Chairty Data EDA

Section 1 of the final group project for 422

This is an Exploratory Data Analysis of the charity data.

#############################################  
# PREDICT 422 Practical Machine Learning #  
# Laura Stotts Gorans, Phil Harm, Tim Kiely #  
#############################################  
  
  
# Load packages. Check if they are installed first though  
list.of.packages <- c(  
 "tidyverse" # tidyverse is a wrapper that contains dplyr, ggplot, tidyr, readr and a bunch of other great Hadley stuff  
 ,"corrplot" #correlation plot  
 ,"lattice" #quantile plots  
 )  
  
new.packages <- list.of.packages[!(list.of.packages %in% installed.packages()[,"Package"])]  
if(length(new.packages)) install.packages(new.packages)  
lapply(list.of.packages,library, character.only=T)

## [[1]]  
## [1] "dplyr" "purrr" "readr" "tidyr" "tibble"   
## [6] "ggplot2" "tidyverse" "stats" "graphics" "grDevices"  
## [11] "utils" "datasets" "methods" "base"   
##   
## [[2]]  
## [1] "corrplot" "dplyr" "purrr" "readr" "tidyr"   
## [6] "tibble" "ggplot2" "tidyverse" "stats" "graphics"   
## [11] "grDevices" "utils" "datasets" "methods" "base"   
##   
## [[3]]  
## [1] "lattice" "corrplot" "dplyr" "purrr" "readr"   
## [6] "tidyr" "tibble" "ggplot2" "tidyverse" "stats"   
## [11] "graphics" "grDevices" "utils" "datasets" "methods"   
## [16] "base"

# Load Data  
rm(list=ls())  
charity <- read\_csv("charity.csv") # load the "charity.csv" file  
  
# quick view of the data  
glimpse(charity)

## Observations: 8,009  
## Variables: 24  
## $ ID <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17...  
## $ reg1 <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, ...  
## $ reg2 <int> 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, ...  
## $ reg3 <int> 1, 1, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, ...  
## $ reg4 <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, ...  
## $ home <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, ...  
## $ chld <int> 1, 2, 1, 1, 0, 1, 3, 3, 2, 3, 3, 3, 0, 0, 0, 2, 1, 0, 0, ...  
## $ hinc <int> 4, 4, 5, 4, 4, 5, 4, 2, 3, 4, 2, 4, 4, 4, 5, 3, 4, 4, 4, ...  
## $ genf <int> 1, 0, 1, 0, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 0, 0, 1, ...  
## $ wrat <int> 8, 8, 8, 8, 4, 9, 8, 5, 5, 7, 8, 6, 9, 8, 8, 8, 9, 9, 1, ...  
## $ avhv <int> 302, 262, 303, 317, 295, 114, 145, 165, 194, 200, 152, 27...  
## $ incm <int> 76, 130, 61, 121, 39, 17, 39, 34, 112, 38, 46, 69, 54, 21...  
## $ inca <int> 82, 130, 90, 121, 71, 25, 42, 35, 112, 58, 46, 69, 54, 36...  
## $ plow <int> 0, 1, 6, 0, 14, 44, 10, 19, 0, 5, 20, 0, 14, 32, 1, 15, 2...  
## $ npro <int> 20, 95, 64, 51, 85, 83, 50, 11, 75, 42, 100, 98, 13, 54, ...  
## $ tgif <int> 81, 156, 86, 56, 132, 131, 74, 41, 160, 63, 414, 169, 34,...  
## $ lgif <int> 81, 16, 15, 18, 15, 5, 6, 4, 28, 12, 25, 29, 9, 5, 36, 5,...  
## $ rgif <int> 19, 17, 10, 7, 10, 3, 5, 2, 34, 10, 14, 36, 7, 4, 20, 4, ...  
## $ tdon <int> 17, 19, 22, 14, 10, 13, 22, 20, 14, 19, 39, 23, 19, 15, 2...  
## $ tlag <int> 6, 3, 8, 7, 6, 4, 3, 7, 4, 3, 7, 7, 11, 9, 8, 9, 7, 4, 6,...  
## $ agif <dbl> 21.05, 13.26, 17.37, 9.59, 12.07, 4.12, 6.50, 3.45, 14.00...  
## $ donr <int> 0, 1, NA, NA, 1, 1, 0, 0, NA, 0, 0, 1, 1, 1, 1, NA, 1, 1,...  
## $ damt <int> 0, 15, NA, NA, 17, 12, 0, 0, NA, 0, 0, 17, 12, 15, 18, NA...  
## $ part <chr> "train", "train", "test", "test", "valid", "train", "vali...

