Exam 3

Michael Pena

2024-06-13

Question 1

We found previously that

$$\nabla_4 X_t = 4\beta_1 + Z_t + Z_{t-4}$$

for the seasonal differencing, we find that

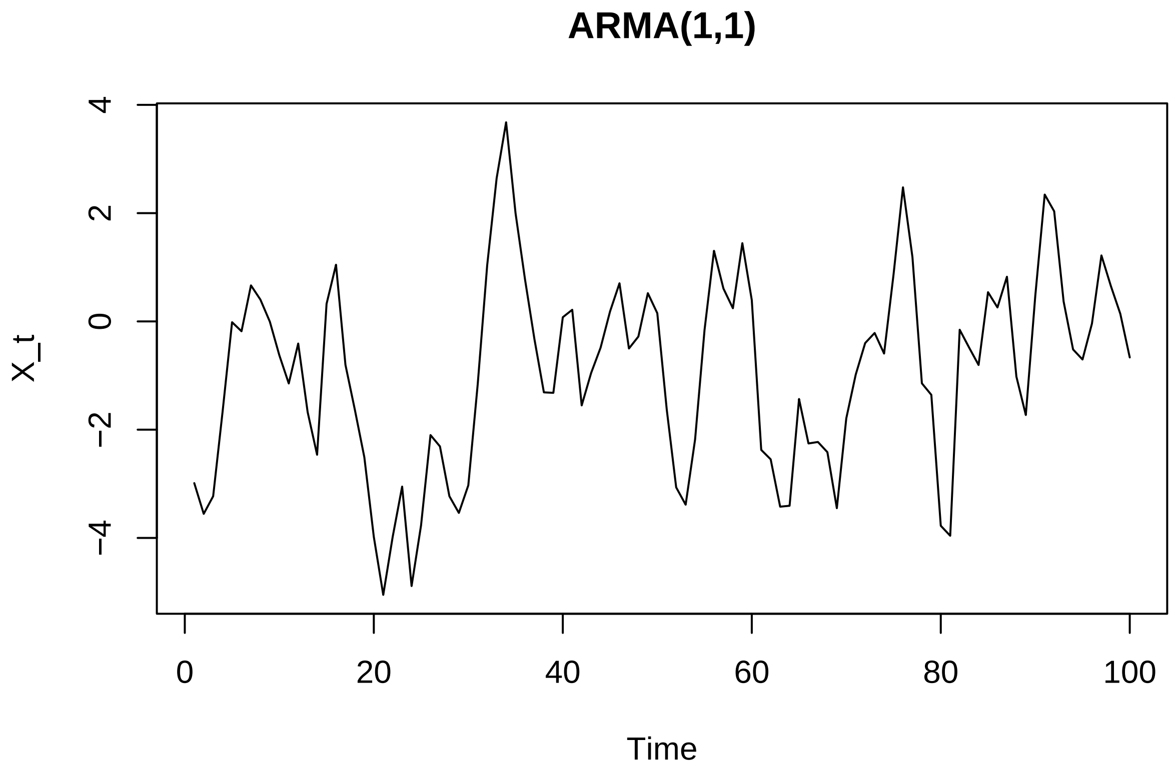
$$\nabla_4 \nabla X_t = -Z_{t-1} - Z_{t-4} + Z_{t-5}$$

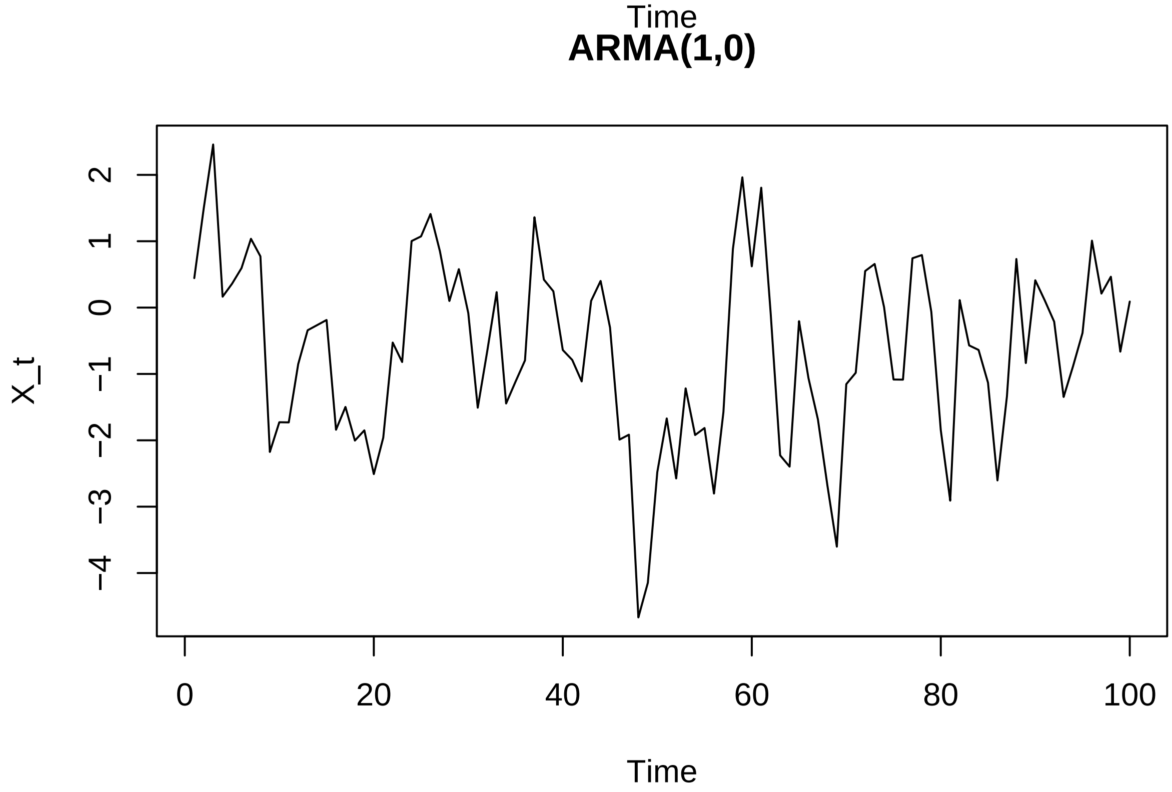
One has more parameters than the other. essentially in the end we want to go with the model that works with less parameters and does the same job in the end; less parameters means less work for the same result (we would pick the seasonal differenced pathway).

