ODI.R

khyati_soni

2020-12-07

```
#Data has been extracted from ESPNCricinfo
#Dataset consists of ODI matches summary which includes the two playing teams,
#and their respective overs played, wickets down and runs per inning.
odi<-read.csv("/Users/khyati_soni/Downloads/matches.csv")
head(odi)
```

```
##         team1        team2 innings1_overs innings1_wickets innings1_runs
## 1   SRI LANKA     ZIMBABWE           50.0                7           272
## 2 NETHERLANDS       CANADA           50.0                7           289
## 3    ZIMBABWE SOUTH AFRICA           50.0                8           174
## 4       INDIA  NEW ZEALAND           49.0               10           276
## 5   AUSTRALIA  NEW ZEALAND           48.4               10           181
## 6   AUSTRALIA        INDIA           50.0                5           359
##   innings2_overs innings2_wickets innings2_runs year
## 1           47.2               10           213 2001
## 2           43.0               10           172 2007
## 3           34.2                1           175 2003
## 4           45.2               10           236 2010
## 5           50.0                8           182 2009
## 6           43.3                1           362 2013
```

```
#Extracting all the matches played by India
crik_india <- odi[odi$team1=="INDIA" | odi$team2 == "INDIA",]
head(crik_india)
```

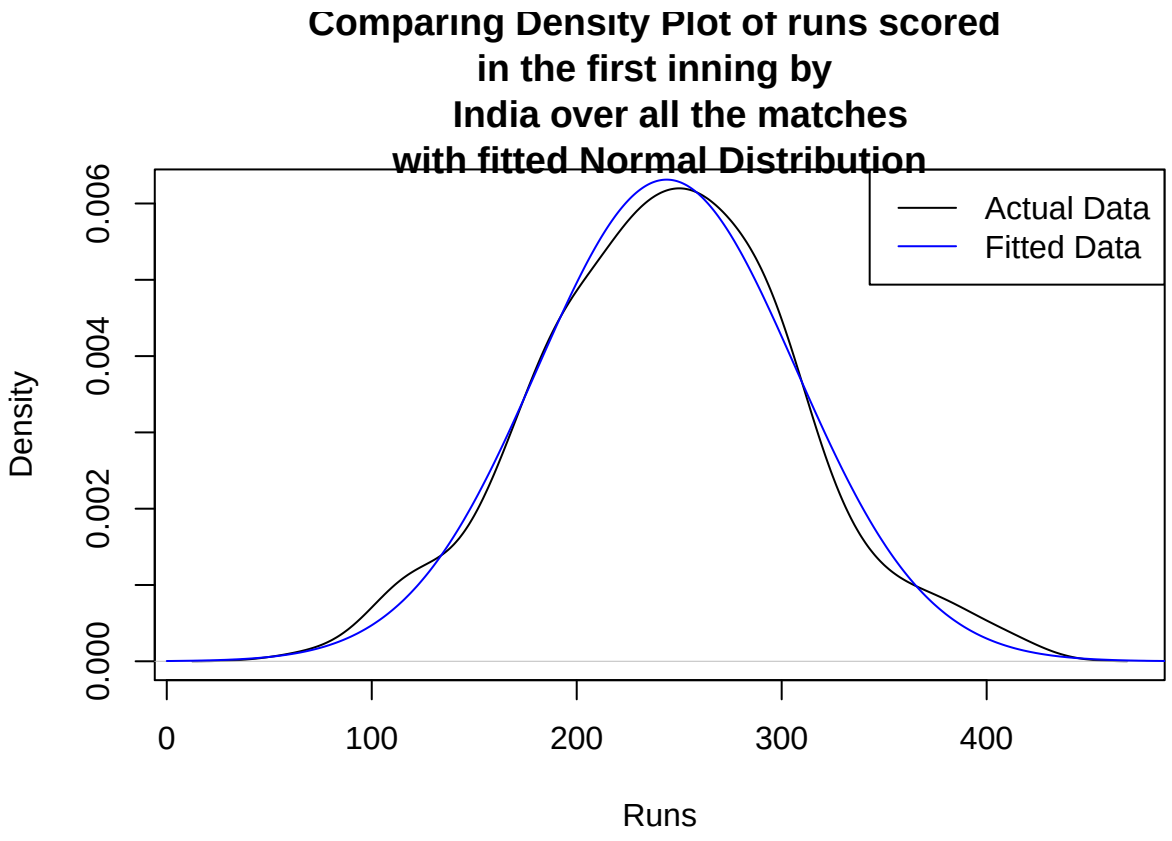
```
##          team1        team2 innings1_overs innings1_wickets innings1_runs
## 4        INDIA  NEW ZEALAND             49               10           276
## 6    AUSTRALIA        INDIA             50                5           359
## 16       INDIA SOUTH AFRICA             50                6           267
## 19 WEST INDIES        INDIA             50                9           192
## 20   AUSTRALIA        INDIA             50                5           313
## 28   AUSTRALIA        INDIA             50                4           350
##    innings2_overs innings2_wickets innings2_runs year
## 4            45.2               10           236 2010
## 6            43.3                1           362 2013
## 16           46.0               10           193 1996
## 19           33.1                4           135 1994
## 20           48.2               10           281 2019
## 28           49.4               10           347 2009
```

```
#Extracting first innings runs of India:
x<-crik_india[crik_india$team1=="INDIA",5]
innings1<-x[1:50]
```

```r
#Density plot of first innings runs
plot(density(x),main="Comparing Density Plot of runs scored \nin the first inning by
     India over all the matches \nwith fitted Normal Distribution",xlab="Runs")

#Finding parameters mean and sigma to fit a distribution
#using MLE (Maximum Likelihood Estimate)
f<-function(params){
  lnL<-dnorm(x,params[1],params[2],log = TRUE) #log likelihood function
  sum(-lnL) #sum of log likelihood function
}

p<-c(mean(x),sd(x))
nlm(f,p)$estimate
```

```
## [1] 243.87205  63.19417
```

```r
mu<-nlm(f,p)$estimate[1]
sigma<-nlm(f,p)$estimate[2]

x1<-seq(0,500)
lines(x1,dnorm(x1,mu,sigma),col="blue")
legend("topright",legend=c("Actual Data","Fitted Data"),col=c("black","blue"),lty=c(1,1))
```

