**Time Series Business Case**

**Objective:- To forecast the total number of passenger’s movement from UK for the next four quarters.**

**Language used: R**

***#importing packages for Time series analysis and forecasting***

***library****(readxl)*

***library****(ggplot2)*

***library****(forecast)*

***library****(xts)*

***library****(fpp2)*

***library****(TTR)*

***library****(dplyr)*

***library****(hrbrthemes)*

***library****(tseries)*

***library****(StatMeasures)*

***#Remove the first record from the excel file (skip = 1)***

*demand <- read\_excel("E:\\dataset\\revexx\\UK Outward Passengers Movement.xls", skip = 1)*

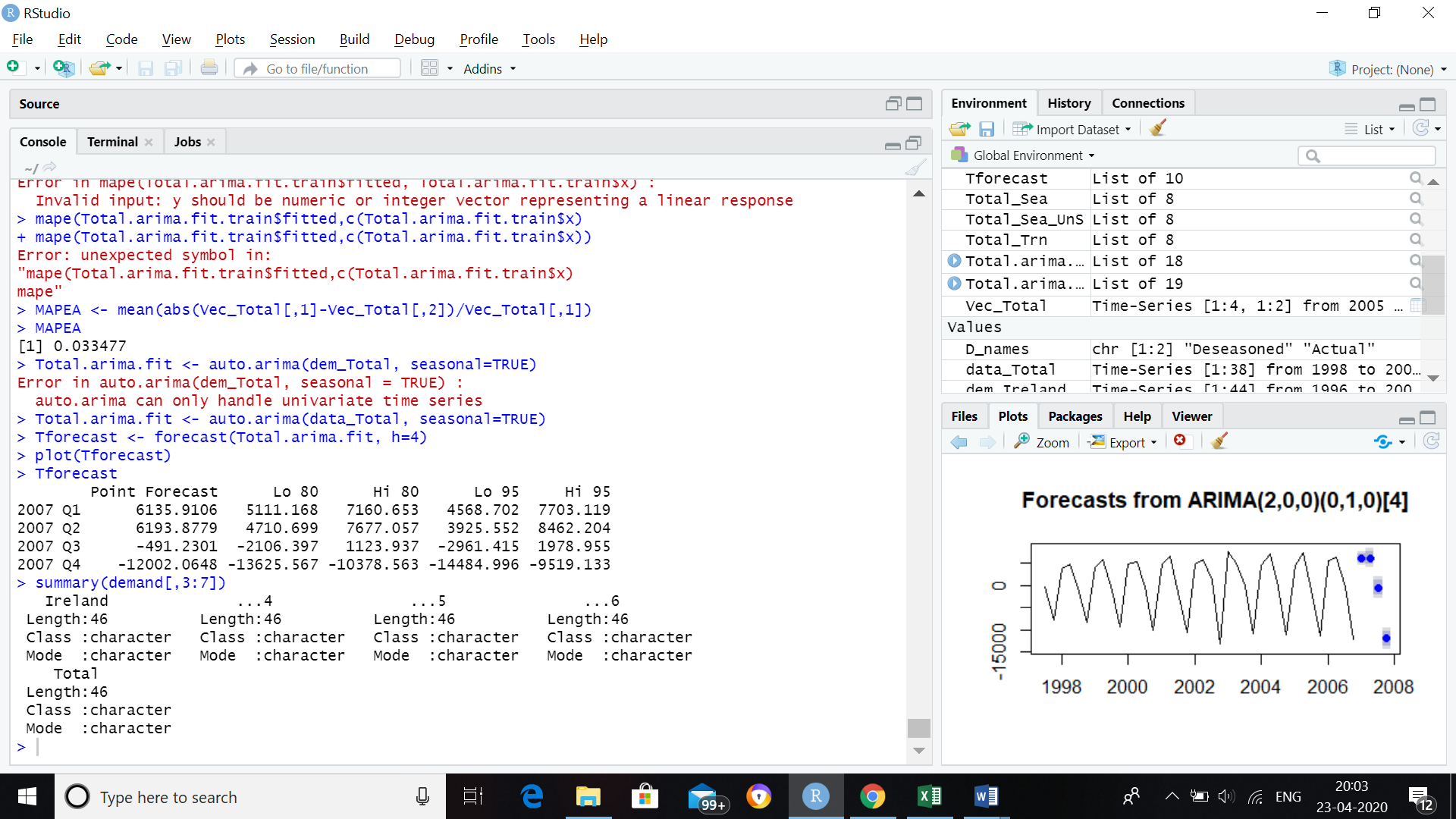
*str(demand)*

*names(demand)[3] <- c("Ireland")*