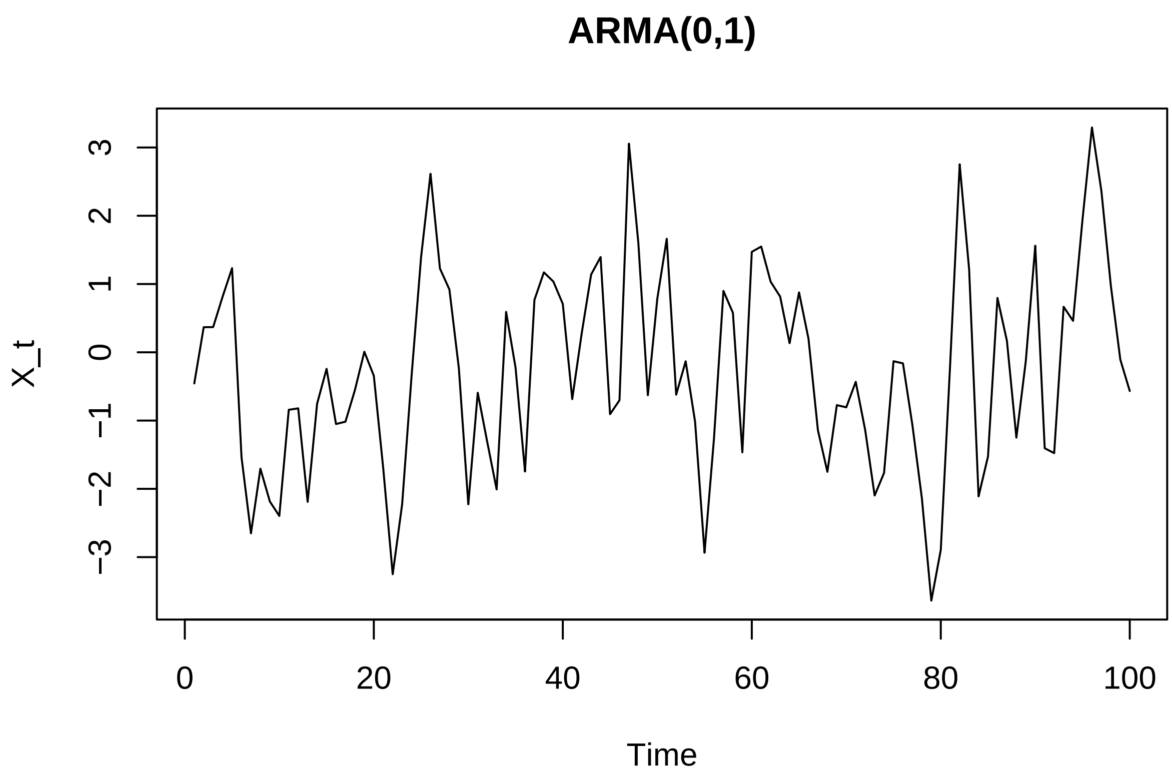
Question 2

```
# define the coeffs
phi = 0.6
the = 0.9
# save simulations into a matrix
SIM  <- matrix(0,100,3)
SIM[,1] <- arima.sim(n=100, list(ar = phi, ma = the))
SIM[,2] <- arima.sim(n=100, list(ar = phi))
SIM[,3] <- arima.sim(n=100, list(ma = the))
```

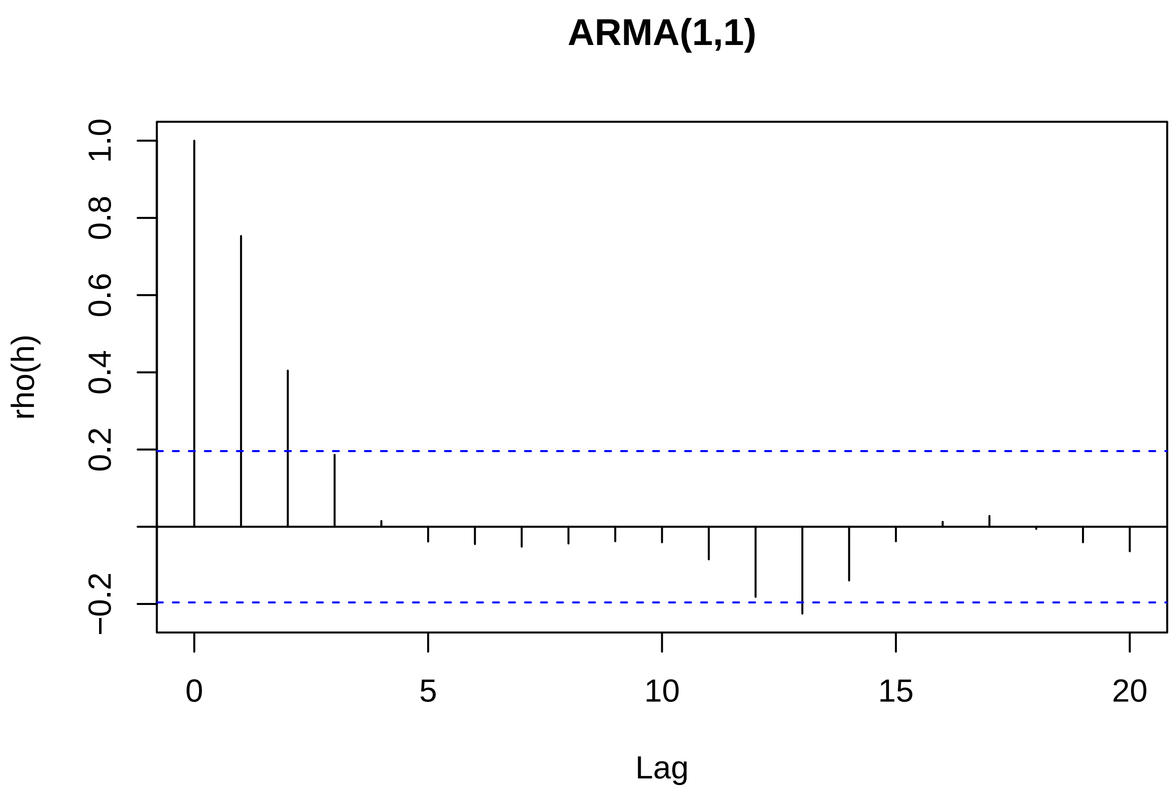
```
# plot the timeseries
name = c("ARMA(1,1)","ARMA(1,0)","ARMA(0,1)")
for(i in 1:3){plot.ts(SIM[,i],main = name[i], ylab = "X_t")}
```

