Part 2 - conversions

jean wills

06/07/2020

arules

# PART 2: “HOUSE OF DATA CONVERSIONS”

### 1. read in 2 files

KEEPING DEFAULT attribute for now

### 2a. the max # of contacts was 275 which is way too large=> delete

### 2b. switch -1 -> 0 in ‘pdays’

### 2c. switch duration in seconds to minutes for easier use

keeping “unknowns” in attributes for now

BM\_mini version - can use same code EXCEPT for numeric to nominal - this is most likely different

# step 1 - read in fresh files  
library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

BM <- read.csv("/Users/jeanwills/Desktop/CKME136/bank\_full.csv", header=T, sep = ";", stringsAsFactors = T, na.strings = "NA")  
# BM\_mini <- read.csv("/Users/jeanwills/Desktop/CKME136/bank.csv", header=T, sep = ";", stringsAsFactors = T, na.strings = "NA")  
# BM<- BM\_mini  
# step 2 - make changes  
# may delete later - attribute 'default' in column 5  
# BM<- select(BM,-5)  
# specific deletion where BM$previous == 275   
BM<- BM[grep("275", BM$previous, invert=TRUE),]  
# switch -1 -> 0 in 'pdays'  
BM$pdays<- ifelse(BM$pdays == "-1", 0, BM$pdays)  
# switch duration in seconds to minutes for easier use  
BM$duration<- BM$duration/60  
# now copy to other datasets, depending on what to do

## for making $y numeric:

#  
BM\_01 <- BM  
BM\_01$y<- ifelse(BM\_01$y==c("yes"), 1, 0)  
# BM\_num$y<- ifelse(BM\_num$y==c("yes"), 1, 0)

# PART 2 - CONVERSIONS

## Part 2 - step 1: Conversion - numeric to nominal - RESULT is BM\_fact

method 1: equal frequency: manually using quantiles to break up the data. method 2: equal width = (max - min)/k, where k is the # of groupings that you choose - not using

# Conversion from Numeric to Categorical  
# using method 1: approx. equal frequency - manually with trial and error  
# 7 numeric attributes to transform: age, balance, day, duration, campaign, pdays, previous  
# set up to work with BM, also works for BM\_mini  
#  
# if not manual, do something like:  
# BM\_fact$age<- cut(BM\_fact$age, breaks = quantile(BM\_fact$age, c(0, 0.25, 0.50, 0.75, 1), na.rm=TRUE, include.lowest=TRUE ))  
BM\_fact<- BM  
# age: (95-18)/5 = ~15 year breaks  
BM\_fact$age<- cut(BM\_fact$age, breaks = c(0,33,39,48,100), labels = c("18-33", "33-39", "39-48", "48-95"))  
BM\_fact$balance<- cut(BM\_fact$balance, breaks = c(-9000, 72, 448, 1428, 105000), labels = c("negative", "72-448", "448-1428", "over1428"))  
# BM\_fact$day< as.factor(BM\_fact$day)  
BM\_fact$day<- cut(BM\_fact$day, breaks = c(0,8,16,21,32), labels = c("1-7", "8-15", "16-20", "21-31"))   
BM\_fact$duration<- cut(BM\_fact$duration, breaks = c(-1,2,3,5,90), labels = c("0-2m", "2-3m", "3-5m", "over5min"))   
BM\_fact$campaign<- cut(BM\_fact$campaign, breaks = c(0, 2, 3, 70), labels = c("1-2", "2-3", "over3"))   
BM\_fact$pdays<- cut(BM\_fact$pdays, breaks = c(-1,100,200,400,600,800,1000), labels = c("0-100", "100-200","200-400","400-600","600-800","over800"))   
BM\_fact$previous<- cut(BM\_fact$previous, breaks = c(-1,1,10,20,30,60), labels = c("0-1", "1-10","10-20","20-30","over30"))   
str(BM\_fact)

