457_A4

Jiayue Wu

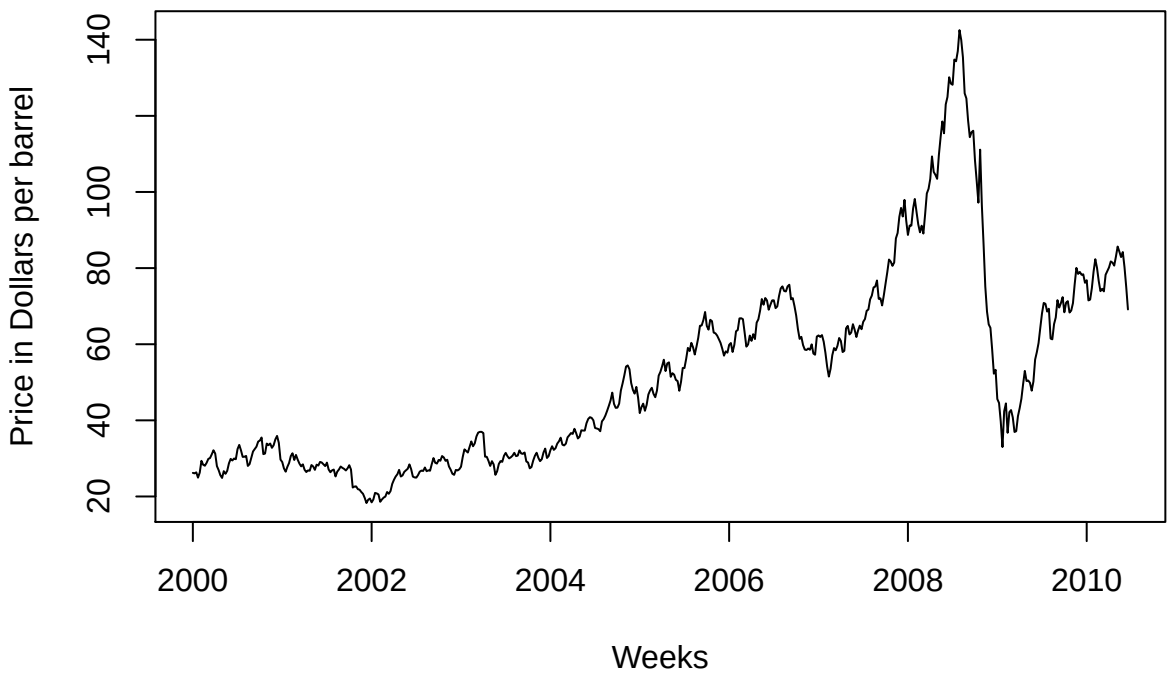
4/1/2021

```
library("astsa")
```

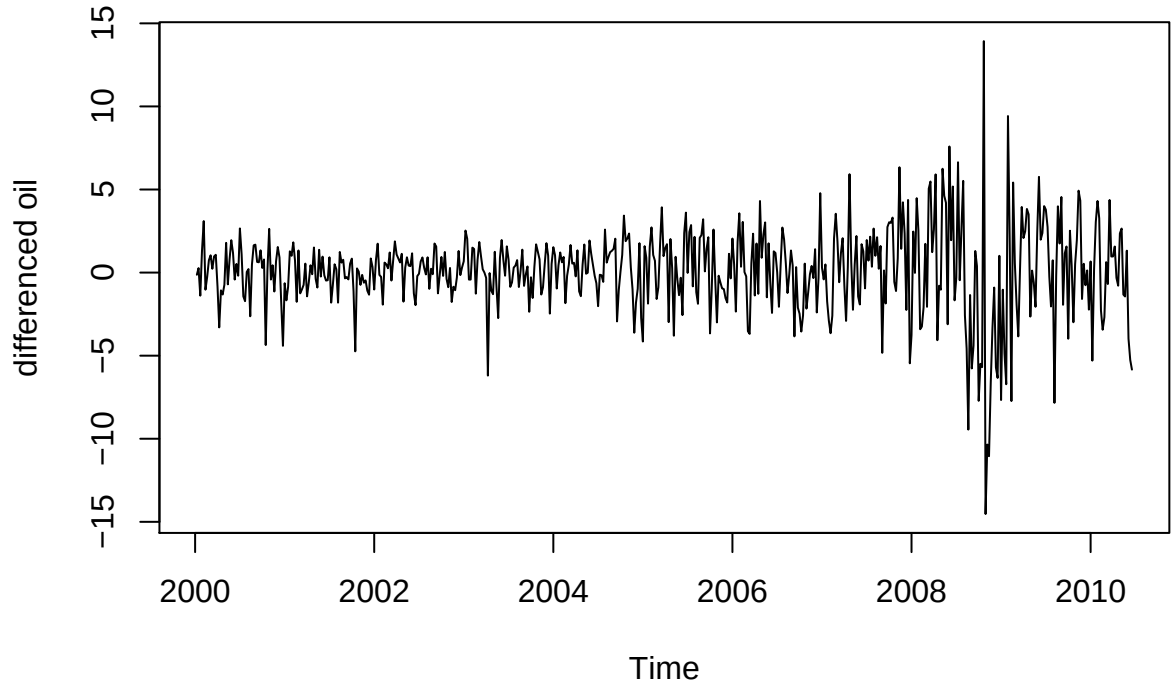
```
## Warning: package 'astsa' was built under R version 3.6.3
```

Q(1)a:

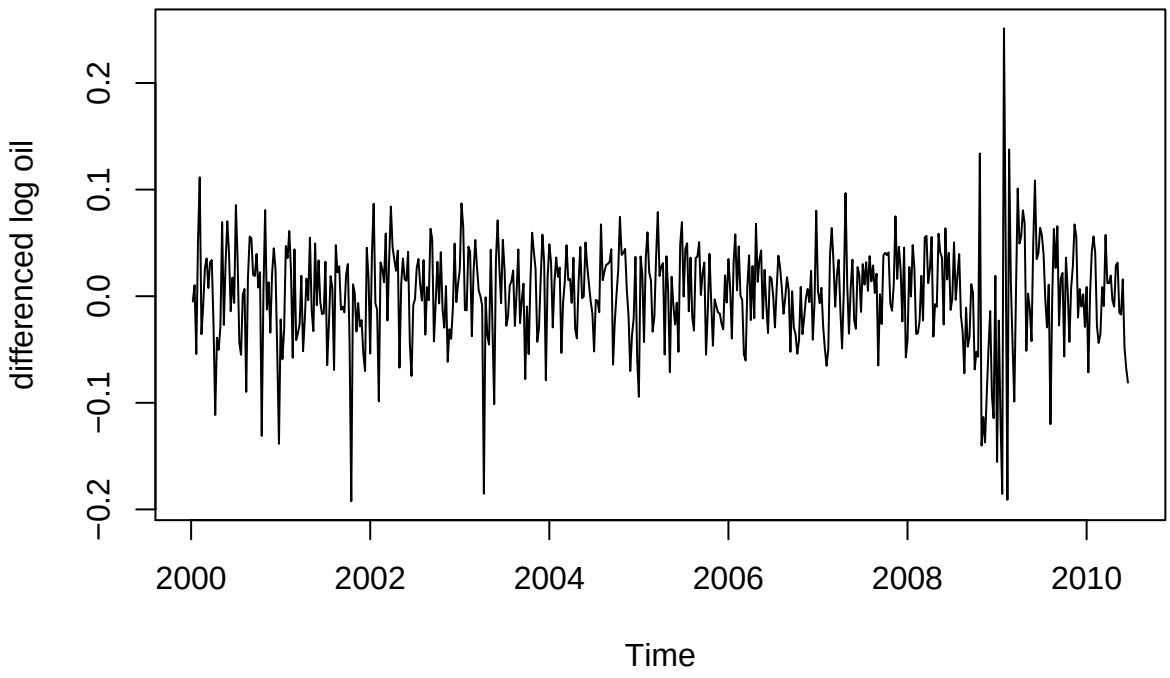
```
plot(oil, xlab = "Weeks", ylab = "Price in Dollars per barrel") # have upward pattern
```



