HW2

Mike Sun

2/3/2022

```
datind2009 <- read.csv("~/Desktop/Fourth_Semester/613/HW2/datind2009.csv")
```

*Exercise*1

Calculate the correlation between Y and X:

```
datind2009_wage_age <- na.omit(select(datind2009,wage,age))

X = datind2009_wage_age$age
Y = datind2009_wage_age$wage

cor(X,Y)
```

```
## [1] -0.1788512
```

Calculate the coefficients on this regression:

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

```
##          [,1]
## [1,] 219.9245
```

Calculate the standard errors of beta: Using the standard formulas of the OLS:

```
sigmasq = sum((Y-(datind2009_wage_age$age)%*%beta)^2)/(nrow(datind2009_wage_age) - 2)
sebeta = sqrt(sigmasq%*%solve((t(X)%*%X)))
sebeta
```

```
##          [,1]
## [1,] 2.779211
```

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

```
set.seed(125)
n<- nrow(datind2009_wage_age)
beta_table_49 = c()

for (i in 1:49){
  sample<- datind2009_wage_age[sample(n, .5*n , replace = FALSE),]
  beta_h <- solve(t(sample[,2])%*%sample[,2])%*%t(sample[,2])%*%sample[,1]
  beta_table_49 = append(beta_table_49,beta_h)
}
```

```
beta_average_49 = mean(beta_table_49)
beta_average_49
```

```
## [1] 219.7277
```

```
set.seed(500)

beta_table_499 = c()

for (i in 1:499){
  sample<- datind2009_wage_age[sample(n, .5*n , replace = FALSE),]
  beta_h <- solve(t(sample[,2])%*%sample[,2])%*%t(sample[,2])%*%sample[,1]
  beta_table_499 = append(beta_table_499,beta_h)
}

beta_average_499 = mean(beta_table_499)
beta_average_499
```

```
## [1] 219.8518
```

Personally, I have no preference. The first one is the solution for full data while bootstrap results are brute force results of different subsets.