# EDA

##########################################  
#REQUIREMENT 1 - Exploratory Data Analyis#  
##########################################  
charity.eda <- charity

# Missing data?

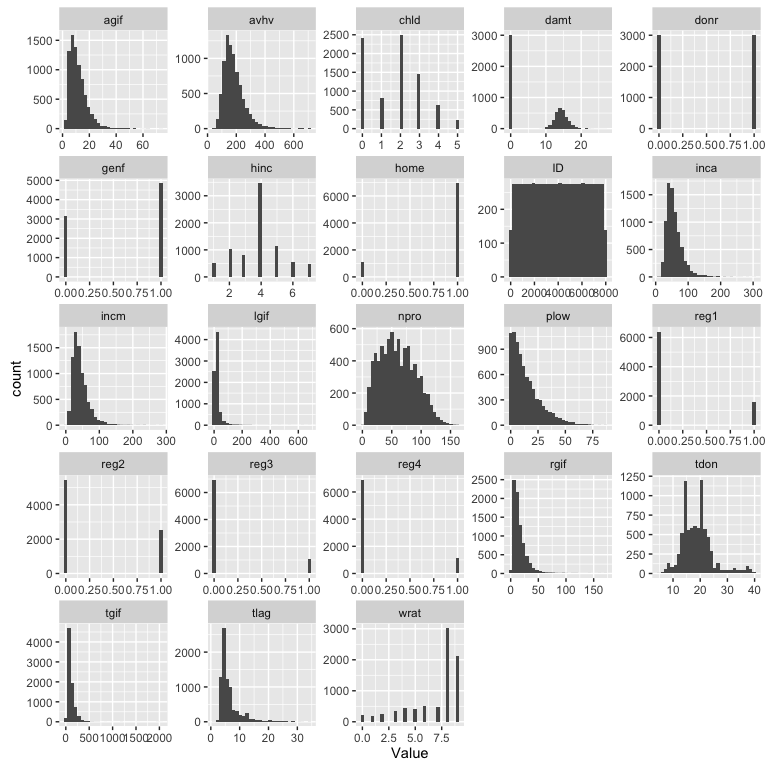
summary(charity.eda)

