Time Series HW 1

Akshay Kumar Varanasi (av32826)

February 5, 2019

Question 1

a. Fit an AR(1) model to the data using methods we learnt in class. Determine the maximum likelihood estimate of the autoregressive coefficientϕ_1 ?, variance of the white noise in the AR(1) model and the variance of the maximum likelihood estimate of the parameter ϕ_1.

```
## To read the data file
library("xlsx")
data<-read.xlsx("HW1_Problem1.xls",sheetIndex = 1, header=FALSE, stringsAsFacto
rs=FALSE)

## Hardcoding to find those parameters
sum=0
sum_sq=0
mu=mean(data$X1)
var_data=0

## Before going forward to do the model fitting, we need to subtract the data b
y its mean
data$X1<-data$X1-mu

## Calculating Phi_1
for (i in 1:49) {
     sum = sum + ((data$X1[i])*(data$X1[i+1]))
     sum_sq = sum_sq+((data$X1[i]))^2 #*(data$X1[i]))
}
phi_1=sum/sum_sq
print(paste("The value of phi_1 is",prettyNum(phi_1)))
```

```
## [1] "The value of phi_1 is -0.08323898"
```

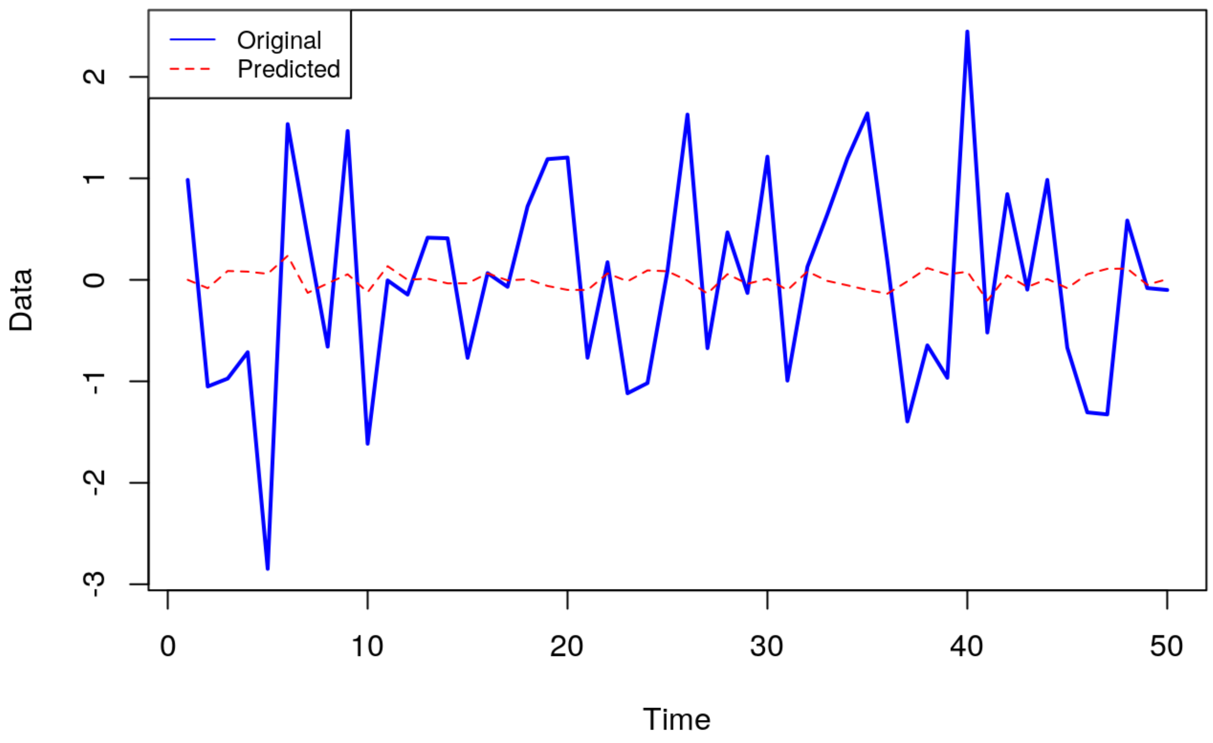
```r
## Calculating variance of white noise and phi_1
for (i in 1:49) {
  var_data=var_data+(data$X1[i+1]-phi_1*data$X1[i])^2
}
var_data=var_data/49
var_phi=var_data/sum_sq

print(paste("The value of variance of white noise is",prettyNum(var_data),"and
that of phi_1 is",prettyNum(var_phi)))
```

```
## [1] "The value of variance of white noise is 1.045042 and that of phi_1 is 0
.01989394"
```

```r
## In built function to fit data in AR model with order 1
model<-ar.mle(x = data$X1, aic = FALSE, order.max = 1)

## Prints out co-efficient and variance value of white noise
model
```

```
##
## Call:
## ar.mle(x = data$X1, aic = FALSE, order.max = 1)
##
## Coefficients:
##       1
## -0.0831
##
## Order selected 1  sigma^2 estimated as  1.043
```

```r
data$time<-1:length(data$X1)
data$predict<-rep(0,length(data$X1))
#data_predicted<-data_frame()
#data$predict<-predict(model, data$X1)

#data$predict<-predict(model, data$X1, n.ahead = 1, se.fit = TRUE)
for (i in 1:49) {
  data$predict[i+1]=data$X1[i]*phi_1
}

plot.ts(data$X1, lwd=2, col="Blue", xlab="Time", ylab="Data")
lines(x=data$time, y=data$predict, col='Red', lty=2 )
legend("topleft", legend=c("Original", "Predicted"),
       col=c("blue", "red"), lty=1:2, cex=0.8)
```



