1,1) and look how good our prediction is to the true last values from x_t.

The red line in the graph is the fitted values of z_t and the dark red line is the last 45 values of x_t. The blue lines is the 95% confidence interval and the green line is 45 step ahead predicted.





All the last 45 values of x_t fall inside the confidens intervall. Unfortunately is the prediction bands very wide which indicates on a less good model.

R-code

1.1

```r
data(chicken)
plot.ts(chicken, main = "x_t")
```

1.2

```r
x_t <- chicken
t <- 1:180
model <- summary(lm(x_t ~ t))
z_t <- ts(model$residuals)
plot(z_t, main = "Detrended data (z_t")
```

1.3

```r
fourier <- fft(z_t)
real_values <- abs(fourier)
w_j <- 1:((180/2)-1) / 180


plot(y = real_values[1:89], x = w_j, type = "l", main = "Periodogram of z_t", ylab= "Periodogram")
abline(h = 15, col = "red")

#takes the largest value
ind= which.max(real_values[1:90])

Upper=2*real_values[ind]/qchisq(0.025,2)
Lower=2*real_values[ind]/qchisq(0.975,2)
CI <- c(Lower, Upper)
names(CI) <- c("Lower", "Upper")
CI
```

1.4

```r
d_j_n <- fft(z_t)
real_values <- abs(d_j_n)


I_0 <- 15
d_j_n[which(real_values < I_0)] <- 0
plot(y = abs(d_j_n[1:89]), x = w_j[1:89], type = "l", main = "Filtered periodogram of z_t", ylab= "Peri
abline(h = I_0, col = "red")


#INVERSE FOURIER TRANSFORMATION
new_xt <- c()
```

```
#this formuala on PP slide nr. 13
for(t in 1:length(d_j_n)){
  sum_vect <- c()
  for(j in 0:(length(d_j_n)-1)){
    sum_vect[j+1] <- d_j_n[j+1]*exp(complex(imaginary = 2*pi* (j)/180 * t))
  }
  new_xt[t] <- 1/length(d_j_n) * sum(sum_vect[1:(length(sum_vect)-1)])
}

#Smoothad
plot(1:180, z_t, type = "l", lwd =2, xlab="Time", main = "The detrended data and IFT fits")
lines(Re(new_xt), col = "blue", lwd = 2)


#PREDICTION OF 36 OBS
pred <- 36

pred_ict <- c()
for(t in (length(d_j_n)+1):(length(d_j_n)+pred)){
  sum_vect <- c()
  for(j in 0:(length(d_j_n)-1)){
    sum_vect[j+1] <- d_j_n[j+1]*exp(complex(imaginary = 2*pi* (j)/180 * t))
  }
  pred_ict[t] <- 1/length(d_j_n) * sum(sum_vect[1:(length(sum_vect)-1)])
}



#predicts of the linear model
coefic <- model$coefficients[,1]
predicts <- coefic[1] + coefic[2]*(181:(180+36))
fits_x <- coefic[1] + coefic[2]*1:180

e_F_hat <- new_xt
#fitted values
x_t_hat <- fits_x + Re(e_F_hat)


pred_x_t_from_F <- predicts + Re(pred_ict[181:(180+36)])
pred_x_t_hat_F <- c(x_t_hat, pred_x_t_from_F)

#Final plot
plot(y = c(rep(NA, 180), pred_x_t_from_F) , x = 1:length(pred_ict), type = "l",
     col = "red", ylim = c(60,(max(x_t)+30) ), lwd = 2, ylab= "Chicken", xlab="Time",main="x_t and predi
lines(as.numeric(x_t), lwd=2)
lines(as.numeric(Re(x_t_hat)), col = "blue", lwd = 2)
lines(c(rep(NA,180), predicts), lwd = 2)
```

