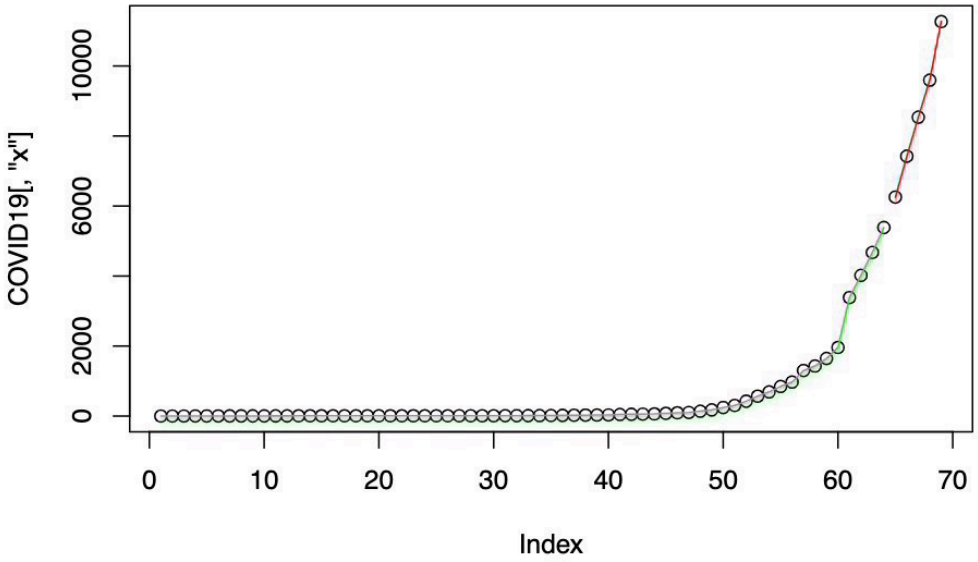
First try to use regression to forecast the future values.

```
COVID19 = read.csv("COVID19.txt")

train = COVID19[1:64,]
test = COVID19[65:69,]

{plot(COVID19[,"x"])
  lines(train, col="green")
  lines(x=seq(65,69,by=1),test,  col="red")
}
```



There is a steep upward trend after around March 15th, no sign of seasonality. The data is not statinary.

```r
t = seq(1/64,1, by = 1/64 )
t2 = t*t
t3 = t2*t
t4 = t3*t
t5 = t4*t


model1 = lm(formula = train~t)
model2 = lm(formula = train~t+t2)
model3 = lm(formula = train~t+t2+t3)
model4 = lm(formula = train~t+t2+t3+t4)
model5 = lm(formula = train~t+t2+t3+t4+t5)

new.t = seq(65/64, 69/64, by = 1/64)
new.t2 = new.t*new.t
new.t3 = new.t2*new.t
new.t4 = new.t3*new.t
new.t5 = new.t4*new.t

new1 = data.frame(t = new.t)
new2 = data.frame(t = new.t, t2 = new.t2  )
new3 = data.frame(t = new.t, t2 = new.t2, t3=new.t3 )
new4 = data.frame(t = new.t, t2 = new.t2, t3=new.t3, t4=new.t4 )
new5 = data.frame(t = new.t, t2 = new.t2, t3=new.t3, t4=new.t4, t5 = new.t5  )

predict.model1 = predict(model1, new=new1)
predict.model2 = predict(model2, new=new2)
predict.model3 = predict(model3, new=new3)
predict.model4 = predict(model4, new=new4)
predict.model5 = predict(model5, new=new5)


sum((predict.model1 - test)^2)
```

```
## [1] 253265290
```

```r
sum((predict.model2 - test)^2)
```

```
## [1] 142906752
```

```r
sum((predict.model3 - test)^2)
```

```
## [1] 55671940
```

```r
sum((predict.model4 - test)^2)
```

```
## [1] 10007096
```

```r
sum((predict.model5 - test)^2)
```

```
## [1] 226635.4
```

PRESS is smallest with p = 5.

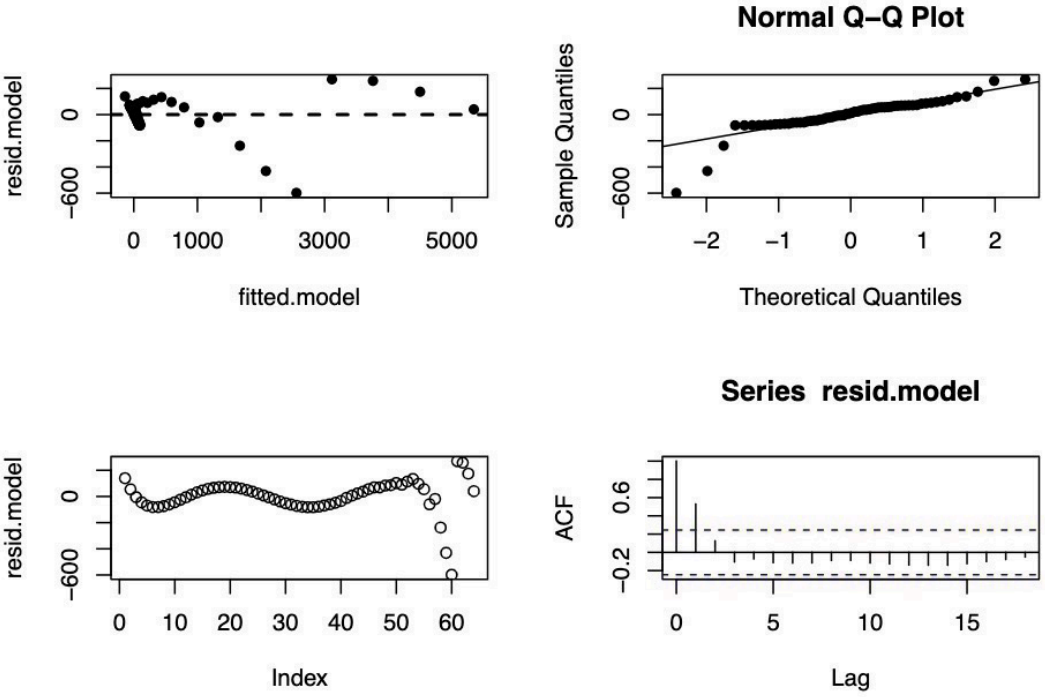
```r
resid.model = model5$residuals
fitted.model = model5$fitted.values

{
par(mfrow=c(2,2))
#resid vs fitted
plot(resid.model~fitted.model, pch = 16)
abline(h=0, lwd=2, lty=2)

#qqplot
qqnorm(resid.model,pch=16)
qqline(resid.model)

#residuals plot
plot(resid.model)

#acf plot
acf(resid.model, pch=16)
}
```




```r
#constant variance fligner
seg = rep(1:4, each = 16)
fligner.test(resid.model, seg)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  resid.model and seg
```

```
## Fligner-Killeen:med chi-squared = 5.5869, df = 3, p-value = 0.1335
```

```
# sharpio test normality

shapiro.test(resid.model)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  resid.model
## W = 0.80658, p-value = 9.844e-08
```

```
# Runs Test
library(lawstat)
runs.test(resid.model)
```

```
##
##  Runs Test - Two sided
##
## data:  resid.model
## Standardized Runs Statistic = -6.5522, p-value = 5.669e-11
```