## ID reg1 reg2 reg3   
## Min. : 1 Min. :0.0000 Min. :0.000 Min. :0.0000   
## 1st Qu.:2003 1st Qu.:0.0000 1st Qu.:0.000 1st Qu.:0.0000   
## Median :4005 Median :0.0000 Median :0.000 Median :0.0000   
## Mean :4005 Mean :0.2004 Mean :0.319 Mean :0.1337   
## 3rd Qu.:6007 3rd Qu.:0.0000 3rd Qu.:1.000 3rd Qu.:0.0000   
## Max. :8009 Max. :1.0000 Max. :1.000 Max. :1.0000   
##   
## reg4 home chld hinc   
## Min. :0.0000 Min. :0.0000 Min. :0.000 Min. :1.000   
## 1st Qu.:0.0000 1st Qu.:1.0000 1st Qu.:0.000 1st Qu.:3.000   
## Median :0.0000 Median :1.0000 Median :2.000 Median :4.000   
## Mean :0.1395 Mean :0.8665 Mean :1.717 Mean :3.909   
## 3rd Qu.:0.0000 3rd Qu.:1.0000 3rd Qu.:3.000 3rd Qu.:5.000   
## Max. :1.0000 Max. :1.0000 Max. :5.000 Max. :7.000   
##   
## genf wrat avhv incm   
## Min. :0.0000 Min. :0.000 Min. : 48.0 Min. : 3.00   
## 1st Qu.:0.0000 1st Qu.:6.000 1st Qu.:133.0 1st Qu.: 27.00   
## Median :1.0000 Median :8.000 Median :169.0 Median : 38.00   
## Mean :0.6053 Mean :6.914 Mean :182.6 Mean : 43.47   
## 3rd Qu.:1.0000 3rd Qu.:9.000 3rd Qu.:217.0 3rd Qu.: 54.00   
## Max. :1.0000 Max. :9.000 Max. :710.0 Max. :287.00   
##   
## inca plow npro tgif   
## Min. : 12.00 Min. : 0.00 Min. : 2.00 Min. : 23.0   
## 1st Qu.: 40.00 1st Qu.: 4.00 1st Qu.: 36.00 1st Qu.: 63.0   
## Median : 51.00 Median :10.00 Median : 58.00 Median : 89.0   
## Mean : 56.43 Mean :14.23 Mean : 60.03 Mean : 113.1   
## 3rd Qu.: 68.00 3rd Qu.:21.00 3rd Qu.: 82.00 3rd Qu.: 137.0   
## Max. :305.00 Max. :87.00 Max. :164.00 Max. :2057.0   
##   
## lgif rgif tdon tlag   
## Min. : 3.00 Min. : 1.00 Min. : 5.00 Min. : 1.000   
## 1st Qu.: 10.00 1st Qu.: 7.00 1st Qu.:15.00 1st Qu.: 4.000   
## Median : 16.00 Median : 12.00 Median :18.00 Median : 5.000   
## Mean : 22.94 Mean : 15.66 Mean :18.86 Mean : 6.363   
## 3rd Qu.: 25.00 3rd Qu.: 20.00 3rd Qu.:22.00 3rd Qu.: 7.000   
## Max. :681.00 Max. :173.00 Max. :40.00 Max. :34.000   
##   
## agif donr damt part   
## Min. : 1.29 Min. :0.0000 Min. : 0.000 Length:8009   
## 1st Qu.: 6.97 1st Qu.:0.0000 1st Qu.: 0.000 Class :character   
## Median :10.23 Median :0.0000 Median : 0.000 Mode :character   
## Mean :11.68 Mean :0.4988 Mean : 7.209   
## 3rd Qu.:14.80 3rd Qu.:1.0000 3rd Qu.:14.000   
## Max. :72.27 Max. :1.0000 Max. :27.000   
## NA's :2007 NA's :2007

Both of our response variables, donr and damt contain 2007 NA's. We might want to interpolate these values or possibly filter them out completely.

## Histogram of each numeric variable

num\_vars <- names(charity.eda)[sapply(charity.eda, is.numeric)]  
  
charity.eda %>%   
 select(one\_of(num\_vars)) %>%   
 gather(Var,Value) %>%   
 mutate(Value=as.numeric(Value)) %>%   
 ggplot()+  
 aes(x=Value)+  
 geom\_histogram()+  
 facet\_wrap(~Var, scales="free")

