```
#Xp=c(25.0,NA,27.0,26.0,28.0,24.0,32.0,31.0,33.0,30.0,29.0,25.0,29.0)

Xp=c(25.0,24.0,27.0,26.0,28.0,24.0,32.0,31.0,33.0,30.0,29.0,25.0,29.0)
Yk=c(32.0,32.5,34.0,33.0,33.5,31.6,37.0,33.5,37.3,36.0,34.8,33.0,35.0)


multregData=read.table("MultRegData.txt",header=T)

head(multregData)

multregDataY=data.frame(multregData$Y)
colnames(multregDataY)=c("y")
multregDataX=data.frame(multregData[,3:6])
colnames(multregDataX)=c("X1","X2","X3","X4")

x=multregDataX
y=multregDataY

RegMatrix = function(x, y){
##########################################################################
# This function carries out Simple or Multiple Linear Regression Analysis
# by using the matrix calculations
# x and y variables are scaled by using scale function
# scale function is equal to z-Transformation
# No missing observation is allowed
#########################################################################
cat("Multple Linear Regression Analysis","\n")
cat("No missing observation is allowed","\n")
cat("\n")

x=as.matrix(data.frame(x))
#colnames(x)=c("x")
cat("Independent variable(s) (x)","\n")
print(x)
cat("\n")

y=as.matrix(data.frame(y))
colnames(y)=c("y")
cat("Dependent variable (y)","\n")
print(y)
cat("\n")

one=matrix(c(1), dim(x)[1], 1)
colnames(one)=c("One")
cat("Variable of One","\n")
print(one)
cat("\n")

## The vectors 1 and x are combined under X matix
 X=cbind(one, x)
 colnames(x)
 colnames(X)=c("One",colnames(x))
 cat("X matrix including one and x matrix","\n")
 print(X)
 cat("\n")

## get sum of y and sum of x*y
 cat("Xy matrix","\n")
 Xy=t(X)%*%y
 print(Xy)
 cat("\n")

## get number of observations, sum of x and sum of x*x
 cat("XX matrix","\n")
 XX=t(X)%*%X
 print(XX)
 cat("\n")

 print(cor(XX))

## get determinant of X'X
 cat("Determinant of XX","\n")
 detXX=det(t(X)%*%X)
 print(detXX)
 cat("\n")

## get inverse of X'X
 cat("Inverse of XX","\n")
 InvXX=solve(t(X)%*%X)
 print(InvXX)
 cat("\n")

## get Least Square Estimate of Beta (Beta0 and Beta1)
 cat("Least Square Estimates of Beta parameters","\n")
 Beta=solve(XX, Xy)
 colnames(Beta)=c("Beta")
 betanames=matrix("",1,dim(XX)[2])
 for(i in 1:dim(XX)[2]){
   betanames[1,i]=paste("Beta_",i-1,sep="")
 }
 rownames(Beta)=c(betanames)
 print(Beta)
 cat("\n")

## get the fitted value vector of yhat
 cat("yhat values","\n")
 yhat=X%*%Beta
 colnames(yhat)=c("yhat")
 print(yhat)
 cat("\n")

## get residuals
 cat("Estimates of Residuals, residualhat","\n")
 Residuals=y - yhat
 colnames(Residuals)=c("Residuals")
 print(Residuals)
 cat("\n")

## get Total Sum of Square (TSS)
 cat("TSS: Total Sum of Squares (Sum of Squares of y)","\n")
 TSS=t(y)%*%y - sum(y)^2/(length(y))
 colnames(TSS)=c("TSS")
 rownames(TSS)=c("")
 print(TSS)
 cat("\n")

## get Residual Sum of Square (RSS)
 cat("RSS: Residual Sum of Squares (Sum of Squares of e)","\n")
 RSS = t(Residuals)%*%Residuals
 colnames(RSS)=c("RSS")
 rownames(RSS)=c("")
 print(RSS)
 cat("\n")

## Regression Model Sum of Squares from yhat
 cat("Regression Model Sum of Squares from Yk_hat=Beta_0+Beta_1*Xp", "\n")
 RMSS=t(yhat)%*%yhat-sum(yhat)^2/length(yhat)
 colnames(RMSS)=c("RMSS")
 rownames(RMSS)=c("yhat")
 print(RMSS)
 cat("\n")

## Coefficient of Determination R2=RMSS/TSS
 cat("Coefficient of Determination R2=RMSS/TSS", "\n")
 R2=RMSS/TSS
 colnames(R2)=c("R2")
 rownames(R2)=c("")
 print(R2)
 cat("\n")
## or
## Coefficient of Determination R2=1-RSS/TSS
 cat("Coefficient of Determination R2=1-RSS/TSS", "\n")
 R2=1-RSS/TSS
 colnames(R2)=c("R2")
 rownames(R2)=c("")
 print(R2)
 cat("\n")
}
```