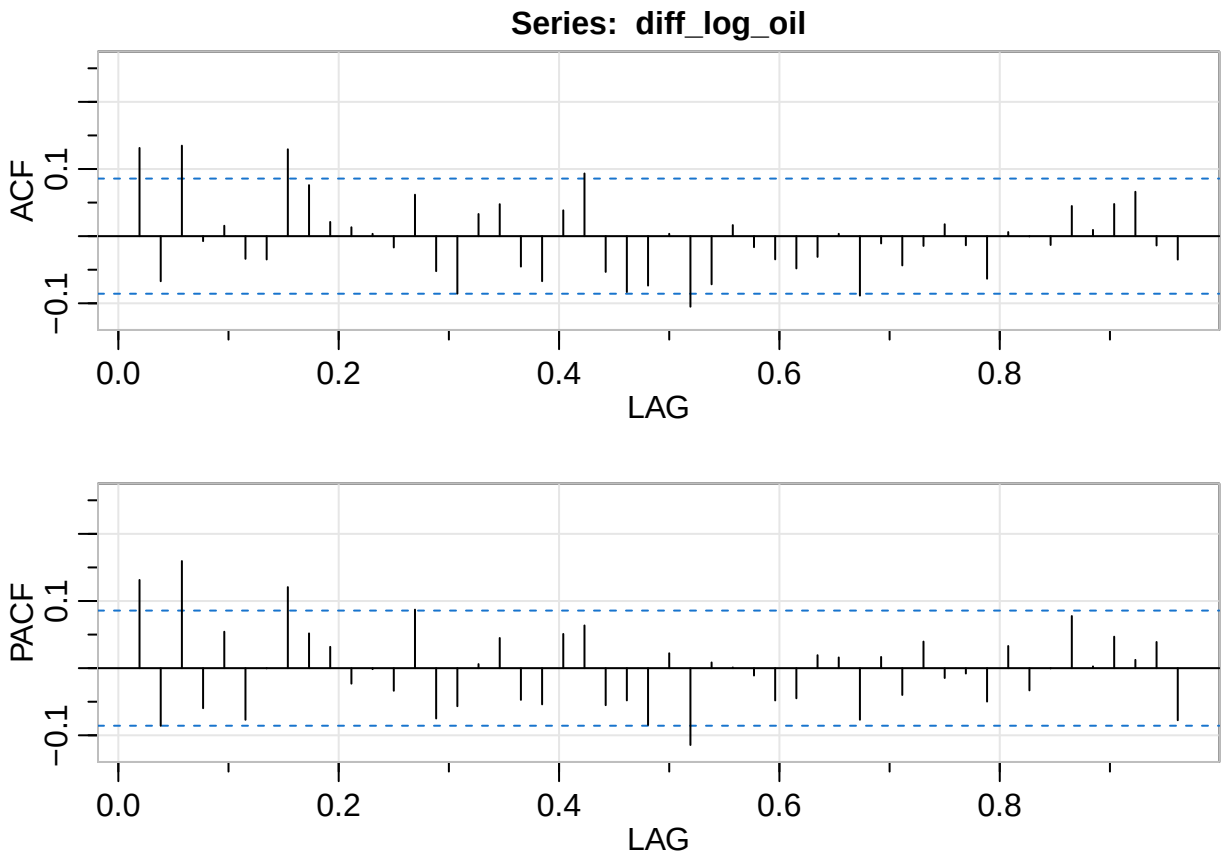
```
plot(diff(oil), ylab = "differenced oil") # get mean = 0, while not constant variance
```




```
plot(diff(log(oil)), ylab = "differenced log oil") # get mean = 0 and constant variance
```




```
diff_log_oil <- diff(log(oil)) # the transformed data: difference and log data oil
acf2(diff_log_oil, 50) # plot the ACF and PACF of transformed data.
```




```
##      [,1]  [,2] [,3]  [,4] [,5]  [,6]  [,7] [,8] [,9] [,10] [,11] [,12] [,13]
## ACF  0.13 -0.07 0.13 -0.01 0.02 -0.03 -0.03 0.13 0.08  0.02  0.01     0 -0.02
## PACF 0.13 -0.09 0.16 -0.06 0.05 -0.08  0.00 0.12 0.05  0.03 -0.02     0 -0.03
##      [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22] [,23] [,24] [,25]
## ACF   0.06 -0.05 -0.09  0.03  0.05 -0.05 -0.07  0.04  0.09 -0.05 -0.08 -0.07
## PACF  0.09 -0.07 -0.06  0.01  0.04 -0.05 -0.05  0.05  0.06 -0.06 -0.05 -0.08
##      [,26] [,27] [,28] [,29] [,30] [,31] [,32] [,33] [,34] [,35] [,36] [,37]
## ACF   0.00 -0.11 -0.07  0.02 -0.02 -0.03 -0.05 -0.03  0.00 -0.09 -0.01 -0.04
## PACF  0.02 -0.11  0.01  0.00 -0.01 -0.05 -0.04  0.02  0.02 -0.08  0.02 -0.04
##      [,38] [,39] [,40] [,41] [,42] [,43] [,44] [,45] [,46] [,47] [,48] [,49]
## ACF  -0.01  0.02 -0.01 -0.06  0.01  0.00 -0.01  0.04  0.01  0.05  0.07 -0.01
## PACF  0.04 -0.01 -0.01 -0.05  0.03 -0.03  0.00  0.08  0.00  0.05  0.01  0.04
##      [,50]
## ACF  -0.03
## PACF -0.08
```

```
# from ACF plot -> MA(3); from PACF plot -> AR(3)
# Proposed ARIMA(3, 0, 3); ARIMA(3, 0, 0); ARIMA(0, 0, 3)
```

From the graph of data oil, we notice that it has an upward pattern, which is not stationary, i.e., not satisfy the white noise assumption. Hence, I make a difference and log transformation for the data oil. And it shows an approximately zero mean and constant variance. I use the difference and log transformation oil as the data after transformation.

Then from the graph of ACF and PACF of difference for transformed data, we can find that there are 3 lags after ACF cutoff, which indicates a MA(3) model; there are 3 lags after PACF cutoff, which means there a

AR(3) model.

Hence, I proposed three possible models: ARIMA(3, 0, 3); ARIMA(3, 0, 0) and ARIMA(0, 0, 3) for transformed oil dara. In other words, I proposed three possible models: ARIMA(3, 1, 3); ARIMA(3, 1, 0) and ARIMA(0, 1, 3) for original globtemp data since I have done a difference for the oil data at the beginning for the data with ACF and PACF.

