ECON 613 Assignment 2 OLS and Probit

Xin Lin

1/28/2022

```
# Load packages
library(tidyverse)
library(magrittr)
library(janitor)
library(knitr)
library(kableExtra)
library(plm)
library(lmtest)

# abadon scientific notation in R
options(scipen = 999)
```

Exercise 1 OLS Estimate

```
# import dataset
dat <- read_csv("data/datind2009.csv")

# clean the data set by excluding values with NA and 0
dat_clean <- dat %>%
  filter(wage != 'NA' & wage != 0) %>%
  filter(age != 'NA' & wage != 0)
```

1. Calculate the correlation between Y and X

```
# direct computatition
X <- dat_clean[,9:10] # keep variables: wage and age
n <- nrow(dat_clean)
Xs <- scale(X, center=TRUE, scale=TRUE)
R <- t(Xs) %*% Xs / (n-1)
R[1,2]
```

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

```
# use function
cor(dat_clean$wage, dat_clean$age)
```

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

Answer: The correlation between Y and X is 0.143492.

2. Calculate the coefficients on this regression: $\hat{\beta} = (X^T X)^{-1} X^T Y$

```r
# direct computatition
y <- dat_clean$wage
x <- cbind(1, dat_clean$age)
beta_hat <- solve(t(x) %*% x) %*% t(x) %*% y
beta_hat
```

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

```r
# use function
lm(dat_clean$wage ~ dat_clean$age)
```

```
##
## Call:
## lm(formula = dat_clean$wage ~ dat_clean$age)
##
## Coefficients:
##   (Intercept)  dat_clean$age
##         14141            231
```

Answer:

	Coefficient
Intercept	14141.1794
Age	230.9923

3. Calculate the standard errors of β using OLS and bootstrap respectively and comment on the difference between the two strategies

```r
# Method 1. use the standard formulas of the OLS
# residuals
y_hat <- x %*% beta_hat
e_hat <- y - y_hat
# residual standard error
sigma2 <- (t(e_hat) %*% e_hat) / (n - 1)
sigma <- sqrt(sigma2)
# standard error
diag_xx <- diag(solve(t(x) %*% x))
beta_se <- sigma * sqrt(diag_xx)
beta_se
```

```
## [1] 645.20671  14.87675
```

```r
# Method 2. use bootstrap with 49 replications
set.seed(999)
R <- 49 # number of bootstraps
nind <- nrow(dat_clean) # number of individuals
nvar <- 2  # number of variables
outs <- mat.or.vec(R, nvar)
for (i in 1:R) {
  samp <- sample(1:nind, nind,rep=TRUE)
  dat_samp <- dat_clean[samp,]
  reg <- lm(wage ~ age,data = dat_samp)
  outs[i,] <- reg$coefficients
}
beta_se_49 <- apply(outs,2,sd)
beta_se_49
```

```
## [1] 646.82185  16.32833
```

```r
# Method 3. use bootstrap with 499 replications
R <- 499 # number of bootstraps
nind <- nrow(dat_clean) # number of individuals
nvar <- 2  # number of variables
outs <- mat.or.vec(R, nvar)
for (i in 1:R) {
  samp <- sample(1:nind, nind,rep=TRUE)
  dat_samp <- dat_clean[samp,]
  reg <- lm(wage ~ age,data = dat_samp)
  outs[i,] <- reg$coefficients
}
beta_se_499 <- apply(outs,2,sd)
beta_se_499
```

```
## [1] 567.14967  15.30548
```

Answer:

	OLS	Bootstrap_49	Bootstrap_499
Intercept	645.20671	646.82185	567.14967
Age	14.87675	16.32833	15.30548

