Assignment3 : Time Series

Paras (1801035), Karthigeyan (1801093), Shanmuga Priya (1801115), Raju (1801166), Niranjan (Acad Sec #1)

24/07/2019

Question 1

Simulating an AR(1) process

Given $\phi_0 = 100$ and $\phi_1 = 0.8$, we simulate simulate a random walk using the equation

$$Y_t = \phi_0 + \phi_1 Y_{t-1} + e_t$$

,Where e_t is the white noise component with $\mu = 0$ and $\sigma = 1$.

Part (a)

Generating the simulation using the above equation.

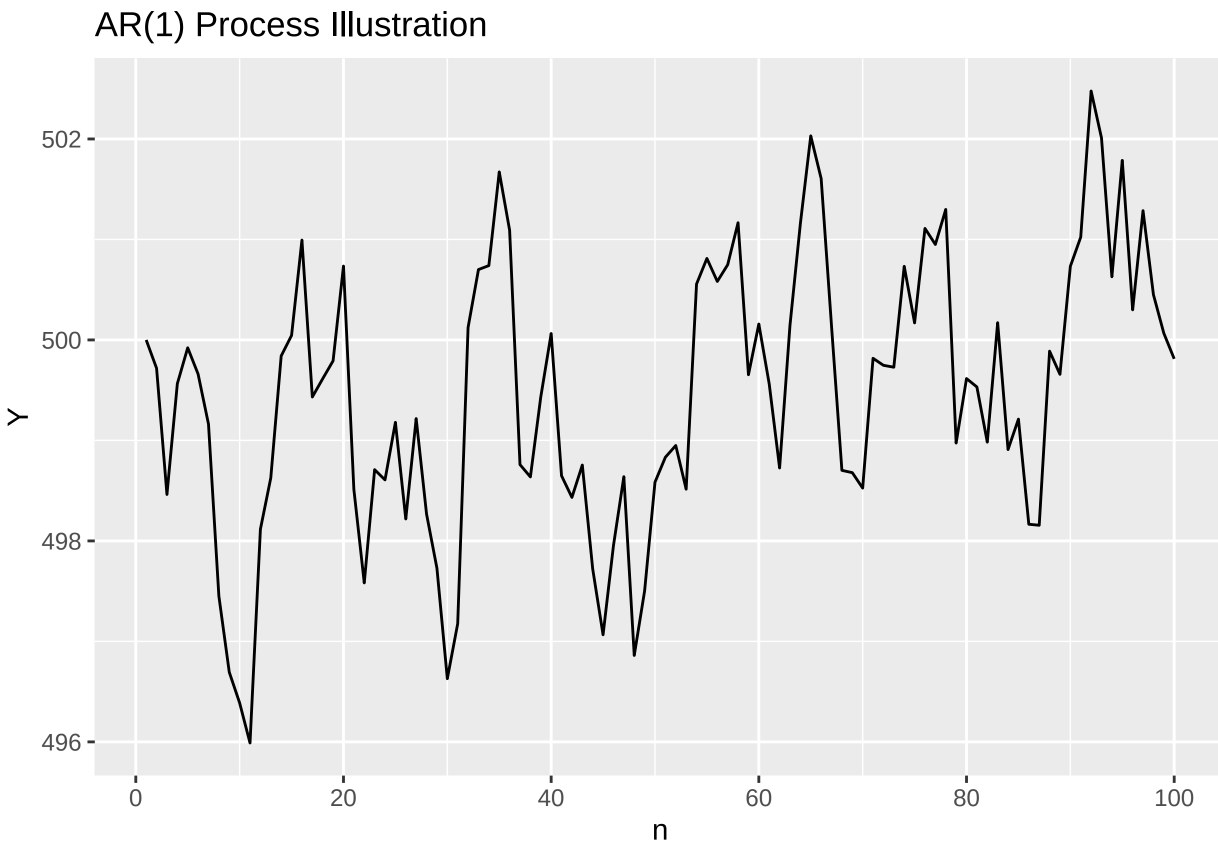
```
# initialize seed for random numbers
set.seed(999)
C<-100
phi_1<-0.8
y<-1:108
y[1]<-C/(1-phi_1); y[1]
```

```
## [1] 500
```

```
# Generating the random walk using equation
for (i in 2:108 ) {y[i]<-(phi_1*y[i-1])+C+rnorm(1, mean = 0, sd = 1)}

#Ploting the random walk for first 100 values
train<-ts(y[1:100],frequency = 1) # creating a training dataset from Y
test<-ts(y[100:108],frequency = 1) # creating a test datatset from Y
library(ggplot2)
library(forecast)
```

```
## Registered S3 method overwritten by 'xts':
##   method     from
##   as.zoo.xts zoo

## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo

## Registered S3 methods overwritten by 'forecast':
##   method             from
##   fitted.fracdiff    fracdiff
##   residuals.fracdiff fracdiff
```

