```
##Author: Tian Chen
> ##Date: 02/03/2022
> ##Purpose: Econ 613 Assignment 2
>
> ########################
> ##      Question 1            ##
> ########################
> require(tidyverse)
> #X: the age of indivduals plus intercept
> #Y: Wage
> #Import data
> datind2009 <- read.csv("~/Desktop/Duke study/Econ613/A2/datind2009.csv")
>
> ##1. Caculate the correlation between Y and X
> cor(datind2009$age, datind2009$wage, use = "complete.obs")
[1] -0.1788512
>
>
> ##2. Calculate the coefficients on this regression. remember
> #Beta = (X'X)^(-1)X'Y
> datind2009_nona <- datind2009 %>% select(age, wage) %>% drop_na()
> age <- as.matrix(datind2009_nona$age)
> one <- rep(1, length(datind2009_nona$age))
> dim(one) <- c(length(datind2009_nona$age), 1)
> X <- cbind(one, age)
> Y <- as.matrix(datind2009_nona$wage)
> beta <- solve(t(X)%*%X)%*%t(X)%*%Y
> beta
             [,1]
[1,] 22075.1066
[2,]    -180.1765
>
>
> ##3.1 Calculate the standard errors of beta
> Y_hat <- X%*%beta
> residual <- Y - Y_hat
> sigma_square <- (t(residual)%*%residual)/length(residual)-2
> SE_beta <- sigma_square[1,1]*solve(t(X)%*%X)
> SE_constant <- sqrt(SE_beta[1,1])
> SE_beta <- sqrt(SE_beta[2,2])
> SE_constant
```

```
[1] 357.8098
> SE_beta
[1] 6.968308
>
> ##3.2 Using bootstrap with 49 and 499 replications respectively. Comment on the difference between
> #the two strategies.
>
> #49 replications
> num <- 1:49
> data <- cbind(Y, X)
> colnames(data) <- c("wage", "one", "age")
> result <- matrix(1, nrow = 49, ncol = 2)
> for (i in num) {
+     sample <- data[sample(nrow(data), 10000), ]
+     beta <- solve(t(sample[,2:3])%*%sample[,2:3])%*%t(sample[,2:3])%*%sample[,1]
+     result[i,1] <- beta[1,1]
+     result[i,2] <- beta[2,1]
+ }
> #bootstrap constant: 49 replication
> mean(result[,1])
[1] 22085.7
> #bootstrap beta
> mean(result[,2])
[1] -180.6486
>
> #499 replications
> num <- 1:499
> result <- matrix(1, nrow = 499, ncol = 2)
> for (i in num) {
+     sample <- data[sample(nrow(data), 10000), ]
+     beta <- solve(t(sample[,2:3])%*%sample[,2:3])%*%t(sample[,2:3])%*%sample[,1]
+     result[i,1] <- beta[1,1]
+     result[i,2] <- beta[2,1]
+ }
> #bootstrap constant: 499 replication
> mean(result[,1])
[1] 22097.94
> #bootstrap beta: 499 replication
> mean(result[,2])
[1] -180.499
>
```

```
> ######################
> ##       Question 2       ##
> ######################
> #Import data
> rm(list = ls())
> datind_file <- list.files(pattern = "^datind")
> datind_combined <- read.csv(datind_file[1])
> datind_file <- datind_file[-1]
> for(file in datind_file){
+      csv <- read.csv(file)
+      datind_combined <- rbind(datind_combined, csv)
+ }
> datind_combined_2005_2018 <- datind_combined %>% as_tibble() %>% filter(year>2004&year<2019)
>
> ##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+”.
> datind_combined_2005_2018 <- datind_combined_2005_2018 %>% mutate(ag = as.factor(ifelse(18 <= datind_combined_2005_2018$age & datind_combined_2005_2018$age<= 25 , '18-25',
+
ifelse(26 <= datind_combined_2005_2018$age & datind_combined_2005_2018$age<= 30, '26-30',
+
ifelse(31 <= datind_combined_2005_2018$age & datind_combined_2005_2018$age<= 35, '31-35',
+
ifelse(36 <= datind_combined_2005_2018$age & datind_combined_2005_2018$age<= 40, '36-40',
+
ifelse(41 <= datind_combined_2005_2018$age & datind_combined_2005_2018$age<= 45, "41-45",
+
ifelse(46 <= datind_combined_2005_2018$age & datind_combined_2005_2018$age<= 50, "46-50",
+
ifelse(51 <= datind_combined_2005_2018$age & datind_combined_2005_2018$age<= 55, "51-55",
+
ifelse(56 <= datind_combined_2005_2018$age & datind_combined_2005_2018$age<= 60, "56-60", "60+")))))))))
>
>
> ##2.Plot the wage of each age group across years. Is there a trend?