b. Comment on your result? What kind of a process does this look like?

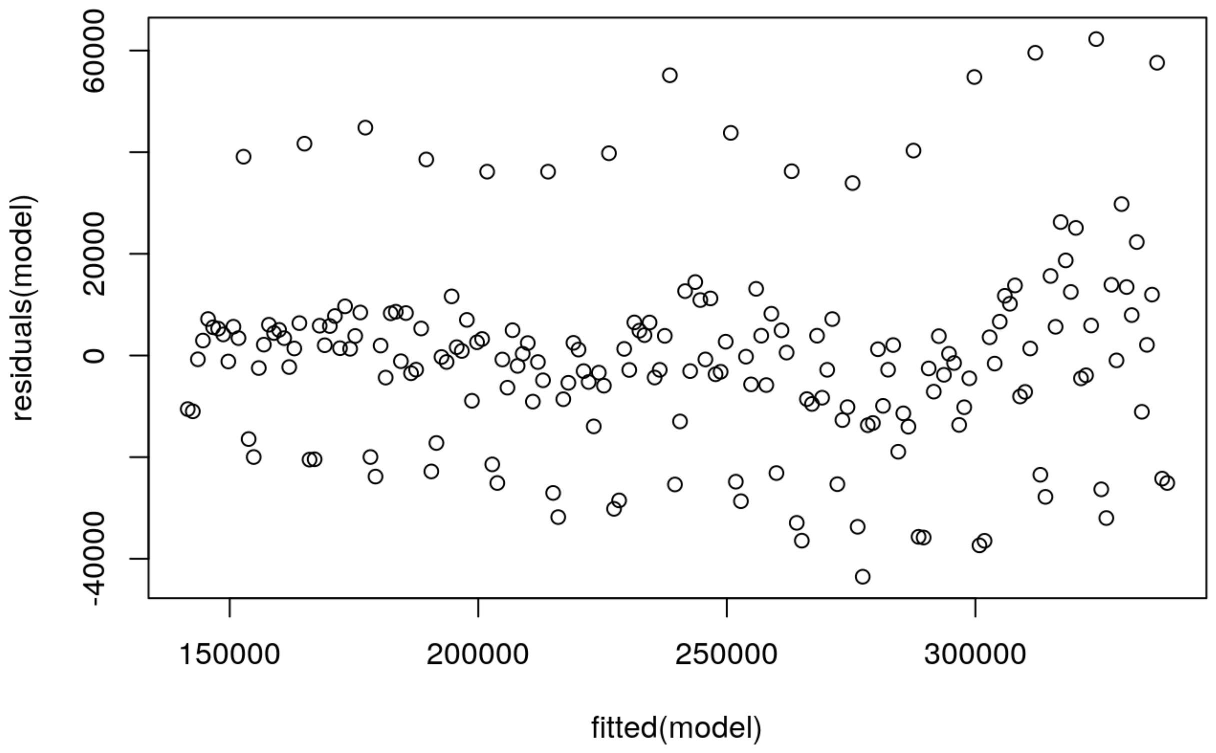
The process which we predicted is more or less stationary around a constant due to low value of phi. This is because the values are not dependent on previous values, i.e the fluctuations or the trend in actual data is mostly due to noise.

