Part 2 - conversions

jean wills

06/07/2020

# PART 2: “HOUSE OF DATA CONVERSIONS”

NOTE: BM\_mini version - can use same code

### 1. read in dataset and make changes - on “BM” dataset

# 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  
#   
# make changes  
# delete 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

### for making $y numeric: all else the same

#  
# 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  
# method 1: equal frequency   
# 7 numeric attributes to transform: age, balance, day, duration, campaign, pdays, previous  
# set up to work with "BM" as input   
BM\_fact<- BM  
# AUTOMATIC: quantiles not working....  
# BM\_fact$age<- cut(BM\_fact$age, breaks = quantile(BM\_fact$age, c(0, 0.25, 0.50, 0.75, 1), na.rm=TRUE, include.lowest=TRUE ))  
# BM\_fact$balance<- cut(BM\_fact$age, breaks = quantile(BM\_fact$balance, c(0, 0.25, 0.50, 0.75, 1), na.rm=TRUE, include.lowest=TRUE ))  
# BM\_fact$day<- cut(BM\_fact$age, breaks = quantile(BM\_fact$day, c(0, 0.25, 0.50, 0.75, 1), na.rm=TRUE, include.lowest=TRUE ))  
# BM\_fact$duration<- cut(BM\_fact$age, breaks = quantile(BM\_fact$duration, c(0, 0.25, 0.50, 0.75, 1), na.rm=TRUE, include.lowest=TRUE ))  
# BM\_fact$campaign<- cut(BM\_fact$age, breaks = quantile(BM\_fact$campaign, c(0, 0.25, 0.50, 0.75, 1), na.rm=TRUE, include.lowest=TRUE ))  
# BM\_fact$pdays<- cut(BM\_fact$age, breaks = quantile(BM\_fact$pdays, c(0, 0.25, 0.50, 0.75, 1), na.rm=TRUE, include.lowest=TRUE ))  
# BM\_fact$previous<- cut(BM\_fact$age, breaks = quantile(BM\_fact$previous, c(0, 0.25, 0.50, 0.75, 1), na.rm=TRUE, include.lowest=TRUE ))  
# MANUAL  
# method 1: approx. equal frequency - manually with trial and error  
#  
# age: (95-18)/5 = ~15 year breaks  
BM\_fact$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)  
# summary(BM\_fact)

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

# numeric still NOT normalized/scaled, actually ALL DATA not normalized/scaled EXCEPT default, housing, loan  
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”

# original numeric still NOT normalized/scaled  
# 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)  
# summary(BM\_dummy)  
# ok so we end up with a lot - then we have to delete the original Factor attributes  
## now 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[17]  
# now take out y in file - to place later at the end  
BM\_dummy<- BM\_dummy[-17]  
# now take out the remaining factors one at a time  
BM\_dummy<- BM\_dummy[-16]  
BM\_dummy<- BM\_dummy[-11]  
BM\_dummy<- BM\_dummy[-9]  
BM\_dummy<- BM\_dummy[-8]  
BM\_dummy<- BM\_dummy[-7]  
BM\_dummy<- BM\_dummy[-5]  
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)  
# summary(BM\_dummy)  
# result: BM\_dummy with dummies for categorical = > all numeric (still not scaled)

## PART 3 - SCALE original NUMERIC DATA - 2 methods to use and MIN\_MAX seems best

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. “scale”: Use min-max normalization: x\_new = (x - x\_min) / (x\_max - x\_min) 2. “z”: Use z-score standardization: x\_new = (x - Mean) / Sd

code for generic version then can swap in datasets: step 1a: BM-> BM\_z step 1b: BM-> BM\_scale step 2: Use For BM\_num ONLY-> BM\_num\_scale [scale MIN\_MAX version] step 3: Use for BM\_dummy ONLY -> BM\_dumax [scale MIN\_MAX version] step 4: Log transformations of numeric data for getting variables to be normally distributed (ONLY works for age, day, campaign) - do NOT use

# step 1a - not used - using preProcess to get - [Standardization with mean=0]

# we start with BM dataset, and we run code on 7 numeric data only  
library(plyr)

## ------------------------------------------------------------------------------

## You have loaded plyr after dplyr - this is likely to cause problems.  
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:  
## library(plyr); library(dplyr)

## ------------------------------------------------------------------------------

##   
## Attaching package: 'plyr'

## The following objects are masked from 'package:dplyr':  
##   
## arrange, count, desc, failwith, id, mutate, rename, summarise,  
## summarize

library(readr)  
library(ggplot2)  
library(GGally)

## Registered S3 method overwritten by 'GGally':  
## method from   
## +.gg ggplot2

##   
## Attaching package: 'GGally'

## The following object is masked from 'package:dplyr':  
##   
## nasa

library(dplyr)  
library(mlbench)  
# glimpse(BM)  
# summary(BM)  
# use caret and preprocess to normalize data with mean of zero  
library(caret)

