HW 2

Ying Liu

```r
setwd("C:/Users/Veronica/Documents/data")
library(tidyverse)
library(ggplot2)
library(readr)
```

Exercise 1

```r
datind2009=read.csv("datind2009.csv", header = TRUE)
#omit the data if either wage or age/empstat is NA
datind2009 = subset(datind2009, select = c("empstat", "age", "wage"))
datind2009 = na.omit(datind2009)
age=datind2009$age
wage=datind2009$wage
```

1.1 The correlation between X and Y is -0.17885

```r
#calculate the correlation between Y and X
corre=cor(wage, age)
corre #-0.17885
```

corre	-0.178851156226984

1.2 The coefficient is -180

```r
#calculate the coefficient on this regression
x = cbind(1, datind2009$age)
beta=solve(t(x) %*% x) %*% (t(x) %*% wage)
beta  #-180
```

beta	num [1:2, 1] 22075 -180

1.3 1) Using the standard formula, the standard error of beta is 6.968652

```r
#calculate the standard error of beta
# 1)using formula
sigma_squre=t(wage-x %*% beta) %*% (wage-x %*% beta)/(nrow(x)-ncol(x))
sigma_squre=as.numeric(sigma_squre)
var_beta_hat=sigma_squre * solve(t(x) %*% x)
std_err_beta = sqrt(diag(var_beta_hat))
std_err_beta[2] #6.968652
```

std_err_beta	num [1:2] 357.83 6.97

2) Using the bootstrap with 49 and 499 replications, the standard error of beta is 6.97. The first method usually draws a sample of normal distribution. For the unknown distribution of beta, the second one can be a more reliable way.

```r
# 2)using bootstrap
reg = lm(wage ~ age, data = datind2009)
reg_sum = summary(reg)
R1  = 49                         # number of bootstraps
R2 =  499
num_ind = nrow(datind2009) # number of individuals in the data
num_var = length(reg$coefficients)  # number of variables in the data

outs1 = mat.or.vec(R1, num_var)
set.seed(123)

for (i in 1:R1)
{
  sample = sample(1:num_ind, num_ind, rep = TRUE)
  data_samp = datind2009[sample, ]
  reg1 = lm(wage ~ age, data = datind2009)
  outs1[i,] = reg1$coefficients
}
```

```r
mean_est1 = apply(outs1, 2, mean)
sd_est1 = apply(outs1, 2, sd)

est1 = cbind(summary(reg1)$coefficients[ , 1], summary(reg1)$coefficients[ , 2], mean_est1, sd_est1)
colnames(est1) = c("CF: estimate","CF: std dev","BT (49): estimate","BT (49): std dev")
est1 #6.968652
```

```r
outs2 = mat.or.vec(R2, num_var)
set.seed(123)

for (i in 1:R2)
{
  sample = sample(1:num_ind, num_ind, rep = TRUE)
  data_samp = datind2009[sample, ]
  reg2 = lm(wage ~ age, data = datind2009)
  outs2[i,] = reg2$coefficients
}

mean_est2 = apply(outs2, 2, mean)
sd_est2 = apply(outs2, 2, sd)

est2 = cbind(summary(reg2)$coefficients[ , 1], summary(reg2)$coefficients[ , 2], mean_est2, sd_est2)
colnames(est2) = c("CF: estimate","CF: std dev","BT (499): estimate","BT (499): std dev")
est2 #6.968652
```

Exercise 2

```r
#combine the data
for(i in 2005:2018)
{
  datind = read.csv(paste('datind',i,'.csv',sep = ""))
  datind$idind = as.character(datind$idind)
  datind$idmen = as.character(datind$idmen)
  assign(paste('datind_',i,sep = ""),datind)

}
datind = rbind(datind_2005,datind_2006,
                 datind_2007,datind_2008,datind_2009,
                 datind_2010,datind_2011,datind_2012,
                 datind_2013,datind_2014,datind_2015,
                 datind_2016,datind_2017,datind_2018
)
datind = subset(datind, select = c("year", "empstat", "age", "wage"))
datind = na.omit(datind)

# Create a categorical variable ag
ag = data.frame(datind, bin = cut(datind$age, c(18, 25, 30, 35, 40, 45, 50, 55, 60, 100), include.lowest = TRUE))
ag = na.omit(ag)
```