*names(demand)[7] <- c("Total")*

*str(demand)*

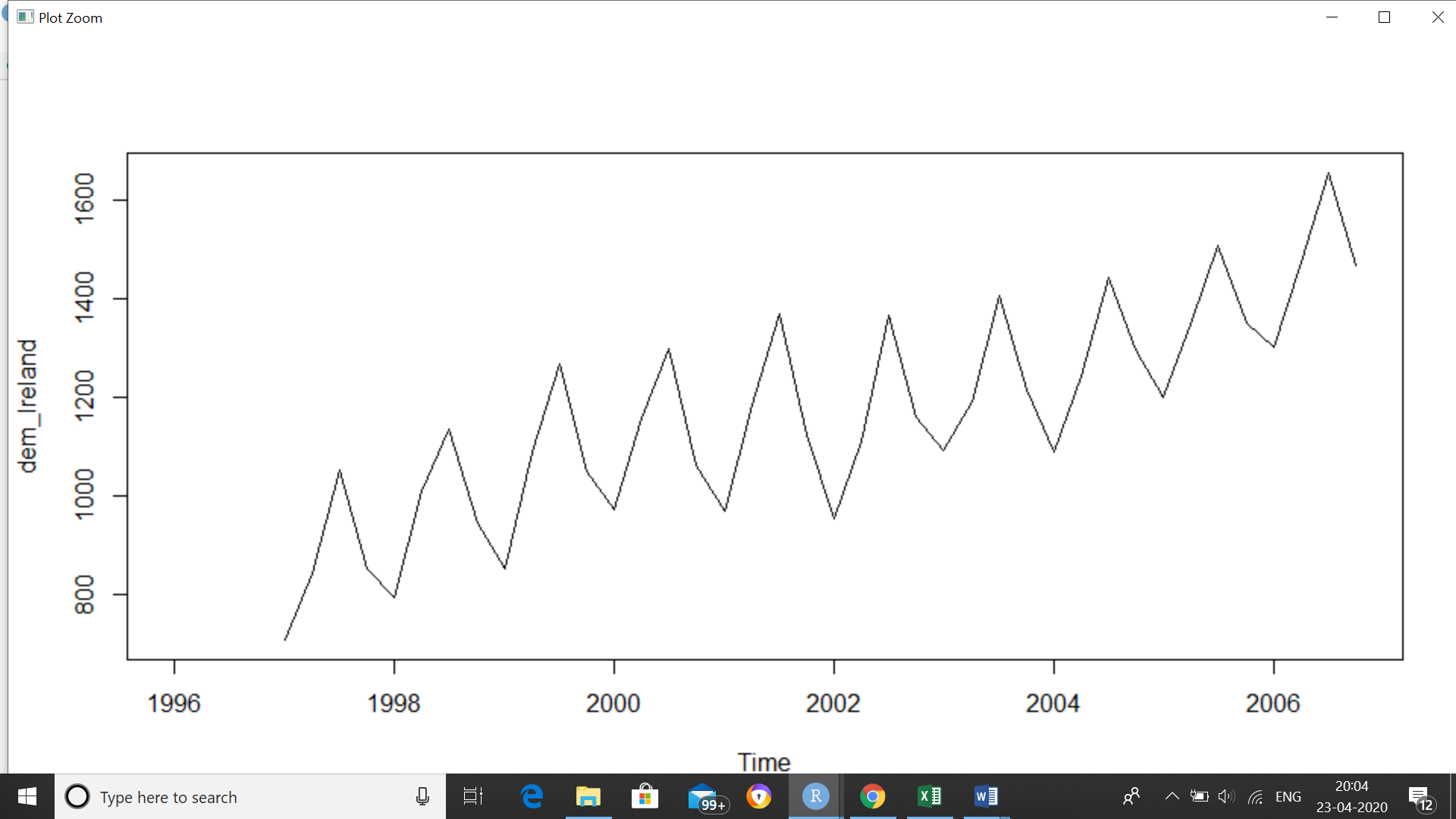
*summary(demand[,3:7])*



***#Plotting Time Series for Ireland for quarterly data from year 1996 to 2006***

*dem\_Ireland <- ts(demand[,3], start=c(1996,1), end=c(2006,4), frequency=4)*

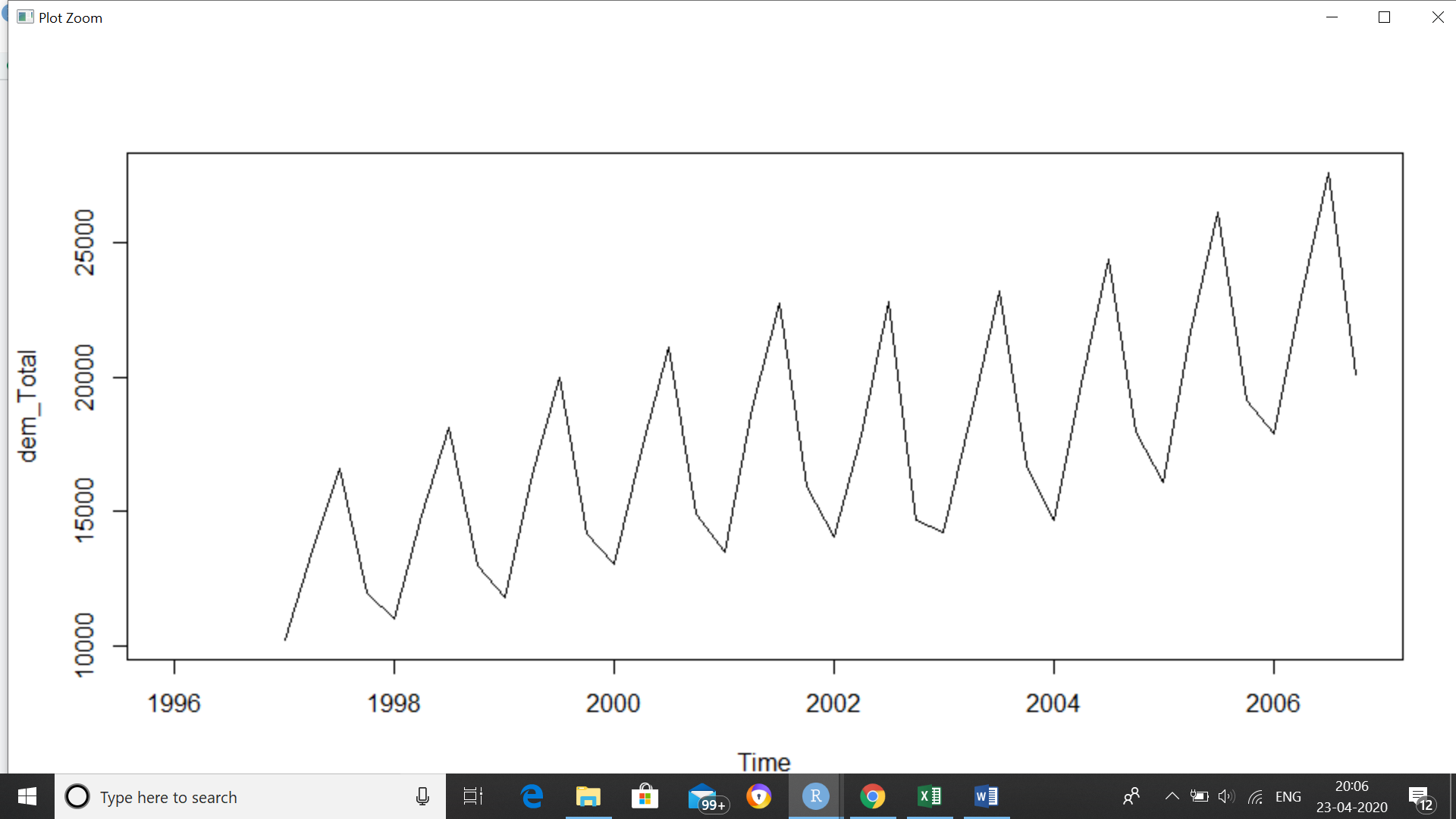
*plot(dem\_Ireland)*



***#Plotting Time Series for Total for quarterly data from year 1996 to 2006***

*dem\_Total <- ts(demand[,7], start=c(1996,1), end=c(2006,4), frequency=4)*

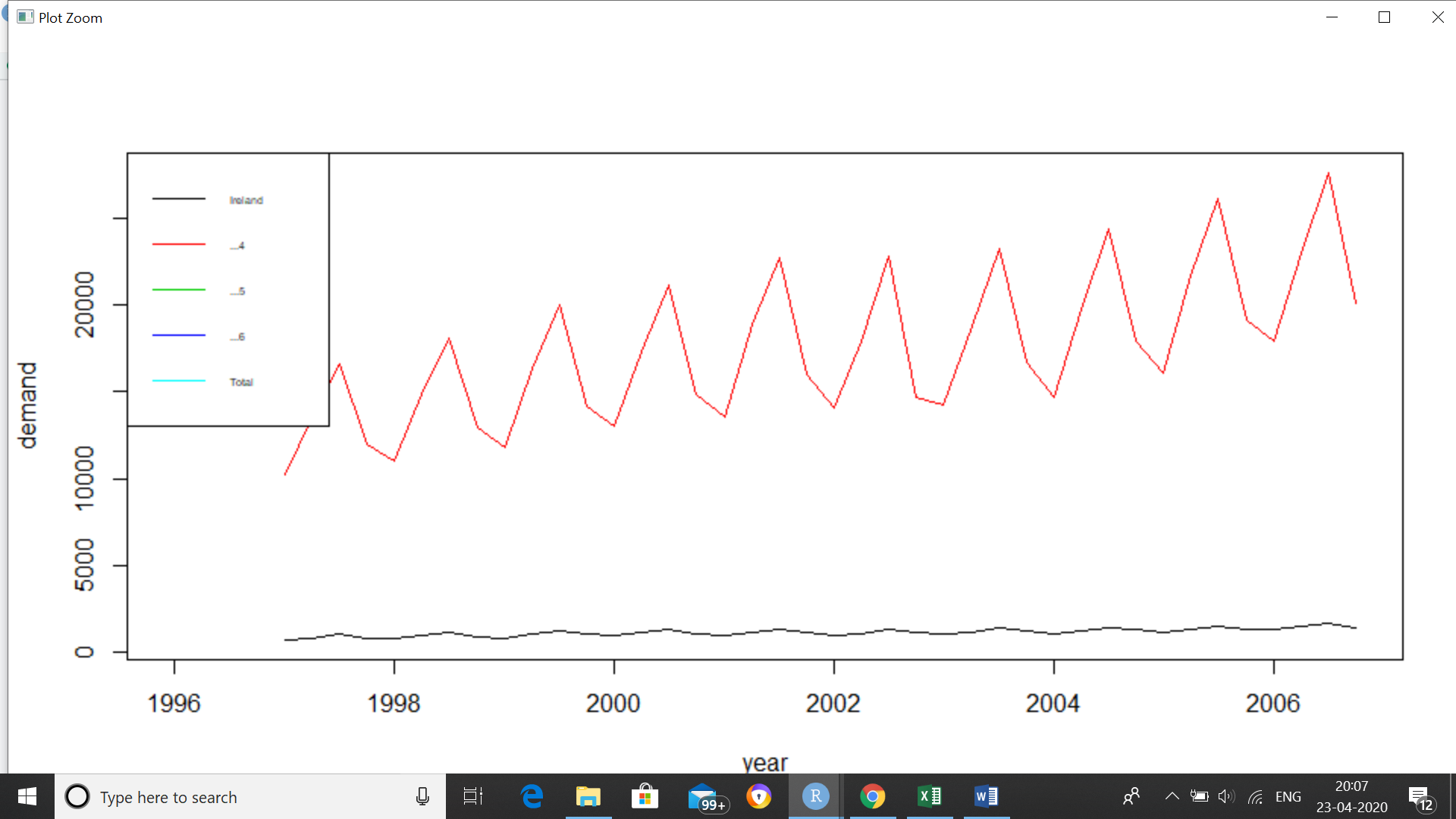
*plot(dem\_Total)*



***#Plotting the Time Series across Ireland and Total***

*ts.plot(dem\_Ireland, dem\_Total, gpars = list(col = c("black", "red")),xlab="year", ylab="demand")*