Question 2 For the retail sale data posted on the class website (originating form
before the economic calamity of 2007/2008), please do the following (a) Fit a line passing through the data (first order polynomial fit). Use regression
methods used in class, with time t being indexed from 0 to length(data)-1
(month is the unit of time)

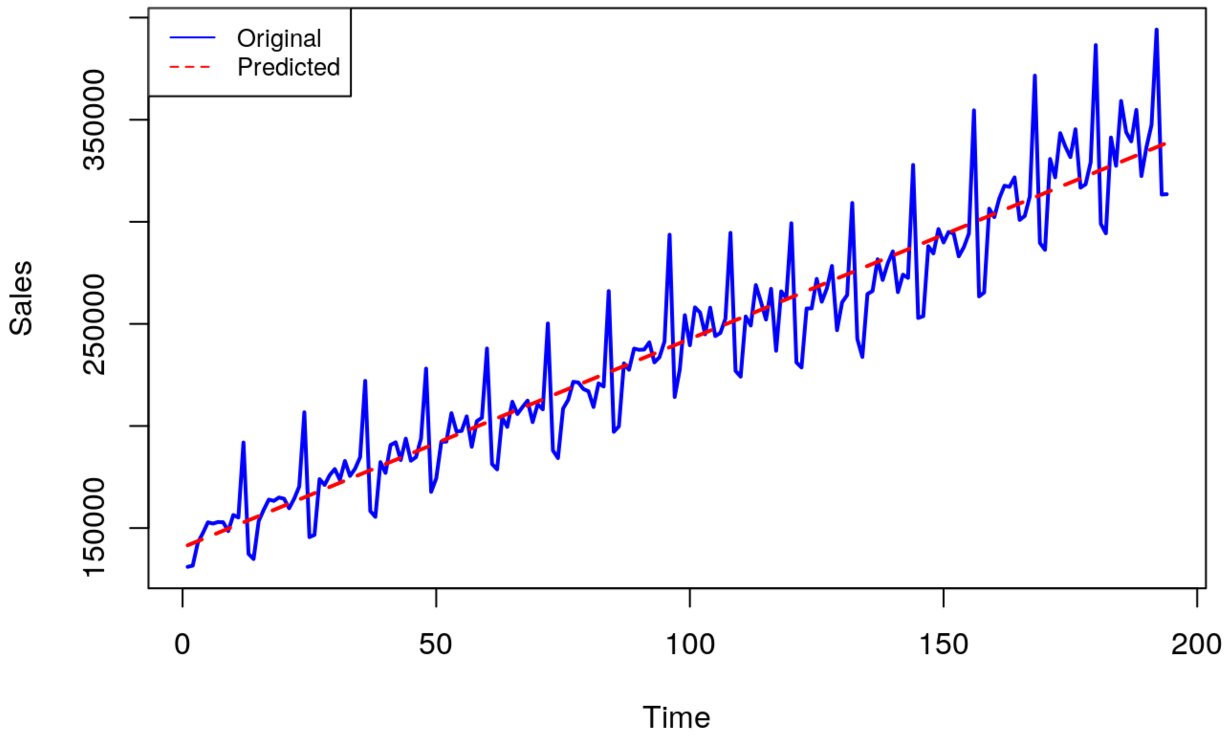
```
## Reading the data
library("xlsx")
data<-read.xlsx("Retail_Sales_Data.xlsx",sheetIndex = 1, header=FALSE, stringsA
sFactors=FALSE)

## Changing the column names for our convinience
colnames(data)<-c("Year","Month","Sales")

## Removing first 2 lines
data_sub<-data[3:length(data$Year), ]

## Converting the date and month to numbers
data_sub$Time<-c(1:length(data_sub$Sales))

## Creating a linear model relating Sales and time
model<-lm(data_sub$Sales~data_sub$Time)
summary(model)
```

```
##
## Call:
## lm(formula = data_sub$Sales ~ data_sub$Time)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -43532  -9425     45   6457  62248
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   140526.68    2756.66   50.98   <2e-16 ***
## data_sub$Time   1020.99      24.52   41.64   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 19120 on 192 degrees of freedom
## Multiple R-squared:  0.9003, Adjusted R-squared:  0.8998
## F-statistic:  1734 on 1 and 192 DF,  p-value: < 2.2e-16
```

```
## Plotting the model
plot(fitted(model),residuals(model))
```



```
## Predicted values
pre_variables<-predict(model,data=data_sub$Time)

## Plotting both original and predicted values
plot.ts(data_sub$Sales, lwd=2, col="Blue", xlab="Time", ylab="Sales")
lines(x=data_sub$Time, lwd=2, y=pre_variables, col='Red', lty=2)
legend("topleft", legend=c("Original", "Predicted"),
       col=c("blue", "red"), lty=1:2, cex=0.8)
```