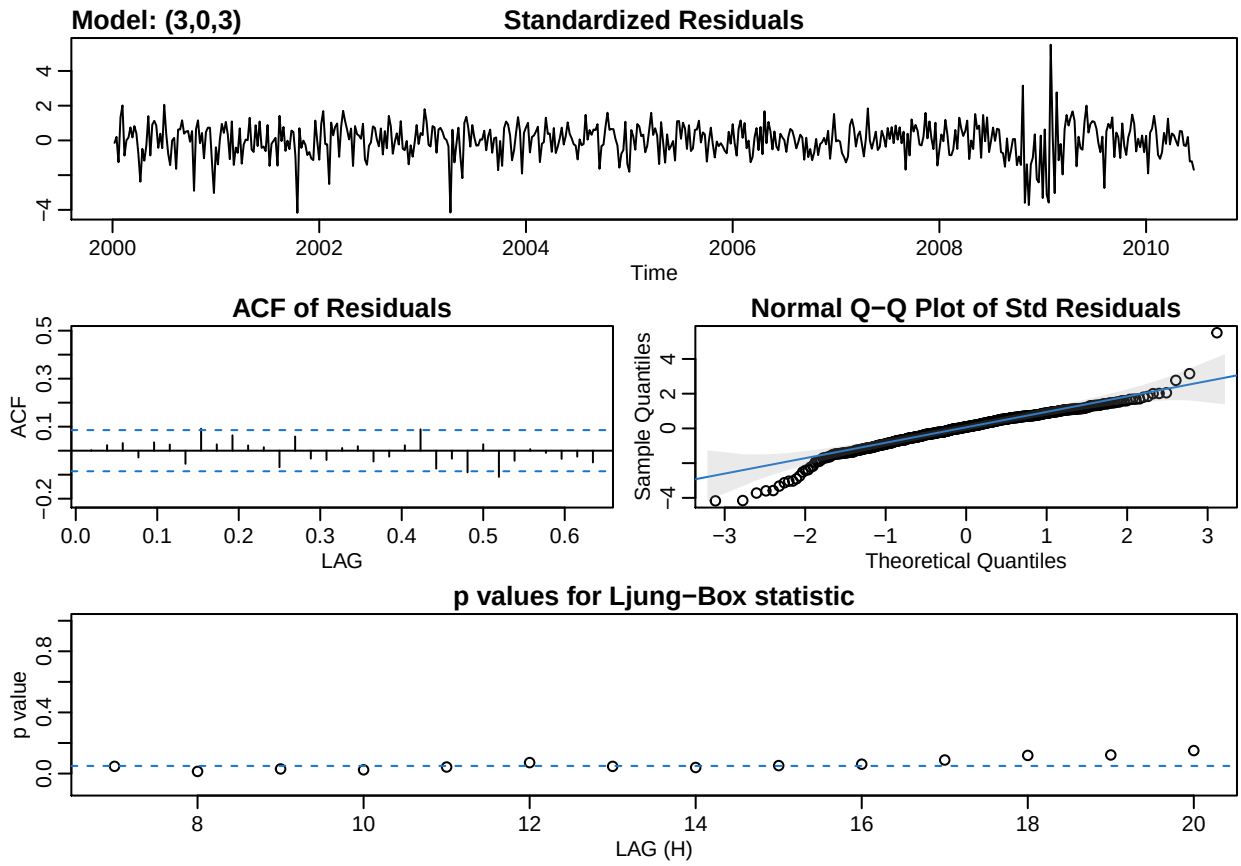
Q(1)b:

Next, I will compare the three proposed models I selected on Q(a).

```
mod303 <- sarima(diff_log_oil, 3, 0, 3)
```

```
## initial  value -3.057088
## iter   2 value -3.075783
## iter   3 value -3.084905
## iter   4 value -3.085417
## iter   5 value -3.085558
## iter   6 value -3.085689
## iter   7 value -3.086109
## iter   8 value -3.086143
## iter   9 value -3.086195
## iter  10 value -3.086538
## iter  11 value -3.086807
## iter  12 value -3.086973
## iter  13 value -3.087073
## iter  14 value -3.087100
## iter  15 value -3.087144
## iter  16 value -3.087867
## iter  17 value -3.087878
## iter  18 value -3.087995
## iter  19 value -3.088122
## iter  20 value -3.088562
## iter  21 value -3.088951
## iter  22 value -3.089486
## iter  23 value -3.089783
## iter  24 value -3.089938
## iter  25 value -3.090155
## iter  26 value -3.090701
## iter  27 value -3.090823
## iter  28 value -3.090850
## iter  29 value -3.090883
## iter  30 value -3.090914
## iter  31 value -3.090960
## iter  32 value -3.090972
## iter  33 value -3.090974
## iter  34 value -3.090980
## iter  35 value -3.090989
## iter  36 value -3.091003
## iter  37 value -3.091005
## iter  38 value -3.091006
## iter  39 value -3.091008
```

```
## iter  40 value -3.091008
## iter  41 value -3.091008
## iter  41 value -3.091008
## iter  41 value -3.091008
## final  value -3.091008
## converged
## initial  value -3.091639
## iter   2 value -3.091651
## iter   3 value -3.091679
## iter   4 value -3.091698
## iter   5 value -3.091729
## iter   6 value -3.091774
## iter   7 value -3.091791
## iter   8 value -3.091848
## iter   9 value -3.091933
## iter  10 value -3.092009
## iter  11 value -3.092051
## iter  12 value -3.092060
## iter  13 value -3.092073
## iter  14 value -3.092098
## iter  15 value -3.092108
## iter  16 value -3.092111
## iter  17 value -3.092112
## iter  18 value -3.092113
## iter  19 value -3.092114
## iter  20 value -3.092116
## iter  21 value -3.092116
## iter  22 value -3.092116
## iter  23 value -3.092116
## iter  24 value -3.092117
## iter  25 value -3.092117
## iter  26 value -3.092117
## iter  27 value -3.092117
## iter  27 value -3.092117
## iter  27 value -3.092117
## final  value -3.092117
## converged
```




```
# ARIMA(3, 0, 3) for transformed data; ARIMA(3,1,3) for original data.
```

The estimate the parameters, and test the significance of the parameter estimates for ARIMA(3, 0, 3) for transformed data are listed below:

```
mod303$ttable # The estimation and significance of the parameters for ARIMA(3, 0, 3)
```

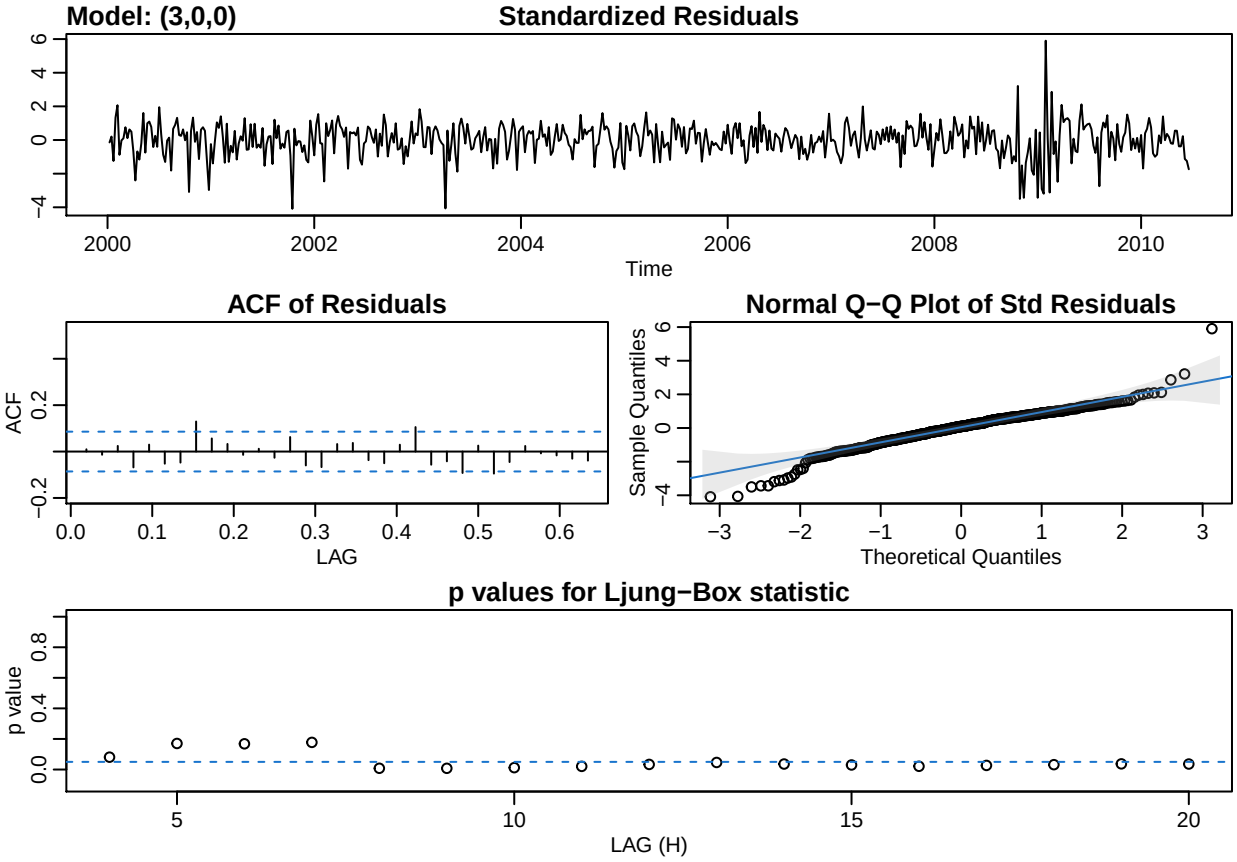
```
##       Estimate     SE t.value p.value
## ar1     0.0674 0.1096  0.6156  0.5384
## ar2    -0.6295 0.0924 -6.8121  0.0000
## ar3    -0.3632 0.1232 -2.9478  0.0033
## ma1     0.0964 0.0926  1.0415  0.2981
## ma2     0.5091 0.0883  5.7653  0.0000
## ma3     0.5992 0.1049  5.7115  0.0000
## xmean   0.0017 0.0022  0.7832  0.4338
```