*legend("topleft", colnames(demand[3:7]), col=1:ncol(demand), lty=1.9, cex=.45)*



***# this shows that our attributes in the dataset are independent of each other and hence we***

***#can perform Univariate time series forecasting on it***

***#working on Total data quarterly***

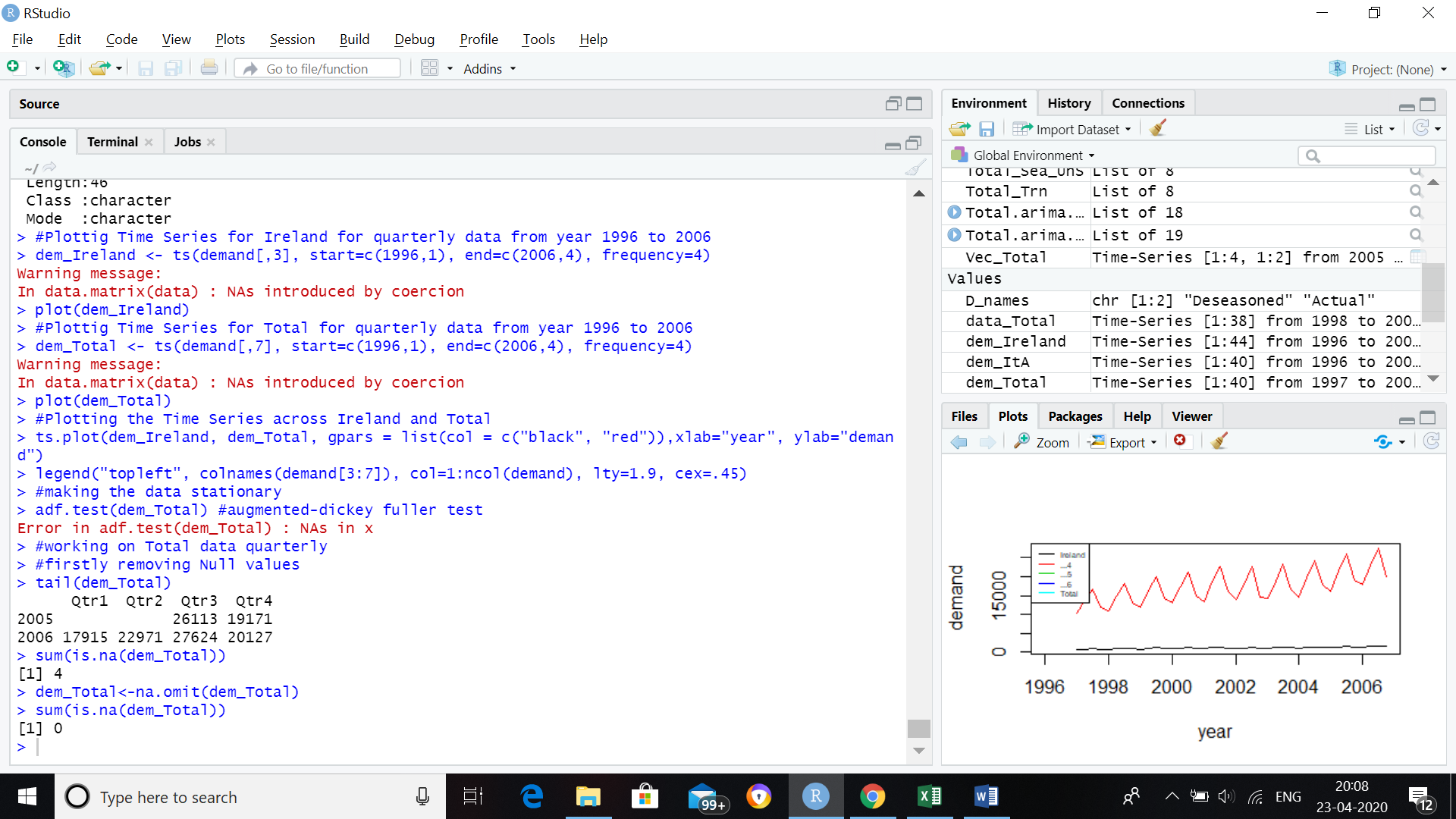
***#firstly removing Null values***

*tail(dem\_Total)*

*sum(is.na(dem\_Total))*

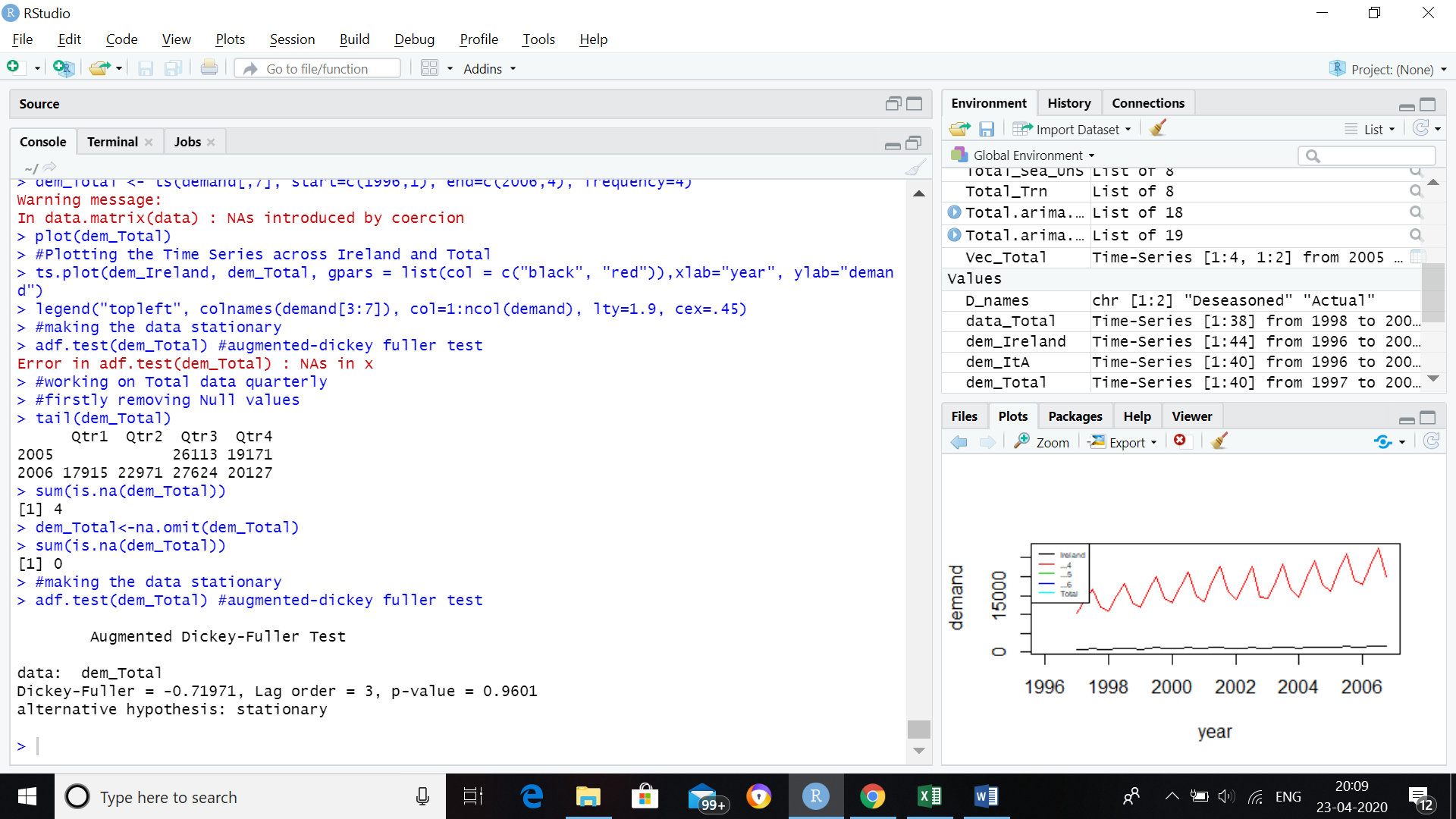
*dem\_Total<-na.omit(dem\_Total)*

*sum(is.na(dem\_Total))*



***#making the data stationary***

*adf.test(dem\_Total)* ***#augmented-dickey fuller test***



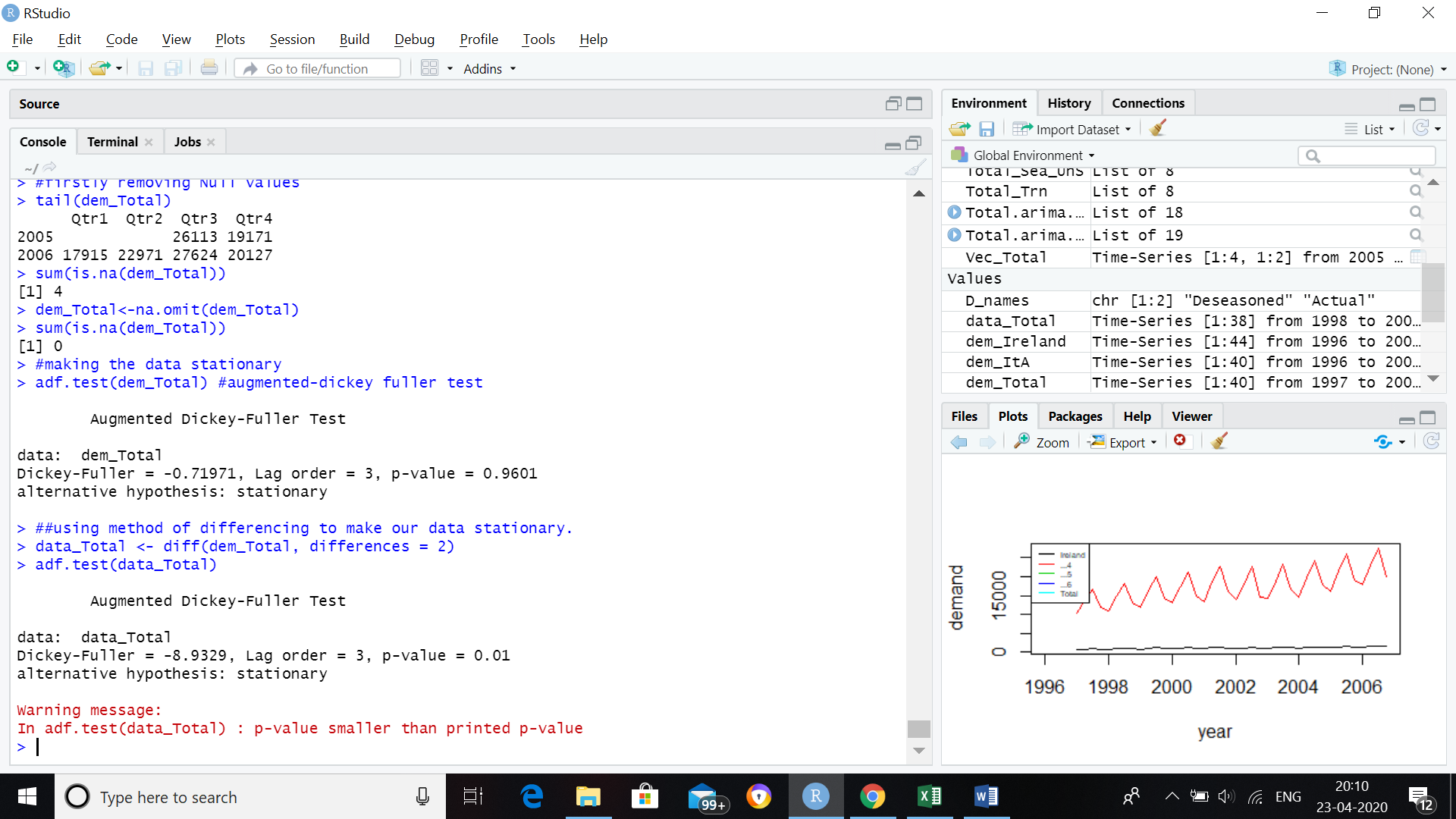
***#The p-value turns out be 0.96.***

***#We thus fail to reject our Ho and conclude that the data is not stationary.***

***##using method of differencing to make our data stationary.***

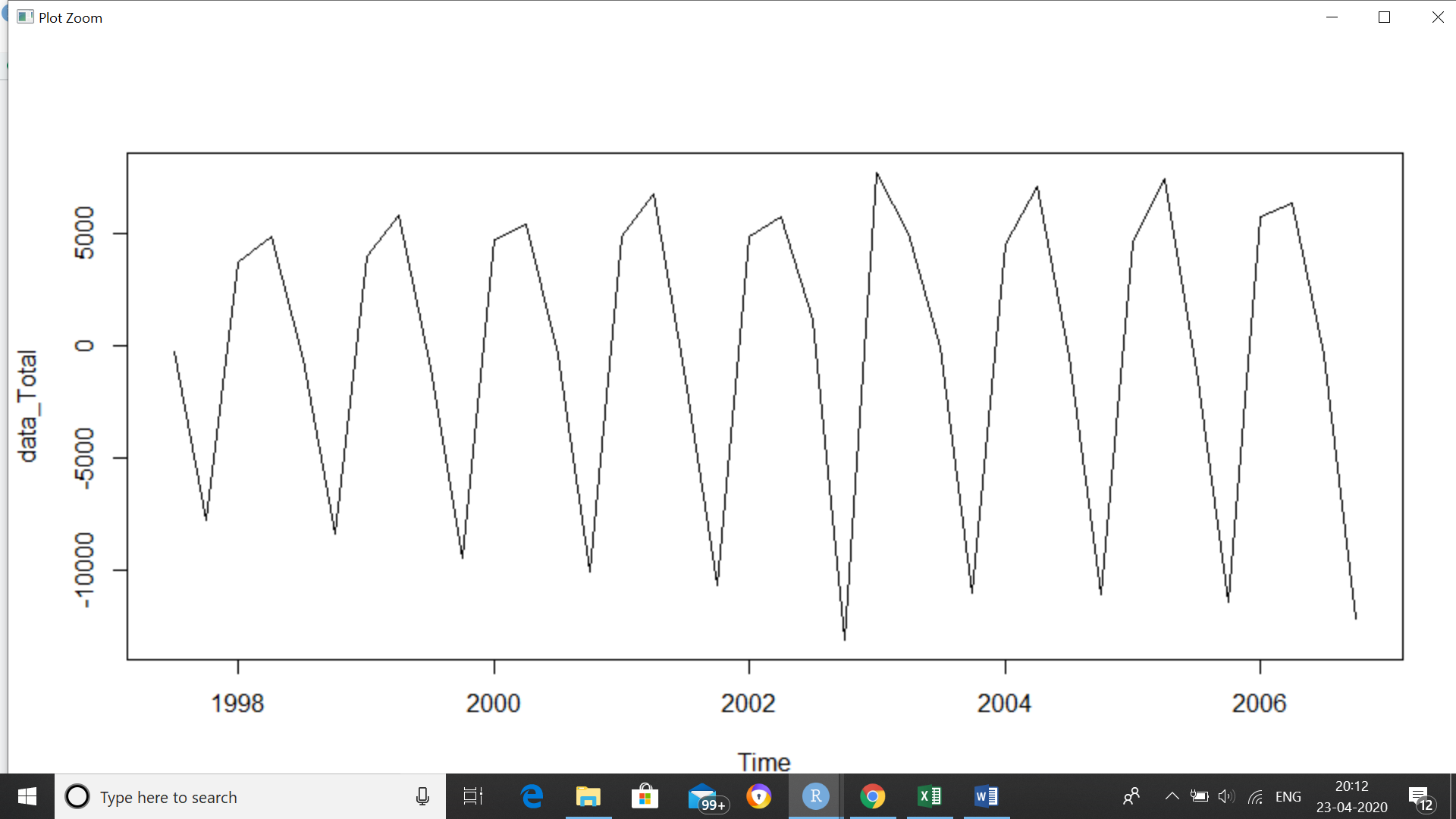
*data\_Total <- diff(dem\_Total, differences = 2)*