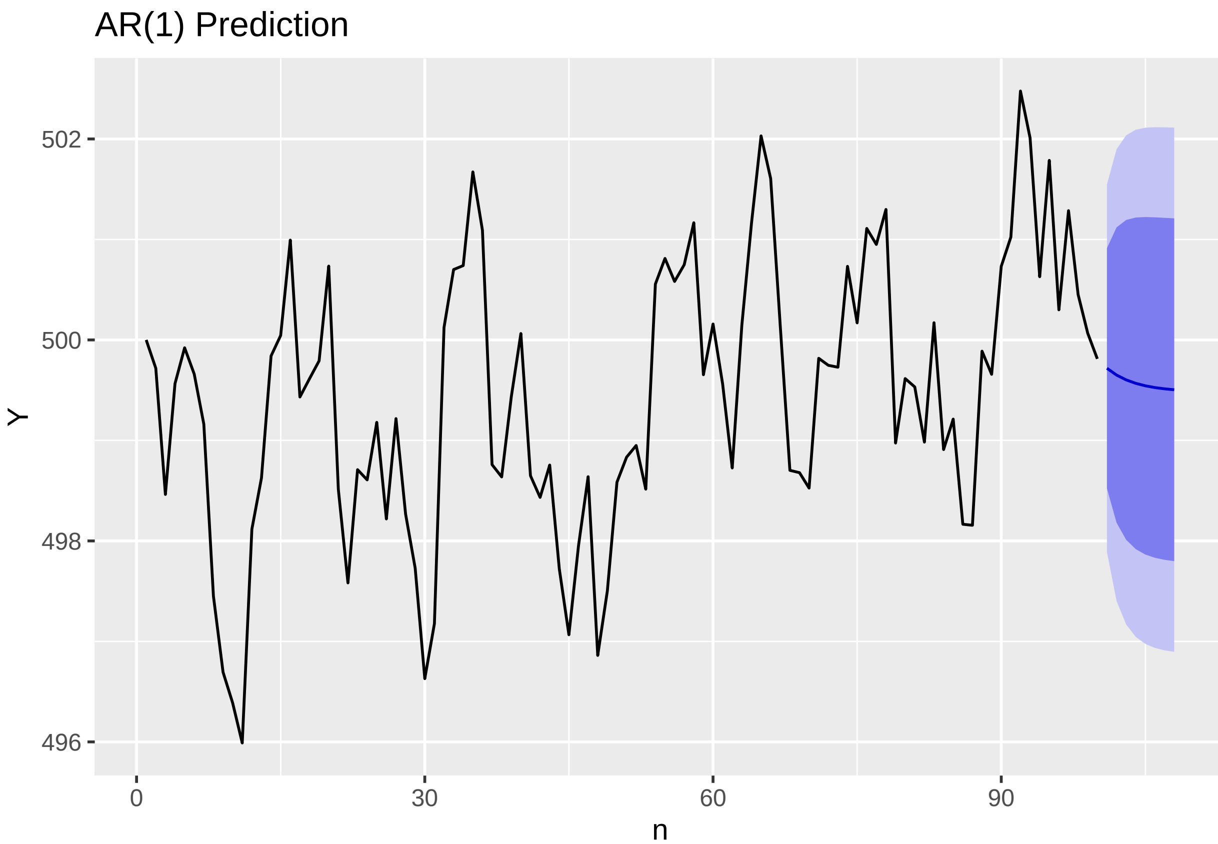
```
autoplot(train , ylab = "Y", xlab = "n", main ="AR(1) Process Illustration")
```