ar2, ar3, ma2, ma3 are statistically significant under significance level $\alpha = 0.05$

```
mod300 <- sarima(diff_log_oil, 3, 0, 0)
```

```
## initial  value -3.057088
## iter   2 value -3.081473
## iter   3 value -3.082699
## iter   4 value -3.082712
```

```
## iter   5 value -3.082712
## iter   6 value -3.082712
## iter   7 value -3.082712
## iter   7 value -3.082712
## iter   7 value -3.082712
## final  value -3.082712
## converged
## initial  value -3.083886
## iter   2 value -3.083887
## iter   3 value -3.083888
## iter   4 value -3.083890
## iter   4 value -3.083890
## iter   4 value -3.083890
## final  value -3.083890
## converged
```




```
# ARIMA(3, 0, 0) for transformed data; ARIMA(3,1,0) for original data.
```

The estimate the parameters, and test the significance of the parameter estimates for ARIMA(3, 0, 0) for transformed data are listed below:

```
mod300$ttable # The estimation and significance of the parameters for ARIMA(3, 0, 0)
```

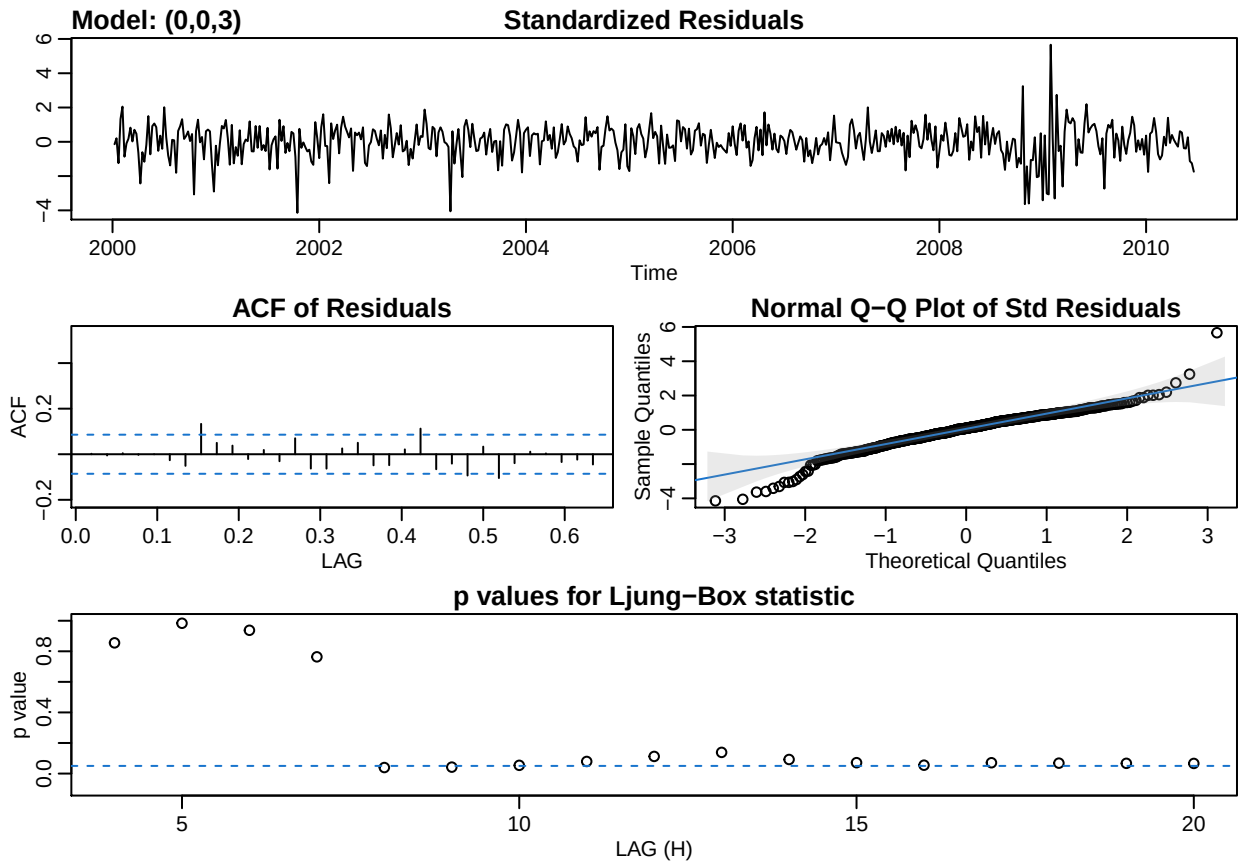
```
##       Estimate     SE t.value p.value
```

```
## ar1     0.1571 0.0424  3.7059  0.0002
## ar2    -0.1078 0.0427 -2.5262  0.0118
## ar3     0.1608 0.0425  3.7853  0.0002
## xmean   0.0017 0.0025  0.6817  0.4957
```

ar1, ar2, ma3 all statistically significant under significance level $\alpha = 0.05$

```
mod003 <- sarima(diff_log_oil, 0, 0, 3)
```

```
## initial  value -3.058495
## iter   2 value -3.086110
## iter   3 value -3.086980
## iter   4 value -3.087501
## iter   5 value -3.087521
## iter   6 value -3.087521
## iter   7 value -3.087522
## iter   8 value -3.087522
## iter   9 value -3.087522
## iter   9 value -3.087522
## iter   9 value -3.087522
## final  value -3.087522
## converged
## initial  value -3.087448
## iter   2 value -3.087448
## iter   3 value -3.087449
## iter   3 value -3.087449
## iter   3 value -3.087449
## final  value -3.087449
## converged
```




```
# ARIMA(0, 0, 3) for transformed data; ARIMA(0,1,3) for original data.
```

The estimate the parameters, and test the significance of the parameter estimates for ARIMA(0, 0, 3) for transformed data are listed below:

```
mod003$ttable # The estimation and significance of the parameters for ARIMA(0, 0, 3)
```

```
##       Estimate     SE t.value p.value
## ma1     0.1688 0.0424  3.9820  0.0001
## ma2    -0.0900 0.0425 -2.1167  0.0347
## ma3     0.1447 0.0430  3.3679  0.0008
## xmean   0.0017 0.0024  0.7147  0.4751
```

ma2, ma2,ma3 are all statistically significant under significance level $\alpha = 0.05$

Q(1)c:

Next I will compare the behavior of ARIMA(3, 0, 3); ARIMA(3, 0, 0) and ARIMA(0, 0, 3) by the plots presented on Q(b).

