HW 1

Adon Rosen

Date: 2020-09-30

Question 1

For the Ambulatory Medical Care data set given below, give one specific example for each of the classification, clustering, association rule mining, and anomaly detection tasks that can be performed on the data (i.e. give an example of a data mining task that would require classification; an example of a data mining task that would require clustering, etc.). For each task, state how the data matrix should be constructed (i.e., specify the rows and columns of the matrix). The Ambulatory Medical Care dataset contains the demographic and medical visit information for each patient (e.g., gender, age, duration of visit, physician's diagnosis, symptoms, medication, etc.).

Classification: Gender can be predicted as a function of symptoms, medication, and duration of visit. Here the rows will be the individuals, and the columns will be symptoms coded as a binary variable indicating presence of the symptom, it will also include medication coded in a similar fashion, finally the gender will be included in the data matrix.

Clustering: Diagnostic severity can attempted to be modeled via an unsupervised classification paradigm and the physicians diagnosis can be used as the outcome. Data included for the clustering will include demographic variables, and symptoms, every row will represent an individual patient. Columns will be the demographic variables such as age and race, age will be coded as a continuous variable, race will be coded as a nominal factor, and symptoms wil be coded as a binary variable for all symptoms indicating presence or not of the symptom.

Association rule mining: Association rule mining can be used to compare symptoms reported and the diagnosis the physician scored for each individual patient. The data will be organized with various symptoms as the columns and binary true/false indicators to indicate whether symptoms are preset in a patient where each row represents an individual patient.

Anomaly Detection: Anomalys for diagnoses can be attempted to be detected as a function of reported symptoms. The data will be organized with various symptoms as the columns and binary true/false indicators to indicate whether symptoms are preset in a patient where each row represents an individual patient.

Question 2

For each attribute given, classify its type as:

- discrete or continuous AND
- qualitative or quantitative AND
- nominal, ordinal, interval, or ratio

Example: Age in years. Answer: Discrete, quantitative, ratio.

1. Daily user traffic volume at YouTube.com (i.e., number of daily visitors who visited the Web site).
2. Air pressure of a car/bicycle tire (in psi).
3. Homeland Security Advisory System ratings - code red/orange/etc.

4. Credit card number.
5. The wealth of a nation measured in terms of gross domestic product (GDP) per capita above the world's average of $10,500.

Answers:

1. Discrete ; Quantitative ; Ratio
2. Continuous ; Quantitative ; Ratio
3. Discrete ; Qualitative ; Ordinal
4. Discrete ; Qualitative ; Nominal
5. Continuous ; Quantitative ; Interval

Question 3 part 1

```
x <- c(0,1,1,1,0,1)
y <- c(0,1,0,1,0,1)
## The SMC is calculated as the number of matches divided by the number of entries
f.zo <- length(which(x==0 & y==1))
f.oz <- length(which(x==1 & y==0))
f.zz <- length(which(x==0 & y==0))
f.oo <- length(which(x==1 & y==1))

# First calc the SMC
smc.one <- sum(f.zz, f.oo) / length(x)

# Now calc the jaccard coef
# number of 11 mathces versus number of non-zero attributes
jac.one <- f.oo / (f.oz + f.zo + f.oo)

## Now calc the cosine correlation
dot.prod <- pracma::dot(x,y)
distance.x <- dot(x,x)^.5
distance.y <- dot(y,y)^.5
cos.value <-  dot.prod / (distance.x * distance.y)
# now confirm
cos.value.lsa <- lsa::cosine(x,y)

## Now calculate the correlation
cor.val <- cor(x, y)

## Now calculate the euclidean distance
first.diff <- sum(x - y ^ 2) ^ 1

## Now do city block
second.diff <- sum(x - y ^ 2) ^ 1/2

## Now create a table with these data
metric <- c("SMC", "Jaccard Coefficient", "Cosine", "Correlation", "Minkowski (r=1)", "Minkowski (r=2)"
values <- c(smc.one, jac.one, cos.value, cor.val, first.diff, second.diff)
to.kable <- rbind(metric, round(values, 2))
to.kable %>%
```

```
  kbl() %>%
  kable_classic_2(full_width = F)
```

metric	SMC	Jaccard Coefficient	Cosine	Correlation	Minkowski (r=1)	Minkowski (r=2)
	0.83	0.75	0.87	0.71	1	0.5

Question 3 part 2

The most appropriate metric to use for these two vectors would be the Mahalanobis distance. This is the most appropriate metric given the difference in scale between the two vectors.the mean of the x values is roughly 30 whereas the mean for the y values is roughly 65000. This difference in scale makes the euclidean difference undesierable because while a differences may be consistenly large it would be difficult to tell if the difference is large in either the Y or X direction in isolation. The simple matching coefficient is not appropriate because the values would nee dot be binarized prior to it's calculation and binarizing continous data typically requires an arbitrary rule set. The Mahalanobis thus is the most appropriate because it looks beyond raw units and uses varainces and covarainces to assign distances between points.

