Assignment 4

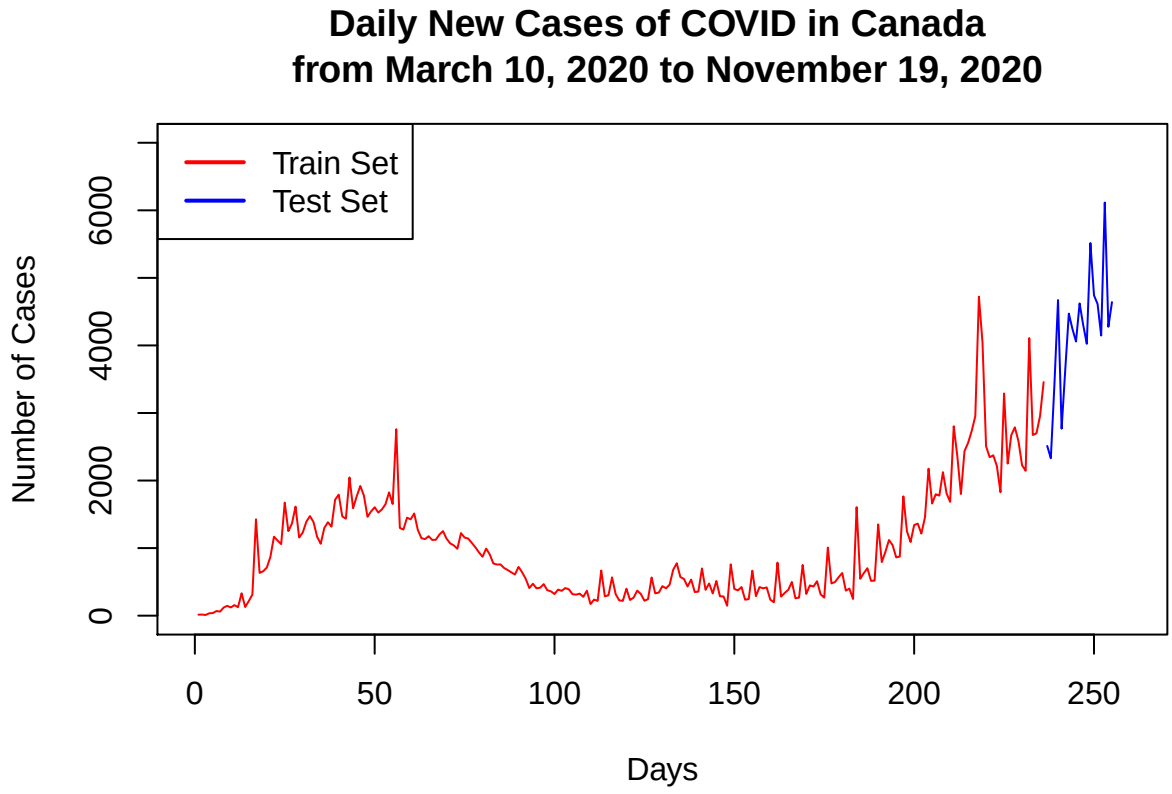
Rayan Roy

06/12/2021

1a:

```
covid <- read.csv("Canada_COVID19.csv")
covid$Time <- 1:nrow(covid)
covidTS <- ts(covid$NewCases,frequency = 7)
train_ind <- 1:236
trainset <- covid[train_ind,]
testset <- covid[-train_ind,]

plot(trainset$Time,trainset$NewCases, main = "Daily New Cases of COVID in Canada \n from March 10, 2020
     ylim = c(0, 7000), xlim = c(0, 260), col = "red", type = "l")
lines(testset$Time, testset$NewCases, type = "l", col ="blue")
legend("topleft", legend = c("Train Set", "Test Set"), col = c("red", "blue"),
       lwd = c(2,2))
```



1b:

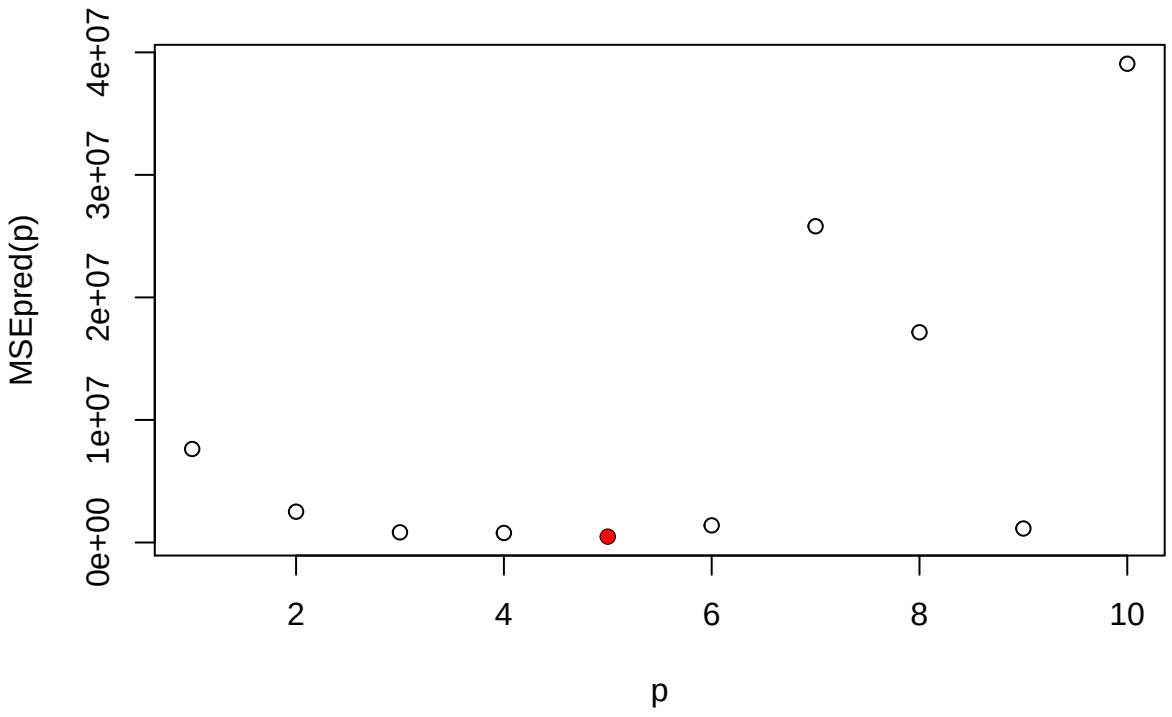
```
covid$Time <- 1:nrow(covid)
season <- as.factor(append(cycle(covidTS),c(4,5,6,7,rep(1:7,1))))

train_ind <- 1:236
trainset <- covid[train_ind,]
testset <- covid[-train_ind,]

MSE <- seq(1,10, by=1)
MSEs <- seq(1,10, by=1)
p <- seq(1,10, by=1)
ps <- seq(1,10,by=1)

for (i in 1:10) {
  model <- lm(NewCases ~ poly(Time, i) + season[Time], data = trainset)
  predictions <- predict(model, testset)
  MSEcalc <- mean((predictions - testset$NewCases)^2)
  MSEs[i] <- MSEcalc
}

plot(ps,MSEs, ylab = "MSEpred(p)", xlab ="p")
indx = which.min(MSEs)
points(indx,MSEs[indx], col="red",pch=16)
```

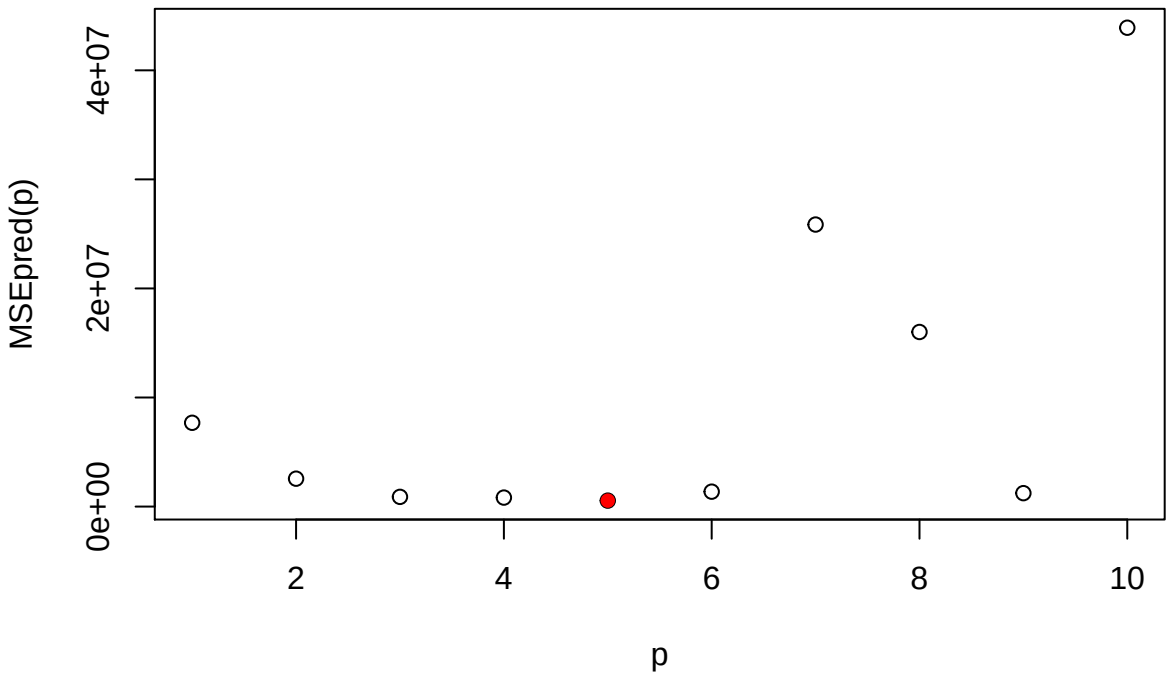



```
MSEs[indx]
```

```
## [1] 478910.7
```

```r
# MSE without seasonality
for (i in 1:10) {
  model <- lm(NewCases ~ poly(Time, i), data = trainset)
  predictions <- predict(model, testset)
  MSEcalc <- mean((predictions - testset$NewCases)^2)
  MSE[i] <- MSEcalc
}

plot(p,MSE, ylab = "MSEpred(p)",xlab ="p")
indx = which.min(MSE)
points(indx,MSE[indx], col = "red",pch=16)
```




```r
MSE[indx]
```

```
## [1] 545383.5
```