From the Standardized residuals plots, we find that all these three models behave like white noise, with approximately mean = 0 and constant variance. From the ACF of Residuals plots, we find that almost all of these three models are within the blue line, i.e., all the lags are in significance. From the Normal QQ Plot

of Standard Residuals plots, we find that all of these three models have similar behaviors: most of points lie on the straight line whlie with a bit tail off for the extreme points. However, for the p-value for Ljung-Box statistic plots, we find that model ARIMA(3, 0, 3) and ARIMA(3, 0, 0) only have approximately 3 points above the significant line, i.e., most of the points they are in significance, which means there are statistically significant evidence to show that they are correlated. While the last model ARIMA(0, 0, 3) has better behavior than the other two, it has more than 14 points are above the significant line, which means most of the points indicate there is no such a statistically significant evidence to show that they are correlated.

Furthermore, I checked the AIC and BIC for these three proposed models:

```
AIC_BIC <- data.frame("model" = c("303", "300", "003"),
                      "AIC" = c(mod303$AIC,mod300$AIC,mod003$AIC),
                      "BIC" = c(mod303$BIC,mod300$BIC,mod003$BIC)
                      )
AIC_BIC
```

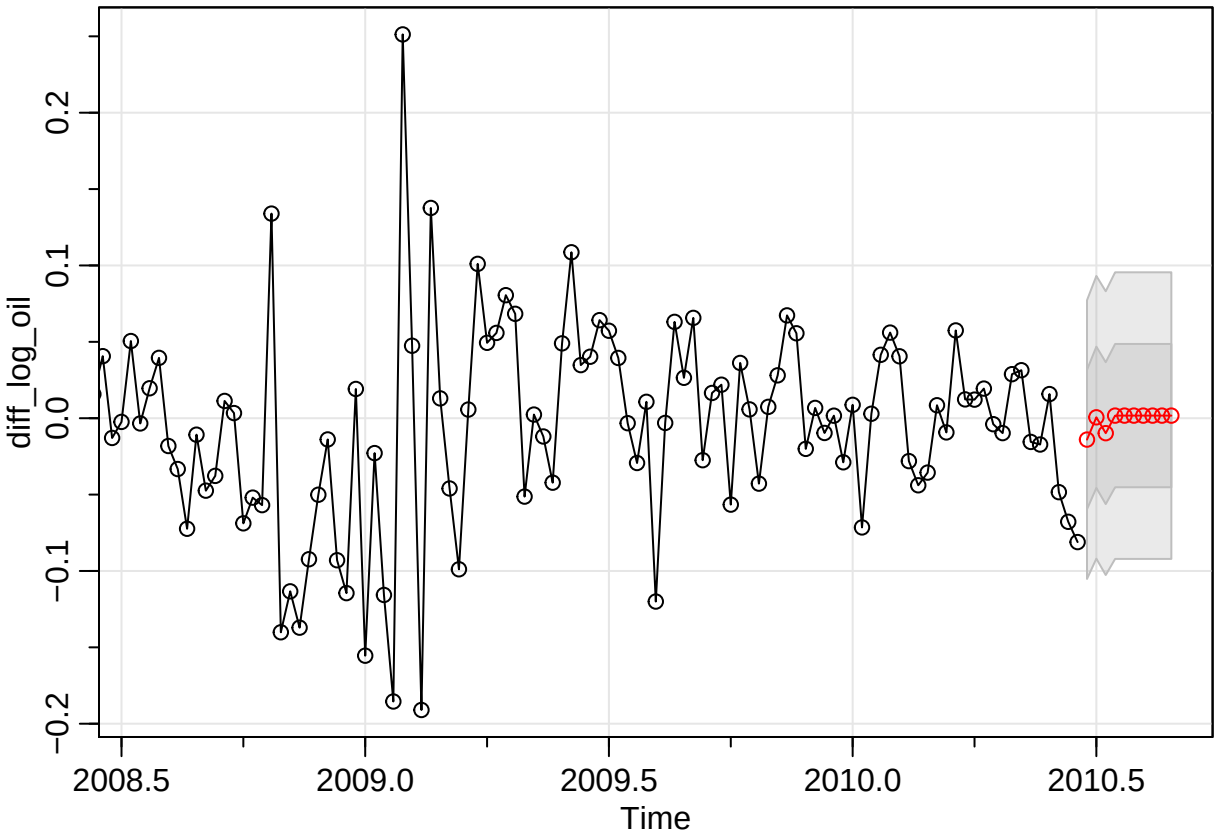
```
##   model       AIC       BIC
## 1   303 -3.316945 -3.253725
## 2   300 -3.311520 -3.272007
## 3   003 -3.318638 -3.279125
```

We prefer the AIC and BIC as small as possible here, and we can find that model ARIMA(0, 0, 3) has least AIC and BIC even with small difference compared to the other two.

Hence, by comparing the behaviors of Ljung-Box statistic plots and AIC_BIC, I will choose the model ARIMA(0, 0, 3). More precisely, model ARIMA(0, 0, 3) for transformed oil data; model ARIMA(0, 1, 3) for original oil data.

Q(1)d:

```
# using ARIMA(0, 0, 3), forecast the next 10 years.
model <- sarima.for(diff_log_oil, 10, 0, 0, 3)
```




```
U <- model$pred + qnorm(0.975)*model$se
L <- model$pred - qnorm(0.975)*model$se
```

```
PI <- data.frame("Predicted_year" = 1:10, "Lower_bound" = L, "Upper_bound" = U)
PI
```

```
##    Predicted_year Lower_bound Upper_bound
## 1               1 -0.10339136  0.07540630
## 2               2 -0.09007321  0.09125373
## 3               3 -0.10081301  0.08122581
## 4               4 -0.09022437  0.09364325
## 5               5 -0.09022437  0.09364325
## 6               6 -0.09022437  0.09364325
## 7               7 -0.09022437  0.09364325
## 8               8 -0.09022437  0.09364325
## 9               9 -0.09022437  0.09364325
## 10             10 -0.09022437  0.09364325
```

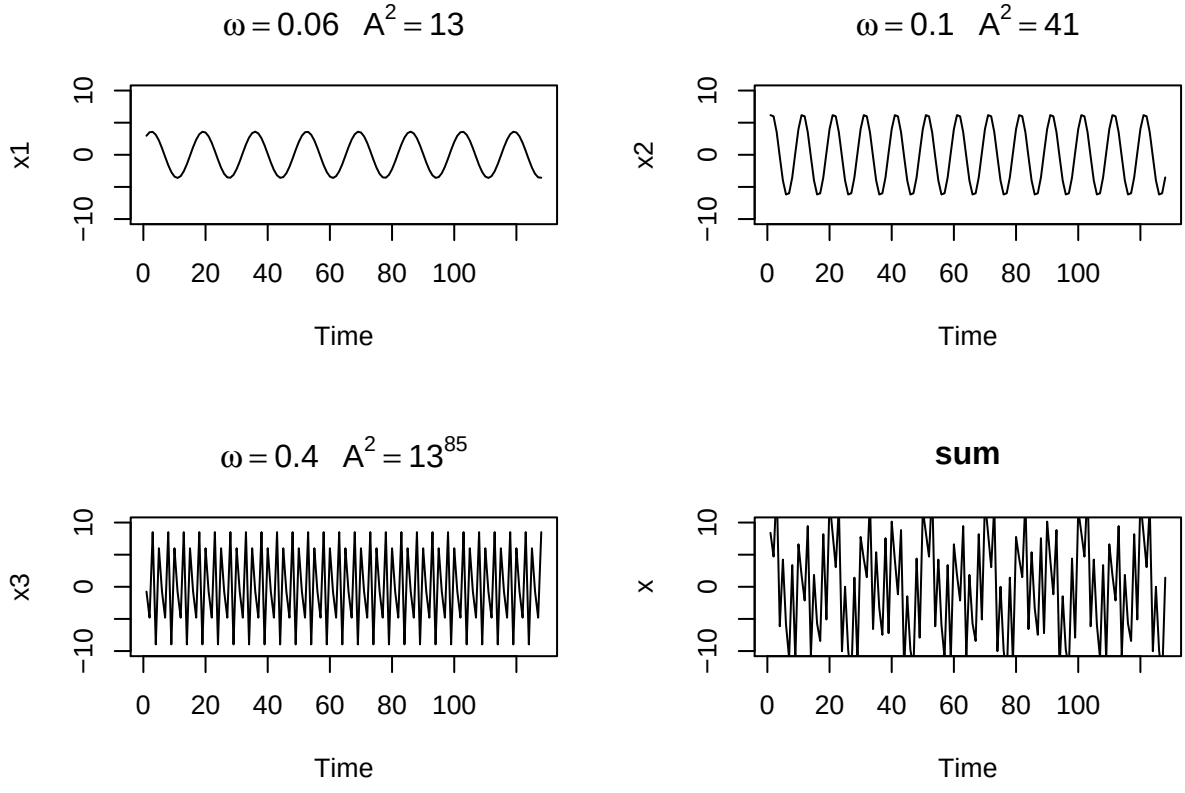
Q(1)e:

As for the limitation of this model, what I conclude is that it is hard to fit the extreme points and the prediction maybe not pretty good due to its too simple and the Normality assumption. For the extreme points, we know the ARIMA model is fitted by the seasonality and trends. However, the extreme points may be out of the general trend of ARIMA model. Hence, it could be hard to fit the extreme points.

For the prediction behavior: since the ARIMA model is a linear regression model, which means it assume that the dependent variable and the independent variables are only linear. However, if the actual data is not linearly separable but more complex, such as containing some polynomial pattern, the model may be incorrect. Furthermore, from the normal QQ-plot of our final model, we can find that the points are a bit tail off, which indicates that the normality assumption fit not very well on the model ARIMA(0, 1, 3) we choose, and it may result in the prediction incorrect somehow.

Q(2)a:

```
n1 <- 128 # set the sample size n1
x1 <- 2*cos(2*pi*0.06*1:n1) +  3*sin(2*pi*0.06*1:n1) # set the formula of x1
x2 <- 4*cos(2*pi*0.1*1:n1) +  5*sin(2*pi*0.1*1:n1) # set the formula of x2
x3 <- 6*cos(2*pi*0.4*1:n1) +  7*sin(2*pi*0.4*1:n1) # set the formula of x2
x <- x1+x2+x3 # set the formula of sum x
par(mfrow = c(2,2))
plot.ts(x1, ylim=c(-10,10), main =expression(omega == 0.06~~~A^2==13)) # x1's plot
plot.ts(x2, ylim=c(-10,10), main =expression(omega == 0.1~~~A^2==41)) # x2's plot
plot.ts(x3, ylim=c(-10,10), main =expression(omega == 0.4~~~A^2==13^85)) # x3's plot
plot.ts(x, ylim=c(-10,10), main = "sum") # sum x's plot
```

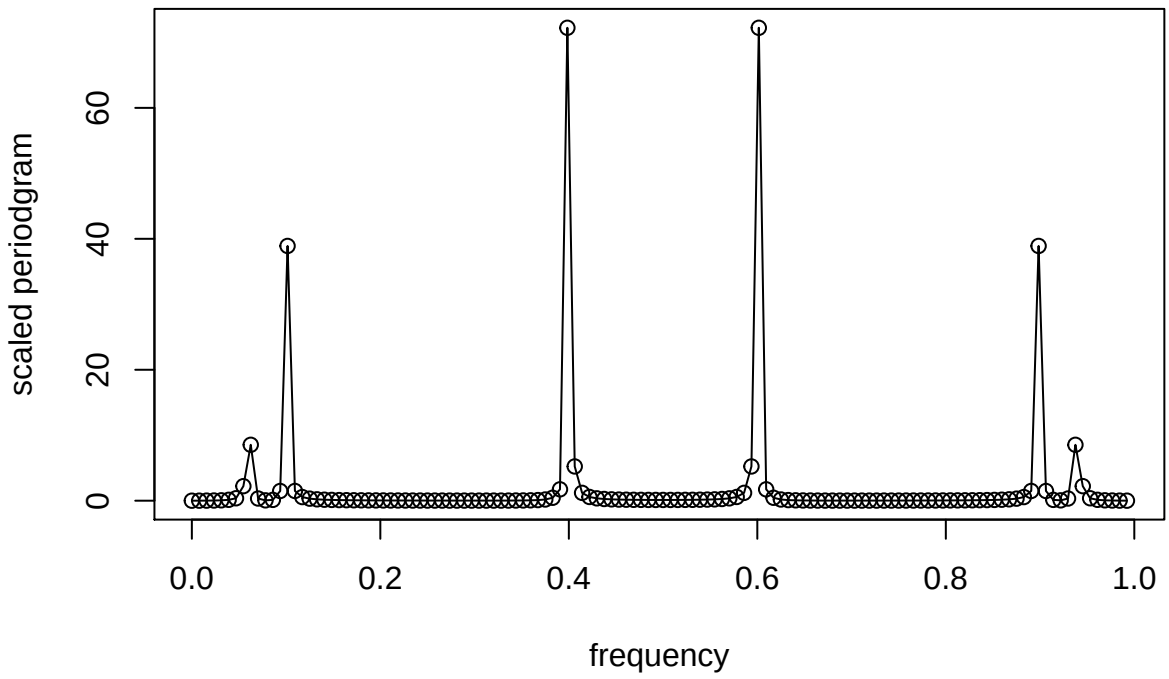


The main difference of Example 4.1 and Q(2)a is the difference between sample size, which shows us a longer time in each graph, i.e., the period in Example 4.1 is 100 while the period in Q(2)a is 128. However, the plots' behaviors in Example 4.1 and Q(2)a are pretty similar, they are almost the same, with the even same corresponding frequencies, squared amplitude and pattern.

Q(2)b:

Below is the scaled periodogram of the data, x_t, stimulated in Q(2)a.

```
P <- Mod(2*fft(x)/n1)^2
Fr <- 0:127/n1
plot(Fr, P, type = "o", xlab = "frequency", ylab = "scaled periodgram")
```

