Data Analytics Lab 6

06/11/2018

Two-stage modelling to impute a variable that can be positive or zero

You are required to follow the example and R codes in pager 537 and 538 to impute the variable "earnings", which could be either positive or zero, assuming no logical imputation based on number of work hours is possible.

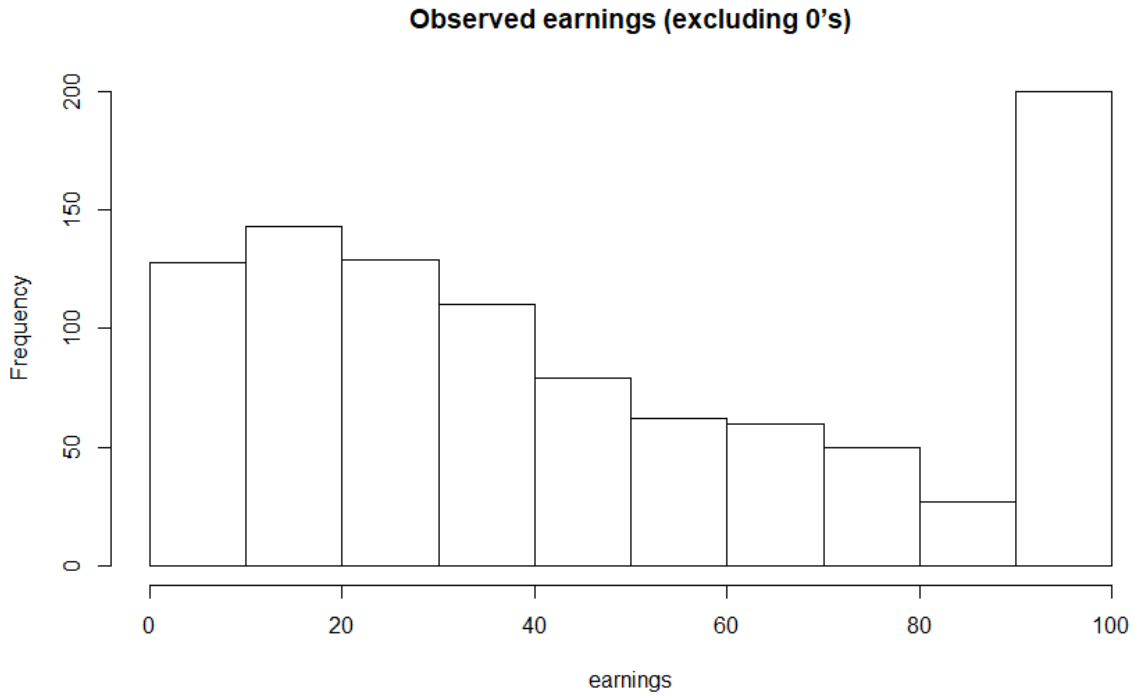
Please be aware that the headings for some variables in the text book and in the dataset might not be exactly the same.

R Code :

```
install.packages("ggplot2")
library(ggplot2)
Data <- read.csv("siswave3v4impute3.csv",  header=TRUE, sep=",")
View(Data)
attach(Data)
n <- nrow (Data)

# earnings variables:
# rearn:  respondent's earnings
# tearn:  spouse's earnings
# set up some simplified variables to work with
na.fix <- function (a) {
  ifelse (a<0 | a==999999, NA, a)
}
earnings <- na.fix(rearn) + na.fix(tearn)
earnings <- earnings/1000
#########################################################################################
#Missing data in R and bugs from Pg 529
cbind (Data$sex, Data$race, Data$educ_r, Data$r_age, earnings, Data$police)[91:95,]

#########################################################################################
#random imputation of single variable  - earnings Pg 534
random.imp <- function (a){
  missing <- is.na(a)
  n.missing <- sum(missing)
  a.obs <- a[!missing]
  imputed <- a
  imputed[missing]  <- sample (a.obs, n.missing, replace=TRUE)
  return (imputed)
}

earnings.imp <- random.imp (earnings)

#Zero coding or topcoding
topcode <- function (a, top){
  return (ifelse (a>top, top, a))
}
earnings.top <- topcode (earnings, 100) # earnings are in $thousands topcoded to 100
```

#Pg 534 fig. 25.1 a

hist (earnings.top[earnings>0], xlab = "earnings", main = "Observed earnings (excluding 0's)")