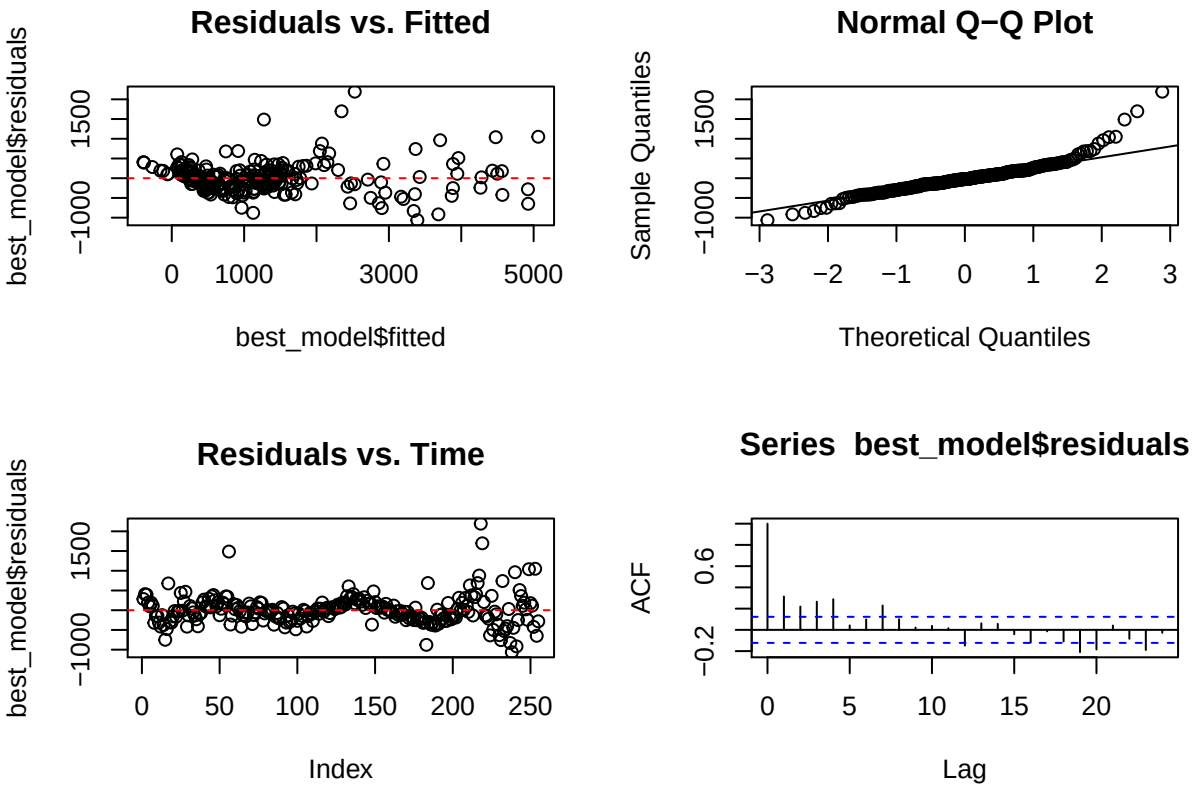
Based on the MSE vs p-graph, we can see that p = 5 gives the lowest MSE prediction on the test data for both the model containing seasonality and not containing seasonality. We can also see by adding the season term, the MSE score is much lower. Therefore the polynomial with degree 5 with the season term added is the best model based on prediction power.

1c:

As for model diagnostics, we will first generate some graphical residual diagnostics

```
best_model <- lm(NewCases ~ poly(Time, 5) + season[Time], data = covid)

# Diagnostic plots for reg model
par(mfrow = c(2, 2)) # Dividing the plotting page into 4 panels
plot(best_model$fitted, best_model$residuals, main = "Residuals vs. Fitted")
abline(h = 0, lty = 2, col = "red")
qqnorm(best_model$residuals) #qq-plot of residuals
qqline(best_model$residuals) # plotting the line where qq-plot should lie
plot(best_model$residuals, main = "Residuals vs. Time") # plotting the residuals vs time
abline(h = 0, lty = 2, col = "red") # plotting a horizontal line at 0
acf(best_model$residuals) #sample acf plot of residuals
```



From the plot of the residuals vs. fitted values, we can see a fanning out shape which implies that the variance of the residuals is not constant. The qq-plot shows that the distribution is heavy tailed since both ends of the qq-plot deviate from the straight line, which violates the normality assumption. The residuals vs. time plot shows a trend indicating that the mean of the residual is not constant at 0. The ACF plot of residuals depicts a few significant spikes indicating that there is correlation among the residuals in different lags.

We will now use Shapiro-Wilk Normality test and Fligner-Killeen test to check the model.

```
# Testing Normality and Homogenity of variance
resid = residuals(best_model) #extracting the residuals
shapiro.test(resid)
```

```
##
```

```
##  Shapiro-Wilk normality test
##
## data:  resid
## W = 0.89602, p-value = 2.987e-12
```

```
indx = factor(rep(1:5, each = 51)) #creating 10 chunks to test variance homogeneity
fligner.test(resid, indx)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  resid and indx
## Fligner-Killeen:med chi-squared = 50.489, df = 4, p-value = 2.854e-10
```

From the Shapiro-Wilk Test, we see that the p-value is close to 0 so we reject the null hypothesis that the residuals are normally distributed. Similarly, from the Fligner-Killeen Test, we see that the p-value is close to 0, confirming that the variance of the residual is not constant.

We will also use the Difference Sign Test and Runs Test

```
difference.sign.test(resid)
```

```
##
##  Difference Sign Test
##
## data:  resid
## statistic = -0.21651, n = 255, p-value = 0.8286
## alternative hypothesis: nonrandomness
```

```
runs.test(resid)
```

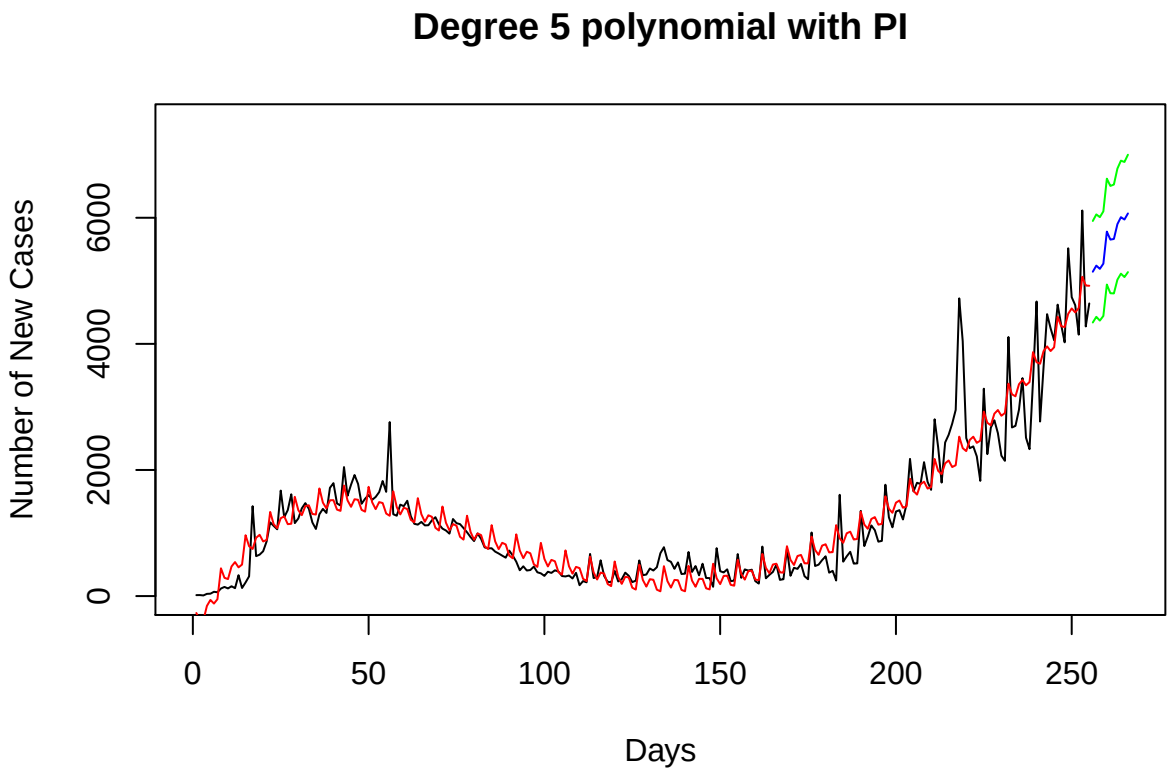
```
##
##  Runs Test
##
## data:  resid
## statistic = -7.0414, runs = 72, n1 = 127, n2 = 127, n = 254, p-value =
## 1.903e-12
## alternative hypothesis: nonrandomness
```

From the Runs test, we can see the p-value is very small, indicating that the residuals are not independent (or data is not random). Although the difference sign test is large with p-value = 0.8286, given other evidence, we still conclude that the residual is not independent.

1d:

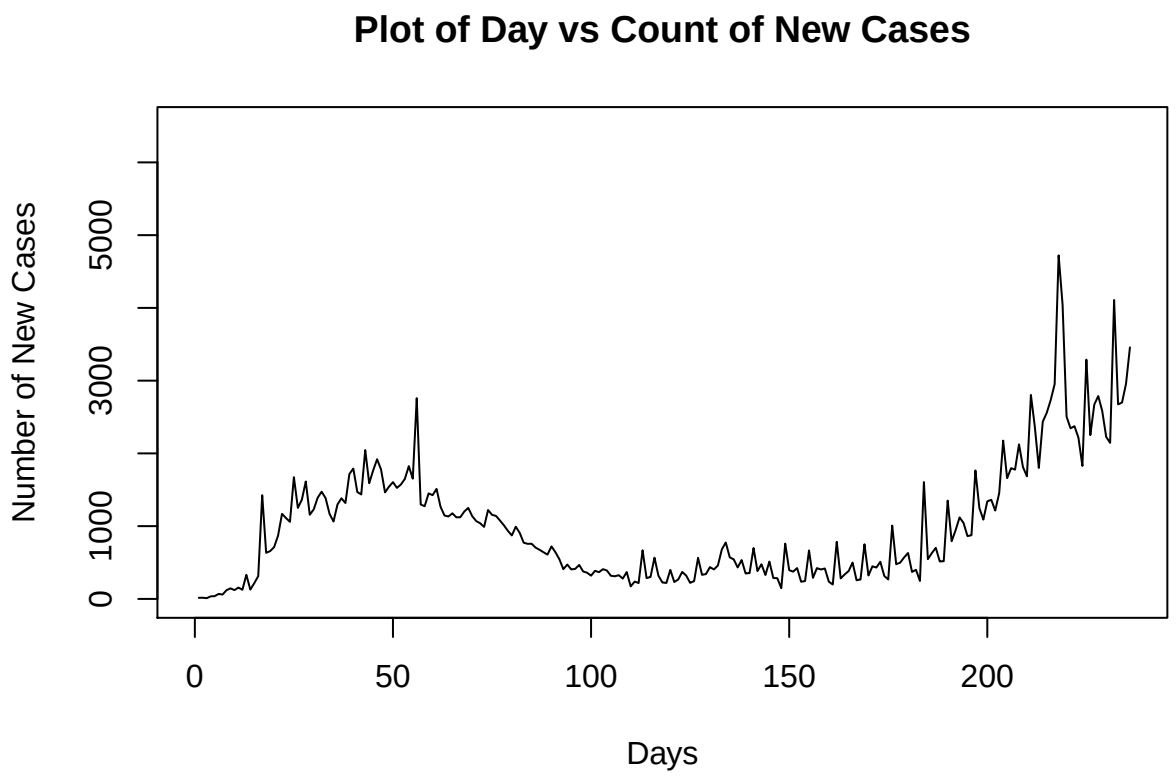
```
data <- data.frame(Time=seq(256,266, by=1))
predictval <- data.frame(predict.lm(best_model,data, interval="prediction"))

plot(covid$Time,covid$NewCases, main = "Degree 5 polynomial with PI",
     xlim = c(0, 266), ylim = c(0,7500), type = "l", xlab = "Days",
     ylab ="Number of New Cases")
points(covid$Time,predict.lm(best_model),type='l',col='red')
points(256:266, predictval[,1],type='l', col="blue")
points(256:266, predictval[,2],type='l', col="green")
points(256:266, predictval[,3],type='l', col="green")
```