1. Based on the graphs, can see a strong pattern in residuals vs fitted plot. Mean is not constant 0.
2. There is also a strong pattern in the residuals plot, residuals are not independent.
3. The acf has 2 spikes outside the bandwidth, which is more that 5% spikes outside the bands so not white noise.
4. The points don't lie very well on the qq plot, shows that the residuals aren't really normal.
5. Fligner's test P value is large, indicating no evidence against homogeneous variance.
6. Shapiro's test P value is small, indicating strong evidence against normality.
7. Run's test p value is really small, indicating strong evidence against randomness.

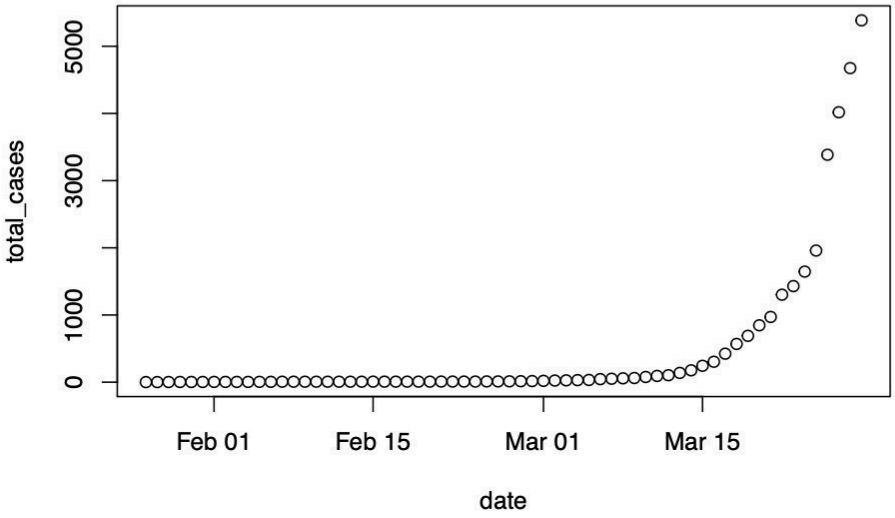
```
new.t = seq(70/64, 76/64, by = 1/64)
new.t2 = new.t*new.t
new.t3 = new.t2*new.t
new.t4 = new.t3*new.t
new.t5 = new.t4*new.t
new4 = data.frame(t = new.t, t2 = new.t2, t3=new.t3, t4=new.t4, t5 = new.t5 )
t = seq(1/64,69/64, by = 1/64 )
t2 = t*t
t3 = t2*t
t4 = t3*t
t5 = t4*t
dat = COVID19[,"x"]
finalModel = lm(dat~t+t2+t3+t4+t5)
predict(finalModel, new=new4, interval = "prediction")
```

```
##        fit      lwr      upr
## 1 12922.33 12581.82 13262.85
## 2 14747.94 14366.14 15129.73
## 3 16766.00 16328.38 17203.63
## 4 18990.73 18481.30 19500.16
## 5 21436.98 20838.68 22035.28
## 6 24120.28 23415.04 24825.53
## 7 27056.85 26225.56 27888.14
```

The predictions for April 4 to April 10 along with their 95% prediction intervals are displayed above.

Checking data with no differencing, the data is obviously not stationary as shown in the plot below.

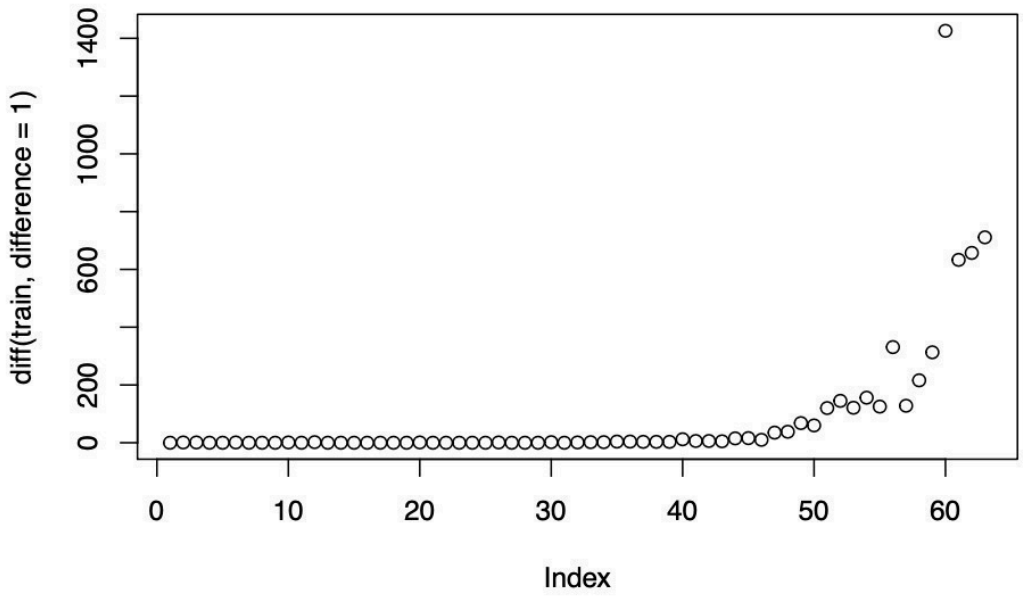
```
plot(trainS)
```



The plot shows the data is not stationary, as there is a clear upward trend.

Next, check data with difference =1.

