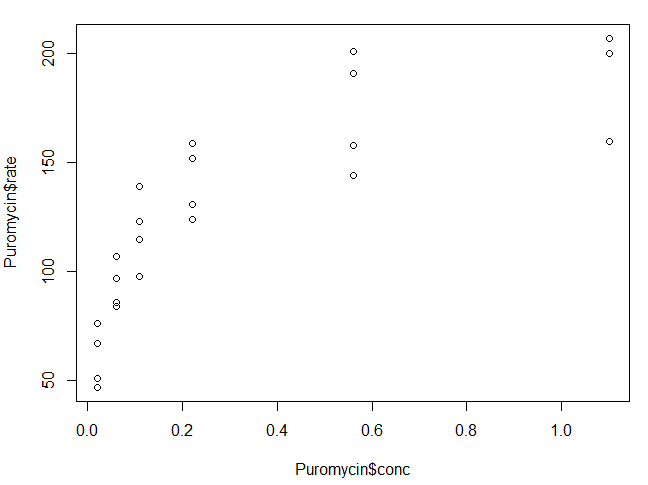
Arun Sangar

Hyunwook Park

CPSC 375

Homework 2

* 1. 
     1. 0.741216
     2. > m <- lm(rate~conc,data=Puromycin)

> residuals <- residuals(m)

> sse <- sum(residuals^2)

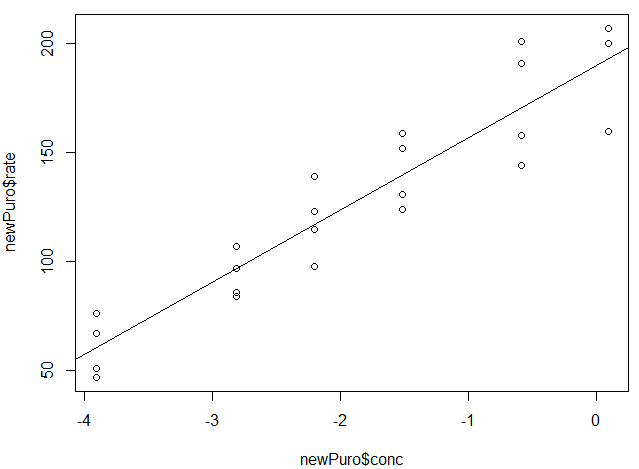
> mean <- mean(Puromycin$rate)

> residuals2 <- Puromycin$rate - mean

> ssr <- sum(residuals2^2)

> sst <- ssr + sse

> rsq <- ssr/sst

* + 1. 0.8888592
    2. Yes, a higher value means it is a more accurate model.
    3. 
    4. > newPuro <- Puromycin

> newPuro$conc <- log(Puromycin$conc)

> m <- lm(rate~conc,data=newPuro)

> residuals <- residuals(m)

> sse <- sum(residuals^2)

> mean <- mean(newPuro$rate)

> residuals2 <- newPuro$rate - mean

> ssr <- sum(residuals2^2)

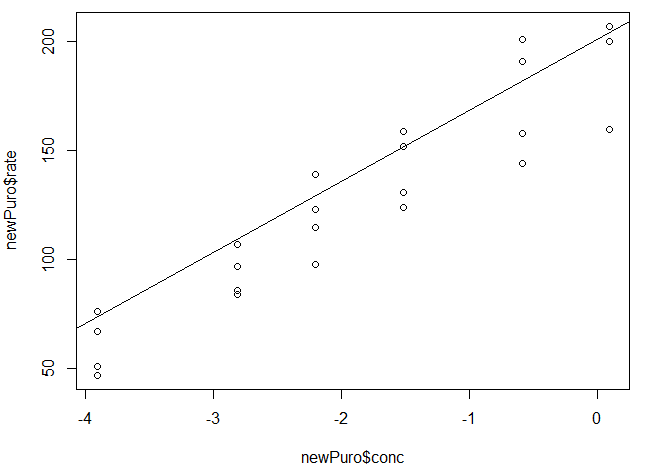
> sst <- ssr + sse

> rsq <- ssr/sst

> plot(newPuro$conc,newPuro$rate)

> coeffs <- coef(m)

> abline(coeffs[1], coeffs[2])