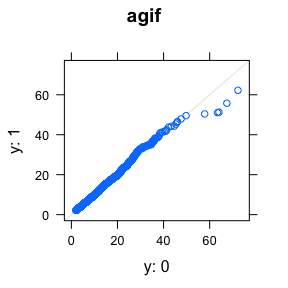
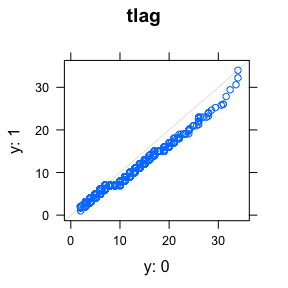
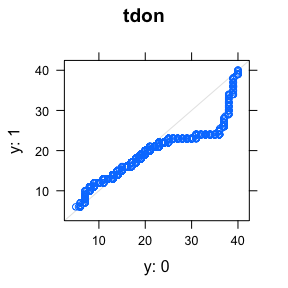
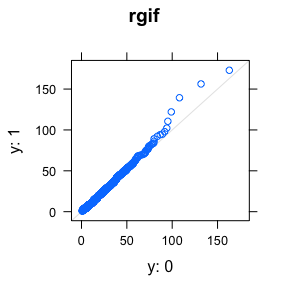
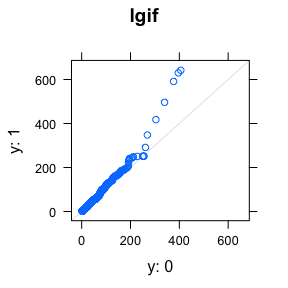
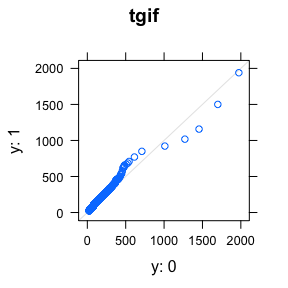
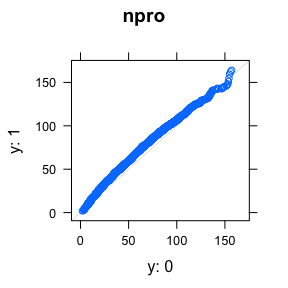
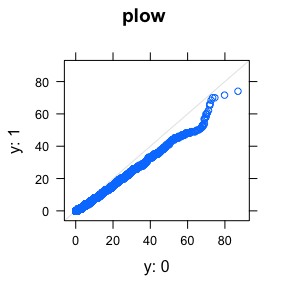
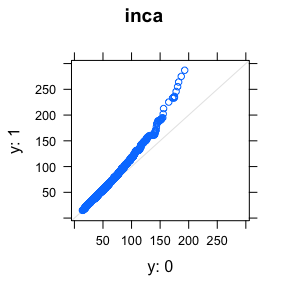
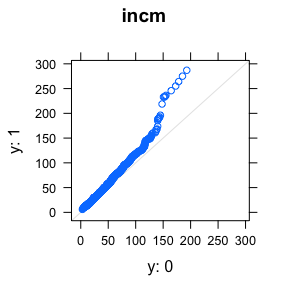
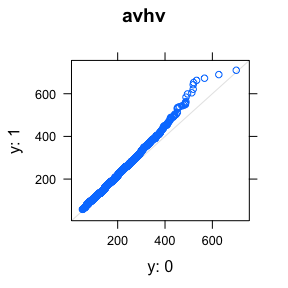
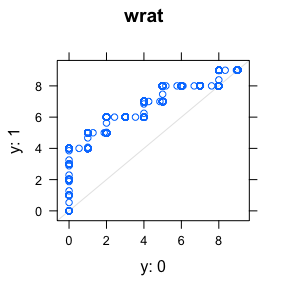
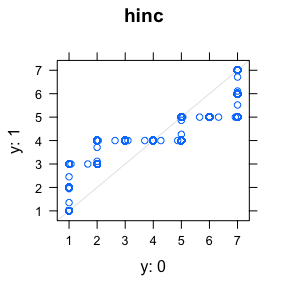
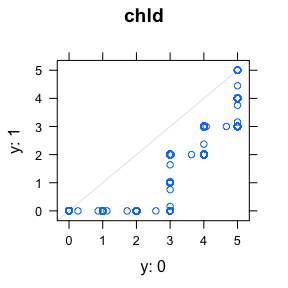
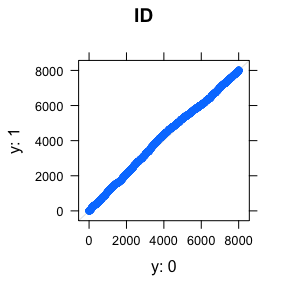