## 'data.frame': 45210 obs. of 17 variables:  
## $ age : Factor w/ 4 levels "18-33","33-39",..: 4 3 1 3 1 2 1 3 4 3 ...  
## $ job : Factor w/ 12 levels "admin.","blue-collar",..: 5 10 3 2 12 5 5 3 6 10 ...  
## $ marital : Factor w/ 3 levels "divorced","married",..: 2 3 2 2 3 2 3 1 2 3 ...  
## $ education: Factor w/ 4 levels "primary","secondary",..: 3 2 2 4 4 3 3 3 1 2 ...  
## $ default : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 2 1 1 ...  
## $ balance : Factor w/ 4 levels "negative","72-448",..: 4 1 1 4 1 2 2 1 2 3 ...  
## $ housing : Factor w/ 2 levels "no","yes": 2 2 2 2 1 2 2 2 2 2 ...  
## $ loan : Factor w/ 2 levels "no","yes": 1 1 2 1 1 1 2 1 1 1 ...  
## $ contact : Factor w/ 3 levels "cellular","telephone",..: 3 3 3 3 3 3 3 3 3 3 ...  
## $ day : Factor w/ 4 levels "1-7","8-15","16-20",..: 1 1 1 1 1 1 1 1 1 1 ...  
## $ month : Factor w/ 12 levels "apr","aug","dec",..: 9 9 9 9 9 9 9 9 9 9 ...  
## $ duration : Factor w/ 4 levels "0-2m","2-3m",..: 3 2 1 1 3 2 3 4 1 1 ...  
## $ campaign : Factor w/ 3 levels "1-2","2-3","over3": 1 1 1 1 1 1 1 1 1 1 ...  
## $ pdays : Factor w/ 6 levels "0-100","100-200",..: 1 1 1 1 1 1 1 1 1 1 ...  
## $ previous : Factor w/ 5 levels "0-1","1-10","10-20",..: 1 1 1 1 1 1 1 1 1 1 ...  
## $ poutcome : Factor w/ 4 levels "failure","other",..: 4 4 4 4 4 4 4 4 4 4 ...  
## $ y : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...

summary(BM\_fact)

## age job marital education   
## 18-33:13083 blue-collar:9732 divorced: 5207 primary : 6851   
## 33-39:10279 management :9457 married :27213 secondary:23202   
## 39-48:10660 technician :7597 single :12790 tertiary :13300   
## 48-95:11188 admin. :5171 unknown : 1857   
## services :4154   
## retired :2264   
## (Other) :6835   
## default balance housing loan contact   
## no :44395 negative:11317 no :20081 no :37966 cellular :29284   
## yes: 815 72-448 :11291 yes:25129 yes: 7244 telephone: 2906   
## 448-1428:11305 unknown :13020   
## over1428:11297   
##   
##   
##   
## day month duration campaign pdays   
## 1-7 :11639 may :13766 0-2m :14044 1-2 :30048 0-100 :38391   
## 8-15 :11718 jul : 6895 2-3m : 8616 2-3 : 5521 100-200: 2866   
## 16-20:10782 aug : 6247 3-5m :10277 over3: 9641 200-400: 3719   
## 21-31:11071 jun : 5341 over5min:12273 400-600: 182   
## nov : 3970 600-800: 41   
## apr : 2932 over800: 11   
## (Other): 6059   
## previous poutcome y   
## 0-1 :39726 failure: 4901 no :39921   
## 1-10 : 5191 other : 1839 yes: 5289   
## 10-20 : 239 success: 1511   
## 20-30 : 43 unknown:36959   
## over30: 11   
##   
##