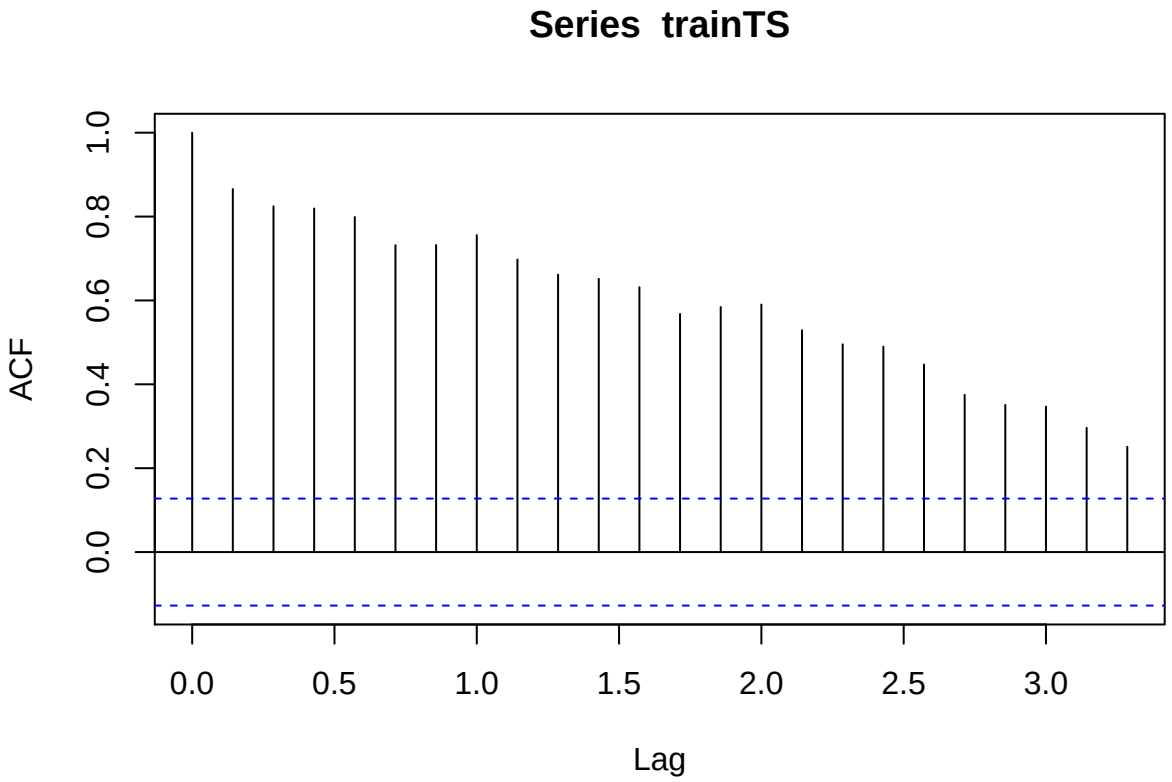


2a:

```
plot(trainset$Time,trainset$NewCases, main = "Plot of Day vs Count of New Cases",
     xlim = c(0, 236), ylim = c(0,6500), type = "l", xlab = "Days",
     ylab ="Number of New Cases")
```



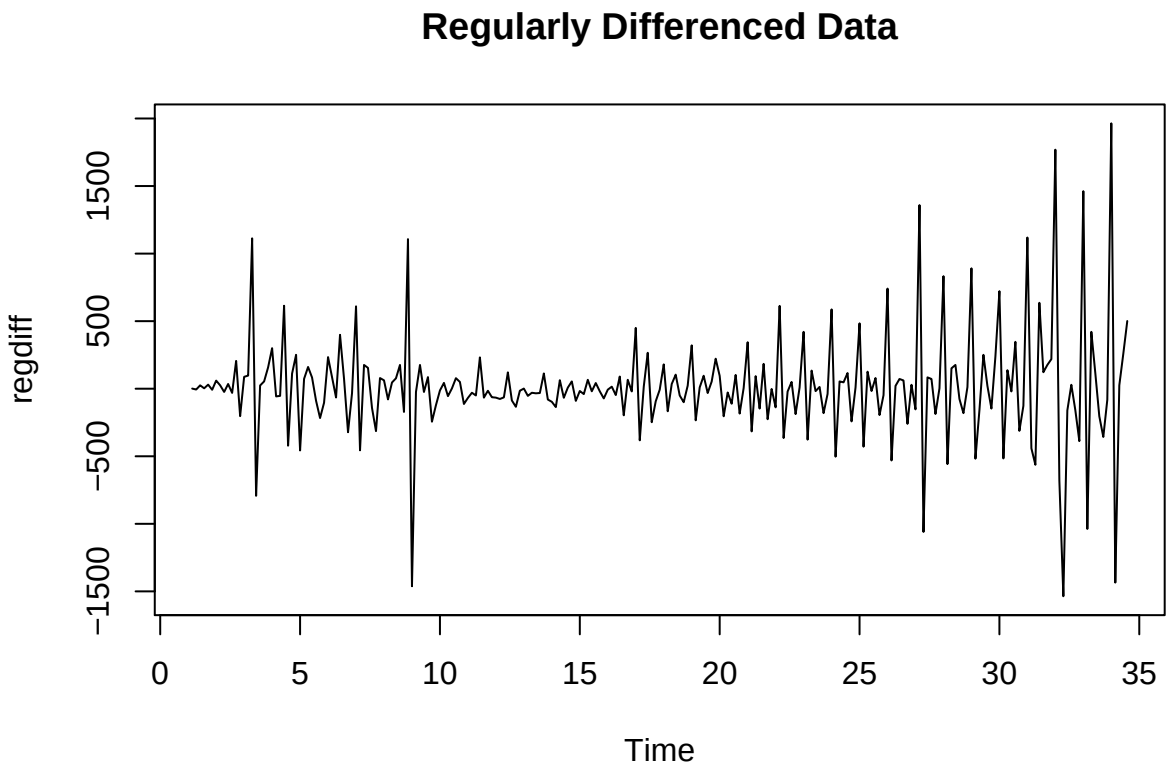

```
trainTS <- ts(trainset$NewCases,frequency = 7)
acf(trainTS)
```



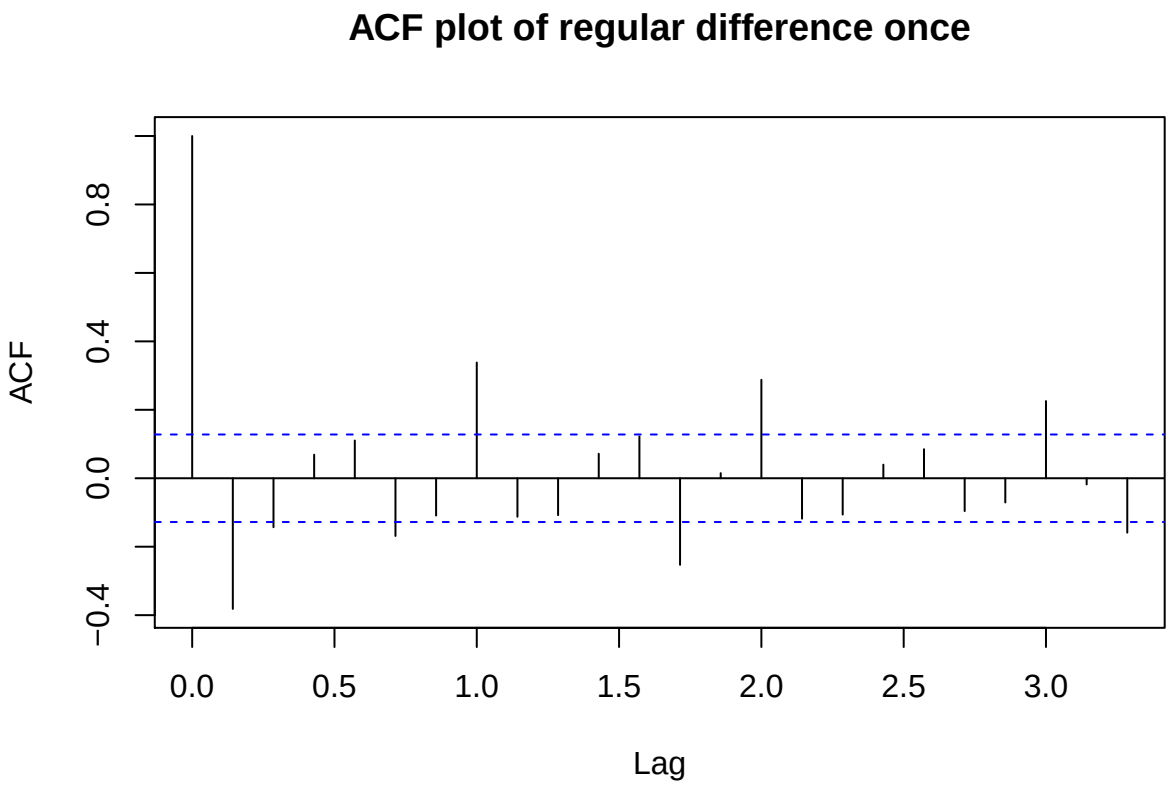
From the above time series plot, we can see the mean is not constant as there is trend in the data. We can also see a seasonal behaviour which indicates the time series is non-stationary. This is further confirmed by the ACF plot with a slow linear decay.

