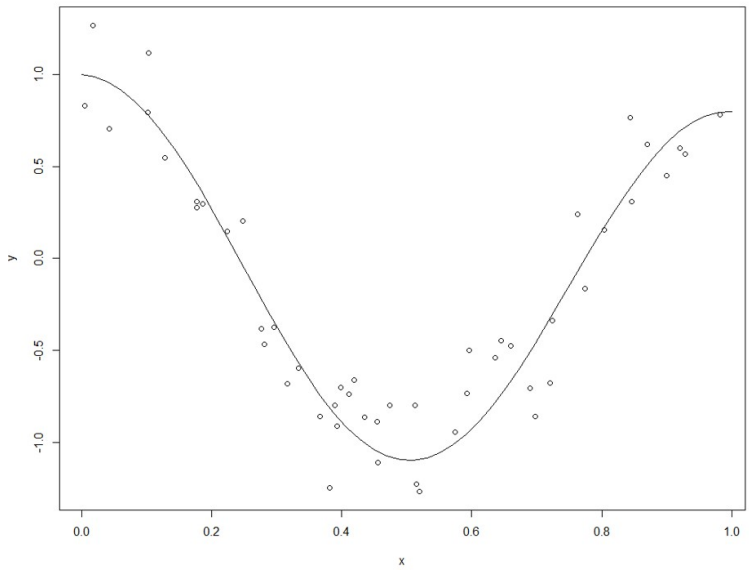

Figure 1: Solution of Question 1


```
rm(list=ls()) # clear all the variables in console
library(splines)
library(gam)
library(pracma)
```



```
################################################################################
#Question 1:
n<-50
set.seed(5) # sets the seed for random number generation making their regeneration possible
e<-rnorm(n,0,0.2)
x<-sort(runif(n,0,1))
a <- seq(0, 1, length=n)
y<-cos(2*pi*x)-0.2*x+e
b<-cos(2*pi*a)-0.2*a
plot(x,y)
lines(a,b)
```



```
################################################################################
#Question 2:
myknots <- quantile(x, probs = c(0.2, 0.4, 0.6, 0.8))
#ns generates a B-spline basis matrix for natural cubic splines, intercept is the first constant term
xns<- ns(x, knots = myknots, intercept = TRUE, Boundary.knots = range(c(0,1)))
#y.fit<- lm(y ~ -1 + xns) # command is used to fit linear models
y.fit <- xns%*%pinv(xns)%*%y
plot(x, y)
lines(a, b, col = "dodgerblue", lty = 1)
lines(x, y.fit, col = "forestgreen", lty = 2)

myknots <- quantile(x, probs = seq(0.05,0.95,length=8))
#ns generates a B-spline basis matrix for natural cubic splines, intercept is the first constant term
xns <- ns(x, knots = myknots, intercept = TRUE, Boundary.knots = range(c(0,1)))
#y.fit<- lm(y ~ -1 + xns) # command is used to fit linear models
y.fit <- xns%*%pinv(xns)%*%y
plot(x, y)
lines(a, b, col = "dodgerblue", lty = 1)
lines(x, y.fit, col = "forestgreen", lty = 2)
# at around about 8 knots, overfitting starts

################################################################################
```