## Part 2 - step 2a: Conversion A- Categorical to Numeric - RESULT is BM\_num

numeric still NOT normalized/scaled

BM\_num <- BM  
BM\_num$job<- as.numeric(BM\_num$job) #12  
# marital: 1-single, 2-married, 3-divorced  
BM\_num$marital<- ifelse(BM\_num$marital == c("single"), 1,   
 ifelse(BM\_num$marital== c("married"), 2, 3))  
# education: 0:unknown, 1: primary, 2:secondary, 3:divorced  
BM\_num$education<- ifelse(BM\_num$education == c("unknown"), 0,   
 ifelse(BM\_num$education == c("primary"), 1,   
 ifelse(BM\_num$education == c("secondary"), 2, 3)))  
# default, housing, loan: if yes then 0 else 1  
# BM\_num$housing<- as.numeric(BM\_num$housing) #2  
BM\_num$default<- ifelse(BM\_num$default == c("yes"), 0, 1) #2  
BM\_num$housing<- ifelse(BM\_num$housing == c("yes"), 0, 1) #2  
BM\_num$loan<- ifelse(BM\_num$loan == c("yes"), 0, 1) #2  
BM\_num$contact<- as.numeric(BM\_num$contact) #3  
# month: jan:1, feb:2.....dec:12  
BM\_num$month<- ifelse(BM\_num$month == "jan", 1,   
 ifelse(BM\_num$month == "feb", 2,   
 ifelse(BM\_num$month == "mar", 3,  
 ifelse(BM\_num$month == "apr", 4,   
 ifelse(BM\_num$month == "may", 5,   
 ifelse(BM\_num$month == "jun", 6,  
 ifelse(BM\_num$month == "jul", 7,  
 ifelse(BM\_num$month == "aug", 8,  
 ifelse(BM\_num$month == "sep", 9,  
 ifelse(BM\_num$month == "oct", 10,  
 ifelse(BM\_num$month == "nov", 11, 12)))))))))))  
# poutcome: 0:unknown,other, 1:failure, 2: success  
BM\_num$poutcome<- ifelse(BM\_num$poutcome == c("failure"), 1, ifelse(BM\_num$poutcome== c("success"), 2, 0))   
# result: BM\_num with only numeric data (NOT scaled)

