MLE : Mid-term

Jinhee Kwak

March 17, 2015

Discussion

I write a function for an OLS regression. Through applying the function to three different types of data in terms of missing value, I present each results of them with coefficients plotting and tables. When conpared the results of model of data (Figure 1) with those of the others, I find that the standard errors of dataMiss and Ameliadata appear smaller than those of the data. It shows that removing missing values (dataMiss, Figure 2) or missing value estimation (Amelia, Figure 3) have some possibilities that its standard errors could be underestimated, even though the results may look better than before treatment of missing data such as the lower estimates of coefficients. It means that uncertainty generated by missing data may not be considered. At this point, the need to make use of MLE arises. In short, the midterm assignment asks questions in a context of how to deal with missing data and its implications.

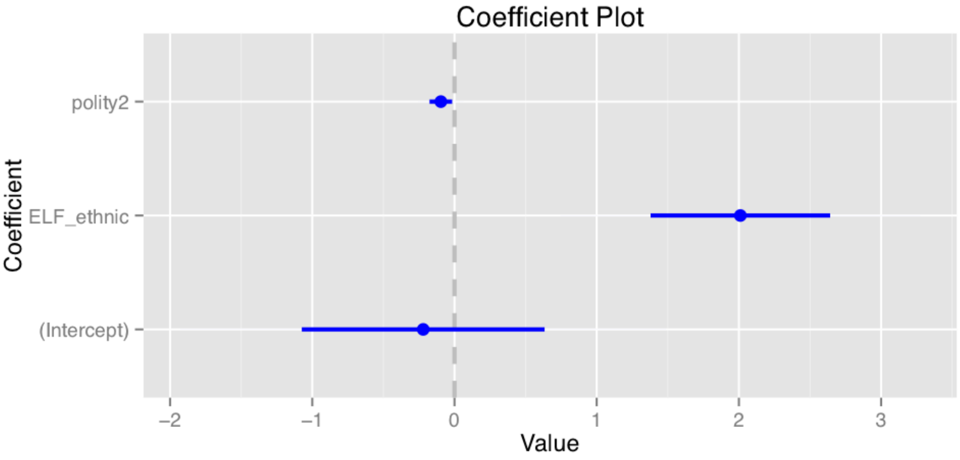

Figure 1: Model Coefficients Plot

	var	Estimate	Std.Error	T-Statistic	P-Value	Lower 95 CI	Upper 95 CI
intercept	intercept	-0.22	0.85	-0.26	1.00	-1.89	1.45
Elf-ethnic	elf-ethnic	2.01	0.63	3.18	0.00	0.77	3.25
polity2	polity2	-0.10	0.08	-1.20	0.00	-0.25	0.06

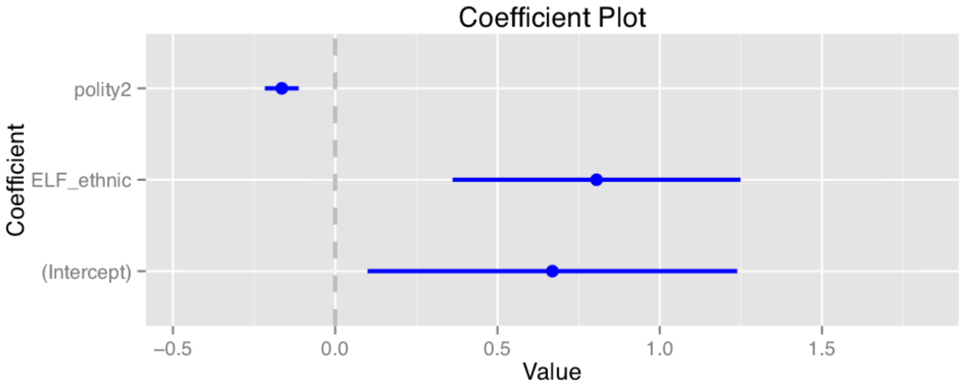

Figure 2: ModelListDel Coefficients Plot

	var	Estimate	Std.Error	T-Statistic	P-Value	Lower 95 CI	Upper 95 CI
intercept	intercept	0.67	0.57	1.18	0.00	-0.45	1.79
Elf-ethnic	elf-ethnic	0.81	0.44	1.82	0.00	-0.06	1.67
polity2	polity2	-0.16	0.05	-3.16	0.00	-0.27	-0.06