Part (b)

Generating an arima model using the first 100 values.

```
## Estimating AR(1) Model
model<-arima(train,order = c(1,0,0))
prediction<-forecast(model,h=8)

# Plotting the Predictions
autoplot(prediction, ylab = "Y", xlab = "n", main ="AR(1) Prediction")
```



Part (c)

Comparing the predicted values with the observed values. We first calculate room mean square error.

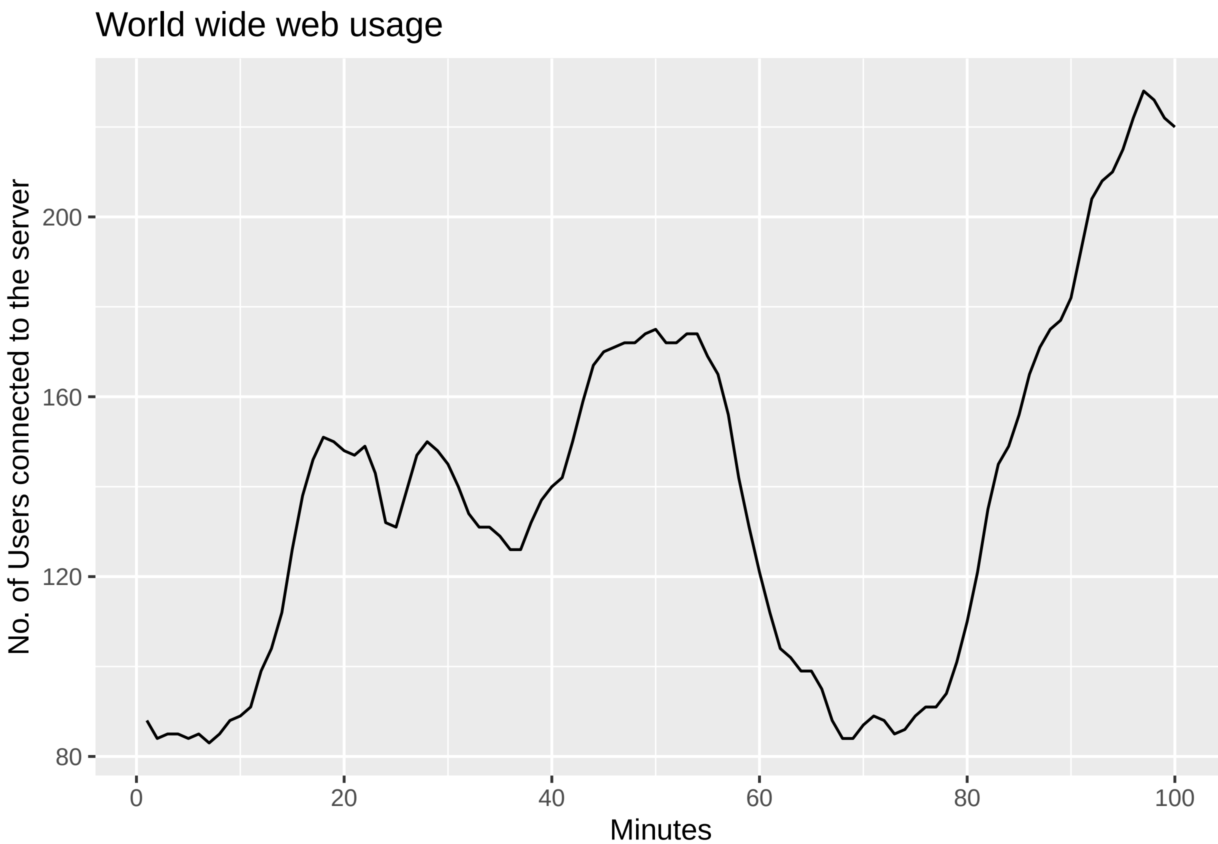
```
#Comparing the preditions with the test data
RMSE= sqrt(mean((prediction$mean-y[101:108])^2))
#Root mean square error
RMSE
```

```
## [1] 1.14924
```