Exercise 2 Detrend Data

```r
# import data sets
dat05ind <- read_csv("data/datind2005.csv")
dat06ind <- read_csv("data/datind2006.csv")
dat07ind <- read_csv("data/datind2007.csv")
dat08ind <- read_csv("data/datind2008.csv")
dat09ind <- read_csv("data/datind2009.csv")
dat10ind <- read_csv("data/datind2010.csv")
dat11ind <- read_csv("data/datind2011.csv")
```

```r
dat12ind <- read_csv("data/datind2012.csv")
dat13ind <- read_csv("data/datind2013.csv")
dat14ind <- read_csv("data/datind2014.csv")
dat15ind <- read_csv("data/datind2015.csv")
dat16ind <- read_csv("data/datind2016.csv")
dat17ind <- read_csv("data/datind2017.csv")
dat18ind <- read_csv("data/datind2018.csv")

# merge the above data sets
dat_ex2 <- rbind(dat05ind, dat06ind, dat07ind, dat08ind, dat09ind, dat10ind, dat11ind,
                dat12ind, dat13ind, dat14ind, dat15ind, dat16ind, dat17ind, dat18ind)

# clean the data set by excluding values with NA and 0
dat_ex2 <- dat_ex2 %>%
  filter(wage != 'NA' & wage != 0) %>%
  filter(age != 'NA' & wage != 0)
```

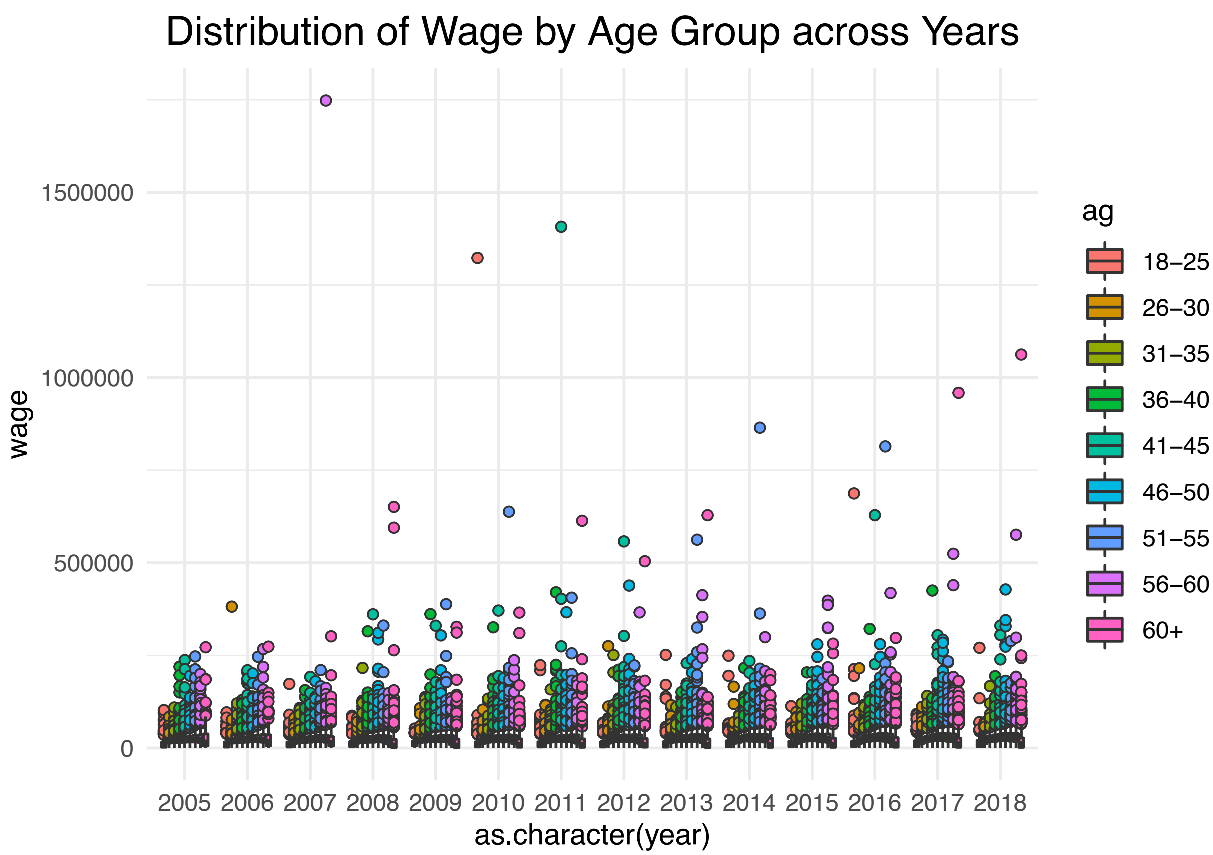
1. Create a categorical variable ag, which bins the age variables into the following groups: "18-25", "26- 30", "31-35", "36-40","41-45", "46-50","51-55", "56-60", and "60+"

```r
dat_ex2 <- dat_ex2 %>%
  filter(age >= 18) %>%
  mutate(ag = ifelse(age%in%c(18:25), "18-25",
                     ifelse(age%in%c(26:30), "26-30",
                            ifelse(age%in%c(31:35), "31-35",
                                   ifelse(age%in%c(36:40), "36-40",
                                          ifelse(age%in%c(41:45), "41-45",
                                                 ifelse(age%in%c(46:50), "46-50",
                                                        ifelse(age%in%c(51:55), "51-55",
                                                               ifelse(age%in%c(56:60), "56-60", "60+"))
# check the results
unique(dat_ex2$ag)
```

```
## [1] "31-35" "26-30" "36-40" "41-45" "51-55" "56-60" "46-50" "18-25" "60+"
```