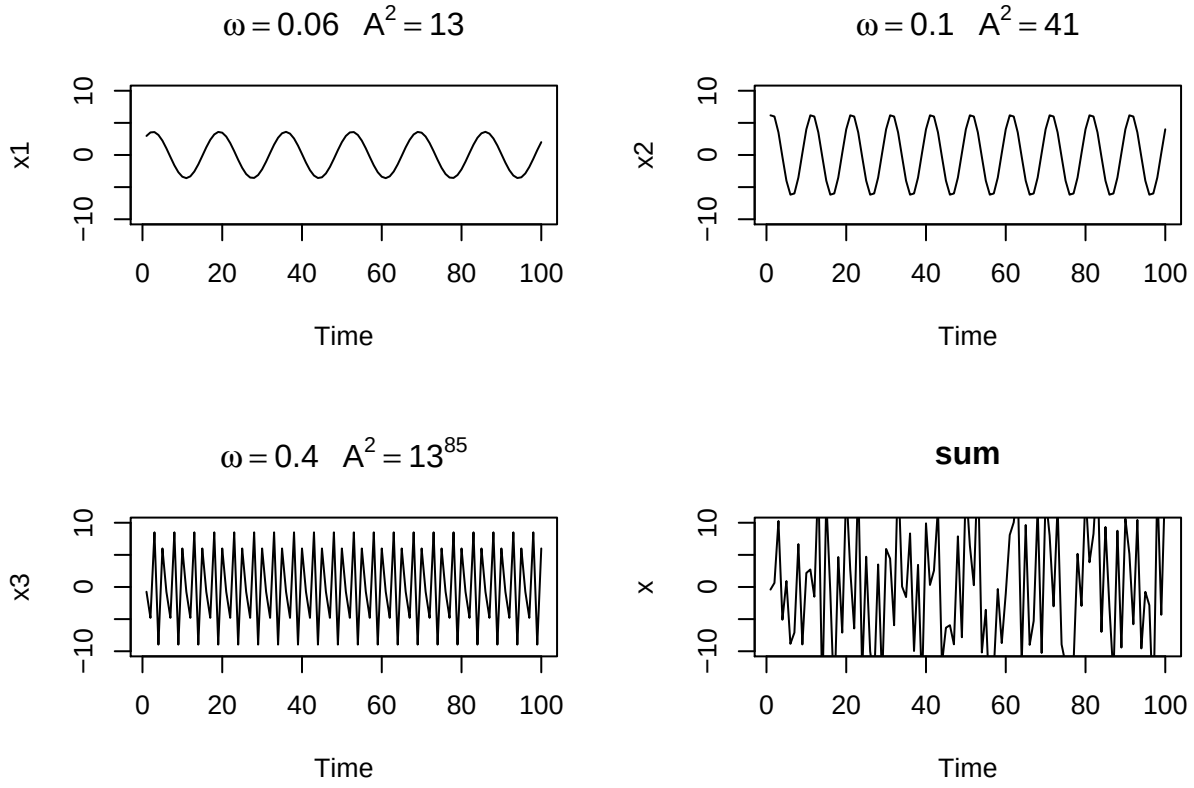


Since P(j/n) = P(1-j/n), j = 0,1,. . . ,n-1, so there is a mirroring effect at the folding frequency of 1/2. And the periodogram is typically not plotted for frequencies higher than the folding frequency. Moreover, we can find that the heights of the scaled periodogram shown in this figure also at peak when P(0.06) = P(0.94); P(0.1) = P(0.9); P(0.4) = P(0.6). However, what Example 4.2 shows is P(j/n) = 0 for values others than the peak points, while what we plotted shows there are still some other points near the peak frequency with periodogran not equal to 0. Furthermore, we have a little lower amplitude here than in Example 4.2, especially by comparing the scale value of scaled periodgram and comparing the value of P(0.1) here and in Example 4.2.

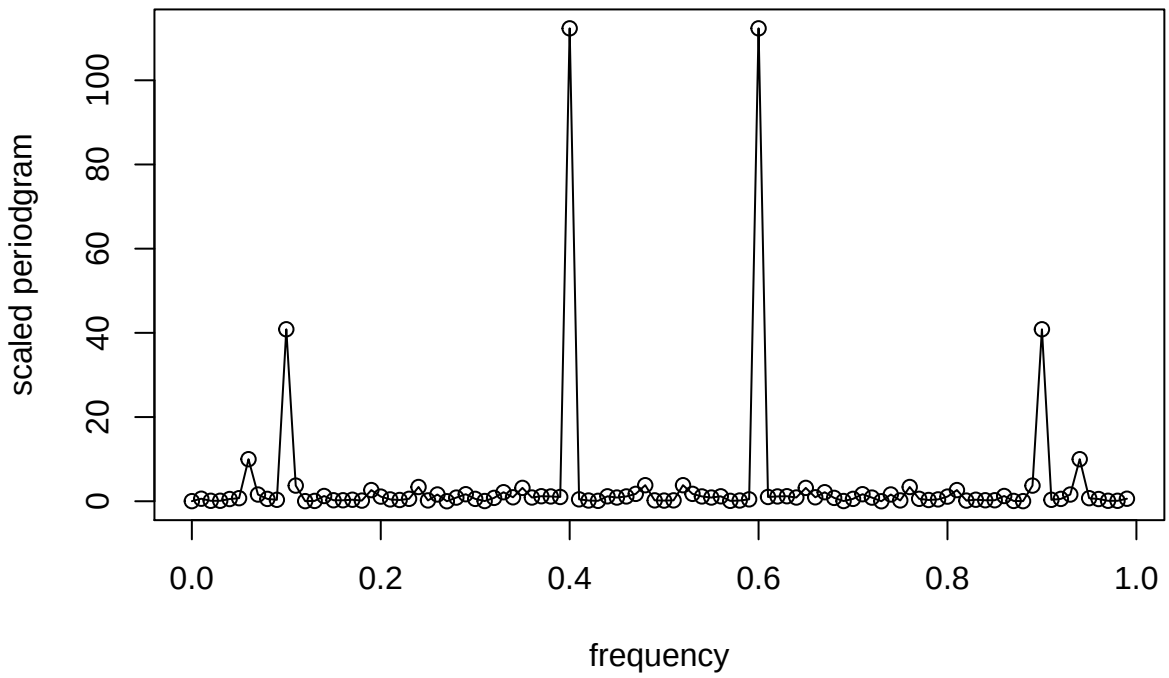
Q(2)c:

```
n <- 100 # set the new sample size n
set.seed(1004768165) # set a seed to always get the same white noise.
w <- rnorm(n, 0, 5) # select the white noise
x1 <- 2*cos(2*pi*0.06*1:n) +  3*sin(2*pi*0.06*1:n) # set the formula of x1
```

```r
x2 <- 4*cos(2*pi*0.1*1:n) +  5*sin(2*pi*0.1*1:n) # set the formula of x2
x3 <- 6*cos(2*pi*0.4*1:n) +  7*sin(2*pi*0.4*1:n) # set the formula of x3
x <- x1+x2+x3+w # set the formula of sum x with white noise.
par(mfrow = c(2,2))
plot.ts(x1, ylim=c(-10,10), main =expression(omega == 0.06~~~A^2==13)) # x1's plot
plot.ts(x2, ylim=c(-10,10), main =expression(omega == 0.1~~~A^2==41)) # x2's plot
plot.ts(x3, ylim=c(-10,10), main =expression(omega == 0.4~~~A^2==13^85)) # x3's plot
plot.ts(x, ylim=c(-10,10), main = "sum") # sum x with white noise's plot
```