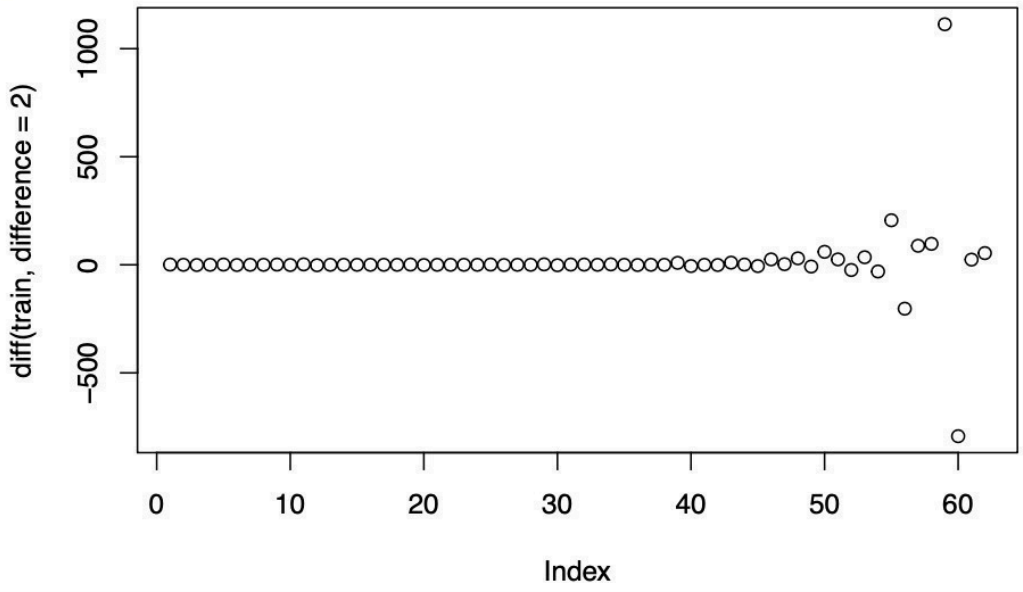
```
plot(diff(train, difference=1))
```



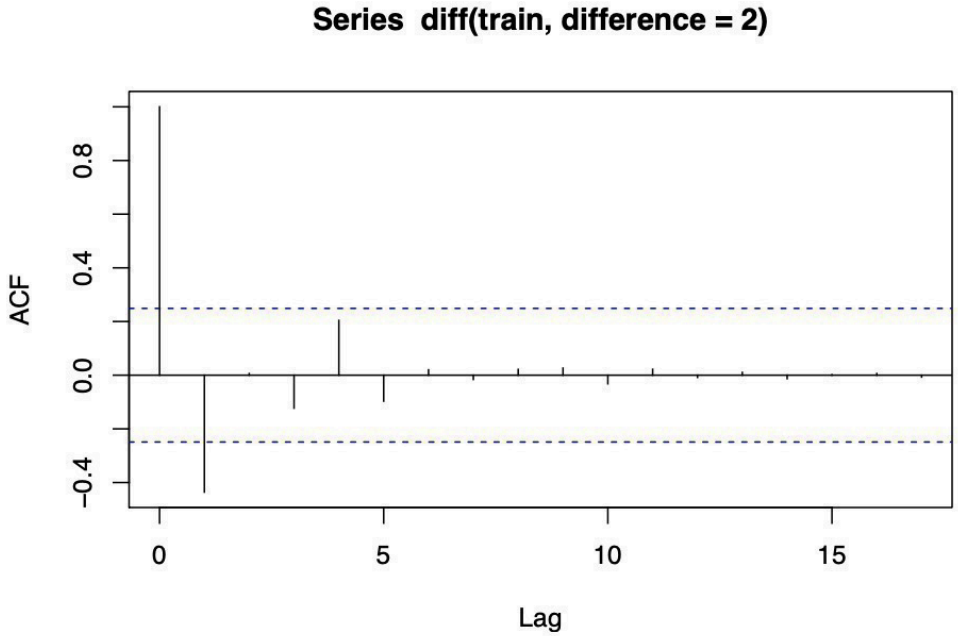
The plot shows the data is still not stationary, as there is a clear upward trend.

Next, check with difference = 2

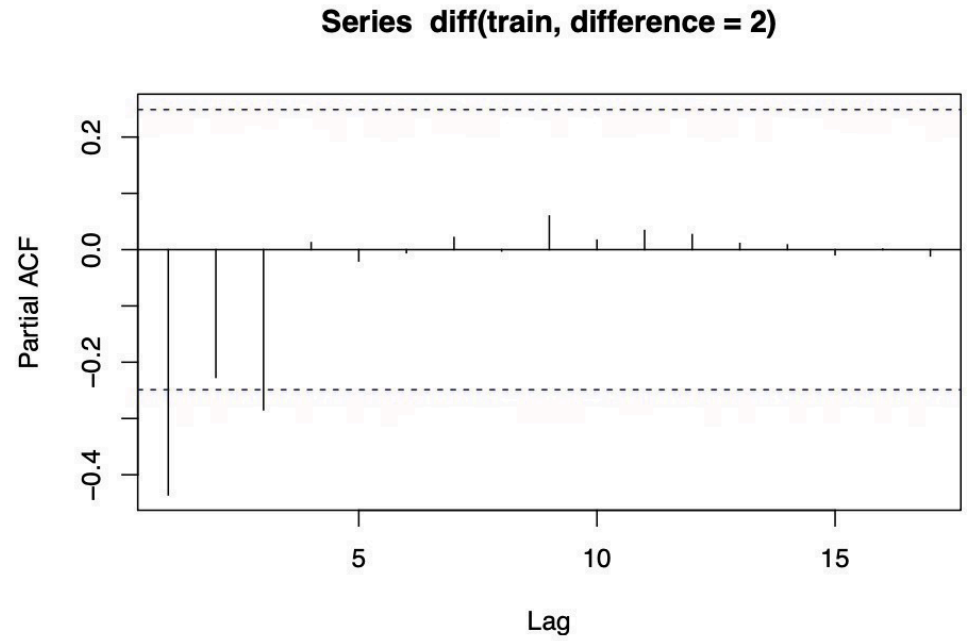
```
plot(diff(train, difference=2))
```




```
acf(diff(train, difference=2))
```



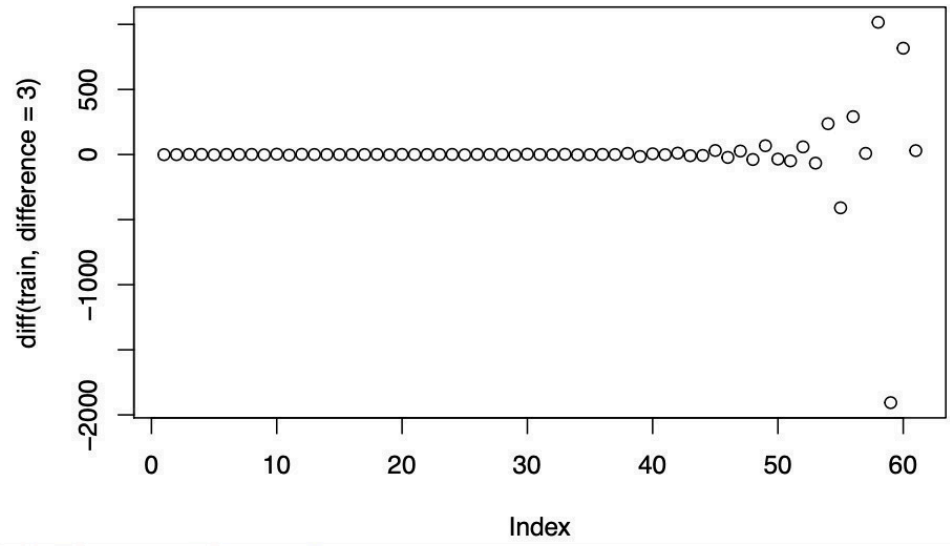

```
pacf(diff(train, difference=2))
```