*adf.test(data\_Total)*



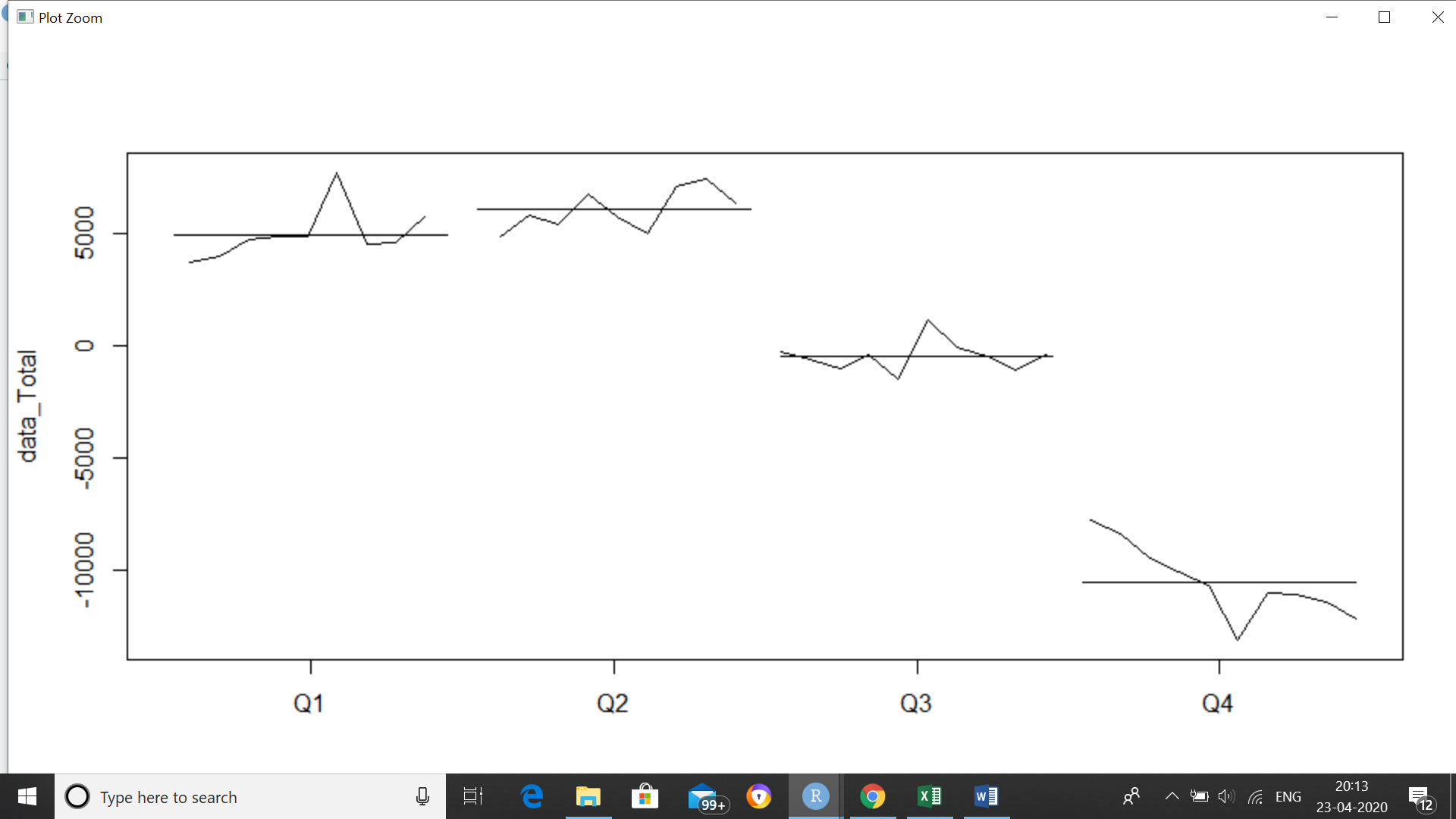
***#p-value is now smaller than printed p-value, that means data is stationary now.***

*plot(data\_Total)*



***#decomposition of time series data set - Seasonal, Trend and Random***

*monthplot(data\_Total)*



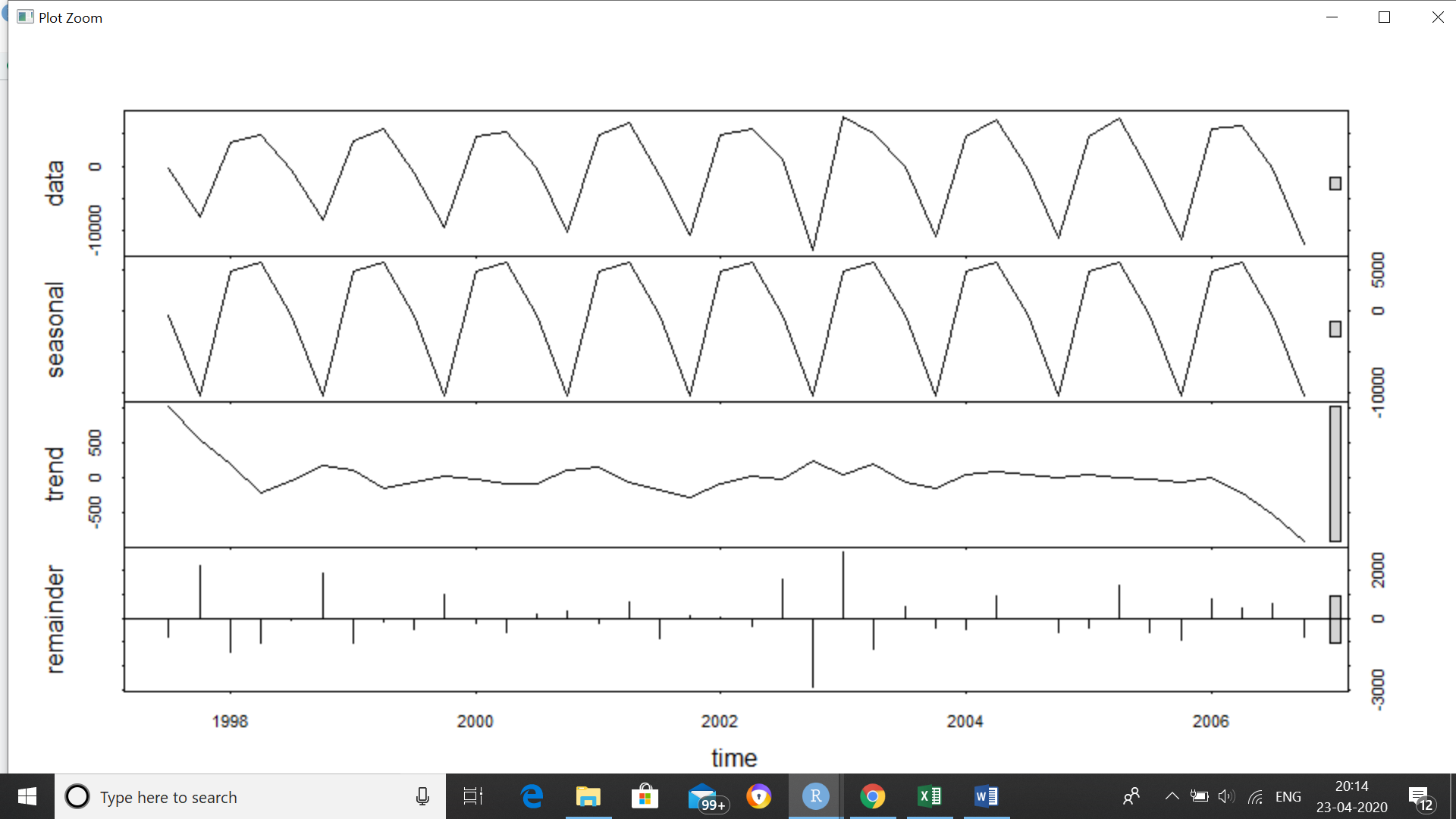
*boxplot (data\_Total ~cycle(data\_Total))*