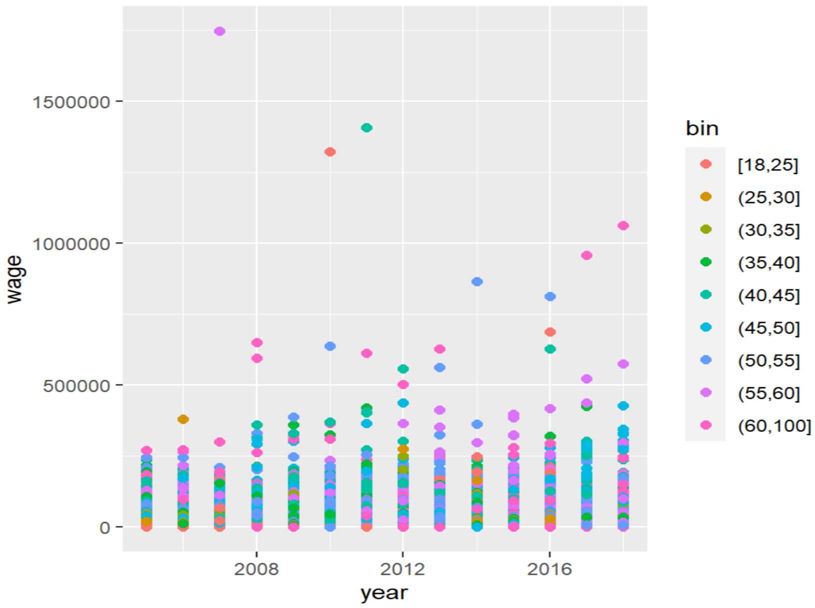
2.1

```r
# Create a categorical variable ag
ag = data.frame(datind, bin = cut(datind$age, c(18, 25, 30, 35, 40, 45, 50, 55, 60, 100), include.lowest = TRUE))
ag = na.omit(ag)
```

	year	empstat	age	wage	bin
1	2005	Inactive	31	12334	(30,35]
3	2005	Employed	32	50659	(30,35]
4	2005	Employed	28	19231	(25,30]
5	2005	Retired	90	0	(60,100]
6	2005	Employed	37	31511	(35,40]
7	2005	Employed	35	24873	(30,35]
8	2005	Employed	41	30080	(40,45]
0	2005	Employed	55	43296	(50,55]
1	2005	Employed	55	20426	(50,55]
2	2005	Employed	57	0	(55,60]
3	2005	Employed	52	0	(50,55]
6	2005	Employed	51	0	(50,55]
7	2005	Employed	47	0	(45,50]
9	2005	Employed	55	49240	(50,55]

2.2

```
# 2.2  plot the wage of each age group
ggplot(ag, aes( x = year, y = wage)) + geom_point( aes( color = bin), size=2)
```



For the group of people with older age, wage goes up as the time goes; But for the group of people with younger age, wage almost stays constant across the year.

2.3

```
# 2.3 Consider Y = beta*X + gamma*Year + e
reg3 = lm(wage ~ age + year, data = ag)
reg3_sum = summary(reg3)
reg3_sum[["coefficients"]]
```

```
                Estimate   Std. Error    t value      Pr(>|t|)
(Intercept) -598559.0541 19374.781368  -30.89372 3.253777e-209
age            -239.4385     2.088296 -114.65738  0.000000e+00
year            310.4354     9.635200   32.21888 2.495656e-227
```

The effect of the age becomes larger after including the year.