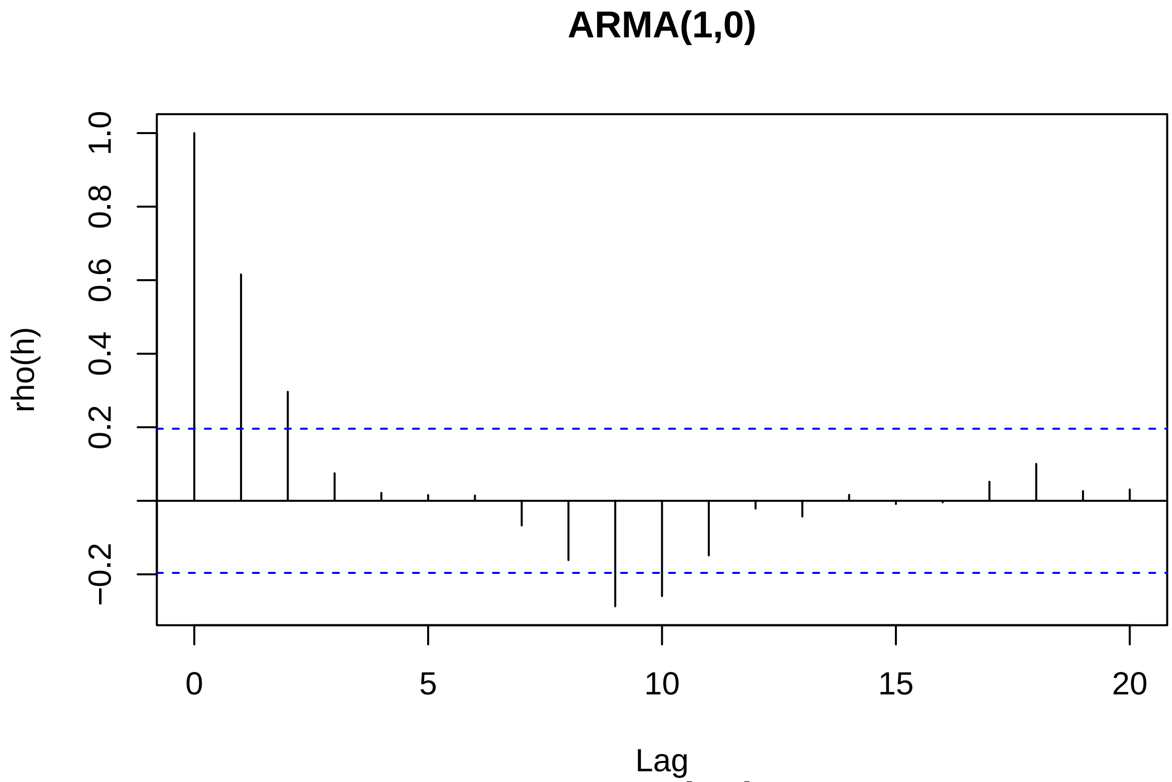

ARMA(1,1)
4
2
0
−2
−4
X_t
0
20
40
60
80
100
Time

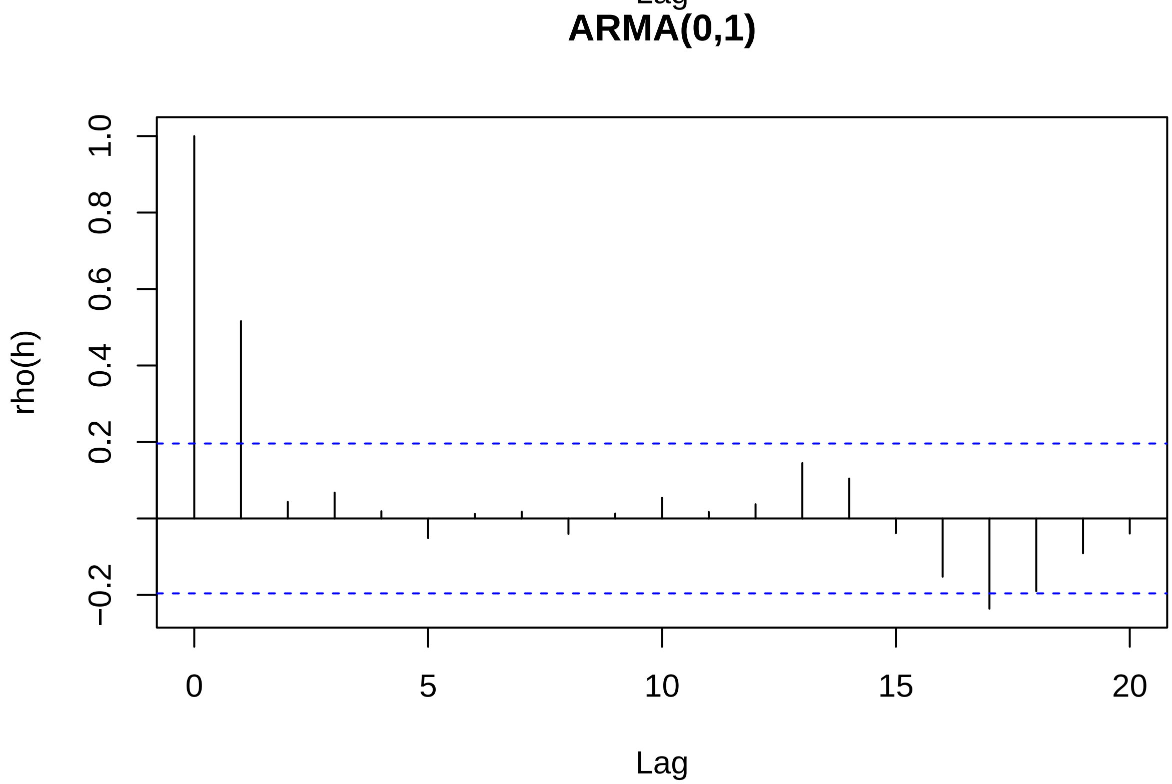

ARMA(1,0)
2
1
0
−1
−2
−3
−4
X_t
0
20
40
60
80
100
Time



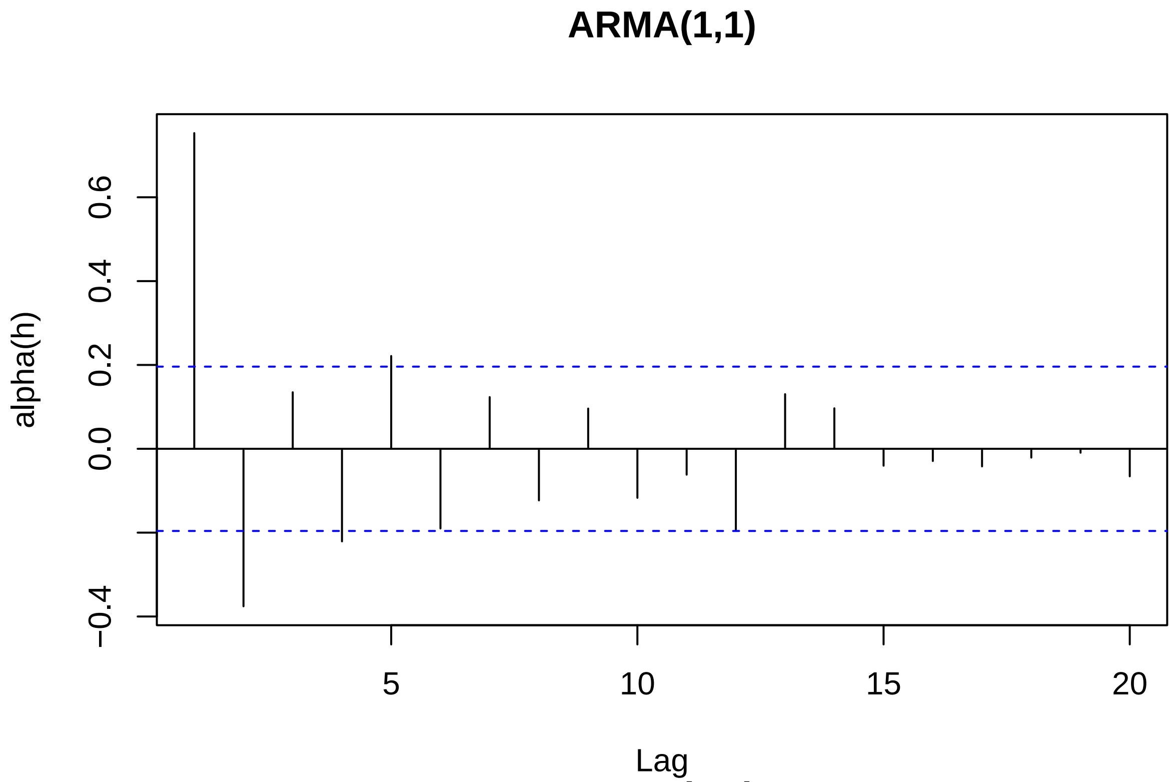

```r
# plot the ACFs
for(i in 1:3){acf(SIM[,i],main = name[i],ylab = "rho(h)")}
```