Question 4

Question 4-1

Write a function that estimates the missing values in the dataset by using the means of the available values as follows: for each continuous attribute that has one or more missing values, compute the mean of the available values in that attribute and replace every missing value in that attribute with the mean. Then use this function to estimate all the missing values of continuous attributes in the dataset.

```
# Load the data
in.data <- read.csv('../Data/auto-mpg.csv', na.strings = '?')

## Create a function to impute using the mean function given a vector
impute.mean <- function(x){
  # Check to see if the values are numeric - if they are run the imputation
  # if values are not numeric - reutnr the input data
  if(!is.numeric(x)){
    out.dat <- x
  }
  else{
    out.dat <- replace(x, is.na(x), mean(x, na.rm = TRUE))
  }
  return(out.dat)
}
orig.data <- in.data
# Now create the imputed dataset
in.data[] <- lapply(in.data, impute.mean)
## Now compare the complete cases between the two
table(complete.cases(orig.data), complete.cases(in.data)) %>%
  kbl() %>%
  kable_classic_2(full_width = F)
```

	TRUE
FALSE	6
TRUE	392

Question 4-2

Draw a random sample of size 50 without replacement out of all rows in the dataset, and discard the rest.

```
# Draw a random sample without replacment
drawn.sample <- in.data[sample.int(dim(in.data)[1], size=50, replace=FALSE),]
```

Question 4-3

Write a function that performs equal frequency discretization of a continuous attribute.

```
# Create a function which will do equal frequncy discretization
equalFreqBins <- function(x,n, na.rm=TRUE){
  ## Find the length
  nx <- length(x)
  ## Deal with uneven values
  nrepl <- floor(nx/n)
  ## Add to those for uneven lengths
  nplus <- sample(1:n,nx - nrepl*n)
  nrep <- rep(nrepl,n)
  nrep[nplus] <- nrepl+1
  ## Now assign the values
  x[order(x)] <- rep(seq.int(n),nrep)
  return(x)
}
```

Question 4-4

Write a function that performs equal width discretization of a continuous attribute.

```
# Create a function which will create equal width bins
equalWidthBins <- function(x, n, na.rm=TRUE){
  ## Find the cut values
  widthVal <- (range(x, na.rm = na.rm)[2] - range(x, na.rm = na.rm)[1])/n
  cutVals <- seq(range(x, na.rm = na.rm)[1], range(x, na.rm = na.rm)[2], widthVal)
  ## Now assign the output
  x.out <- x
  for(i in 1:length(cutVals)-1){x.out[which(x>=cutVals[i])] <- i}
  return(x.out)
}
```

Question 4-5

Do the following:

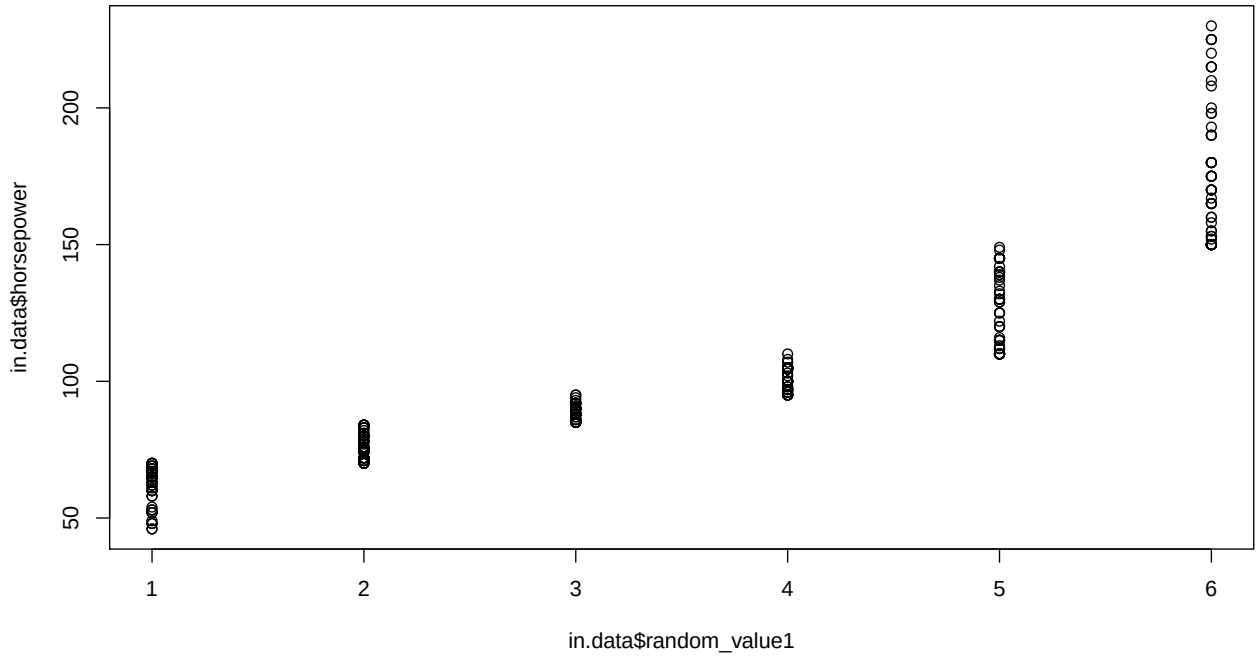
a. Use the function implemented in Task 3 to discretize the horsepower attribute, and plot the original values of this attribute along with the split points resulting from the discretization function (like the plot in Figure 2.13 on Page 67 in the textbook). To do this, create a new attribute named 'random_value' and assign a random value for this 'random_value' attribute for every row in the dataset, and then plot horsepower vs. 'random_value'.

b. Repeat Task 5.a, but use the discretization function implemented in Task 4.

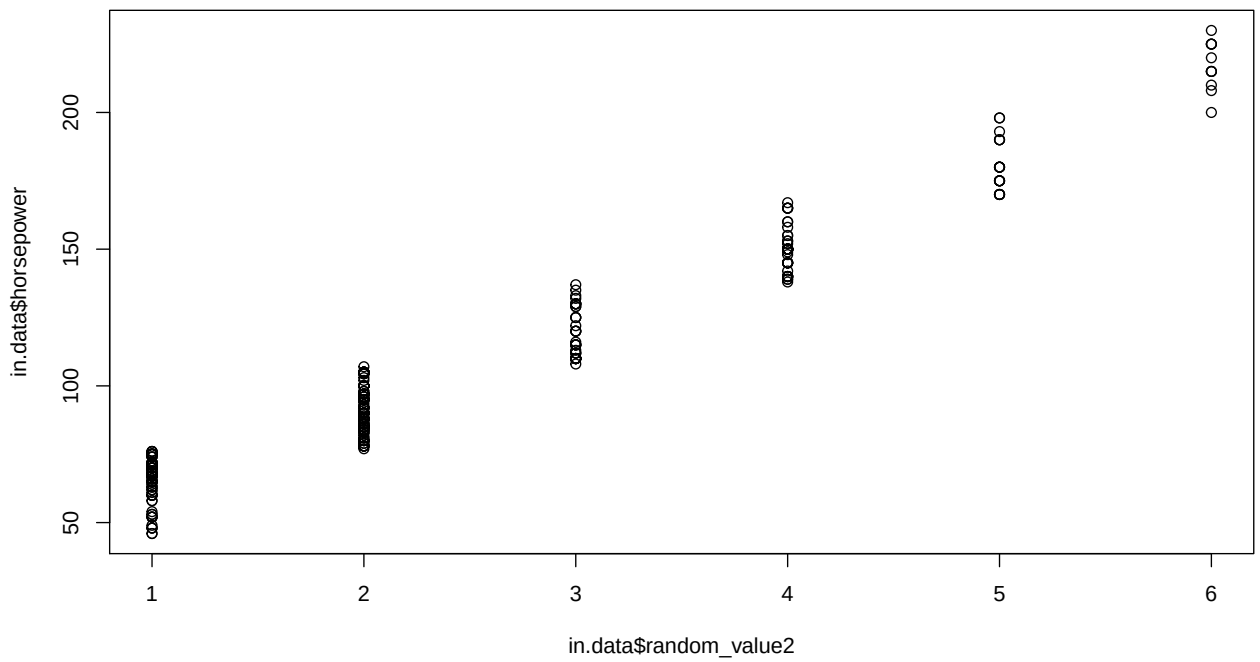
c. Inspect the two plots produced in Tasks 5.a and 5.b and discuss what you have observed.

```
in.data$random_value1 <- equalFreqBins(in.data$horsepower, 6)
in.data$random_value2 <- equalWidthBins(in.data$horsepower, 6)
```

```r
## Create the first plot
plot(in.data$random_value1, in.data$horsepower)
```




```r
## Now do the second plot
plot(in.data$random_value2, in.data$horsepower)
```

