A4

Shiqi Zhou

4/6/2022

```
#set path
getwd()
```

```
## [1] "/Users/zhoushiqi/Desktop/613/A4/A4.R"
```

```
mainpath = "/Users/zhoushiqi/Desktop/613/A4/A4.R"
setwd(mainpath)
getwd()
```

```
## [1] "/Users/zhoushiqi/Desktop/613/A4/A4.R"
```

```
datapath = "/Data.A4"

#import datasets
DATA <- list.files(paste0(mainpath,datapath))
for (i in 1:length(DATA)) {
  assign(sub(".csv","",DATA[i]), fread(paste0(paste0(mainpath,datapath),"/",
                                              DATA[i]),encoding = "UTF-8"))
}
```

Exercise 1 Preparing the Data

```
names(dat_A4)
```

```
##  [1] "V1"                        "X"
##  [3] "PUBID_1997"                "KEY_SEX_1997"
##  [5] "KEY_BDATE_M_1997"          "KEY_BDATE_Y_1997"
##  [7] "CV_SAMPLE_TYPE_1997"       "CV_HGC_BIO_DAD_1997"
##  [9] "CV_HGC_BIO_MOM_1997"       "CV_HGC_RES_DAD_1997"
## [11] "CV_HGC_RES_MOM_1997"       "KEY_RACE_ETHNICITY_1997"
## [13] "TRANS_SAT_MATH_HSTR"       "CV_HH_SIZE_2019"
## [15] "CV_MARSTAT_COLLAPSED_2019" "CV_BIO_CHILD_HH_U18_2019"
## [17] "CV_URBAN.RURAL_2019"       "CV_WKSWK_JOB_DLI.01_2019"
## [19] "CV_WKSWK_JOB_DLI.02_2019"  "CV_WKSWK_JOB_DLI.03_2019"
## [21] "CV_WKSWK_JOB_DLI.04_2019"  "CV_WKSWK_JOB_DLI.05_2019"
## [23] "CV_WKSWK_JOB_DLI.06_2019"  "CV_WKSWK_JOB_DLI.07_2019"
## [25] "CV_WKSWK_JOB_DLI.08_2019"  "CV_WKSWK_JOB_DLI.09_2019"
## [27] "CV_WKSWK_JOB_DLI.10_2019"  "CV_WKSWK_JOB_DLI.11_2019"
## [29] "YSCH.3113_2019"            "YINC_1700_2019"
```

```
dat_A4=select(dat_A4,-V1,)
names(dat_A4_panel)
```

```
##   [1] "V1"                              "PUBID_1997"
```

```
##   [3] "YINC-1700_1997"                "KEY_SEX_1997"
##   [5] "KEY_BDATE_M_1997"              "KEY_BDATE_Y_1997"
##   [7] "CV_MARSTAT_COLLAPSED_1997"     "CV_WKSWK_JOB_DLI.01_1997"
##   [9] "CV_WKSWK_JOB_DLI.02_1997"      "CV_WKSWK_JOB_DLI.03_1997"
##  [11] "CV_WKSWK_JOB_DLI.04_1997"      "CV_WKSWK_JOB_DLI.05_1997"
##  [13] "CV_WKSWK_JOB_DLI.06_1997"      "CV_WKSWK_JOB_DLI.07_1997"
##  [15] "CV_SAMPLE_TYPE_1997"           "KEY_RACE_ETHNICITY_1997"
##  [17] "YINC-1700_1998"                "CV_HIGHEST_DEGREE_9899_1998"
##  [19] "CV_MARSTAT_COLLAPSED_1998"     "CV_WKSWK_JOB_DLI.01_1998"
##  [21] "CV_WKSWK_JOB_DLI.02_1998"      "CV_WKSWK_JOB_DLI.03_1998"
##  [23] "CV_WKSWK_JOB_DLI.04_1998"      "CV_WKSWK_JOB_DLI.05_1998"
##  [25] "CV_WKSWK_JOB_DLI.06_1998"      "CV_WKSWK_JOB_DLI.07_1998"
##  [27] "CV_WKSWK_JOB_DLI.08_1998"      "CV_WKSWK_JOB_DLI.09_1998"
##  [29] "YINC-1700_1999"                "CV_HIGHEST_DEGREE_9900_1999"
##  [31] "CV_MARSTAT_COLLAPSED_1999"     "CV_WKSWK_JOB_DLI.01_1999"
##  [33] "CV_WKSWK_JOB_DLI.02_1999"      "CV_WKSWK_JOB_DLI.03_1999"
##  [35] "CV_WKSWK_JOB_DLI.04_1999"      "CV_WKSWK_JOB_DLI.05_1999"
##  [37] "CV_WKSWK_JOB_DLI.06_1999"      "CV_WKSWK_JOB_DLI.07_1999"
##  [39] "CV_WKSWK_JOB_DLI.08_1999"      "CV_WKSWK_JOB_DLI.09_1999"
##  [41] "YINC-1700_2000"                "CV_HIGHEST_DEGREE_0001_2000"
##  [43] "CV_MARSTAT_COLLAPSED_2000"     "CV_WKSWK_JOB_DLI.01_2000"
##  [45] "CV_WKSWK_JOB_DLI.02_2000"      "CV_WKSWK_JOB_DLI.03_2000"
##  [47] "CV_WKSWK_JOB_DLI.04_2000"      "CV_WKSWK_JOB_DLI.05_2000"
##  [49] "CV_WKSWK_JOB_DLI.06_2000"      "CV_WKSWK_JOB_DLI.07_2000"
##  [51] "CV_WKSWK_JOB_DLI.08_2000"      "CV_WKSWK_JOB_DLI.09_2000"
##  [53] "YINC-1700_2001"                "CV_HIGHEST_DEGREE_0102_2001"
##  [55] "CV_MARSTAT_COLLAPSED_2001"     "CV_WKSWK_JOB_DLI.01_2001"
##  [57] "CV_WKSWK_JOB_DLI.02_2001"      "CV_WKSWK_JOB_DLI.03_2001"
##  [59] "CV_WKSWK_JOB_DLI.04_2001"      "CV_WKSWK_JOB_DLI.05_2001"
##  [61] "CV_WKSWK_JOB_DLI.06_2001"      "CV_WKSWK_JOB_DLI.07_2001"
##  [63] "CV_WKSWK_JOB_DLI.08_2001"      "YINC-1700_2002"
##  [65] "CV_HIGHEST_DEGREE_0203_2002"   "CV_MARSTAT_COLLAPSED_2002"
##  [67] "CV_WKSWK_JOB_DLI.01_2002"      "CV_WKSWK_JOB_DLI.02_2002"
##  [69] "CV_WKSWK_JOB_DLI.03_2002"      "CV_WKSWK_JOB_DLI.04_2002"
##  [71] "CV_WKSWK_JOB_DLI.05_2002"      "CV_WKSWK_JOB_DLI.06_2002"
##  [73] "CV_WKSWK_JOB_DLI.07_2002"      "CV_WKSWK_JOB_DLI.08_2002"
##  [75] "CV_WKSWK_JOB_DLI.09_2002"      "CV_WKSWK_JOB_DLI.10_2002"
##  [77] "CV_WKSWK_JOB_DLI.11_2002"      "CV_HIGHEST_DEGREE_0304_2003"
##  [79] "CV_MARSTAT_COLLAPSED_2003"     "CV_WKSWK_JOB_DLI.01_2003"
##  [81] "CV_WKSWK_JOB_DLI.02_2003"      "CV_WKSWK_JOB_DLI.03_2003"
##  [83] "CV_WKSWK_JOB_DLI.04_2003"      "CV_WKSWK_JOB_DLI.05_2003"
##  [85] "CV_WKSWK_JOB_DLI.06_2003"      "CV_WKSWK_JOB_DLI.07_2003"
##  [87] "CV_WKSWK_JOB_DLI.08_2003"      "CV_WKSWK_JOB_DLI.09_2003"
##  [89] "CV_WKSWK_JOB_DLI.10_2003"      "YINC-1700_2003"
##  [91] "CV_HIGHEST_DEGREE_0405_2004"   "CV_MARSTAT_COLLAPSED_2004"
##  [93] "CV_WKSWK_JOB_DLI.01_2004"      "CV_WKSWK_JOB_DLI.02_2004"
##  [95] "CV_WKSWK_JOB_DLI.03_2004"      "CV_WKSWK_JOB_DLI.04_2004"
##  [97] "CV_WKSWK_JOB_DLI.05_2004"      "CV_WKSWK_JOB_DLI.06_2004"
##  [99] "CV_WKSWK_JOB_DLI.07_2004"      "YINC-1700_2004"
## [101] "CV_HIGHEST_DEGREE_0506_2005"   "CV_MARSTAT_COLLAPSED_2005"
## [103] "CV_WKSWK_JOB_DLI.01_2005"      "CV_WKSWK_JOB_DLI.02_2005"
## [105] "CV_WKSWK_JOB_DLI.03_2005"      "CV_WKSWK_JOB_DLI.04_2005"
## [107] "CV_WKSWK_JOB_DLI.05_2005"      "CV_WKSWK_JOB_DLI.06_2005"
## [109] "CV_WKSWK_JOB_DLI.07_2005"      "CV_WKSWK_JOB_DLI.08_2005"
```

```
## [111] "CV_WKSWK_JOB_DLI.09_2005"         "YINC-1700_2005"
## [113] "CV_HIGHEST_DEGREE_0607_2006"      "CV_MARSTAT_COLLAPSED_2006"
## [115] "CV_WKSWK_JOB_DLI.01_2006"         "CV_WKSWK_JOB_DLI.02_2006"
## [117] "CV_WKSWK_JOB_DLI.03_2006"         "CV_WKSWK_JOB_DLI.04_2006"
## [119] "CV_WKSWK_JOB_DLI.05_2006"         "CV_WKSWK_JOB_DLI.06_2006"
## [121] "CV_WKSWK_JOB_DLI.07_2006"         "CV_WKSWK_JOB_DLI.08_2006"
## [123] "CV_WKSWK_JOB_DLI.09_2006"         "YINC-1700_2006"
## [125] "CV_HIGHEST_DEGREE_0708_2007"      "CV_MARSTAT_COLLAPSED_2007"
## [127] "CV_WKSWK_JOB_DLI.01_2007"         "CV_WKSWK_JOB_DLI.02_2007"
## [129] "CV_WKSWK_JOB_DLI.03_2007"         "CV_WKSWK_JOB_DLI.04_2007"
## [131] "CV_WKSWK_JOB_DLI.05_2007"         "CV_WKSWK_JOB_DLI.06_2007"
## [133] "CV_WKSWK_JOB_DLI.07_2007"         "CV_WKSWK_JOB_DLI.08_2007"
## [135] "YINC-1700_2007"                   "CV_HIGHEST_DEGREE_0809_2008"
## [137] "CV_MARSTAT_COLLAPSED_2008"        "CV_WKSWK_JOB_DLI.01_2008"
## [139] "CV_WKSWK_JOB_DLI.02_2008"         "CV_WKSWK_JOB_DLI.03_2008"
## [141] "CV_WKSWK_JOB_DLI.04_2008"         "CV_WKSWK_JOB_DLI.05_2008"
## [143] "CV_WKSWK_JOB_DLI.06_2008"         "CV_WKSWK_JOB_DLI.07_2008"
## [145] "CV_WKSWK_JOB_DLI.08_2008"         "YINC-1700_2008"
## [147] "CV_HIGHEST_DEGREE_0910_2009"      "CV_MARSTAT_COLLAPSED_2009"
## [149] "CV_WKSWK_JOB_DLI.01_2009"         "CV_WKSWK_JOB_DLI.02_2009"
## [151] "CV_WKSWK_JOB_DLI.03_2009"         "CV_WKSWK_JOB_DLI.04_2009"
## [153] "CV_WKSWK_JOB_DLI.05_2009"         "CV_WKSWK_JOB_DLI.06_2009"
## [155] "CV_WKSWK_JOB_DLI.07_2009"         "CV_WKSWK_JOB_DLI.08_2009"
## [157] "CV_WKSWK_JOB_DLI.09_2009"         "YINC-1700_2009"