*Exercise*2

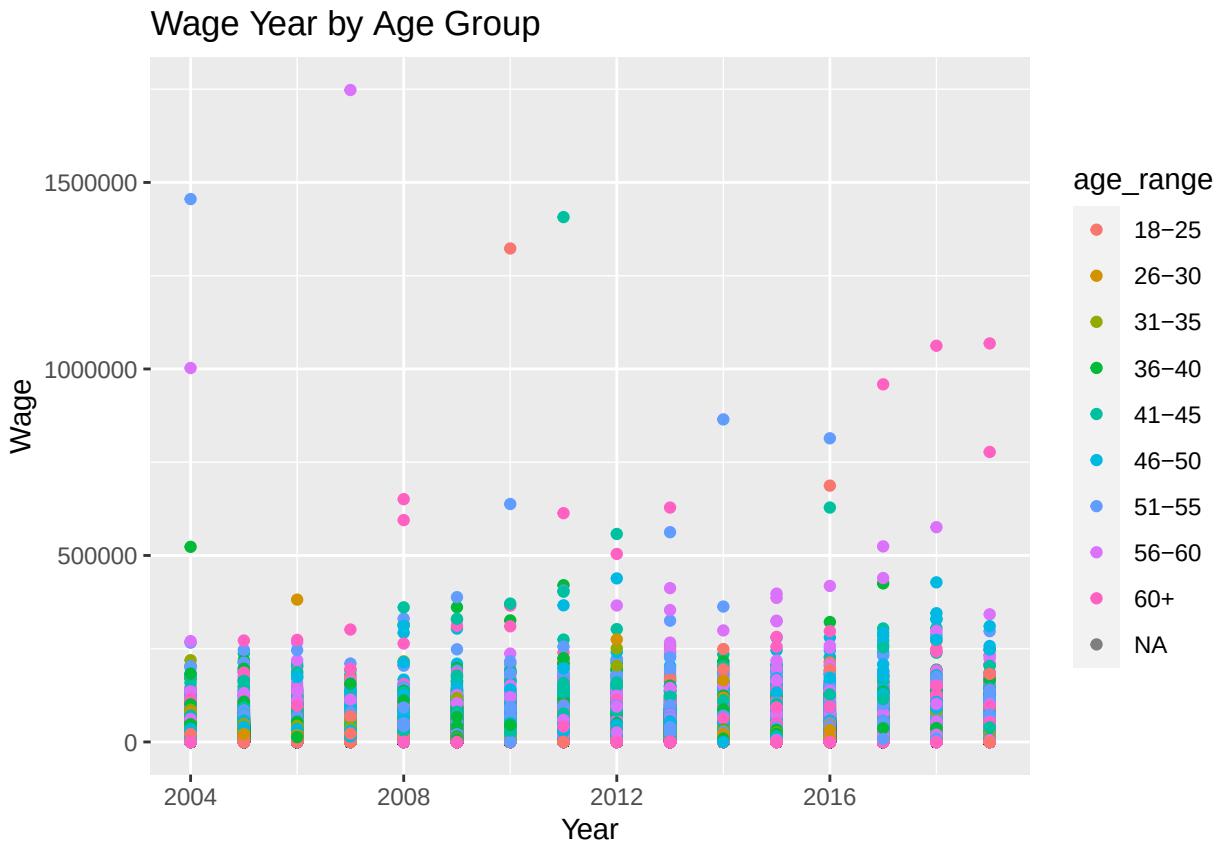
```
datind_full = unique(rbind(datind2004,datind2005,datind2006,datind2007,datind2008,datind2009,
                           datind2010,datind2011,datind2012,datind2013,datind2014,datind2015,
                           datind2016,datind2017,datind2018,datind2019))
```

```
datind_full = datind_full %>% mutate(age_range = as.factor(
  ifelse(18 <= age & age<= 25 , '18-25',
         ifelse(26 <= age & age<= 30, '26-30',
                ifelse(31 <= age & age<= 35, '31-35',
                       ifelse(36 <= age & age<= 40, '36-40',
                              ifelse(41 <= age & age<= 45, "41-45",
                                     ifelse(46 <= age & age<= 50, "46-50",
                                            ifelse(51 <= age & age<= 55, "51-55",
                                                   ifelse(56 <= age & age<= 60, "56-60", "60+")))))))))
```

```
wage_group_plot = ggplot(data=datind_full, aes(y=wage,x=year,group=age_range,colour = age_range)) +
  geom_point() +
  ggtitle("Wage Year by Age Group") + xlab("Year") + ylab("Wage")
```

```
wage_group_plot
```

```
## Warning: Removed 85184 rows containing missing values (geom_point).
```



Based on the graph, middle age group has higher income in general, and people get higher income in as year passes.

After including a time fixed effect, how do the estimated coefficients change?

```
datind_full$year = as.factor(datind_full$year)
summary(lm(wage ~ age + year,datind_full))
```

```
##
## Call:
## lm(formula = wage ~ age + year, data = datind_full)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
##  -21830  -11576   -7243    8510 1738132
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 20449.893    179.673 113.817  < 2e-16 ***
## age          -186.215      1.903 -97.864  < 2e-16 ***
## year2005      194.309    216.418   0.898   0.3693
## year2006      216.073    214.466   1.007   0.3137
## year2007      488.706    212.350   2.301   0.0214 *
## year2008     1618.920    212.912   7.604 2.89e-14 ***
## year2009     1913.789    212.662   8.999  < 2e-16 ***
## year2010     2062.714    210.751   9.787  < 2e-16 ***
## year2011     2308.995    209.673  11.012  < 2e-16 ***
## year2012     2794.046    207.242  13.482  < 2e-16 ***
```

```
## year2013     2671.356    210.959  12.663  < 2e-16 ***
## year2014     2941.960    210.021  14.008  < 2e-16 ***
## year2015     3312.957    210.316  15.752  < 2e-16 ***
## year2016     3602.019    210.249  17.132  < 2e-16 ***
## year2017     3670.662    212.223  17.296  < 2e-16 ***
## year2018     3827.609    213.489  17.929  < 2e-16 ***
## year2019     4359.895    210.089  20.753  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 20480 on 328303 degrees of freedom
##   (85184 observations deleted due to missingness)
## Multiple R-squared:  0.03094,    Adjusted R-squared:  0.03089
## F-statistic:   655 on 16 and 328303 DF,  p-value: < 2.2e-16
```

age now have negative effect on income across years, but years does have a positive effect on income.