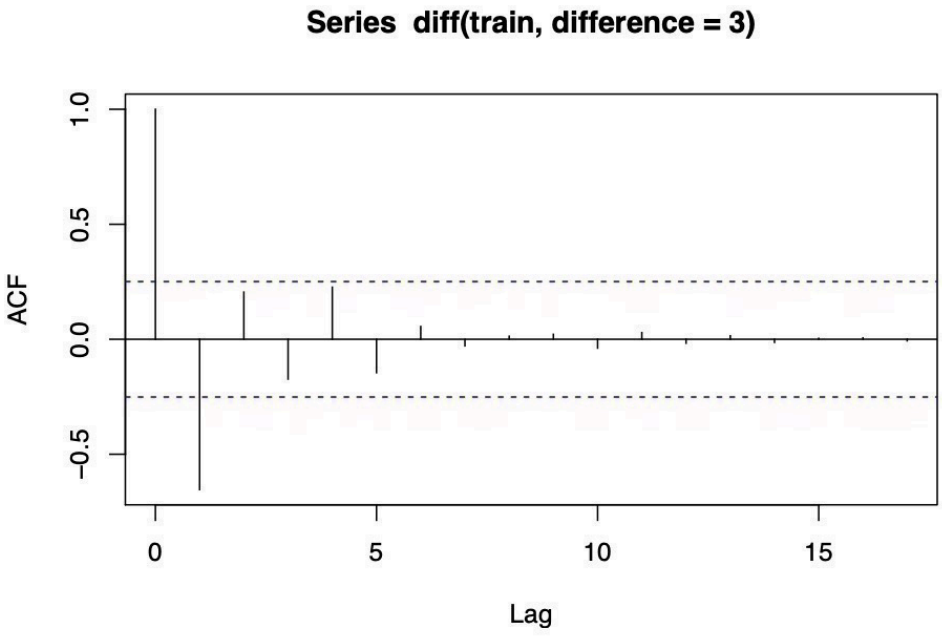
Now, the data is stationary as shown in the data plot. There is no time dependent pattern in the acf plot and pacf plot. The Acf plot has spikes cut off after lag 1 with exponential deday. Pacf also has spike cut off after lag 1 with exponential decay.

Trying third differencing adds variance as shown below.

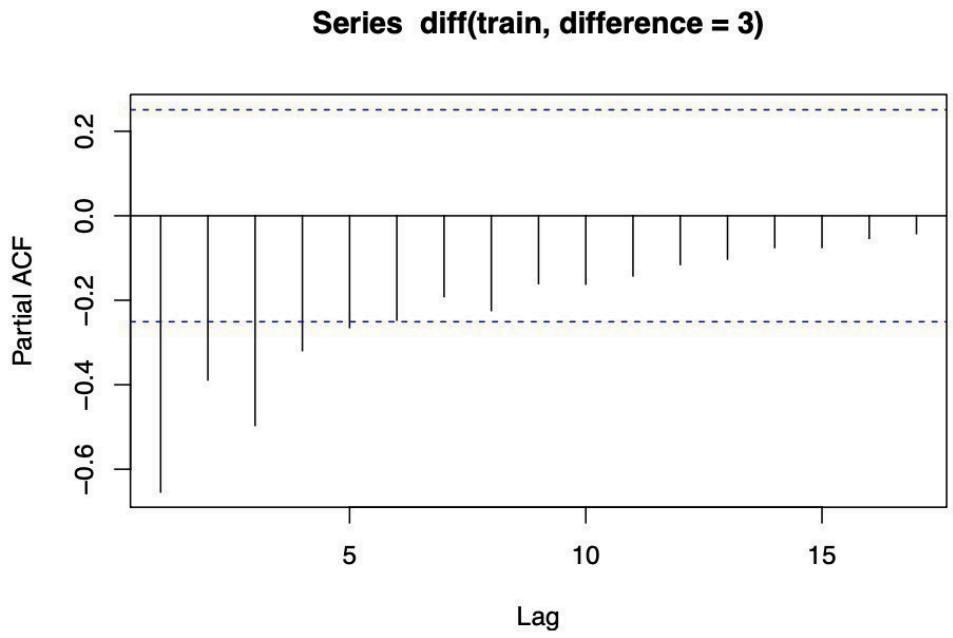
```
plot(diff(train, difference=3))
```




```
acf(diff(train, difference=3))
```

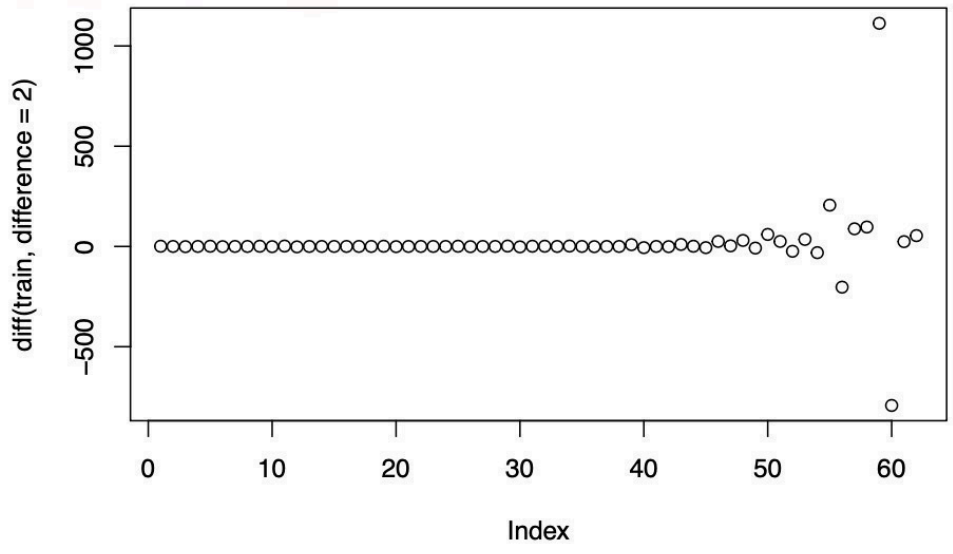


```
pacf(diff(train, difference=3))
```