So we will perform a regular differencing,

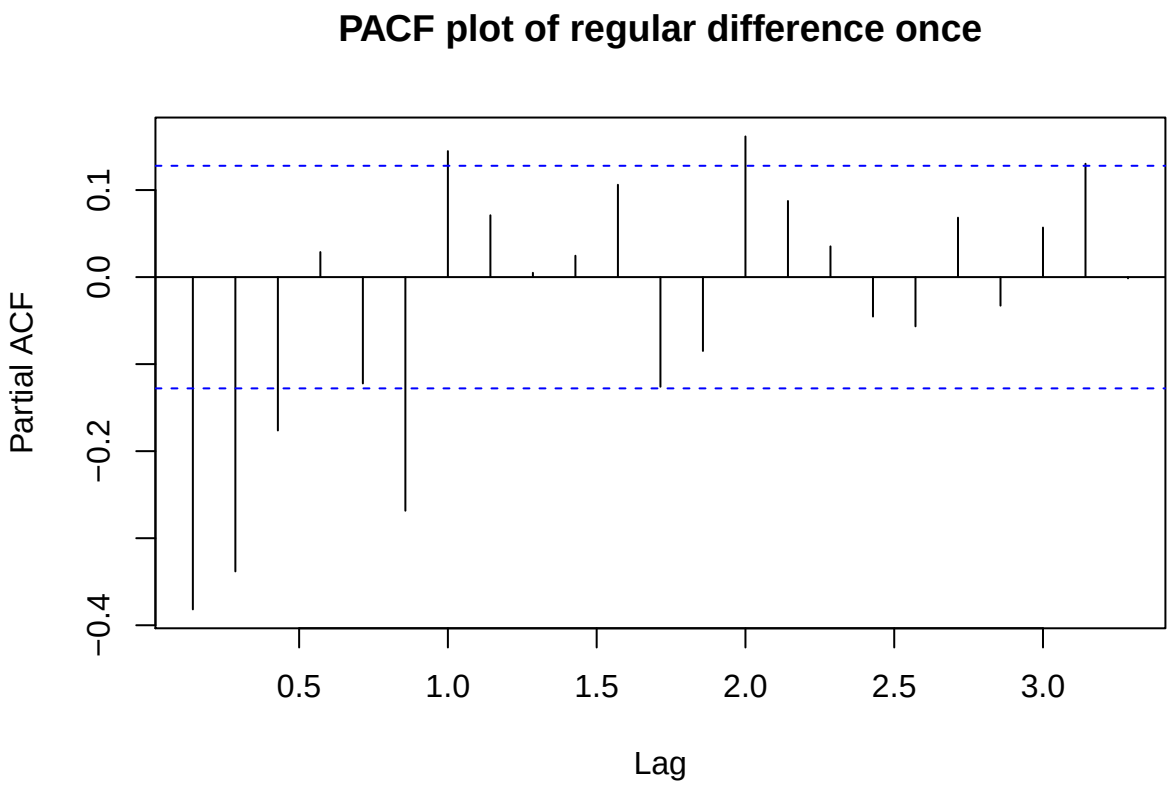
```
regdiff<- diff(trainTS)
plot(regdiff, main = "Regularly Differenced Data")
```




```
acf(regdiff, main="ACF plot of regular difference once")
```

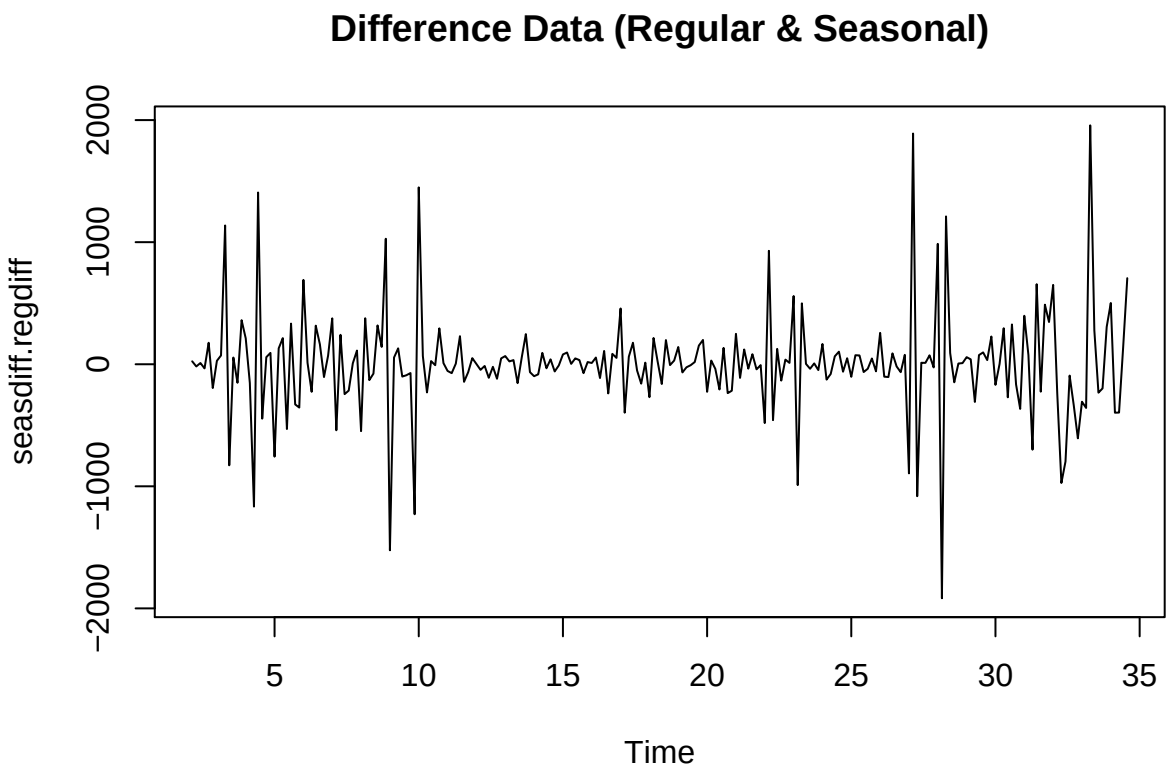


```r
pacf(regdiff,main="PACF plot of regular difference once")
```

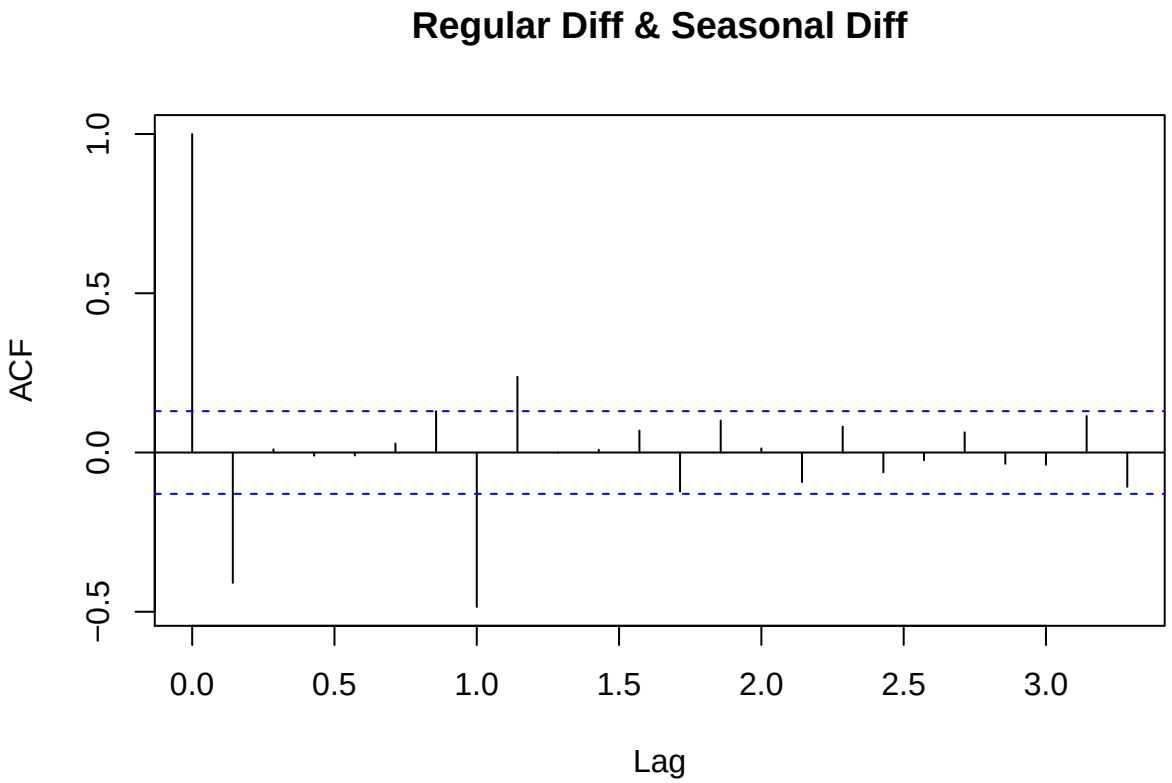


We can see the trend which was predicted by the linear decay is gone now. There is a periodic behaviour in the differenced data with period about 7 (since we added the season part), so we perform one time seasonal differencing in lag (7).

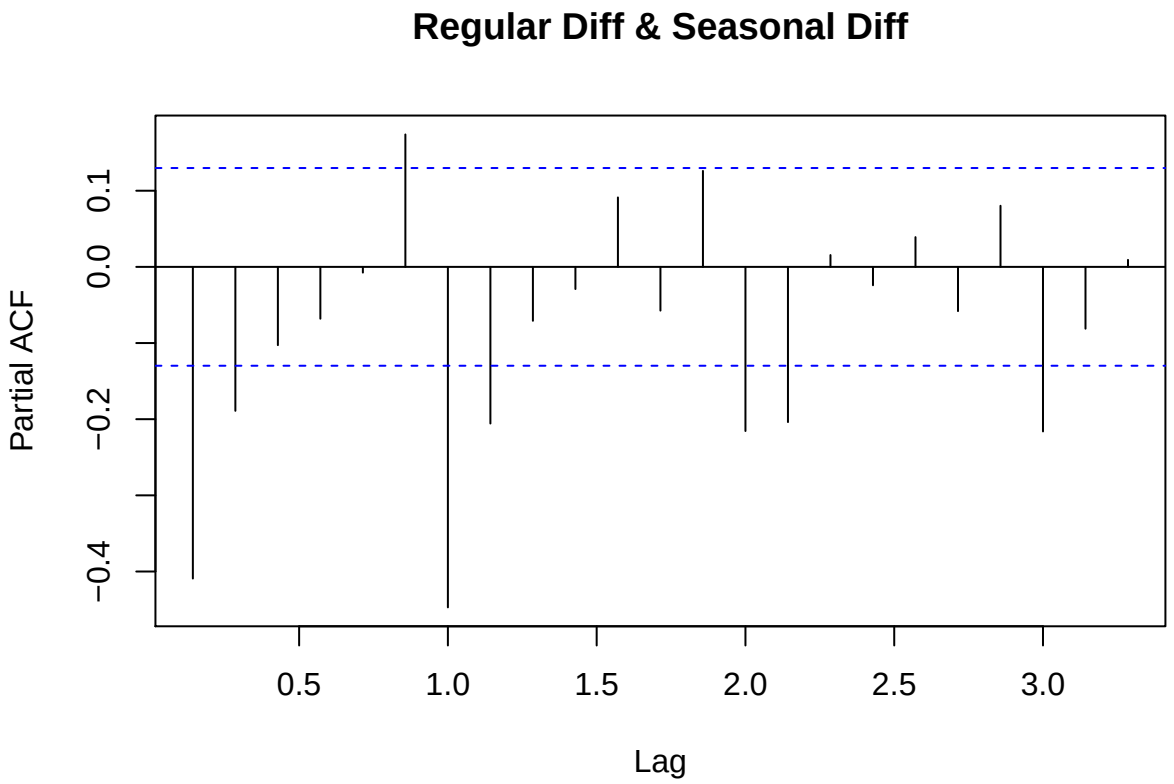
```r
seasdiff.regdiff <- diff(regdiff, lag = 7)
plot(seasdiff.regdiff, main = "Difference Data (Regular & Seasonal)")
```




```r
acf(seasdiff.regdiff, main="Regular Diff & Seasonal Diff")
```



```
pacf(seasdiff.regdiff, main="Regular Diff & Seasonal Diff")
```



We can see from the above ACF plot that there is no slow linear decay or periodicity left. From the time series plot we don't see any seasonality or trend. Hence, one regular differencing and one seasonal differencing (lag 7) leads to a stationary process.

