Zhaoxing Wu
Stat 443
Homework 4

Many R packages are designed to create tree and forest models. For example, "rpart" can create tree models, and "party" can create tree and forest models by using `ctree()` and `cforest()` functions. In the consumer expenditure survey data, we want to estimate the population mean interest and dividend (INTRDVX) using those R packages. To do that, we first need to remove all the non-excluded variables from the data set according to the description file. After reading in the modified dataset into R, we need to edit the flag variable of INTRDVX. If the value of INTRDVX_ is "don't know", refusal, or other types of non-response, we change it to 0; if the value of INTRDVX_ is valid data value, or topcoded value, we change it to 1. In this way, it is easier to estimate the probability that the variable of interest is non-missing. Then we can implement different statistical methods to estimate INTRDVX with sampling weight FINLWT_21.

First, we want to use the IPW weighing method to estimate mean of INTRDVX with `rpart()` classification tree. When using `rpart()` , it is necessary to set the method as "class" and the variable of INTRDVX_ as factor for classification tree. Then we fit a classification tree of INTRDVX_ with all other variables except INTRDVX and FINLWT_21. When using `predict()` on the tree model, we get one column of the estimated probability of missing INTRDVX and another column of the estimated probability of non-missing INTRDVX. The second column is the $\widehat{\pi_i}$ we want to get. Then using the IPW formula with the values calculated, we get the result 4549.218.

Second, we want to use the imputation method to estimate mean of INTRDVX with `rpart()` regression tree. Different from the classification tree, it is necessary to set the method as "anova" to tell R this is for regression. Then we fit a regression tree of INTRDVX with all other variables except INTRDVX_ and the sampling weight is also FINLWT_21. After getting the regression tree model, we use `predict()` on it to get the imputed value of the non-missing INTRDVX. Then apply the formulate of imputation, we get the result 3996.971.

Lastly, we want to use the IPW weighing method to estimate mean of INTRDVX with `ctree()` and `cforest()` classification tree. Similar to the first one, we need to as the variable of INTRDVX_ as factor variables, and then fit a classification tree or forest of INTRDVX_ with all other variables except INTRDVX and FINLWT_21. When using `predict()` on the tree or forest model, we need to set the type to "prob" to get probability of missing and non-missing INTRDVX. It returns a list of 4693 lists of probability of missing and non-missing INTRDVX for each INTRDVX. Then using the IPW formula with the values calculated, we get 4445.513 for the tree model, and 4680.598 for the forest model.

(R code is attached in the second page)

```r
library(rpart)
vartype <- rep("numeric",638)
# variable= "C"
vartype[c(1,2,4,6,8,10,12,14,16,18,20,22,23,24,25,26,27,29,31,33,34,35,37,39,
41,43,45,47,49,52,54,56,58,60,62,64,66,68,70,72,73,74,75,76,77,78,79,
80,82,84,86,88,89,90,92,94,96,97,98,99,100,102,104,106,108,109,110,
111,112,113,114,115,116,118,119,120,122,123,124,125,126,128,130,131,
132,133,135,137,139,303,304,305,306,307,308,309,310,311,312,313,314,
315,316,317,318,319,321,323,325,331,333,407,409,410,411,453,454,456,
458,460,462,464,465,466,467,468,470,472,474,476,477,478,479,482,484,
486,488,490,492,494,496,497,498,499,500,502,504,506,508,510,512,514,
516,518,520,522,524,526,528,530,532,534,536,538,540,542,544,546,548,
550,552,554,556,558,560,562,564,566,568,570,572,574,576,578,580,582,
584,585,586,588,590,592,594,596,598,600,602,604,606,608,610,612,614,
616,618,620,622,624,626,628,630,632,635,637)] <- "factor"
# colClasses: use vartype to identify is it a factor variable or a numeric va
riable
z <- read.table("subset.txt",header=TRUE,colClasses=vartype)
tmp <- rep(NA,nrow(z))
#in order to predict the non-missing value
#set C as 0, and D,T as 1
tmp[z$INTRDVX_ == "C"] <- 0
tmp[z$INTRDVX_ == "D" | z$INTRDVX_ == "T"] <- 1
z$INTRDVX_ <- tmp ### convert INTRDVX to binary variable
z$INTRDVX_ = as.factor(z$INTRDVX_)
w <- z$FINLWT21
y <- z$INTRDVX
gp <- !is.na(y)

### classification tree without INTRDVX and FINLWT21
rp_class <- rpart(INTRDVX_ ~ . - INTRDVX - FINLWT21, data=z, method="class")
p <- predict(rp_class) ### predicted prob(INTRDVX_ = 1)
ipw <- sum(w[gp]*y[gp]/p[, 2][gp])/sum(w[gp]/p[,2][gp])
print(ipw)

## [1] 4549.218

### regression tree without INTRDVX and FINLWT21
rp_reg <- rpart(INTRDVX ~ . - as.numeric(INTRDVX_), weight=FINLWT21, data=z,
method="anova")
miss <- is.na(y) ## obs with missing INTRDVX
yhat <- predict(rp_reg, newdata=z)
popmean <- (sum(w[!miss]*y[!miss])+sum(w[miss]*yhat[miss]))/sum(w)
print(popmean)

## [1] 3996.971

library(party)

### classification tree without INTRDVX and FINLWT21
ct <- ctree(INTRDVX_ ~ . - INTRDVX - FINLWT21, data=z)
```

```
p <- predict(ct, type = "prob")
a = c()
for (i in 1:length(p)){
  a = c(a, p[[i]][2])
}
p = a
ipw <- sum(w[gp]*y[gp]/p[gp])/sum(w[gp]/p[gp])
print(ipw)
```

```
## [1] 4445.513
```

```
cf <- cforest(INTRDVX_ ~ . - INTRDVX - FINLWT21, data=z)

p <- predict(cf,newdata=z, type = "prob")

a = c()
for (i in 1:length(p)){
  a = c(a, p[[i]][2])
}
p = a
w <- z$FINLWT21
y <- z$INTRDVX
gp <- !is.na(y)
ipw <- sum(w[gp]*y[gp]/p[gp])/sum(w[gp]/p[gp])
print(ipw)
```

```
## [1] 4680.598
```