ECON613 HW2

Zhilin Tang, zt53

2022-02-04

```
library(ggplot2)
library(gridExtra)
library(dplyr)
library(data.table)
library(tinytex)
```

Exercise 1 OLS estimate

```
datind2009 = data.frame(fread(paste('datind2009.csv',sep=''),header=TRUE))
df1 = datind2009[,c('empstat','wage','age')]
df1 = df1[which(complete.cases(df1) & df1$wage!=0),]
```

1.1 Calculate the correlation between Y and X

```
Y = matrix(df1$wage)
X = matrix(cbind(rep(1,length(df1$age)),df1$age),ncol=2)
cov_XY = mean(df1$age*df1$wage)-mean(df1$age)*mean(df1$wage)
var_X = mean(df1$age^2)-mean(df1$age)^2
var_Y = mean(df1$wage^2)-mean(df1$wage)^2
cov_XY/sqrt(var_X*var_Y)
```

```
## [1] 0.143492
```

1.2 Calculate the coefficients on this regression

$$\hat{\beta} = (X^T X)^{-1} X^T Y$$

```
beta_hat = solve(t(X) %*% X) %*% t(X) %*% Y
beta_hat
```

```
##             [,1]
## [1,] 14141.1794
## [2,]   230.9923
```

1.3 Calculate the standard errors of $\hat{\beta}$

1.3.1 Using the standard formulas of the OLS

$$\widehat{Var(\hat{\beta})} = \hat{\sigma}^2 (X^T X)^{-1}$$

$$SE(\hat{\beta}_i) = \hat{\sigma}\sqrt{(X^T X)^{-1}_{ii}}$$

```
Y_hat = X %*% beta_hat
e = Y - Y_hat
sigmasq_hat = (t(e) %*% e)/(length(Y)-2)
var_bata_hat = sigmasq_hat[1]*diag(2) %*% solve(t(X) %*% X)
se_bata_hat = sqrt(diag(var_bata_hat))
se_bata_hat
```

```
## [1] 645.2348  14.8774
```

1.3.2 Using bootstrap with 49 and 499 replications respectively. Comment on the difference between the two strategies

```
bootstrap = function(X,Y,R){
  betas = c()
  for (r in 1:R) {
    chosen_row = sample(nrow(X),nrow(X),replace=TRUE)
    X_boot = X[chosen_row,]
    Y_boot = Y[chosen_row]
    beta_hat_boot = solve(t(X_boot) %*% X_boot) %*% t(X_boot) %*% Y_boot
    betas = cbind(betas,beta_hat_boot)
  }
  return(betas)
}
```

```
set.seed(123)
apply(bootstrap(X,Y,49),MARGIN=1,sd)
```

```
## [1] 596.08216  15.72919
```

```
apply(bootstrap(X,Y,499),MARGIN=1,sd)
```

```
## [1] 625.2644  16.4461
```

The first strategy uses residual to estimate variance of random error term, and then use the estimated variance to calculate the standard errors of $\hat{\beta}$; the second method infers for unknown distribution of $\hat{\beta}$ using resampling method.