1.5

```r
k=kernel("modified.daniell", c(2,2))

smooth_z_t = mvspec(z_t, kernel=k, log="no", plot = FALSE)


#CI for every frequency value
matr_CI <- matrix(NA, nrow = length(smooth_z_t$spec)-1, ncol = 2)
for(i in 1:nrow(matr_CI)){
  Upper=2*smooth_z_t$Lh*smooth_z_t$spec[i] / qchisq(0.025,2*smooth_z_t$Lh)
  Lower=2*smooth_z_t$spec[i] / qchisq(0.975,2*smooth_z_t$Lh)
  matr_CI[i,] <- c(Lower, Upper)

}

w_j <- 1:(180/2-1) / 180
#Plot
w_0 <- 19
plot(y = smooth_z_t$spec[1:(length(smooth_z_t$spec)-1)], x = w_j, type = "l",
     main = "Smoothened periodogram of z_t", ylab= "Sprectrum",lwd = 2)
lines(y=matr_CI[,1],x = w_j,col= "blue",lwd = 2)
lines(y=matr_CI[,2],x = w_j,col= "blue",lwd = 2)
abline(v = w_j[w_0], col = "red",lwd = 2)


d_j_nNext <- fft(z_t)

#set all numbers below w_0 to 0
d_j_nNext[(w_0+1):(length(d_j_nNext)-w_0-1)] <- 0

#tar fram periodogrammet och jämför med uppg. 4)
plot(y = abs(d_j_nNext[1:89]), x = w_j[1:89], type = "l", main = "Frequencies from task 5)", ylab = "Pe
plot(y = abs(d_j_n[1:89]), x = w_j[1:89], type = "l", main = "Frequencies from task 4)", ylab = "Period
```

1.6

```r
p=2; d=1; q=0
P=0; D=0; Q=1; S=12

a <- stats::arima(x_t,order = c(p,d,q),seasonal = list(order = c(P,D,Q), period = S))
predd <- forecast(a,h=36)
plot(predd)
grid(col = "dark grey",lwd = 1.7)

plot(y = c(rep(NA, 180), pred_x_t_from_F) , x = 1:length(pred_ict), type = "l",
     col = "red", ylim = c(60,(max(x_t)+30) ), lwd = 2, ylab= "Chicken", xlab="Time",main="x_t and predi
lines(as.numeric(Re(x_t)), lwd = 2, type="l")
grid(col = "dark grey",lwd = 1.7)
```

1.7

```
p=3; d=0; q=0
P=0; D=0; Q=1; S=12


model_season <- stats::arima(z_t,order = c(p,d,q),seasonal = list(order = c(P,D,Q), period = S))
#according to the diagnostic plots, the model seem really nice!
fit_arima <- fitted(model_season)

a <- mvspec(fit_arima, log="no", plot = TRUE)


b <- mvspec(z_t,  kernel = k, log="no", plot = TRUE)
```

2.1

```
data(oil)
x<-diff(log(oil,base = exp(1)))
z<-x[1:((c(1:length(time(x)))[time(x)==(2009+33/52)])-1)]
z<-ts(z,start =  c(2000, 2) ,frequency = 52)


plot(z,main="Data = Z")
grid(col = "dark grey",lwd = 1.7)

par(mfrow = c(1,2))
acf(z,40,main="ACF of Z")
pacf(z,40,main="PACF of z")
par(mfrow = c(1,1))

eacf(x)

p<-1;d<-0;q<-1
P<-0;D<-0;Q<-0;S=0
model1<-sarima(z,p=p,d=d,q=q,
               P=P,D=D,Q=Q,S=S)
r<-model1$fit$residuals

model1$ttable
names(model1$AIC)<-"AIC"
model1$AIC
names(model1$BIC)<-"BIC"
model1$BIC
```

2.2

```
r<-ts(r,start =  c(2000, 2) ,frequency = 52)

plot(r,main="Plot of r")
```

```
grid(col = "dark grey",lwd = 1.7)

par(mfrow = c(1,2))
acf(r^2,40)
pacf(r^2,40)
par(mfrow = c(1,1))
#Båda avtagande mönster börja med 1,1
```