## Part 2 - step 2b: Conversion B- Categorical to Numeric - as dummy variables - RESULT is BM\_dummy

numeric still NOT normalized/scaled

# extra:  
# BM$y<- factor(BM$y, levels = c("yes", "no"), labels = c("yes", "no"))  
# round(prop.table(table(BM$y)) \* 100, digits=1)  
# dummy coding  
# keep y as factor  
# dataset is too large to run through Select Attrbutes - try BM\_mini version  
BM\_dummy <- BM  
# now create new attributes for each component in attribute less 1 category  
# for example, marital has 3 attributes so we need 2 dummy variables (each of 0,1)  
# BM\_fact$y<- ifelse(BM\_fact$y==c("yes"), 0, 1)  
#  
BM\_dummy$job1 <- ifelse(BM\_dummy$job == c("admin."), 1, 0)  
BM\_dummy$job2 <- ifelse(BM\_dummy$job == c("blue-collar"), 1, 0)  
BM\_dummy$job3 <- ifelse(BM\_dummy$job == c("entrepreneur"), 1, 0)  
BM\_dummy$job4 <- ifelse(BM\_dummy$job == c("housemaid"), 1, 0)  
BM\_dummy$job5 <- ifelse(BM\_dummy$job == c("management"), 1, 0)  
BM\_dummy$job6 <- ifelse(BM\_dummy$job == c("retired"), 1, 0)  
BM\_dummy$job7 <- ifelse(BM\_dummy$job == c("self-employed"), 1, 0)  
BM\_dummy$job8 <- ifelse(BM\_dummy$job == c("services"), 1, 0)  
BM\_dummy$job9 <- ifelse(BM\_dummy$job == c("student"), 1, 0)  
BM\_dummy$job10 <- ifelse(BM\_dummy$job == c("technician"), 1, 0)  
BM\_dummy$job11 <- ifelse(BM\_dummy$job == c("unemployed"), 1, 0)  
#  
BM\_dummy$mar1 <- ifelse(BM\_dummy$marital== c("divorced"), 1, 0)  
BM\_dummy$mar2 <- ifelse(BM\_dummy$marital== c("married"), 1, 0)  
#  
BM\_dummy$ed1 <- ifelse(BM\_dummy$education == c("primary"), 1, 0)  
BM\_dummy$ed2 <- ifelse(BM\_dummy$education == c("secondary"), 1, 0)  
BM\_dummy$ed3 <- ifelse(BM\_dummy$education == c("tertiary"), 1, 0)  
#  
BM\_dummy$hous1 <- ifelse(BM\_dummy$housing == c("no"), 1, 0)  
BM\_dummy$def1 <- ifelse(BM\_dummy$default == c("no"), 1, 0)  
BM\_dummy$loan1 <- ifelse(BM\_dummy$loan == c("no"), 1, 0)  
#  
BM\_dummy$cont1 <- ifelse(BM\_dummy$contact == c("cellular"), 1, 0)  
BM\_dummy$cont2 <- ifelse(BM\_dummy$contact == c("telephone"), 1, 0)  
#  
BM\_dummy$mon1 <- ifelse(BM\_dummy$month == c("jan"), 1, 0)  
BM\_dummy$mon2 <- ifelse(BM\_dummy$month == c("feb"), 1, 0)  
BM\_dummy$mon3 <- ifelse(BM\_dummy$month == c("mar"), 1, 0)  
BM\_dummy$mon4 <- ifelse(BM\_dummy$month == c("apr"), 1, 0)  
BM\_dummy$mon5 <- ifelse(BM\_dummy$month == c("may"), 1, 0)  
BM\_dummy$mon6 <- ifelse(BM\_dummy$month == c("jun"), 1, 0)  
BM\_dummy$mon7 <- ifelse(BM\_dummy$month == c("jul"), 1, 0)  
BM\_dummy$mon8 <- ifelse(BM\_dummy$month == c("aug"), 1, 0)  
BM\_dummy$mon9 <- ifelse(BM\_dummy$month == c("sep"), 1, 0)  
BM\_dummy$mon10 <- ifelse(BM\_dummy$month == c("oct"), 1, 0)  
BM\_dummy$mon11 <- ifelse(BM\_dummy$month == c("nov"), 1, 0)  
#  
BM\_dummy$pout1 <- ifelse(BM\_dummy$poutcome == c("failure"), 1, 0)  
BM\_dummy$pout2 <- ifelse(BM\_dummy$poutcome == c("success"), 1, 0)  
BM\_dummy$pout3 <- ifelse(BM\_dummy$poutcome == c("other"), 1, 0)  
# ok so we end up with a lot - then we have to delete the original Factor attributes

next step in step 2b process…

## Part 2 - step 2 Conversion B cont’d. take out the original attributes and the answer Y column and RESULT: BM\_dummy

# keep the y column in separate file for now  
BM\_y<- BM\_dummy[16]  
# now take out y in file - to place later at the end  
BM\_dummy<- BM\_dummy[-16]  
# now take out the remaining factors one at a time  
BM\_dummy<- BM\_dummy[-15]  
BM\_dummy<- BM\_dummy[-10]  
BM\_dummy<- BM\_dummy[-8]  
BM\_dummy<- BM\_dummy[-7]  
BM\_dummy<- BM\_dummy[-6]  
BM\_dummy<- BM\_dummy[-4]  
BM\_dummy<- BM\_dummy[-3]  
BM\_dummy<- BM\_dummy[-2]  
# add back y at the end   
# y is still a factor  
BM\_dummy<- cbind(BM\_dummy, BM\_y)  
# result: BM\_dummy with dummies for categorical = > all numeric (still not scaled)

## PART 3 - SCALE original NUMERIC DATA - 2 methods to use

for models that use (i.e. Euclidian) distance between 2 points or require normal data (regression, PCA, etc.) 2 methods of normalizing numeric data: 1. Use min-max normalization: x\_new = (x - x\_min) / (x\_max - x\_min) 2. Use z-score standardization: x\_new = (x - Mean) / Sd

code for generic version then can swap in datasets: step 1: BM-> BM\_scale, BM\_z step 2: BM\_num-> BM\_num\_scale, BM\_num\_z