## [159] "CV_HIGHEST_DEGREE_EVER_EDT_2010" "CV_HIGHEST_DEGREE_1011_2010"
## [161] "CV_MARSTAT_COLLAPSED_2010"        "CV_WKSWK_JOB_DLI.01_2010"
## [163] "CV_WKSWK_JOB_DLI.02_2010"         "CV_WKSWK_JOB_DLI.03_2010"
## [165] "CV_WKSWK_JOB_DLI.04_2010"         "CV_WKSWK_JOB_DLI.05_2010"
## [167] "CV_WKSWK_JOB_DLI.06_2010"         "CV_WKSWK_JOB_DLI.07_2010"
## [169] "CV_WKSWK_JOB_DLI.08_2010"         "CV_WKSWK_JOB_DLI.09_2010"
## [171] "YINC-1700_2010"                   "CV_HIGHEST_DEGREE_EVER_EDT_2011"
## [173] "CV_HIGHEST_DEGREE_1112_2011"      "CV_MARSTAT_COLLAPSED_2011"
## [175] "CV_WKSWK_JOB_DLI.01_2011"         "CV_WKSWK_JOB_DLI.02_2011"
## [177] "CV_WKSWK_JOB_DLI.03_2011"         "CV_WKSWK_JOB_DLI.04_2011"
## [179] "CV_WKSWK_JOB_DLI.05_2011"         "CV_WKSWK_JOB_DLI.06_2011"
## [181] "CV_WKSWK_JOB_DLI.07_2011"         "CV_WKSWK_JOB_DLI.08_2011"
## [183] "CV_WKSWK_JOB_DLI.09_2011"         "CV_WKSWK_JOB_DLI.10_2011"
## [185] "CV_WKSWK_JOB_DLI.11_2011"         "CV_WKSWK_JOB_DLI.12_2011"
## [187] "CV_WKSWK_JOB_DLI.13_2011"         "YINC-1700_2011"
## [189] "CV_HIGHEST_DEGREE_EVER_EDT_2013" "CV_HIGHEST_DEGREE_1314_2013"
## [191] "CV_MARSTAT_COLLAPSED_2013"        "CV_WKSWK_JOB_DLI.01_2013"
## [193] "CV_WKSWK_JOB_DLI.02_2013"         "CV_WKSWK_JOB_DLI.03_2013"
## [195] "CV_WKSWK_JOB_DLI.04_2013"         "CV_WKSWK_JOB_DLI.05_2013"
## [197] "CV_WKSWK_JOB_DLI.06_2013"         "CV_WKSWK_JOB_DLI.07_2013"
## [199] "CV_WKSWK_JOB_DLI.08_2013"         "CV_WKSWK_JOB_DLI.09_2013"
## [201] "CV_WKSWK_JOB_DLI.10_2013"         "YINC-1700_2013"
## [203] "CV_HIGHEST_DEGREE_EVER_EDT_2015" "CV_MARSTAT_COLLAPSED_2015"
## [205] "CV_WKSWK_JOB_DLI.01_2015"         "CV_WKSWK_JOB_DLI.02_2015"
## [207] "CV_WKSWK_JOB_DLI.03_2015"         "CV_WKSWK_JOB_DLI.04_2015"
## [209] "CV_WKSWK_JOB_DLI.05_2015"         "CV_WKSWK_JOB_DLI.06_2015"
## [211] "CV_WKSWK_JOB_DLI.07_2015"         "CV_WKSWK_JOB_DLI.08_2015"
## [213] "CV_WKSWK_JOB_DLI.09_2015"         "CV_WKSWK_JOB_DLI.10_2015"
## [215] "CV_WKSWK_JOB_DLI.11_2015"         "CV_WKSWK_JOB_DLI.12_2015"
## [217] "YINC-1700_2015"                   "CV_HIGHEST_DEGREE_EVER_EDT_2017"
```

```
## [219] "CV_MARSTAT_COLLAPSED_2017"       "CV_WKSWK_JOB_DLI.01_2017"
## [221] "CV_WKSWK_JOB_DLI.02_2017"        "CV_WKSWK_JOB_DLI.03_2017"
## [223] "CV_WKSWK_JOB_DLI.04_2017"        "CV_WKSWK_JOB_DLI.05_2017"
## [225] "CV_WKSWK_JOB_DLI.06_2017"        "CV_WKSWK_JOB_DLI.07_2017"
## [227] "CV_WKSWK_JOB_DLI.08_2017"        "CV_WKSWK_JOB_DLI.09_2017"
## [229] "CV_WKSWK_JOB_DLI.10_2017"        "CV_WKSWK_JOB_DLI.11_2017"
## [231] "CV_WKSWK_JOB_DLI.12_2017"        "CV_WKSWK_JOB_DLI.13_2017"
## [233] "CV_WKSWK_JOB_DLI.14_2017"        "CV_WKSWK_JOB_DLI.15_2017"
## [235] "YINC-1700_2017"                  "CV_HIGHEST_DEGREE_EVER_EDT_2019"
## [237] "CV_MARSTAT_COLLAPSED_2019"       "CV_WKSWK_JOB_DLI.01_2019"
## [239] "CV_WKSWK_JOB_DLI.02_2019"        "CV_WKSWK_JOB_DLI.03_2019"
## [241] "CV_WKSWK_JOB_DLI.04_2019"        "CV_WKSWK_JOB_DLI.05_2019"
## [243] "CV_WKSWK_JOB_DLI.06_2019"        "CV_WKSWK_JOB_DLI.07_2019"
## [245] "CV_WKSWK_JOB_DLI.08_2019"        "CV_WKSWK_JOB_DLI.09_2019"
## [247] "CV_WKSWK_JOB_DLI.10_2019"        "CV_WKSWK_JOB_DLI.11_2019"
## [249] "YINC-1700_2019"
```

```
dat_A4_panel=select(dat_A4_panel,-V1)
colnames(dat_A4_panel)[248]="YINC_1700_2019"
colnames(dat_A4_panel)[2]="YINC_1700_1997"
```

#1.1 Create additional variable for the age of the agent "age", total work experience measured in years "work exp".