Exercise 2 Detrend Data

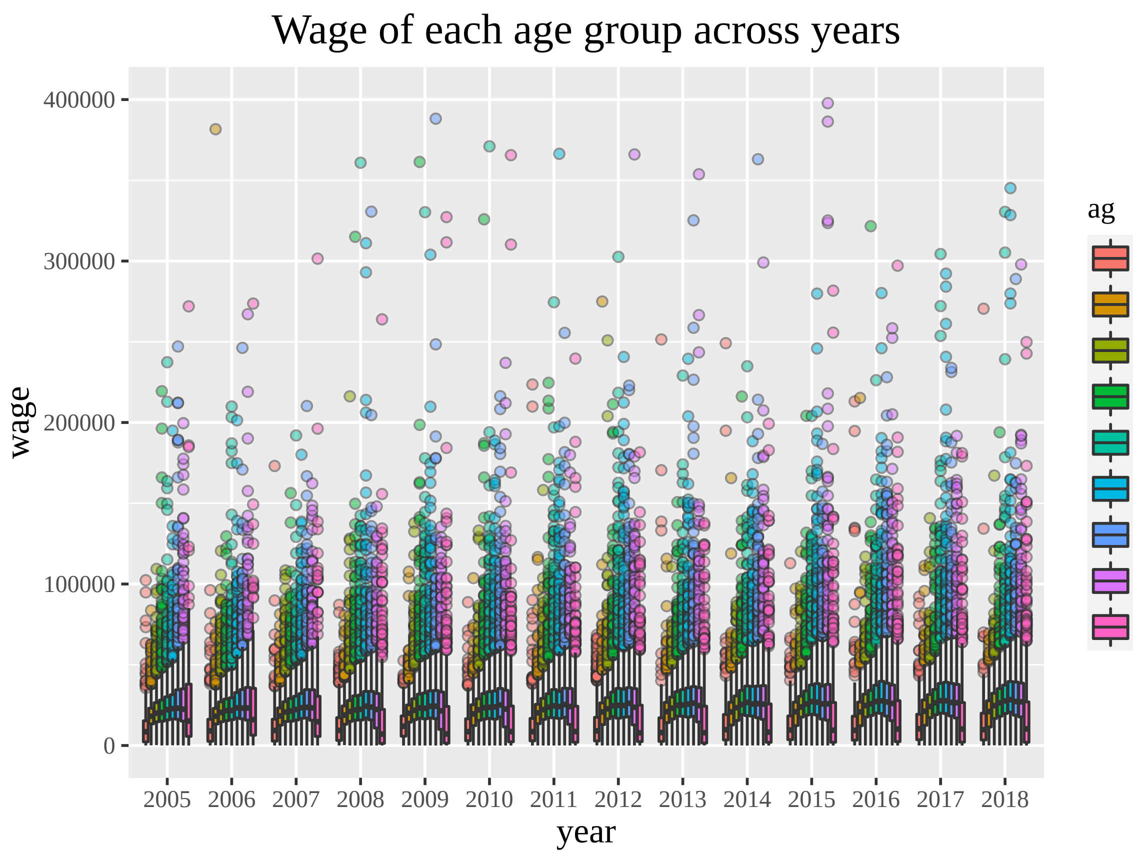
```
for (year in 2005:2018){
  datind_file = data.frame(fread(paste('datind',year,'.csv',sep=''),header=TRUE))
  assign(paste('datind',year,sep=''),datind_file)

  if (year==2005){
    datind = datind_file
  }else{
    datind = rbind(datind,datind_file)
  }
}
df2 = datind[,c('year','empstat','wage','age')]
```

2.1 Create a categorical variable ag

```
df2['ag'] = case_when(
  (df2$age>=18 & df2$age<=25) ~ '18-25',
  (df2$age>=26 & df2$age<=30) ~ '26-30',
  (df2$age>=31 & df2$age<=35) ~ '31-35',
  (df2$age>=36 & df2$age<=40) ~ '36-40',
  (df2$age>=41 & df2$age<=45) ~ '41-45',
  (df2$age>=46 & df2$age<=50) ~ '46-50',
  (df2$age>=51 & df2$age<=55) ~ '51-55',
  (df2$age>=56 & df2$age<=60) ~ '56-60',
  (df2$age>=61) ~ '60+'
)
# ignore observations under 18, and drop wage 0
df2 = df2[which(complete.cases(df2) & df2$wage!=0),]
```

2.2 Plot the wage of each age group across years. Is there a trend?

```
ggplot(df2,aes(as.character(year),wage)) +
  geom_boxplot(aes(fill=ag),outlier.shape=21,outlier.alpha=0.5) +
  ylim(0,400000) +
  ggtitle('Wage of each age group across years') +
  xlab('year') +
  theme(axis.title.x = element_text(size = 13),
        axis.title.y = element_text(size = 13),
        plot.title = element_text(hjust = 0.5,size=16),
        text=element_text(family='Times'))
```



Yes, there is a trend for wage of each age group. As age getting larger, the wage would first increase and then decrease.

2.3 Consider $Y_{it} = \beta X_{it} + \gamma_t + e_{it}$. After including a time fixed effect, how do the estimated coefficients change?

```
# Generate time dummy variables
for (year in 2005:2018){
  df2[as.character(year)] = as.numeric(df2$year==year)
}
Y = as.matrix(df2$wage)
X = as.matrix(cbind(rep(1,length(df2$age)),df2[,c(4,7:19)])) # delete 2005 as base group
beta_hat2 = solve(t(X) %*% X) %*% t(X) %*% Y
beta_hat2
```

```
##                                 [,1]
## rep(1, length(df2$age)) 10625.07352
## age                       297.81653
## 2006                      114.58388
## 2007                      135.20977
## 2008                      -90.79323
## 2009                      758.23827
## 2010                      568.06612
## 2011                     1106.64241
## 2012                     1550.65094
## 2013                     1424.06266
```

```
## 2014                         2029.96350
## 2015                         2305.74479
## 2016                         2982.29116
## 2017                         2940.21715
## 2018                         3044.37546
```

The intercept becomes smaller,and the effect of age becomes larger.