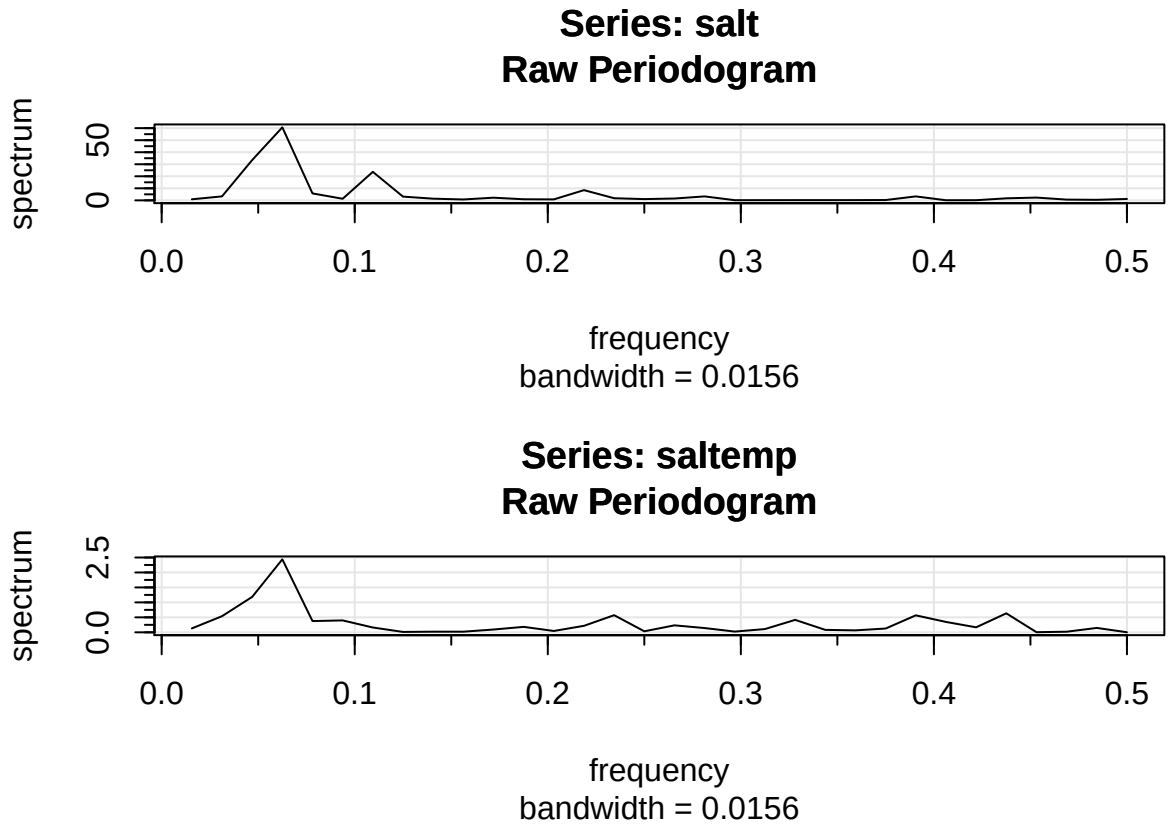

```r
# Plot the periodogran of x
par(mfrow = c(1,1))
P <- Mod(2*fft(x)/n)^2
Fr <- 0:99/n
plot(Fr, P, type = "o", xlab = "frequency", ylab = "scaled periodgram")
```



Here I stimulated and plot the data adding white noise, and also plot the periodogram of x_t. Generally speaking, after adding the white noise, the plot of x_t still behaves almost the same as before, with zero mean and constant variance. The periodogram also has similar behavior as before. Although with more points not equal to 0, the frequency reach the peaks and the corresponding value are almost the same.

Q(3)a:

```
data(salt) # data of salt
data(saltemp) # data of saltemp
par(mfrow = c(2,1))
# perform the spectral analyses
salt.per <- mvspec(salt, log = "no") # spectral analyses on salt
saltemp.per <- mvspec(saltemp, log = "no") # spectral analyses on saltemp
```



Here I perform the separate spectral analyses on the two series first.

```
# Here I order by the spectrum of salt in decreasing order, i.e., the largest
# spectrum is on the first and the smallest spectrum is listed on the last.
spec_salt <- salt.per$details[order(salt.per$details[,3], decreasing = TRUE),]
# The first column is frequency, and I constructed a data frame to list the first three.
data.frame(
  "order" = c("first", "second", "third"),
  "frequency" = c(spec_salt[1,][1], spec_salt[2,][1],spec_salt[3,][1]),
  "spectrum" = c(spec_salt[1,][3], spec_salt[2,][3],spec_salt[3,][3])
)
```

```
##    order frequency spectrum
## 1  first    0.0625  60.6665
## 2 second    0.0469  33.4859
## 3  third    0.1094  23.6903
```

Hence, the first three dominant frequencies by performing separate spectral analyses for the data salt is 0.0625, 0.0469, 0.1094, with corresponding spectrum 60.6665, 33.4859, 23.6903.

```
# Here I order by the spectrum of salt in decreasing order, i.e., the largest
# spectrum is on the first and the smallest spectrum is listed on the last.
spec_saltemp <- saltemp.per$details[order(saltemp.per$details[,3], decreasing = TRUE),]
# The first column is frequency,
# and I constructed a data frame to list the first three.
data.frame(
```

```
  "order" = c("first", "second", "third"),
  "frequency" = c(spec_saltemp[1,][1], spec_saltemp[2,][1],spec_saltemp[3,][1]),
  "spectrum" = c(spec_saltemp[1,][3], spec_saltemp[2,][3],spec_saltemp[3,][3])
)
```

```
##    order frequency spectrum
## 1  first    0.0625   2.4379
## 2 second    0.0469   1.1836
## 3  third    0.4375   0.6353
```

Hence, the first three dominant frequencies by performing separate spectral analyses for the data saltemp is 0.0625, 0.0469, 0.4375, with corresponding spectrum 2.4379, 1.1836, 0.6353.

Q(3)b:

```
# CI for first dominant
dom_salt1 <- spec_salt[1,][3] # the largest spectrum in data salt
dom_saltemp1 <- spec_saltemp[1,][3] # the largest spectrum in data saltemp
x_0.025 <- qchisq(0.025, 2)
x_0.975 <- qchisq(0.975, 2)
salt_upper <- 2*dom_salt1/x_0.025 # upper bound
salt_lower <- 2*dom_salt1/x_0.975 # lower bound
saltemp_upper <- 2*dom_saltemp1/x_0.025 # upper bound
saltemp_lower <- 2*dom_saltemp1/x_0.975 # lower bound
# construct a data frame.
data.frame(
  "first" = c("salt", "saltemp"),
  "lower_bound" = c(salt_lower, saltemp_lower),
  "upper_bound" = c(salt_upper, saltemp_upper)
)
```

```
##     first lower_bound upper_bound
## 1    salt  16.4457800  2396.19876
## 2 saltemp   0.6608782    96.29191
```

Hence, the 95% confidence interval for first dominant of data salt is (16.4457800, 2396.19876). And the 95% confidence interval for first dominant of data saltemp is (0.6608782, 96.29191).

```
# CI for second dominant
dom_salt2 <- spec_salt[2,][3] # the second largest spectrum in data salt
dom_saltemp2 <- spec_saltemp[2,][3] # the second largest spectrum in data saltemp
salt_upper <- 2*dom_salt2/x_0.025 # upper bound
salt_lower <- 2*dom_salt2/x_0.975 # lower bound
saltemp_upper <- 2*dom_saltemp2/x_0.025 # upper bound
saltemp_lower <- 2*dom_saltemp2/x_0.975 # lower bound
# construct a data frame.
data.frame(
  "second" = c("salt", "saltemp"),
  "lower_bound" = c(salt_lower, saltemp_lower),
  "upper_bound" = c(salt_upper, saltemp_upper)
)
```

```
##    second lower_bound upper_bound
## 1    salt   9.0775262   1322.6224
## 2 saltemp   0.3208562     46.7497
```

Hence, the 95% confidence interval for second dominant of data salt is (9.0775262, 1322.6224). And the 95% confidence interval for second dominant of data saltemp is (0.3208562, 46.7497).

```
# CI for third dominant
dom_salt3 <- spec_salt[3,][3] # the third largest spectrum in data salt
dom_saltemp3 <- spec_saltemp[3,][3] # the third largest spectrum in data saltemp
salt_upper <- 2*dom_salt3/x_0.025 # upper bound
salt_lower <- 2*dom_salt3/x_0.975 # lower bound
saltemp_upper <- 2*dom_saltemp3/x_0.025 # upper bound
saltemp_lower <- 2*dom_saltemp3/x_0.975 # lower bound
# construct a data frame.
data.frame(
  "third" = c("salt", "saltemp"),
  "lower_bound" = c(salt_lower, saltemp_lower),
  "upper_bound" = c(salt_upper, saltemp_upper)
)
```

```
##     third lower_bound upper_bound
## 1    salt   6.4220857   935.71687
## 2 saltemp   0.1722203    25.09301
```

Hence, the 95% confidence interval for third dominant of data salt is (6.4220857, 935.71687). And the 95% confidence interval for third dominant of data saltemp is (0.1722203, 25.09301).

I have listed the 95% confidence interval for the three dominant frequencies in part(a).

However, the CI is somehow too large that provide little constructive suggestion on predicting the true data.