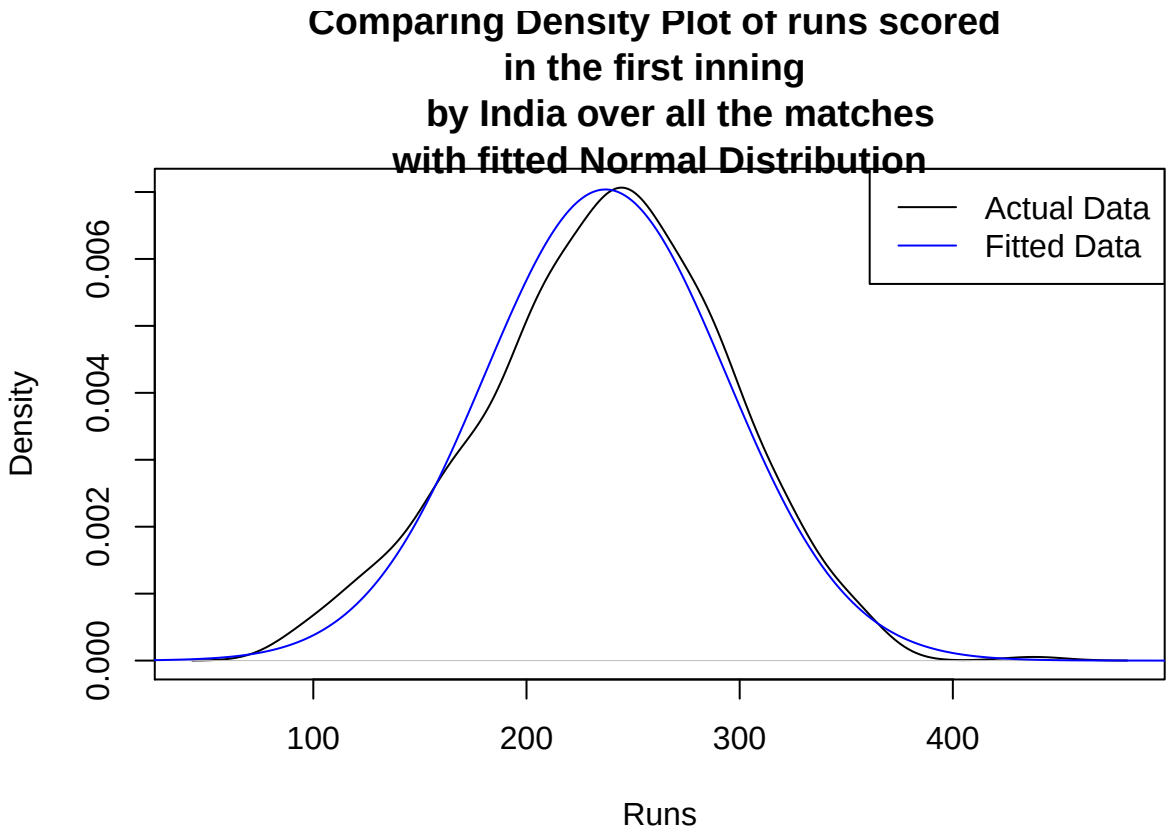



```r
#Therefore, we can conclude that the given estimate provides a good fit to our model.
#Knowing the parameters, we can perform various analysis such as calculation of
#confidence interval and hypothesis testing.

#Similarly for the second innings:
```

```r
y<-crik_india[crik_india$team2=="INDIA",5]
innings2<-y[1:50]

#Density plot of second innings
plot(density(y),main="Comparing Density Plot of runs scored \nin the first inning
     by India over all the matches \nwith fitted Normal Distribution",xlab="Runs")
#Finding parameters mean and sigma to fit a distribution using
#MLE (Maximum Likelihood Estimate)
f<-function(params){
  lnL<-dnorm(y,params[1],params[2],log = TRUE)
  sum(-lnL)
}

p<-c(mean(x),sd(x))
nlm(f,p)$estimate
```

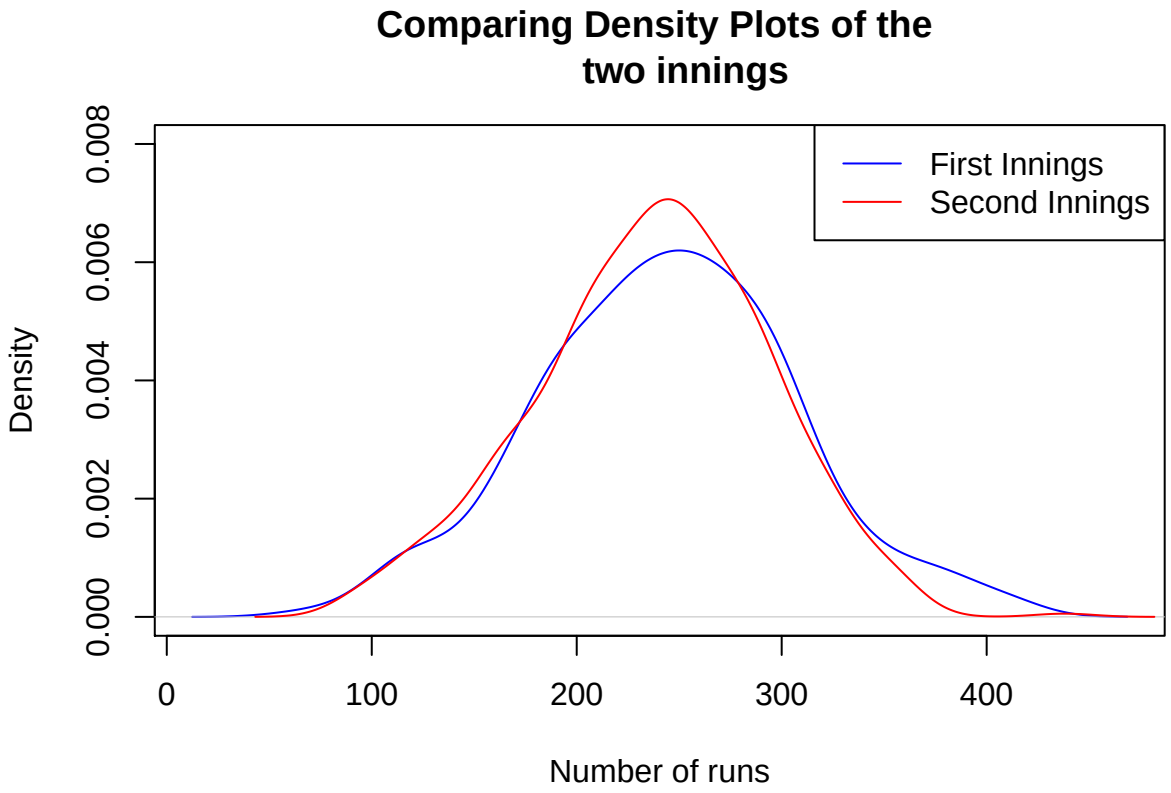
```
## [1] 237.05940  56.68371
```

```r
mu<-nlm(f,p)$estimate[1]
sigma<-nlm(f,p)$estimate[2]

y1<-seq(0,500)
lines(y1,dnorm(y1,mu,sigma),col="blue")
legend("topright",legend=c("Actual Data","Fitted Data"),col=c("black","blue"),lty=c(1,1))
```




```r
#Therefore, we can conclude that the given estimate provides a good fit to our model.
#Knowing the parameters, we can perform various analysis such as calculation of
#confidence interval and hypothesis testing.
```

```r
#Effect of the second innings adjustment:
#Does India play better in the first inning or in the second innning?
#Comparing Density Plots of First and Second Innings played by India

plot(density(x),col="blue",ylim=c(0,0.008),main="Comparing Density Plots of the
     two innings",xlab="Number of runs")
lines(density(y),col="red")
legend("topright",legend=c("First Innings","Second Innings"),
       col=c("blue","red"),lty=c(1,1))
```




```r
#Conclusion: India plays better in the Second Innings
#Therefore, they should choose to bowl first if the coin flip
#turns out to be in their favour.



#Extracting number of wickets down if India chooses to bat in the first
#inning over all the matches:
wik1<-crik_india[crik_india$team1=="INDIA",4]
#Total number of wickets taken in the first inning:
tot_wik1<-sum(wik1)

#Extracting number of wickets down if India chooses to bat in the first
#inning over all the matches:
```

```r
wik2<-crik_india[crik_india$team2=="INDIA",7]
#Total number of wickets taken in the first inning:
tot_wik2<-sum(wik2)

#Thus, calculating the percentage of wickets down in the
#first vs second innings for India:
tot_wik1/(length(wik1)*10)*100
```