> plot(datind_combined_2005_2018$ag, datind_combined_2005_2018$wage)
> #trend: with age increasing, wage concentrates on higher income
>
> ##3.Fixed effect model
> #we use fast dummy package to create time fixed effect dummy
```

```
> #install.packages("fastDummies")
> require(fastDummies)
> datind_combined_2005_2018 <- dummy_cols(datind_combined_2005_2018,select_columns = "year")
> data <- datind_combined_2005_2018 %>% select(wage, age, year_2005:year_2018) %>% drop_na()
> X <- as.matrix(select(data, age, year_2005:year_2018))
> Y <- as.matrix(select(data, wage))
> results <- solve(t(X)%*%X)%*%t(X)%*%Y
> results
                 wage
age         -186.8793
year_2005 20675.0583
year_2006 20696.9955
year_2007 20969.8609
year_2008 22100.2489
year_2009 22395.4188
year_2010 22544.5834
year_2011 22791.0759
year_2012 23276.2858
year_2013 23153.9017
year_2014 23424.7333
year_2015 23796.0275
year_2016 24085.1717
year_2017 24154.0902
year_2018 24311.2098
> #the OLS results underestimate the coefficient of age
>
>
> ########################
> ##       Question3       ##
> ########################
> datind_2007 <- read.csv("~/Desktop/Duke study/Econ613/A2/datind2007.csv")
> datind_2007_tbl <- datind_2007 %>% as_tibble()
>
> ##1. exclude inactive respondents
> datind_2007_wage_age <- datind_2007_tbl %>% select(wage, age, empstat) %>% drop_na()
>
>
> ##2.
> # Decide feature and label
> df_tbl <- datind_2007_wage_age %>% select(empstat, age, wage) %>% drop_na()
> df_tbl <- df_tbl %>% mutate(employ_dummy = ifelse(df_tbl$empstat == "employed", 1,0 ))
```

```
> X <- as.matrix(df_tbl$age)
> Y <- as.matrix(df_tbl$employ_dummy)
>
> # Likelihood Function
> Likelihood <- function(X = X, y = Y, beta){
+     num <- 1:length(X)
+     L = 0
+     for (i in num) {
+        x_i = X[i,]
+        y_i = y[i]
+        L = L * ((pnorm(x_i%*%beta,0,1))^y_i) * ((1- pnorm(x_i%*%beta,0,1))^(1-y_i))
+     }
+     return(L)
+ }
>
>
>
>
> ###################
> ##    Question4    ``##
> ###################
> ##Exclude all individuals who are inactive.
> datind_2005_2015 <- datind_combined %>% filter(year >= 2005 & year <= 2015)
> datind_2005_2015 <- dummy_cols(datind_2005_2015,select_columns = "year") %>% select(age, empstat, year) %>% drop_na()
> #empstat: individual's participation in the labor market
> df <- datind_2005_2015 %>%mutate(employ_dummy = ifelse(datind_2005_2015$empstat == "Employed", 1,0), year_factor = as.factor(year))
>
> ##Write and optimize the probit, logit, and the linear probability models.
> ##OLS, probit, logit
> require(glm2)
>
> #LPM
> LPM <- lm(employ_dummy~age + year_factor, data = df)
> summary(LPM)

Call:
lm(formula = employ_dummy ~ age + year_factor, data = df)

Residuals:
```

```
    Min      1Q  Median      3Q     Max 
-0.4929 -0.4173 -0.3563  0.5907  0.6461 

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)    
(Intercept)      3.489e-01  3.471e-03 100.506  < 2e-16 ***
age              1.373e-03  3.894e-05  35.261  < 2e-16 ***
year_factor2006 -1.701e-03  4.410e-03  -0.386 0.699620    
year_factor2007  2.001e-03  4.369e-03   0.458 0.646869    
year_factor2008  6.683e-03  4.385e-03   1.524 0.127553    
year_factor2009 -2.250e-03  4.381e-03  -0.514 0.607552    
year_factor2010 -3.187e-03  4.344e-03  -0.734 0.463139    
year_factor2011  1.164e-03  4.324e-03   0.269 0.787694    
year_factor2012 -1.080e-03  4.272e-03  -0.253 0.800431    
year_factor2013 -1.043e-02  4.352e-03  -2.396 0.016584 *  
year_factor2014 -1.070e-02  4.336e-03  -2.468 0.013594 *  
year_factor2015 -1.558e-02  4.342e-03  -3.589 0.000331 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.4889 on 288115 degrees of freedom
Multiple R-squared:  0.004407,	Adjusted R-squared:  0.004369 
F-statistic: 115.9 on 11 and 288115 DF,  p-value: < 2.2e-16