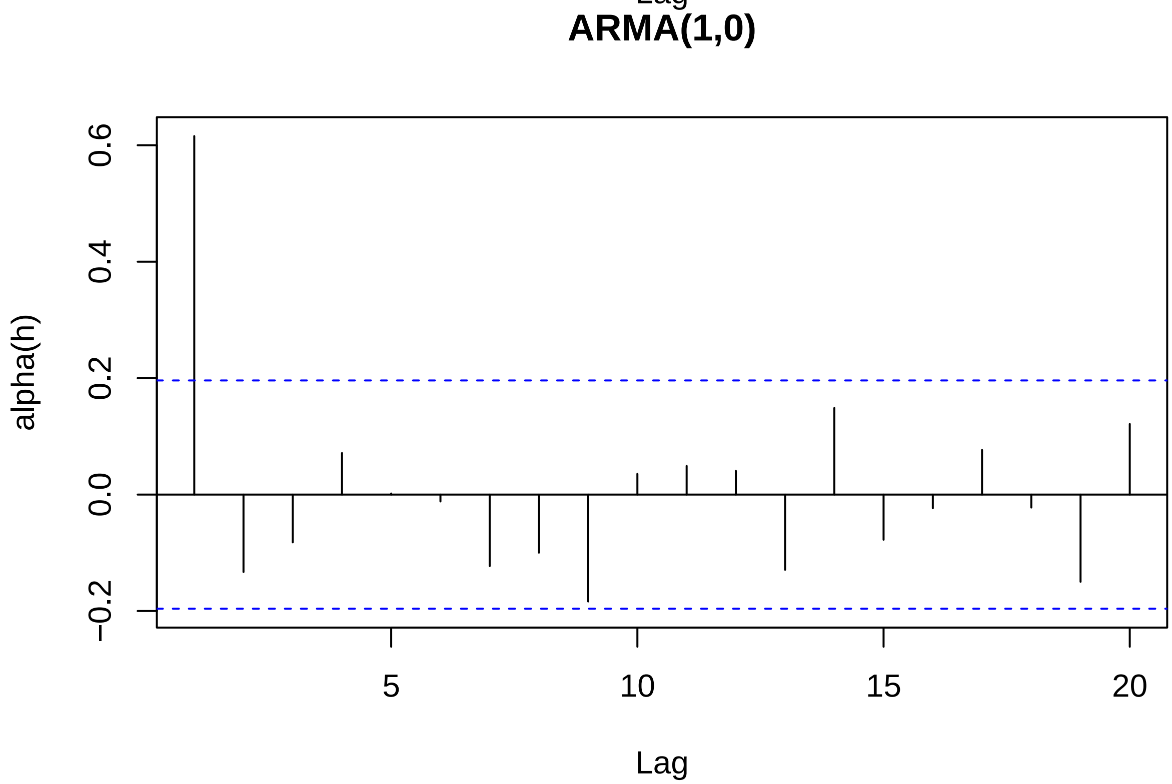


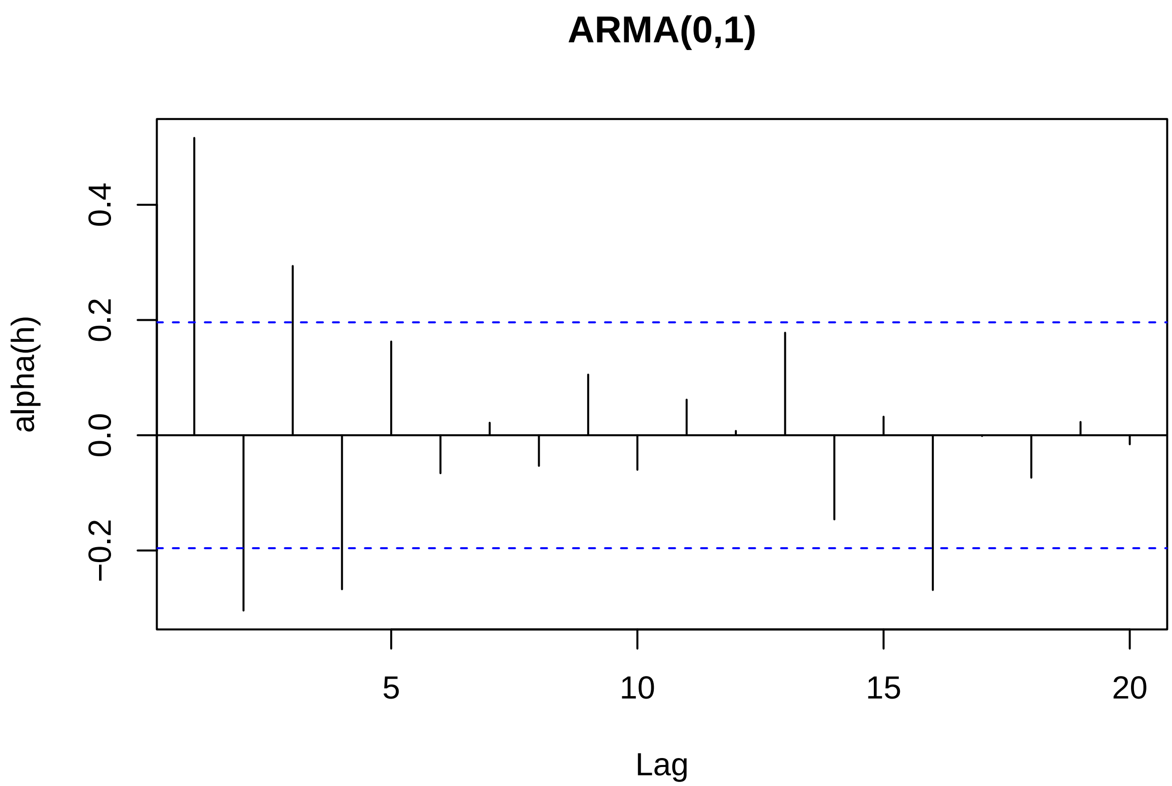





```
# plot the PACFs
for(i in 1:3){pacf(SIM[,i],main = name[i],ylab = "alpha(h)")}
```


ARMA(1,1)
alpha(h)
0.6
0.4
0.2
0.0
−0.4
5
10
15
20
Lag


ARMA(1,0)
alpha(h)
0.6
0.4
0.2
0.0
−0.2
5
10
15
20
Lag



The models seem to match the table we went over in class; I do think that even though there seems to be a significant peak in the ARIMA(1,0) PACF, it seems not to peak over enough to claim to be an ARIMA(2,0) model.

Question 3

```
# get the data
library(astsa)
```

```
##
## Attaching package: 'astsa'
## The following object is masked from 'package:forecast':
##
##     gas
```

```
# make models
ar.ols(cmort,order= 2) -> cmort.ar2
arima(cmort, order = c(2,0,0)) -> cmort.arima200
cmort.ar2$x.intercept
```

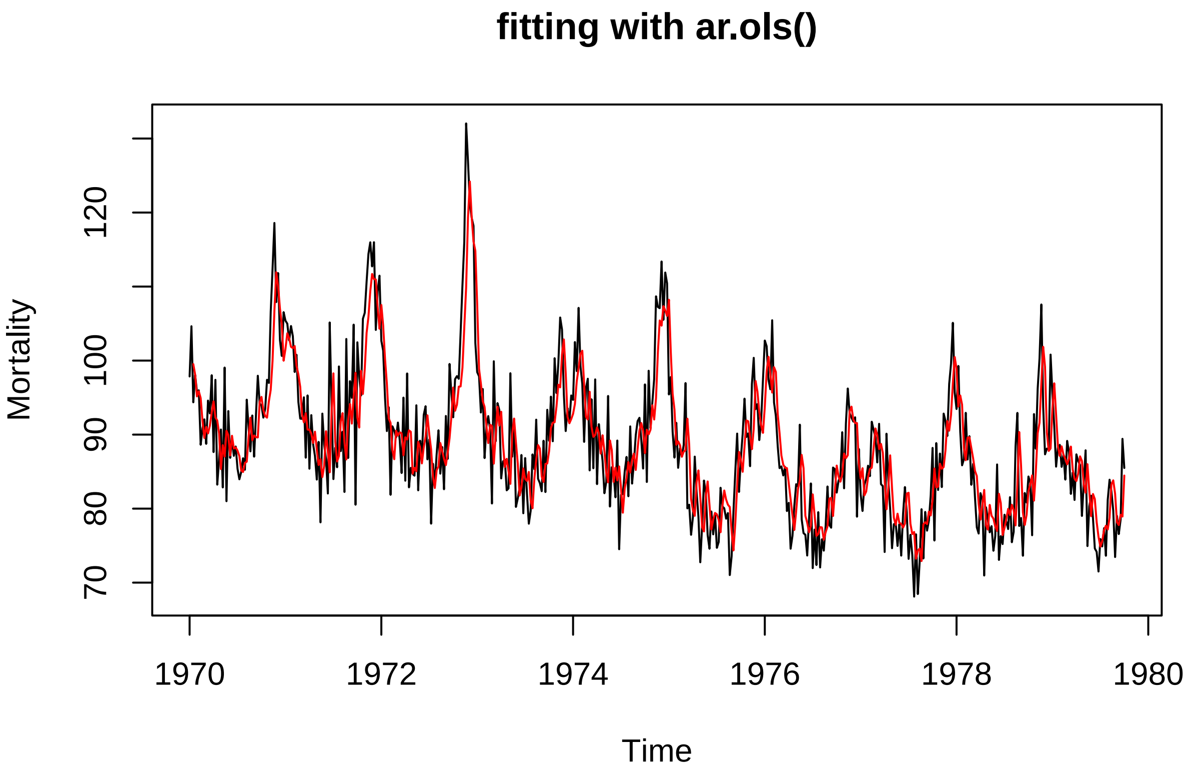
```
## [1] -0.04671956
```

```
summary(cmort.arima200)
```