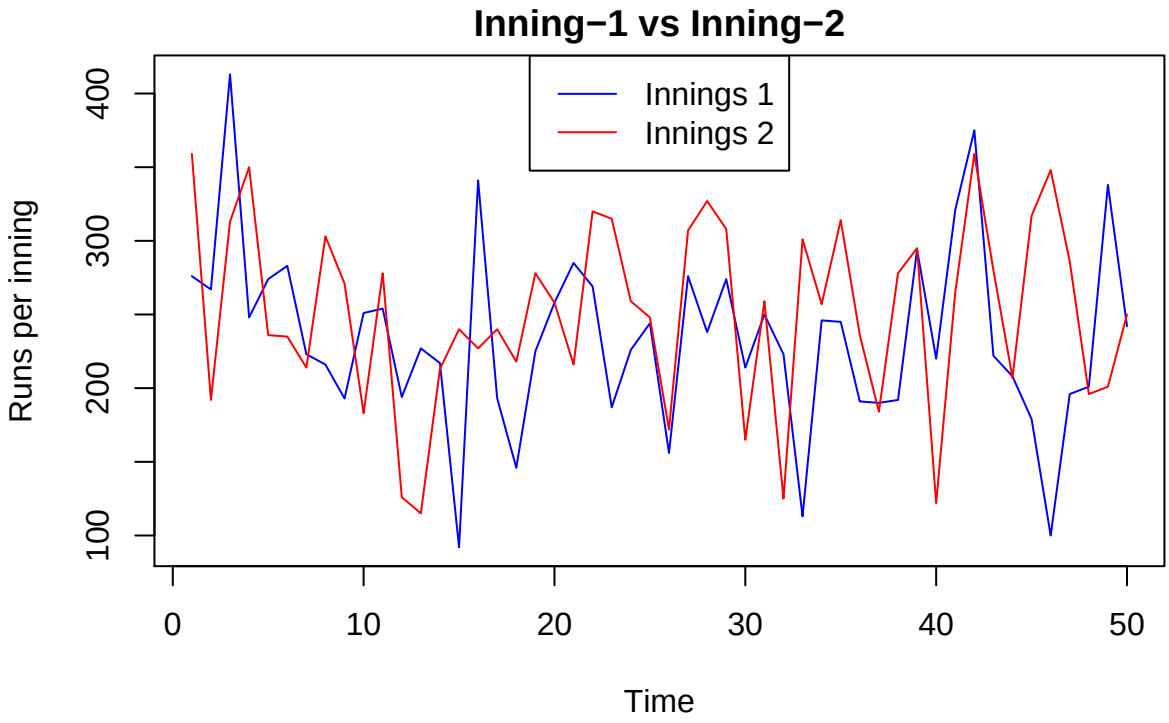
```
## [1] 74.01826
```

```r
tot_wik2/(length(wik2)*10)*100
```

```
## [1] 61.46825
```

```r
#We observe that the percentage of wickets down over all the matches if India plays in the
#first innings is higher than that if India plays in the second innings.
#Therefore, we again conclude that India should choose to bowl first if the coin
#flip turns out to be in their favour.




#Generating Time Series of the first inning and plotting
TS1<-ts(innings1,start=1,end=length(innings1),frequency=1)
ts.plot(TS1,col="blue",ylab="Runs per inning",main="TIME SERIES (Number of runs):
        \nInning-1 vs Inning-2")

#Generating Time Series of the second inning and plotting to compare
#it to the first inning
TS2<-ts(innings2,start=1,end=length(innings2),frequency=1)
lines(TS2,col="red")
legend("top",legend=c("Innings 1","Innings 2"),col=c("blue","red"),lty=c(1,1))
```

TIME SERIES (Number of runs):




```
#Fitting time series process to our model:
#The following function calculates the aic of all the possible combinations for
```

p', 'd', 'q' for an ARIMA-p,d,q process.

```
ans<-numeric(4)
for(p in 0:2){
  for(d in 0:2){
    for(q in 0:2){
      aic<-arima(TS1,order=c(p,d,q))$aic #Extracting akaike's information criterion
      row<-c(p,d,q,aic)
      ans<-rbind(ans,row)
    }
  }
}

head(ans)
```

```
##      [,1] [,2] [,3]     [,4]
## ans     0    0    0   0.0000
## row     0    0    0 557.7396
## row     0    0    1 559.3292
## row     0    0    2 561.2321
## row     0    1    0 574.5231
## row     0    1    1 551.5668
```

```
ans<-ans[-1,]
head(ans)
```

```
##      [,1] [,2] [,3]     [,4]
```

```
## row    0    0    0 557.7396
## row    0    0    1 559.3292
## row    0    0    2 561.2321
## row    0    1    0 574.5231
## row    0    1    1 551.5668
## row    0    1    2 552.9848
```

```
#Now, we extract the minimum aic and the respective p,d and q values of our ARIMA model.
which(ans[,4]==min(ans[,4])) #Extracts the row number of the lowest aic
```

```
## row
##   9
```