*Exercise*3

Exclude all individuals who are inactive:

```
datind2007_active = datind2007 %>% filter(empstat != 'Inactive')
```

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

```
datind2007_active = datind2007_active %>% mutate(employed = ifelse(empstat =="Employed",1,0))
datind2007_short = na.omit(datind2007_active %>% select(age,employed))

probit.ll <- function(beta, x, y) {
  x.matrix <- as.matrix(x)
  x.beta <- x.matrix %*% beta
  proby <- pnorm(x.beta)
  proby[proby > 0.99999] = 0.99999
  proby[proby < 0.00001] = 0.00001
  ll <- sum((y*log(proby)) + (1-y)*log(1-proby),na.rm = TRUE)
  return(ll)
}
```

Optimize the model and interpret the coefficients.

```
list = seq(-1,1, by = 0.01)
best_l = -1000000

for(i in list){
  l = probit.ll(i,datind2007_active$age,datind2007_active$employed)
  if(l > best_l){
    best_l = l
    best_beta = i
  }
}
best_l
```

```
## [1] -11533.28
```

```
best_beta
```

```
## [1] 0
```

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

Yes, since if you receive wages, you are employed.

*Exercise*4

Exclude all individuals who are inactive.

```
datind2005_2015 = unique(rbind(datind2005,datind2006,datind2007,datind2008,datind2009,
                         datind2010,datind2011,datind2012,datind2013,datind2014,datind2015))

datind2005_2015$year = as.factor(datind2005_2015$year)
```

```
datind2005_2015 = datind2005_2015 %>% filter(empstat != 'Inactive') %>%
  mutate(employed = ifelse(empstat =="Employed",1,0))
```

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

```
logit <- glm(employed ~ age + year, family=binomial(link="logit"), data=datind2005_2015)
probit <- glm(employed ~ age + year, family = binomial(link = "probit"), data=datind2005_2015)
OLS <- lm(employed ~ age + year, data=datind2005_2015)
```

```
summary(logit)
```

```
##
## Call:
## glm(formula = employed ~ age + year, family = binomial(link = "logit"),
##     data = datind2005_2015)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.2421  -0.5466   0.2595   0.6449   2.8174
##
## Coefficients:
##              Estimate Std. Error  z value Pr(>|z|)
## (Intercept)  7.030702   0.037402  187.975  < 2e-16 ***
## age         -0.124143   0.000554 -224.098  < 2e-16 ***
## year2006     0.020509   0.031865    0.644  0.51983
## year2007     0.061578   0.031485    1.956  0.05049 .
## year2008     0.081826   0.031555    2.593  0.00951 **
## year2009     0.005498   0.031364    0.175  0.86085
## year2010     0.014255   0.030945    0.461  0.64505
## year2011     0.068384   0.030746    2.224  0.02614 *
## year2012     0.055240   0.030322    1.822  0.06848 .
## year2013     0.012652   0.030801    0.411  0.68125
## year2014     0.068207   0.030611    2.228  0.02587 *
## year2015     0.055309   0.030554    1.810  0.07026 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 255165  on 190295  degrees of freedom
## Residual deviance: 153801  on 190284  degrees of freedom
## AIC: 153825
##
```

```
## Number of Fisher Scoring iterations: 5
```

```
summary(probit)
```

```
##
## Call:
## glm(formula = employed ~ age + year, family = binomial(link = "probit"),
##     data = datind2005_2015)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.3309  -0.6057   0.3056   0.7375   2.8148
##
## Coefficients:
##              Estimate Std. Error  z value Pr(>|z|)
## (Intercept)  3.572379   0.018300  195.213   <2e-16 ***
## age         -0.063590   0.000258 -246.455   <2e-16 ***
## year2006     0.008744   0.017372    0.503   0.6147
## year2007     0.036271   0.017174    2.112   0.0347 *
## year2008     0.042294   0.017208    2.458   0.0140 *
## year2009    -0.008325   0.017115   -0.486   0.6267
## year2010    -0.002783   0.016909   -0.165   0.8693
## year2011     0.024255   0.016801    1.444   0.1488
## year2012     0.010830   0.016567    0.654   0.5133
## year2013    -0.017260   0.016829   -1.026   0.3051
## year2014     0.012913   0.016741    0.771   0.4405
## year2015     0.001498   0.016716    0.090   0.9286
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 255165  on 190295  degrees of freedom
## Residual deviance: 159784  on 190284  degrees of freedom
## AIC: 159808
##
## Number of Fisher Scoring iterations: 5
```

```
summary(OLS)
```

```
##
## Call:
## lm(formula = employed ~ age + year, data = datind2005_2015)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.27519 -0.23173  0.02784  0.28419  1.09432
##
## Coefficients:
##               Estimate Std. Error  t value Pr(>|t|)
## (Intercept)  1.5446069  0.0038082  405.599   <2e-16 ***
## age         -0.0184661  0.0000484 -381.532   <2e-16 ***
## year2006     0.0021767  0.0041665    0.522   0.6014
## year2007     0.0065161  0.0041151    1.583   0.1133
## year2008     0.0075729  0.0041161    1.840   0.0658 .
```

```
## year2009    -0.0017817  0.0041056   -0.434   0.6643
## year2010    -0.0008924  0.0040617   -0.220   0.8261
## year2011     0.0053247  0.0040339    1.320   0.1868
## year2012     0.0035873  0.0039811    0.901   0.3675
## year2013    -0.0022910  0.0040519   -0.565   0.5718
## year2014     0.0045557  0.0040345    1.129   0.2588
## year2015     0.0023945  0.0040345    0.594   0.5528
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3675 on 190284 degrees of freedom
## Multiple R-squared:  0.4344, Adjusted R-squared:  0.4344
## F-statistic: 1.329e+04 on 11 and 190284 DF,  p-value: < 2.2e-16
```