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

```r
ggplot(dat_ex2, aes(x=as.character(year), y=wage)) +
  geom_boxplot(aes(fill=ag), outlier.shape = 21) +
  theme_minimal() +
  labs(title = "Distribution of Wage by Age Group across Years") +
  theme(plot.title = element_text(face = "plain", size = 15, hjust = 0.5, color = "black"))
```



From the above graph, we can see that there is a trend accross the years: as age gets older, wage increases until individuals achieve their middle age and then declines.

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

```
reg_ex2 <- plm(wage ~ age,
            data = dat_ex2,
            index = "year",
            model = "within")
coeftest(reg_ex2)
```

```
##
## t test of coefficients:
##
##     Estimate Std. Error t value              Pr(>|t|)
## age 297.8165     4.3695  68.159 < 0.00000000000000022 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Answer: After including a time fixed effect, the estimated coefficient on age becomes larger as shown below

	Coefficient on Age
No Time Fixed Effect	230.9923
With Time Fixed Effect	297.8165

Exercise 3 Numerical Optimization

1. Exclude all individuals who are inactive

```
dat_ex3 <- dat07ind %>%
  filter(empstat %in% c("Unemployed", "Employed")) %>% # exclue inactive
  mutate(empstat=ifelse(empstat == "Employed", 1, 0)) # 1 if employed and 0 if unemployed
```

2. Write a function that returns the likelihood of the probit of being employed

```
# write the function
# b is the coefficients, x is regressor, and y is response variable
flike <- function(b, x, y) {
  xbeta <- b[1] + b[2]*x
  prob <- pnorm(xbeta) # standard normal
  prob[prob > 0.999999] <- 0.999999
  prob[prob < 0.000001] <- 0.000001
  # to avoid log(0) in log(b) and log(1-b)
  # ensure that the probability is less than one and greater than 0
  p1 <- log(prob) # represent the log prob of (y=1)
  p0 <- log(1-prob) # represent the log prob of (y=0)
  like <- y * p1 + (1-y) * p0 #
  return(-sum(like)) # return negative allows us to maximize
}
```

3. Optimize the model and interpret the coefficients. You can use pre-programmed optimization packages

```
# initialize
time <- 100
result <- mat.or.vec(time, 3) # the first two rows are coef and the third row is minimizing value

# optimize
for (i in 1:time) {
  searchv = runif(2, -5, 5) # random starting search value
  res = optim(searchv,
              fn = flike,
              method = "BFGS",
              control = list(trace = 6, maxit = 3000),
              x = dat_ex3$age,
              y = dat_ex3$empstat)
  result[i,] = c(res$par, res$value)
}
```

```
## initial  value 144620.765593
## final  value 144620.765593
## converged
## initial  value 14810.237786
```

```
## initial  value 144620.765593
## final  value 144620.765593
## converged
## initial  value 14810.237786
## final  value 14810.237786
## converged
## initial  value 14810.237786
## final  value 14810.237786
## converged
## initial  value 14810.237786
## final  value 14810.237786
## converged
## initial  value 14810.237786
## final  value 14810.237786
## converged
## initial  value 144620.765593
## final  value 144620.765593
## converged
## initial  value 144620.765593
## final  value 144620.765593
## converged
## initial  value 144620.765593
## final  value 144620.765593
## converged
## initial  value 14810.237786
## final  value 14810.237786
## converged
## initial  value 144620.765593
## final  value 144620.765593
## converged
```