## Loading required package: lattice

preproc1<- preProcess(BM[,c(1,6,10,12:15)], method=c("center", "scale"))  
BM\_norm1<- predict(preproc1, BM[,c(1,6,10,12:15)])  
# summary(BM\_norm1)  
# now recombine dataframes with the nominal components  
BM\_norm1$job<-BM$job  
BM\_norm1$marital<-BM$marital  
BM\_norm1$education<-BM$education  
BM\_norm1$default<-BM$default  
BM\_norm1$housing<-BM$housing  
BM\_norm1$loan<-BM$loan  
BM\_norm1$contact<-BM$contact  
BM\_norm1$month<-BM$month  
BM\_norm1$poutcome<-BM$poutcome  
BM\_norm1$y<-BM$y  
# summary(BM\_norm1)

## step 1a - BM-> BM\_z - [Standardization with mean=0] - same results as above

# KEEP: age-1, balance-6, day-10, duration-12, campaign-13, pdays-14, previous-15  
# uses BM  
# method 2  
z\_norm<- function(x) {  
 return ((x - mean(x)) / sd(x))  
}  
BM\_z<- as.data.frame(lapply(BM[,c(1,6,10,12:15)], 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  
# summary(BM\_z)  
# 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)

# 1b: using preProcess to get scale - [MIN\_MAX version] - not used

# uses BM  
# we start with BM dataset, and we run code on 7 numeric data only  
preproc1<- preProcess(BM[,c(1,6,10,12:15)], method=c("range"))  
BM\_norm2<- predict(preproc1, BM[,c(1,6,10,12:15)])  
# summary(BM\_norm2)  
# now recombine dataframes with the nominal components  
BM\_norm2$job<-BM$job  
BM\_norm2$marital<-BM$marital  
BM\_norm2$education<-BM$education  
BM\_norm2$default<-BM$default  
BM\_norm2$housing<-BM$housing  
BM\_norm2$loan<-BM$loan  
BM\_norm2$contact<-BM$contact  
BM\_norm2$month<-BM$month  
BM\_norm2$poutcome<-BM$poutcome  
BM\_norm2$y<-BM$y  
# summary(BM\_norm2)

## step 1b - BM-> BM\_scale - [MIN\_MAX version] - same results as above

# KEEP: age-1, balance-6, day-10, duration-12, campaign-13, pdays-14, previous-15  
# uses BM  
normalize<- function(x) {  
 return ((x - min(x)) / (max(x) - min(x)))  
}  
BM\_scale<- as.data.frame(lapply(BM[,c(1,6,10,12:15)], normalize))  
# now recombine dataframes with the nominal components  
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  
# summary(BM\_scale)  
# 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)

## step 2 - USE For BM\_num ONLY-> BM\_num\_scale - [MIN\_MAX version]

get BM\_num first from above

# keep default(5), housing(7) and loan(8) as is as they are binary, the rest can ALL be scaled  
# run section of code above in Part 3 step 2, to get BM\_num first   
normalize<- function(x) {  
 return ((x - min(x)) / (max(x) - min(x)))  
}  
BM\_num\_scale<- as.data.frame(lapply(BM\_num[,c(1:4,6,9:16)], normalize))  
# now recombine dataframes with the nominal components  
BM\_num\_scale$default<-BM\_num$default  
BM\_num\_scale$housing<-BM\_num$housing  
BM\_num\_scale$loan<-BM\_num$loan  
BM\_num\_scale$y<-BM\_num$y  
summary(BM\_num\_scale)