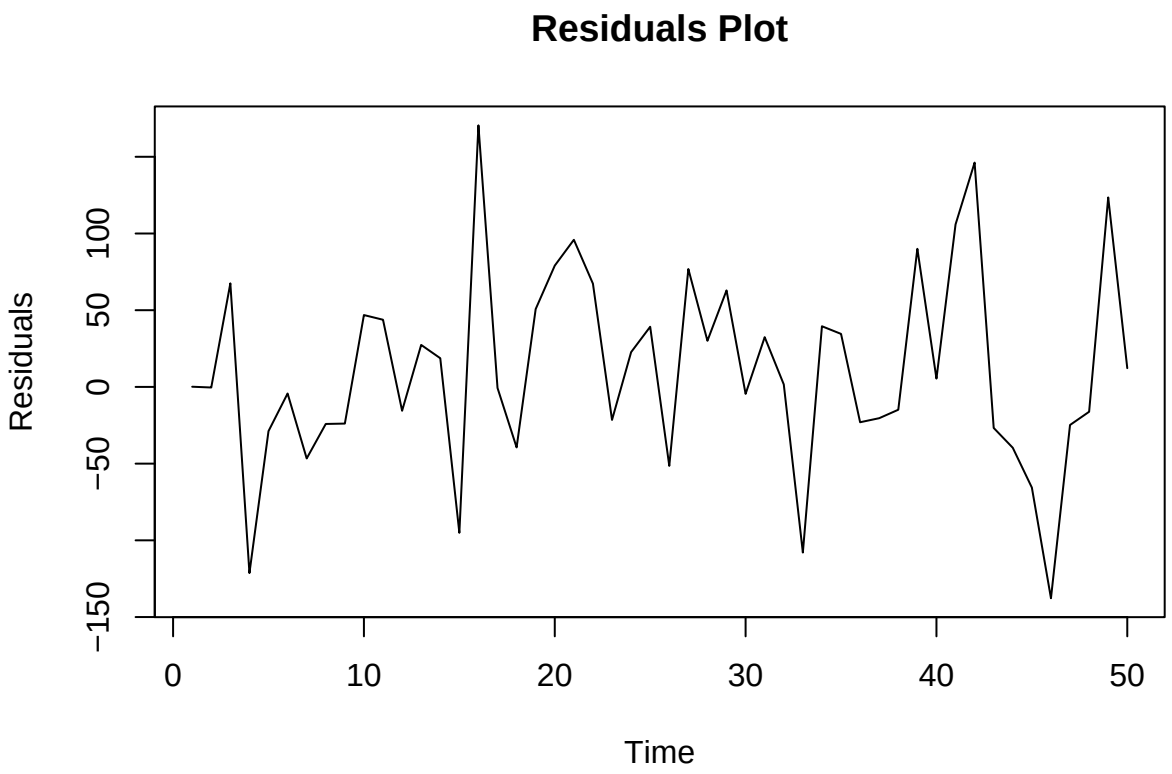
```
ans[9,]
```

```
## [1]   0.0000   2.0000   2.0000 551.2728
```

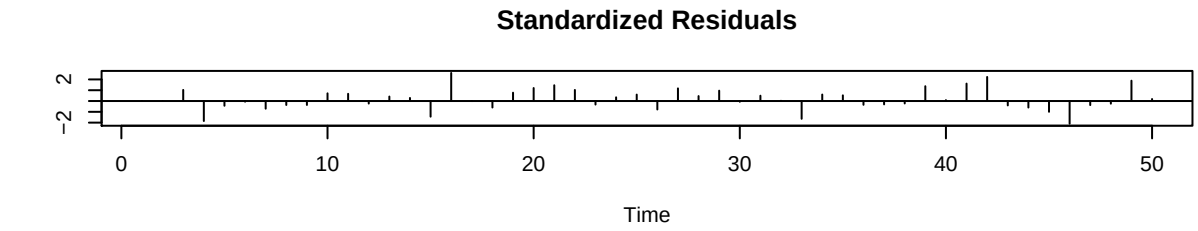
```
fit<-arima(TS1,order=c(0,2,2))
fit
```

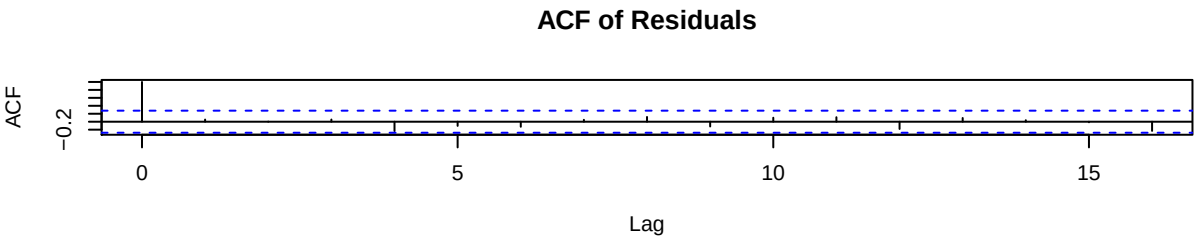
```
##
## Call:
## arima(x = TS1, order = c(0, 2, 2))
##
## Coefficients:
##           ma1     ma2
##       -1.8699  0.8803
## s.e.   0.1365  0.1373
##
## sigma^2 estimated as 4298:  log likelihood = -272.64,  aic = 551.27
```

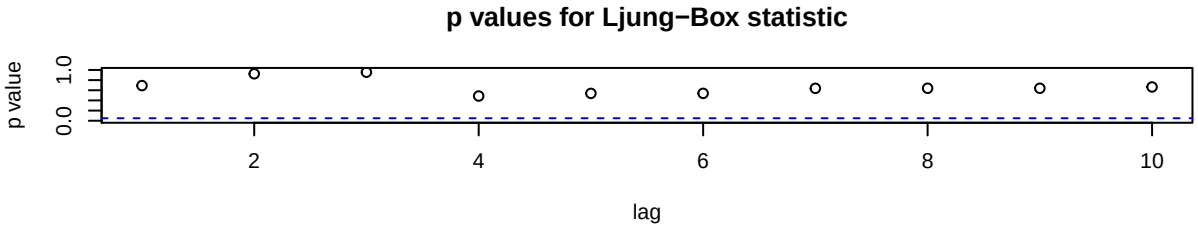
```
#beta1 = -1.8699
#beta2 = 0.8803
plot(fit$residuals,main="Residuals Plot",ylab="Residuals")
```