```r
# print the minimum negative log likelihood
result <- as.data.frame(result)
result[which(result$V3 == min(result$V3)), ]
```

```
##           V1          V2       V3
## 65 1.042158 0.006976191 3555.892
```

Answer: By optimizing the model, I get that the estimated coefficient on intercept is 1.043905, the estimated coefficient on age is 0.0069316, and the minimum negative loglikelihood is 3555.891. The coefficients can not be interpreted directly in the probit model, but we can know that age has a positive effect on market participation without controlling other factors.

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

```r
# write the function
# b is the coefficients, x1 and x2 are regressors, and y is response variable
flike2 <- function(b, x1, x2, y) {
  xbeta <- b[1] + b[2]*x1 + b[3]*x2
  prob <- pnorm(xbeta)
```

```r
  prob[prob > 0.999999] <- 0.999999
  prob[prob < 0.000001] <- 0.000001
  p1 <- log(prob)
  p0 <- log(1-prob)
  like <- y * p1 + (1-y) * p0
  return(-sum(like))
}

# initialize
time <- 100
result2 <- mat.or.vec(time, 4)

# clean the data set with NA
dat_ex3_clean <- dat_ex3 %>%
  filter(wage != 'NA',
         age != 'NA',
         empstat != 'NA')

# optimize
for (i in 1:time) {
  searchv = runif(3, -5, 5)
  res = optim(searchv,
              fn = flike2,
              method = "BFGS",
              control = list(trace = 6, maxit = 1000),
              x1 = dat_ex3_clean$age,
              x2 = dat_ex3_clean$wage,
              y = dat_ex3_clean$empstat)
  result2[i,] = c(res$par, res$value)
}
```

```
## initial  value 21510.783124
## final  value 21496.944394
## converged
## initial  value 137685.379777
## final  value 137685.379777
## converged
## initial  value 144427.348441
## final  value 144427.348441
## converged
## initial  value 21451.877538
## final  value 14754.975730
## converged
## initial  value 14754.975730
## final  value 14754.975730
## converged
## initial  value 9603.671210
## final  value 8448.671502
## converged
## initial  value 137699.195286
## final  value 137699.195286
## converged
## initial  value 144427.348441
```

```
## final  value 21317.342770
## converged
## initial  value 137671.565067
## final  value 137657.748766
## converged
## initial  value 144427.348441
## final  value 144427.348441
## converged
```

```r
# print the minimum negative log likelihood
result2 <- as.data.frame(result2)
result2[which(result2$V4 == min(result2$V4)), ]
```

```
##          V1        V2       V3       V4
## 6 -0.112352 0.0118834 1.130802 8448.672
```

No, we cannot estimate the same model including wages as a determinant of labor market participation because unemployed people generally have zero wage; however, there are a lot of outliers in our data set: some unemployed people have large wages. Therefore, we cannot include wages in our model.

Exercise 4 Discrete Choice

```r
# merge the 2005-2015 data sets
dat_ex4 <- rbind(dat05ind, dat06ind, dat07ind, dat08ind, dat09ind, dat10ind, dat11ind,
                 dat12ind, dat13ind, dat14ind, dat15ind)
```

1. Exclude all individuals who are inactive

```r
dat_ex4 <- dat_ex4 %>%
  # exclue inactive
  filter(empstat %in% c("Unemployed", "Employed")) %>%
  # 1 if employed and 0 if unemployed
  mutate(empstat=ifelse(empstat == "Employed", 1, 0)) %>%
  # make year dummy variables
  mutate(dum = 1) %>%
  pivot_wider(names_from = year, values_from = dum, values_fill = 0)
```

2. Write and optimize the probit, logit, and the linear probability models

```r
# get the regressors
y <- dat_ex4$empstat
x <- dat_ex4$age
y06 <- dat_ex4$'2006'
y07 <- dat_ex4$'2007'
y08 <- dat_ex4$'2008'
y09 <- dat_ex4$'2009'
```