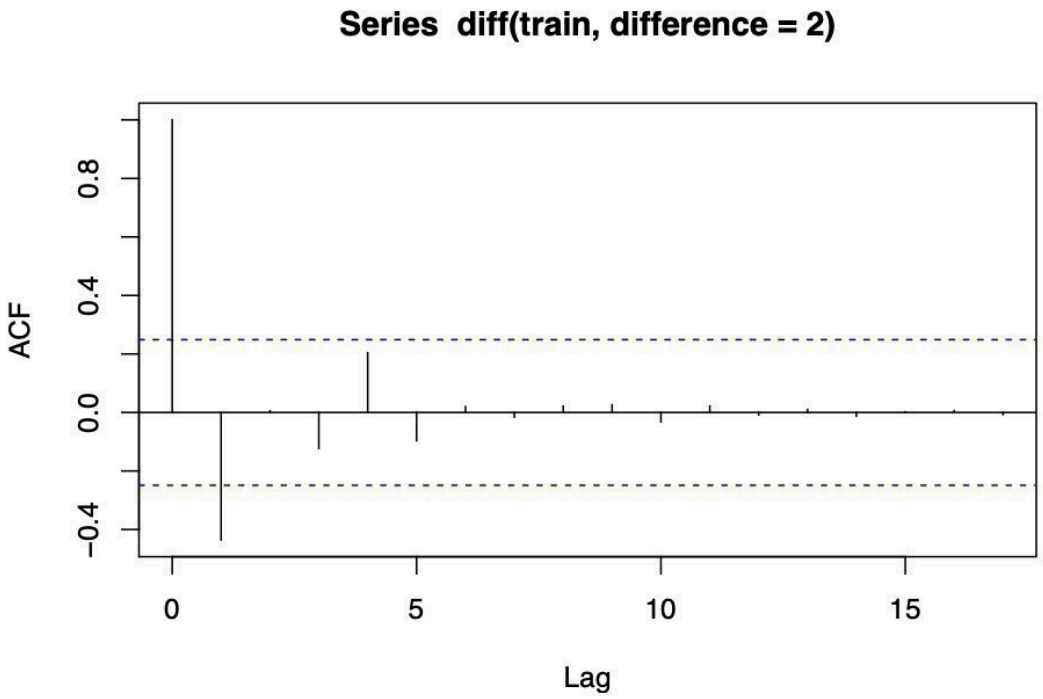


a) part 2, find p,q, propose model

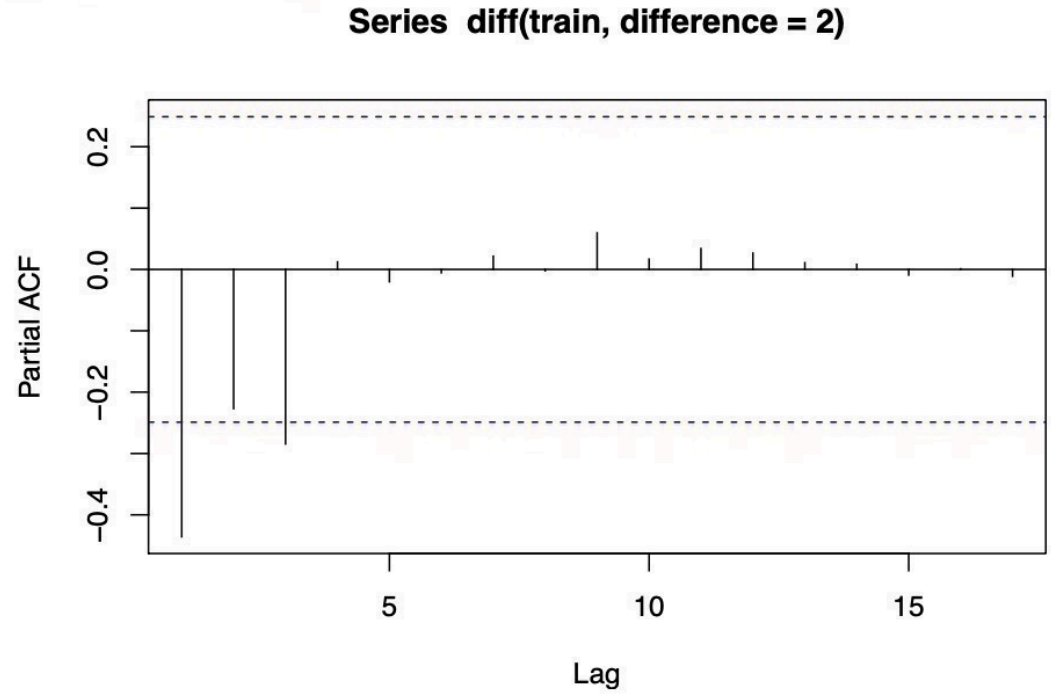
```
plot(diff(train, difference=2))
```



```
acf(diff(train, difference=2))
```



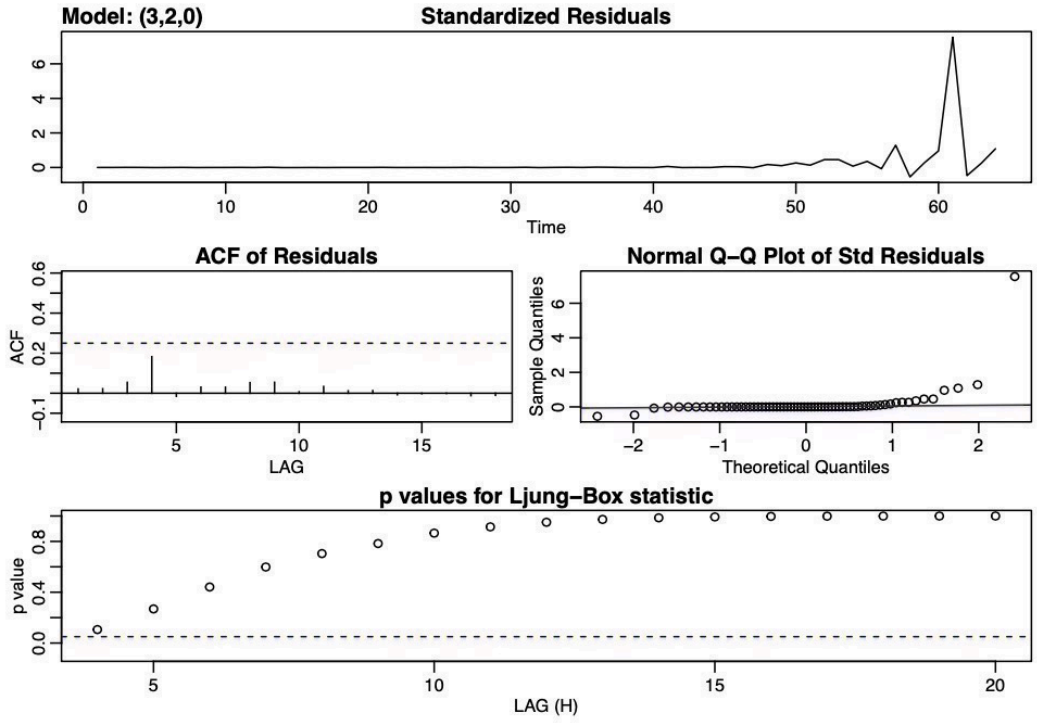

```
pacf(diff(train, difference=2))
```