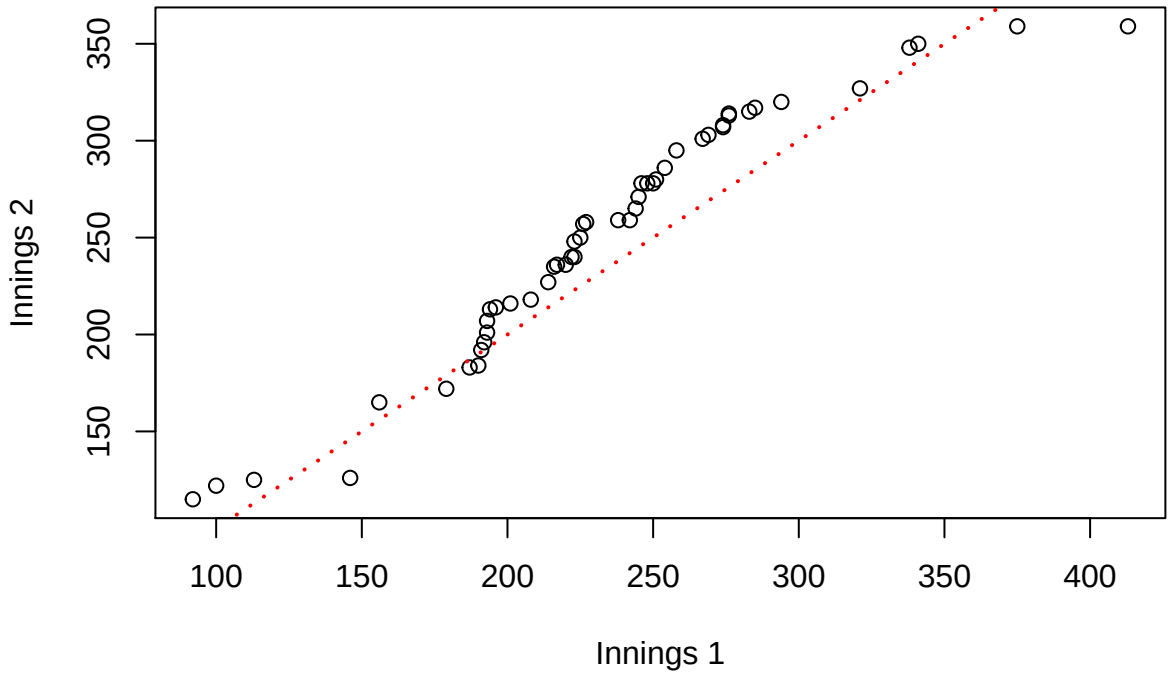

```
#The residuals plot is patternless
#There are almost equal number of positive and negative values
#Therefore, the ARIMA(0,2,2) process provides a good fit our model.



#MODELLING:
tsdiag(fit)
```



