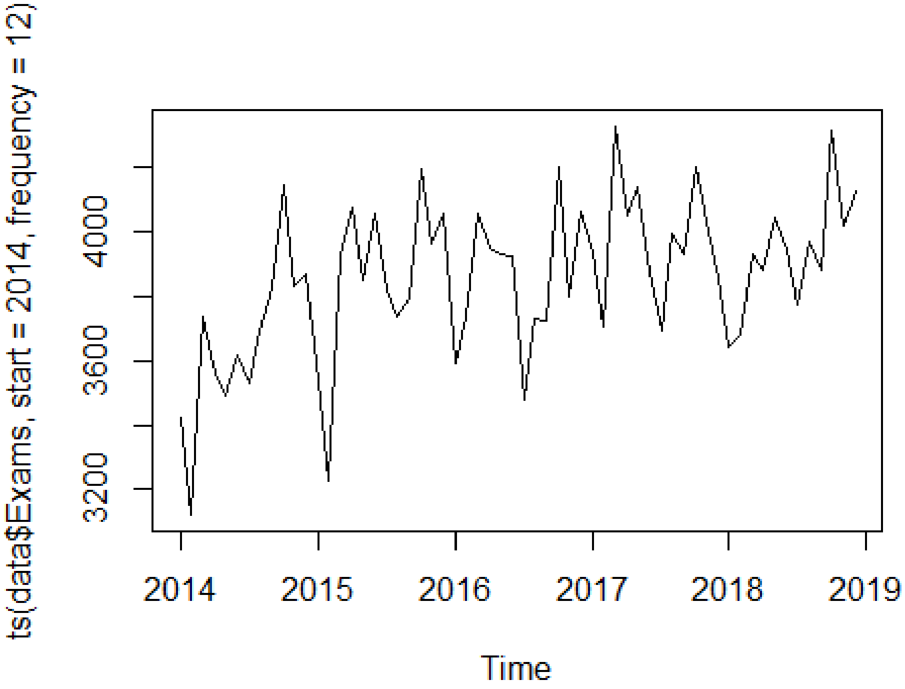
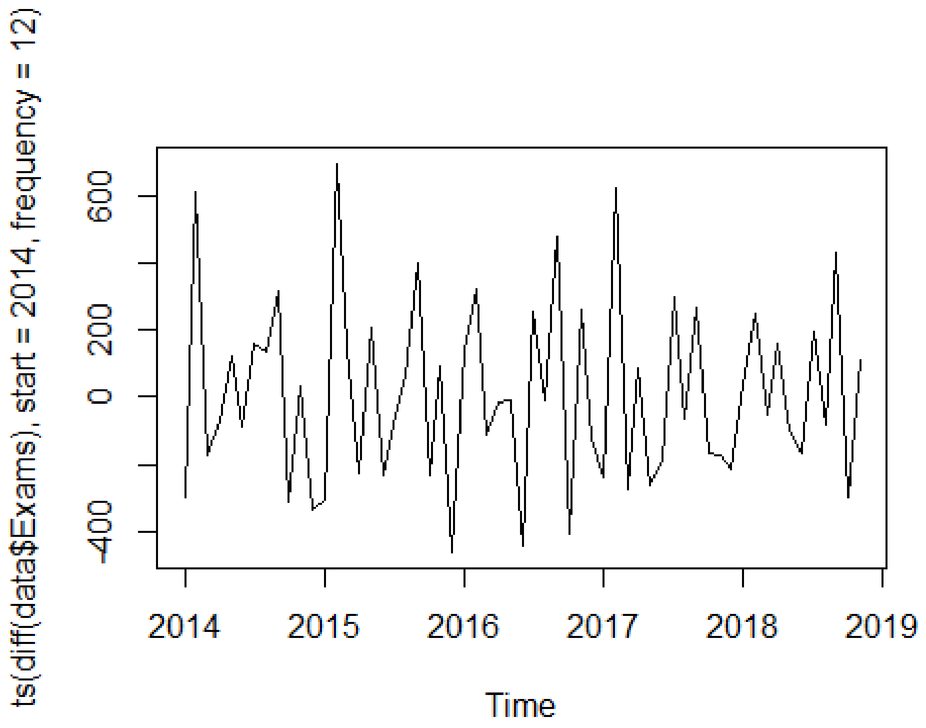
```r
data = read.csv(file.choose(), header=TRUE)

# Spit data into training and testing

train = data[1:(nrow(data)-6),]
test = data[(nrow(data)-5):nrow(data),]

plot(ts(data$Exams, start=2014, frequency = 12))
```



Stationarity is violated as there is a clear upwards trend in the data. Thus, there is not a constant mean. There also does not appear to be constant variance.