Using differencing = 2 makes data stationary. We can see from the ACF plot that the spike cuts off after lag 1 at an exponential damped sinusoid. PACF cuts off after lag 3 and is at exponential decay as well. Based on these, the three proposed models from second differencing are ARIMA(3,2,0), ARIMA(0,2,1), ARIMA(1,2,1).

```
library(astsa)
fit1 <- sarima(train, p=3, d=2, q=0)
```

```
## initial  value 5.210732
## iter   2 value 5.102407
## iter   3 value 5.051953
## iter   4 value 5.034617
## iter   5 value 5.033833
## iter   6 value 5.033818
## iter   7 value 5.033817
## iter   7 value 5.033817
## iter   7 value 5.033817
## final  value 5.033817
## converged
## initial  value 5.015498
## iter   2 value 5.015242
## iter   3 value 5.015151
## iter   4 value 5.015137
## iter   4 value 5.015137
## iter   4 value 5.015137
## final  value 5.015137
## converged
```



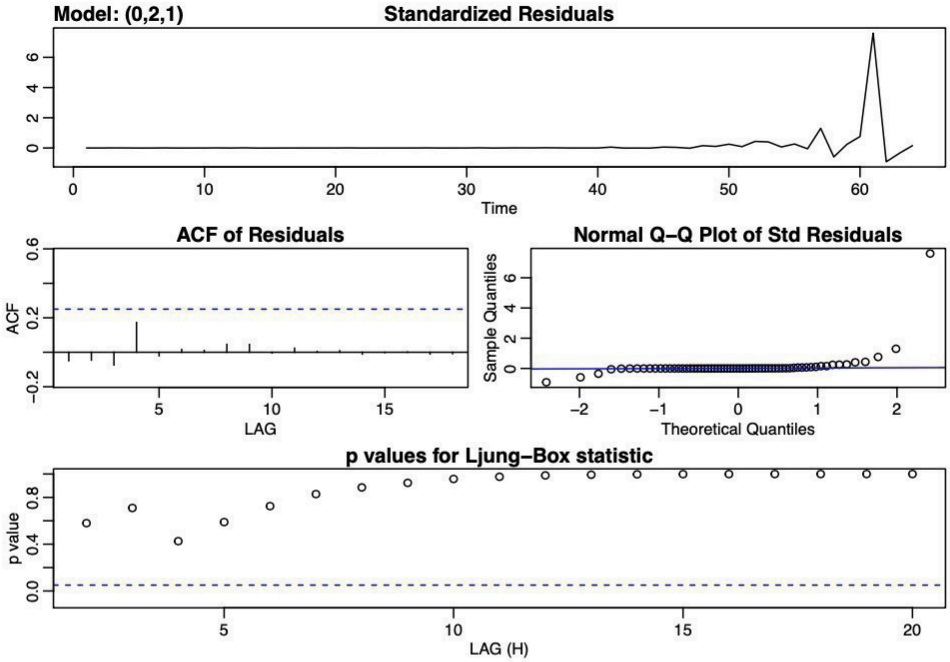
```r
seg = rep(1:4, each = 16)
fligner.test(resid(fit1$fit), seg)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  resid(fit1$fit) and seg
## Fligner-Killeen:med chi-squared = 33.748, df = 3, p-value = 2.24e-07
```

For ARMA(3,2,0), there is a high bump for the residuals at around time 61-63, otherwise the residuals are very constant around 0 and have no pattern. The acf shows the residuals are whitenoise as no spikes are outside the dotted bands. All points are above the dash line in the Ljung-Box statistic. The points lie quite well on a straight line for the qq-plot which means it's gaussian. The fligner's test has a small p-value, indicating evidence against homogeneous variance.

```r
fit2 <- sarima(train, p=0, d=2, q=1)
```

```
## initial  value 5.185934
## iter   2 value 5.054903
## iter   3 value 5.047572
## iter   4 value 5.043822
## iter   5 value 5.043574
## iter   6 value 5.043040
## iter   7 value 5.043037
## iter   7 value 5.043037
## final  value 5.043037
## converged
## initial  value 5.046078
## iter   2 value 5.046055
## iter   3 value 5.046030
## iter   3 value 5.046030
## iter   3 value 5.046030
## final  value 5.046030
## converged
```




```
seg = rep(1:4, each = 16)
fligner.test(resid(fit2$fit), seg)
```

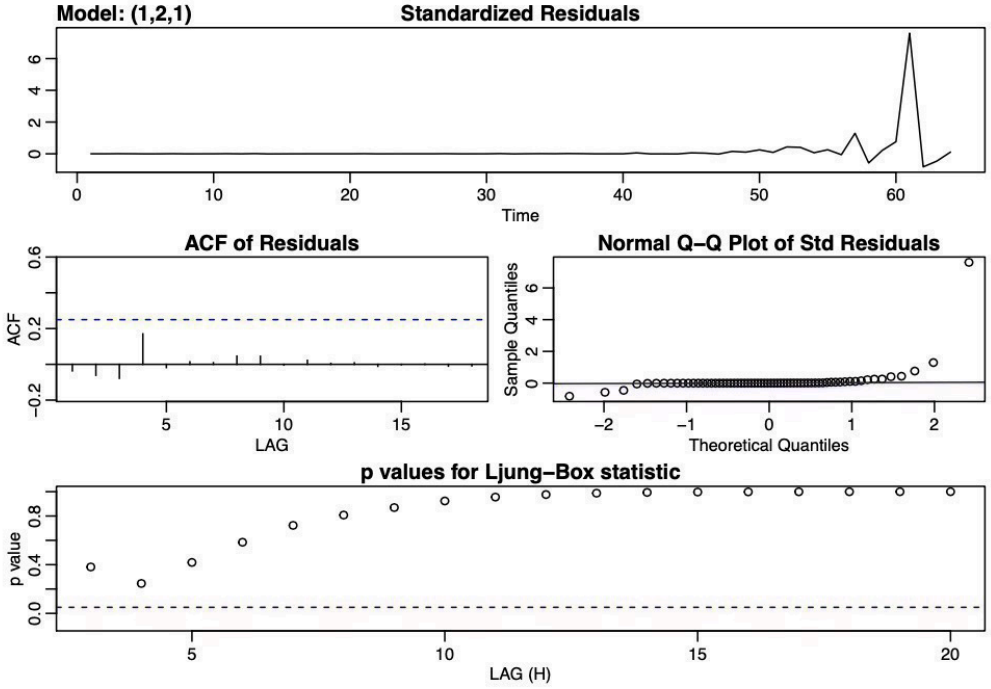
```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  resid(fit2$fit) and seg
## Fligner-Killeen:med chi-squared = 44.308, df = 3, p-value = 1.298e-09
```