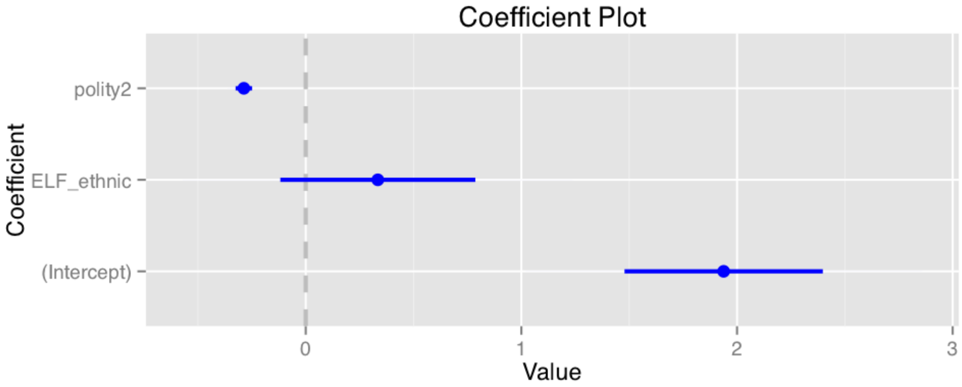

Figure 3: ModelAmelia Coefficients Plot

	var	Estimate	Std.Error	T-Statistic	P-Value	Lower 95 CI	Upper 95 CI
intercept	intercept	1.94	0.46	4.22	0.00	1.04	2.84
Elf-ethnic	elf-ethnic	0.33	0.45	0.74	0.00	-0.55	1.22
polity2	polity2	-0.29	0.04	-7.49	0.00	-0.36	-0.21

```
###################################################
# Set up workspace
rm(list=ls())
setwd("/Users/paxhistory/Dropbox/Duke 2015 Spring term/MLE lab/lab7")
load("midTermData.rda")
set.seed(6886)

# Function to load packages
loadPkg=function(toLoad){
  for(lib in toLoad){
  if(! lib %in% installed.packages()[,1])
    { install.packages(lib, repos='http://cran.rstudio.com/') }
        suppressMessages( library(lib, character.only=TRUE) ) }
}

# Load libraries
packs=c('foreign', 'lmtest', 'sandwich', 'Amelia', 'sbgcop')
loadPkg(packs)
###################################################

ols=function(formula,data,impute=FALSE){

  if(impute==TRUE){
    set.seed(6886)
    data=amelia(x=data,m=1)
    data=data$imp$imp1}
    data=data[complete.cases(data),]

    # Retrieve vars from formula input
    dv = all.vars(form)[1]
    ivs = all.vars(form)[ 2:length(all.vars(form)) ]
    # Create matrix with column for intercept and
    ## data from independent variables
    y = data[,dv]
    x = data.matrix(cbind(1, data[,ivs]))
    # General parameters
    n = nrow(x) # Number of observations
    p = length(ivs) # Number of parameters
    df=n-p-1# degrees of freedom

    #coefficient

    # Beta = (X'X)^-1 %*% X'Y
    # calculating X'X
    xTx = t(x) %*% x

    # calculating X'y
    xTy = t(x) %*% y

    # calculating Beta
    beta = solve(xTx) %*% xTy

    #standard errors
    #se(Beta) = sqrt( sigma^2 * (X'X)^-1 )
    # First lets calculate our yhat
    yhat=x%*%beta
    # First get out residuals
    e=y-yhat
    # calculating e'e (sum of squared residuals): assuming homoskedasticity
    # Adjust by degrees of freedom
    sigma2 = sum(e^2)/df
    varcov = sigma2*solve(xTx)
    # Pull out the standard errors for the coefficient estimates
    se = sqrt(diag(varcov))
    # Calculate t values
    tval = beta/se
    # Calculate p-values
    pval = round(2*pt(abs(tval),df, lower.tail=FALSE))
    up95=beta+qnorm(0.975)*se
    lo95=beta-qnorm(0.975)*se
    # R squared
    ssReg = sum((yhat-mean(y))^2)
    ssTot = sum((y-mean(y))^2)
    R2 = ssReg/ssTot
    # F statistic
    msReg = sum((yhat-mean(y))^2)/p
    msRes = sum(e^2)/df
    Fstat = round(msReg/msRes,3)
    Fpval = round(pf(Fstat,p,df,lower.tail=F),3)
    resul=paste("F-statistic:",Fstat,"on",p,"and",df,"DF,","p-value:",Fpval,sep=' ')
    # creating matrix
    coef=cbind(beta, se, tval, pval,lo95, up95)
    colnames(coef)=c("Estimate","Std.Error","T-Statistic","P-Value",
                     "Lower 95% CI","Upper 95% CI")
    rownames(coef)=c("intercept", "Elf-ethnic","polity2")
    colnames(varcov)=c("intercept","Elf-ethnic","polity2")
    rownames(varcov)=c("intercept","Elf-ethnic","polity2")
    a<-list("coefficients"=coef, "varcov"=varcov, "Rsq"=R2, "Fstat"=resul)
    return(a)
  }


# First set a seed

set.seed(6886)

# Set up the model formula

form = formula(gini_net_std ~ ELF_ethnic + polity2)

# Run the various models

model = ols(formula=form, data=data)
a<-lm(gini_net_std ~ ELF_ethnic + polity2, data=data)
summary(a)
```

```r
modelListDel = ols(formula = form, data=dataMiss)
b<-lm(gini_net_std ~ ELF_ethnic + polity2, data=dataMiss)
summary(b)

modelAmelia = ols(formula = form, data=dataMiss, impute=TRUE)


set.seed(6886)
Ameliadata=amelia(x=dataMiss,m=1)$imp$imp1
c<-lm(gini_net_std ~ ELF_ethnic + polity2, data=Ameliadata )
summary(c)
```

```r
library(coefplot)
library(xtable)
library(ggplot2)
coefplot(a)
```

```r
coefplot(b)
```

```r
coefplot(c)
```