Exercise 3 Numerical Optimization

```
datind2007 = data.frame(fread(paste('datind2007.csv',sep=''),header=TRUE))
df3 = datind2007[,c('empstat','wage','age')]
df3 = df3[which(complete.cases(df3)),]
```

3.1 Exclude all individuals who are inactive

```
df3 = df3[which(df3$empstat!="Inactive"),]
df3 = df3[which(df3$empstat!='Retired'),] # someone who is retired is inactive
# Generate variable of being employed
df3['employed'] = case_when(
  (df3$empstat=='Employed') ~ 1,
  (df3$empstat=='Unemployed') ~ 0
)
```

3.2 Write a function that returns the likelihood of the probit of being employed

```
Flike_1 = function(beta,x1,y){ # one explanatory variable
  Xbeta = beta[1] + beta[2]*x1
  cdf = pnorm(Xbeta)
  cdf[cdf>0.999999] = 0.999999
  cdf[cdf<0.000001] = 0.000001 # cannot have log of zero
  log_likelihood = sum(y*log(cdf) + (1-y)*log(1-cdf))
  return(-log_likelihood)
}
```

3.3 Optimize the model and interpret the coefficients

```
num_try = 1000
out = mat.or.vec(num_try,3)
for (i in 1:num_try){
  start = runif(2,-10,10)
  res = optim(start,fn=Flike_1,method='BFGS',control=list(trace=6,maxit=1000),
                    x1=df3$age,y=df3$employed) # minimize minus log likelihood
  out[i,] = c(res$par,res$value)
}
out[which(out[,3]==min(out[,3])),]
```

```
[1]    1.052605555    0.006737124 3545.692253741
```

The coefficient of age is positive, which means age has a positive effect on being employed.

3.4 Can you estimate the same model including wages as a determinant of labor market participation? Explain

```
Flike_2 = function(beta,x1,x2,y){ # two explanatory variable
  Xbeta = beta[1] + beta[2]*x1 + beta[3]*x2
  cdf = pnorm(Xbeta)
  cdf[cdf>0.999999] = 0.999999
  cdf[cdf<0.000001] = 0.000001 # cannot have log of zero
  log_likelihood = sum(y*log(cdf) + (1-y)*log(1-cdf))
  return(-log_likelihood)
}
```

```
num_try = 1000
out2 = mat.or.vec(num_try,4)
for (i in 1:num_try){
  start = c(runif(1,0,0.5),runif(2,0,0.01))
  res = optim(start,fn=Flike_2,method='BFGS',control=list(trace=6,maxit=1000),
                x1=df3$age,x2=df3$wage,y=df3$employed)
  out2[i,] = c(res$par,res$value)
}
out2[which(out2[,4]==min(out2[,4])),]
```

```
[1]    0.36037232958    0.00366476077    0.00006562897 2791.22922489534
```

No, we cannot estimate the same model including wages as a determinant of labor market participation. Generally, unemployed people have wage zero, but there are still some of them have large wages, which may be outliers. So, we should not include wages in our model.

Exercise 4 Discrete choice

```
for (year in 2005:2015){
  datind_file = data.frame(fread(paste('datind',year,'.csv',sep=''),header=TRUE))
  assign(paste('datind',year,sep=''),datind_file)

  if (year==2005){
    datind = datind_file
  }else{
    datind = rbind(datind,datind_file)
  }
}
df4 = datind[,c('year','empstat','wage','age')]
```

4.1 Exclude all individuals who are inactive

```
df4 = df4[which(df4$empstat!="Inactive"),]
df4 = df4[which(df4$empstat!='Retired'),]
df4 = df4[which(complete.cases(df4)),]
df4['employed'] = case_when(
  (df4$empstat=='Employed') ~ 1,
  (df4$empstat=='Unemployed') ~ 0
)
```

```r
# Generate time dummy variables
for (year in 2005:2015) {
  df4[as.character(year)] = as.numeric(df4$year==year)
}
```