## Part 3 - step 1a - BM-> BM\_scale

# KEEP: age-1, balance-6, day-10, duration-12, campaign-13, pdays-14, previous-15  
BM\_step<-BM  
# use grep ?  
BM\_step<-BM\_step[-17]  
BM\_step<-BM\_step[-16]  
BM\_step<-BM\_step[-11]  
BM\_step<-BM\_step[-9]  
BM\_step<-BM\_step[-8]  
BM\_step<-BM\_step[-7]  
BM\_step<-BM\_step[-5]  
BM\_step<-BM\_step[-4]  
BM\_step<-BM\_step[-3]  
BM\_step<-BM\_step[-2]  
# method 1  
normalize<- function(x) {  
 return ((x - min(x)) / (max(x) - min(x)))  
}  
BM\_scale<- as.data.frame(lapply(BM\_step[1:7], normalize))  
# now recombine dataframes with the nominal components  
# at this stage we only have numeric data  
BM\_scale$job<-BM$job  
BM\_scale$marital<-BM$marital  
BM\_scale$education<-BM$education  
BM\_scale$default<-BM$default  
BM\_scale$housing<-BM$housing  
BM\_scale$loan<-BM$loan  
BM\_scale$contact<-BM$contact  
BM\_scale$month<-BM$month  
BM\_scale$poutcome<-BM$poutcome  
BM\_scale$y<-BM$y  
# str(BM\_scale)  
# result used BM file but now BM\_scale with normalized numeric data  
# and y is factor  
# to convert y to numeric use next line  
# BM\_scale$y<- ifelse(BM\_scale$y==c("yes"), 1, 0)

## Part 3 - step 1b - BM-> BM\_z

# KEEP: age-1, balance-6, day-10, duration-12, campaign-13, pdays-14, previous-15  
BM\_step<-BM  
# use grep ?  
BM\_step<-BM\_step[-17]  
BM\_step<-BM\_step[-16]  
BM\_step<-BM\_step[-11]  
BM\_step<-BM\_step[-9]  
BM\_step<-BM\_step[-8]  
BM\_step<-BM\_step[-7]  
BM\_step<-BM\_step[-5]  
BM\_step<-BM\_step[-4]  
BM\_step<-BM\_step[-3]  
BM\_step<-BM\_step[-2]  
#  
# method 2  
z\_norm<- function(x) {  
 return ((x - mean(x)) / sd(x))  
}  
BM\_z<- as.data.frame(lapply(BM\_step[1:7], z\_norm))  
# now recombine dataframes with the nominal components  
# at this stage we only have numeric data  
BM\_z$job<-BM$job  
BM\_z$marital<-BM$marital  
BM\_z$education<-BM$education  
BM\_z$default<-BM$default  
BM\_z$housing<-BM$housing  
BM\_z$loan<-BM$loan  
BM\_z$contact<-BM$contact  
BM\_z$month<-BM$month  
BM\_z$poutcome<-BM$poutcome  
BM\_z$y<-BM$y  
# result used BM file but now BM\_z with normalized numeric data  
# and y is factor  
# to convert y to numeric use next line  
# BM\_z$y<- ifelse(BM\_z$y==c("yes"), 1, 0)