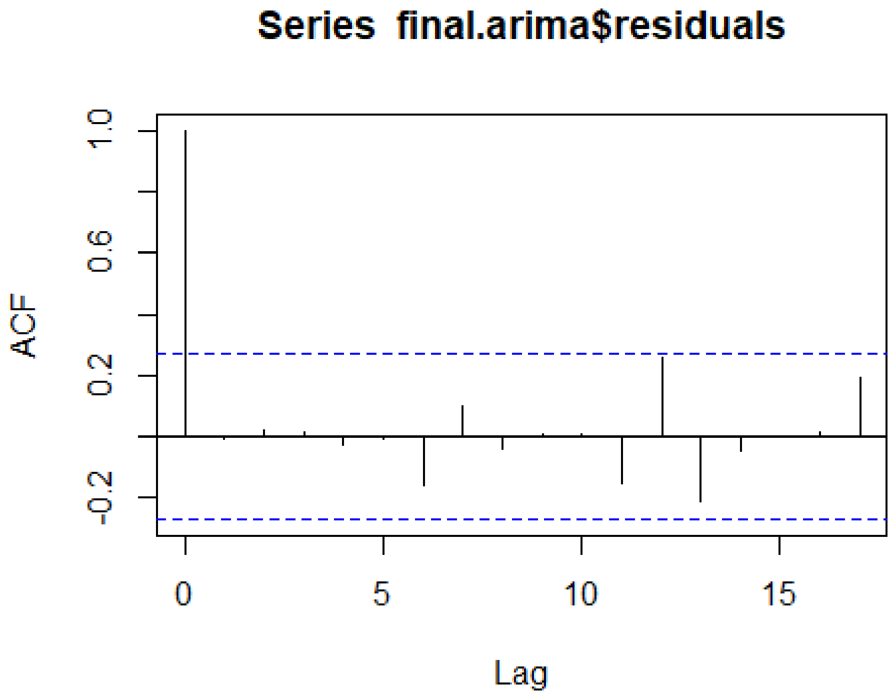
```r
plot(ts(diff(data$Exams), start=2014, frequency = 12))
```



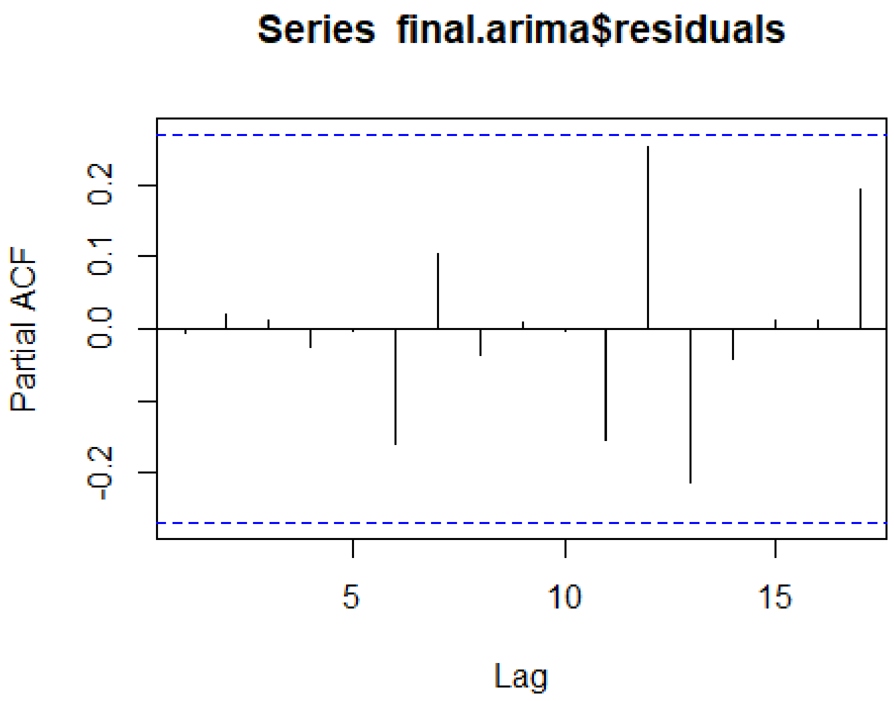
The plot of the first order difference appears to be stationary. There now seems to be a constant mean and variance. The trend also appears to have been eliminated.

```
final.aic = Inf
final.order = c(0,0,0)
norder = 5

for (p in 1:norder) for (q in 1:norder) {

  current.aic=AIC(arima(diff(train$Exams), order=c(p,0,q), method='ML'))

  if (current.aic < final.aic) {

    final.aic = current.aic

    final.order = c(p,0,q)

    final.arima=arima(diff(train$Exams), order=final.order, method='ML')

  }
}

final.aic

## [1] 714.9967
```

```
final.arima$coef

##         ar1        ar2        ar3        ar4        ma1        ma2
##  0.2034696 -0.6680344 -0.2797134 -0.4212283 -1.0916508  1.2949092
##        ma3  intercept
## -0.4909931  8.5441046

acf(final.arima$residuals)
```



```
pacf(final.arima$residuals)
```



```
Box.test(final.arima$residuals, lag = (4+3+1), type = "Ljung-Box", fitdf = (4
+3))

##
##  Box-Ljung test
##
## data:  final.arima$residuals
## X-squared = 2.4444, df = 1, p-value = 0.1179
```

