Peter Mamaev

MGMT-4600 Lab 2 Part 2

* Verifying that regress\_data is attached.

> head(regress\_data)

YEAR ROLL UNEM HGRAD INC

1 1 5501 8.1 9552 1923

2 2 5945 7.0 9680 1961

3 3 6629 7.3 9731 1979

4 4 7556 7.5 11666 2030

5 5 8716 7.0 14675 2112

6 6 9369 6.4 15265 2192

* Qqplot between 90K High school grads and 7% Unemployment rate.

> #Using the unemployment rate (UNEM) and

> #number of spring high school graduates

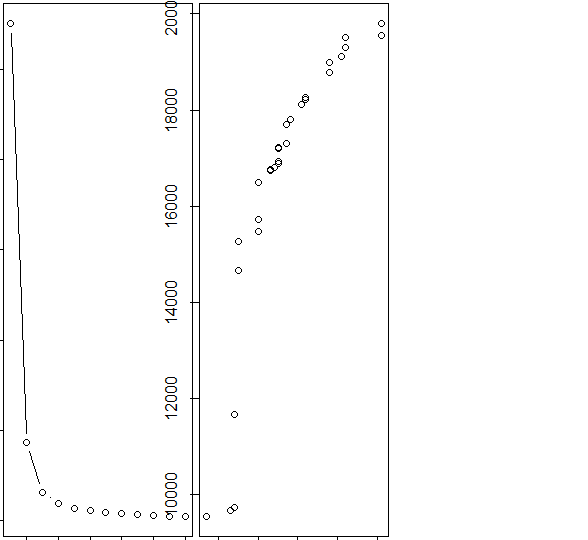
> #(HGRAD), predict the fall enrollment (ROLL)

> #for this year by knowing that UNEM=7% and

> #HGRAD=90,000.

> qqplot(regress\_data$UNEM, regress\_data$HGRAD, data = regress\_data, geom = "point")

There were 12 warnings (use warnings() to see them)



* Two-predictor model for UNEM and GRAD.

> twoPredictorModel <- lm(ROLL ~ UNEM + HGRAD, data = regress\_data)

> twoPredictorModel

Call:

lm(formula = ROLL ~ UNEM + HGRAD, data = regress\_data)

Coefficients:

(Intercept) UNEM HGRAD

-8255.7511 698.2681 0.9423

* Find Fall enrollement at 90K High School Grads, 7% Unemployment Rate, and the prior-mentioned two predictors.
* Predict ROLL if INC = $25,000.

> #The formula I found goes:

> #Fall Enrollment = Intercept + UNEM \* Unemployment Rate + HGRAD \* Number of Spring High School Graduates

> #with:

> #Intercept = -8255.7511

> #UNEM = 698.2681

> #HGRAD = 0.9423

> #Unemployment rate = 7%

> #Spring High School Grads = 90,000

> -8255.7511 + 698.2681 \* 9 + 0.9423 \* 90000 #= 82835.66

[1] 82835.66

> #Repeat and add per capita income (INC) to

> #the model. Predict ROLL if INC=$25,000

> threePredictorModel <- lm(ROLL ~ UNEM + HGRAD + INC, data = regress\_data)

> #Repeat and add per capita income (INC) to

> #the model. Predict ROLL if INC=$25,000

> threePredictorModel <- lm(ROLL ~ UNEM + HGRAD + INC, data = regress\_data)

> threePredictorModel

Call:

lm(formula = ROLL ~ UNEM + HGRAD + INC, data = regress\_data)

Coefficients:

(Intercept) UNEM HGRAD INC

-9153.2545 450.1245 0.4065 4.2749

> #Intercept = -9153.2545

> #UNEM = 450.1245

> #HGRAD = 0.4065

> #INC (third variable) = 4.2749

> #Unemployment rate = 7%

> #Spring High School Grads = 90,000

> #Fall Enrollment = Intercept + UNEM \* Unemployment Rate + UGRAD \* Number of Spring High School Graduates + 4.3 \* Per Capita Income

> -9153.2545 + 450.1245 \* 9 + 0.4065 \* 90000 + 4.2749 \* 25000 #= 138355.4

[1] 138355.4

* Getting a summary of abalone rings.