Question 2

ARIMA Model for WWWusage

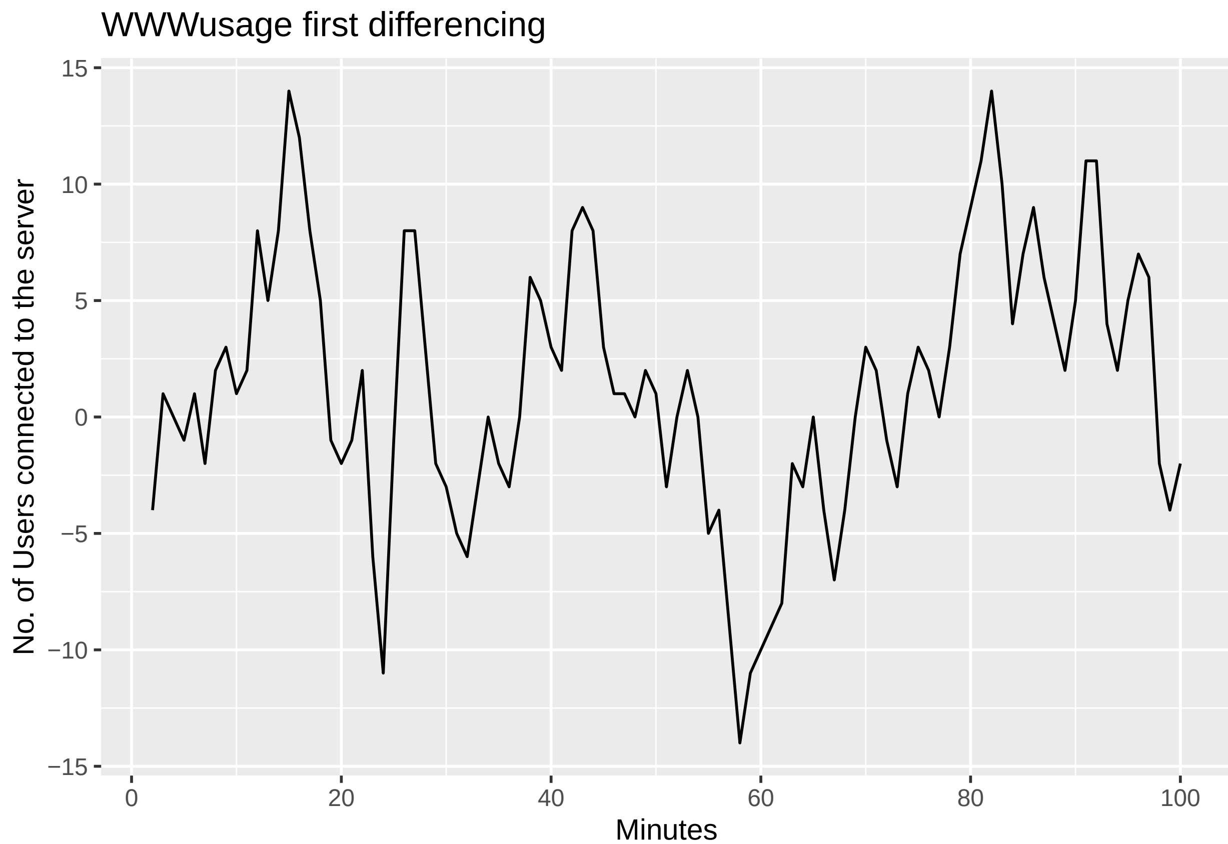
```
### Question 2 - Developing and ARIMA process using "Modeling Procedure"

# We Start with the time series model
autoplot(WWWusage, ylab = "No. of Users connected to the server",
         xlab = "Minutes", main ="World wide web usage")
```




```r
#Visual inspection does not give us any idea about stationarity
#Next we try to confirm if the data is stationary or not using the ADF test

##Staionarity Tests
library(tseries)
adf.test(WWWusage)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  WWWusage
## Dickey-Fuller = -2.6421, Lag order = 4, p-value = 0.3107
## alternative hypothesis: stationary
```