```r
y10 <- dat_ex4$'2010'
y11 <- dat_ex4$'2011'
y12 <- dat_ex4$'2011'
y13 <- dat_ex4$'2012'
y14 <- dat_ex4$'2014'
y15 <- dat_ex4$'2015'
```

1) Probit Model

```r
# return the likelihood of probit function
probit_like <- function(b, x, y, y06, y07, y08, y09, y10, y11, y12, y13, y14, y15) {
  xbeta <- b[1] + b[2]*x + b[3]*y06 + b[4]*y07 + b[5]*y08 + b[6]*y09 +
    b[7]*y10 + b[8]*y11 + b[9]*y12 + b[10]*y13 + b[11]*y14 + b[12]*y15
  prob <- pnorm(xbeta) # standard normal
  prob[prob > 0.999999] <- 0.999999
  prob[prob < 0.000001] <- 0.000001
  p1 <- log(prob)
  p0 <- log(1-prob)
  like <- y * p1 + (1-y) * p0
  return(-sum(like))
}

# initialize
time <- 100
result_probit <- mat.or.vec(time, 13)

# optimize
for (i in 1:time) {
  searchv <- runif(12, -5, 5)
  res <- optim(searchv,
               fn = probit_like,
               method = "BFGS",
               control = list(trace = 6, maxit = 1000),
               x = x, y = y,
               y06 = y06, y07 = y07, y08 = y08, y09 = y09, y10 = y10,
               y11 = y11, y12 = y12, y13 = y13, y14 = y14, y15 = y15)
  result_probit[i,] <- c(res$par, res$value)
}
```

```
## initial  value 183608.250661
## final  value 183608.250661
## converged
## initial  value 1593563.894109
## final  value 1593563.894109
## converged
## initial  value 183608.250661
## final  value 183608.250661
## converged
## initial  value 1593563.894109
## final  value 1593563.894109
## converged
```

```
## converged
## initial  value 183608.250661
## final  value 183608.250661
## converged
## initial  value 146318.629193
## iter  10 value 92214.185233
## iter  20 value 42609.462614
## final  value 42245.245534
## converged
## initial  value 155400.889796
## iter  10 value 89487.658361
## iter  20 value 57527.769414
## iter  30 value 56339.412897
## iter  30 value 56339.412897
## iter  30 value 56339.412897
## final  value 56339.412897
## converged
## initial  value 1593563.894109
## final  value 1593563.894109
## converged
## initial  value 183608.250661
## final  value 183608.250661
## converged
## initial  value 183608.250661
## final  value 183608.250661
## converged
## initial  value 183608.250661
## final  value 183608.250661
## converged
## initial  value 646689.480198
## final  value 183608.250661
## converged
## initial  value 1534375.295120
## final  value 183608.250661
## converged
## initial  value 183608.250661
## final  value 183608.250661
## converged
## initial  value 183608.250661
## final  value 183608.250661
## converged
```

```
# print the minimum negative log likelihood
result_probit <- as.data.frame(result_probit)
result_probit[which(result_probit$V13 == min(result_probit$V13)), ]
```

```
##           V1         V2         V3        V4        V5         V6         V7
## 91 0.7287374 0.01231544 0.03801882 0.1009017 0.1299625 0.04710529 0.04252999
##          V8        V9        V10         V11         V12      V13
## 91 3.938964 -3.862953 0.03068647 -0.01281685 -0.03223671 42245.25
```

2) Logit Model

```r
# return the likelihood of logit function
logit_like <- function(b, x, y, y06, y07, y08, y09, y10, y11, y12, y13, y14, y15) {
  xbeta <- b[1] + b[2]*x + b[3]*y06 + b[4]*y07 + b[5]*y08 + b[6]*y09 +
    b[7]*y10 + b[8]*y11 + b[9]*y12 + b[10]*y13 + b[11]*y14 + b[12]*y15
  prob <- exp(xbeta) / (1+exp(xbeta))
  prob[prob > 0.999999] <- 0.999999
  prob[prob < 0.000001] <- 0.000001
  p1 <- log(prob)
  p0 <- log(1-prob)
  like <- y * p1 + (1-y) * p0
  return(-sum(like))
}