Exercise 3

3.1

```
datind2007=read.csv("datind2007.csv", header = TRUE)
#omit the data if either wage or age is NA
datind2007 = subset(datind2007, select = c("empstat", "age","wage"))

# 3.1 exclude individuals who are inactive
datind2007 = na.omit(datind2007) %>% filter(empstat != "Inactive", empstat != "Retired")
```

3.2

```
# Create the dummy variable
datind2007$empstat[ which(datind2007$empstat == "Employed") ] = 1
datind2007$empstat[ which(datind2007$empstat == "Unemployed") ] = 0
empstat2007 = as.numeric(datind2007$empstat)
age2007 =datind2007$age
wage2007 =datind2007$wage
# 3.2 Define function
Probit_model <- function(empst, age, cf)
{ XB = cf[1] + cf[2]*age
  Prob = pnorm(XB)
  Prob[Prob>0.999999] = 0.999999 # These two lines ensure that the probability is less than one and greater than
  Prob[Prob<0.000001] = 0.000001
  p1 = log(Prob)
  p0 = log(1-Prob)
  log_likelihood = sum(empst * p1 + (1-empst) * p0)
  return( -log_likelihood )
}
```

3.3

```
# 3.3 Optimize the model

num= 1000
out3 = mat.or.vec(num,3)
for (i in 1:num){
  random_start     = runif(2,-5,5)
  res       = optim(random_start,fn=Probit_model,method='BFGS',control=list(trace=6,maxit=1000),
                    age=age2007,empst=empstat2007) # minimize minus log likelihood
  out3[i,] = c(res$par,res$value)
}
out3[which(out3[,3]==min(out3[,3])),] # collect the estimate minimize the value
```

```
> out3[which(out3[,3]==min(out3[,3])),] # collect the estimate minimize the value
[1] 1.052429e+00 6.742876e-03 3.545692e+03
```

The coefficient of age is 1.0524, which is relatively small but positive, meaning that as the age grows, the person has greater probability of being employed.

3.4

```
# 3.4 Estimate the same model including wage
Probit_model2 = function(cf,age,wage,empst){
  Xb = cf[1] + cf[2]*age + cf[3]*wage
  Prob = pnorm(Xb)
  Prob[Prob>0.999999] = 0.999999
  Prob[Prob<0.000001] = 0.000001
  p1 = log(Prob)
  p0 = log(1-Prob)
  log_likelihood = sum(empst*p1 + (1-empst)*p0)
  return(-log_likelihood)# use the negative one to calculate the minimum
}

out4 = mat.or.vec(num,4)
for (i in 1:num){
  random_start     = c(runif(1,0,0.05),runif(3,0,0.01))
  res       = optim(random_start,fn=Probit_model2,method='BFGS',control=list(trace=6,maxit=1000),age=age2007,wage=
  out4[i,] = c(res$par,res$value)
}
out4[which(out4[,4]==min(out4[,4])),]
```

```
converged
> out4[which(out4[,4]==min(out4[,4])),]
[1] 4.290375e-02 7.936833e-03 7.613043e-05 2.807630e+03
>
```

The answer is no. We cannot estimate the same model including wage. Because the wage has tight relation with the age.

Exercise 4

4.1

```r
#combine the data
for(i in 2005:2015)
{
  datind = read.csv(paste('datind',i,'.csv',sep = ""))
  datind$idind = as.character(datind$idind)
  datind$idmen = as.character(datind$idmen)
  assign(paste('datind_',i,sep = ""),datind)

}
datind_2005to2015 = rbind(datind_2005,datind_2006,
                datind_2007,datind_2008,datind_2009,
                datind_2010,datind_2011,datind_2012,
                datind_2013,datind_2014,datind_2015
)
datind_2005to2015 = subset(datind_2005to2015, select = c("year", "empstat", "age", "wage"))
datind_2005to2015 = na.omit(datind_2005to2015)

# 4.1 Exclude individulas who are inactive
datind_2005to2015 = na.omit(datind_2005to2015) %>% filter(empstat != "Inactive", empstat != "Retired")
#generate the empstat dummy and year dummy
datind_2005to2015$empstat[ which(datind_2005to2015$empstat == "Employed") ] = 1
datind_2005to2015$empstat[ which(datind_2005to2015$empstat == "Unemployed") ] = 0
```