The residuals do not appear uncorrelated as the Ljung-Box is low.

```
final.seas.aic = Inf
final.seas.order = c(0,0,0)

for (P in 1:norder) for (Q in 1:norder) {

  current.aic=AIC(arima(diff(train$Exams), order=c(4,0,3), seasonal = list(or
der = c(P,0,Q))))

  if (current.aic < final.aic) {

    final.seas.aic = current.aic

    final.seas.order = c(P,0,Q)

    final.seas.arima=arima(diff(train$Exams), order=c(4,0,3), seasonal = list
(order = c(P,0,Q)))
```

```
  }
}
```

```
## Warning in log(s2): NaNs produced
```

```
## Warning in arima(diff(train$Exams), order = c(4, 0, 3), seasonal =
## list(order = c(P, : possible convergence problem: optim gave code = 1
```

```
## Warning in log(s2): NaNs produced
```

```
## Warning in log(s2): NaNs produced
```

```
## Warning in log(s2): NaNs produced
```

```
## Warning in log(s2): NaNs produced
```

```
## Warning in arima(diff(train$Exams), order = c(4, 0, 3), seasonal =
## list(order = c(P, : possible convergence problem: optim gave code = 1
```

```
## Warning in log(s2): NaNs produced
```

```
## Warning in log(s2): NaNs produced
```

```
## Warning in arima(diff(train$Exams), order = c(4, 0, 3), seasonal =
## list(order = c(P, : possible convergence problem: optim gave code = 1
```

```
## Warning in log(s2): NaNs produced
```

```
## Warning in arima(diff(train$Exams), order = c(4, 0, 3), seasonal =
## list(order = c(P, : possible convergence problem: optim gave code = 1
```

```
## Warning in arima(diff(train$Exams), order = c(4, 0, 3), seasonal =
## list(order = c(P, : possible convergence problem: optim gave code = 1
```

```
## Warning in log(s2): NaNs produced
```

```
## Warning in log(s2): NaNs produced
```

```
## Warning in arima(diff(train$Exams), order = c(4, 0, 3), seasonal =
## list(order = c(P, : possible convergence problem: optim gave code = 1
```

```
## Warning in arima(diff(train$Exams), order = c(4, 0, 3), seasonal =
## list(order = c(P, : possible convergence problem: optim gave code = 1
```

```
## Warning in arima(diff(train$Exams), order = c(4, 0, 3), seasonal =
## list(order = c(P, : possible convergence problem: optim gave code = 1
```