> # As shown above, the “rings” variable has a range between 1-29.

> # This is the variable that we want to predict, and predicting this many levels

> # might not give us the insight we’re looking for.

> # For now, we’ll break the rings variable

> # into 3 levels" “young” for abalones less than 8, “adult” for abalones between 8-11, and “old” for abalones older than 11.

> # So from my understanding we're basically stratifying the data, after converting it into a numeric format.

> abalone$Rings <- as.numeric(abalone$Rings)

> abalone$Rings <- cut(abalone$Rings, br=c(-1,8,11,35), labels = c("young", 'adult', 'old'))

> abalone$Rings <- as.factor(abalone$Rings)

> summary(abalone$Rings)

young adult old

1407 1810 960

> str(abalone)

'data.frame': 4177 obs. of 9 variables:

$ Sex : chr "M" "M" "F" "M" ...

$ Length : num 0.455 0.35 0.53 0.44 0.33 0.425 0.53 0.545 0.475 0.55 ...

$ Diameter : num 0.365 0.265 0.42 0.365 0.255 0.3 0.415 0.425 0.37 0.44 ...

$ Height : num 0.095 0.09 0.135 0.125 0.08 0.095 0.15 0.125 0.125 0.15 ...

$ Whole.weight : num 0.514 0.226 0.677 0.516 0.205 ...

$ Shucked.weight: num 0.2245 0.0995 0.2565 0.2155 0.0895 ...

$ Viscera.weight: num 0.101 0.0485 0.1415 0.114 0.0395 ...

$ Shell.weight : num 0.15 0.07 0.21 0.155 0.055 0.12 0.33 0.26 0.165 0.32 ...

$ Rings : Factor w/ 3 levels "young","adult",..: 3 1 2 2 1 1 3 3 2 3 ...

> aba <- abalone

> aba$Sex <- NULL

> # normalize the data using min max normalization

> # Normalization techniques enables us to reduce the scale of the variables and thus it affects the statistical distribution of the data in a positive manner.

> normalize <- function(x) {

+ return ((x - min(x)) / (max(x) - min(x)))

+ }

> aba[1:7] <- as.data.frame(lapply(aba[1:7], normalize))

> summary(aba$shucked\_wieght)

Length Class Mode

0 NULL NULL

* K-Nearest Neighbor Prediction for abalone rings.

> # k-nearest neighbour classification for test set from training set.

> # For each row of the test set, the k nearest (in Euclidean distance) training set

> # vectors are found, and the classification is decided by majority vote,

> # with ties broken at random. If there are ties for the kth nearest

> # vector, all candidates are included in the vote.

> KNNpred <- knn(train = KNNtrain[1:7], test = KNNtest[1:7], cl = KNNtrain$Rings, k = 55)