> #probit
> Probit <- glm(employ_dummy~age + year_factor, data = df, family = binomial(link = "probit"))
> summary(Probit)

Call:
glm(formula = employ_dummy ~ age + year_factor, family = binomial(link = "probit"), 
    data = df)

Deviance Residuals: 
    Min       1Q   Median       3Q      Max  
-1.1759  -1.0405  -0.9348   1.3368   1.4451  

Coefficients:
                  Estimate Std. Error z value Pr(>|z|)    
(Intercept)     -0.3953542  0.0090318 -43.774  < 2e-16 ***
age              0.0037591  0.0001012  37.133  < 2e-16 ***
year_factor2006 -0.0044737  0.0114527  -0.391 0.696075    
```

```
year_factor2007  0.0051530  0.0113415   0.454 0.649574
year_factor2008  0.0172643  0.0113781   1.517 0.129182
year_factor2009 -0.0059321  0.0113782  -0.521 0.602115
year_factor2010 -0.0084036  0.0112829  -0.745 0.456387
year_factor2011  0.0028293  0.0112239   0.252 0.800982
year_factor2012 -0.0030504  0.0110900  -0.275 0.783274
year_factor2013 -0.0273775  0.0113113  -2.420 0.015505 *
year_factor2014 -0.0281670  0.0112681  -2.500 0.012429 *
year_factor2015 -0.0409693  0.0112899  -3.629 0.000285 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 387903  on 288126  degrees of freedom
Residual deviance: 386562  on 288115  degrees of freedom
AIC: 386586

Number of Fisher Scoring iterations: 4

> #logit
> Logit <- glm(employ_dummy~age + year_factor, data = df, family = binomial(link = "logit"))
> summary(Logit)

Call:
glm(formula = employ_dummy ~ age + year_factor, family = binomial(link = "logit"),
    data = df)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.1691  -1.0389  -0.9393   1.3373   1.4405

Coefficients:
                  Estimate Std. Error z value Pr(>|z|)
(Intercept)     -0.6204247  0.0145623 -42.605  < 2e-16 ***
age              0.0057317  0.0001631  35.135  < 2e-16 ***
year_factor2006 -0.0071359  0.0184455  -0.387 0.698856
year_factor2007  0.0083168  0.0182600   0.455 0.648776
year_factor2008  0.0277757  0.0183116   1.517 0.129308
year_factor2009 -0.0094823  0.0183242  -0.517 0.604827
year_factor2010 -0.0134115  0.0181714  -0.738 0.460481
```

```
year_factor2011  0.0047471  0.0180693   0.263 0.792771
year_factor2012 -0.0046276  0.0178567  -0.259 0.795517
year_factor2013 -0.0437949  0.0182267  -2.403 0.016271 *
year_factor2014 -0.0449385  0.0181566  -2.475 0.013322 *
year_factor2015 -0.0655199  0.0181991  -3.600 0.000318 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 387903  on 288126  degrees of freedom
Residual deviance: 386632  on 288115  degrees of freedom
AIC: 386656

Number of Fisher Scoring iterations: 4

>
> ##Interpret and compare the estimated coefficients. How significant are they?
> #All model's coefficient are statistically significant at 5 % level, but the linear probability model
> #returns the extremely small coefficient. In conclusion, age is positively correlated with employment
> #Older people are morelily to get a job.
>
>
> #######################
> ##      Question 5        ##
> #######################
> #install.packages("ggeffects")
> #install.packages("prediction")
> require(ggeffects, prediction)
>
>
> ##Marginal effects
> #LPM
> LPM_effects <- ggpredict(LPM, "age")
> LPM_effects
# Predicted values of employ_dummy

age | Predicted |         95% CI
------------------------------
 -5 |      0.34 | [0.34, 0.35]
 10 |      0.36 | [0.36, 0.37]
```

```
 20 |           0.38 | [0.37, 0.38]
 35 |           0.40 | [0.39, 0.40]
 50 |           0.42 | [0.41, 0.42]
 65 |           0.44 | [0.43, 0.44]
 80 |           0.46 | [0.45, 0.47]
105 |           0.49 | [0.49, 0.50]

Adjusted for:
* year_factor = 2005
> #Probit
> Probit_effects <- ggpredict(Probit, "age")
Data were 'prettified'. Consider using `terms="age [all]"` to get smooth plots.
