Exercise 7

1.

For the regression models we'll choose: single linear regression model and multiple linear regression model.

```
library(boot)

computersData <- read.table('ComputerData.txt', header = T)
computersData <- computersData[, -c(1, 2, 10)]

set.seed(17)

cvs.error <- rep(0,10)

cvm.error <- rep(0,10)

degree = 1:10

# Cross validation k=10
# Evaluating performance of single regression model for polynomials of different degrees
for (d in degree) {
  computer_single.glm <- glm(PRP ~ poly(MMAX, d), data = computersData)
  cvs.error[d] <- cv.glm(computersData, computer_single.glm, K=10)$delta[1]
  cat("degree =", d, "cvs.error =", cvs.error[d], "\n")
}
```

```
## degree = 1 cvs.error = 6943.999
## degree = 2 cvs.error = 4655.769
## degree = 3 cvs.error = 5127.221
## degree = 4 cvs.error = 5125.833
## degree = 5 cvs.error = 4678.775
## degree = 6 cvs.error = 5791.531
## degree = 7 cvs.error = 5287.725
## degree = 8 cvs.error = 4980.649
## degree = 9 cvs.error = 5927.106
## degree = 10 cvs.error = 5356.779
```

```
# Cross validation k=10
# Evaluating performance of multiple regression model
for (d in degree) {
  computer_multiple.glm <-
    glm(PRP ~ poly(MYCT + MMIN + MMAX + CACH + CHMAX, d), data = computersData)
  cvm.error[d] <- cv.glm(computersData, computer_multiple.glm, K=10)$delta[1]
  cat("degree =", d, "cvm.error =", cvm.error[d], "\n")
}
```

```
## degree = 1 cvm.error = 5842.91
## degree = 2 cvm.error = 4421.637
## degree = 3 cvm.error = 4106.998
## degree = 4 cvm.error = 24569.65
## degree = 5 cvm.error = 105683.7
## degree = 6 cvm.error = 182612.6
```

```
## degree = 7 cvm.error = 3440.06
## degree = 8 cvm.error = 7453.816
## degree = 9 cvm.error = 101222076090
## degree = 10 cvm.error = 7218854663
```

```
computer_single.mod <- glm(PRP ~ poly(MMAX, 2), data=computersData)
training_single.MSE <- mean(computer_single.mod$residuals^2)

computer_multiple.mod <- glm(PRP ~ poly(MYCT + MMIN + MMAX + CACH + CHMAX, 7), data = computersData)
training_mult.MSE <- mean(computer_multiple.mod$residuals^2)

cat("Single regression: Test MSE = ", cvs.error[2], "| Training MSE = ",
    training_single.MSE, "\n")
```

```
## Single regression: Test MSE =  4655.769 | Training MSE =  4328.388
```

```
cat("Multiple regression: Test MSE  = ",cvm.error[7], "| Training MSE = ",
    training_mult.MSE, "\n")
```

```
## Multiple regression: Test MSE  =  3440.06 | Training MSE =  2358.308
```

2.

According to the **knn.reg** documentation(library FNN):

If test is not supplied, Leave one out cross-validation is performed and R-square is the predicted R-square.

```
library('FNN')

set.seed(1)

PRP <- computersData$PRP

#preprocessing
computers_n <- as.data.frame(scale(computersData[-7]))
computers_n <- cbind(computers_n, PRP)

#knn regression LOOCV because test = null
pc.knn <- knn.reg(train = computers_n[-7], test = NULL, y = PRP, k = 2)
```

3.

Now we can compare the MSE of the multiple linear regression model from 1) and k-NN regression from 2), but first we will recompute multiple linear regression for the normalized data, just to make sure that there is no difference caused by the normalization:

```
set.seed(1)

glm.mod <- glm(PRP ~ MYCT + MMIN + MMAX + CACH + CHMAX, data = computers_n)
cv.error <- cv.glm(computers_n, glm.mod, K=10)$delta[1]

cat("k-NN LOOCV MSE =", mean(pc.knn$residuals^2), "\n")
```

```
## k-NN LOOCV MSE = 4350.194
```

```r
cat("Mult. regression MSE =",cv.error)
```

```
## Mult. regression MSE = 4588.66
```

We can see that k-NN has a smaller mean square error(MSE) than multiple linear regression, the reason behind that is that in multiple linear regression we make an **assumption about the data**: *that they are linear and when it's not then it affects our performance*, we can show that if we use polynomial of degree 2 in the predictors:

```r
glm.mod <- glm(PRP ~ poly(MYCT + MMIN + MMAX + CACH + CHMAX, 2), data = computers_n)
cv.error <- cv.glm(computers_n, glm.mod, K=10)$delta[1]

cat("Mult. regression MSE(degree 2) =",cv.error)
```

```
## Mult. regression MSE(degree 2) = 3631.142
```

So, we see that we have much smaller test MSE with the quadratic polynomial which means that there is a degree of non-linearity on our data.

4.

Based on *test mean square error(MSE)* and also on the *test with quadratic terms* in the model which shows that *linear fit is not adequate for this data*; so **the best model in our case is k-NN**, the **advantage** of k-NN regression is: it *doesn't do any strong assumption about the relationship between X and Y* and then when the extent of non-linearity increases, there is little change in test set MSE for the non-parametric k-NN method, but there is a large increase in the test set MSE of linear regression, **drawbacks**: we dont't know the amount each predictor contributes in the model and *the curse of dimensionality*.

But we might prefer linear regression to k-NN from *an interpretabiliy standpoint*; we might loose something on accuracy but we would accept that for the sake of an easier interpretability of the model, also we can include higher polynomial terms in model(quadratic) and we saw that we get a much better fit(smaller MSE).