4.2

1)Probit

```r
#1) probit model
Probit_mol = function(cf,x1,x2,empst){
  temp = x2 %*% as.matrix(cf[3:12]) # describe cf[i]* certain year
  Xb = cf[1] + cf[2]*x1 + temp
  Prob = pnorm(Xb)
  Prob[Prob>0.999999] = 0.999999
  Prob[Prob<0.000001] = 0.000001
  p1 = log(Prob)
  p0 = log(1-Prob)
  log_likelihood = sum(empst*p1 + (1-empst)*p0)
  return(-log_likelihood)
}
# Optimize probit model
num = 10
result_probit = mat.or.vec(1,12)
minLocprobit = 0
minLikeprobit = Inf
for (i in 1:num){
  random_start = runif(12,-5,5)
  outcome_probit = optim(random_start,fn=Probit_mol,method='BFGS',control=list(trace=6,maxit=3000),
            x1=datind_2005to2015$age,x2=as.matrix(datind_2005to2015[,6:15]),empst=empstat_2005to2015,hessian=T


if(outcome_probit$value <  minLikeprobit){
  minLikeprobit = outcome_probit$value
  minLocprobit = i
  result_probit = outcome_probit$par
}
}
parameterprobit=result_probit
parameterprobit
```

```
> parameterprobit
 [1]  0.5338187  1.1715625 -2.6612653 -1.8295368 -4.3603026
 [6]  2.4700207  0.5234257  1.4546254 -4.9028956  1.6147311
[11]  3.2737585  3.9764814
```

2)Logit

```r
# (2) logit model
Logit_mol = function(cf,x1,x2,empst){
  temp = x2 %*% as.matrix(cf[3:12]) # describe cf[i]* certain year
  Xb = cf[1] + cf[2]*x1 + temp
  Prob=exp(Xb)/(1+exp(Xb))
  Prob[Prob>0.999999] = 0.999999
  Prob[Prob<0.000001] = 0.000001
  p1 = log(Prob)
  p0 = log(1-Prob)
  log_likelihood = sum(empst*p1 + (1-empst)*p0)
  return(-log_likelihood)
}
# Optimize logit model
num = 10
result_logit = mat.or.vec(1,12)
minLoclogit = 0
minLikelogit = Inf
for (i in 1:num){
  random_start = runif(12,-5,5)
  outcome_logit = optim(random_start,fn=Logit_mol,method='BFGS',control=list(trace=6,maxit=3000),
                        x1=datind_2005to2015$age,x2=as.matrix(datind_2005to2015[,6:15]),empst=empstat_2005to2015,


if(outcome_logit$value <  minLikelogit){
  minLikelogit = outcome_logit$value
  minLoclogit = i
  result_logit = outcome_logit$par
}
}
parameterlogit=result_logit
parameterlogit
```

```
> parameterlogit
 [1]  1.11925123  0.02538901  0.02760052  0.15623551
 [5]  0.20972404  0.04279752  0.03758098  0.09715678
 [9]  0.01040283 -0.08738293 -0.07389215 -0.11636665
```