For ARMA(0,2,1), there is a high bump for the residuals at around time 61-63, otherwise the residuals are very constant around 0 and have no pattern. The acf shows the residuals are whitenoise as no spikes are outside the dotted bands. All points are above the dash line in the Ljung-Box statistic. The points lie quite well on a straight line for the qq-plot which means it's gaussian. The fligner's test has a small p-value, indicating evidence against homogeneous variance.

```
fit3 <- sarima(train, p=1, d=2, q=1)
```

```
## initial  value 5.194064
## iter   2 value 5.096934
## iter   3 value 5.061979
## iter   4 value 5.057735
## iter   5 value 5.054122
## iter   6 value 5.051993
## iter   7 value 5.051364
## iter   8 value 5.051029
## iter   9 value 5.050933
## iter  10 value 5.050891
## iter  11 value 5.050888
```

```
## iter  12 value 5.050888
## iter  12 value 5.050888
## iter  12 value 5.050888
## final  value 5.050888
## converged
## initial  value 5.045847
## iter   2 value 5.045809
## iter   3 value 5.045795
## iter   3 value 5.045795
## iter   3 value 5.045795
## final  value 5.045795
## converged
```

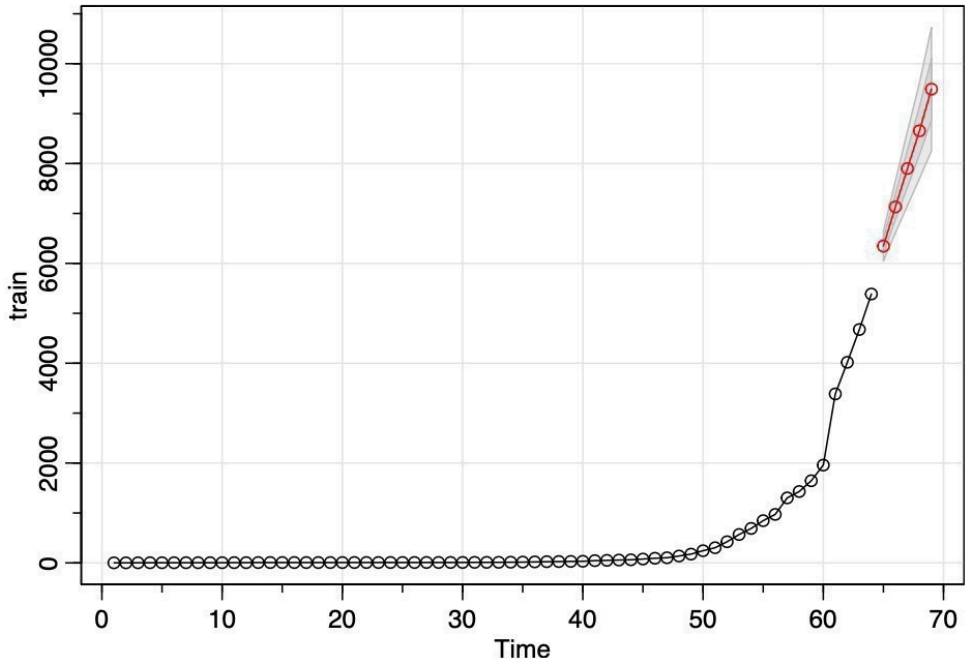



```
seg = rep(1:4, each = 16)
fligner.test(resid(fit3$fit), seg)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  resid(fit3$fit) and seg
## Fligner-Killeen:med chi-squared = 44.717, df = 3, p-value = 1.063e-09
```

For ARMA(1,2,1), there is a high bump for the residuals at around time 61-63, otherwise the residuals are very constant around 0 and have no pattern. The acf shows the residuals are whitenoise as no spikes are outside the dotted bands. All points are above the dash line in the Ljung-Box statistic. The points lie quite well on a straight line for the qq-plot which means it's gaussian. The fligner's test has a small p-value, indicating evidence against homogeneous variance.

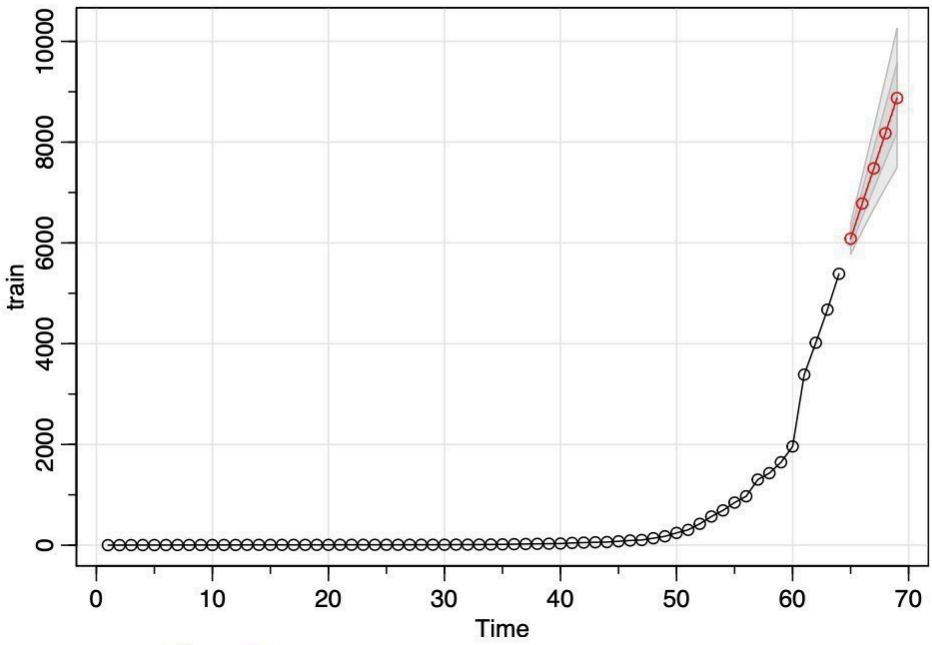
```
for1 <- sarima.for(train,n.ahead=5, p=3, d=2, q=0)
```




```
sum((test - for1$pred)^2)
```

```
## [1] 4540473
```