We observe the following:

* AGIF (Average dollar amount of gifts to date): Mostly normal with skew to right
* ID (data identifier): unique ID (remeber that the binwidth is ~30 in the histograms)
* reg1, reg2, reg3, reg4: Binary region indicators. Reg 3 and 4 have about 1/4th what reg 1 and 2 have. May need to overweight/undersample for those regions
* home (Homeowner): 0's are vastly underweighted here. We may need to apply under-sampling techniques
* chld (numebr of children): roughly normally distributed around the value 2, but with many zero values
* hinc (household income in seven categories): Category 4 is by far the most popular, which makes sense.
* genf (gender): sample is roughly 65% female
* wrat (Wealth Rating, 9 is highest 0 is lowest): Many 8's and 9's. This data is skewed towards wealthier people
* avhv (Average Home Value in $ thousands): Nearly normal with a mean of approx $150K, skewed right
* incm (Median Family Income in neighborhood in $ thousands): Nearly normal with a mean of approx $40K
* inca (Average family income in neighborhood in $ thousands): Nearly normal with a mean of approx $50K
* plow (% categorized as low income in neighborhood): long tailed, highest concentration around 0
* npro (total number of promotions received to date): normally distributed
* tgif (dollar amount total gifts to date): Normally distributed, but with outliers. may need to trim
* lgif (Dollar amount of largest gift to date): Normally distributed, but with outliers. may need to trim
* rgif (Dollar amount of most recent gift): Normally distributed, but with outliers. may need to trim
* tdon (Number of months since last donation): Nealry normal, but with some high concentrations around 15 and 21. Also skewed right
* tlag (Number of months between first and second gift): Skewed right
* agif (Average dollar amount of gifts to date): Nealry normal. Might be a good candidate to create a multiplier vs largest gift
* donr (Donor, binary): Response Variable. Evenly distributed
* damt (Donation in dollars): Response Variable. Zero-inflated distribution with an otherwise normal distribution

# QQ Plots of donr

ranges <- sapply(charity.eda,max)  
binary\_vars <- ranges==1  
continuous\_vars <- ranges>1  
cont\_select <- continuous\_vars[continuous\_vars==T]  
cont\_select <- names(na.omit(cont\_select))  
cont\_select <- cont\_select[1:15]  
  
  
for(i in 1:length(cont\_select)){  
 var <- charity.eda %>% select(one\_of(cont\_select[i]))  
 name <- names(var)  
 mod <- as.formula(paste0("donr~",name))  
 a <- qq(mod, data = charity.eda, main = name)  
 print(a)  
}



## Overlap of Region data?

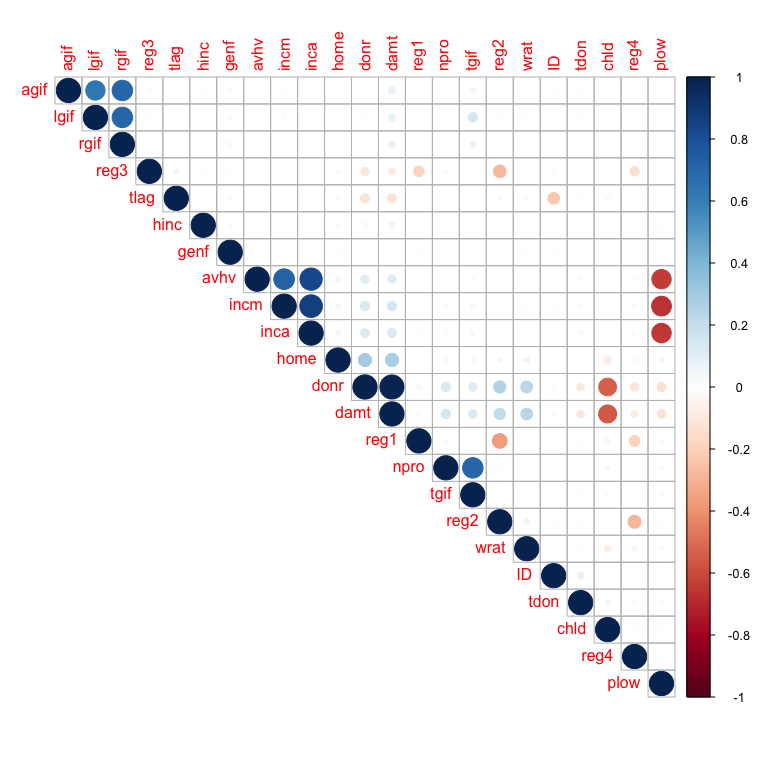
charity.eda %>%   
 group\_by(reg1,reg2,reg3,reg4) %>%   
 summarise(count=n())