```
#create "age" variables
dat = mutate(dat_A4, age_1997=1997-KEY_BDATE_Y_1997,
             age_2019=2019-KEY_BDATE_Y_1997)
#age in 1997
(count(group_by(dat,age_1997)))
```

```
## # A tibble: 5 x 2
## # Groups:   age_1997 [5]
##   age_1997     n
##      <dbl> <int>
## 1       13  1771
## 2       14  1807
## 3       15  1841
## 4       16  1874
## 5       17  1691
```

```
#age in 1997
(count(group_by(dat,age_2019)))
```

```
## # A tibble: 5 x 2
## # Groups:   age_2019 [5]
##   age_2019     n
##      <dbl> <int>
## 1       35  1771
## 2       36  1807
## 3       37  1841
## 4       38  1874
## 5       39  1691
```

```
#create work experience in year "work exp"

#first, create work time in weeks
a = dat[,c(1,17:27)]
```

```r
a[is.na(a)]<-0
a$work_exp_week = rowSums(a[,2:12])
a1 = a[,c(1,13)]
dat = left_join(dat,a1,by="X")
#then, translate it into years (assume that there are 52 weeks in a year)
dat$work_exp_years = dat$work_exp_week/52
```

#1.2 Create additional education variables indicating total years of schooling from all variables related to education.

```r
#all variables related to education
b = dat[,c(1,7:10,28)]
b$bio.fa.edu=ifelse(b$CV_HGC_BIO_DAD_1997==95,0,b$CV_HGC_BIO_DAD_1997)
b$bio.mo.edu=ifelse(b$CV_HGC_BIO_MOM_1997==95,0,b$CV_HGC_BIO_MOM_1997)
b$res.fa.edu=ifelse(b$CV_HGC_RES_DAD_1997==95,0,b$CV_HGC_RES_DAD_1997)
b$res.mo.edu=ifelse(b$CV_HGC_RES_MOM_1997==95,0,b$CV_HGC_RES_MOM_1997)

#translate the highest degree to schooling year
#GED equals to high school degree for 12 years(2,3);
#2 years for AA(4)=12+2;
#4 years for Bachelor(5)=12+4;
#take 2 years for Master and all have Bachelor degree(usually 1.5-2 years)=18
#for PHD and professional degree take them as 20 years or more
b$self.edu.2019=
  ifelse(b$YSCH.3113_2019==1,0,
         ifelse(b$YSCH.3113_2019==2,12,