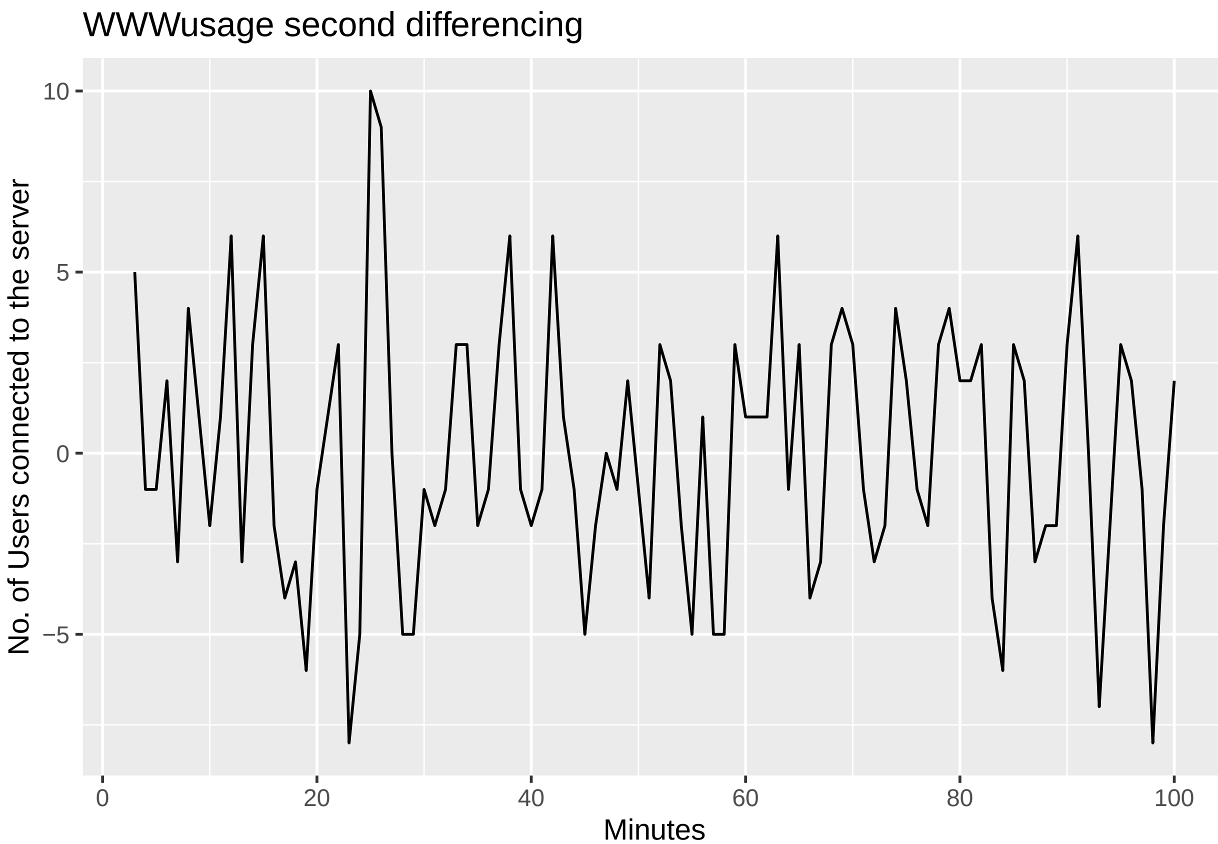
```r
#p-value> 0.05 : Fail to reject null hypothesis = Non-stationarity
##So the time series contains a random walk in it


#Taking first differences
diff.series<-diff(WWWusage, lag = 1, differences = 1)
autoplot(diff.series, ylab = "No. of Users connected to the server",
         xlab = "Minutes", main ="WWWusage first differencing")
```




```
##Staionarity Tests
adf.test(diff.series)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  diff.series
## Dickey-Fuller = -2.5459, Lag order = 4, p-value = 0.3506
## alternative hypothesis: stationary
```

```
#p-value> 0.05 : Fail to reject null hypothesis = Non-stationarity

#Taking second differences
diff2.series<-diff(diff.series, lag = 1, differences = 1)
autoplot(diff2.series, ylab = "No. of Users connected to the server",
         xlab = "Minutes", main ="WWWusage second differencing")
```