## Source: local data frame [5 x 5]  
## Groups: reg1, reg2, reg3 [?]  
##   
## reg1 reg2 reg3 reg4 count  
## <int> <int> <int> <int> <int>  
## 1 0 0 0 0 1661  
## 2 0 0 0 1 1117  
## 3 0 0 1 0 1071  
## 4 0 1 0 0 2555  
## 5 1 0 0 0 1605

The region data is mutually exclusive

# Cross correlations?

M <- cor(charity.eda %>% select(one\_of(num\_vars)),use = "na.or.complete")  
corrplot(M, order = "hclust", type="upper")



Some auto-correlations between RGIF (Dollar amount of most recent gift) LGID (Last gift) and AGIF (average gift). Auto-correlations also between Median and Average Income and home prices. Inverse correlation between PLOT and the income variables.

# Do the NA Values in the reponse vars have different distributions?

the\_nas <- charity.eda %>% filter(is.na(donr))  
the\_complete <- charity.eda %>% filter(!is.na(donr))  
  
  
# full set represented  
dim(the\_nas)

## [1] 2007 24

dim(the\_complete)

## [1] 6002 24

dim(the\_nas)[1]+dim(the\_complete)[1]==nrow(charity.eda)

## [1] TRUE

na\_means <- suppressWarnings(sapply(the\_nas,mean))  
complete\_means <- suppressWarnings(sapply(the\_complete,mean))  
a<-as.data.frame(na\_means)  
a$rows <- rownames(a)  
b<-as.data.frame(complete\_means)  
b$rows <- rownames(b)  
c <- left\_join(a,b)  
c <- c %>% mutate(abs\_diff = abs(na\_means-complete\_means))  
c %>% arrange(-abs\_diff)

## na\_means rows complete\_means abs\_diff  
## 1 4083.0284006 ID 3978.9081973 1.041202e+02  
## 2 104.9058296 tgif 115.7995668 1.089374e+01  
## 3 56.0742402 npro 61.3543819 5.280142e+00  
## 4 178.8799203 avhv 183.9053649 5.025445e+00  
## 5 42.0548082 incm 43.9488504 1.894042e+00  
## 6 55.3477828 inca 56.7894035 1.441621e+00  
## 7 15.2745391 plow 13.8850383 1.389501e+00  
## 8 2.1155954 chld 1.5839720 5.316234e-01  
## 9 6.5879422 wrat 7.0231589 4.352167e-01  
## 10 19.0861983 tdon 18.7892369 2.969614e-01  
## 11 6.4967613 tlag 6.3185605 1.782009e-01  
## 12 22.8181365 lgif 22.9813396 1.632030e-01  
## 13 3.8176383 hinc 3.9390203 1.213821e-01  
## 14 0.2351769 reg2 0.3470510 1.118741e-01  
## 15 0.8126557 home 0.8845385 7.188278e-02  
## 16 0.1709018 reg3 0.1212929 4.960894e-02  
## 17 15.6851021 rgif 15.6539487 3.115346e-02  
## 18 0.1604385 reg4 0.1324558 2.798262e-02  
## 19 11.6896163 agif 11.6776658 1.195056e-02  
## 20 0.5979073 genf 0.6077974 9.890077e-03  
## 21 0.1973094 reg1 0.2014329 4.123439e-03  
## 22 NA donr 0.4988337 NA  
## 23 NA damt 7.2090970 NA  
## 24 NA part NA NA