2b:

The two candidate SARIMA models are:

- SARIMA(1,1,1) x $(0, 1, 2)_7$ From the regular differencing, we can say that there is cutoff after lag 1 in ACF plot and cutoff after lag 1 in PACF plot leading to (1,1,1) while from the seasonal differencing we can see there is cutoff after lag 2 in ACF plot but a fast exponential decay in PACF plot leading to $(0, 1, 2)_7$.
- SARIMA(0,1,1) x $(3, 1, 0)_7$ From the regular differencing, we can say that the PACF plot shows sinusodial pattern and there is cutoff after lag 1 in ACF plot leading to (0,1,1) while from the seasonal differencing we can see there is cutoff after lag 3 in the PACF but a fast exponential decay in ACF plot, leading to $(3, 1, 0)_7$.

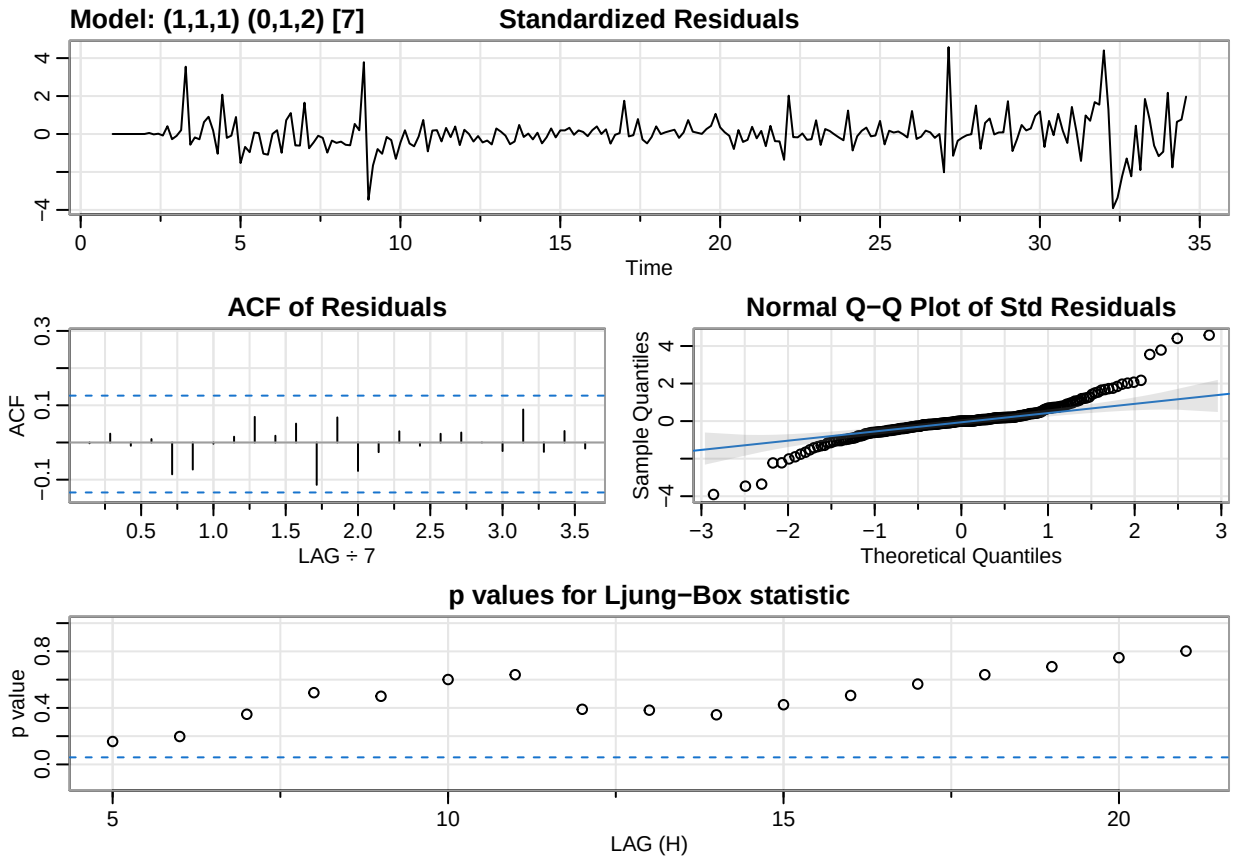
2c:

Fitting in the first proposed model, we get:

SARIMA(1,1,1) x $(0, 1, 2)_7$

```
model1 = sarima(trainTS, p=1,d=1,q=1,P=0,D=1,Q=2,S=7)
```

```
## initial  value 6.068643
## iter   2 value 5.796603
## iter   3 value 5.761849
## iter   4 value 5.732571
## iter   5 value 5.704813
## iter   6 value 5.703948
## iter   7 value 5.698527
## iter   8 value 5.696782
## iter   9 value 5.695467
## iter  10 value 5.694467
## iter  11 value 5.694441
## iter  12 value 5.694313
## iter  13 value 5.694310
## iter  14 value 5.694291
## iter  15 value 5.694290
## iter  16 value 5.694289
## iter  16 value 5.694289
## iter  16 value 5.694289
## final  value 5.694289
## converged
## initial  value 5.704278
## iter   2 value 5.703813
## iter   3 value 5.703784
## iter   4 value 5.703778
## iter   5 value 5.703777
## iter   6 value 5.703776
## iter   7 value 5.703776
## iter   8 value 5.703776
## iter   8 value 5.703776
## iter   8 value 5.703776
## final  value 5.703776
## converged
```




```
# AIC, AICs, BIC
c(model1$AIC,model1$AICc,model1$BIC)
```

```
## [1] 14.28929 14.29008 14.36449
```

Looking at the Standard Residuals plot, we see no trend or no seasonality in the data. From the ACF of the Residuals and Ljung Box statistic, we can see there is no serial correlation among residuals. From the QQ-plot, the residuals seems to be normal.

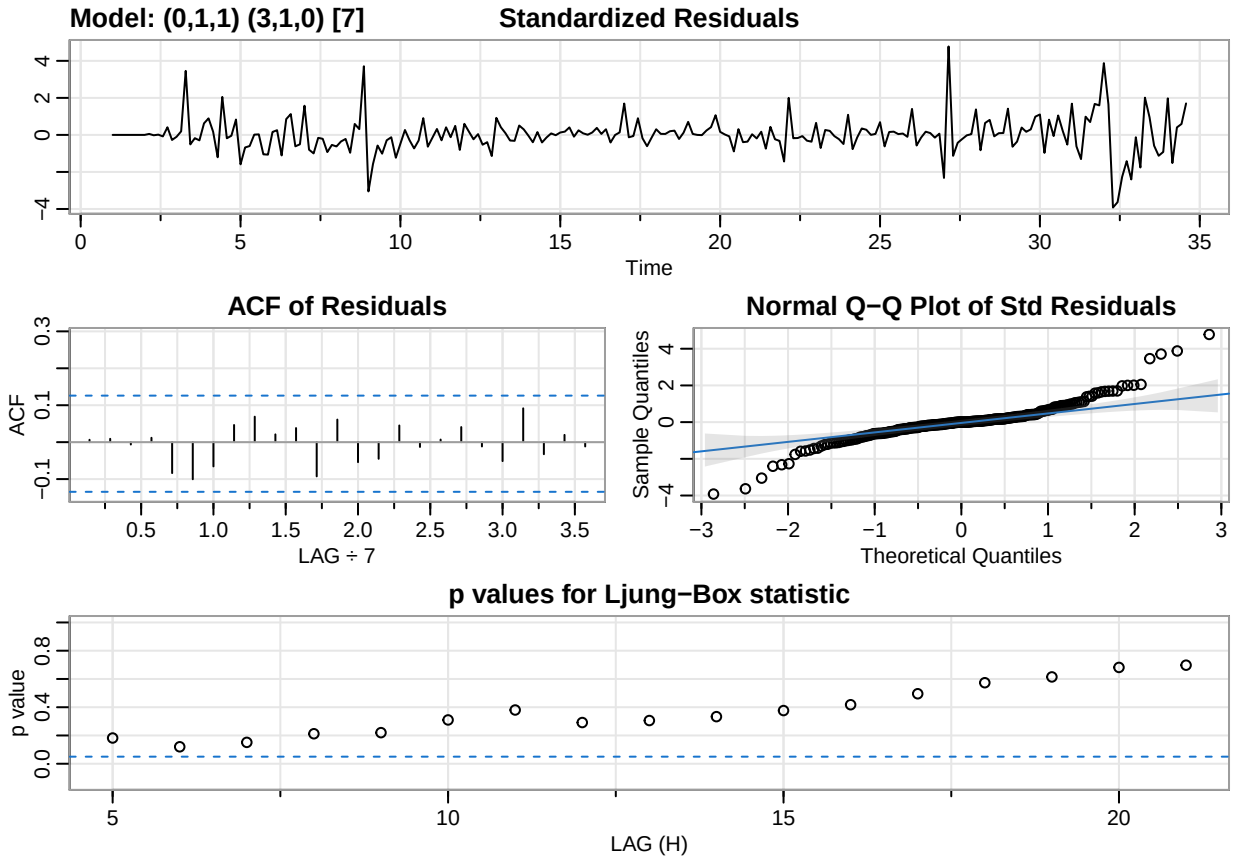
Fitting in the second proposed model, we get:

SARIMA(0,1,1) x $(3,1,0)_7$

```
model2 = sarima(trainTS, p=0,d=1,q=1,P=3,D=1,Q=0,S=7)
```

```
## initial  value 6.032862
## iter   2 value 5.797292
## iter   3 value 5.729837
## iter   4 value 5.697173
## iter   5 value 5.688235
## iter   6 value 5.687683
## iter   7 value 5.687634
## iter   8 value 5.687632
## iter   8 value 5.687632
```

```
## iter   8 value 5.687632
## final  value 5.687632
## converged
## initial  value 5.708493
## iter   2 value 5.708307
## iter   3 value 5.708226
## iter   4 value 5.708223
## iter   5 value 5.708223
## iter   5 value 5.708223
## iter   5 value 5.708223
## final  value 5.708223
## converged
```




```
# AIC, AICs, BIC
c(model2$AIC, model2$AICc, model2$BIC)
```