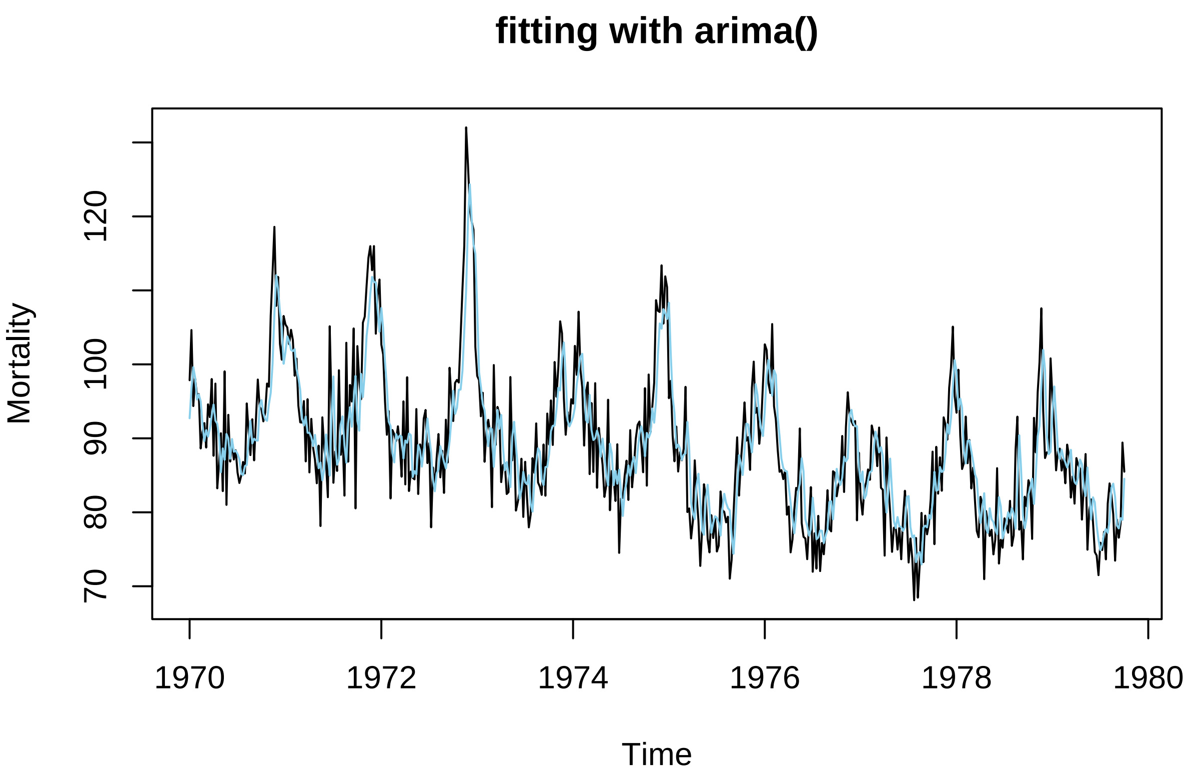
```
##
## Call:
## arima(x = cmort, order = c(2, 0, 0))
##
## Coefficients:
##          ar1     ar2  intercept
```

```
##        0.4301  0.4424    88.8538
## s.e.  0.0397  0.0398     1.9407
##
## sigma^2 estimated as 32.37:  log likelihood = -1604.71,  aic = 3217.43
##
## Training set error measures:
##                       ME     RMSE      MAE        MPE    MAPE     MASE
## Training set -0.04047717 5.689543 4.493995 -0.4445741 5.06128 0.846808
##                     ACF1
## Training set -0.01043152
```

```r
ts.plot(cmort, main = "fitting with ar.ols()", ylab = "Mortality")
lines(fitted(cmort.ar2),col = 'red')
```

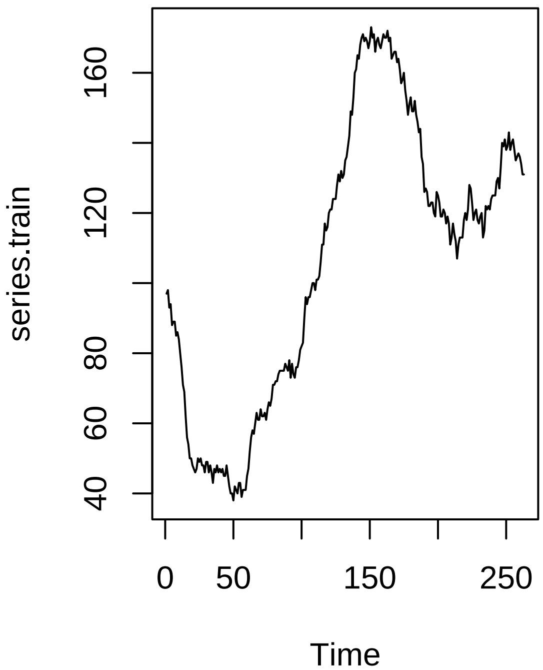


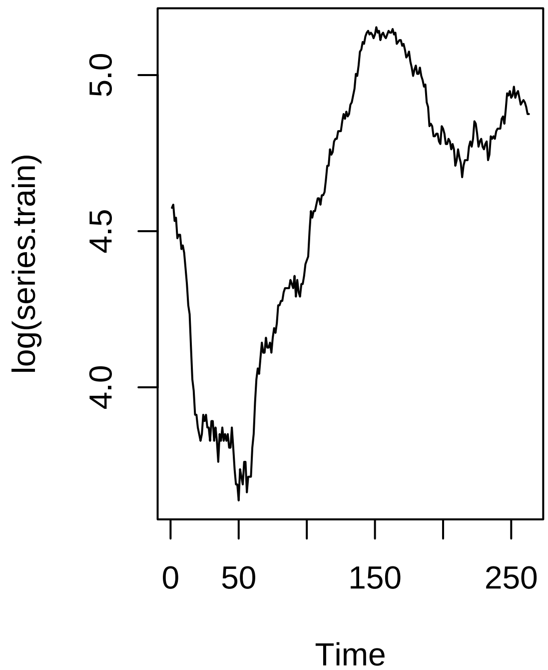

```r
ts.plot(cmort, main = "fitting with arima()", ylab = "Mortality")
lines(fitted(cmort.arima200),col = 'skyblue')
```