## Part 3 - step 2a - BM\_num-> BM\_num\_scale

You must do Part 2 -step 2a first to get BM\_num

# KEEP: age-1, balance-6, day-10, duration-12, campaign-13, pdays-14, previous-15  
# run section of code above in Part 3 step 2, to get BM\_num first (or BM\_dummy)  
# or BM\_step<- BM\_dummy  
BM\_step<-BM\_num  
BM\_step<-BM\_step[-17]  
BM\_step<-BM\_step[-16]  
BM\_step<-BM\_step[-11]  
BM\_step<-BM\_step[-9]  
BM\_step<-BM\_step[-8]  
BM\_step<-BM\_step[-7]  
BM\_step<-BM\_step[-5]  
BM\_step<-BM\_step[-4]  
BM\_step<-BM\_step[-3]  
BM\_step<-BM\_step[-2]  
# method 1  
normalize<- function(x) {  
 return ((x - min(x)) / (max(x) - min(x)))  
}  
BM\_num\_scale<- as.data.frame(lapply(BM\_step[1:7], normalize))  
# now recombine dataframes with the nominal components  
BM\_num\_scale$job<-BM\_num$job  
BM\_num\_scale$marital<-BM\_num$marital  
BM\_num\_scale$education<-BM\_num$education  
BM\_num\_scale$default<-BM\_num$default  
BM\_num\_scale$housing<-BM\_num$housing  
BM\_num\_scale$loan<-BM\_num$loan  
BM\_num\_scale$contact<-BM\_num$contact  
BM\_num\_scale$month<-BM\_num$month  
BM\_num\_scale$poutcome<-BM\_num$poutcome  
BM\_num\_scale$y<-BM\_num$y  
str(BM\_num\_scale)

## 'data.frame': 45210 obs. of 17 variables:  
## $ age : num 0.519 0.338 0.195 0.377 0.195 ...  
## $ balance : num 0.0923 0.0731 0.0728 0.0865 0.0728 ...  
## $ day : num 0.133 0.133 0.133 0.133 0.133 ...  
## $ duration : num 0.0531 0.0307 0.0155 0.0187 0.0403 ...  
## $ campaign : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ pdays : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ previous : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ job : num 5 10 3 2 12 5 5 3 6 10 ...  
## $ marital : num 2 1 2 2 1 2 1 3 2 1 ...  
## $ education: num 3 2 2 0 0 3 3 3 1 2 ...  
## $ default : num 1 1 1 1 1 1 1 0 1 1 ...  
## $ housing : num 0 0 0 0 1 0 0 0 0 0 ...  
## $ loan : num 1 1 0 1 1 1 0 1 1 1 ...  
## $ contact : num 3 3 3 3 3 3 3 3 3 3 ...  
## $ month : num 5 5 5 5 5 5 5 5 5 5 ...  
## $ poutcome : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ y : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...

# result used BM\_num file but now BM\_num\_scale with normalized numeric data  
# and y is still a factor  
# to convert y to numeric use next line  
# BM\_num\_scale$y<- ifelse(BM\_num\_scale$y==c("yes"), 1, 0)

## Part 3 - step 2b - BM\_num-> BM\_num\_z

You must do Part 2 -step 2a first to get BM\_num

# KEEP: age-1, balance-6, day-10, duration-12, campaign-13, pdays-14, previous-15  
# or BM\_step<- BM\_dummy  
BM\_step<-BM\_num  
BM\_step<-BM\_step[-17]  
BM\_step<-BM\_step[-16]  
BM\_step<-BM\_step[-11]  
BM\_step<-BM\_step[-9]  
BM\_step<-BM\_step[-8]  
BM\_step<-BM\_step[-7]  
BM\_step<-BM\_step[-5]  
BM\_step<-BM\_step[-4]  
BM\_step<-BM\_step[-3]  
BM\_step<-BM\_step[-2]  
#  
# method 2  
z\_norm<- function(x) {  
 return ((x - mean(x)) / sd(x))  
}  
BM\_num\_z<- as.data.frame(lapply(BM\_step[1:7], z\_norm))  
# now recombine dataframes with the nominal components  
BM\_num\_z$job<-BM\_num$job  
BM\_num\_z$marital<-BM\_num$marital  
BM\_num\_z$education<-BM\_num$education  
BM\_num\_z$default<-BM\_num$default  
BM\_num\_z$housing<-BM\_num$housing  
BM\_num\_z$loan<-BM\_num$loan  
BM\_num\_z$contact<-BM\_num$contact  
BM\_num\_z$month<-BM\_num$month  
BM\_num\_z$poutcome<-BM\_num$poutcome  
BM\_num\_z$y<-BM\_num$y  
# result used BM file but now BM\_num\_z with normalized numeric data  
# and y is still a factor  
# to convert y to numeric use next line  
# BM\_num\_z$y<- ifelse(BM\_num\_z$y==c("yes"), 1, 0)