b. Calculate residuals after you removed the first order fit and then fit an autoregressive model of order 2 – AR(2) to those residuals (again, months are units of time). Please mark the residual sum of squares after you fitted the AR(2) model.

```
data_sub$Predicted<-pre_variables
data_sub$Resdiuals<-as.numeric(data_sub$Sales)-as.numeric(data_sub$Predicted)

# AR(2)
res<-ar.ols(x = data_sub$Resdiuals, aic = FALSE, order.max = 2)
res
```

```
##
## Call:
## ar.ols(x = data_sub$Resdiuals, aic = FALSE, order.max = 2)
##
## Coefficients:
##       1        2
## -0.0521  -0.2451
##
## Intercept: 184.8 (1336)
##
## Order selected 2  sigma^2 estimated as  342373908
```

```
# Residual Sum of squares
print(paste("The value of residual sum of squares for AR(2) is",prettyNum((leng
th(data_sub$Year)-1)*res$var.pred)))
```

```
## [1] "The value of residual sum of squares for AR(2) is 66078164280"
```

```
#data_sub$Resdiuals_pre<-predict(res,data_sub$Resdiuals)
#plot.ts(data_sub$Resdiuals)
#lines(x=data_sub$Time,y=data_sub$Resdiuals_pre)
```

c. Take the same residuals obtained after fitting a line through the sales data, but now please fit an AR(4) model. Once again mark the residual sum of squares after you fitted the AR(4) model.

```
# AR(4)

res_4<-ar.ols(x = data_sub$Resdiuals, aic = FALSE, order.max = 4)
res_4
```