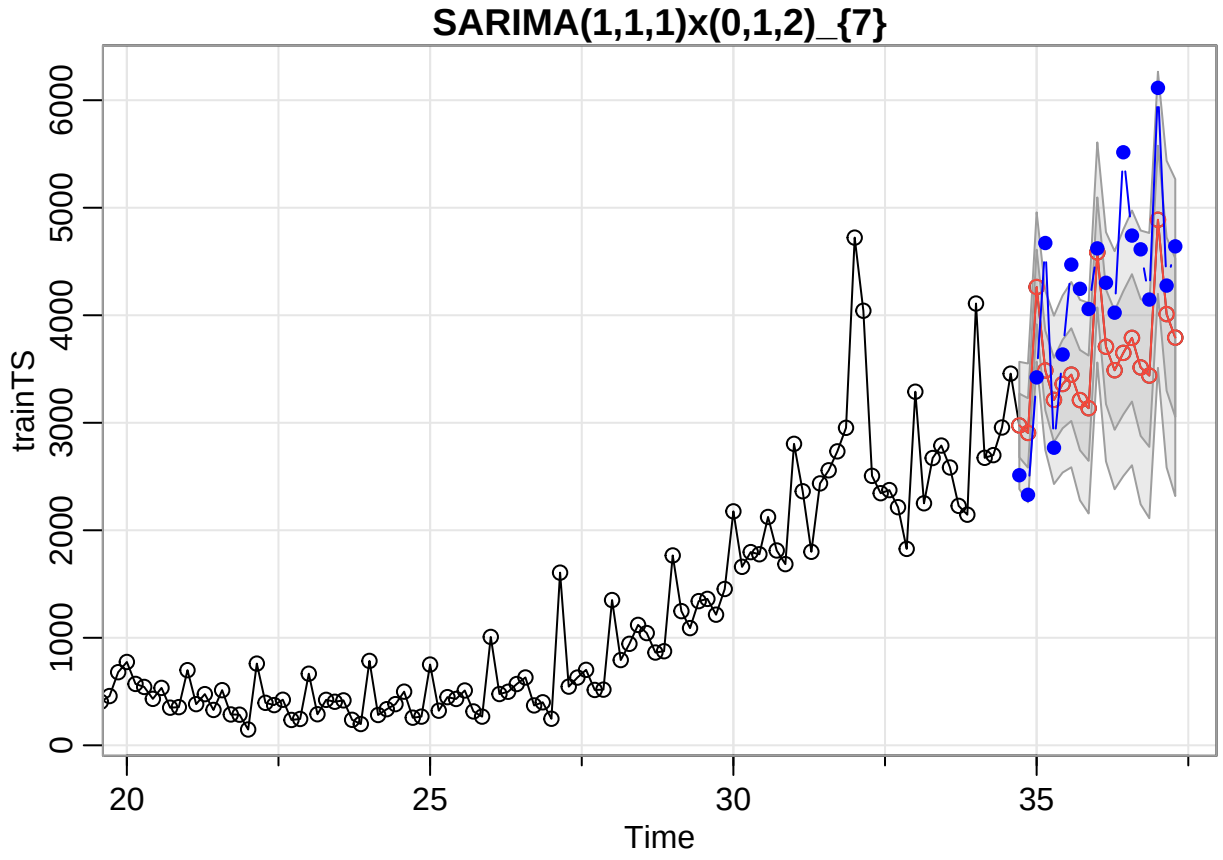
```
## [1] 14.29818 14.29897 14.37339
```

Looking at the Standard Residuals plot, we see no trend or seasonality in the data. From the ACF of the Residuals and Ljung Box statistic, we can see there is no serial correlation among residuals. From the QQ-plot, the residuals seems to be normal.

We can see AIC, AICc and BIC scores of both the models are very close to each other.

2d:

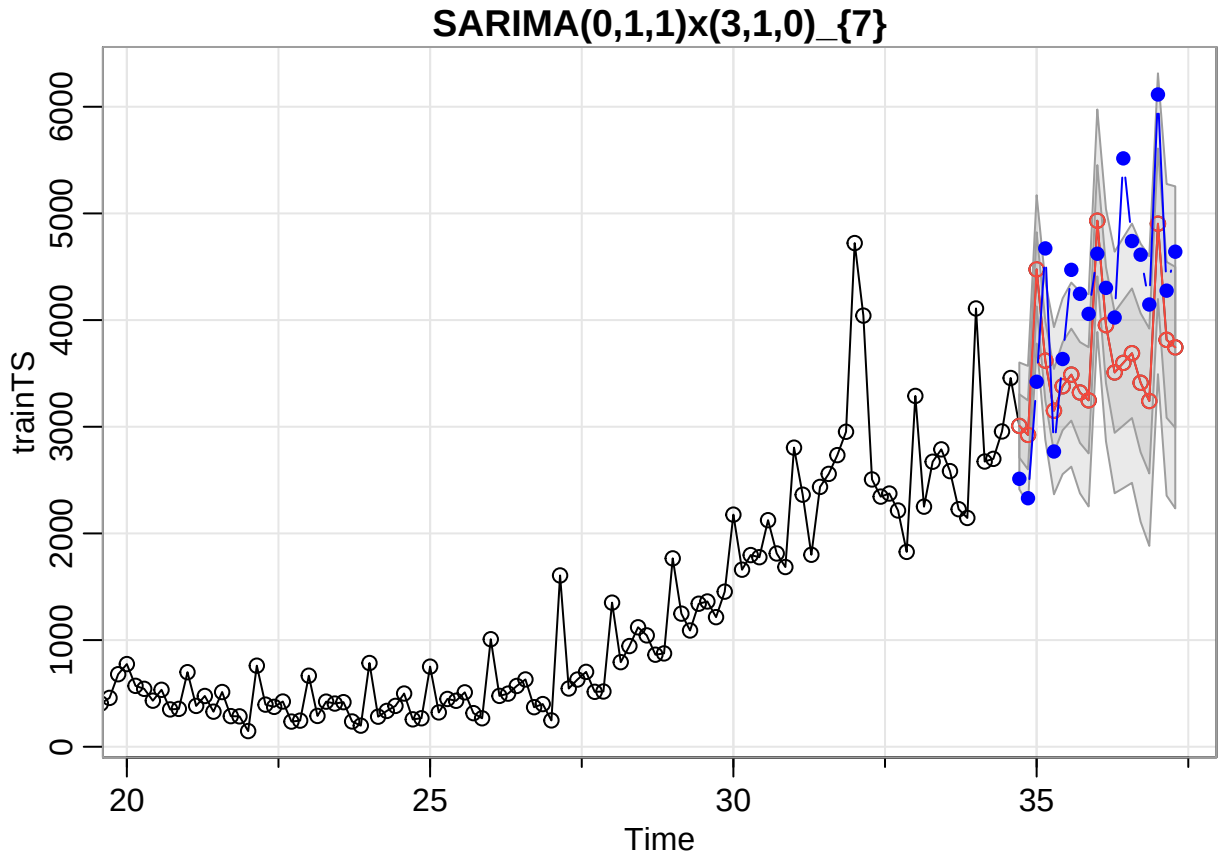
```
tim <- time(covidTS)
fore1 <- sarima.for(trainTS, n.ahead=19,
                    p=1,d=1,q=1,P=0,D=1,Q=2,S=7)
title("SARIMA(1,1,1)x(0,1,2)_{7}")
lines(tim[237:255],testset$NewCases,col='blue',type='b',pch=16)
```




```
mean((fore1$pred-testset$NewCases)^2)
```

```
## [1] 785856.4
```

```
fore2 <- sarima.for(trainTS, n.ahead=19,
                    p=0,d=1,q=1,P=3,D=1,Q=0,S=7)
title("SARIMA(0,1,1)x(3,1,0)_{7}")
lines(tim[237:255],testset$NewCases,col='blue',type='b',pch=16)
```



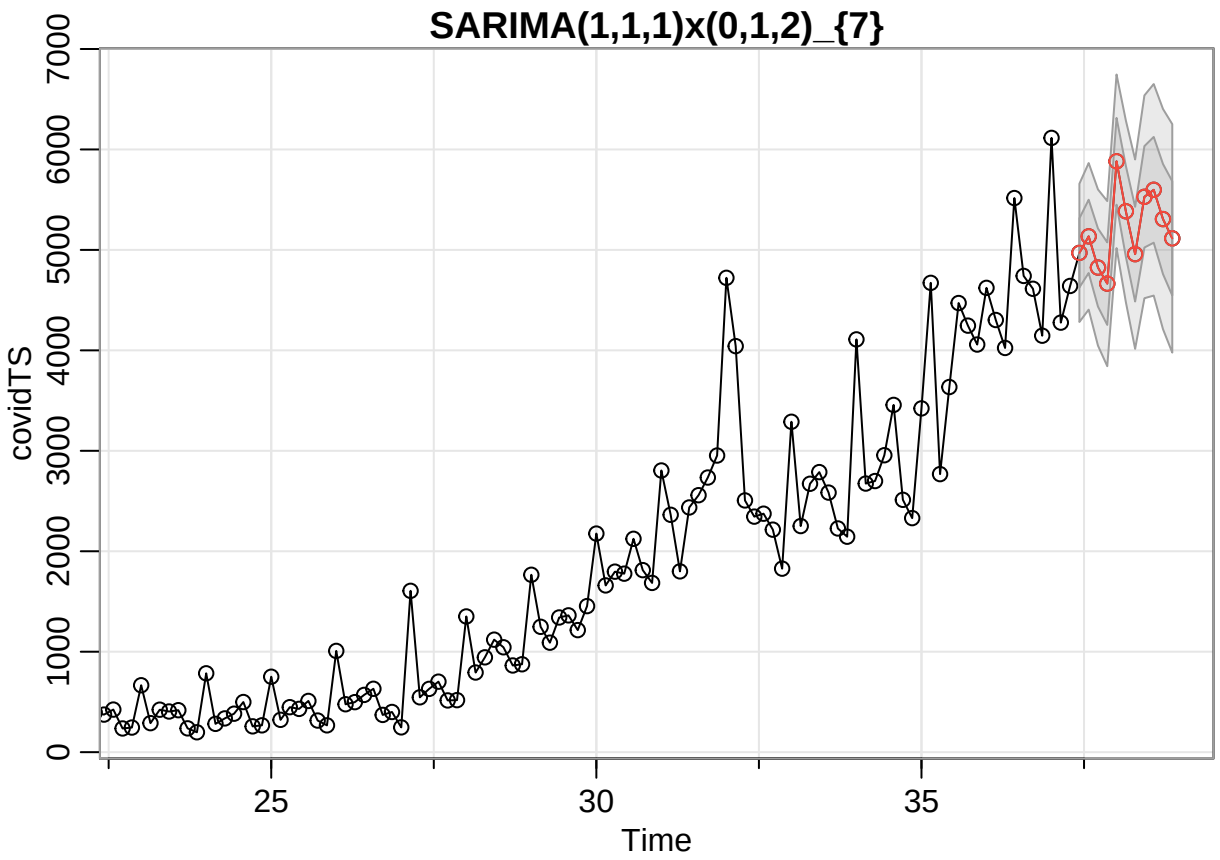

```
mean((fore2$pred-testset$NewCases)^2)
```

```
## [1] 816449.5
```

Based on the prediction power, model1 seems to be better with a lower MSE_pred. Hence the chosen model is SARIMA(1,1,1) x $(0, 1, 2)_7$