4.2 Write and optimize the probit, logit, and the linear probability model

```r
# (1) probit model
probit_like_2 = function(beta,x1,x2,y){
  Xbeta = beta[1] + beta[2]*x1 + x2 %*% as.matrix(beta[3:12])
  cdf = pnorm(Xbeta)

  cdf[cdf>0.999999] = 0.999999
  cdf[cdf<0.000001] = 0.000001
  log_likelihood = sum(y*log(cdf) + (1-y)*log(1-cdf))
  return(-log_likelihood)
}
```

```r
# Optimize probit model
num_try = 100
out4_probit = mat.or.vec(num_try,13)
out4_probit_hessian_s = list()
for (i in 1:num_try){
  start = runif(12,-1,1)
  res = optim(start,fn=probit_like_2,method='BFGS',control=list(trace=6,maxit=1000),
              x1=df4$age,x2=as.matrix(df4[,7:16]),y=df4$employed,hessian=TRUE)
  out4_probit[i,] = c(res$par,res$value)
  out4_probit_hessian_s[[i]] = res$hessian
}
out4_probit[which(out4_probit[,13]==min(out4_probit[,13])),]
```

```
 [1]  0.749491095  0.012338886  0.015260163  0.080064612  0.108384529  0.024805176
 [7]  0.021875311  0.053627994  0.009377134 -0.040946963 -0.034002549 -0.055725796
```

```r
# (2) logit model
logit_like_2 = function(beta,x1,x2,y){
  Xbeta = beta[1] + as.matrix(beta[2]*x1) + x2 %*% as.matrix(beta[3:12])
  logistic_fun = exp(Xbeta)/(1+exp(Xbeta))

  logistic_fun[logistic_fun>0.999999] = 0.999999
  logistic_fun[logistic_fun<0.000001] = 0.000001
  log_likelihood = sum(y*log(logistic_fun) + (1-y)*log(1-logistic_fun))
  return(-log_likelihood)
}
```

```r
# Optimize logit model
num_try = 100
out4_logit = mat.or.vec(num_try,13)
out4_logit_hessian_s = list()
for (i in 1:num_try){
  start = c(runif(1,0,2),runif(11,-1,1))
```

```r
  res = optim(start,logit_like_2,method='BFGS',control=list(trace=6,maxit=1000),
              x1=df4$age,x2=as.matrix(df4[,7:16]),y=df4$employed,hessian=TRUE)
  out4_logit[i,] = c(res$par,res$value)
  out4_logit_hessian_s[[i]] = res$hessian
}
out4_logit[which(out4_logit[,13]==min(out4_logit[,13])),]
```

```
 [1]  1.120342385  0.025356952  0.028218084  0.157070387  0.210172797  0.043742223
 [7]  0.037527893  0.097303190  0.009805422 -0.086859102 -0.073267494 -0.116072108
```

```r
# (3) linear probability model
Y = as.matrix(df4$employed)
X = as.matrix(cbind(rep(1,length(df4$age)),df4[,c(4,7:16)])) # year 2005 :base group
beta_hat4 = solve(t(X) %*% X) %*% t(X) %*% Y
beta_hat4
```

```
##                                  [,1]
## rep(1, length(df4$age))  0.7978781222
## age                      0.0023386254
## 2006                     0.0025310551
## 2007                     0.0138135121
## 2008                     0.0181377017
## 2009                     0.0038035183
## 2010                     0.0033095513
## 2011                     0.0085217166
## 2012                     0.0007194678
## 2013                    -0.0085849411
## 2014                    -0.0072380277
## 2015                    -0.0114074788
```