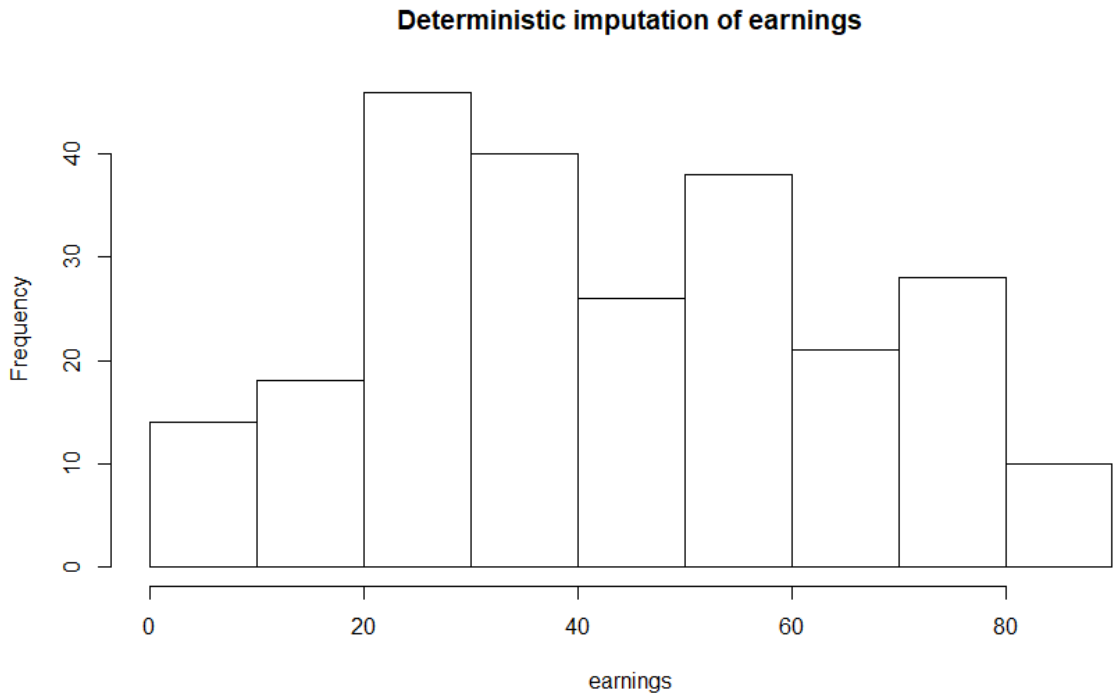
###

#Deterministic imputation of single variable - earnings Pg 535

```
#calculate each variable
white <- ifelse (race==1, 1, 0)
white[is.na(race)] <- 0
male <- ifelse (sex==1, 1, 0)
over65 <- ifelse (r_age>65, 1, 0)
immig[is.na(immig)] <- 0
educ_r[is.na(educ_r)] <- 2.5
workhrs.top <- topcode (workhrs, 40)
is.any <- function (a) {
 any.a <- ifelse (a>0, 1, 0)
 any.a[is.na(a)] <- 0
 return(any.a)
}

workmos <- workmos
earnings[workmos==0] <- 0
any.ssi <- is.any (ssi)
any.welfare <- is.any (welfare)
any.charity <- is.any (charity)

#setting up a data frame with all the variables we shall use in our analysis:
sis <- data.frame (cbind (earnings, earnings.top, male, over65, white,
                  immig, educ_r, workmos, workhrs.top, any.ssi, any.welfare, any.charity))

#fit a regression to positive values of earnings
lm.imp.1 <- lm (earnings ~ male + over65 + white + immig + educ_r +
```

```
        workmos + workhrs.top + any.ssi + any.welfare + any.charity,
        data=sis, subset=earnings>0)
#predictions for the data
pred.1 <- predict (lm.imp.1, sis)

#imputing predictions into missing values
impute <- function (a, a.impute){
  ifelse (is.na(a), a.impute, a)
}
#compute missing earnings
earnings.imp.1 <- impute (earnings, pred.1)
View(earnings.imp.1)
#transforming and top coding
lm.imp.2.sqrt <- lm (I(sqrt(earnings.top)) ~ male + over65 + white +
            immig + educ_r + workmos + workhrs.top + any.ssi + any.welfare +
             any.charity, data=sis, subset=earnings>0)


pred.2.sqrt <- predict (lm.imp.2.sqrt, sis)
pred.2 <- topcode (pred.2.sqrt^2, 100)
earnings.imp.2 <- impute (earnings.top, pred.2)


#as tabulated on Pg 536
summary(lm.imp.2.sqrt)

Call:
lm(formula = I(sqrt(earnings.top)) ~ male + over65 + white +
    immig + educ_r + workmos + workhrs.top + any.ssi + any.welfare +
    any.charity, data = sis, subset = earnings > 0)

Residuals:
    Min      1Q  Median      3Q     Max
-6.6140 -1.3569 -0.0302  1.3124  6.3782

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.667735   0.439786  -3.792 0.000159 ***
male         0.318698   0.130479   2.443 0.014761 *
over65      -1.435181   0.582410  -2.464 0.013903 *
white        0.959027   0.151147   6.345 3.40e-10 ***
immig       -0.616299   0.135138  -4.561 5.75e-06 ***
educ_r       0.788973   0.066665  11.835  < 2e-16 ***
workmos      0.326720   0.030532  10.701  < 2e-16 ***
workhrs.top  0.057996   0.009261   6.262 5.68e-10 ***
any.ssi     -0.973930   0.554129  -1.758 0.079131 .
any.welfare -1.351159   0.367475  -3.677 0.000249 ***
any.charity -1.171515   0.601405  -1.948 0.051705 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.964 on 977 degrees of freedom
  (241 observations deleted due to missingness)
Multiple R-squared:  0.4364,      Adjusted R-squared:  0.4307
F-statistic: 75.66 on 10 and 977 DF,  p-value: < 2.2e-16
```