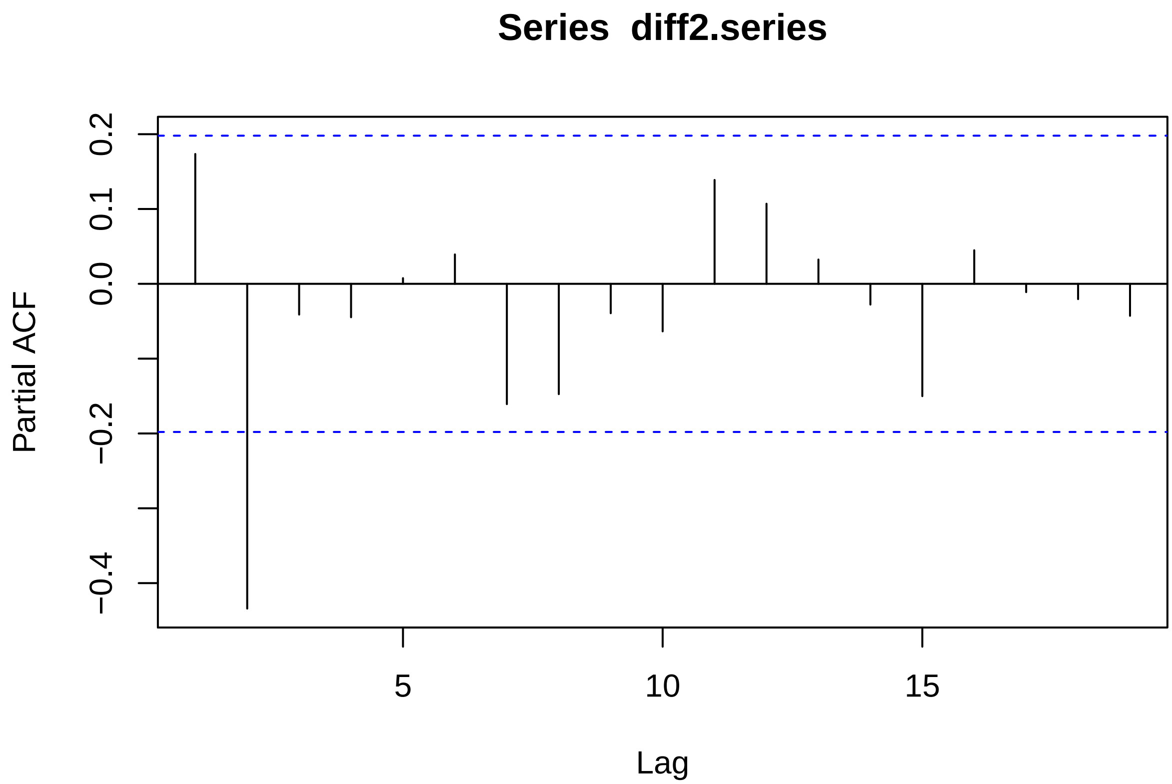



```r
##Staionarity Tests
adf.test(diff2.series)
```