3)Linear

```r
#3) linear probability model
Linear_mol = function(cf,x1,x2,empst){
  temp = x2 %*% as.matrix(cf[3:12]) # describe cf[i]* certain year
  Xb = cf[1] + cf[2]*x1 + temp
  Prob=Xb
  Prob[Prob>0.999999] = 0.999999
  Prob[Prob<0.000001] = 0.000001
  p1 = log(Prob)
  p0 = log(1-Prob)
  log_likelihood = sum(empst*p1 + (1-empst)*p0)
  return(-log_likelihood)
}
# Optimize linear model
num = 10
result_linear = mat.or.vec(1,12)
minLoclinear = 0
minLikelinear = Inf
for (i in 1:num){
  random_start = runif(12,-5,5)
  outcome_linear = optim(random_start,fn=Linear_mol,method='BFGS',control=list(trace=6,maxit=3000),
                        x1=datind_2005to2015$age,x2=as.matrix(datind_2005to2015[,6:15]),empst=empstat_2005to2015,h
  if(outcome_linear$value <  minLikelinear){
    minLikelinear = outcome_linear$value
    minLoclinear = i
    result_linear = outcome_linear$par
  }
}
parameterlinear=result_linear
parameterlinear
```

```
> parameterlinear
 [1]  3.6998377  1.7145697  0.9643769 -4.8814752  1.0828294
 [6] -2.2240179  4.4761399 -2.7850928  0.6316834  4.4621345
[11]  0.8016458  2.6597640
```

4.3 Firstly, let us see the coefficient of age, beta1. All the three models show that the age has a positive effect on the employment status. Secondly, for beta0, all the three models show that in 2005 people has pos*itive* probability of being employed. Thirdly, when we consider the impact of the year, we can see that probit and logit models show the relatively similar outputs.

Exercise 5

5.1

```
parprobit= parameterprobit[1:2]
xbar = mean(datind_2005to2015$age)
ME_Probit = dnorm(parprobit[1]+parprobit[2]*xbar)*parprobit[2]
ME_Probit #0
#Logit Model
parlogit = parameterlogit[1:2]
epow = exp(parlogit[1]-parlogit[2]*xbar)
ME_Logit = parlogit[2]*epow/((1+epow)^2)
ME_Logit #0.006339
```

5.2

```
#5.2 Construct the standard error of marginal effect
CF = function(fn,dathind){
  num = 100
  result = mat.or.vec(1,12)
  minLoc = 0
  minLike = Inf
  for (i in 1:num){
    random_start = runif(12,-5,5)
    outcome = optim(random_start,fn=Linear_mol,method='BFGS',control=list(trace=6,maxit=3000),
                    x1=datind_2005to2015$age,x2=as.matrix(datind_2005to2015[,6:15]),empst=empstat_2005to2015,hessian=TRUE)
  if(outcome$value <  minLike){
    minLike = outcome$value
    minLoc = i
    result = outcome$par
  }
}
  return(result)
}

R =  49
num_ind = nrow(datind_2005to2015) # number of individuals in the data
resultsprobit = mat.or.vec(R, 1)

for (i in 1:R)
{
  sample = sample(1:num_ind, num_ind, rep = TRUE)
  data_samp = datind_2005to2015[sample, ]
  reg1 = CF(Probit_mol,data_samp)
  data_samp_2005=data_samp[data_samp["year"] == 2005]
  x_samp_bar = mean(data_samp_2005$age)
  resultsprobit[i] = dnorm(coef[1]+coef[2]*x_samp_bar)*coef[2]
}
```

```
> mean_est = mean(resultsprobit)
> sd_est = sd(resultsprobit)
> sd_est
[1] 0
```

```
mean_est = mean(resultsprobit)
sd_est = sd(resultsprobit)
sd_est

R =  49
num_ind = nrow(datind_2005to2015) # number of individuals in the data
resultslogit = mat.or.vec(R, 1)

for (i in 1:R)
{
  sample = sample(1:num_ind, num_ind, rep = TRUE)
  data_samp = datind_2005to2015[sample, ]
  reg1 = CF(Logit_mol,data_samp)
  data_samp_2005=data_samp[data_samp["year"] == 2005]
  x_samp_bar = mean(data_samp_2005$age)
  resultslogit[i] = dnorm(coef[1]+coef[2]*x_samp_bar)*coef[2]
}

mean_est = mean(resultslogit)
sd_est_logit = sd(resultslogit)
sd_est_logit
```

```
> mean_est = mean(resultslogit)
> sd_est_logit = sd(resultslogit)
> sd_est_logit
[1] 0
```