## age job marital education   
## Min. :0.0000 Min. :0.00000 Min. :0.0000 Min. :0.0000   
## 1st Qu.:0.1948 1st Qu.:0.09091 1st Qu.:0.0000 1st Qu.:0.6667   
## Median :0.2727 Median :0.36364 Median :0.5000 Median :0.6667   
## Mean :0.2979 Mean :0.39452 Mean :0.4161 Mean :0.6868   
## 3rd Qu.:0.3896 3rd Qu.:0.63636 3rd Qu.:0.5000 3rd Qu.:1.0000   
## Max. :1.0000 Max. :1.00000 Max. :1.0000 Max. :1.0000   
## balance contact day month   
## Min. :0.00000 Min. :0.0000 Min. :0.0000 Min. :0.0000   
## 1st Qu.:0.07346 1st Qu.:0.0000 1st Qu.:0.2333 1st Qu.:0.3636   
## Median :0.07687 Median :0.0000 Median :0.5000 Median :0.4545   
## Mean :0.08517 Mean :0.3201 Mean :0.4936 Mean :0.4677   
## 3rd Qu.:0.08577 3rd Qu.:1.0000 3rd Qu.:0.6667 3rd Qu.:0.6364   
## Max. :1.00000 Max. :1.0000 Max. :1.0000 Max. :1.0000   
## duration campaign pdays previous   
## Min. :0.00000 Min. :0.00000 Min. :0.00000 Min. :0.000000   
## 1st Qu.:0.02094 1st Qu.:0.00000 1st Qu.:0.00000 1st Qu.:0.000000   
## Median :0.03660 Median :0.01613 Median :0.00000 Median :0.000000   
## Mean :0.05249 Mean :0.02845 Mean :0.04708 Mean :0.009901   
## 3rd Qu.:0.06486 3rd Qu.:0.03226 3rd Qu.:0.00000 3rd Qu.:0.000000   
## Max. :1.00000 Max. :1.00000 Max. :1.00000 Max. :1.000000   
## poutcome default housing loan   
## Min. :0.00000 Min. :0.000 Min. :0.0000 Min. :0.0000   
## 1st Qu.:0.00000 1st Qu.:1.000 1st Qu.:0.0000 1st Qu.:1.0000   
## Median :0.00000 Median :1.000 Median :0.0000 Median :1.0000   
## Mean :0.08762 Mean :0.982 Mean :0.4442 Mean :0.8398   
## 3rd Qu.:0.00000 3rd Qu.:1.000 3rd Qu.:1.0000 3rd Qu.:1.0000   
## Max. :1.00000 Max. :1.000 Max. :1.0000 Max. :1.0000   
## y   
## no :39921   
## yes: 5289   
##   
##   
##   
##

# 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)

# step 3: Use for BM\_dummy ONLY -> BM\_dumax: [MIN\_MAX version]

get BM\_dummy first from above

BM\_dumax<-BM\_dummy  
#  
normalize<- function(x) {  
 return ((x - min(x)) / (max(x) - min(x)))  
}  
BM\_2<- as.data.frame(lapply(BM\_dummy[,c(1:7)], normalize))  
# delete first 7 of original BM\_dummy  
 BM\_dumax<- BM\_dumax[-1]  
 BM\_dumax<- BM\_dumax[-1]  
 BM\_dumax<- BM\_dumax[-1]  
 BM\_dumax<- BM\_dumax[-1]  
 BM\_dumax<- BM\_dumax[-1]  
 BM\_dumax<- BM\_dumax[-1]  
 BM\_dumax<- BM\_dumax[-1]  
# now add back normalized first 7 numerical  
BM\_dumax<- cbind(BM\_2, BM\_dumax)  
# summary(BM\_dumax)

# step 4: Log transformations of numeric data for getting variables to be normally distributed - NOT GOOD -do not use

ONLY works for age, day, campaign because the other numeric data has zeros or negative numbers….

# run code   
BMlog<- BM  
BMlog$age<- log(BM$age)  
# log2balance = log(BM$balance)  
BMlog$day = log(BM$day)  
# log2duration = log(BM$duration)  
BMlog$campaign = log(BM$campaign)  
# log2pdays = log(BM$pdays)  
# log2previous = log(BM$previous)  
summary(BMlog)

## age job marital education   
## Min. :2.890 blue-collar:9732 divorced: 5207 primary : 6851   
## 1st Qu.:3.497 management :9457 married :27213 secondary:23202   
## Median :3.664 technician :7597 single :12790 tertiary :13300   
## Mean :3.679 admin. :5171 unknown : 1857   
## 3rd Qu.:3.871 services :4154   
## Max. :4.554 retired :2264   
## (Other) :6835   
## default balance housing loan contact   
## no :44395 Min. : -8019 no :20081 no :37966 cellular :29284   
## yes: 815 1st Qu.: 72 yes:25129 yes: 7244 telephone: 2906   
## Median : 448 unknown :13020   
## Mean : 1362   
## 3rd Qu.: 1428   
## Max. :102127   
##   
## day month duration campaign   
## Min. :0.000 may :13766 Min. : 0.000 Min. :0.0000   
## 1st Qu.:2.079 jul : 6895 1st Qu.: 1.717 1st Qu.:0.0000   
## Median :2.773 aug : 6247 Median : 3.000 Median :0.6931   
## Mean :2.562 jun : 5341 Mean : 4.303 Mean :0.7068   
## 3rd Qu.:3.045 nov : 3970 3rd Qu.: 5.317 3rd Qu.:1.0986   
## Max. :3.434 apr : 2932 Max. :81.967 Max. :4.1431   
## (Other): 6059   
## pdays previous poutcome y   
## Min. : 0.00 Min. : 0.0000 failure: 4901 no :39921   
## 1st Qu.: 0.00 1st Qu.: 0.0000 other : 1839 yes: 5289   
## Median : 0.00 Median : 0.0000 success: 1511   
## Mean : 41.01 Mean : 0.5743 unknown:36959   
## 3rd Qu.: 0.00 3rd Qu.: 0.0000   
## Max. :871.00 Max. :58.0000   
##