Assignment 6

edelsonc

September 11, 2016

This assignment will investigate descritizing data in R. This will be done using a data set looking as Chronic Kidney Disease, provided from the University of California Irvine.

First the data is downloaded and extracted using bash.

```
# file url
url="https://archive.ics.uci.edu/ml/machine-learning-databases/00336/Chronic_Kidney_Disease.rar"

# use wget command to download the file
wget "$url"

# extract the file's content
unrar e Chronic_Kidney_Disease.rar

# record the data providence
echo "Downloaded from $url at $( date )" > KidneyProvidence.txt
```

Now lets investigate the extracted files

```
# list content of current directory, looking for items with chronic in
# their name
ls | grep chronic
```

```
## chronic_kidney_disease.arff
## chronic_kidney_disease.info.txt
## chronic_kidney_disease_full.arff
```

The data files have been extracted. However, they are in an unusual format, a `.arff` document. In order to read this exotic file into memory, we'll need to use a special package, `RWeka`

```
# load RWeka library
library(RWeka)

# use arff reading function to load data into memory
kidney <- read.arff("chronic_kidney_disease_full.arff")
str(kidney)
```

```
## 'data.frame':    400 obs. of  25 variables:
##  $ age  : num  48 7 62 48 51 60 68 24 52 53 ...
##  $ bp   : num  80 50 80 70 80 90 70 NA 100 90 ...
##  $ sg   : Factor w/ 5 levels "1.005","1.010",..: 4 4 2 1 2 3 2 3 3 4 ...
##  $ al   : Factor w/ 6 levels "0","1","2","3",..: 2 5 3 5 3 4 1 3 4 3 ...
##  $ su   : Factor w/ 6 levels "0","1","2","3",..: 1 1 4 1 1 1 1 5 1 1 ...
##  $ rbc  : Factor w/ 2 levels "normal","abnormal": NA NA 1 1 1 NA NA 1 1 2 ...
##  $ pc   : Factor w/ 2 levels "normal","abnormal": 1 1 1 2 1 NA 1 2 2 2 ...
##  $ pcc  : Factor w/ 2 levels "present","notpresent": 2 2 2 1 2 2 2 2 1 1 ...
```

```
##  $ ba   : Factor w/ 2 levels "present","notpresent": 2 2 2 2 2 2 2 2 2 2 ...
##  $ bgr  : num  121 NA 423 117 106 74 100 410 138 70 ...
##  $ bu   : num  36 18 53 56 26 25 54 31 60 107 ...
##  $ sc   : num  1.2 0.8 1.8 3.8 1.4 1.1 24 1.1 1.9 7.2 ...
##  $ sod  : num  NA NA NA 111 NA 142 104 NA NA 114 ...
##  $ pot  : num  NA NA NA 2.5 NA 3.2 4 NA NA 3.7 ...
##  $ hemo : num  15.4 11.3 9.6 11.2 11.6 12.2 12.4 12.4 10.8 9.5 ...
##  $ pcv  : num  44 38 31 32 35 39 36 44 33 29 ...
##  $ wbcc : num  7800 6000 7500 6700 7300 7800 NA 6900 9600 12100 ...
##  $ rbcc : num  5.2 NA NA 3.9 4.6 4.4 NA 5 4 3.7 ...
##  $ htn  : Factor w/ 2 levels "yes","no": 1 2 2 1 2 1 2 2 1 1 ...
##  $ dm   : Factor w/ 2 levels "yes","no": 1 2 1 2 2 1 2 1 1 1 ...
##  $ cad  : Factor w/ 2 levels "yes","no": 2 2 2 2 2 2 2 2 2 2 ...
##  $ appet: Factor w/ 2 levels "good","poor": 1 1 2 2 1 1 1 1 1 2 ...
##  $ pe   : Factor w/ 2 levels "yes","no": 2 2 2 1 2 1 2 1 2 2 ...
##  $ ane  : Factor w/ 2 levels "yes","no": 2 2 1 1 2 2 2 2 1 1 ...
##  $ class: Factor w/ 2 levels "ckd","notckd": 1 1 1 1 1 1 1 1 1 1 ...
```

Completeness of data

We can investigate how many rows are complete using the `complete.cases` command

```
sum(complete.cases(kidney))/nrow(kidney) * 100
```

```
## [1] 39.5
```

So 39.5% or the rows contain an entry for every column.

We wish to investigate the potassium column, and inorder to do this we'll eliminate rows where there is an `NA` for potassium

```
kidney<- kidney[!(is.na(kidney$pot)),]
```

Descritizing the Data

There are a number of different schreme for descritizing data. A common one is to simply break the data into n many chunks, each with the same number of points in it

```
bin_loc <- quantile(kidney$pot, probs= seq(0,1, by=0.2))

kidney$dis_fix_freq <- with(kidney, cut(pot, breaks=bin_loc, include.lowest=TRUE, right=FALSE))

summary(kidney$dis_fix_freq)
```

```
## [2.5,3.7) [3.7,4.1) [4.1,4.7)   [4.7,5)    [5,47]
##        58        54        74        60        66
```

An alternate scheme for descretizing is to divide the range into equal parts

```
MIN <- range(kidney$pot)[1]
MAX <- range(kidney$pot)[2]

bin_loc <- seq(MIN, MAX, (MAX - MIN)/5)

kidney$dis_width <- with(kidney, cut(pot, breaks=bin_loc, include.lowest=TRUE, right=FALSE))

summary(kidney$dis_width)
```

```
##  [2.5,11.4) [11.4,20.3) [20.3,29.2) [29.2,38.1)   [38.1,47]
##         310           0           0           0           2
```

We can see that for this example, this is not the best discretization.

An alternate regime is to use a topdown discretization process. The `R` package `discretization` lets you do that using the function `disc.Topdown` with anyone of three algorithims: CAIM, CACC, Ameva.

```
library("discretization")

kidney_comp <- kidney[complete.cases(kidney),]

top_down <- disc.Topdown(kidney_comp[c(13,14)], method=1)

kidney_comp$top_down <- factor(top_down$Disc.data[[c(2)]])

summary(kidney_comp$top_down)
```

```
## 2.5 2.9   3 3.2 3.3 3.4 3.5 3.6 3.7 3.8 3.9   4 4.1 4.2 4.3 4.4 4.5 4.6
##   1   2   1   1   2   1  23   5   7   7   8   6   3   2   3   4   7   4
## 4.7 4.8 4.9   5 5.2 5.4 5.5 5.6 5.7 5.8 6.4 6.5 7.6  47
##  11   8  16  23   2   1   1   1   3   1   1   1   1   1
```

Alternatively, a bottom up approach can be used

```
bottom_up <- chiM(kidney_comp[c(13, 14)])

kidney_comp$bottom_up <- factor(bottom_up$Disc.data[[c(2)]])

summary(kidney_comp$bottom_up)
```

```
## 2.5 2.9   3 3.2 3.3 3.4 3.5 3.6 3.7 3.8 3.9   4 4.1 4.2 4.3 4.4 4.5 4.6
##   1   2   1   1   2   1  23   5   7   7   8   6   3   2   3   4   7   4
## 4.7 4.8 4.9   5 5.2 5.4 5.5 5.6 5.7 5.8 6.4 6.5 7.6  47
##  11   8  16  23   2   1   1   1   3   1   1   1   1   1
```