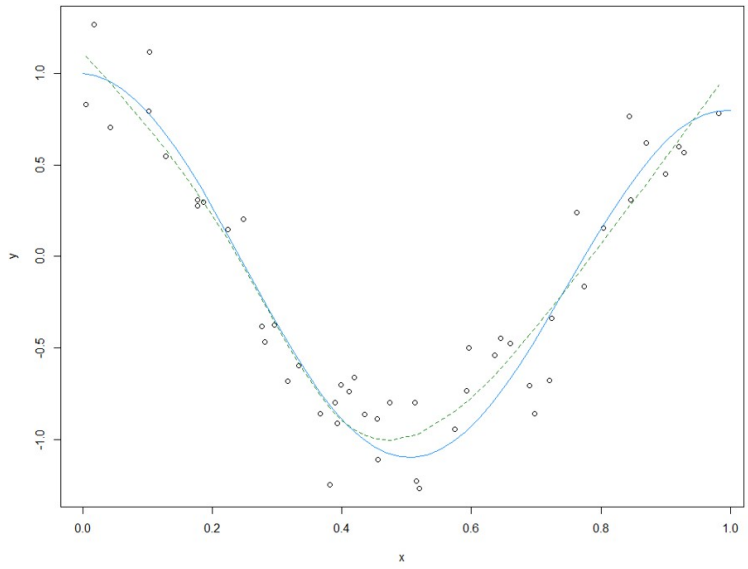

Figure 2: Solution of Question 2


```
#Question 3:
xss <- gam(y ~ s(x, df = 6))
yfit <- predict(xss)
plot(x, y)
lines(a, b, type = "l", col = "dodgerblue3", lty = 1)
lines(x, yfit, type = "l", col = "forestgreen", lty = 2)
```



```
##############################################################################
#Question 4:
results <- numeric(15)
for (i in 1:15) {
  xss <- gam(y ~ s(x, df = i))
  yfit <- predict(xss)
  results[i] <- sum((yfit - b)^2)/length(yfit)
}
plot(2:15, results[2:15], type = "b", col = "dodgerblue2", xlab = "DoF", ylab = "MSE", pch = 19, lwd = 3)
df = which.min(results)
# optimal number found to be at index 7 so df = 7 is optimal
```



```
##############################################################################
#Question 5:
data<-read.table("D:/R/data.txt") #Change the path according to your file location
x<-as.numeric(data[2:222,1])
y<-as.numeric(data[2:222,2])
xps <- smooth.spline(x,y,spar =0.9,all.knots = FALSE)
yfit <- predict(xps,x)$y
plot(x, y)
lines(x, yfit, type = "l", col = "dodgerblue3", lty = 2)
# we have to check this manually the overfitting starts at around about 0.5 and underfitting at 1
```