> Probit_effects
# Predicted probabilities of employ_dummy

age | Predicted |          95% CI
------------------------------
 -5 |           0.34 | [0.33, 0.35]
 10 |           0.36 | [0.35, 0.37]
 20 |           0.37 | [0.37, 0.38]
 35 |           0.40 | [0.39, 0.40]
 50 |           0.42 | [0.41, 0.42]
 65 |           0.44 | [0.43, 0.45]
 80 |           0.46 | [0.46, 0.47]
105 |           0.50 | [0.49, 0.51]

Adjusted for:
* year_factor = 2005
> #Logit
> Logit_effects <- ggpredict(Logit, "age")
Data were 'prettified'. Consider using `terms="age [all]"` to get smooth plots.
> Logit_effects
# Predicted probabilities of employ_dummy

age | Predicted |          95% CI
------------------------------
 -5 |           0.34 | [0.34, 0.35]
 10 |           0.36 | [0.36, 0.37]
 20 |           0.38 | [0.37, 0.38]
 35 |           0.40 | [0.39, 0.40]
 50 |           0.42 | [0.41, 0.42]
```

```
 65 |            0.44 | [0.43, 0.45]
 80 |            0.46 | [0.45, 0.47]
105 |            0.50 | [0.49, 0.50]

Adjusted for:
* year_factor = 2005
>
> ##Standard error of marginal effects
> #LPM
> SD_LPM_effects <- cbind(as.matrix(LPM_effects)[,1], as.matrix(LPM_effects)[,3])
> SD_LPM_effects
      [,1]  [,2]
 [1,] " -5" "0.003558843"
 [2,] "  0" "0.003471461"
 [3,] "  5" "0.003393014"
 [4,] " 10" "0.003324135"
 [5,] " 15" "0.003265428"
 [6,] " 20" "0.003217450"
 [7,] " 25" "0.003180688"
 [8,] " 30" "0.003155532"
 [9,] " 35" "0.003142263"
[10,] " 40" "0.003141030"
[11,] " 45" "0.003151848"
[12,] " 50" "0.003174593"
[13,] " 55" "0.003209012"
[14,] " 60" "0.003254735"
[15,] " 65" "0.003311292"
[16,] " 70" "0.003378141"
[17,] " 75" "0.003454684"
[18,] " 80" "0.003540292"
[19,] " 85" "0.003634324"
[20,] " 90" "0.003736145"
[21,] " 95" "0.003845136"
[22,] "100" "0.003960705"
[23,] "105" "0.004082293"
> #Probit
> SD_Probit_effects <- cbind(as.matrix(Probit_effects)[,1], as.matrix(Probit_effects)[,3])
> SD_Probit_effects
      [,1]  [,2]
 [1,] " -5" "0.009260736"
 [2,] "  0" "0.009031807"
```

```
 [3,] "   5" "0.008826000"
 [4,] " 10" "0.008644967"
 [5,] " 15" "0.008490291"
 [6,] " 20" "0.008363436"
 [7,] " 25" "0.008265683"
 [8,] " 30" "0.008198072"
 [9,] " 35" "0.008161353"
[10,] " 40" "0.008155944"
[11,] " 45" "0.008181905"
[12,] " 50" "0.008238942"
[13,] " 55" "0.008326414"
[14,] " 60" "0.008443377"
[15,] " 65" "0.008588625"
[16,] " 70" "0.008760751"
[17,] " 75" "0.008958208"
[18,] " 80" "0.009179359"
[19,] " 85" "0.009422537"
[20,] " 90" "0.009686083"
[21,] " 95" "0.009968382"
[22,] "100" "0.010267886"
[23,] "105" "0.010583136"
> #Logit
> SD_Logit_effects <- cbind(as.matrix(Logit_effects)[,1], as.matrix(Logit_effects)[,3])
> SD_Logit_effects
      [,1]   [,2]
 [1,] " -5" "0.01493298"
 [2,] "   0" "0.01456228"
 [3,] "   5" "0.01422872"
 [4,] " 10" "0.01393498"
 [5,] " 15" "0.01368362"
 [6,] " 20" "0.01347702"
 [7,] " 25" "0.01331725"
 [8,] " 30" "0.01320602"
 [9,] " 35" "0.01314455"
[10,] " 40" "0.01313356"
[11,] " 45" "0.01317315"
[12,] " 50" "0.01326289"
[13,] " 55" "0.01340176"
[14,] " 60" "0.01358826"
[15,] " 65" "0.01382046"
[16,] " 70" "0.01409610"
```

```
[17,] " 75" "0.01441268"
[18,] " 80" "0.01476758"
[19,] " 85" "0.01515811"
[20,] " 90" "0.01558158"
[21,] " 95" "0.01603539"
[22,] "100" "0.01651704"
[23,] "105" "0.01702415"
>
```