Upward Mobility - model validation

Alex Haase

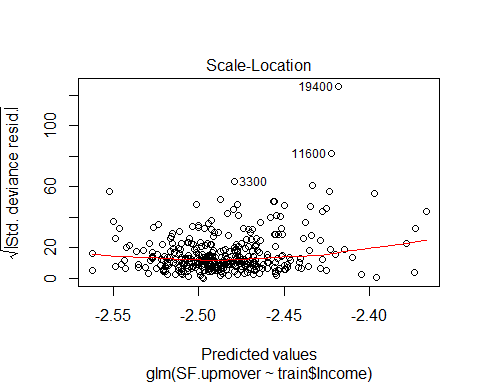
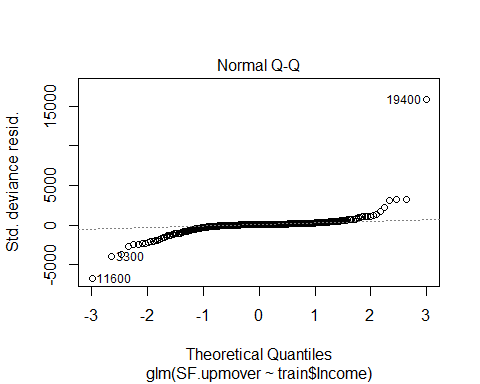
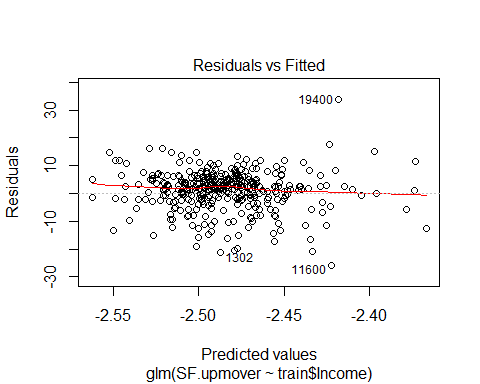
rm(list=ls()) # clear global environment  
getwd() # double check wd

mobility = read.csv('http://dept.stat.lsa.umich.edu/~bbh/s485/data/Mobility.csv', row.names = 1)  
  
#\*\*\*\*\*\*\*\*\*\*\*\* TRAIN DATA \*\*\*\*\*\*\*\*\*\*\*\*  
  
#1.)   
  
# 50% of the sample size - test sample

sample\_size = floor(0.50 \* nrow(mobility))  
  
# set the seed to make the partition reproductible  
set.seed(123)  
train\_ind = sample(seq\_len(nrow(mobility)), size = sample\_size)  
  
train = mobility[train\_ind, ]  
test = mobility[-train\_ind, ]  
  
#nrow(train) #370 rows  
#nrow(test) #371 rows  
  
#nrow(train$N.child)  
#nrow(test$N.child)  
  
write.csv(test, file="testdata.csv", row.names=T) # creates file: testdata.csv which is a separate file for the test data  
  
#2.)   
#model.frame(fitted\_model)  
# produces a data frame for the model without missing values  
  
Income\_traindata = model.frame(train$p.upmover~train$Income)  
Gini\_traindata = model.frame(train$p.upmover~train$Gini)  
Seg\_poverty\_traindata = model.frame(train$p.upmover~train$Seg\_poverty)  
Seg\_racial\_traindata = model.frame(train$p.upmover~train$Seg\_racial)  
Seg\_income\_traindata = model.frame(train$p.upmover~train$Seg\_income)  
  
  
#Likelihood Ratio Test  
#choosing significance: 0.05  
  
SF.upmover = with(train, N.child \* prop.lowstart \* cbind(S=train$p.upmover, F=(1-train$p.upmover)))  
SF.upmover = round(SF.upmover)  
  
fit1 = glm(SF.upmover ~ 1, family = binomial('logit'), data = train)  
fit2 = glm(SF.upmover ~ train$Income, family = binomial('logit'), data = train)  
fit3 = glm(SF.upmover ~ train$Income \* train$Gini, family = binomial('logit'), data = train)  
fit4 = glm(SF.upmover ~ train$Income \* train$Gini \* train$Seg\_poverty, family = binomial('logit'), data = train)  
fit5 = glm(SF.upmover ~ train$Income \* train$Gini \* train$Seg\_poverty \* train$Seg\_racial, family = binomial('logit'), data = train)  
fit6 = glm(SF.upmover ~ train$Income \* train$Gini \* train$Seg\_poverty \* train$Seg\_racial \* train$Seg\_income, family = binomial('logit'), data = train)  
  
anova(fit1, fit2, fit3, fit4, fit5, test = "LRT")

## Analysis of Deviance Table  
##   
## Model 1: SF.upmover ~ 1  
## Model 2: SF.upmover ~ train$Income  
## Model 3: SF.upmover ~ train$Income \* train$Gini  
## Model 4: SF.upmover ~ train$Income \* train$Gini \* train$Seg\_poverty  
## Model 5: SF.upmover ~ train$Income \* train$Gini \* train$Seg\_poverty \*   
## train$Seg\_racial  
## Resid. Df Resid. Dev Df Deviance Pr(>Chi)   
## 1 362 18944   
## 2 361 18734 1 209.9 < 2.2e-16 \*\*\*  
## 3 359 17195 2 1539.9 < 2.2e-16 \*\*\*  
## 4 355 13457 4 3737.8 < 2.2e-16 \*\*\*  
## 5 347 10456 8 3000.6 < 2.2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