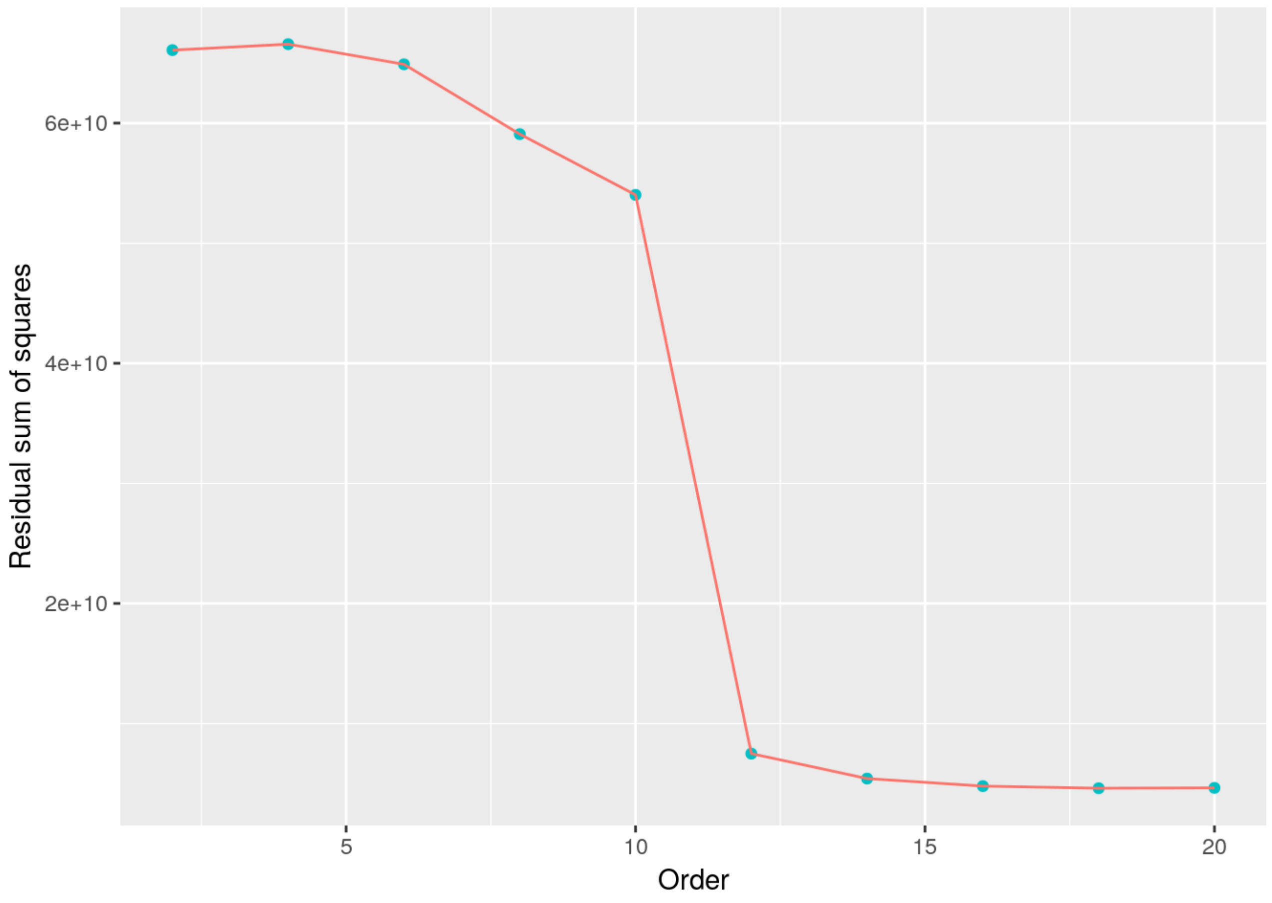
```
##
## Call:
## ar.ols(x = data_sub$Resdiuals, aic = FALSE, order.max = 4)
##
## Coefficients:
##       1        2        3        4
## -0.0632  -0.2438  -0.0499   0.0181
##
## Intercept: 211.1 (1348)
##
## Order selected 4  sigma^2 estimated as  344935201
```

```
# Residual Sum of squares
print(paste("The value of residual sum of squares for AR(4) is",prettyNum((leng
th(data_sub$Year)-1)*res_4$var.pred)))
```

```
## [1] "The value of residual sum of squares for AR(4) is 66572493864"
```

d. Increase the order of the AR(k) model by fitting AR(6), AR(8), AR(10) and so on models. Do this until you fit AR(20). Please plot the residual sums of squares of these models as you increased the order of the AR models. Please comment on what you see

```
res_sum_sq=rep(0L,10)
for (i in seq(2,20,2)){
  res<-ar.ols(x = data_sub$Resdiuals, aic = FALSE, order.max = i)

  res_sum_sq[i/2]=res$var.pred*(length(data_sub$Year)-1)
}
res_sum<-data.frame(res_sum_sq)
#res_sum$res_sum_sq<-res_sum_sq
res_sum$model_num=seq(2,20,2)

library(ggplot2)
ggplot(res_sum,aes(x=res_sum$model_num, y=res_sum$res_sum_sq))+geom_point(aes(c
olor = 'Red'),show.legend = FALSE)+geom_line(aes(color = 'green'),show.legend =
FALSE)+xlab("Order")+ylab("Residual sum of squares")
```



From the plot, we can see how residual sum of squares drops off suddenly after some increase in order number then stays around that value as we increase further. It is because of as we increase number of variables to fit the fit becomes better.