# initialize
time <- 100
result_logit <- mat.or.vec(time, 13)

# optimize
for (i in 1:time) {
  searchv <- runif(12, -5, 5)
  res <- optim(searchv,
               fn = logit_like,
               method = "BFGS",
               control = list(trace = 6, maxit = 1000),
               x = x, y = y,
               y06 = y06, y07 = y07, y08 = y08, y09 = y09, y10 = y10,
               y11 = y11, y12 = y12, y13 = y13, y14 = y14, y15 = y15)
  result_logit[i,] <- c(res$par, res$value)
}
```

```
## initial  value 183608.250661
## final  value 183608.250661
## converged
## initial  value 183608.250661
## final  value 183608.250661
## converged
## initial  value 1593563.894109
## final  value 1593563.894109
## converged
## initial  value 1593563.894109
## final  value 1593563.894109
## converged
## initial  value 183608.250661
## final  value 183608.250661
## converged
## initial  value 676690.494929
## final  value 183608.250661
## converged
## initial  value 183608.250661
## final  value 183608.250661
## converged
```

```
## initial  value 1593563.894109
## final  value 1593563.894109
## converged
## initial  value 1116863.628445
## final  value 183608.250661
## converged
## initial  value 183608.250661
## final  value 183608.250661
## converged
## initial  value 1593563.894109
## final  value 1593563.894109
## converged
## initial  value 1530041.604240
## final  value 183608.250661
## converged
## initial  value 1593563.894109
## final  value 1593563.894109
## converged
## initial  value 1593563.894109
## final  value 1593563.894109
## converged
## initial  value 183608.250661
## final  value 183608.250661
## converged
## initial  value 1593142.307118
## final  value 183608.250661
## converged
## initial  value 183608.250661
## final  value 183608.250661
## converged
```

```
# print the minimum negative log likelihood
result_logit <- as.data.frame(result_logit)
result_logit[which(result_logit$V13 == min(result_logit$V13)), ]
```

```
##          V1         V2         V3        V4        V5         V6         V7
## 70 1.077785 0.02526293 0.07577846 0.2017179 0.2570684 0.09011007 0.08098643
##            V8        V9        V10         V11        V12      V13
## 70 -0.2722411 0.4175975 0.05630649 -0.02676383 -0.0679094 42215.74
```

3) Linear Model

```
# get the value of regrssors and response variable
y_ex4 <- dat_ex4$empstat
x_ex4 <- rbind(rep(1,nrow(dat_ex4)), dat_ex4$age, dat_ex4$'2006', dat_ex4$'2007',
               dat_ex4$'2008', dat_ex4$'2009', dat_ex4$'2010', dat_ex4$'2011',
               dat_ex4$'2012', dat_ex4$'2013', dat_ex4$'2014', dat_ex4$'2015')

# solve for the coefficients
beta_linear <- solve(x_ex4 %*% t(x_ex4)) %*% x_ex4 %*% y_ex4
beta_linear
```

```
##                 [,1]
##  [1,]  0.7977483650
##  [2,]  0.0023358617
##  [3,]  0.0029332879
##  [4,]  0.0139479286
##  [5,]  0.0184425587
##  [6,]  0.0040834132
##  [7,]  0.0033035718
##  [8,]  0.0088873932
##  [9,]  0.0008988494
## [10,] -0.0083476676
## [11,] -0.0070497876
## [12,] -0.0109176395
```

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

Answer: Here are the estimated coefficients I get from the previous part:

	Probit Model	Logit Model	Linear Model
Intercept	0.7287374	1.0777849	0.7977484
Age	0.0123154	0.0252629	0.0023359
2006	0.0380188	0.0757785	0.0029333
2007	0.1009017	0.2017179	0.0139479
2008	0.1299625	0.2570684	0.0184426
2009	0.0471053	0.0901101	0.0040834
2010	0.0425300	0.0809864	0.0033036
2011	3.9389642	-0.2722411	0.0088874
2012	-3.8629528	0.4175975	0.0008988
2013	0.0306865	0.0563065	-0.0083477
2014	-0.0128169	-0.0267638	-0.0070498
2015	-0.0322367	-0.0679094	-0.0109176

The estimated coefficients are different among three models.

Firstly, let's look at the coefficeint on Intercept. It is positive across the three models; however, their values do not have explicit meaning, so, we cannot interpret them directly. But together with the coefficient on Age (which is also positive across the three models), we can conclude that there is a positive probability that individuals get employed. For the linear model, we can conclude that the probability for an eighteen-year-old individual getting employed is 0.7977 + 18 * 0.002336 = 83.97%, holding the other variables constant.

Next, let's look at the coefficient on Age. It is also positive across the three models. For the probit and logit model, we can conclude that when the age increases, the probability of being employed also increases. For the linear model, we can conclude that when the age increases by one year, the probability of being employed increases by 0.2933%.

Also, given the asymptotic of MLE and OLS estimators, the coefficients from the three models are all statistically significant.

Exercise 5 Marginal Effects

1. Compute the marginal effect of the previous probit and logit models