                ifelse(b$YSCH.3113_2019==3,12,
                       ifelse(b$YSCH.3113_2019==4,14,
                              ifelse(b$YSCH.3113_2019==5,16,
                                     ifelse(b$YSCH.3113_2019==6,18,
                                            ifelse(b$YSCH.3113_2019==7,20,
                                                   ifelse(b$YSCH.3113_2019==8,
                                                          20,0))))))))
b1=b[,c(1,7:11)]
b1[is.na(b1)]<-0

#create the indicator for schooling year
b1$sy.edu.parents=rowSums(b1[,2:5])
b1$sy.edu.all=rowSums(b1[,2:6])

b2=b1[,c(1,7:8)]

dat = left_join(dat,b2,by="X")
```

#1.3 Provide the following visualizations.

```r
#set up dataset used in this problem,
#include income in panel data because censor problem
c=select(dat,PUBID_1997,YINC_1700_2019,age_1997,age_2019,KEY_SEX_1997,
         CV_MARSTAT_COLLAPSED_2019,CV_BIO_CHILD_HH_U18_2019)
c=filter(c,!is.na(YINC_1700_2019))
#the top-coded income is 1e+05
max(c$YINC_1700_2019)
```

```
## [1] 1e+05
```

```
#include income in panel data as YINC_1700_2019.y
u=select(dat_A4_panel,PUBID_1997,YINC_1700_2019)
c=left_join(c,u,by="PUBID_1997")
names(c)
```

```
## [1] "PUBID_1997"                "YINC_1700_2019.x"
## [3] "age_1997"                  "age_2019"
## [5] "KEY_SEX_1997"              "CV_MARSTAT_COLLAPSED_2019"
## [7] "CV_BIO_CHILD_HH_U18_2019"  "YINC_1700_2019.y"
```

```
#the real max income in panel data is 328451
max(c$YINC_1700_2019.y)
```

```
## [1] 328451
```

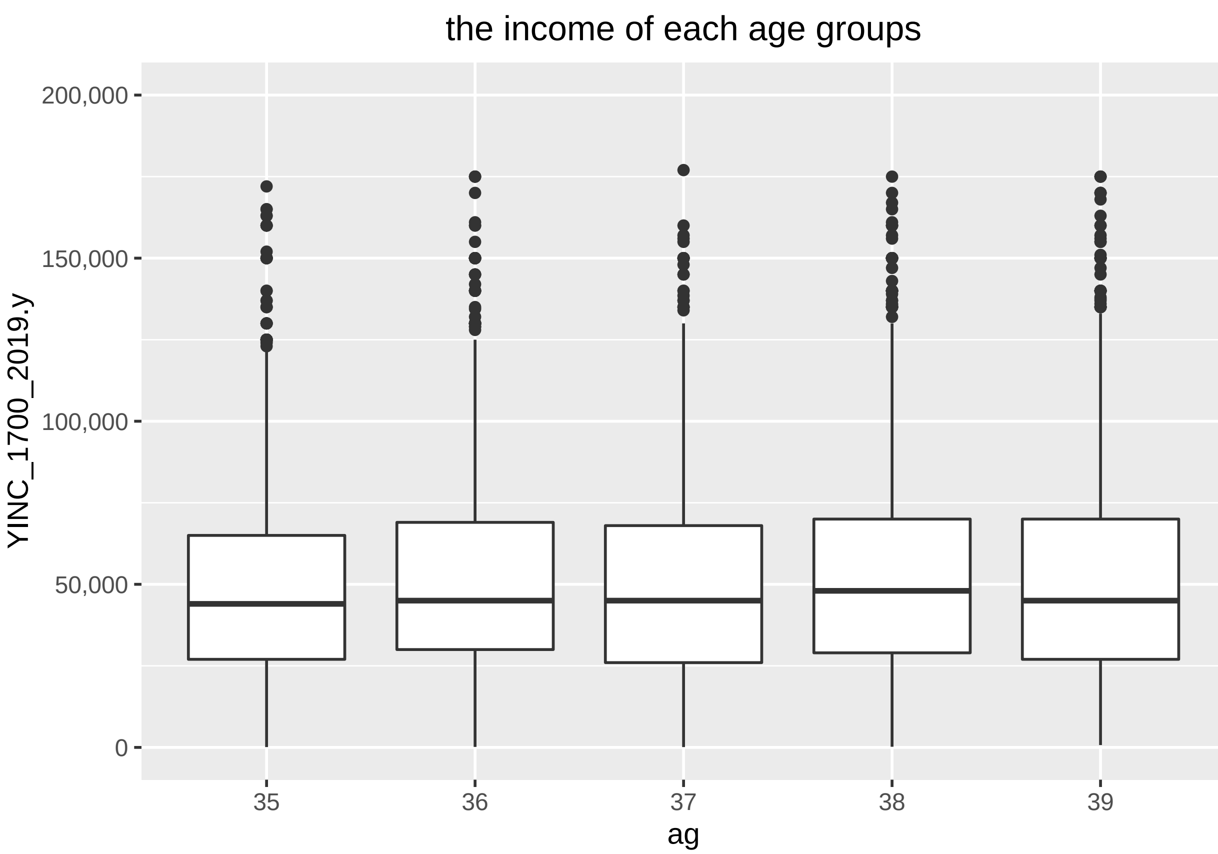
```
max(c$YINC_1700_2019.x)
```

```
## [1] 1e+05
```

```
#group the income variable
c <- mutate(c,income.group.x=case_when(YINC_1700_2019.x == 0  ~ "0",
                                       YINC_1700_2019.x >= 1 & YINC_1700_2019.x <= 4999 ~ "1-4999",