***#Decomposing the time series using STL (Seasonal and Trend decomposition using Loss).***

*Total\_Sea <- stl(data\_Total, s.window="p")* ***#constant seasonality***

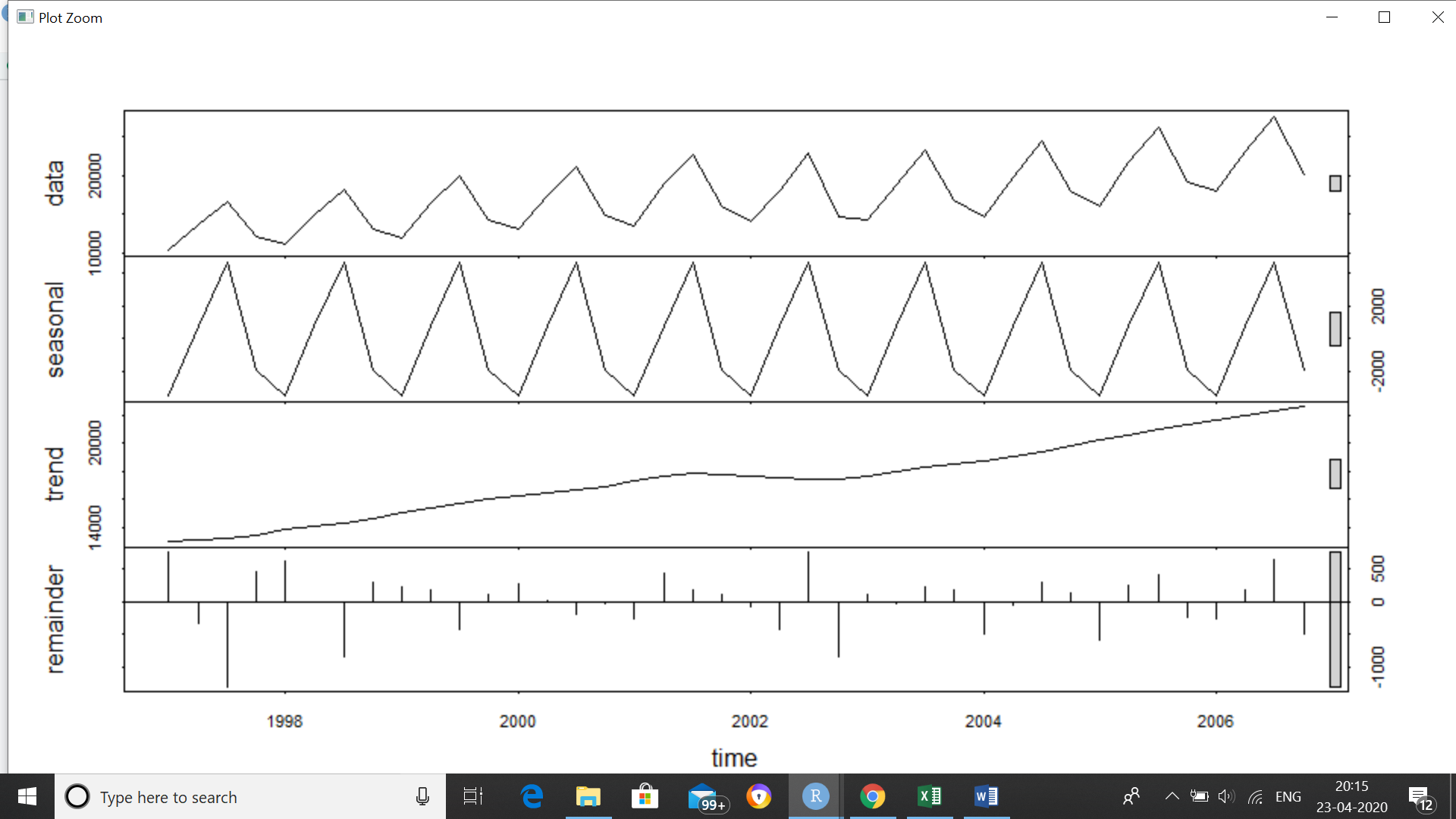
*plot(Total\_Sea)*



***#on original data, i.e., non-stationary***

*Total\_Sea\_UnS <- stl(dem\_Total, s.window="p")* ***#constant seasonality***

*plot(Total\_Sea\_UnS)*



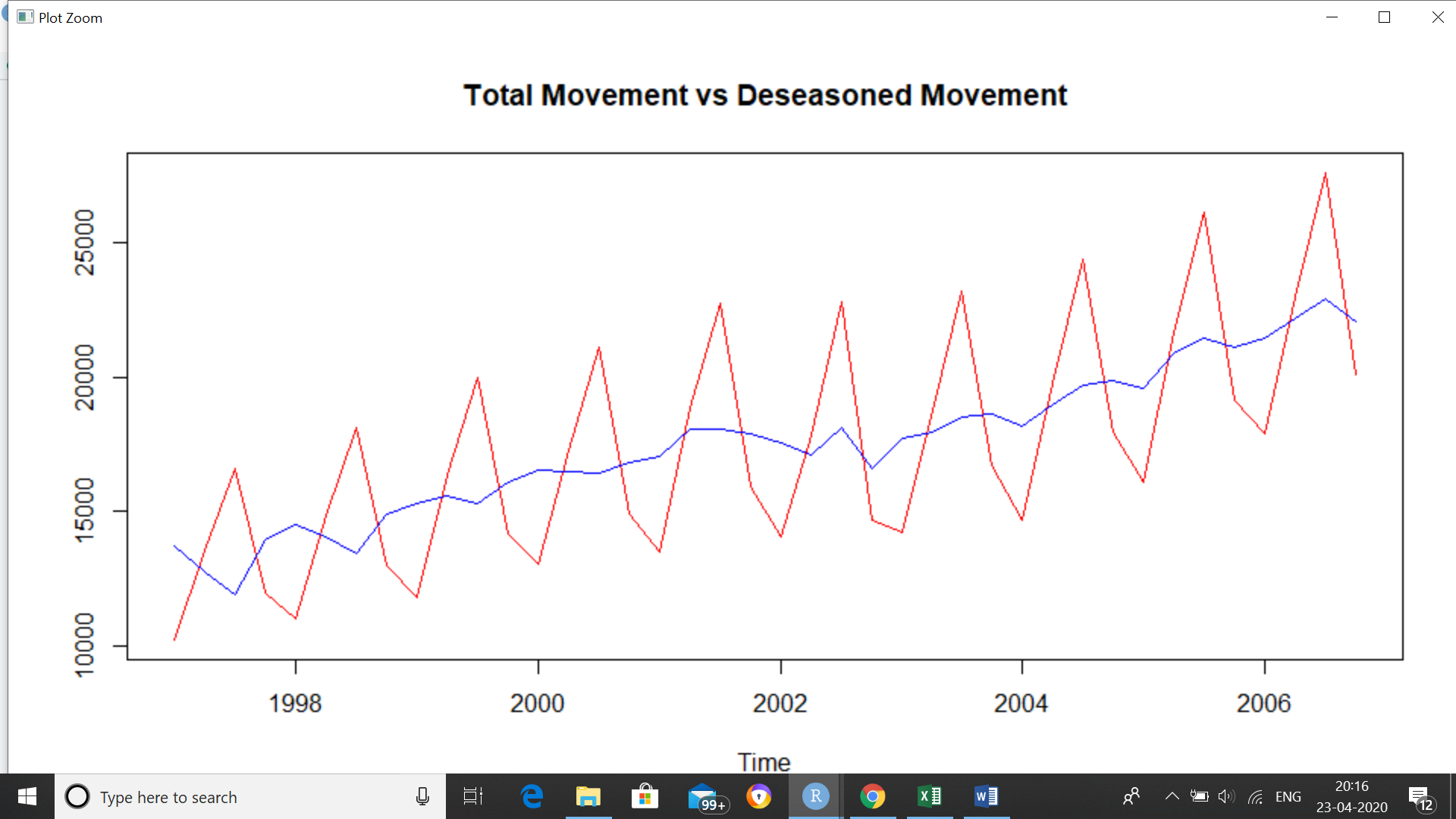
***#according to seasonality there was increase in overall movement from 1996 to around 2002 then there was slight fall near 2003 and then it increased again till 2006***

***#to forecast for next quarter we have to take into account both trend and seasonality***

*D\_names <- c('Deseasoned', 'Actual')*

*Deseason\_Total <- (Total\_Sea\_UnS$time.series[,2]+Total\_Sea\_UnS$time.series[,3])*

*ts.plot(dem\_Total, Deseason\_Total, col=c("red", "blue"), main="Total Movement vs Deseasoned Movement")*



***#show Movement in Red and de-seasoned movement in Blue***

***#we can see that there is increasing trend of movement.***

***#dividing data into test and train***

*Total\_Train <- window(dem\_Total, start=c(1996,1), end=c(2004,4), frequency=4)*