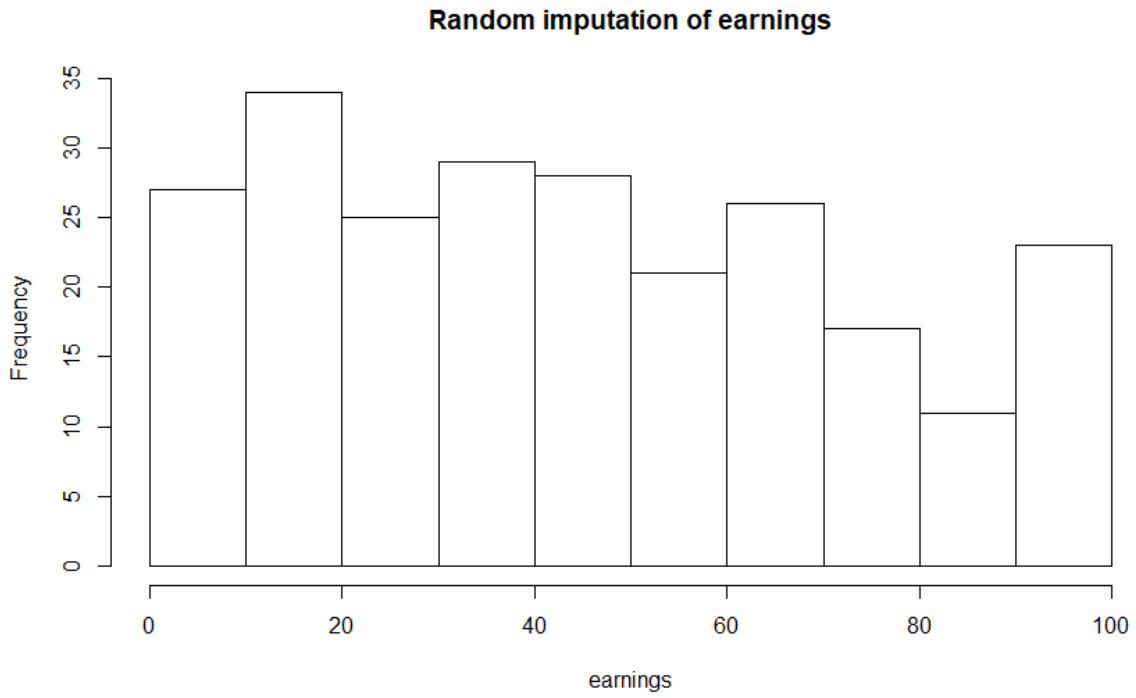
```
#Plot deterministic imputation of earnings, Pg 534 fig. 25.1b
hist (earnings.imp.2[is.na(earnings)], xlab = "earnings", main = "Deterministic imputation of
earnings")

#plot using ggplot2 just for practice
frame2 = data.frame(earnings = earnings.imp.2[is.na(earnings)])
p2 <- ggplot(frame2,aes(earnings)) +
 geom_histogram(colour = "black", fill = "white",binwidth=7) +
 theme_bw() +
 labs(title="Deterministic imputation of earnings")
plot(p2)
```




```
################################################################################

#?rnorm
## Random regression imputation
pred.4.sqrt <- rnorm (n, pred.2.sqrt, sigma(lm.imp.2.sqrt))
pred.4 <- topcode (pred.4.sqrt^2, 100)
earnings.imp.4 <- impute (earnings.top, pred.4)
```

#Plot random imputation of earnings, Pg 534 fig. 25.1c

hist (earnings.imp.4[is.na(earnings)], xlab = "earnings", main = "Random imputation of earnings")



##

#Two-stage modeling to impute a variable that can be positive or zero, Pg 538

#Applying generalized linear model

```
glmfit <- glm (I(earnings>0) ~ male + over65 + white +
                 immig + educ_r + any.ssi + any.welfare + any.charity,
               data=sis, family=binomial(link=logit))
summary(glmfit)
lm.ifpos.sqrt <- lm (I(sqrt(earnings.top)) ~ male + over65 + white +
                     immig + educ_r + any.ssi + any.welfare + any.charity,
                   data=sis, subset=earnings>0) # (same as lm.imp.2 from above)
View(lm.ifpos.sqrt)

#?rbinom
#impute whether missing earnings are positive
pred.sign <- rbinom (n, 1, predict(glmfit, type = "response"))
pred.pos.sqrt <- rnorm (n, predict (lm.ifpos.sqrt, sis),
                        sigma(lm.ifpos.sqrt))
#then impute the earnings themselves
pred.pos <- topcode (pred.pos.sqrt^2, 100)
earnings.imp <- impute (earnings, pred.sign*pred.pos)
```