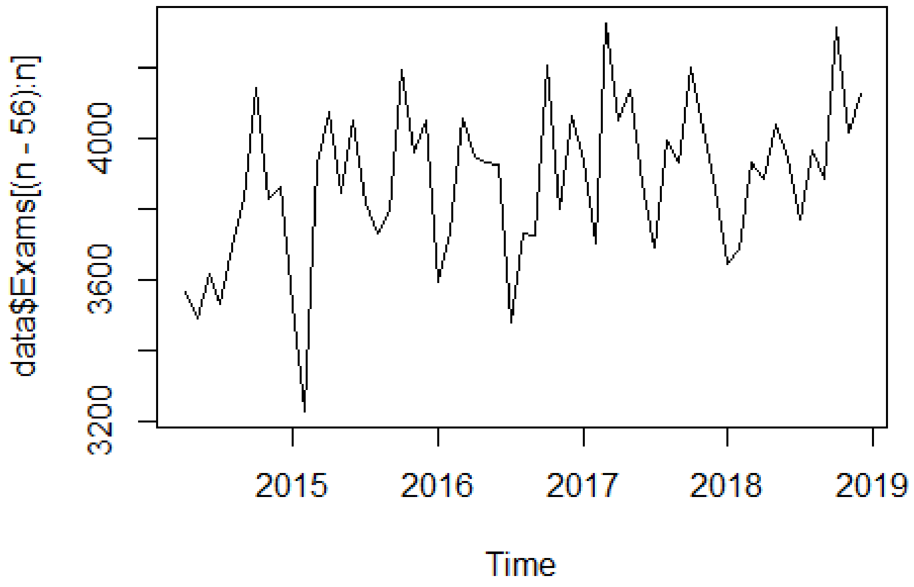
```
final.seas.aic
```

```
## [1] 714.9881
```

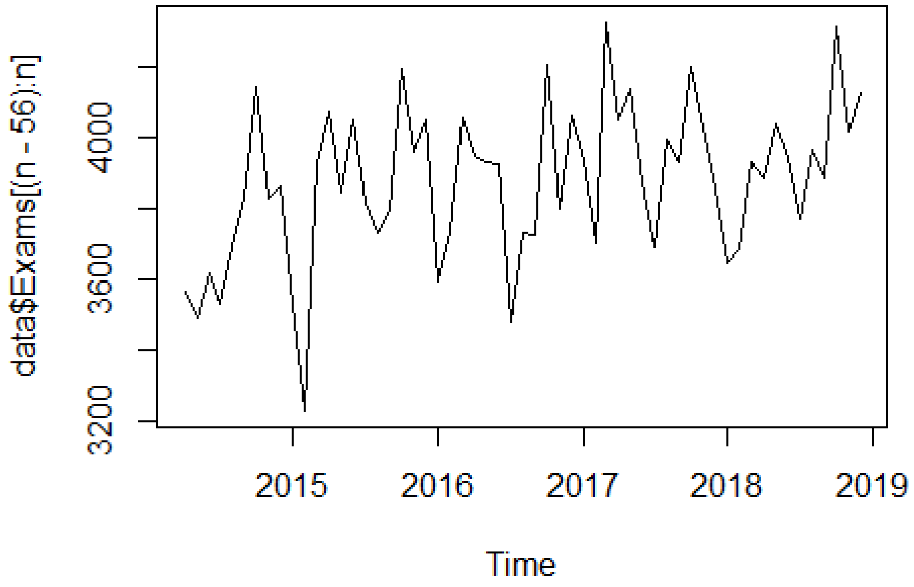
```
final.aic

## [1] 714.9967
```

I would choose the seasonal model based on AIC.

```
arima.pred = as.vector(predict(final.arima, newdata=test$Exams, n.ahead = 6))
seas.pred = as.vector(predict(final.seas.arima,n.ahead=6))

arima.pred$pred

## Time Series:
## Start = 54
## End = 59
## Frequency = 1
## [1] -196.77568  117.42923   93.36525   56.21617   17.61035  -91.04888

seas.pred$pred

## Time Series:
## Start = 54
## End = 59
## Frequency = 1
## [1] -211.90881  291.15483   33.96307  236.13435 -177.70681 -111.21441

times=time(ts(data$Exams, start=2014, frequency = 12))
n = length(data$Exams)
nfit = n-6
plot(times[(n-56):n],data$Exams[(n-56):n],type="l", xlab="Time")
points(times[(nfit+1):n],arima.pred$pred,col="red")
```



```
plot(times[(n-56):n],data$Exams[(n-56):n],type="l", xlab="Time")
points(times[(nfit+1):n],seas.pred$pred,col="red")
```


data$Exams[(n - 56):n]
Time