                                       YINC_1700_2019.x >= 5000 & YINC_1700_2019.x <= 9999 ~ "5000-9999",
                                       YINC_1700_2019.x >= 10000 & YINC_1700_2019.x <= 14999 ~ "10000-149
                                       YINC_1700_2019.x >= 15000 & YINC_1700_2019.x <= 19999 ~ "15000-199
                                       YINC_1700_2019.x >= 20000 & YINC_1700_2019.x <= 24999 ~ "20000-249
                                       YINC_1700_2019.x >= 25000 & YINC_1700_2019.x <= 29999 ~ "25000-299
                                       YINC_1700_2019.x >= 30000 & YINC_1700_2019.x <= 39999 ~ "30000-399
                                       YINC_1700_2019.x >= 40000 & YINC_1700_2019.x <= 49999 ~ "40000-499
                                       YINC_1700_2019.x >= 50000 & YINC_1700_2019.x <= 59999 ~ "50000-599
                                       YINC_1700_2019.x >= 60000 & YINC_1700_2019.x <= 69999 ~ "60000-699
                                       YINC_1700_2019.x >= 70000 & YINC_1700_2019.x <= 79999 ~ "70000-799
                                       YINC_1700_2019.x >= 80000 & YINC_1700_2019.x <= 89999 ~ "80000-899
                                       YINC_1700_2019.x >= 90000 & YINC_1700_2019.x <= 99999 ~ "90000-999
                                       YINC_1700_2019.x >= 100000 ~ "100000+"
                                       ))


c <- mutate(c,income.group.y=case_when(YINC_1700_2019.y == 0  ~ "0",
                                       YINC_1700_2019.y >= 1 & YINC_1700_2019.y <= 4999 ~ "1-4999",
                                       YINC_1700_2019.y >= 5000 & YINC_1700_2019.y <= 9999 ~ "5000-9999",
                                       YINC_1700_2019.y >= 10000 & YINC_1700_2019.y <= 14999 ~ "10000-149