*Total\_Test <- window(dem\_Total, start=c(2005,1), frequency=4)*

*Total\_Test*

***#Convert into seasonal, trend and irregular component using STL***

*Total\_Trn <- stl(Total\_Train, s.window="p")*

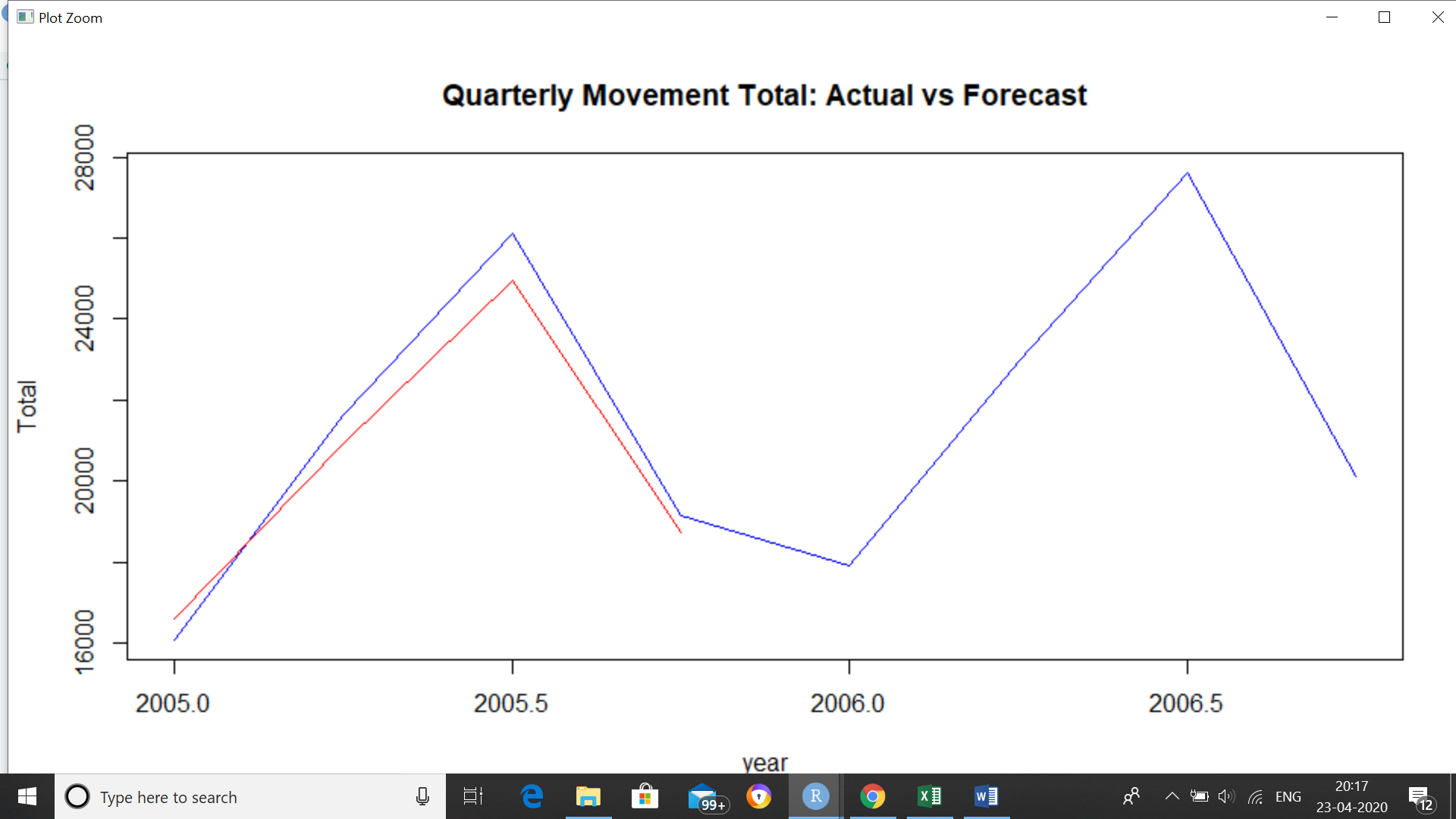
***#model building***

***##Random walk with drift model - Forecast on train data***

*fcst.Total.stl <- forecast(Total\_Trn, method="rwdrift", h=4)*

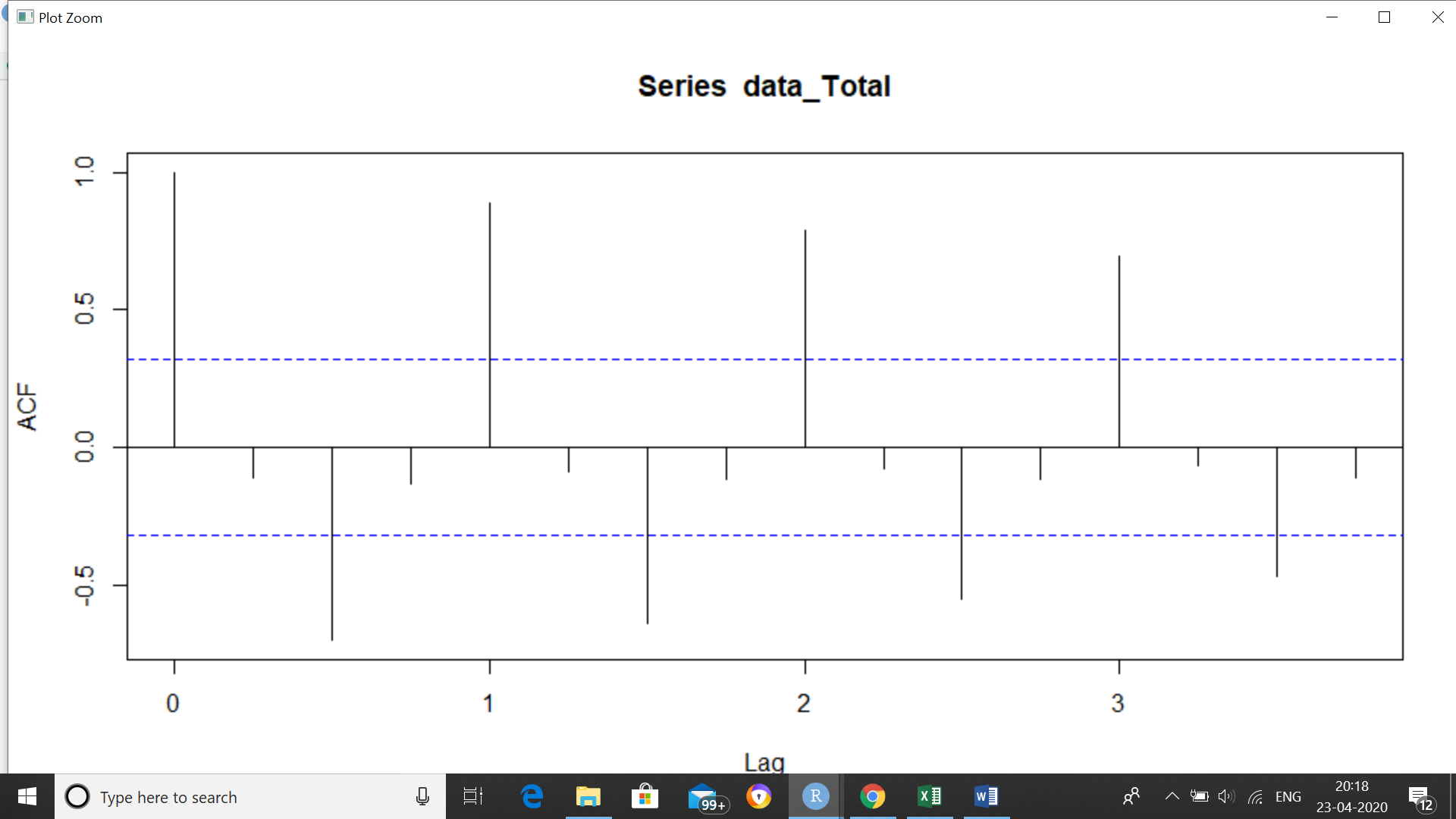
*Vec\_Total<- cbind(Total\_Test,fcst.Total.stl$mean)*

*ts.plot(Vec\_Total, col=c("blue", "red"),xlab="year", ylab="Total", main="Quarterly Movement Total: Actual vs Forecast")*

