IS467: Assignment 2

Jasmine Dumas

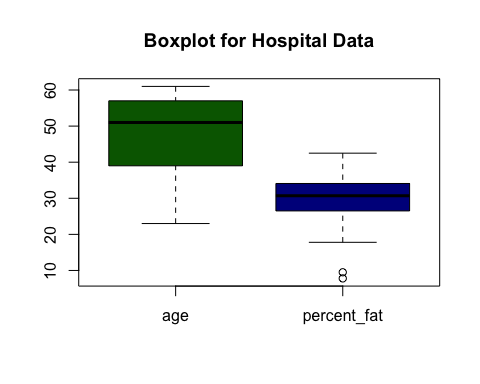
January 28, 2016

### Problem 1 (15 points): This problem is an example of data preprocessing needed in a data mining process.

# create the data frame - manually from HW  
hospital <- data.frame(x = c(23, 23, 27, 27, 39, 41, 47, 49, 50,  
 52, 54, 54, 56, 57, 58, 58, 60, 61),   
 y = c(9.5, 26.5, 7.8, 17.8, 31.4, 25.9,   
 27.4, 27.2,31.2, 34.6, 42.5, 28.8,   
 33.4, 30.2, 34.1, 32.9, 41.2, 35.7))  
# add column names  
colnames(hospital) <- c("age", "percent\_fat")

1. (3 points) Draw the box-plots for age and %fat. Interpret the distribution of the data: **The distribution for age is greater than for %fat meaning the standard deviation for age is greater at 13.2186242 versus 9.2543948 also the boxplot for age is larger in size and has a longer tail between the min-value and 25th percentile that the percent\_fat.**

boxplot(hospital, main = "Boxplot for Hospital Data", col=c("darkgreen", "darkblue"))



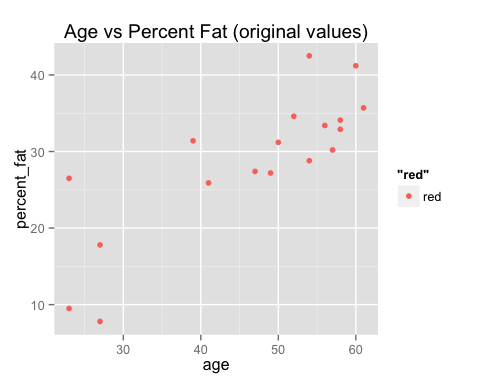
1. (3 points) Normalize the two attributes based on z-score normalization. (textbook 114-115): ***The function below displays the approach of supplying each observation (or row) and substracting the mean of the entire column and dividing by the column standard deviation.***

# function to code data from x to u (z score)  
# original source: source('~/Desktop/R-directory/coded\_x\_u.R', echo=TRUE)  
coded\_x\_2u <- function(x){  
 (x - mean(x)) / sd(x)  
}  
  
z\_age <- coded\_x\_2u(hospital$age)  
hospital$z\_age <- z\_age  
z\_fat <- coded\_x\_2u(hospital$percent\_fat)  
hospital$z\_fat <- z\_fat  
hospital

## age percent\_fat z\_age z\_fat  
## 1 23 9.5 -1.77359187 -2.083694688  
## 2 23 26.5 -1.77359187 -0.246729622  
## 3 27 7.8 -1.47098852 -2.267391195  
## 4 27 17.8 -1.47098852 -1.186823509  
## 5 39 31.4 -0.56317846 0.282748545  
## 6 41 25.9 -0.41187679 -0.311563683  
## 7 47 27.4 0.04202824 -0.149478530  
## 8 49 27.2 0.19332992 -0.171089884  
## 9 50 31.2 0.26898076 0.261137191  
## 10 52 34.6 0.42028243 0.628530204  
## 11 54 42.5 0.57158411 1.482178676  
## 12 54 28.8 0.57158411 0.001800946  
## 13 56 33.4 0.72288579 0.498862082  
## 14 57 30.2 0.79853662 0.153080422  
## 15 58 34.1 0.87418746 0.574501820  
## 16 58 32.9 0.87418746 0.444833697  
## 17 60 41.2 1.02548914 1.341704877  
## 18 61 35.7 1.10113998 0.747392650

1. (3 points) Regardless of the original ranges of the variables, normalization techniques transform the data into new ranges that allow to compare and use variables on the same scales. What are the values ranges of the following normalization methods? Explain your answer.
2. *Min-max normalization:* **The range is [, ] or commonly [0.0, 1.0] or [ -1.0, 1.0 ] which preserves the relationships among the original data values.**
3. *Z-score normalization:* **The range is [-infinity, infinity] since the actual minimum and maximum of attribute A are unknown, or the outliers dominate the min-max normalization.**
4. *Normalization by decimal scaling:* **The range is [absolute value of the min / 1000 , absolute value of the max / 1000] which moves the decimal point of values of attribute A.**
5. (3 points) Draw a scatter-plot based on the two variables and interpret the relationship between the two variables.: ***There seems to be a positive linear relationship between age and percent fat variables or another phrasing of the relationship is that the older person is in age the more body fat percentage.***

library(ggplot2)  
# original data  
a = ggplot(hospital, aes(x = age, y = percent\_fat, color = "red")) +  
 geom\_point() +  
 ggtitle("Age vs Percent Fat (original values)")  
a



