**Problem 1 (10 points). This problem is about the smoothing by binning, which we discussed in the class. Consider the following variable:**

**(1). Smooth the variable using the equal-width binning method and bin means. Use three bins.**

Range: 147 – 23 = 124

Bin width: 124/3 = 41.33

Bin1: 23, 23+41.33 = 64.33

Bin2: 64.33, 64.33+41.33 = 105.66

Bin3: 105.66, 147

So:

Bin1: 23, 26, 30, 32, 35, 47, 48, 52, 59, 63

Bin2: 92

Bin3: 110, 132, 147

We can get:

Bin1 mean: 41.5.

Bin2 mean: 92.

Bin3 mean: 129.67.

Smoothed:

<41.5, 41.5, 41.5, 41.5, 41.5, 41.5, 41.5, 41.5, 41.5, 41.5, 92, 129.67, 129.67, 129.67>

**(2). Smooth the variable using the equal-width binning method and bin boundaries. Use three bins.**

Bin1: 23, 64.33

Bin2: 64.33, 105.66

Bin3: 105.67, 147

Check the number near which boundaries.

Smoothed:

<23, 23, 23, 23, 23, 64.33, 64.33, 64.33, 64.33, 64.33, 105.66, 105.66, 147, 147>

**Problem 2 (10 points). Use the hw3\_p2.csv file for this problem. The dataset has 9 attributes. The first 8 attributes are predictor attributes and the last attribute, area, is the class attribute. (1). Which predictor attribute has the highest correlation with the class attribute?**

> data <- read.csv("hw3\_p2.csv")

>

> #(1):

> # Calculate the correlation of each predictor attribute with the class attribute

> correlations <- cor(data[, -9], data$area)

>

> # Find the predictor attribute with the highest correlation with the class attribute

> max\_correlation\_attribute <- rownames(correlations)[max\_cor\_index]

> print(paste("The predictor attribute with the highest correlation to the class attribute is:", max\_correlation\_attribute))

[1] "The predictor attribute with the highest correlation to the class attribute is: DMC"

**So, it is DMC.**

**(2). Among the 8 predictor attributes, which pair of attributes has the highest correlation?**

> cor\_matrix <- cor(data[, -9])

>

> # Since the diagonal of the correlation matrix is 1, we set the diagonal to 0

> diag(cor\_matrix) <- 0

>

> # Find the pair of attributes with the highest correlation

> max\_cor\_pair <- which(abs(cor\_matrix) == max(abs(cor\_matrix), na.rm = TRUE), arr.ind = TRUE)

>

> print(paste("The pair of predictor attributes with the highest correlation is:",

+ colnames(data)[max\_cor\_pair[1,1]], "and", colnames(data)[max\_cor\_pair[1,2]]))

[1] "The pair of predictor attributes with the highest correlation is: DC and DMC"

**So, it is DC and DMC.**

**Problem 3 (10 points). Use the hw3\_p3.csv file for this problem. Determine whether there is a correlation between housing and class using the chi-square test method that we discussed in the class. You may use any tool to derive a contingency table. However, you must do all calculations yourself after that, including the calculation of expected values and the test statistic. You may use a spreadsheet software or R just for the purpose of calculation.**

> data <- read.csv("hw3\_p3.csv")

>

> # Create contingency table

> contingency\_table <- table(data$housing, data$class)

> print(contingency\_table)

not\_recom priority spec\_prior very\_recom

convenient 1440 1618 1052 208

critical 1440 1252 1608 20

less\_conv 1440 1396 1384 100

>

> #Calculate Expected Frequencies

> row\_totals <- rowSums(contingency\_table)

> col\_totals <- colSums(contingency\_table)

> grand\_total <- sum(contingency\_table)

>

> expected <- outer(row\_totals, col\_totals, FUN = "\*") / grand\_total

> print(expected)

not\_recom priority spec\_prior very\_recom

convenient 1439.555 1421.561 1347.584 109.2996

critical 1440.222 1422.219 1348.208 109.3502

less\_conv 1440.222 1422.219 1348.208 109.3502

>

> #Calculate the Chi-Square Statistic

> chi\_square\_statistic <- sum((contingency\_table - expected)^2 / expected)

> print(chi\_square\_statistic)

[1] 326.7832

>

> #Determine the Significance

> test <- chisq.test(contingency\_table)

> print(test$p.value)

[1] 1.481377e-67

This value is too small, there is a significant relationship between the housing and class variables.