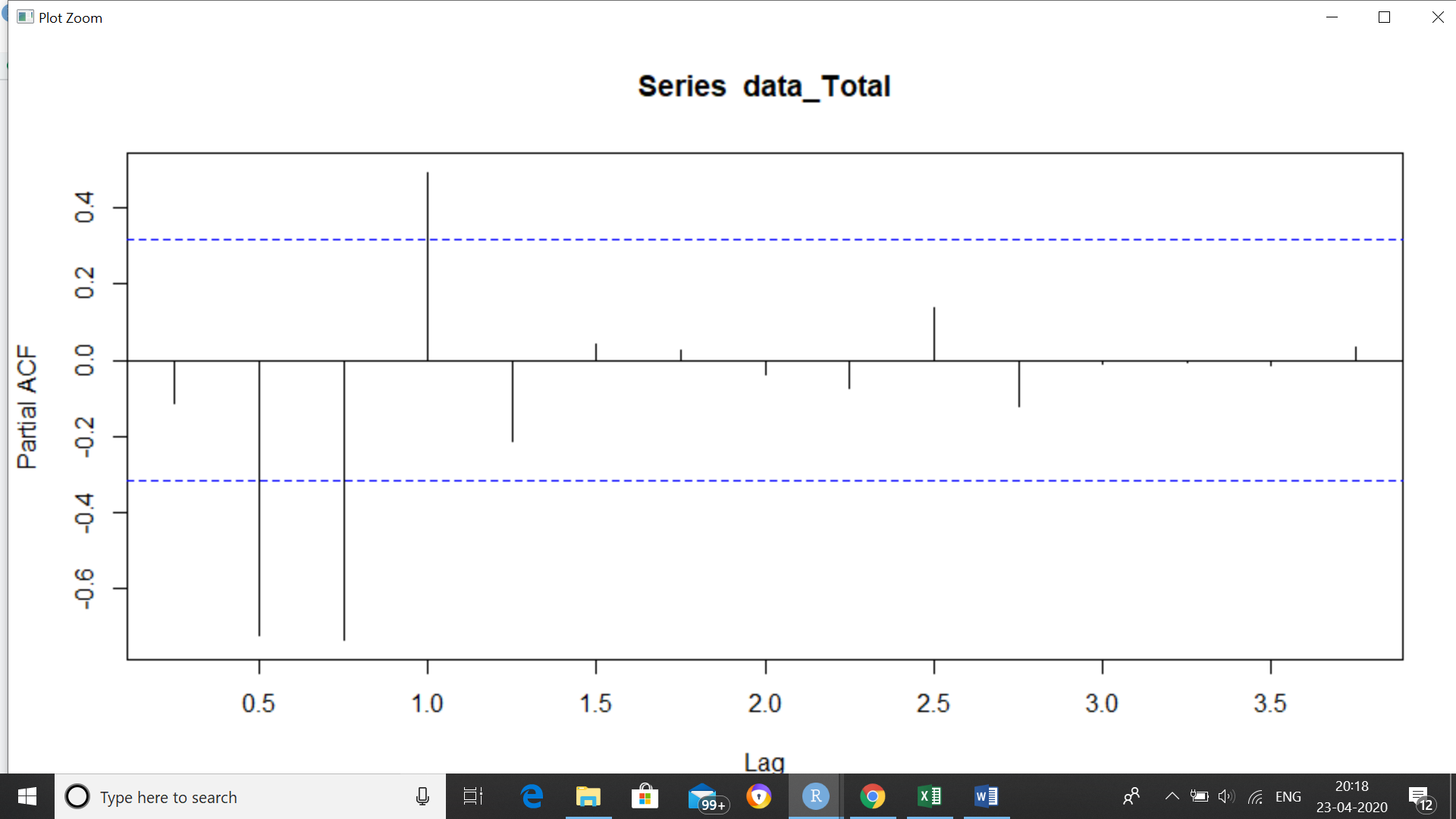


***#using acf and pacf to check the lags***

*acf(data\_Total)*



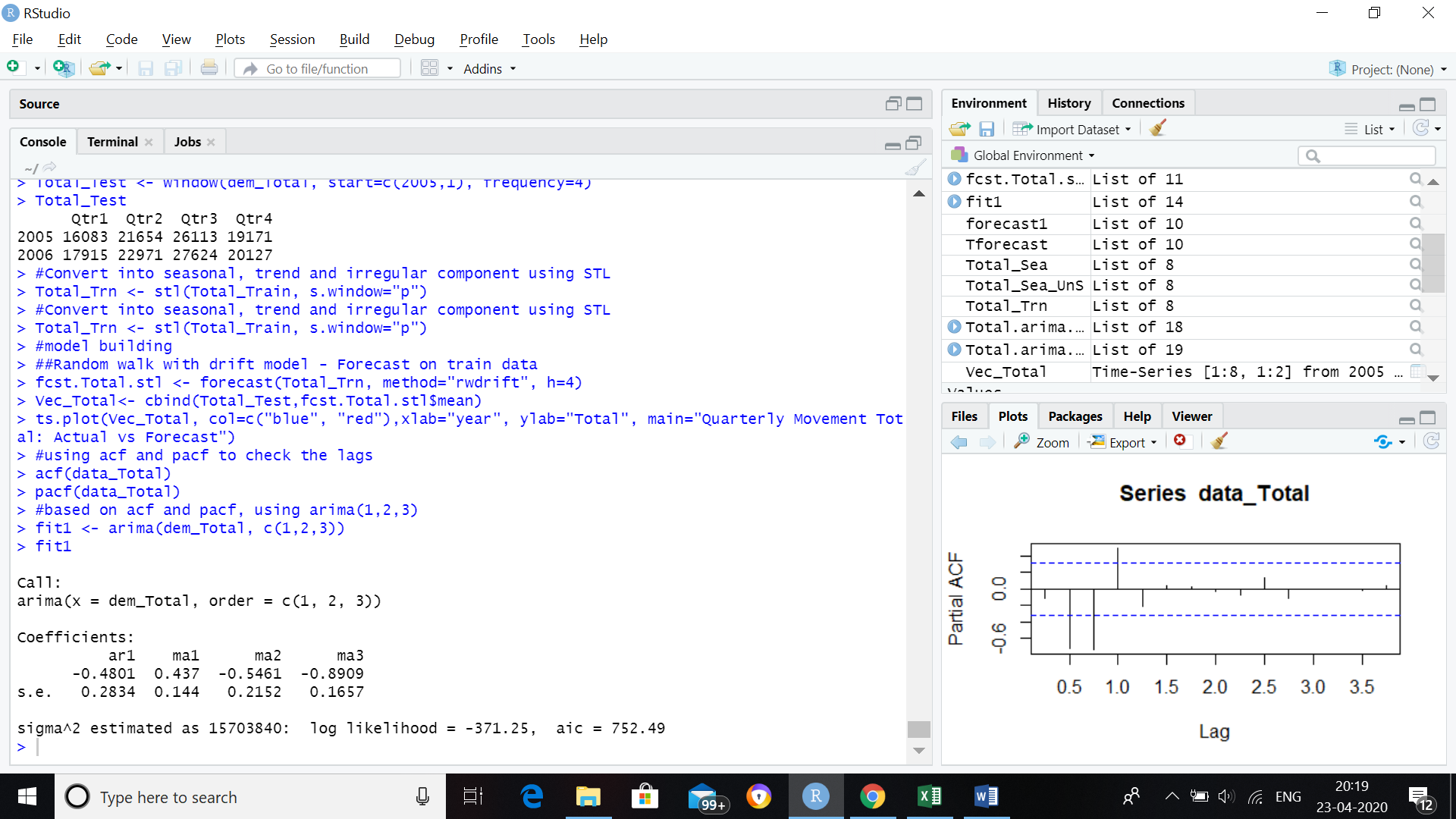
*pacf(data\_Total)*



***#based on acf and pacf, using arima(1,2,3)***

*fit1 <- arima(dem\_Total, c(1,2,3))*

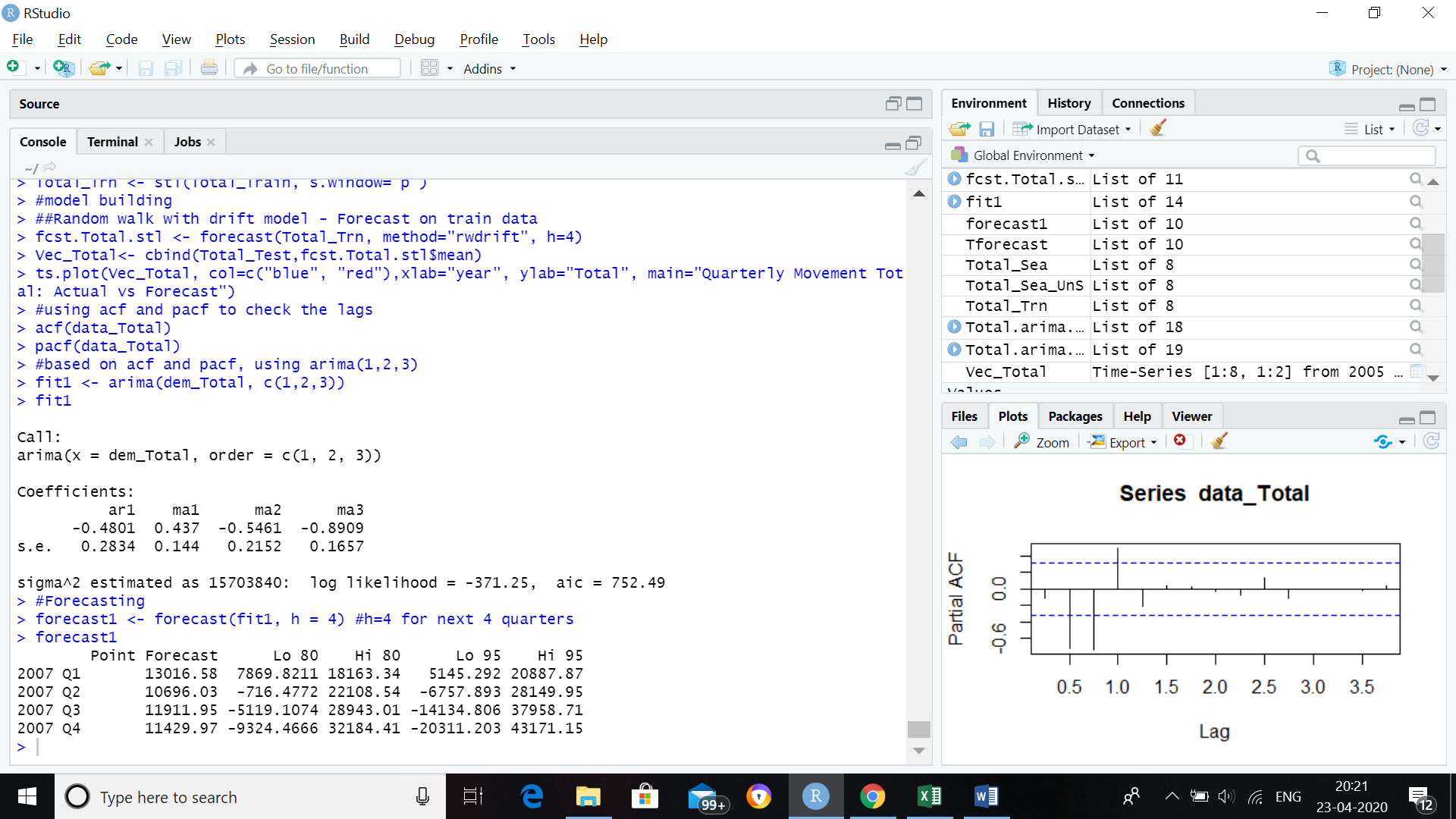
*fit1*



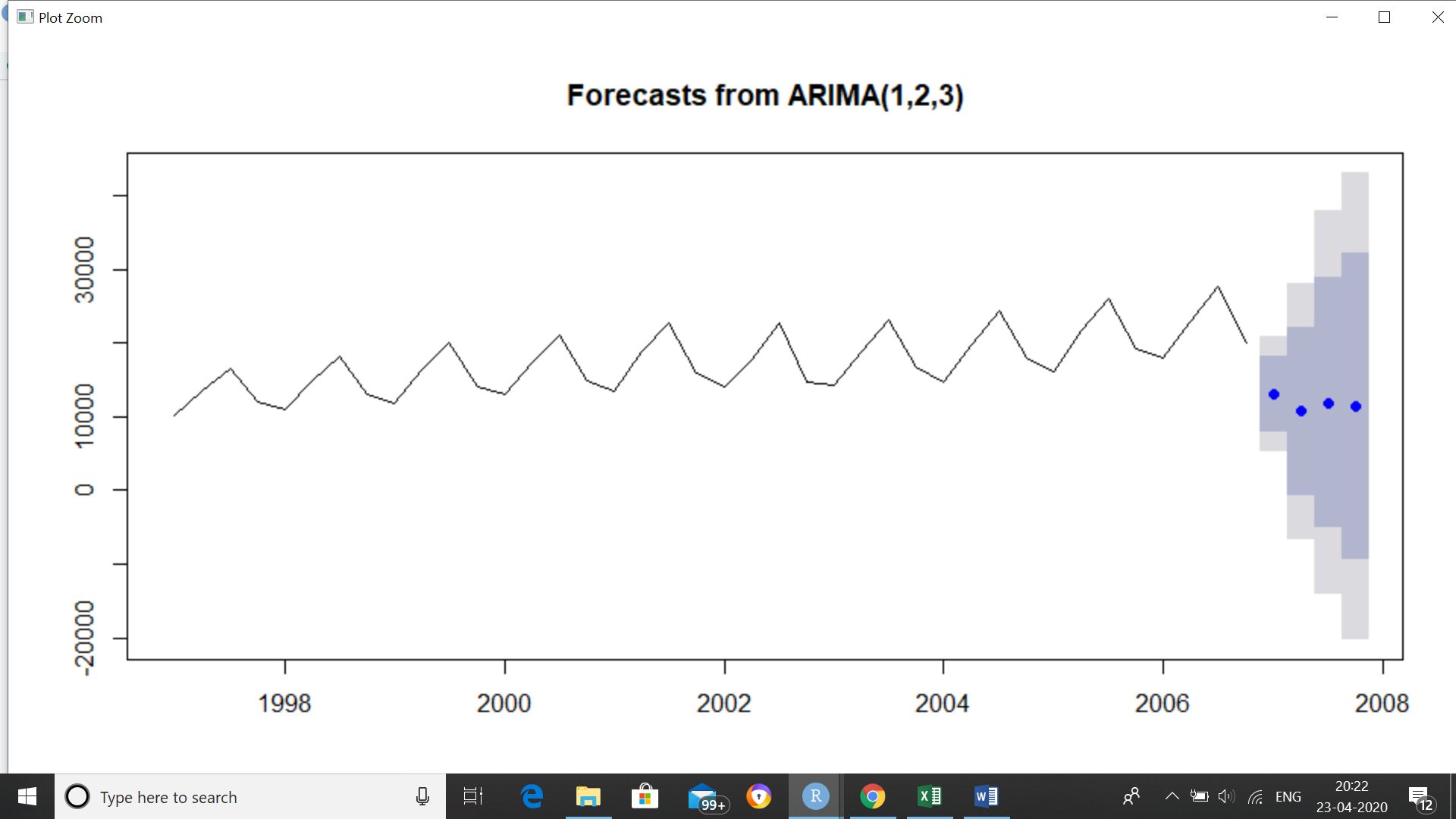
***#Forecasting***

*forecast1 <- forecast(fit1, h = 4) #h=4 for next 4 quarters*

*forecast1*



*plot(forecast1)*



***#the last Total point forecast will be 11429***

***#Mean absolute percentage error (MAPE)***

***## It calculates the mean absolute percentage error (Deviation) function for the forecast and the eventual outcomes.***