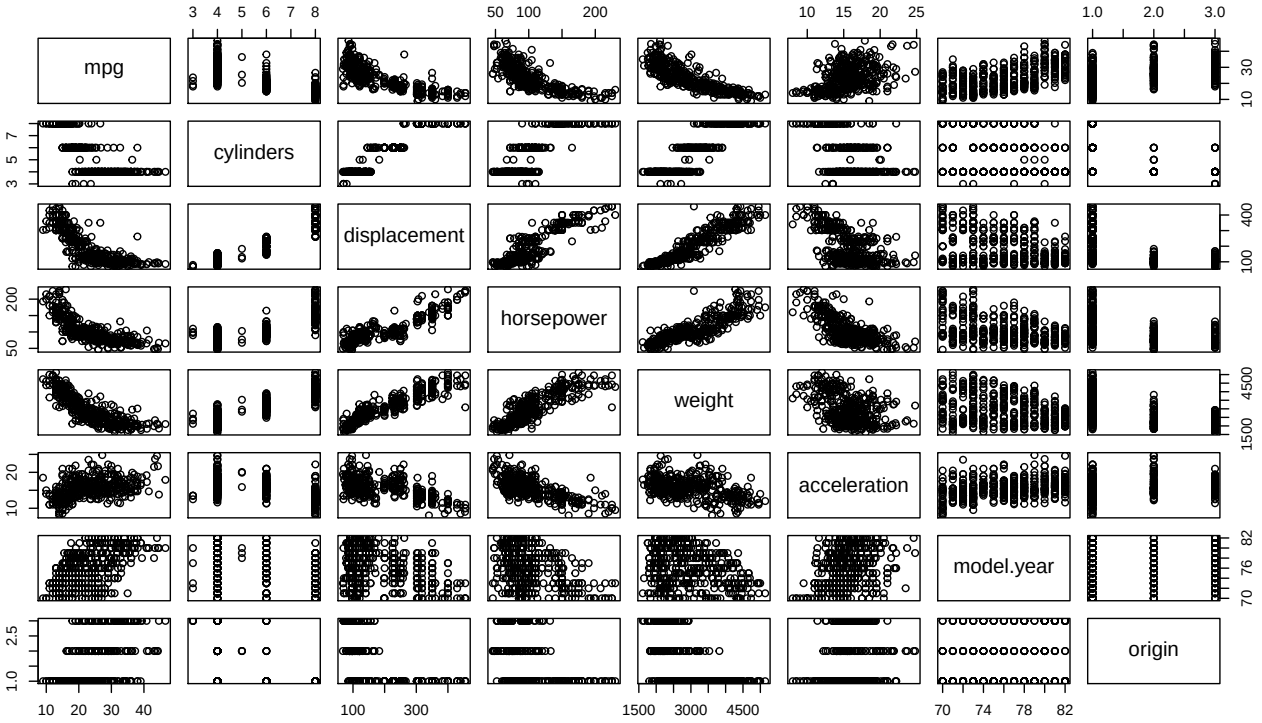


The graphs above suggest the horsepower variable is skewed to the right. This is illustrated by the difference in the variability in the higher bins in the first graph and the difference in the number of values in the second graph.

Question 4-6

Create a matrix of scatter plots of the continuous attributes. Of these attributes, which ones do you think exhibit the greatest variability? Do you observe any correlation between any pair of attributes? Why?

```
pairs(in.data[,1:8])
```



The scatter plots do suggest there is increased variablity in the displacement, horsepower, and weight variables; however, by nature these variables are graded on a continuous rather than a discrete scale. Furthermore, these same variables appear to be highly correlated with each other, suggesting that increases in any one of them yields a rather linear increase in the others. This is likely due to the nature of engines and how larger engines yield more power as well as weight.