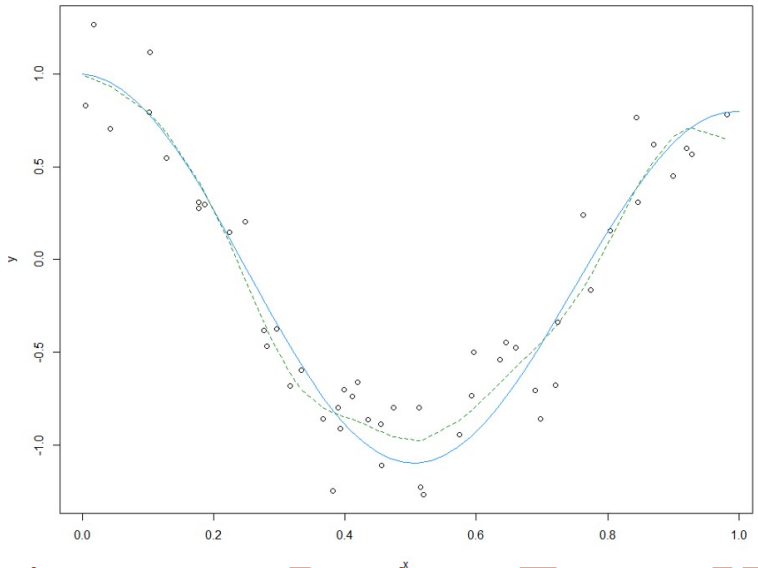

Figure 3: Solution of Question 2

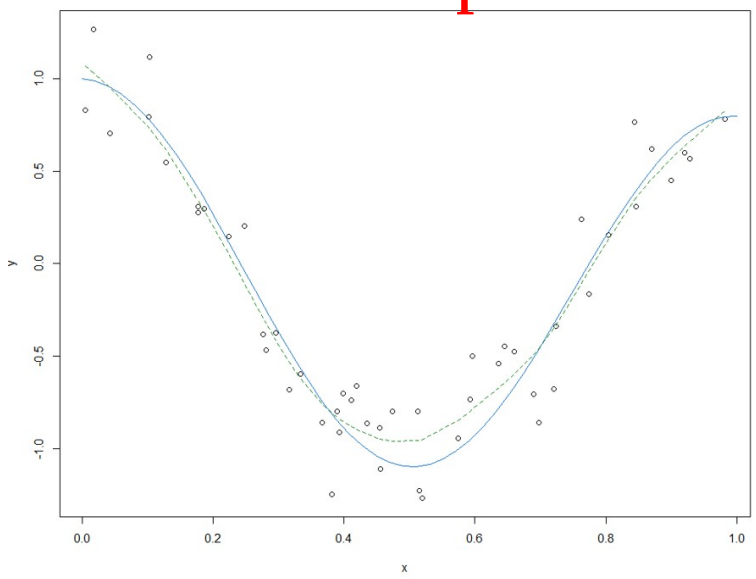

Figure 4: Solution of Question 3

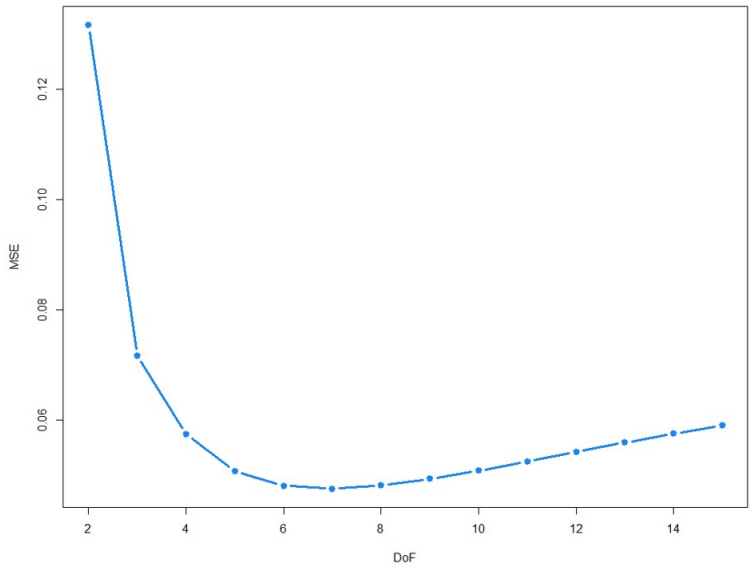

Figure 5: Solution of Question 4

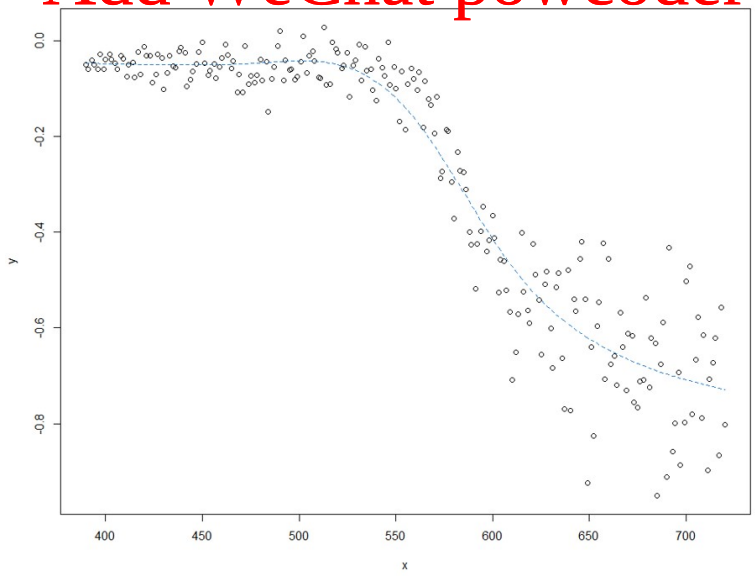

Figure 6: Solution of Question 5