2.3

```
### Garch 1,1
garch_model1<-garchFit(~arma(1,1)+garch(1,1),data=z, cond.dist="std")
garch_model1@fit$matcoef<-round(garch_model1@fit$matcoef,3)

### Garch 2,2
garch_model2<-garchFit(~arma(1,1)+garch(2,2),data=z, cond.dist="std")
garch_model2@fit$matcoef<-round(garch_model2@fit$matcoef,3)

### Garch 1,2
garch_model3<-garchFit(~arma(1,1)+garch(1,2),data=z, cond.dist="std")
garch_model3@fit$matcoef<-round(garch_model3@fit$matcoef,3)


### Garch 2,1
garch_model4<-garchFit(~arma(1,1)+garch(2,1),data=z, cond.dist="std")
garch_model4@fit$matcoef<-round(garch_model4@fit$matcoef,3)


garch_model1@fit$matcoef
garch_model2@fit$matcoef
garch_model3@fit$matcoef
garch_model4@fit$matcoef

# standardize residuals model 1 och 2 ####
standardize_residuals_model1<-residuals(garch_model1,standardize = TRUE)
standardize_residuals_model2<-residuals(garch_model2,standardize = TRUE)

par(mfrow = c(1,1))
plot.ts(standardize_residuals_model1,ylab="Standardized residuals",main="Model 1: Standardized residuals
grid(col = "dark grey",lwd = 1.7)
plot.ts(standardize_residuals_model2,ylab="Standardized residuals",main="Model 2: Standardized residuals
grid(col = "dark grey",lwd = 1.7)
par(mfrow = c(1,1))

# ACF
par(mfrow = c(2,2))
acf(standardize_residuals_model1,main="Model 1: Standardized residuals")
acf(standardize_residuals_model2,main="Model 2: Standardized residuals")
acf(standardize_residuals_model1^2,main="Model 1: Squared standardized residuals")
acf(standardize_residuals_model2^2,main="Model 2: Squared standardized residuals")
par(mfrow = c(1,1))
```

```
# QQ plot
par(mfrow = c(1,2))
plot(garch_model1,which=13)
title(sub="Model1")
plot(garch_model2,which=13)
title(sub="Model2")
par(mfrow = c(1,1))

# Jarque-Bera Test, Ljung-Box Test, AIC and BIC
summary(garch_model1)
summary(garch_model2)
```

2.4

```
plot(1:length(z),z,type="l",main="Predicted volatility of z",ylab="",xlab="Time")
lines(garch_model1@sigma.t,col="red")
lines(-garch_model1@sigma.t,col="red")
```

2.5

```
mu<-garch_model1@fit$coef[str_detect(names(garch_model1@fit$coef),"mu")]
ar<-garch_model1@fit$coef[str_detect(names(garch_model1@fit$coef),"ar")]
ma<-garch_model1@fit$coef[str_detect(names(garch_model1@fit$coef),"ma")]
omega<-garch_model1@fit$coef[str_detect(names(garch_model1@fit$coef),"omega")]
alpha<-garch_model1@fit$coef[str_detect(names(garch_model1@fit$coef),"alpha")]
beta<-garch_model1@fit$coef[str_detect(names(garch_model1@fit$coef),"beta")]
shape<-garch_model1@fit$coef[str_detect(names(garch_model1@fit$coef),"shape")]


lista<-list(mu=mu,ar=ar,ma=ma,omega=omega,alpha=alpha,beta=beta,shape=shape)
set.seed(12345)
data_sim=garchSim(spec=garchSpec(model=lista), n=500)

par(mfrow = c(1,1))
plot(1:500,z,type="l", ylim=c(-0.2,0.2),main="Data=Z")
grid(col = "dark grey",lwd = 1.7)
# lines(1:500,data_sim,col="red")
plot(1:500,data_sim,type="l",ylim=c(-0.2,0.2),main="Simulated data")
grid(col = "dark grey",lwd = 1.7)
par(mfrow = c(1,1))
```

2.6

```
ahead<-45
prediktion<-predict(garch_model1,n.ahead = ahead,conf=0.95,plot=TRUE)


tillbaka<-544
kolla<-(545-tillbaka):545
```

```
true_values<-x[(length(z)+1):length(x)]

plot(NULL, xlim=c(min(kolla),max(kolla)), ylim=c(-0.25,0.25), main="diff(ln(Oil data)): All data + forca
lines(kolla,z[kolla])
lines(kolla,garch_model1@fitted[kolla],col="red")
lines(501:545,prediktion$meanForecast,col="green")
lines(501:545,prediktion$lowerInterval,col="blue")
lines(501:545,prediktion$upperInterval,col="blue")
lines(501:(length(true_values)+500),true_values,col="dark red")
grid(col = "dark grey",lwd = 1.7)


tillbaka<-200
kolla<-(500-tillbaka+45):545
true_values<-x[(length(z)+1):length(x)]

plot(NULL, xlim=c(min(kolla),max(kolla)), ylim=c(-0.25,0.25), main="diff(ln(Oil data)): last 200 obs +
lines(kolla,z[kolla])
lines(kolla,garch_model1@fitted[kolla],col="red")
lines(501:545,prediktion$meanForecast,col="green")
lines(501:545,prediktion$lowerInterval,col="blue")
lines(501:545,prediktion$upperInterval,col="blue")
lines(501:(length(true_values)+500),true_values,col="dark red")
grid(col = "dark grey",lwd = 1.7)
```