> KNNpred

[1] young young adult young young young adult adult adult old

[11] young young young young young young adult young young young

[21] old old adult young adult adult adult adult old adult

[31] adult adult adult old young old young old young young

[41] adult old adult young young adult old old old old

[51] old old adult adult young adult adult young adult young

[61] old adult young adult adult adult young old young young

[71] young young young old old adult adult old young young

[81] young adult old old old adult adult adult adult young

[91] adult adult young adult young young young young young adult

[101] adult adult adult adult adult old young adult adult adult

[111] adult adult adult adult adult old adult old adult adult

[121] adult young adult old adult young adult adult adult adult

[131] adult adult young adult old adult young adult young young

[141] old adult adult adult adult adult adult adult adult adult

[151] old adult old young young young young adult young adult

[161] young young young young adult adult young adult adult young

[171] adult old old young adult adult old adult adult old

[181] young young young adult young young old young young young

[191] young old adult adult adult adult old adult young old

[201] old adult old young adult young young young young young

[211] old adult old old old adult adult old old adult

[221] adult old adult old old old old young adult young

[231] adult young young young young young young young young young

[241] young young adult young adult adult adult adult adult adult

[251] adult adult adult adult adult adult adult old adult young

[261] young young young young young young young young young young

[271] young young young young young young young young young adult

[281] young young young young young young adult adult adult adult

[291] adult adult adult adult adult adult adult adult adult adult

[301] adult adult adult adult adult adult adult adult young young

[311] young young young young young young young young adult young

[321] adult adult adult adult adult adult adult adult adult adult

[331] adult adult adult adult adult adult adult adult adult adult

[341] adult adult adult adult adult adult adult adult adult young

[351] young young young young young young young young young young

[361] young young young young young young young adult young adult

[371] young adult adult young adult adult adult adult adult adult

[381] adult adult adult adult adult adult adult adult adult adult

[391] adult adult adult adult adult adult adult adult old adult

[401] adult adult adult adult adult adult adult adult young young

[411] young young young young young young young adult adult adult

[421] adult adult adult adult old adult adult adult adult adult

[431] old adult adult adult adult adult adult adult adult young

[441] young young young young young young young young young young

[451] young young young young young young young young adult young

[461] young young young young adult adult young old adult adult

[471] adult adult adult adult adult adult adult adult adult adult

[481] adult adult old adult adult adult adult adult adult adult

[491] adult adult adult adult adult adult adult adult adult adult

[501] adult adult adult adult adult adult adult adult adult adult

[511] adult adult adult adult adult adult adult old adult adult

[521] adult adult adult adult adult adult adult adult young young

[531] young adult adult adult adult adult adult adult adult adult

[541] adult adult old adult young young young young young young

[551] adult adult young young young young adult adult adult adult

[561] old adult adult adult adult adult adult adult adult adult

[571] adult adult adult old adult adult adult adult adult old

[581] adult adult adult adult adult adult old adult old young

[591] young young young young young young adult adult adult adult

[601] adult adult young young young young young young young young

[611] young young young adult adult adult adult adult adult adult

[621] old adult young young young young old young young young

[631] young young young adult young young young adult adult adult

[641] young adult old young young adult young old old adult

[651] adult adult adult adult old old old old old young

[661] old old adult adult young adult adult adult young adult

[671] old adult young old adult old adult adult adult old

[681] old adult adult adult young adult adult young adult adult

[691] young adult adult adult young old adult old adult adult

[701] adult adult adult adult old adult old young young young

[711] young adult young young adult adult adult old adult adult

[721] young young young adult adult old adult young old old

[731] young young old old adult old adult adult adult adult

[741] old old old young young young young young young adult

[751] young adult adult adult adult adult adult adult adult adult

[761] adult adult adult adult young young young young young young

[771] young young adult adult adult adult adult adult adult adult

[781] adult adult adult adult adult adult adult adult adult young

[791] young young young adult adult adult young adult adult adult

[801] adult adult adult adult adult adult adult young young young

[811] young young young young young young adult young adult young

[821] adult adult adult adult adult adult adult adult adult adult

[831] adult adult old adult adult old adult young young young

[841] young young young young adult adult adult adult adult adult

[851] adult adult adult adult adult adult young young young young

[861] adult adult adult adult adult adult adult adult adult adult

[871] adult adult adult adult adult adult adult adult adult adult

[881] adult adult adult adult adult adult adult old young young

[891] young young adult adult adult adult old adult adult adult

[901] adult adult adult old adult adult adult adult old adult

[911] old young young young young adult young adult adult young

[921] young young adult adult adult adult adult adult adult adult

[931] old young adult adult adult young adult adult young young

[941] young young adult old adult old adult young young young

[951] old young adult young adult old old young young young

[961] adult old adult old old adult adult young young young

[971] young young adult adult adult young adult adult old adult

[981] old adult adult adult adult adult adult old young adult

[991] young young adult young young adult adult adult young young

[ reached getOption("max.print") -- omitted 259 entries ]

Levels: young adult old

* Importing iris dataset.