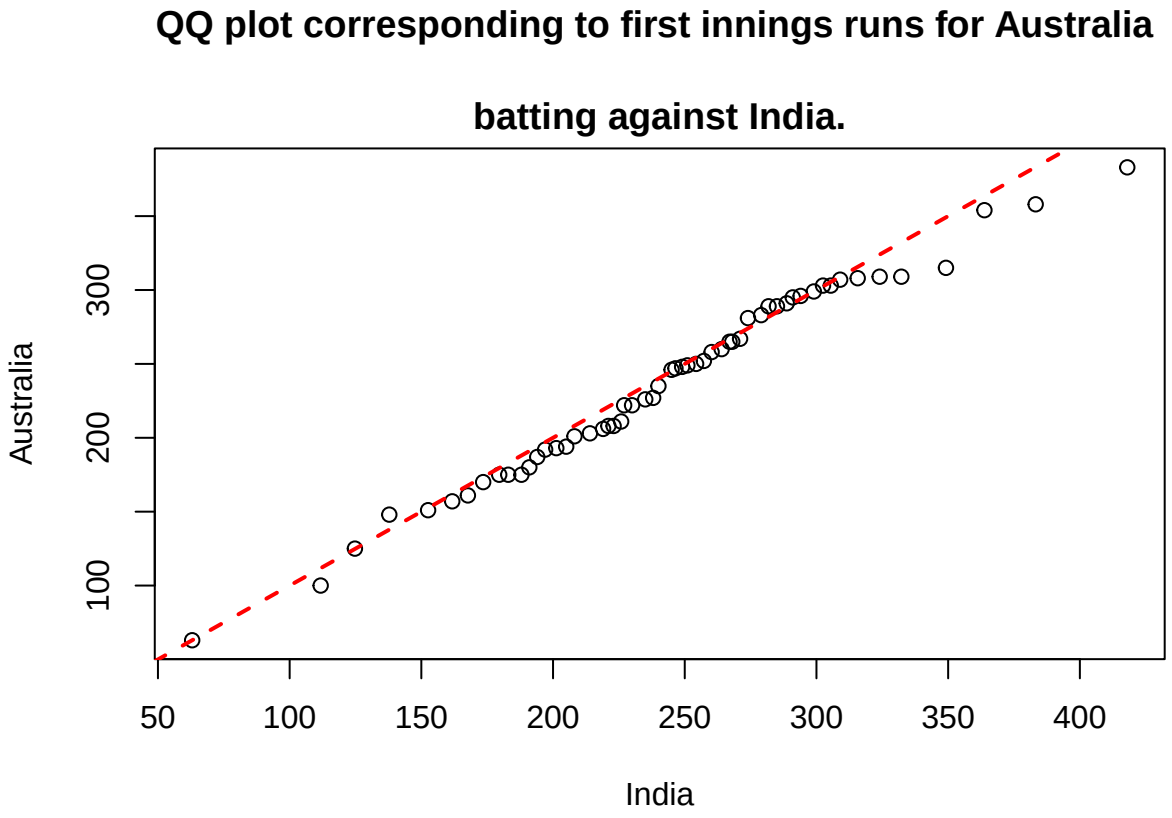





```
#Thus, for Standardized residuals:
#1. It is patternless
#2. There are equal number of positive and negative values
#3. Standardized Residuals lie between the range (-2,2) (since 96% of the values should
#lie between -2 and 2)
#ACF gets cut off for k>=1
#Thus, fit is very good




#qqplot between innings1 and innings 2 of India
qqplot(innings1,innings2,main="QQPlot: Innings1 vs Innings2",
       xlab="Innings 1",ylab="Innings 2")
abline(0,1,col="red",lty=3,lwd=2)
```




```
#QQ plot corresponding to first innings runs for Australia batting against India
Ind<-crik_india[crik_india$team1=="INDIA",5]
Aus<-crik_india[crik_india$team2=="AUSTRALIA",5]
qqplot(Ind,Aus,main="QQ plot corresponding to first innings runs for Australia
       \nbatting against India.",
       xlab="India",ylab="Australia")
abline(0,1,col="red",lty=2,lwd=2)
```




```
#COMPARING- Number of overs played in the First innings by Australia and by India:
Aus_data<-odi[odi$team1=="AUSTRALIA",]
head(Aus_data)
```

```
##        team1       team2 innings1_overs innings1_wickets innings1_runs
## 5  AUSTRALIA NEW ZEALAND           48.4               10           181
## 6  AUSTRALIA       INDIA           50.0                5           359
## 20 AUSTRALIA       INDIA           50.0                5           313
## 28 AUSTRALIA       INDIA           50.0                4           350
## 61 AUSTRALIA       INDIA           50.0                5           235
## 64 AUSTRALIA     ENGLAND           50.0                9           342
##    innings2_overs innings2_wickets innings2_runs year
## 5            50.0                8           182 2009
## 6            43.3                1           362 2013
## 20           48.2               10           281 2019
## 28           49.4               10           347 2009
## 61           41.5               10           198 2003
## 64           41.5               10           231 2015
```

```
n1<-length(Aus_data)
c1=0
for(i in 1:n1){
  if(Aus_data$innings1_overs[i]==50){
    c1=c1+1
  }
}

c1
```

```
## [1] 8
```

```
Ind_data<-odi[odi$team1=="INDIA",]
n2<-length(Ind_data)
c2=0
for(i in 1:n2){
  if(Ind_data$innings1_overs[i]==50){
    c2=c2+1
  }
}

c2
```

```
## [1] 5
```

```
#Thus, % of the time Australia uses all its overs in innings 1:
c1/n1*100
```

```
## [1] 88.88889
```

```
#Thus, % of the time India uses all its overs in innings 1:
c2/n2*100
```

```
## [1] 55.55556
```

```
#This suggests that there is merit in our modification of aggressiveness in first innings
#batting by Australia as compared to India.
#This supports our previous claim as well where had concluded that India plays better
#if they bat in the second innings.



#Analysing Sachin Tendulkkar as a cricketer:
#Data has been extracted from ESPNCricinfo
#Dataset consists summary statistics of Sachin Tendulkar as a batsman from 1989 to 2013.
data<-read.csv("/Users/khyati_soni/Documents/Khyati/Internship/cricketr/cricketr/data/tendulkar.csv")
head(data)
```

```
##   X Runs Mins  BF X4s X6s    SR Pos Dismissal Inns Opposition     Ground
## 1 1   15   28  24   2   0 62.50   6    bowled    2 v Pakistan    Karachi
## 2 2  DNB    -   -   -   -     -   -         -    4 v Pakistan    Karachi
## 3 3   59  254 172   4   0 34.30   6       lbw    1 v Pakistan Faisalabad
## 4 4    8   24  16   1   0 50.00   6   run out    3 v Pakistan Faisalabad
## 5 5   41  124  90   5   0 45.55   7    bowled    1 v Pakistan     Lahore
## 6 6   35   74  51   5   0 68.62   6       lbw    1 v Pakistan    Sialkot
##    Start.Date
## 1 15 Nov 1989
## 2 15 Nov 1989
## 3 23 Nov 1989
## 4 23 Nov 1989
## 5  1 Dec 1989
## 6  9 Dec 1989
```

```
#Exctracting runs
runs<-data[,2]
head(runs)
```

```
## [1] 15  DNB 59  8   41  35
## 149 Levels: 0 0* 1 10 10* 100 100* 101 103 103* 104* 105* 106 109 11 11* ... TDNB
```

```
#Creating Time Series from runs data:
ts<-ts(runs,start=1989,end=2013)

#The following function calculates the aic of all the possible combinations for
```

p', 'd', 'q' for an ARIMA-p,d,q process.

```
ans<-numeric(4)
for(p in 0:2){
  for(d in 0:2){
    for(q in 0:2){
      aic<-arima(ts,order=c(p,d,q))$aic
      row<-c(p,d,q,aic)
      ans<-rbind(ans,row)
    }
  }
}
head(ans)
```

```
##     [,1] [,2] [,3]     [,4]
## ans    0    0    0   0.0000
## row    0    0    0 264.3487
## row    0    0    1 265.8347
## row    0    0    2 267.7625
## row    0    1    0 272.3882
## row    0    1    1 258.0200
```

```
ans<-ans[-1,]
head(ans)
```

```
##     [,1] [,2] [,3]     [,4]
## row    0    0    0 264.3487
## row    0    0    1 265.8347
## row    0    0    2 267.7625
## row    0    1    0 272.3882
## row    0    1    1 258.0200
## row    0    1    2 259.3006
```

```
#Now, we extract the minimum aic and the respective p,d and q values of our ARIMA model.

which(ans[,4]==min(ans[,4]))
```

```
## row
##   9
```

```
ans[9,]
```