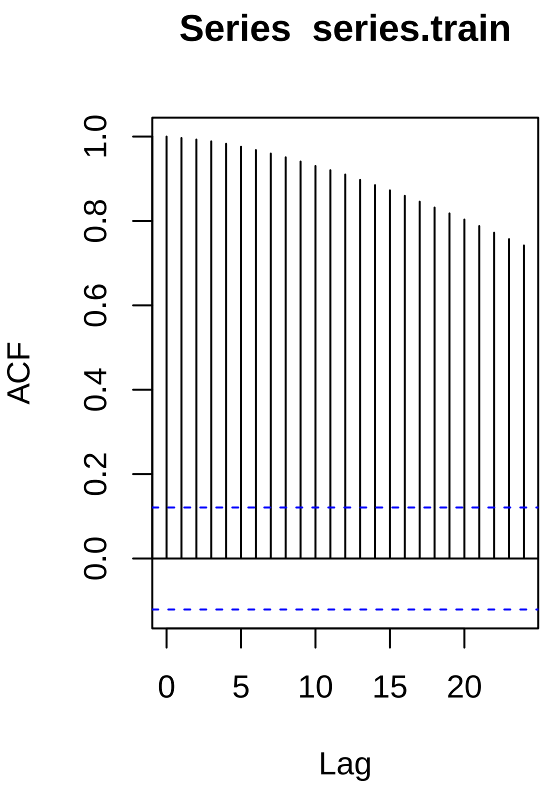
Question 4

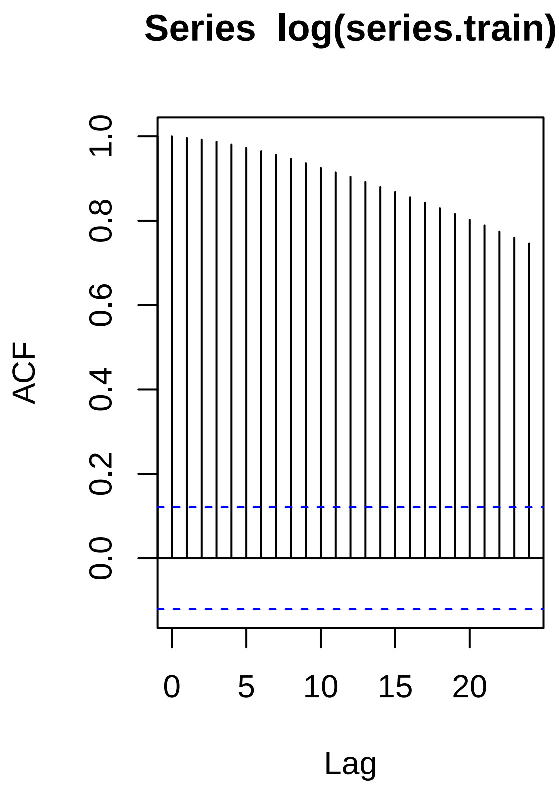
```
# getting data
demand <- as.vector(read.csv("Demand-2.txt", head = T)[,1])
series <- ts(demand, start =c(1992,1), frequency = 12)
series.train <- series[1:263]
series.val <- series[264:287]

par(mfrow=c(1,2))
plot.ts(series.train)
plot.ts(log(series.train))
```



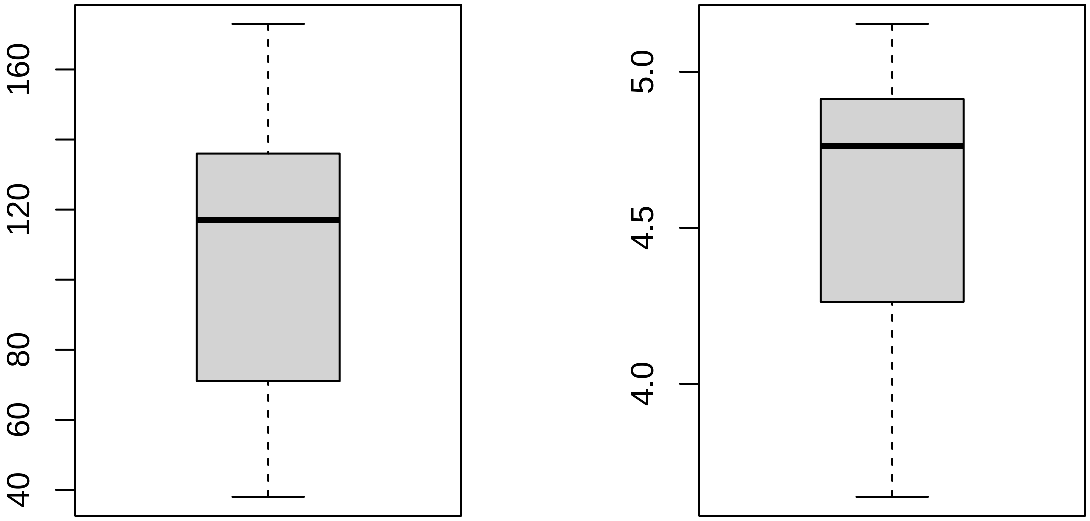



```r
acf(series.train)
acf(log(series.train))
```



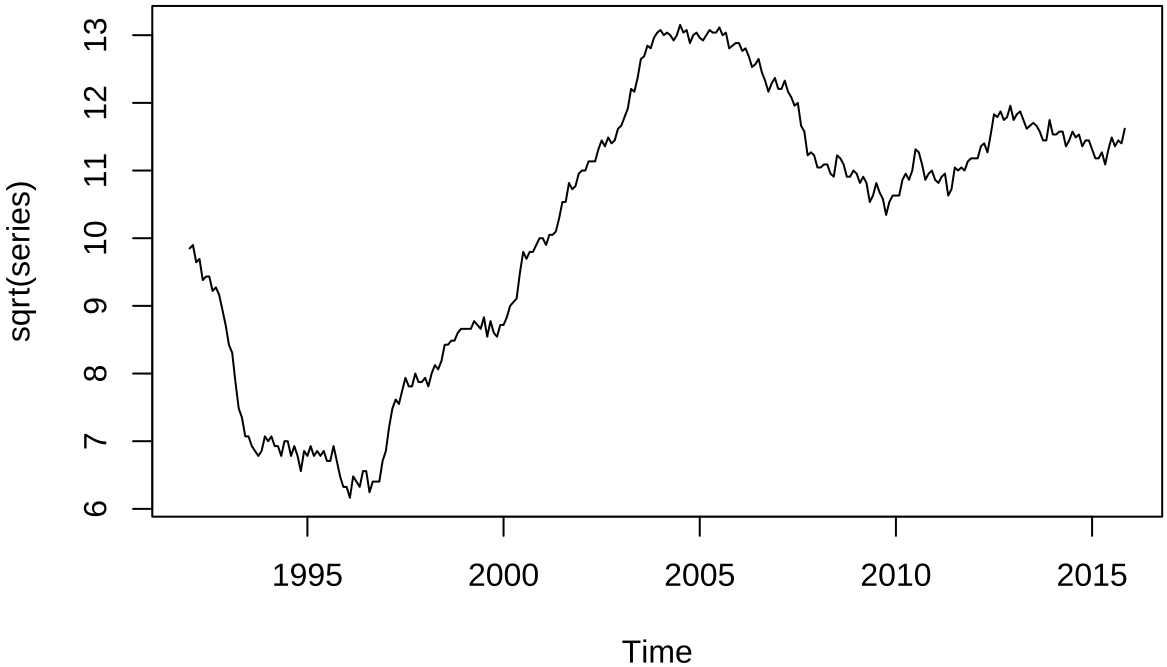



```r
boxplot(series.train)
boxplot(log(series.train))
```