4.3 Interpret and compare the estimated coefficients. How significant are they?

```r
# (1) probit model
out4_probit_min_postion = which(out4_probit[,13]==min(out4_probit[,13]))
out4_probit_min_beta = out4_probit[out4_probit_min_postion,-13]
out4_probit_hessian = solve(out4_probit_hessian_s[[out4_probit_min_postion]])
out4_probit_se = sqrt(diag(out4_probit_hessian))
out4_probit_Z = out4_probit_min_beta/out4_probit_se
out4_probit_sig = as.numeric((out4_probit_Z > 1.96)|(out4_probit_Z<(-1.96)))

out4_probit_table = cbind(out4_probit_min_beta,out4_probit_se,out4_probit_sig)
row.names(out4_probit_table) = c('Intercept','age',paste('(year)',2006:2015,sep=''))
colnames(out4_probit_table) = c('probit_Estimate','probit_SE','5%_significant')
out4_probit_table
```

```
           probit_Estimate    probit_SE 5%_significant
Intercept      0.749491095 0.0228574647              1
age            0.012338886 0.0004071411              1
(year)2006     0.015260163 0.0228681218              0
(year)2007     0.080064612 0.0230297576              1
(year)2008     0.108384529 0.0232489862              1
(year)2009     0.024805176 0.0227781846              0
(year)2010     0.021875311 0.0225888750              0
(year)2011     0.053627994 0.0226358965              1
(year)2012     0.009377134 0.0221149719              0
(year)2013    -0.040946963 0.0223541506              0
(year)2014    -0.034002549 0.0223470073              0
(year)2015    -0.055725796 0.0223246712              1
```

```r
# (2) logit model
out4_logit_min_postion = which(out4_logit[,13]==min(out4_logit[,13]))
out4_logit_min_beta = out4_logit[out4_logit_min_postion,-13]
out4_logit_hessian = solve(out4_logit_hessian_s[[out4_logit_min_postion]])
out4_logit_se = sqrt(diag(out4_logit_hessian))
out4_logit_Z = out4_logit_min_beta/out4_logit_se
out4_logit_sig = as.numeric((out4_logit_Z > 1.96)|(out4_logit_Z<(-1.96)))

out4_logit_table = cbind(out4_logit_min_beta,out4_logit_se,out4_logit_sig)
row.names(out4_logit_table) = c('Intercept','age',paste('(year)',2006:2015,sep=''))
colnames(out4_logit_table) = c('logit_Estimate','logit_SE','5%_significant')
out4_logit_table
```

```
           logit_Estimate     logit_SE 5% significant
Intercept     1.120342385 0.0442157530              1
age           0.025356952 0.0008141604              1
(year)2006    0.028218084 0.0442068134              0
(year)2007    0.157070387 0.0449431837              1
(year)2008    0.210172797 0.0455189733              1
(year)2009    0.043742223 0.0440513998              0
(year)2010    0.037527893 0.0436893073              0
(year)2011    0.097303190 0.0439555482              1
(year)2012    0.009805422 0.0426954378              0
(year)2013   -0.086859102 0.0429089997              1
(year)2014   -0.073267494 0.0429722829              0
(year)2015   -0.116072108 0.0428234168              1
```

```r
# linear probability model
Y_hat4 = X %*% beta_hat4
e4 = Y - Y_hat4
sigmasq_hat4 = (t(e4) %*% e4)/(length(Y)-2)
var_bata_hat4 = sigmasq_hat4[1]*diag(12) %*% solve(t(X) %*% X)
out4_linear_se = sqrt(diag(var_bata_hat4))
out4_linear_Z = beta_hat4/out4_linear_se
out4_linear_sig = as.numeric((out4_linear_Z > 1.96)|(out4_linear_Z<(-1.96)))
```

```
out4_linear_table = cbind(beta_hat4,out4_linear_se,out4_linear_sig)
row.names(out4_linear_table) = c('Intercept','age',paste('(year)',2006:2015,sep=''))
colnames(out4_linear_table) = c('linear_Estimate','linear_SE','5%_significant')
out4_linear_table
```

```
             linear_Estimate      linear_SE 5%_significant
Intercept       0.7978781222 0.00420986549              1
age             0.0023386254 0.00007444258              1
(year)2006      0.0025310551 0.00409818093              0
(year)2007      0.0138135121 0.00406143412              1
(year)2008      0.0181377017 0.00406948012              1
(year)2009      0.0038035183 0.00407046954              0
(year)2010      0.0033095513 0.00403729345              0
(year)2011      0.0085217166 0.00401278362              1
(year)2012      0.0007194678 0.00396040875              0
(year)2013     -0.0085849411 0.00404646165              1
(year)2014     -0.0072380277 0.00403410147              0
(year)2015     -0.0114074788 0.00404680434              1
```

The estimated coefficients are different among three models, for probit model, the effect of age, year2007, 2008, 2011, 2015 are statistically significant at 5% significant level; for logit and linear probability model, in addition to the significant effect in probit model, the effect of year 2013 is also significant. From the sign of the coefficients, age has a positive effect on being employed. The probability of employment from year 2006 to 2012 is larger than that in year 2005; whereas, the probability of employment from year 2013 to 2015 is smaller than that in year 2005.