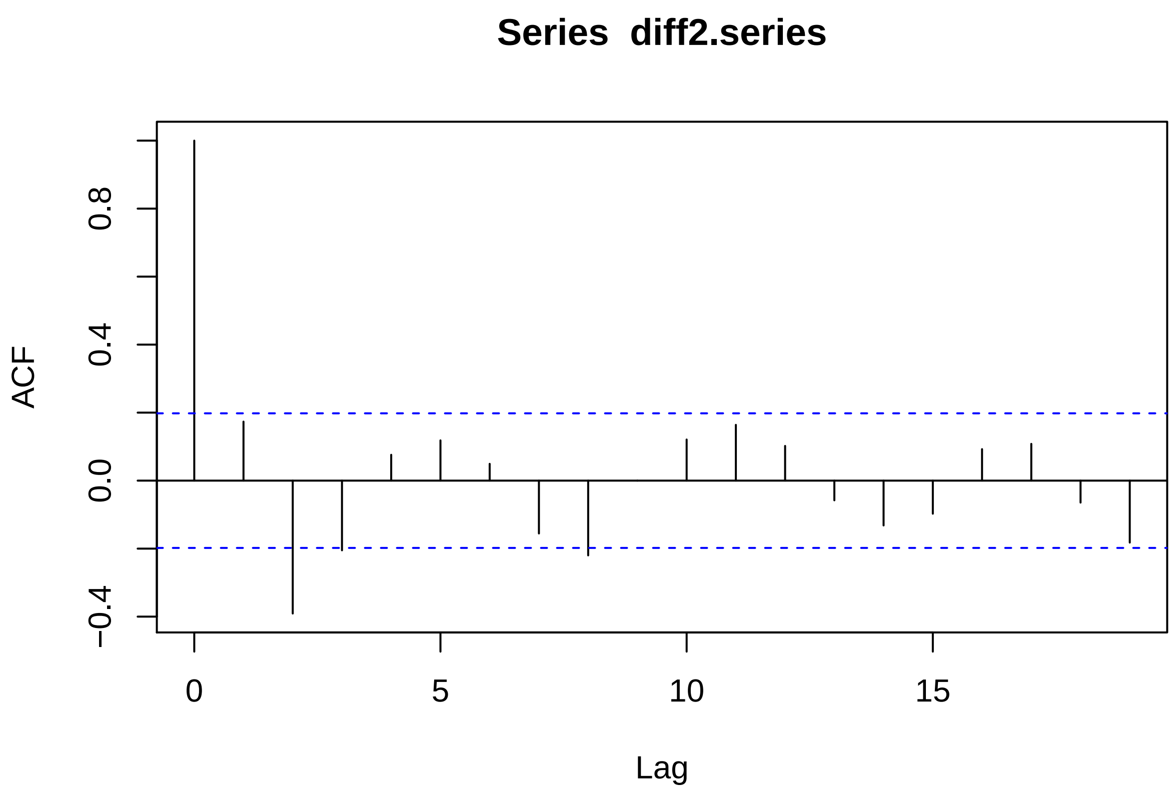
```
## Warning in adf.test(diff2.series): p-value smaller than printed p-value
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  diff2.series
## Dickey-Fuller = -4.828, Lag order = 4, p-value = 0.01
## alternative hypothesis: stationary
```

```r
#p-value< 0.05 : Reject null hypothesis = Non-stationarity
#Therefore the differencing parameter d=2

# To find the value of p, we plot the PACF of diff2.series
pacf(diff2.series)
```




```
#From the graph we see that a lag of 2 is significantly greater than zero. Hence p=2

# To find the value of q, we plot the ACF of diff2.series
acf(diff2.series)
```



```r
#From the we see that a lag of 1 has a value significantly greater than zero

# Thus our ARIMA (p,d,q) model is is ARIMA (2,2,1)
fit1<-arima(WWWusage,order = c(2,2,1))
fit1
```