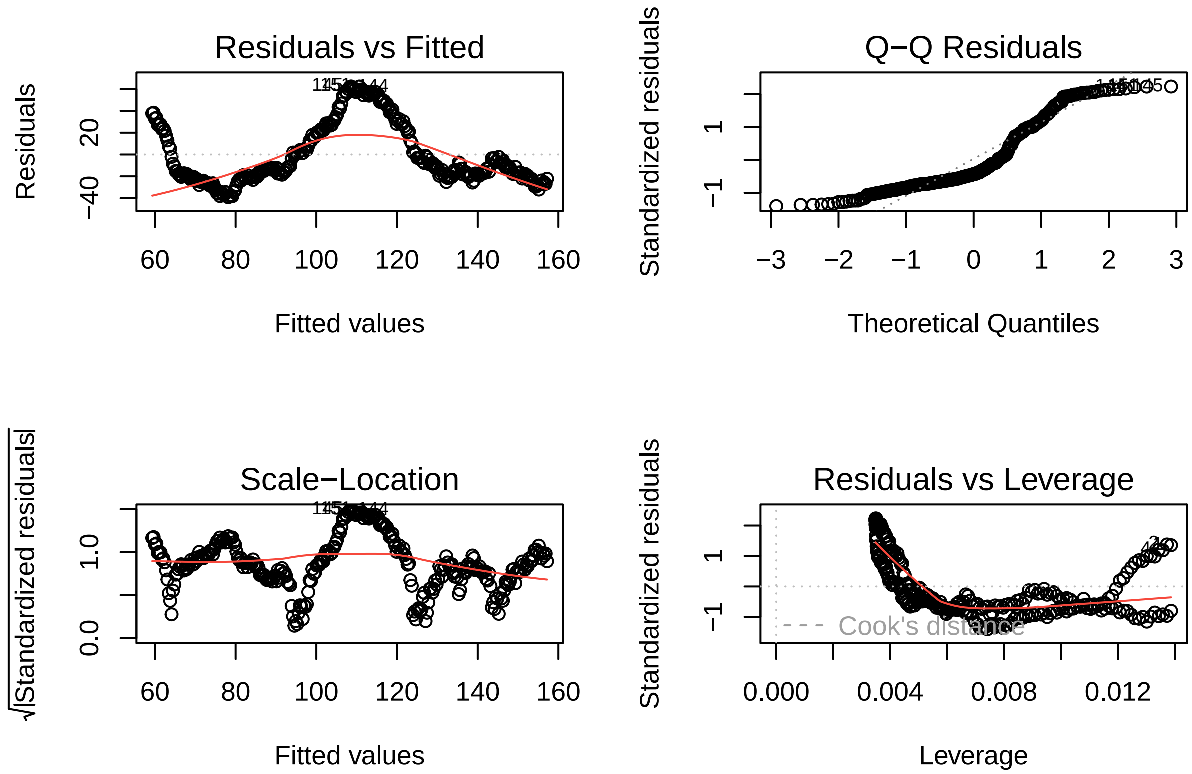
Going to try more transforms

```r
plot.ts(sqrt(series))
```

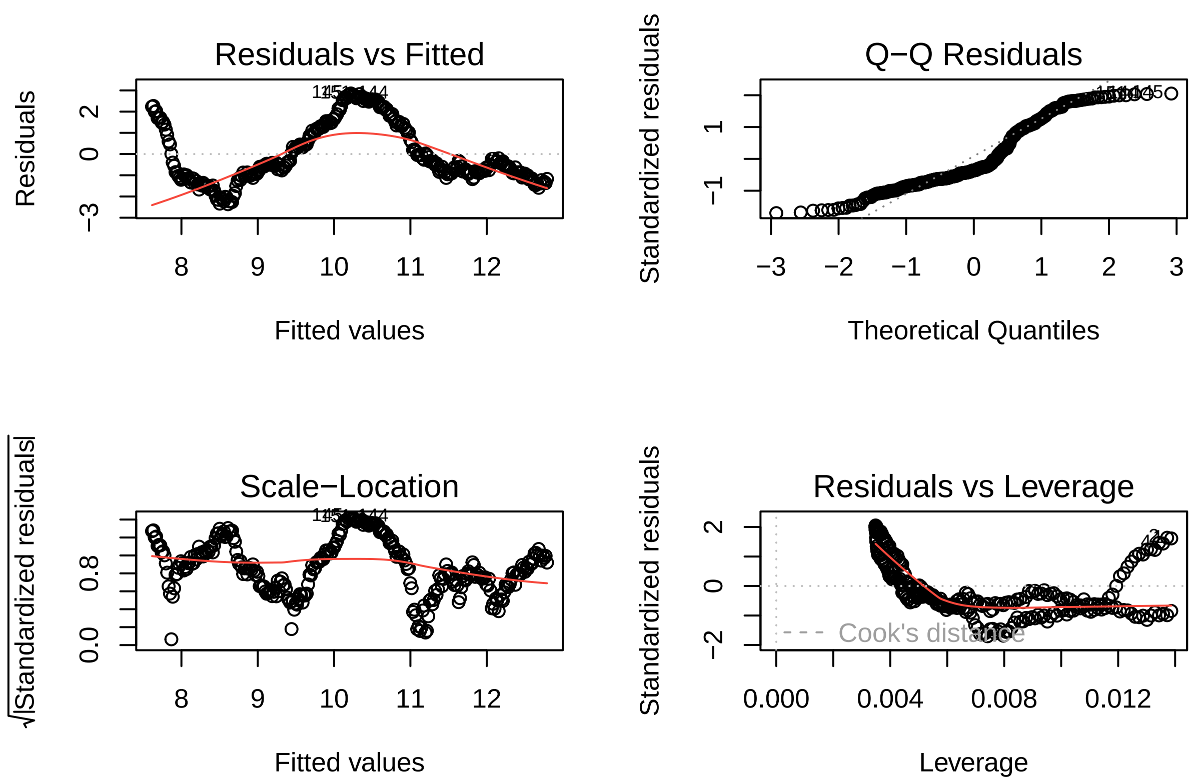



```r
plot.ts((series)^(1/3))
```

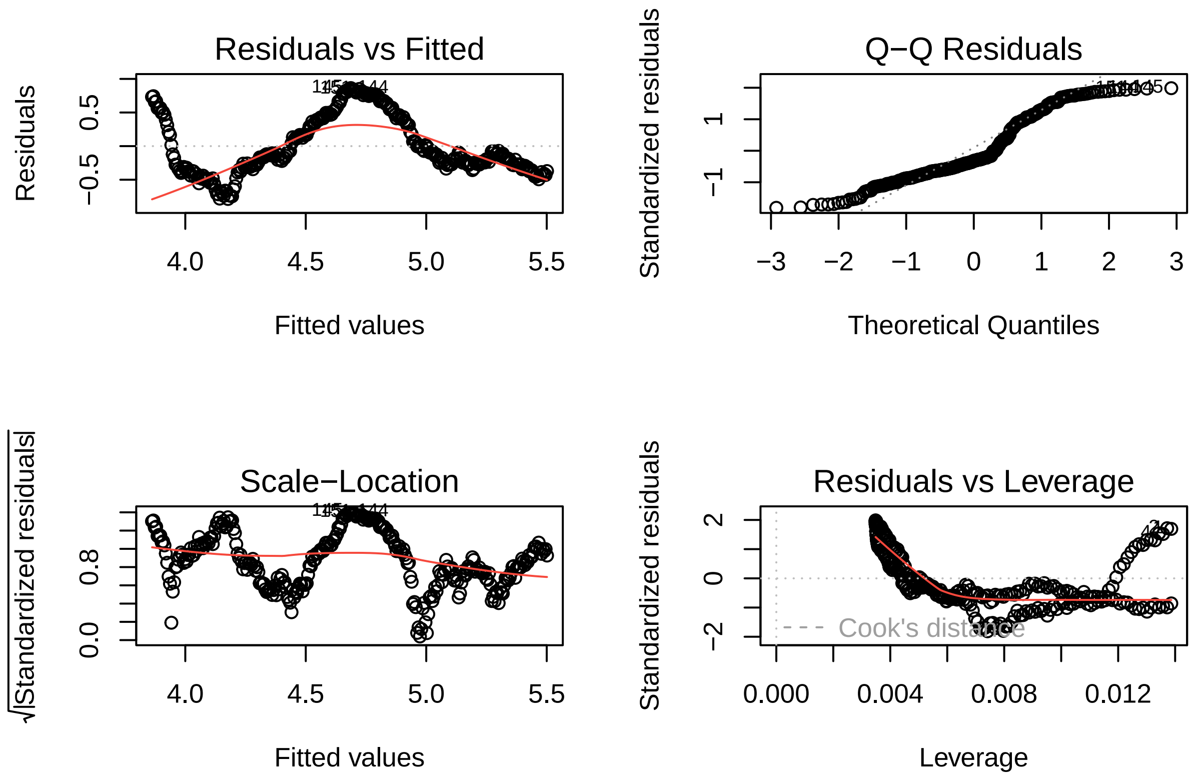

```r
# fit linear model :/
t = time(series)
fit0 <- lm(series ~ t)
fit1 <- lm(series ~ t + t^2)
fit2 <- lm(series ~ t + t^2 + t^3)
fit3 <- lm(series ~ t + t^2 + t^3 + t^4)
fit4 <- lm(sqrt(series) ~ t)
fit5 <- lm((series)^(1/3) ~ t)

par(mfrow = c(2,2))
plot(fit0)
plot(fit1)
```




```r
plot(fit2)
plot(fit3)
plot(fit4)
```



```r
plot(fit5)
```