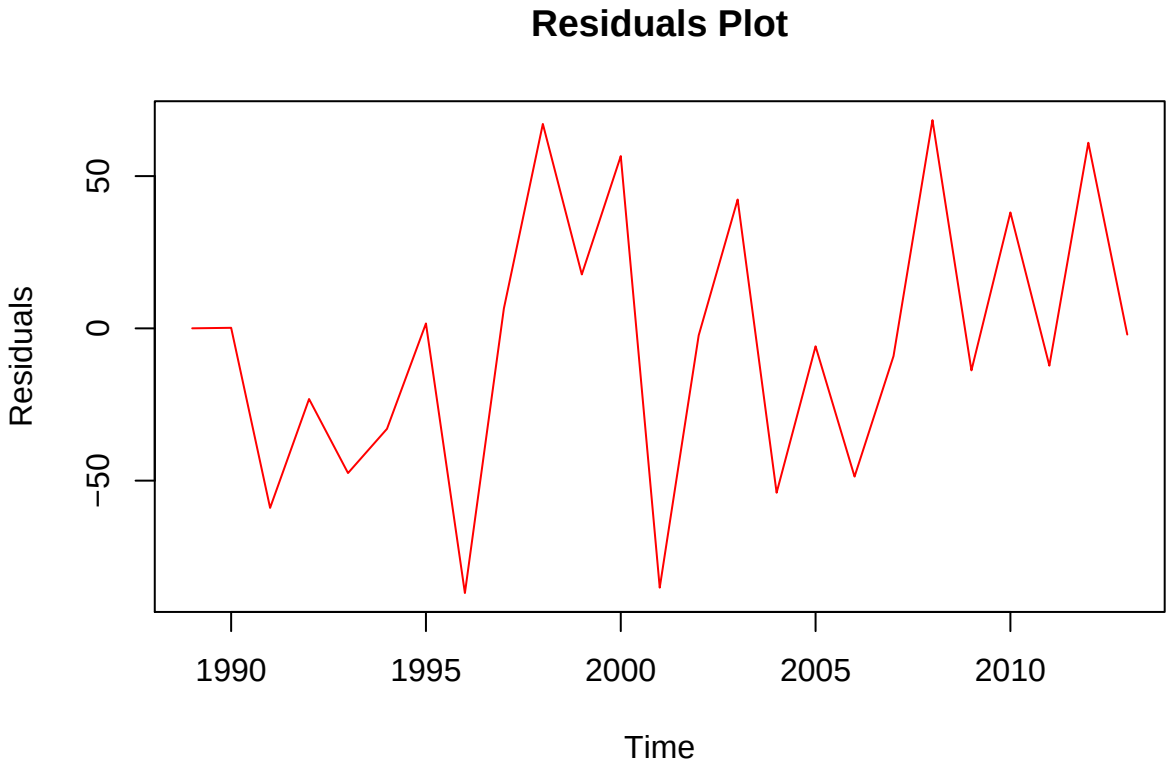
```
## [1]   0.0000   2.0000   2.0000 255.4478
```

```
#Therefore, the model follows an ARIMA(0,2,2) process

fit<-arima(ts,order=c(0,2,2))
fit
```

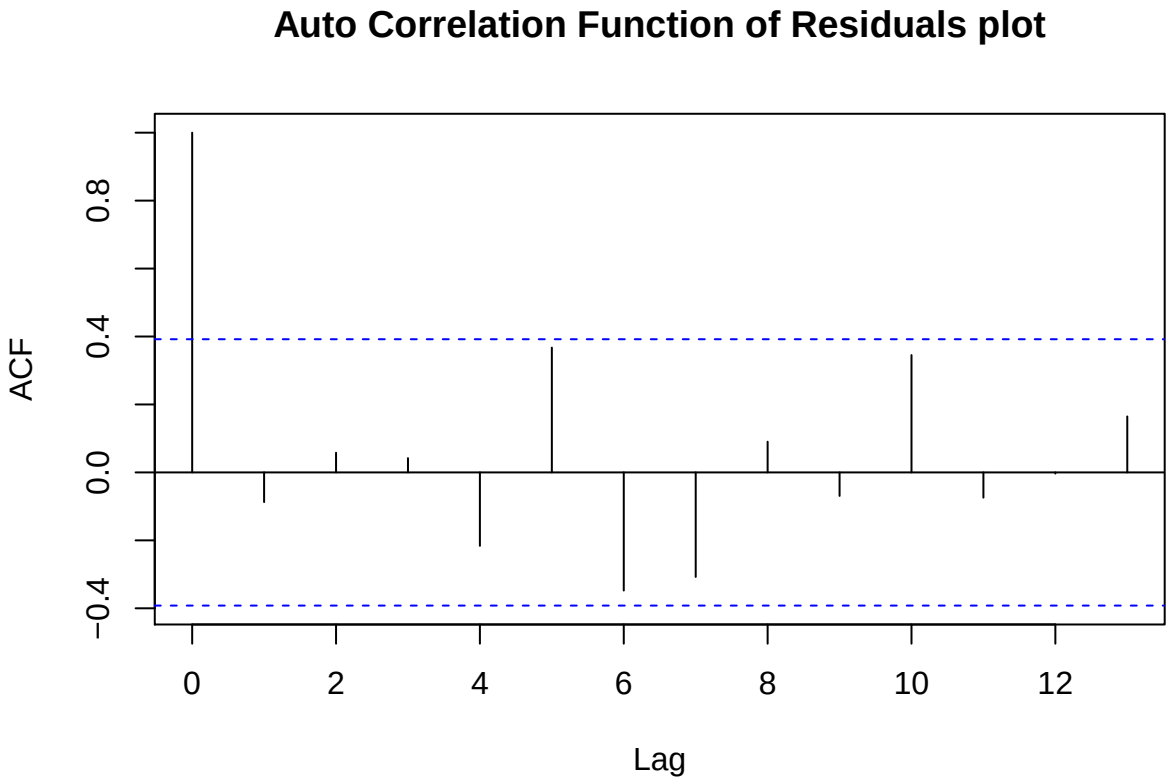
```
##
## Call:
```

```
## arima(x = ts, order = c(0, 2, 2))
##
## Coefficients:
##           ma1     ma2
##       -1.9714  1.0000
## s.e.   0.2083  0.2078
##
## sigma^2 estimated as 2070:  log likelihood = -124.72,  aic = 255.45
```