```
# calculate the marginal effect evaluated at the mean on the base year: 2005
dat_ex5 <- dat_ex4[dat_ex4$'2005' == 1, ]
xbar <- mean(dat_ex5$age)

# probit model
probit_b <- probit[1:2,]
probit_ME <- dnorm(probit_b[1]+probit_b[2]*xbar)*probit_b[2]
probit_ME
```

```
##          V1
## 0.002308115
```

```
# logit model
logit_b <- logit[1:2,]
e2 <- exp(-logit_b[1]-logit_b[2]*xbar)
logit_ME <- logit_b[2]*e2/((1+e2)^2)
logit_ME
```

```
##         V2
## 0.00244694
```

<u>Answer:</u> Here are the marginal effects for probit and logit models:

	Probit Model	Logit Model
Marginal Effect	0.0023081	0.0024469

2. Construct the standard errors of the marginal effects

```
# get the regressors and response variable
dat2 <- cbind.data.frame(y, x, y06, y07, y08, y09, y10, y11, y12, y13, y14, y15)

# write the function that returns the coefficients of the model
get_coef <- function(likefn, dat){
  time <- 100
  result <- mat.or.vec(time, 13)
  for (i in 1:time){
    searchv <- runif(12, -2, 2)
    res <- optim(searchv,
                 fn = likefn,
                 method = "BFGS",
                 control=list(trace=6, REPORT=1, maxit=1000),
                 x = dat$x, y = dat$y, y06 = dat$y06,
                 y07 = dat$y07, y08 = dat$y08, y09 = dat$y09,
                 y10 = dat$y10, y11 = dat$y11, y12 = dat$y12,
                 y13 = dat$y13, y14 = dat$y14, y15 = dat$y15)
```

```r
    result[i,] <- c(res$par, res$value)
  }
  result <- as.data.frame(result)
  coef <- result[which(result$V13 == min(result$V13)), ]
  return(coef[,-13])
}

# bootstrap: replicate 49 times for 12 coefficients
set.seed(999)
R <- 49
n <- nrow(dat2)

# probit model
resultsProbit <- mat.or.vec(R, 1)
for (i in 1:R) {
  # sample data using bootstrap
  dat_boot <- dat2[sample(1:n, n, replace=TRUE), ]
  # use the above function to get the coefficients of probit model
  coef <- get_coef(probit_like, dat_boot)
  # estimate the probit model
  x_bar <- mean(dat_boot$x)
  resultsProbit[i] <- dnorm(as.numeric(coef[1])+as.numeric(coef[2])*x_bar)*as.numeric(coef[2])
}
sd_probit <- sd(resultsProbit)

# logit model
resultsLogit <- mat.or.vec(R, 1)
for (i in 1:R) {
  # sample data using bootstrap
  dat_boot <- dat2[sample(1:n, n, replace=TRUE), ]
  # use the above function to get the coefficients of logit model
  coef <- get_coef(logit_like, dat_boot)
  # estimate the probit model
  x_bar <- mean(dat_boot$x)
  epow <- exp(-as.numeric(coef[1])-as.numeric(coef[2])*x_bar)
  resultsLogit[i] <- as.numeric(coef[2])*epow / ((1+epow)^2)
}
sd_logit <- sd(resultsLogit)
```

Answer: Here are the standard errors of the marginal effects for probit and logit models:

	Probit Model	Logit Model
Standard Error	0.0001002	0.0000959