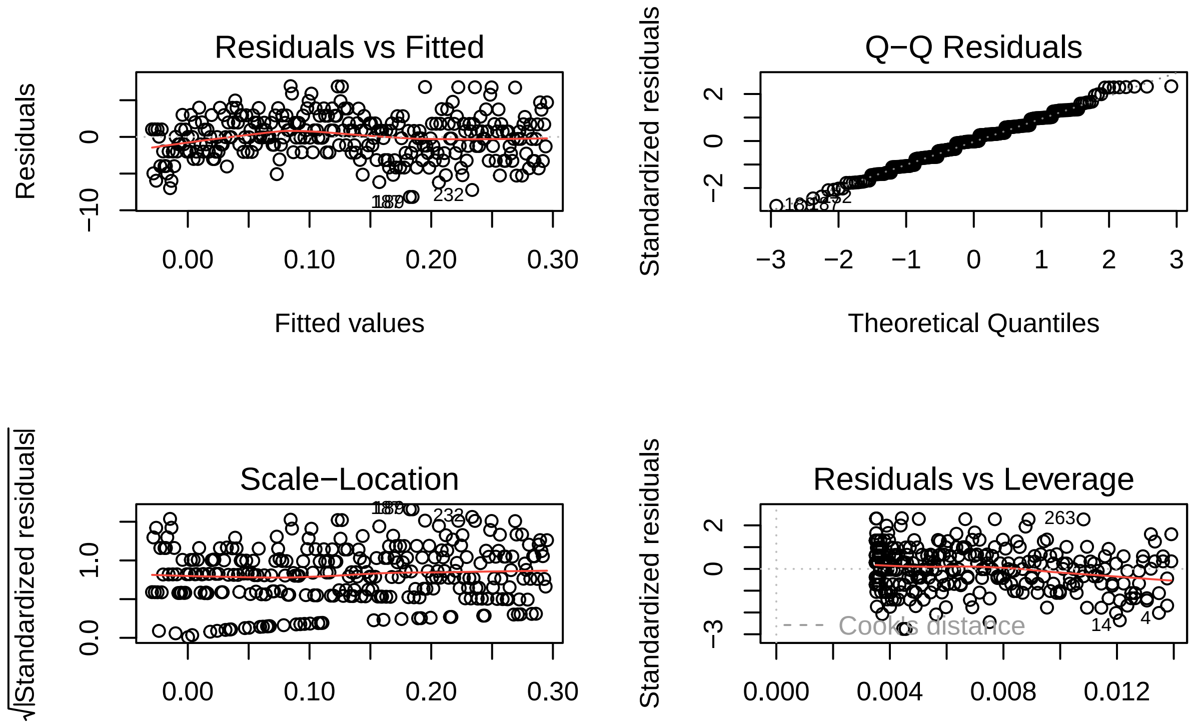
the scale-location line doesn't seem to straighten out for any linear fit we do. I am going to difference.

```r
#difference once
seriesD <- diff(series,diff = 1)
# run kpss test
seriesD_decomp <- decompose(seriesD,type = c("additive"))
seriesD_decomp <- na.omit(seriesD_decomp)
kpss.test(seriesD_decomp$random)
```

```
## Warning in kpss.test(seriesD_decomp$random): p-value greater than printed
## p-value

##
##  KPSS Test for Level Stationarity
##
## data:  seriesD_decomp$random
## KPSS Level = 0.018164, Truncation lag parameter = 5, p-value = 0.1
```

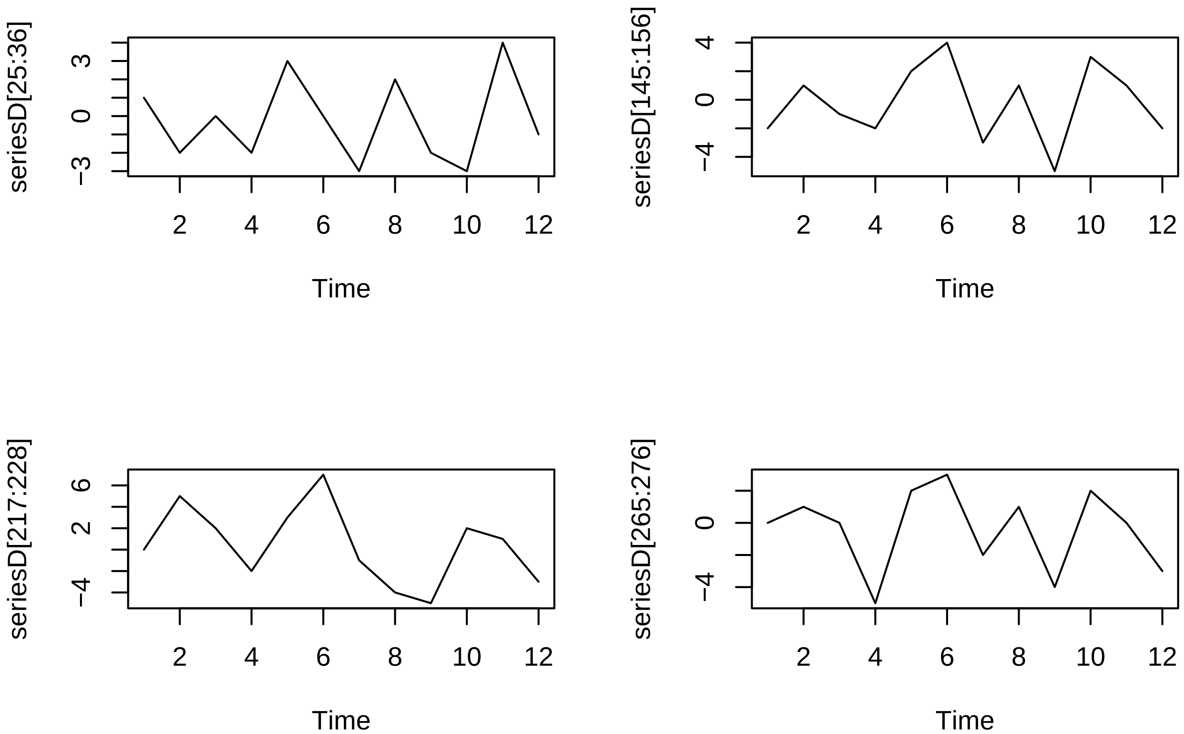
The test concludes a failure to reject null, there is a chance that the system is stationary, so lets check variance again.

```r
par(mfrow=c(2,2))
plot(lm(seriesD ~ time(seriesD)))
```



Yay. It is stabalized now!

```
# render graphs the single out random cycles of 12
par(mfrow=c(2,2))
plot.ts(seriesD[25:36])
plot.ts(seriesD[145:156])
plot.ts(seriesD[217:228])
plot.ts(seriesD[265:276])
```



It seems in the later months, there is a peak it