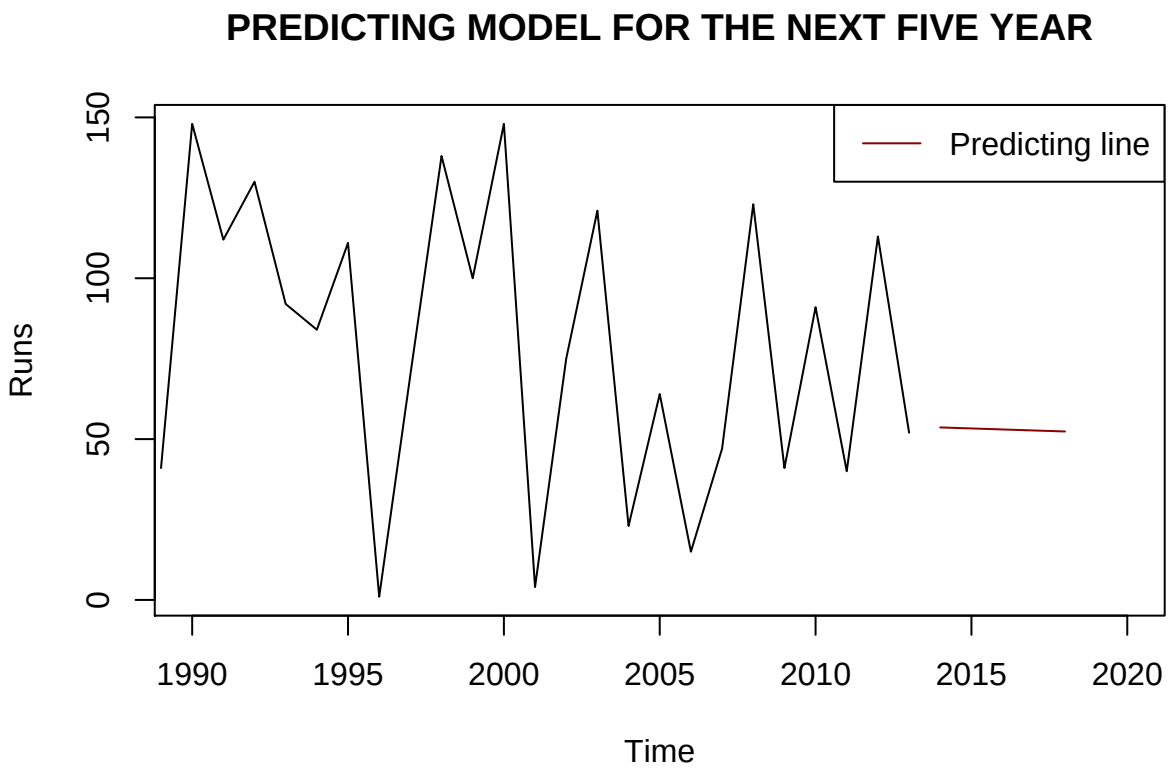
```
#Extracting the residuals of the time series
et<-fit$residuals
plot(et,main="Residuals Plot",ylab="Residuals",col="red")
```



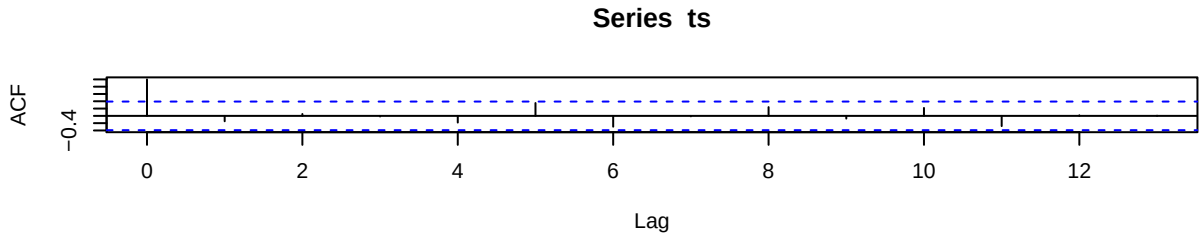

```
#Plot of Residuals is patternless
#There are almost equal number of positive and negative values
#Thus, we can conclude that ARIMA(0,2,2) process provied a good fit to our model
#We can therefore, predict future values using this model.

acf(et,main="Auto Correlation Function of Residuals plot")
```

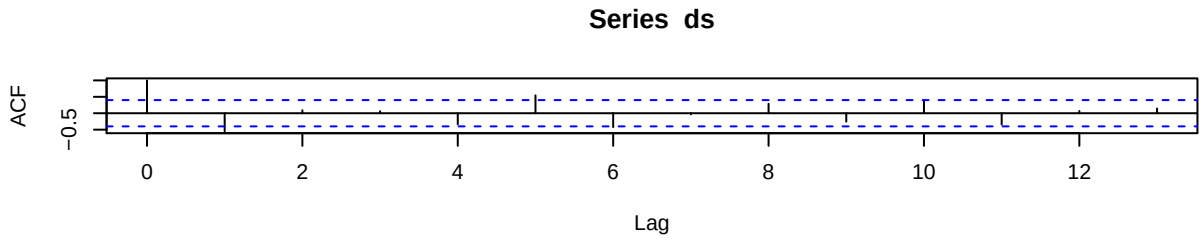


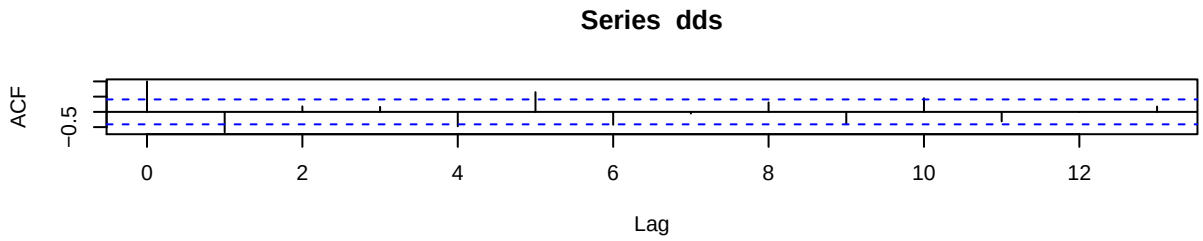

```
#Predicting future 5 years time series model:
pd<-predict(fit,n.ahead = 5)$pred
ts.plot(ts,xlim=c(1990,2020),main="PREDICTING MODEL FOR THE NEXT FIVE YEAR",ylab="Runs")
lines(pd,col="dark red")
legend("topright",legend="Predicting line",col="dark red",lty=1)
```



```r
#Differencing Data for Stationarity:
ds<-diff(ts)
dds<-diff(ds)

par(mfrow=c(3,1))
acf(ts)
acf(ds)
acf(dds)
```








```r
par(mfrow=c(1,1))

#Variance Test: Should be least
var(ts)
```

```
## [1] 2032.24
```

```r
var(ds)
```

```
## [1] 4773.389
```

```r
var(dds)
```

```
## [1] 14687.77
```

```r
#Since variance of ts is least and we can also observe from the diagram above,
#the model does not need to be differenced at all for the process to attain
#stationarity.


#In this article, I have used performed simulation and used Time Series Analysis
#for modelling and predicting the model.
```

```
#With respect to India's performance in ODI matches, I have conluded that India plays
#better if given a chance to bowl first and bat second in terms of runs and wickets down.

#For prediction of Sachin Tendulkar's performance, I have fitted an accurate time series
#process based on his performances over the years. Accuracy is certain since the residuals
#plot was patternless and there were equal number of negative and positive values.
#Using this fit, I have predicted his performance over the next 5 years.


#BIBLIOGRAPHY:
#https://stats.espncricinfo.com/ci/engine/records/index.html
```