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

For and OLS we have: as age increase by 1, the probability of being employed goes down by 0.18%. However, for logit and probit, we can only say that, increasing in age leads to lower probability of employment. Results of three model are all significant. Probit and Logit results are the same, just different scaling.

*Exercise*5

logit:

```
logit_m = summary(margins(logit))
```

```
logit_m$AME
```

```
##           age      year2006      year2007      year2008      year2009
## -0.0160045386  0.0026506959  0.0079441147  0.0105466186  0.0007110323
##      year2010      year2011      year2012      year2013      year2014
##  0.0018428831  0.0088194680  0.0071285854  0.0016357507  0.0087966853
##      year2015
##  0.0071374285
```

probit:

```
probit_m=summary(margins(probit))
```

```
probit_m$AME
```

```
##           age      year2006      year2007      year2008      year2009
## -0.0156852062  0.0021590566  0.0089372289  0.0104162883 -0.0020583116
##      year2010      year2011      year2012      year2013      year2014
## -0.0006877079  0.0059820957  0.0026737042 -0.0042702792  0.0031876515
##      year2015
##  0.0003701047
```

Construct the standard errors of the marginal effects

logit:

```
logit_m$SE
```

```
##      Var_dydx_age Var_dydx_year2006 Var_dydx_year2007 Var_dydx_year2008
##      2.156884e-05      4.118529e-03      4.062046e-03      4.067145e-03
## Var_dydx_year2009 Var_dydx_year2010 Var_dydx_year2011 Var_dydx_year2012
##      4.056324e-03      4.000750e-03      3.965723e-03      3.913460e-03
## Var_dydx_year2013 Var_dydx_year2014 Var_dydx_year2015
```

```
##      3.982366e-03      3.948070e-03      3.942911e-03
```

probit:

```
probit_m$SE
```

```
##      Var_dydx_age Var_dydx_year2006 Var_dydx_year2007 Var_dydx_year2008
##      2.380243e-05      4.289748e-03      4.231912e-03      4.237933e-03
## Var_dydx_year2009 Var_dydx_year2010 Var_dydx_year2011 Var_dydx_year2012
##      4.231577e-03      4.178870e-03      4.144008e-03      4.090387e-03
## Var_dydx_year2013 Var_dydx_year2014 Var_dydx_year2015
##      4.163402e-03      4.132598e-03      4.130048e-03
```