Homework 13

Jon Flees

12/10/2021

Problem 1

```
df <- read.csv('00 kc_house_data.csv')
(nData = nrow(df))
(set.seed(0))
(trainIdx <- sample(seq_len(nData), floor(nData * 0.70 )))
(Train <- df[trainIdx, ])
(Test <-  df[-trainIdx, ])
```

```
(Avg_house_price_train <- mean(Train[ ,3]))
```

```
## [1] 539998.8
```

```
(Avg_house_price_test <- mean(Test[ ,3]))
```

```
## [1] 540296.6
```

Problem 2

1.

```
ar <- read.csv('arrythmia.csv')
ar <- ar[1:100,1:5]
standard.data <- scale(ar)
colMeans(standard.data)       # Check that column means are 0
```

```
##           X75            X0          X190           X80           X91
##  2.282896e-16 -1.199041e-16  3.652634e-16 -2.351938e-16 -2.046974e-18
```

```
apply(standard.data, 2, sd)   # Check that std.devs are 1
```

```
##  X75   X0 X190  X80  X91
##    1    1    1    1    1
```

```
standard.data[1:5,1]          # Normalized data from first 5 Rows of 1st Column
```

```
##         1         2         3         4         5
##  0.6269961  0.4927357  0.5598659  1.9024701 -2.2596028
```

2.

```
NumCategories <- 10
(c1 = cut(ar[,1], breaks=NumCategories))    # Equi-Width Bin Ranges
```

```
##   [1] (55.6,63.4] (47.8,55.6] (47.8,55.6] (71.2,79.1] (8.8,16.6]  (32.2,40]
##   [7] (47.8,55.6] (40,47.8]   (47.8,55.6] (55.6,63.4] (40,47.8]   (47.8,55.6]
##  [13] (24.4,32.2] (40,47.8]   (40,47.8]   (40,47.8]   (40,47.8]   (71.2,79.1]
##  [19] (55.6,63.4] (24.4,32.2] (40,47.8]   (32.2,40]   (55.6,63.4] (32.2,40]
##  [25] (40,47.8]   (32.2,40]   (24.4,32.2] (55.6,63.4] (47.8,55.6] (47.8,55.6]
```

```
##  [31] (55.6,63.4] (47.8,55.6] (47.8,55.6] (63.4,71.2] (40,47.8]   (47.8,55.6]
##  [37] (32.2,40]   (55.6,63.4] (40,47.8]   (40,47.8]   (32.2,40]   (24.4,32.2]
##  [43] (32.2,40]   (32.2,40]   (71.2,79.1] (63.4,71.2] (24.4,32.2] (40,47.8]
##  [49] (32.2,40]   (71.2,79.1] (24.4,32.2] (32.2,40]   (16.6,24.4] (47.8,55.6]
##  [55] (71.2,79.1] (32.2,40]   (40,47.8]   (40,47.8]   (24.4,32.2] (0.922,8.8]
##  [61] (32.2,40]   (32.2,40]   (24.4,32.2] (47.8,55.6] (40,47.8]   (47.8,55.6]
##  [67] (24.4,32.2] (40,47.8]   (63.4,71.2] (32.2,40]   (32.2,40]   (24.4,32.2]
##  [73] (40,47.8]   (32.2,40]   (32.2,40]   (40,47.8]   (32.2,40]   (55.6,63.4]
##  [79] (63.4,71.2] (32.2,40]   (55.6,63.4] (71.2,79.1] (47.8,55.6] (55.6,63.4]
##  [85] (71.2,79.1] (16.6,24.4] (32.2,40]   (63.4,71.2] (71.2,79.1] (47.8,55.6]
##  [91] (63.4,71.2] (55.6,63.4] (55.6,63.4] (24.4,32.2] (47.8,55.6] (32.2,40]
##  [97] (32.2,40]   (47.8,55.6] (32.2,40]   (47.8,55.6]
## 10 Levels: (0.922,8.8] (8.8,16.6] (16.6,24.4] (24.4,32.2] ... (71.2,79.1]
```

```
(count1 = as.vector(table(c1)))   # Counts for 1st Column
```

```
##  [1]  1  1  2 11 23 18 18 12  6  8
```

```
elementCounts <- matrix(count1)   # Initialize matrix to store each column's counts per bin

for ( i in 2:length(ar) ) {       # Loop to Calc each column's counts per bin
  (ci = cut(ar[,i], breaks=NumCategories))
  (counti = as.vector(table(ci)))
  ( elementCounts <- cbind(elementCounts,counti))   # Add counts vector to matrix
}
colnames(elementCounts) <- c("Col_1", "Col_2", "Col_3", "Col_4", "Col_5")
elementCounts     # Completed matrix
```

```
##       Col_1 Col_2 Col_3 Col_4 Col_5
##  [1,]     1    32     1     1    30
##  [2,]     1     0     0     0    35
##  [3,]     2     0     0     0    21
##  [4,]    11     0     0     2     9
##  [5,]    23     0     4    21     1
##  [6,]    18     0    23    29     0
##  [7,]    18     0    47    25     2
##  [8,]    12     0    16    12     0
##  [9,]     6     0     7     6     0
## [10,]     8    68     2     4     2
```

```
elementCounts[,1]     # Counts from 1st Column from matrix
```

```
##  [1]  1  1  2 11 23 18 18 12  6  8
```

Problem 3

a.

```
raw_us <- read.csv('RawDataUSCities.csv')
(nRaw <- nrow(raw_us))
(colsRaw <- length(raw_us))
standard.us <- scale(raw_us[1:nRaw,3:colsRaw])
```

```
standard.us[1:6,]
```

```
##   PercentageBlack PercentageHispanic PercentageAsian   MedianAge
## 1     -1.17872113          1.2389537     -0.36257405  0.06134197
## 2      2.35518849         -0.7644344     -0.45230197 -0.43961742
```

```
## 3     -0.68176509          0.5104489     -0.27284613 -1.44153619
## 4      1.91344978         -0.8251431     -0.45230197  0.56230135
## 5      0.09127764         -0.2180558     -0.09339029 -0.94057681
## 6      0.42258167         -0.8251431     -0.36257405  0.06134197
##   UnemploymentRate PerCapitaIncomeThousand
## 1       -0.7514633              -0.8752312
## 2       -0.7514633               0.3243864
## 3       -1.4953360              -0.5753268
## 4        1.4801550               0.3243864
## 5       -0.7514633               0.9241951
## 6       -1.4953360              -0.2754224
```

b.

```
normalize = function(x) {
  return( (x-min(x)) / (max(x)-min(x)) )
}
us_norm <- as.data.frame(lapply(raw_us[3:colsRaw], normalize))
head(us_norm,6)
```

```
##   PercentageBlack PercentageHispanic PercentageAsian MedianAge UnemploymentRate
## 1      0.02666667         0.50000000      0.01428571 0.4444444              0.2
## 2      0.88000000         0.01470588      0.00000000 0.3333333              0.2
## 3      0.14666667         0.32352941      0.02857143 0.1111111              0.0
## 4      0.77333333         0.00000000      0.00000000 0.5555556              0.8
## 5      0.33333333         0.14705882      0.05714286 0.2222222              0.2
## 6      0.41333333         0.00000000      0.01428571 0.4444444              0.0
##   PerCapitaIncomeThousand
## 1               0.2777778
## 2               0.5000000
## 3               0.3333333
## 4               0.5000000
## 5               0.6111111
## 6               0.3888889
```