***#removing NA in Vec\_Total***

*tail(Vec\_Total)*

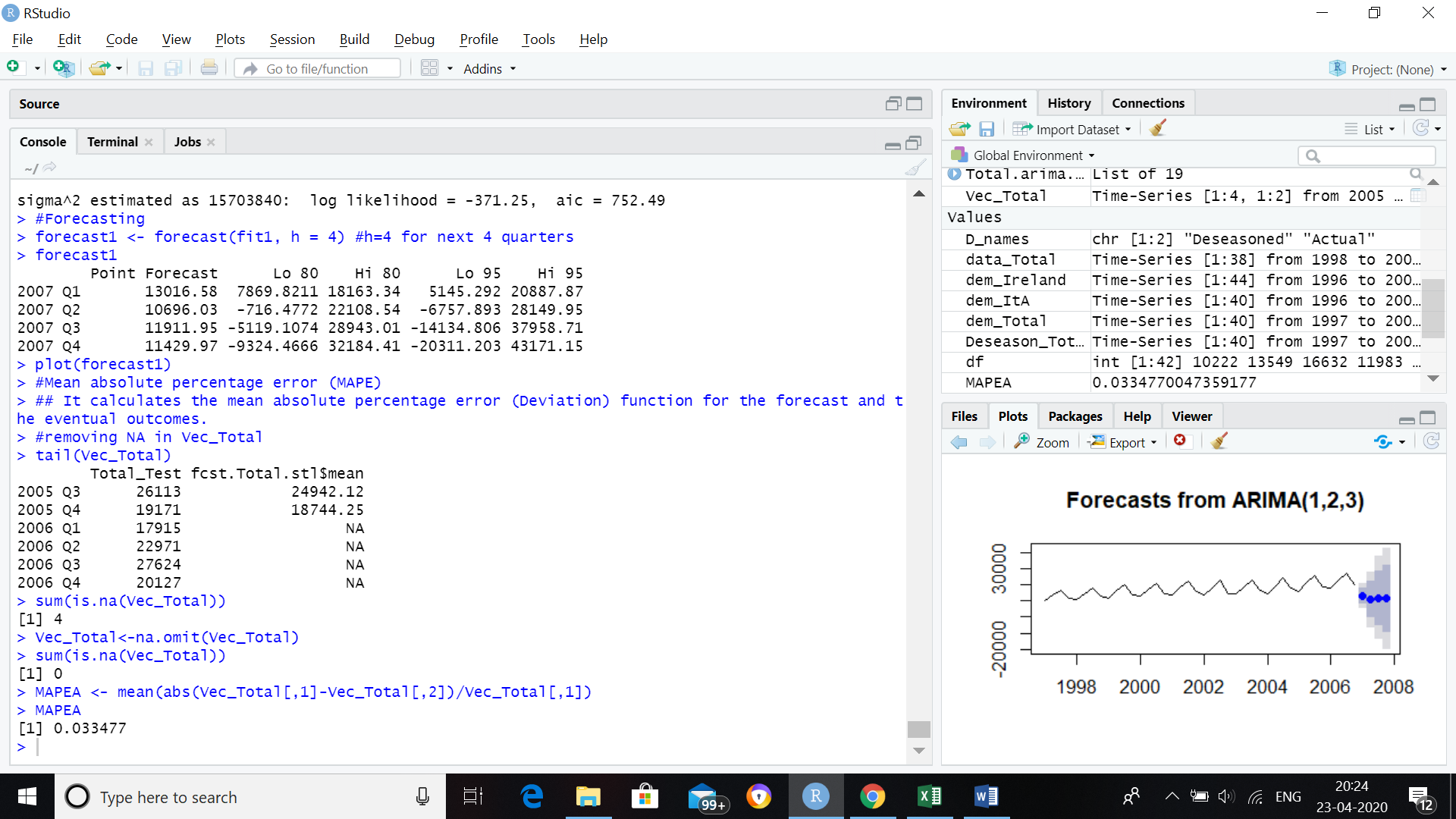
*sum(is.na(Vec\_Total))*

*Vec\_Total<-na.omit(Vec\_Total)*

*sum(is.na(Vec\_Total))*

*MAPEA <- mean(abs(Vec\_Total[,1]-Vec\_Total[,2])/Vec\_Total[,1])*

*MAPEA*



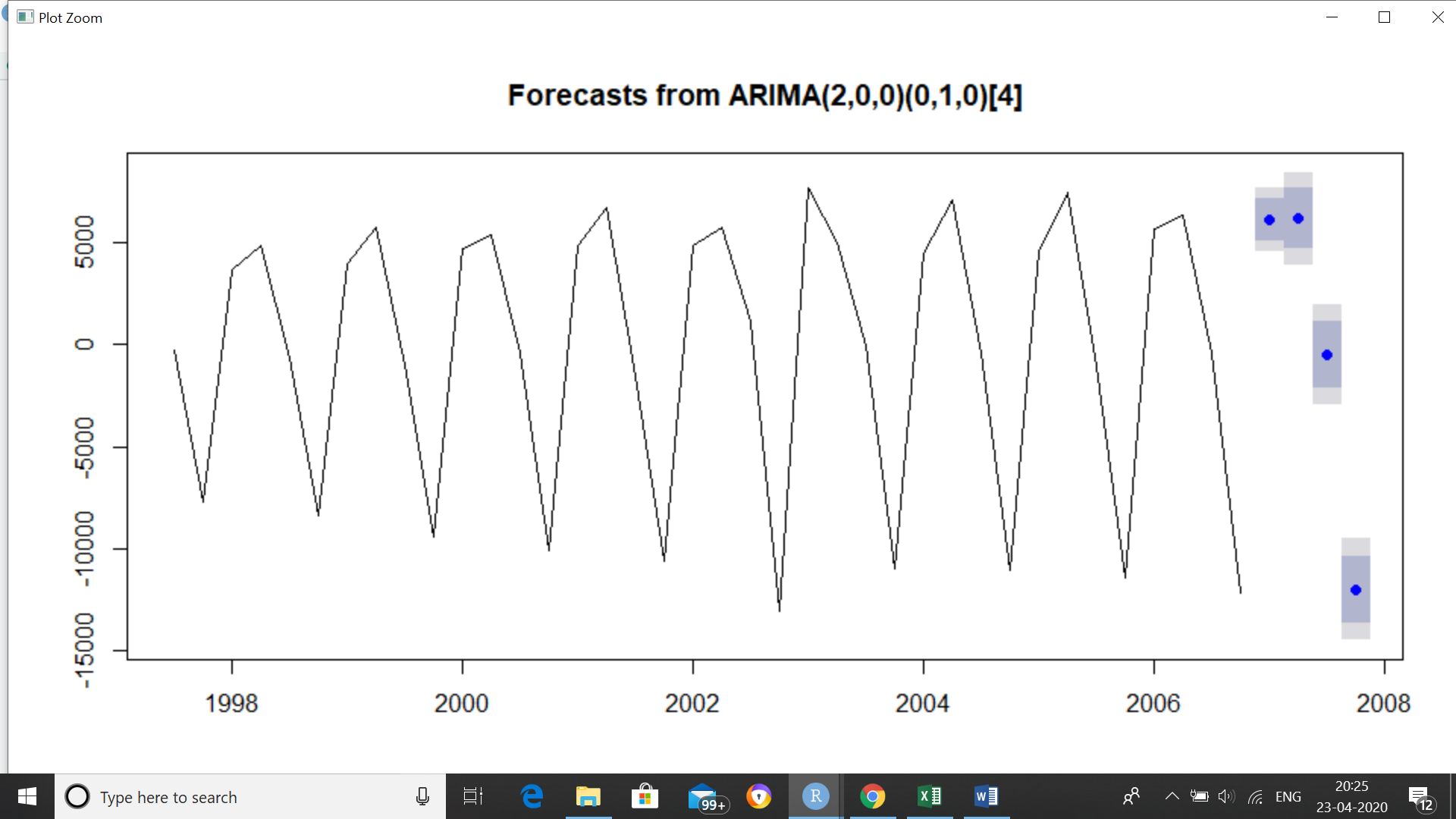
***#From the above MAPE results we can see the 3.3 % less accuracy in model.***

***#Forecasting using Arima model- another method***

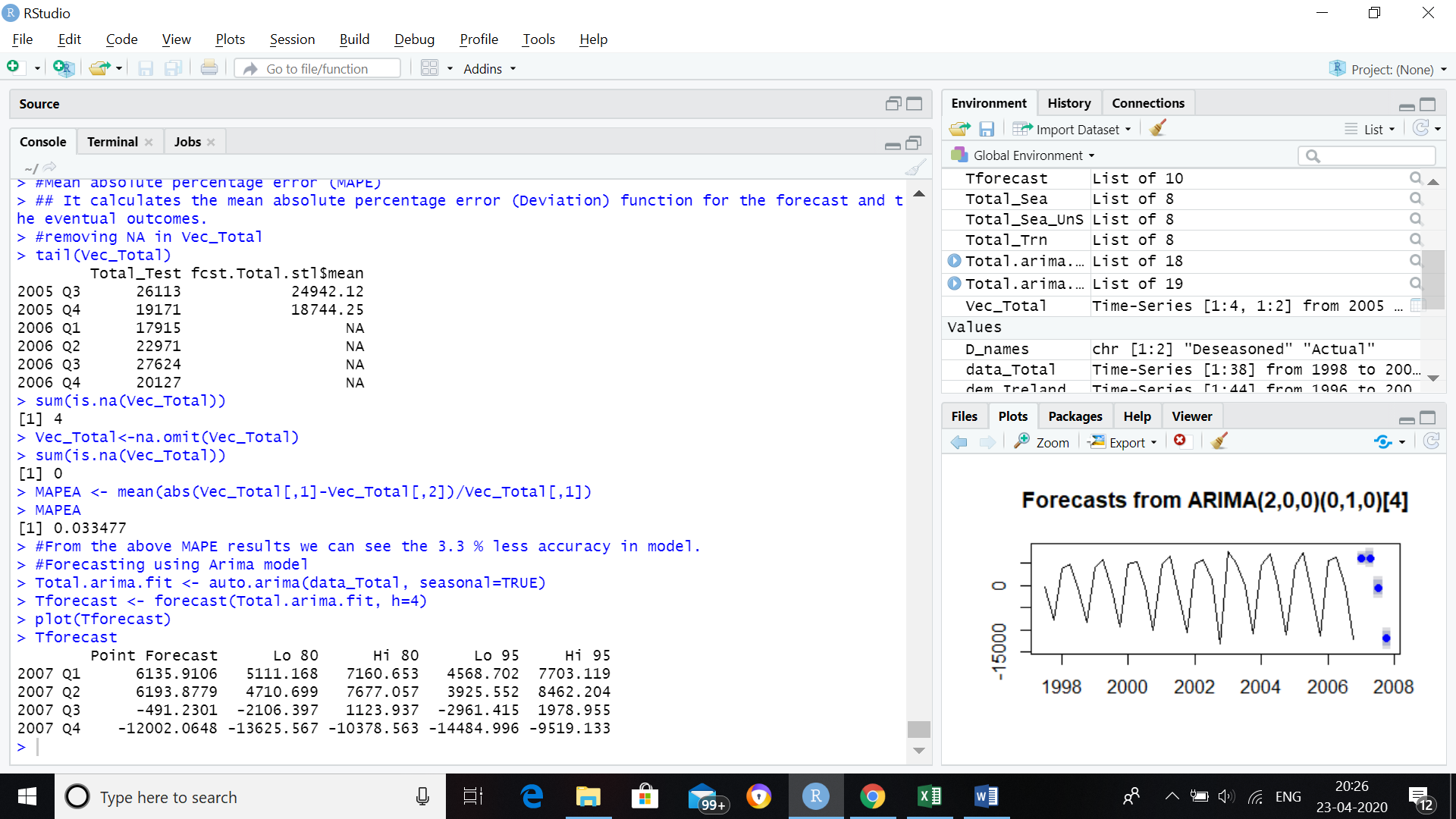
*Total.arima.fit <- auto.arima(data\_Total, seasonal=TRUE)*

*Tforecast <- forecast(Total.arima.fit, h=4)*

*plot(Tforecast)*



*Tforecast*



***#the point forecast will decrease in the later part of the quarter***