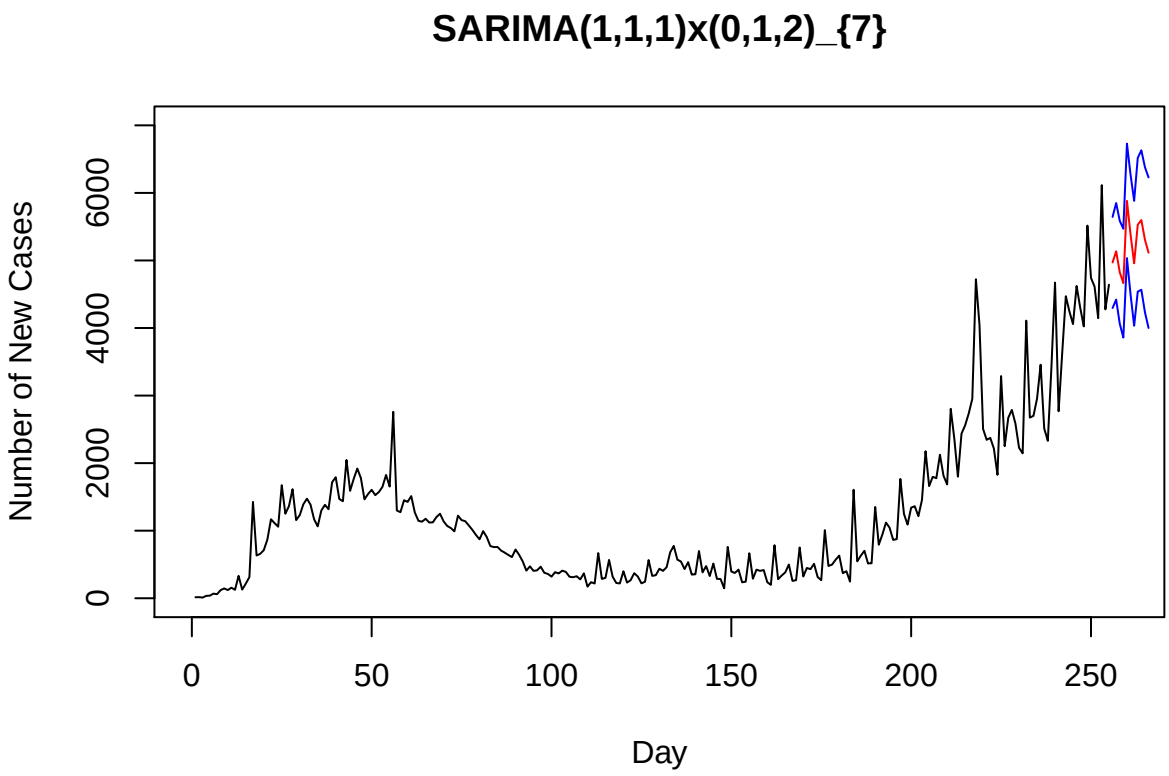
2e:

```
future.forecast <- sarima.for(covidTS, n.ahead=11,
                              p=1,d=1,q=1,P=0,D=1,Q=2,S=7)
title("SARIMA(1,1,1)x(0,1,2)_{7}")
```



We can see the plot of the whole dataset which includes the predicted values and the 95% prediction interval. We will show another plot since the plot above cuts off a little bit in the beginning.

```
lower <- future.forecast$pred-1.96*future.forecast$se
upper <- future.forecast$pred+1.96*future.forecast$se
fit <- future.forecast$pred
yband <- c(0,1000)
plot(covid$NewCases,type = "l",xlab = "Day", ylim = c(0,7000), xlim = c(0,260),
       ylab="Number of New Cases",main='SARIMA(1,1,1)x(0,1,2)_{7}')
points(256:266,fit,col='red',type='l')
points(256:266,lower,col='blue',type ="l")
points(256:266,upper,col='blue',type ="l")
```


SARIMA(1,1,1)x(0,1,2)_{7}
Number of New Cases
0
2000
4000
6000
0
50
100
150
200
250
Day

3a:

We will fit all 4 model namely Simple exponential smoothing, Double Exponential smoothing, Additive Holt-Winters and Multiplicative Holt-Winters model.

```
es <- HoltWinters(trainTS, gamma = FALSE , beta = FALSE) # Simple exponential smoothing
es
```

```
## Holt-Winters exponential smoothing without trend and without seasonal component.
##
## Call:
## HoltWinters(x = trainTS, beta = FALSE, gamma = FALSE)
##
## Smoothing parameters:
##  alpha: 0.3731889
##  beta : FALSE
##  gamma: FALSE
##
## Coefficients:
##       [,1]
## a 3087.364
```

```
des <- HoltWinters(trainTS, gamma = FALSE) # Double Exponential smoothing
des
```

```
## Holt-Winters exponential smoothing with trend and without seasonal component.
##
## Call:
## HoltWinters(x = trainTS, gamma = FALSE)
##
## Smoothing parameters:
##  alpha: 0.3262693
##  beta : 0.03470391
##  gamma: FALSE
##
## Coefficients:
##         [,1]
## a 3117.79759
## b   38.41223
```

```
hw.ad <- HoltWinters(trainTS, seasonal = "additive") #Additive HW method
hw.ad
```

```
## Holt-Winters exponential smoothing with trend and additive seasonal component.
##
## Call:
## HoltWinters(x = trainTS, seasonal = "additive")
##
## Smoothing parameters:
##  alpha: 0.4024811
##  beta : 0.0248438
##  gamma: 0.3097384
##
```

```
## Coefficients:
##          [,1]
## a  3181.09259
## b    35.95480
## s1 -221.46140
## s2 -248.68744
## s3  921.11056
## s4   48.15694
## s5 -178.31992
## s6  -44.99647
## s7   49.06300
```

```
hw.mul <- HoltWinters(trainTS, seasonal = "multiplicative") #Multiplicative HW method
hw.mul
```

```
## Holt-Winters exponential smoothing with trend and multiplicative seasonal component.
##
## Call:
## HoltWinters(x = trainTS, seasonal = "multiplicative")
##
## Smoothing parameters:
##  alpha: 0.3613207
##  beta : 0
##  gamma: 0.4063899
##
## Coefficients:
##            [,1]
## a  4707.9473249
## b    18.4030612
## s1    0.5684682
## s2    0.5302421
## s3    0.9293855
## s4    0.6764661
## s5    0.6216621
## s6    0.6578074
## s7    0.6876572
```

3b:

```
HW.predictes = predict(es, n.ahead = 19)
MSEes <- mean((HW.predictes-testset$NewCases)^2)
MSEes
```

```
## [1] 1995296
```

```
HW.predictdes = predict(des, n.ahead= 19)
MSEdes <- mean((HW.predictdes-testset$NewCases)^2)
MSEdes
```

```
## [1] 1048089
```

```
HW.A = predict(hw.ad, n.ahead= 19)
MSEad <- mean((HW.A-testset$NewCases)^2)
MSEad
```

```
## [1] 845134.6
```

```
HW.MUL = predict(hw.mul, n.ahead= 19)
MSEmul <- mean((HW.MUL-testset$NewCases)^2)
MSEmul
```

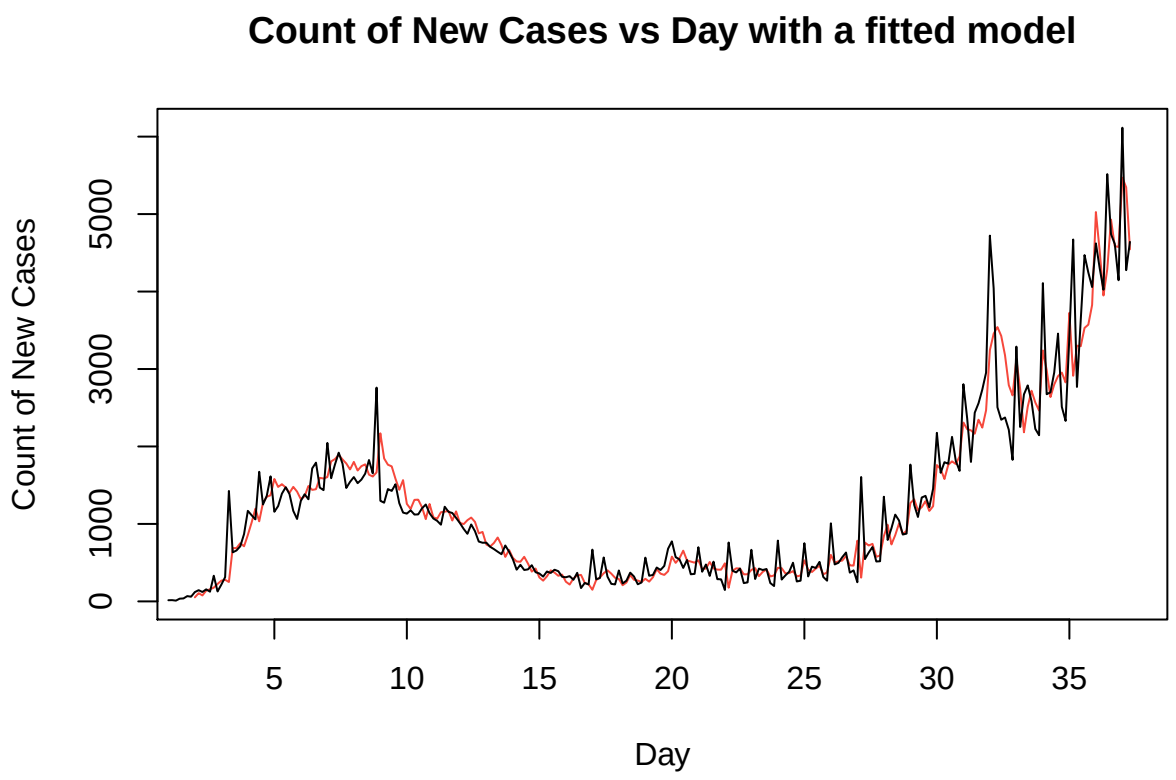
```
## [1] 1491992
```

Looking at the MSE of all 4 models, the Additive Holt-Winters model seems to be the best model with lowest MSE score (845134.6).

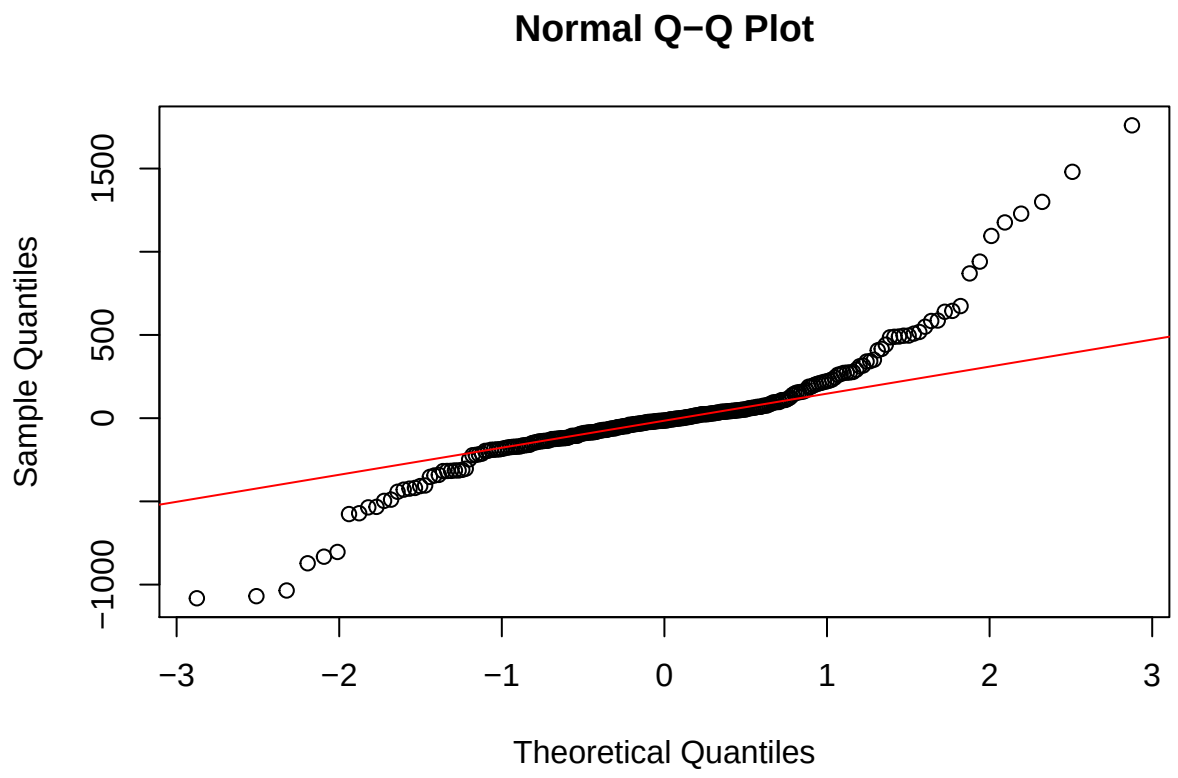
3c:

From part(b), we said the Additive Holt-Winters model was the best model, so we will fit it on the whole dataset.