**Problem 4 (10 points). Use the hw3\_p4.csv file for this problem. This problem is about the PCA that we discussed in the class. Use R for this problem.**

**(1). Standardize the first 8 attributes (predictors) using the z-score method.**

RCode

# Standardize the data

data[, 1:8] <- scale(data[, 1:8])

**(2). Split the dataset into a training dataset and a test dataset with the ratio of 66:34. Use 31 as the seed (so that I may replicate your code) and you must do stratified splitting.**

Rcode

# Install and load the required package

if (!require(caret)) {

install.packages("caret", dependencies = TRUE)

library(caret)

}

# Stratified split based on the 'class' column

set.seed(31)

index <- createDataPartition(data$class, p = 0.66, list = FALSE)

train\_data <- data[index, ]

test\_data <- data[-index, ]

**(3). Apply PCA on the training dataset. If you want to keep 90% of total variability, how many principal components you should keep? If you want to keep 70% of total variability, how many principal components you should keep?**

> pca\_model <- prcomp(train\_data[, 1:8], center = TRUE, scale. = TRUE)

> explained\_var <- cumsum(pca\_model$sdev^2) / sum(pca\_model$sdev^2)

> n\_90 <- which(explained\_var >= 0.90)[1]

> n\_70 <- which(explained\_var >= 0.70)[1]

> print(paste("Number of principal components to retain 90% variability:", n\_90))

[1] "Number of principal components to retain 90% variability: 6"

> print(paste("Number of principal components to retain 70% variability:", n\_70))

[1] "Number of principal components to retain 70% variability: 4"

**90%: 6**

**70: 4**

**(4). Transform (or project) both the training dataset and the test dataset to new datasets with new attributes (principal components) and show the first 6 tuples of each dataset.**

> train\_transformed <- as.data.frame(predict(pca\_model, newdata = train\_data[, 1:8]))

> test\_transformed <- as.data.frame(predict(pca\_model, newdata = test\_data[, 1:8]))

>

> #print first 6 tuples of the transformed training dataset:

> print(head(train\_transformed))

PC1 PC2 PC3 PC4 PC5 PC6 PC7 PC8

3 -0.4879559 0.8023952 -1.2162910 -0.6348981 0.6422054 -0.51883450 1.1993803 0.5019050

5 -2.6812067 -1.1847038 -0.2109701 0.3735212 2.9452350 0.88846089 -0.6888653 -1.0253764

6 0.8032524 1.0395401 1.3705604 0.1917006 -0.1628263 0.68264433 0.2145107 0.1725617

7 0.5743705 1.2570571 -0.2607765 -0.6213125 -0.1675232 -0.01642395 -0.1404216 -0.1474141

9 -0.7393810 -1.3703196 0.2220055 0.8763599 -0.0667635 -0.66718008 0.3041993 -0.1643207

10 0.4083325 0.3207018 -0.2910972 -0.1951770 -0.4239334 -0.32438169 0.4177096 -0.6466624

>

> #print first 6 tuples of the transformed test dataset:

> print(head(test\_transformed))

PC1 PC2 PC3 PC4 PC5 PC6 PC7 PC8

1 -3.267111 0.4033363 1.44167077 0.3320431 -0.1370759 -0.50421217 -0.23363472 0.1499428

2 -0.262277 1.0369554 -1.40083623 -0.8765347 0.6342607 -0.25754991 1.07853078 0.1206940

4 -3.006797 -2.0008445 1.52731817 -0.9067686 2.5978337 0.18662337 -0.36348623 -0.8750620

8 -1.012434 -2.2096786 -0.57509682 1.0469865 2.1789473 0.01331764 0.13649563 0.2477515

15 1.541912 0.8672929 0.02715926 -0.2561275 -1.0937910 -0.17151175 -0.01142486 -0.0478117