                                       YINC_1700_2019.y >= 15000 & YINC_1700_2019.y <= 19999 ~ "15000-199
                                       YINC_1700_2019.y >= 20000 & YINC_1700_2019.y <= 24999 ~ "20000-249
                                       YINC_1700_2019.y >= 25000 & YINC_1700_2019.y <= 29999 ~ "25000-299
                                       YINC_1700_2019.y >= 30000 & YINC_1700_2019.y <= 39999 ~ "30000-399
                                       YINC_1700_2019.y >= 40000 & YINC_1700_2019.y <= 49999 ~ "40000-499
                                       YINC_1700_2019.y >= 50000 & YINC_1700_2019.y <= 59999 ~ "50000-599
                                       YINC_1700_2019.y >= 60000 & YINC_1700_2019.y <= 69999 ~ "60000-699
                                       YINC_1700_2019.y >= 70000 & YINC_1700_2019.y <= 79999 ~ "70000-799
                                       YINC_1700_2019.y >= 80000 & YINC_1700_2019.y <= 89999 ~ "80000-899
                                       YINC_1700_2019.y >= 90000 & YINC_1700_2019.y <= 99999 ~ "90000-999
                                       YINC_1700_2019.y >= 100000 & YINC_1700_2019.y <= 149999 ~ "100000-
                                       YINC_1700_2019.y >= 150000 ~ "150000+"
                                       ))
c <- mutate(c,ag=as.factor(age_2019))
```

```
c <- mutate(c,gender=as.factor(KEY_SEX_1997))
c <- mutate(c,child.num=as.factor(CV_BIO_CHILD_HH_U18_2019))
c <- mutate(c,marital=as.factor(CV_MARSTAT_COLLAPSED_2019))

#1.3.1 Plot the income data (where income is positive) by
c1=filter(c,YINC_1700_2019.y>0)


#i) age groups
#bar chart
ggplot(c1, aes(x = ag, y = YINC_1700_2019.y)) +
  geom_boxplot() +
  scale_y_continuous(labels = label_comma(), limits = c(NA, 200000)) +
  ggtitle("the income of each age groups")+