> data("iris")

> library(ggplot2) # we will use ggplot2 to visualize the data.

> head(iris) # first 6 rows of the iris dataset

Sepal.Length Sepal.Width Petal.Length Petal.Width Species

1 5.1 3.5 1.4 0.2 setosa

2 4.9 3.0 1.4 0.2 setosa

3 4.7 3.2 1.3 0.2 setosa

4 4.6 3.1 1.5 0.2 setosa

5 5.0 3.6 1.4 0.2 setosa

6 5.4 3.9 1.7 0.4 setosa

> str(iris) # structure of the iris data using str() function in R.

'data.frame': 150 obs. of 5 variables:

$ Sepal.Length: num 5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...

$ Sepal.Width : num 3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...

$ Petal.Length: num 1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...

$ Petal.Width : num 0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...

$ Species : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1 1 1 ...

* Removing the fifth non-numeric column.

> #Create a new data frame and remove the fifth

> iris2 <- data.frame(iris)

> # Removing the non-numerical Species column.

> iris2$Species = NULL

> head(iris2)

Sepal.Length Sepal.Width Petal.Length Petal.Width

1 5.1 3.5 1.4 0.2

2 4.9 3.0 1.4 0.2

3 4.7 3.2 1.3 0.2

4 4.6 3.1 1.5 0.2

5 5.0 3.6 1.4 0.2

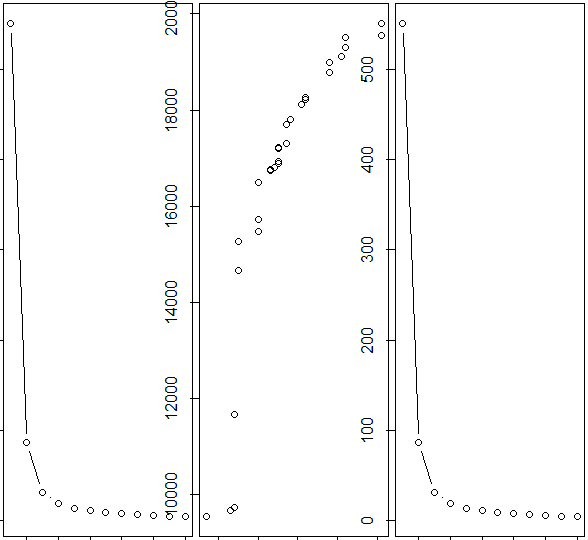
6 5.4 3.9 1.7 0.4

> wss<- sapply(1:k.max,function(k){kmeans(iris2[,3:4],k,nstart = 20,iter.max = 20)$tot.withinss})

> wss # within sum of squares.

[1] 550.895333 86.390220 31.371359 19.465989 13.916909 11.025145

[7] 9.236596 7.615402 6.456495 5.528149 5.115650 4.691349



> icluster <- kmeans(iris2[,3:4],3, nstart = 20)

> icluster

K-means clustering with 3 clusters of sizes 52, 48, 50

Cluster means:

Petal.Length Petal.Width

1 4.269231 1.342308

2 5.595833 2.037500

3 1.462000 0.246000

Clustering vector:

[1] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3

[34] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

[67] 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

[100] 1 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 1 2 2 2 2 2

[133] 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2

Within cluster sum of squares by cluster:

[1] 13.05769 16.29167 2.02200

(between\_SS / total\_SS = 94.3 %)

Available components:

[1] "cluster" "centers" "totss" "withinss"

[5] "tot.withinss" "betweenss" "size" "iter"

[9] "ifault"

> table(icluster$cluster,iris$Species)

setosa versicolor virginica

1 0 48 4

2 0 2 46

3 50 0 0