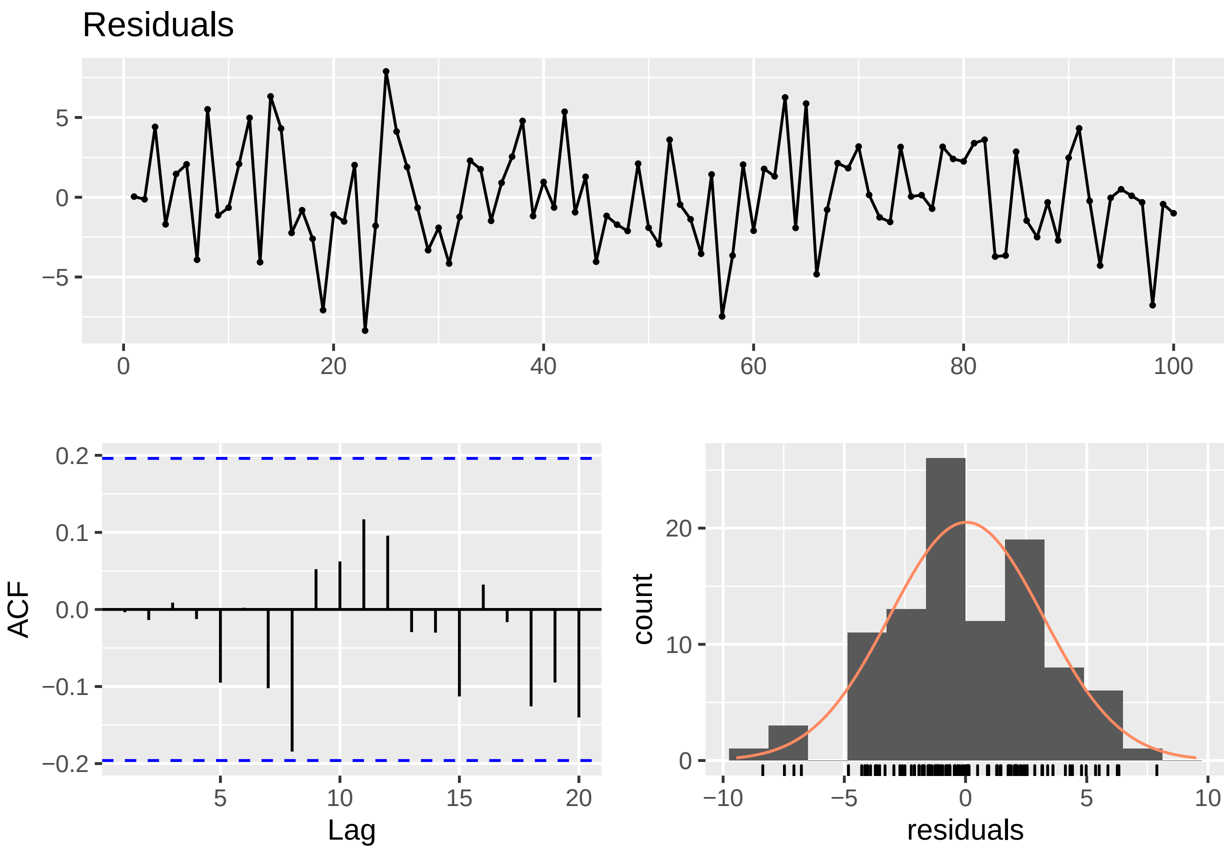
```
##
## Call:
## arima(x = WWWusage, order = c(2, 2, 1))
##
## Coefficients:
##          ar1      ar2      ma1
##       0.3512  -0.4572  -0.1161
## s.e.  0.2189   0.0937   0.2502
##
## sigma^2 estimated as 10.1:  log likelihood = -252.63,  aic = 513.26
```

```r
#The AIC score is 513.26
fit2<-auto.arima(WWWusage)
fit2
```

```
## Series: WWWusage
## ARIMA(1,1,1)
##
## Coefficients:
##          ar1     ma1
##       0.6504  0.5256
## s.e.  0.0842  0.0896
##
## sigma^2 estimated as 9.995:  log likelihood=-254.15
## AIC=514.3   AICc=514.55   BIC=522.08
```

```r
# The AIC score is 514.3
# ARIMA(2,2,1) has the lowest AIC value, hence we will select this model

#Now we check the residuals of our chosen model
checkresiduals(fit1$residuals)
```

```
## Warning in modeldf.default(object): Could not find appropriate degrees of
## freedom for this model.
```




```
#Now the residuals look like white noise and the ACF values are almost zero.
#Hence our models is correct.
```