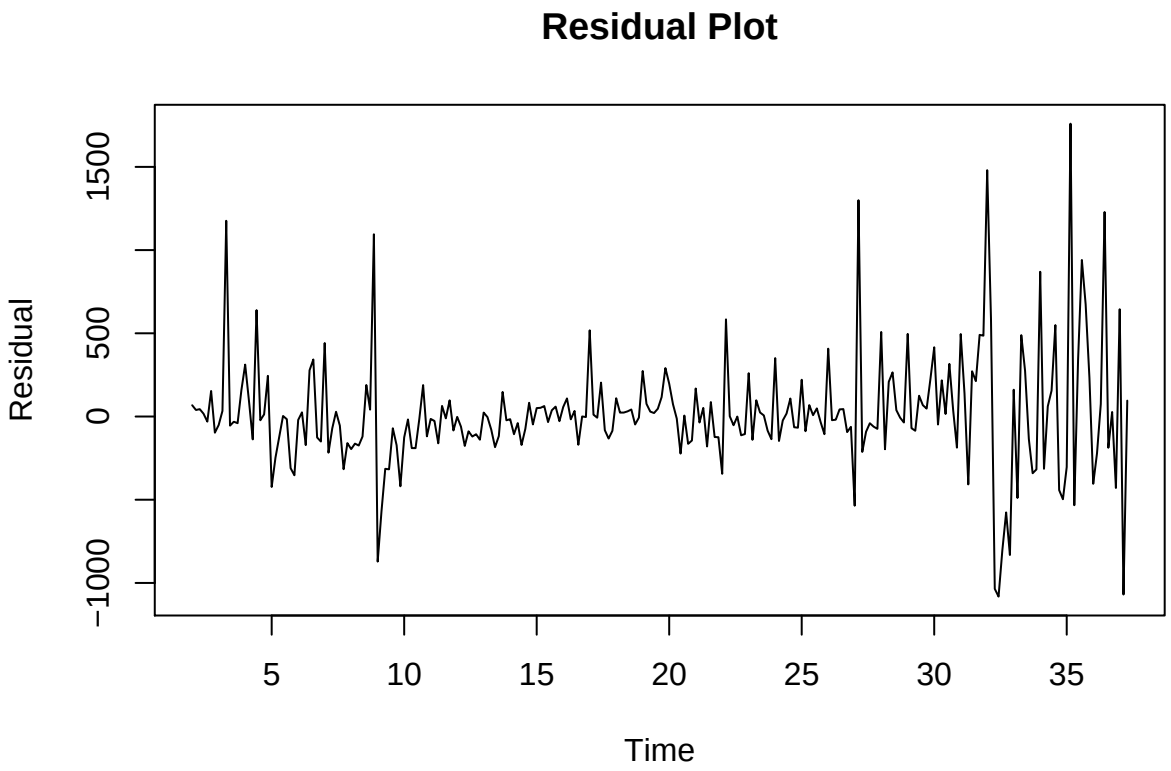
```
mbest <- HoltWinters(covidTS, seasonal = "additive")
plot(mbest, ylab = "Count of New Cases", xlab = "Day",
     main = "Count of New Cases vs Day with a fitted model")
```



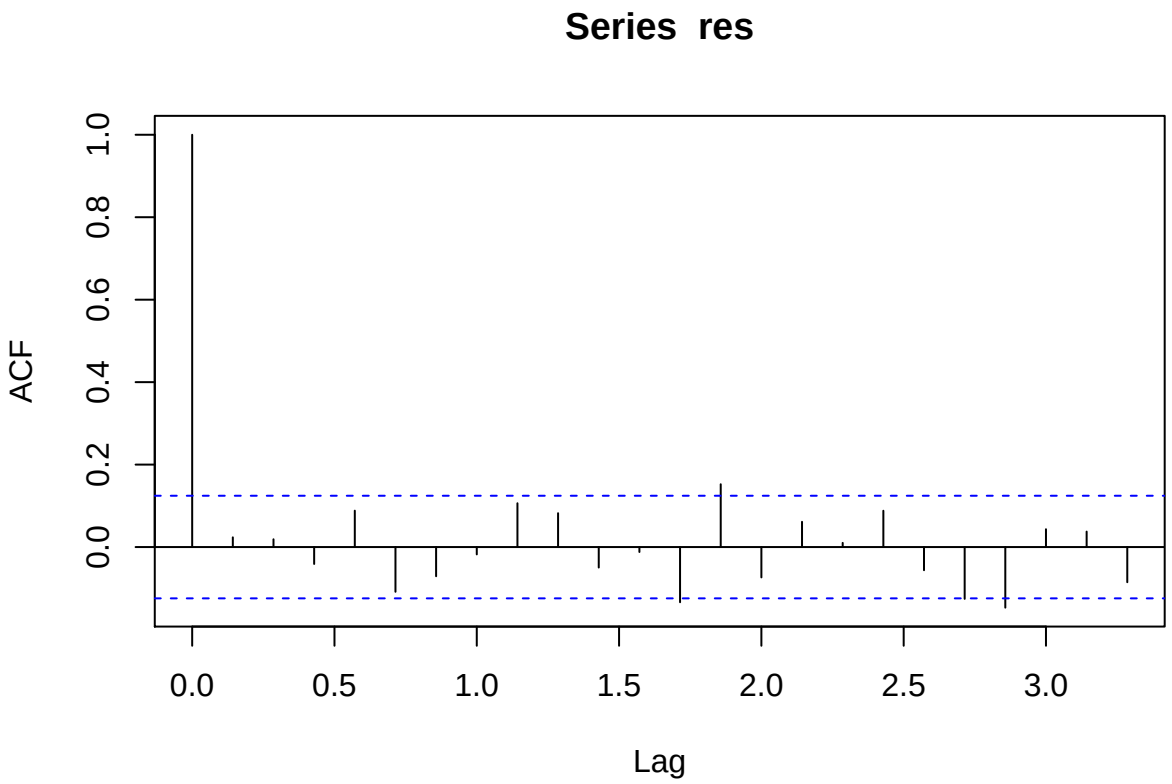

```
res = residuals(mbest)
qqnorm(res)
qqline(res, col = "red")
```




```r
plot(res , type="l", pch=16, ylab = "Residual", main = "Residual Plot")
```



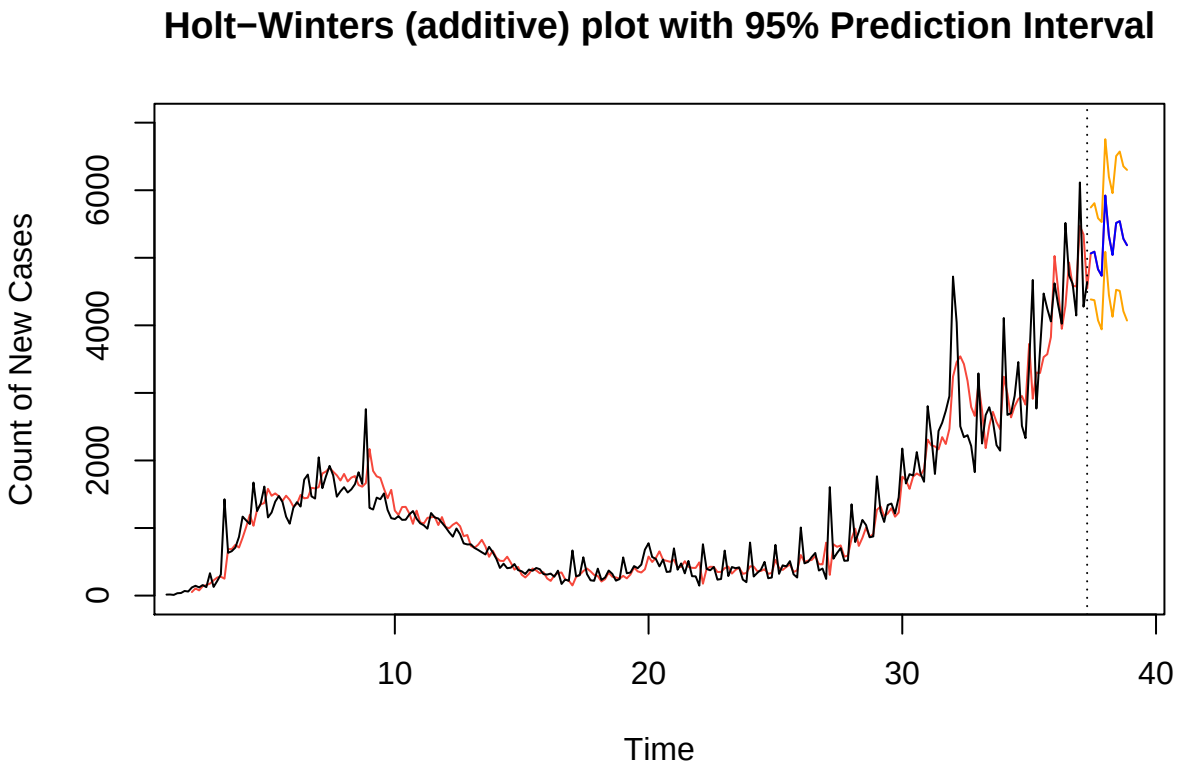
```r
acf(res)
```



From the time-series plot of residual, we can see no sign of trend or periodic behaviour. In the ACF plot, we don't see any significant spikes, indicating that the residuals are not correlated. However, from the QQ-plot, we see the points deviate from the straight line and contain heavy tails indicating that the residuals are not normally distributed. As a result, the residuals of the regression model are not realizations of Gaussian white noise.

3d:

```
hw.predict <- predict(mbest, 11 , prediction.interval = TRUE, level=0.95)
plot(mbest, predict(mbest, n.ahead=11), ylim = c(0,7000),ylab = "Count of New Cases"
     ,main="Holt-Winters (additive) plot with 95% Prediction Interval")
lines(hw.predict[,1], type ="l", col="blue")
lines(hw.predict[,2], type ="l", col="orange")
lines(hw.predict[,3], type = "l", col="orange")
```



3e:

```
# Regression model
MSEs[5]
```

```
## [1] 478910.7
```

```
# Model from SARIMA
mean((fore1$pred-testset$NewCases)^2)
```

```
## [1] 785856.4
```

```
# Model from Holt-Winters
MSEad
```

```
## [1] 845134.6
```

Looking at the prediction power between the regression model from Q1(b), the SARIMA model of Q2(d), and the additive Holt-Winters models from 3(d), the Regression model from Q1(b) (i.e Regression with degree 5 polynomial and including season) is the best model with lowest MSE score.