1. (3 points) Calculate the correlation coefficient. Are these two attributes positively or negatively correlated? Compute the covariance matrix.: **Yes, These attributes have a strong/high positive correlation coefficient in which the max is 1.0. The cov() function is calculated by first creating a matrix of means, then create a difference matrix by subtracting the matrix of means from the original data matrix, and finally create the covariance matrix by multiplying the transposed the difference matrix with a normal difference matrix and inverse of the number of obserations -1.**

# correlation coefficient  
cor(hospital$age, hospital$percent\_fat)

## [1] 0.8176188

# the covariance matrix  
hospital$z\_age <- NULL # remove z-score for covariance matrix calculation  
hospital$z\_fat <- NULL  
cov(hospital)

## age percent\_fat  
## age 174.7320 100.01961  
## percent\_fat 100.0196 85.64382

### Problem 2 (10 points): This problem is an example of data preprocessing needed in a data mining process.

Suppose a group of 12 sales price records has been sorted as follows: 5, 10, 11, 13, 15, 35, 50,55,72,92,204,215

Partition them into bins by each of the following method, smooth the data and interpret the results: a. (5 points) equal-depth (frequency) partitioning with 3 values per bin. ***The partitioning is done by specifying the amount of bins and then equally dividing the list. The smoothing functions display the approach of calculating the means and medians of the previous and applying thats to each bin list.***

price <- c(5, 10, 11, 13, 15, 35, 50,55,72,92,204,215) # 12 values, 4 bins of 3 each  
bin\_equal\_depth = split(price, ceiling(seq\_along(price)/3))  
bin\_equal\_depth

## $`1`  
## [1] 5 10 11  
##   
## $`2`  
## [1] 13 15 35  
##   
## $`3`  
## [1] 50 55 72  
##   
## $`4`  
## [1] 92 204 215

## ---- Smoothing function - mean ----- ##  
bin\_smooth\_mean <- function(x, bins) {  
 lapply(x, function(x){ rep(mean(x), times = bins) })  
}  
## ---- Smoothing function - median ----- ##  
bin\_smooth\_median <- function(x, bins) {  
 lapply(x, function(x){ rep(median(x), times = bins) })  
}  
  
bin\_smooth\_mean(bin\_equal\_depth, bins = 3)

## $`1`  
## [1] 8.666667 8.666667 8.666667  
##   
## $`2`  
## [1] 21 21 21  
##   
## $`3`  
## [1] 59 59 59  
##   
## $`4`  
## [1] 170.3333 170.3333 170.3333

bin\_smooth\_median(bin\_equal\_depth, bins = 3)

## $`1`  
## [1] 10 10 10  
##   
## $`2`  
## [1] 15 15 15  
##   
## $`3`  
## [1] 55 55 55  
##   
## $`4`  
## [1] 204 204 204

1. (5 points) equal-width (distance) partitioning with 3 bins

width = ((max(price) - min(price)) / 3)  
width # each interval/bin has a width of approx. 70

## [1] 70

bin\_equal\_width = split(x=price, cut(price, 3, labels=c("1", "2", "3")))  
bin\_equal\_width

## $`1`  
## [1] 5 10 11 13 15 35 50 55 72  
##   
## $`2`  
## [1] 92  
##   
## $`3`  
## [1] 204 215

# Smoothing  
bin\_smooth\_mean(bin\_equal\_width, bins = 3)

## $`1`  
## [1] 29.55556 29.55556 29.55556  
##   
## $`2`  
## [1] 92 92 92  
##   
## $`3`  
## [1] 209.5 209.5 209.5

bin\_smooth\_median(bin\_equal\_width, bins = 3)

## $`1`  
## [1] 15 15 15  
##   
## $`2`  
## [1] 92 92 92  
##   
## $`3`  
## [1] 209.5 209.5 209.5

***These approaches to binning and smoothing allow the removal of noise (random error or variance in a measured variable) from the data. Binning methods help smooth a data value by associated it with its "neighboorhood" or the values surrounding it.***

### Problem 3 (10 points):

1. (2 points) Figure 1 illustrates the plots for some data with respect to two variables: balance and employment status. If you have to select one of these two variables to classify the data into two classes (circle class and plus class), which one would you select? Is there any approach/criterion that you can use to support your selection? Explain your answer.:

***I would use the unemployed / employed variable to classify the data back into either circle class or plus class because there seems to be a distictive grouping/cluster. I would utilize the clustering technique which partitions the objects into groups, or clusters so that objects within a cluster are "similar" (mean centroids) to one another and "disimilar" to objects in other clusters.***

1. (8 points) For the data in Figure 2 with three variables and two classes: which variable you would choose to classify the data? Show all the steps of your calculations and interpret your answer. ***I would use the variable 'C' since it clearly partitions the data frame into two clusters for factor "I" and "II"***

df <- data.frame(x = c(1, 1, 0, 1),   
 y = c(1, 1, 0, 0),  
 z = c(1, 0, 1, 0),   
 c = c("I", "I", "II", "II"))  
bb <- ggplot(df) +  
 geom\_point(aes(x=factor(c), y=x)) +   
 geom\_point(aes(x=factor(c), y=y)) +   
 geom\_point(aes(x=factor(c), y=z))   
bb