* + 1. 0.9504869
    2. Yes, again it is higher therefore better.
    3. 
    4. > newPuro <- Puromycin

> newPuro$conc <- log(Puromycin$conc)

> m <- lm(rate~conc+state,data=newPuro)

> residuals <- residuals(m)

> sse <- sum(residuals^2)

> mean <- mean(newPuro$rate)

> residuals2 <- newPuro$rate - mean

> ssr <- sum(residuals2^2)

> sst <- ssr + sse

> rsq <- ssr/sst

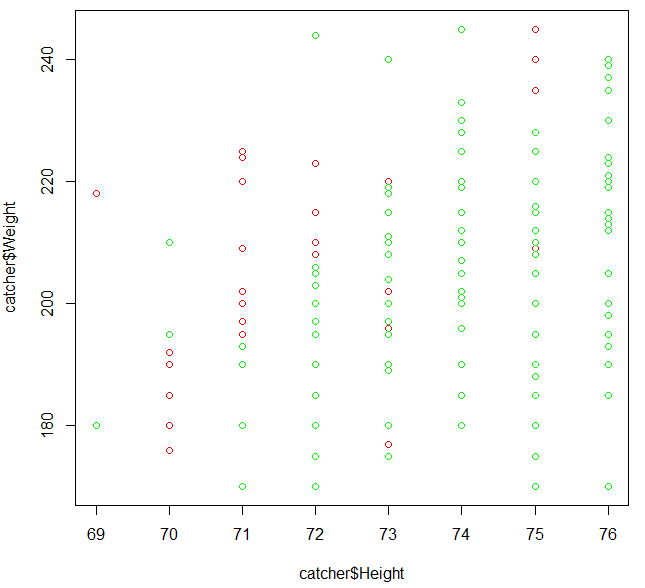
> plot(newPuro$conc,newPuro$rate)

> coeffs <- coef(m)

> abline(coeffs[1],coeffs[2])

* + 1. > mydata <- read.csv("C:\\Users\\aruns\\Desktop\\baseball.csv")
    2. > subdata <- subset(mydata, mydata$Position=="Catcher"|mydata$Position=="Starting\_Pitcher")
    3. > subdata <- droplevels(subdata)
    4. > subdata <- subdata[order(subdata$Name),]
    5. > nrow(subdata)

> subdata[296,]

* + 1. 
    2. No, linear regression would not give a good model because the height variable seems to be categorical. To add to this, the weight values vary greatly for each height value, using a linear model would not accurately represent that.
    3. > catcher <- subset(subdata,subdata$Position=="Catcher")

> pitcher <- subset(subdata,subdata$Position=="Starting\_Pitcher")

> plot(catcher$Height,catcher$Weight,col="Red")

> points(pitcher$Height,pitcher$Weight,col="Green")

* + 1. Error rate: 2/6 or 33.3%
    2. > trainindex <- 1:290

> testindex <- 291:296

> traindata <- subdata[trainindex,4:5]

> testdata <- subdata[testindex,4:5]

> trainlabels <- subdata[trainindex,3]

> testlabals <- subdata[testindex,3]

> predicted <- knn(test=testdata,train=traindata,cl=trainlabels,k=1)

> table(testlabels,predicted)

* + 1. Error rate: 0/6 or 0%
    2. > predicted <- knn(test=testdata,train=traindata,cl=trainlabels,k=3)

> table(testlabels,predicted)

* + 1. Error rate: 2/6 or 33.3%
    2. Considering these observations, when k = 3 there is a 0% error rate, which suggests that 3 would be the best value for k.
    3. > predicted <- knn(test=testdata,train=traindata,cl=trainlabels,k=25)

> table(testlabels,predicted)

* + 1. Error rate: 1/6 or 16.7%
    2. No, the error rate does not always decrease with larger number of parameters. This is because the additional parameters may not be related to the dependent variable.
    3. > trainindex <- 1:290

> testindex <- 291:296

> traindata <- subdata[trainindex,4:6]

> testdata <- subdata[testindex,4:6]

> trainlabels <- subdata[trainindex,3]

> testlabels <- subdata[testindex,3]

> predicted <- knn(test=testdata,train=traindata,cl=trainlabels,k=1)

> table(testlabels,predicted)