Exercise 5 Marginal Effects

5.1 Compute the marginal effect of the previous probit and logit models

```
# Marginal effect of probit model
ME_probit = function(beta,x1,x2){
  Xbeta = beta[1] + beta[2]*x1 + x2 %*% beta[3:12]
  pdf_beta = matrix(dnorm(Xbeta)) %*% t(beta)
  return(pdf_beta)
}
```

```
beta_probit = out4_probit[which(out4_probit[,13]==min(out4_probit[,13])),-13]
ME_probit_full = ME_probit(beta_probit,df4$age,x2=as.matrix(df4[,7:16]))
apply(ME_probit_full,MARGIN=2,mean)
```

```
 [1]  0.133202495  0.002192915  0.002712096  0.014229397  0.019262523
 [6]  0.004408473  0.003887766  0.009530977  0.001666541 -0.007277255
[11] -0.006043066 -0.009903807
```

```
# Marginal effect of logit model
ME_logit = function(beta,x1,x2){
  Xbeta = beta[1] + beta[2]*x1 + x2 %*% beta[3:12]
  return(as.matrix(exp(Xbeta)/(1+exp(Xbeta))^2) %*% t(beta))
}
```

```r
beta_logit = out4_logit[which(out4_logit[,13]==min(out4_logit[,13])),-13]
ME_logit_full = ME_logit(beta_logit,df4$age,x2=as.matrix(df4[,7:16]))
apply(ME_logit_full,MARGIN=2,mean)
```

```
##  [1]  0.1028244546  0.0023272482  0.0025898414  0.0144158403  0.0192895525
##  [6]  0.0040146390  0.0034442910  0.0089304374  0.0008999367 -0.0079718843
## [11] -0.0067244534 -0.0106530390
```

5.2 Construct the standard errors of the marginal effect. Hint: Bootstrap

```r
ME_boot = function(data,R){
  ME_probit_s = c()
  ME_logit_s = c()

  for (r in 1:R){
    chosen_row = sample(nrow(data),nrow(data),replace=TRUE)
    boot_data = data[chosen_row,]
    x1 = boot_data$age
    x2 = as.matrix(boot_data[,7:16])
    y = boot_data$employed

    num_try = 20
    out5_probit = mat.or.vec(num_try,13)
    for (i in 1:num_try){
      start = c(runif(1,-5,5),runif(11,-1,1))
      try({
        res = optim(start,fn=probit_like_2,method='BFGS',
                    control=list(trace=6,maxit=1000),x1=x1,x2=x2,y=y)
        out5_probit[i,] = c(res$par,res$value)},silent=TRUE)
    }
    beta_probit_boot = data.frame(out5_probit[which(out5_probit[,13]==min(out5_probit[,13])),-13])

    if (dim(beta_probit_boot)[2]==1){
      beta_probit_boot1 = as.numeric(t(beta_probit_boot))
    }else{
      beta_probit_boot1 = as.numeric(beta_probit_boot[1,])
    }
    ME_probit_full_boot = ME_probit(beta_probit_boot1,x1,x2)
    ME_probit_s = cbind(ME_probit_s,apply(ME_probit_full_boot,MARGIN=2,mean))

    out5_logit = mat.or.vec(num_try,13)
    for (j in 1:num_try){
      start = c(runif(1,-5,5),runif(11,-1,1))
      try({
        res = optim(start,fn=logit_like_2,method='BFGS',
                    control=list(trace=6,maxit=1000),x1=x1,x2=x2,y=y)
        out5_logit[j,] = c(res$par,res$value)},silent=TRUE)
    }
    beta_logit_boot = data.frame(out5_logit[which(out5_logit[,13]==min(out5_logit[,13])),-13])

    if (dim(beta_logit_boot)[2]==1){
      beta_logit_boot1 = as.numeric(t(beta_logit_boot))
    }else{
```

```
      beta_logit_boot1 = as.numeric(beta_logit_boot[1,])
    }
    ME_logit_full_boot = ME_logit(beta_logit_boot1,x1,x2)
    ME_logit_s = cbind(ME_logit_s,apply(ME_logit_full_boot,MARGIN=2,mean))
  }
  return(list(ME_probit_s,ME_logit_s))

}
```

```
ME_boot_results49 = ME_boot(df4,49)
ME_probit_boot49 = ME_boot_results49[[1]]
ME_logit_boot49 = ME_boot_results49[[2]]
apply(ME_probit_boot49,MARGIN=1,sd)
apply(ME_logit_boot49,MARGIN=1,sd)
```

```
 [1] 0.00414797349 0.00008994219 0.00335198511 0.00414087636 0.00375614551 0.00346814009
 [7] 0.00408425982 0.00391832618 0.00370442553 0.00370052486 0.00362701319 0.00387299015

 [1] 0.00427687853 0.00009653801 0.00334763846 0.00419993592 0.00378106479 0.00344055944
 [7] 0.00411909178 0.00390732377 0.00370924886 0.00366574142 0.00364341829 0.00381077252
```