# run a t-test, hypothesis is that both means are the same:  
t.test(c$na\_means,c$complete\_means)

##   
## Welch Two Sample t-test  
##   
## data: c$na\_means and c$complete\_means  
## t = 0.086337, df = 40.903, p-value = 0.9316  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## -500.4846 545.1847  
## sample estimates:  
## mean of x mean of y   
## 220.2875 197.9374

# fail to reject

The NAs in the repsonse variables have similar population means, therefore, it might be permissible to interpolate the responses.

# PARTITION THE DATA FOR CROSS VALIDATION

#Set-up Transformations  
  
  
charity.t <- charity  
charity.t$avhv <- log(charity.t$avhv)  
  
#Partition Data  
#Training  
data.train <- charity.t[charity$part=="train",]  
x.train <- data.train[,2:21]  
c.train <- data.train[,22] # donr  
n.train.c <- length(c.train) # 3984  
y.train <- data.train[c.train==1,23] # damt for observations with donr=1  
n.train.y <- length(y.train) # 1995  
  
#Validation  
data.valid <- charity.t[charity$part=="valid",]  
x.valid <- data.valid[,2:21]  
c.valid <- data.valid[,22] # donr  
n.valid.c <- length(c.valid) # 2018  
y.valid <- data.valid[c.valid==1,23] # damt for observations with donr=1  
n.valid.y <- length(y.valid) # 999  
  
#Testing  
data.test <- charity.t[charity$part=="test",]  
n.test <- dim(data.test)[1] # 2007  
x.test <- data.test[,2:21]  
  
x.train.mean <- apply(x.train, 2, mean)  
x.train.sd <- apply(x.train, 2, sd)  
x.train.std <- t((t(x.train)-x.train.mean)/x.train.sd) # standardize to have zero mean and unit sd  
apply(x.train.std, 2, mean) # check zero mean

## reg1 reg2 reg3 reg4 home   
## 2.151811e-17 -2.526099e-17 3.693258e-17 -6.017778e-17 -9.663428e-18   
## chld hinc genf wrat avhv   
## -2.051129e-17 -1.463197e-17 4.465563e-17 -1.062688e-16 -3.616154e-16   
## incm inca plow npro tgif   
## -1.335981e-16 8.260581e-17 4.848389e-17 -6.655491e-17 2.532023e-17   
## lgif rgif tdon tlag agif   
## 2.669731e-17 5.429393e-18 -1.835154e-16 1.010477e-16 -1.258480e-16

apply(x.train.std, 2, sd) # check unit sd

## reg1 reg2 reg3 reg4 home chld hinc genf wrat avhv incm inca plow npro tgif   
## 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1   
## lgif rgif tdon tlag agif   
## 1 1 1 1 1

data.train.std.c <- data.frame(x.train.std, donr=c.train) # to classify donr  
data.train.std.y <- data.frame(x.train.std[c.train==1,], damt=y.train) # to predict damt when donr=1  
  
x.valid.std <- t((t(x.valid)-x.train.mean)/x.train.sd) # standardize using training mean and sd  
data.valid.std.c <- data.frame(x.valid.std, donr=c.valid) # to classify donr  
data.valid.std.y <- data.frame(x.valid.std[c.valid==1,], damt=y.valid) # to predict damt when donr=1  
  
x.test.std <- t((t(x.test)-x.train.mean)/x.train.sd) # standardize using training mean and sd  
data.test.std <- data.frame(x.test.std)