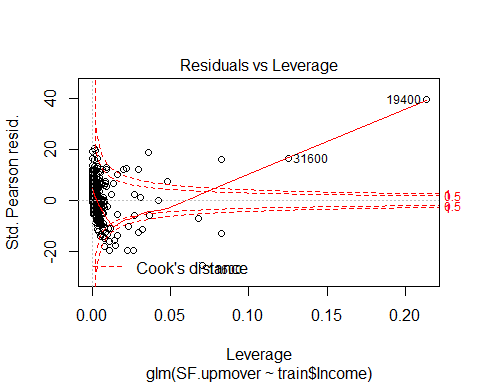
#model with the best fit:  
#fit 2: Income with Gini  
  
#Generate Residual Plots  
plot(fit2)



#Modeling with natural spline bases  
#install.packages("dplyr")  
#install.packages("Rcpp")  
#install.packages("DAAG")  
dat0 = dplyr::filter(train)  
#ns\_mods = paste("SF.upmover ~ ns(train$Income, df=", 1:4, ")")  
  
library(splines)  
library(DAAG)

## Warning: package 'DAAG' was built under R version 3.4.4

## Loading required package: lattice



library(MASS)

##   
## Attaching package: 'MASS'

## The following object is masked from 'package:DAAG':  
##   
## hills

library(boot)

## Warning: package 'boot' was built under R version 3.4.2

##   
## Attaching package: 'boot'

## The following object is masked from 'package:lattice':  
##   
## melanoma

crossval = cv.glm(train, fit2)

anova(fit2, crossval, test = "LRT")

## Analysis of Deviance Table  
##   
## Model: binomial, link: logit  
##   
## Response: SF.upmover  
##   
## Terms added sequentially (first to last)  
##   
##   
## Df Deviance Resid. Df Resid. Dev Pr(>Chi)   
## NULL 362 18944   
## train$Income 1 209.88 361 18734 < 2.2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

#SF.upmover and train$Income  
#SF.upmover  
#train$Income #length = 370  
length(train$Income)

## [1] 370

dim(SF.upmover) #dim = 370 2

## [1] 370 2

#SF.upmover[,'S']  
#SF.upmover[,'F']  
  
#using R spline() function  
splineOutput = spline(x=SF.upmover[,'S'], y=train$Income)  
  
#splineOutput$x #length = 792  
#splineOutput$y # length = 792  
  
#appending the above spline outputs to fit train data length  
spline1 = splineOutput$x[2:371]  
spline2 = splineOutput$y[2:371]  
  
class(spline1)

## [1] "numeric"

class(SF.upmover[,'S'])

## [1] "numeric"

fitSx = glm(SF.upmover ~ 1, family = binomial('logit'), data = train)  
fitSy = glm(SF.upmover ~ train$Income, family = binomial('logit'), data = train)  
  
anova(fitSx, fitSy, test = "LRT")

## Analysis of Deviance Table  
##   
## Model 1: SF.upmover ~ 1  
## Model 2: SF.upmover ~ train$Income  
## Resid. Df Resid. Dev Df Deviance Pr(>Chi)   
## 1 362 18944   
## 2 361 18734 1 209.88 < 2.2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

#\*\*\*\*\*\*\*\*\*\*\*\* TEST DATA \*\*\*\*\*\*\*\*\*\*\*\*  
  
testdata = read.csv("testdata.csv", TRUE, ",")  
#testdata  
   
#hand selected: Commute and Local\_gov\_spending  
  
Commute\_testdata = model.frame(testdata$p.upmover~testdata$Commute)  
Local\_gov\_spending\_testdata = model.frame(testdata$p.upmover~testdata$Local\_gov\_spending)  
  
# LR-test  
#choosing significance: 0.05  
  
testdata = na.omit(testdata)  
p.upmover2 = na.omit(testdata$p.upmover)  
Commute = na.omit(testdata$Commute)  
Local\_gov\_spending = na.omit(testdata$Local\_gov\_spending)  
  
  
SF.upmover2 = with(testdata, N.child \* prop.lowstart \* cbind(S=p.upmover2, F=(1-p.upmover2)))  
SF.upmover2 = round(SF.upmover2)  
dim(SF.upmover2)

## [1] 205 2

fit1b = glm(SF.upmover2 ~ 1, family = binomial('logit'), data = testdata)  
fit2b = glm(SF.upmover2 ~ Commute, family = binomial('logit'), data = testdata)  
fit3b = glm(SF.upmover2 ~ Commute \* Local\_gov\_spending, family = binomial('logit'), data = testdata)  
  
logit\_mod = anova(fit1b, fit2b, fit3b, test = "LRT")  
logit\_mod

## Analysis of Deviance Table  
##   
## Model 1: SF.upmover2 ~ 1  
## Model 2: SF.upmover2 ~ Commute  
## Model 3: SF.upmover2 ~ Commute \* Local\_gov\_spending  
## Resid. Df Resid. Dev Df Deviance Pr(>Chi)   
## 1 204 12908   
## 2 203 11781 1 1127.58 < 2.2e-16 \*\*\*  
## 3 201 10965 2 816.21 < 2.2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

#each of the variables were significant  
#look at coefficients  
#newfit = update(logit\_mod, data = testdata)  
#coeffs = coef(summary(newfit))  
#coeffs[, "Pr(>|z|)"] = p.adjust(coeffs[, "Pr(>|z|)"], "holm")  
#coeffs