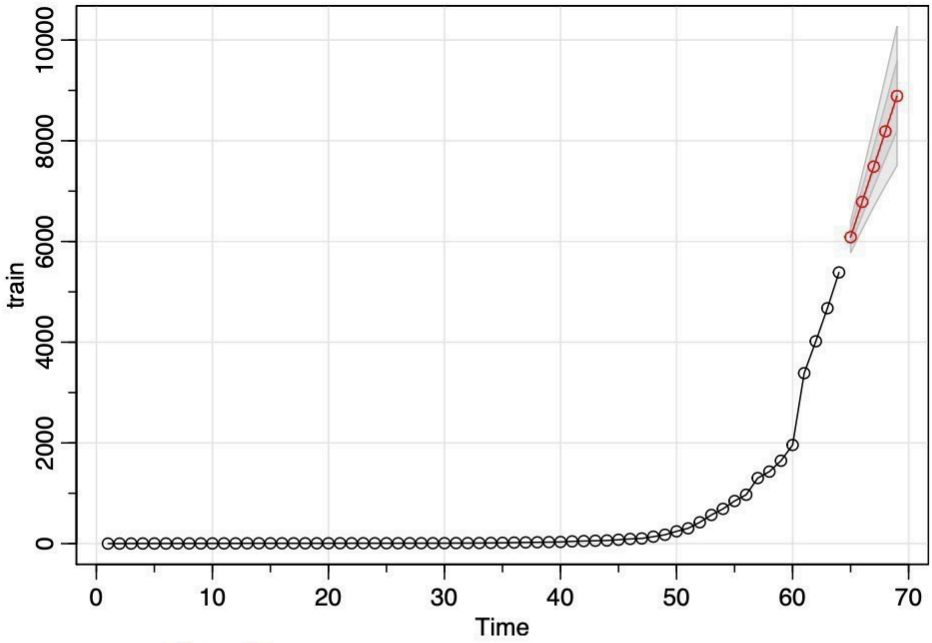
```
for2 <- sarima.for(train,n.ahead=5, p=0, d=2, q=1)
```




```
sum((test - for2$pred)^2)
```

```
## [1] 9298127
```

```
for3 <- sarima.for(train,n.ahead=5, p=1, d=2, q=1)
```



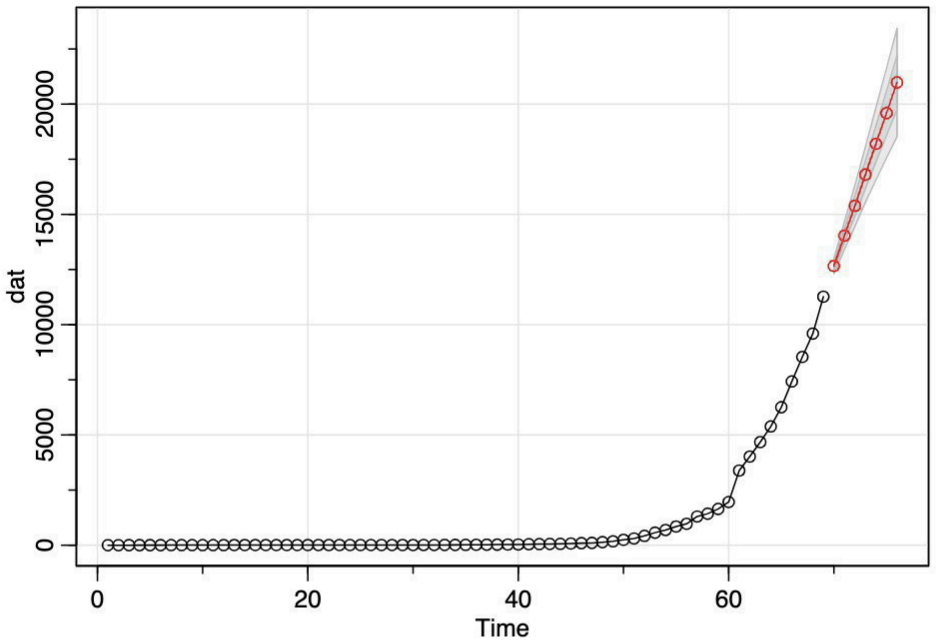

```
sum((test - for3$pred)^2)
```

```
## [1] 9181456
```

Based on PRESS score, ARMA(3,2,0) has the lowest at 4540473, so ARMA (3,2,0) has a better prediction power.

d)

```
for4 <- sarima.for(dat,n.ahead=7, p=3, d=2, q=0)
```



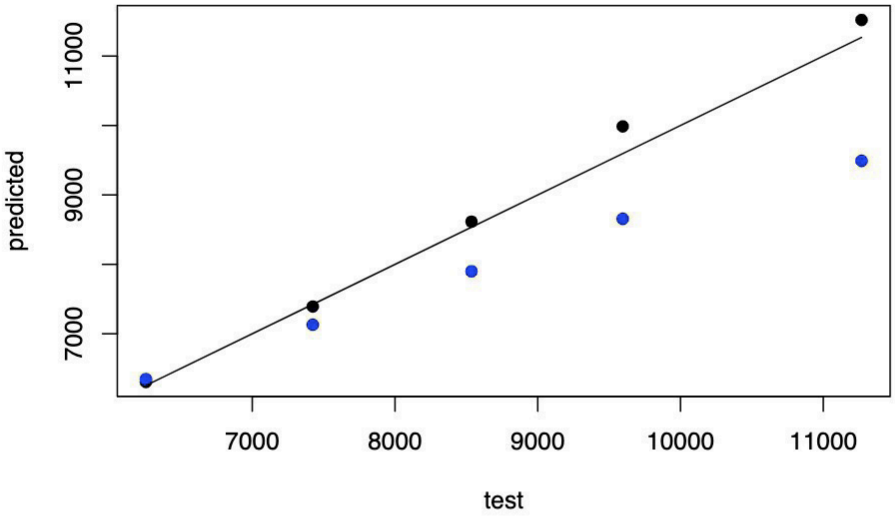

```
lower <- for4$pred-1.96*for4$se
upper <- for4$pred+1.96*for4$se
fit <- for4$pred


res = data.frame(lower, fit, upper)
res
```

```
##      lower      fit    upper
## 1 12319.82 12659.73 12999.63
## 2 13422.94 14036.40 14649.86
## 3 14475.73 15387.58 16299.43
## 4 15573.85 16805.07 18036.30
## 5 16601.95 18199.05 19796.14
## 6 17598.07 19591.47 21584.87
## 7 18561.82 20979.27 23396.72
```

e)

```
{
  plot(test, predict.model5, ylab="predicted", pch = 19)
  points(test, for1$pred, col="blue", pch=19)
  lines(test, test)
}
```



Based on the plot, the regression model appears to lie closer to the test data than the ARINM model.

PRESS Numbers:

```
sum((test - for1$pred)^2)
```

```
## [1] 4540473
```

ARIMA(1,2,1) has PRESS 9200174

```
sum((predict.model5 - test)^2)
```

```
## [1] 226635.4
```

Regression has PRESS 226635.4.

Based on the PRESS numbers, regression also has a smaller PRESS than the ARIMA model, showing regression predicts the dataset better.