  theme(plot.title=element_text(hjust=0.5))
```

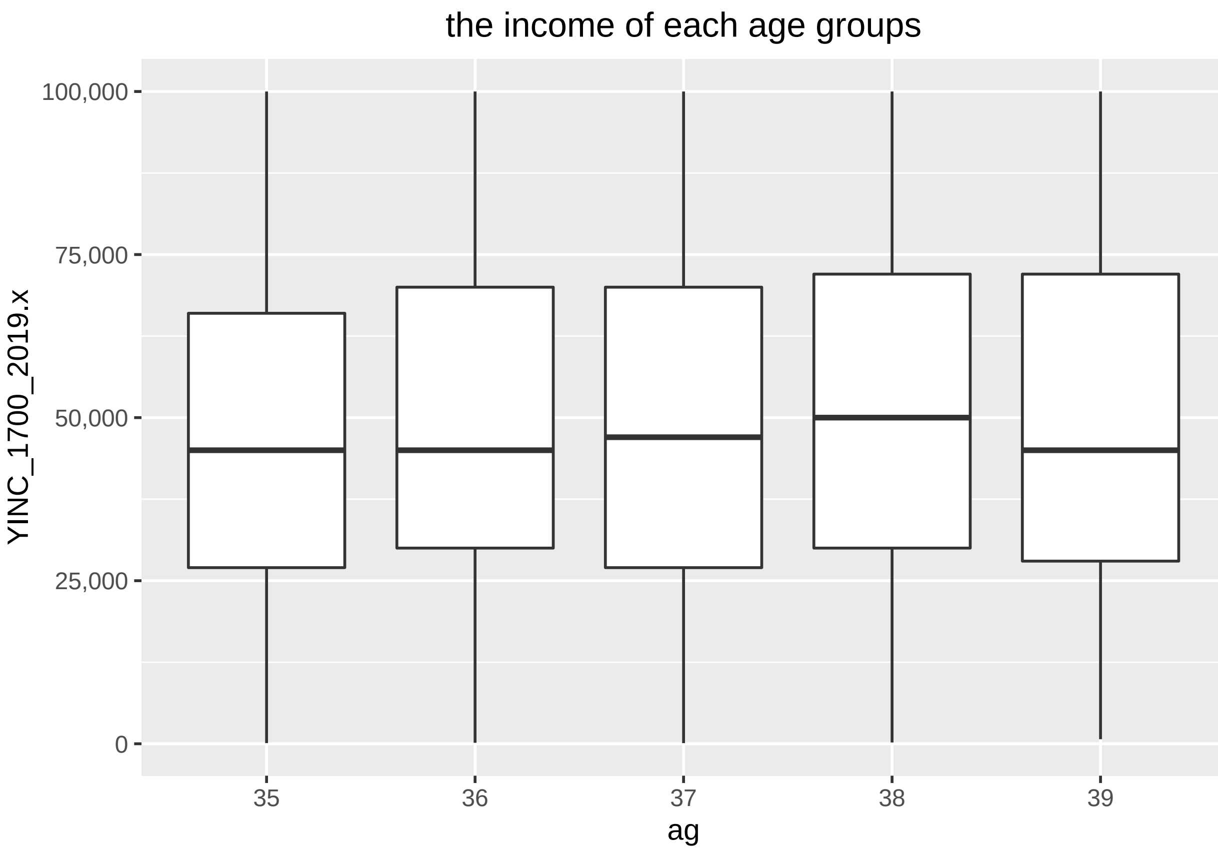
```
## Warning: Removed 120 rows containing non-finite values (stat_boxplot).
```



#Part 1 in 1.3: plot with income variable in cross section data where censor problem exist, then plot with income variable in panel data

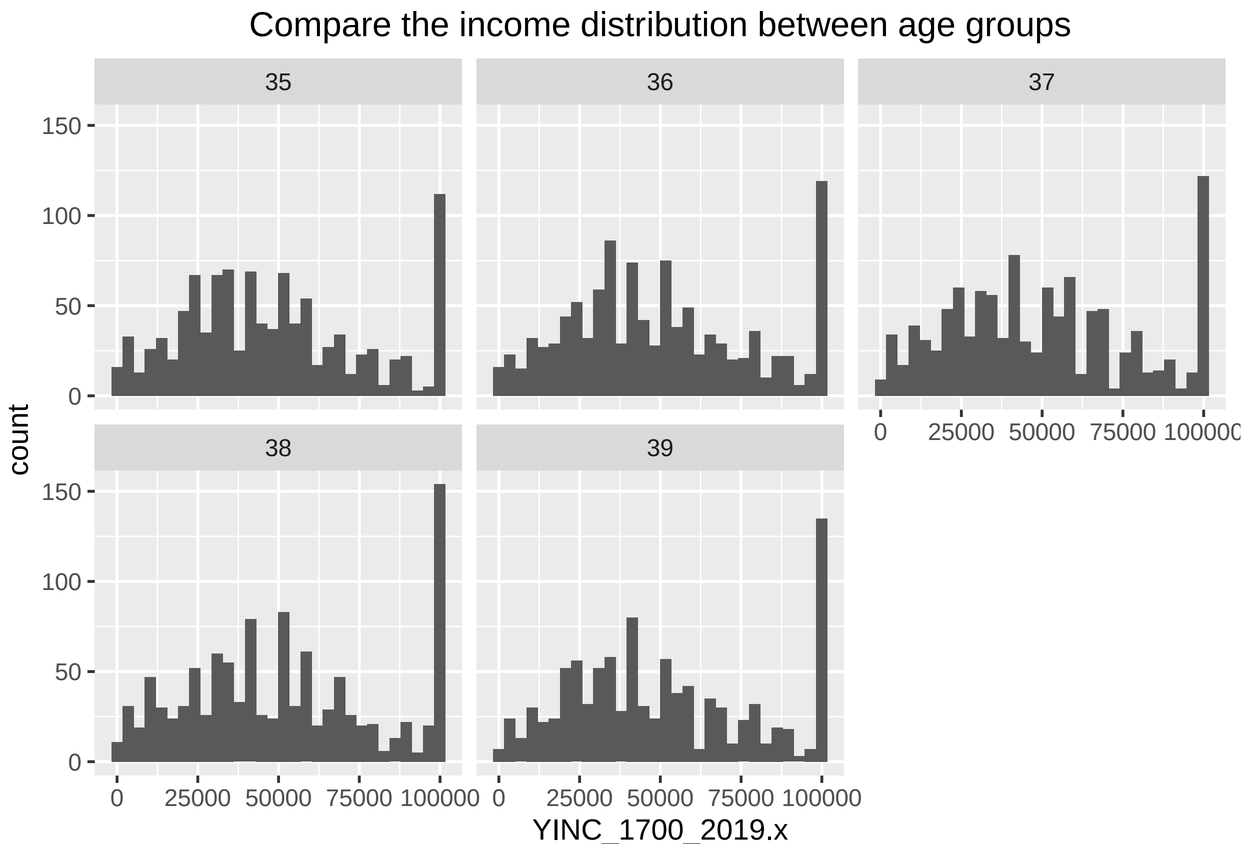
```
#1.3.1 Plot the income data (where income is positive) with income variable
#in cross section data where censor problem exist
c1=filter(c,YINC_1700_2019.x>0)

#i) age groups
#bar chart
```

```r
ggplot(c1, aes(x = ag, y = YINC_1700_2019.x)) +
  geom_boxplot() +
  scale_y_continuous(labels = label_comma(), limits = c(NA, 100000)) +
  ggtitle("the income of each age groups")+
  theme(plot.title=element_text(hjust=0.5))
```

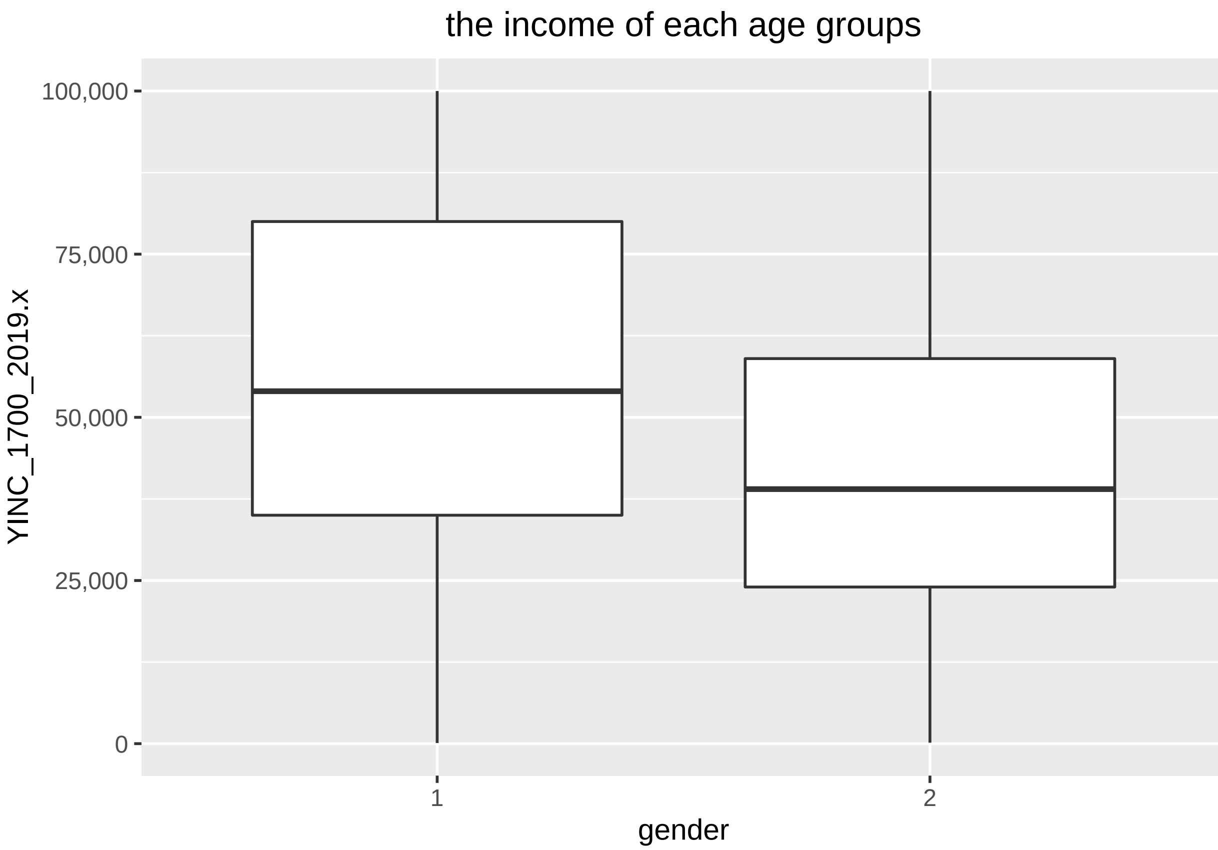



```r
#1.3.1 i) by age groups
#histogram
ggplot(c1, aes(x=YINC_1700_2019.x)) +
  geom_histogram()+
  facet_wrap( ~ag)+
  ggtitle("Compare the income distribution between age groups")+
  theme(plot.title=element_text(hjust=0.5))
```

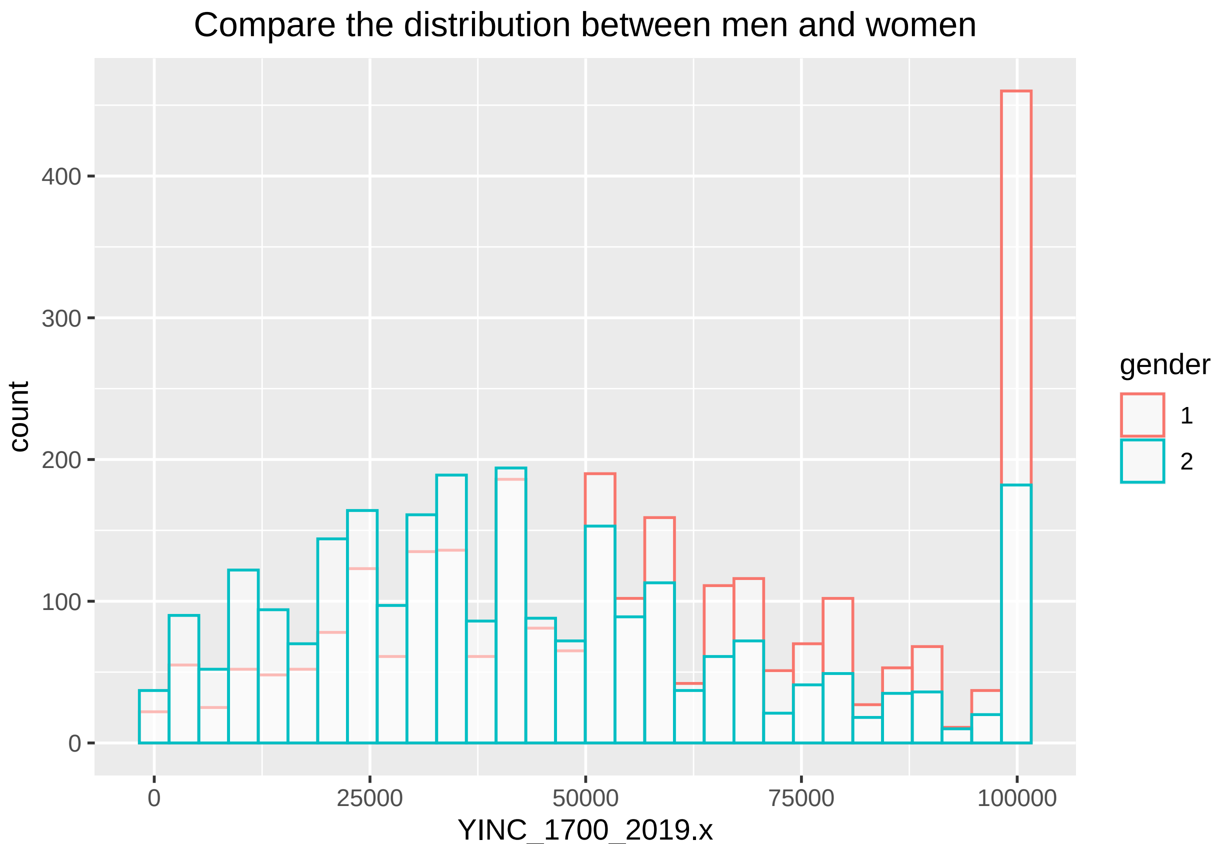
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



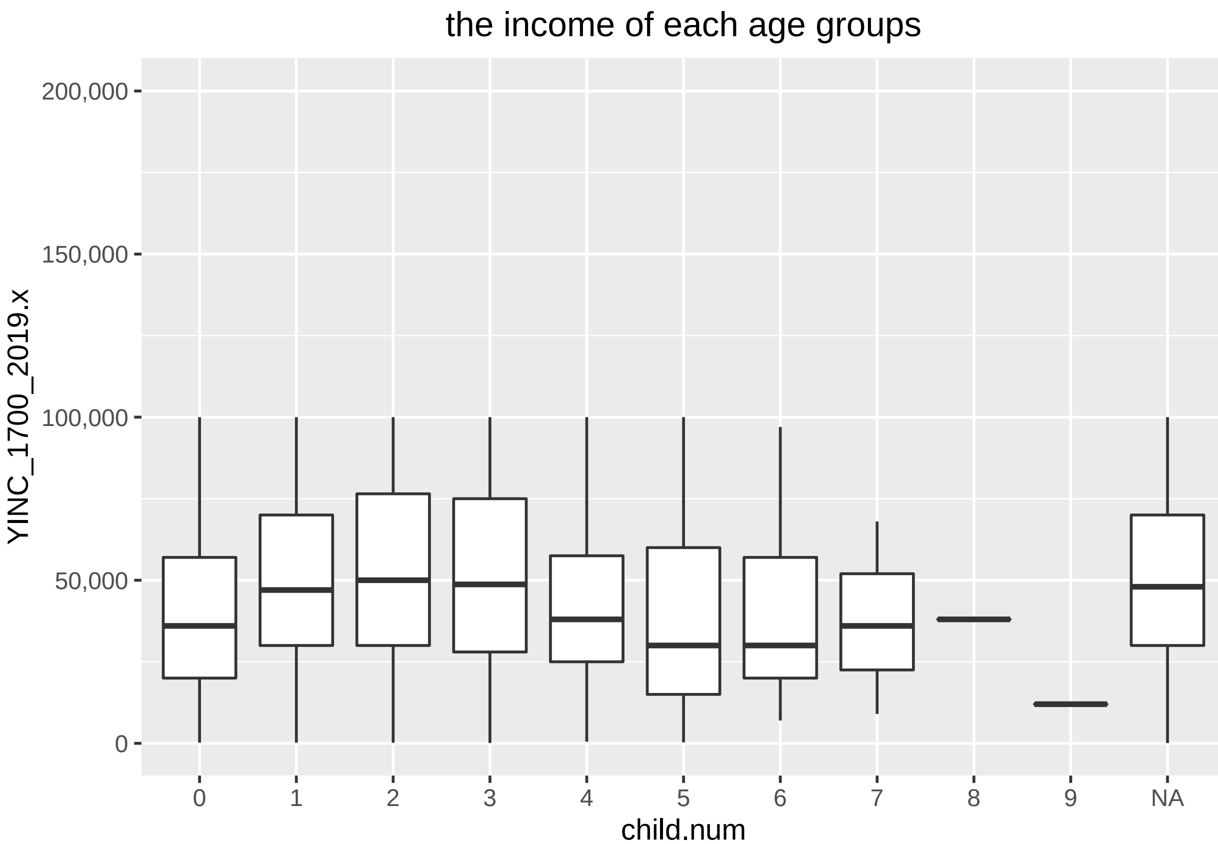

```
#1.3.1 ii) by gender groups and
#bar chart
ggplot(c1, aes(x = gender, y = YINC_1700_2019.x)) +
  geom_boxplot() +
  scale_y_continuous(labels = label_comma(), limits = c(NA, 100000)) +
  ggtitle("the income of each age groups")+
  theme(plot.title=element_text(hjust=0.5))
```




```
#1.3.1 ii) by gender groups and
#histogram
ggplot(c1, aes(x=YINC_1700_2019.x, color=gender)) +
  geom_histogram(fill="white", alpha=0.5,bins=30,
                 position="identity")+
  ggtitle("Compare the distribution between men and women")+
  theme(plot.title=element_text(hjust=0.5))
```

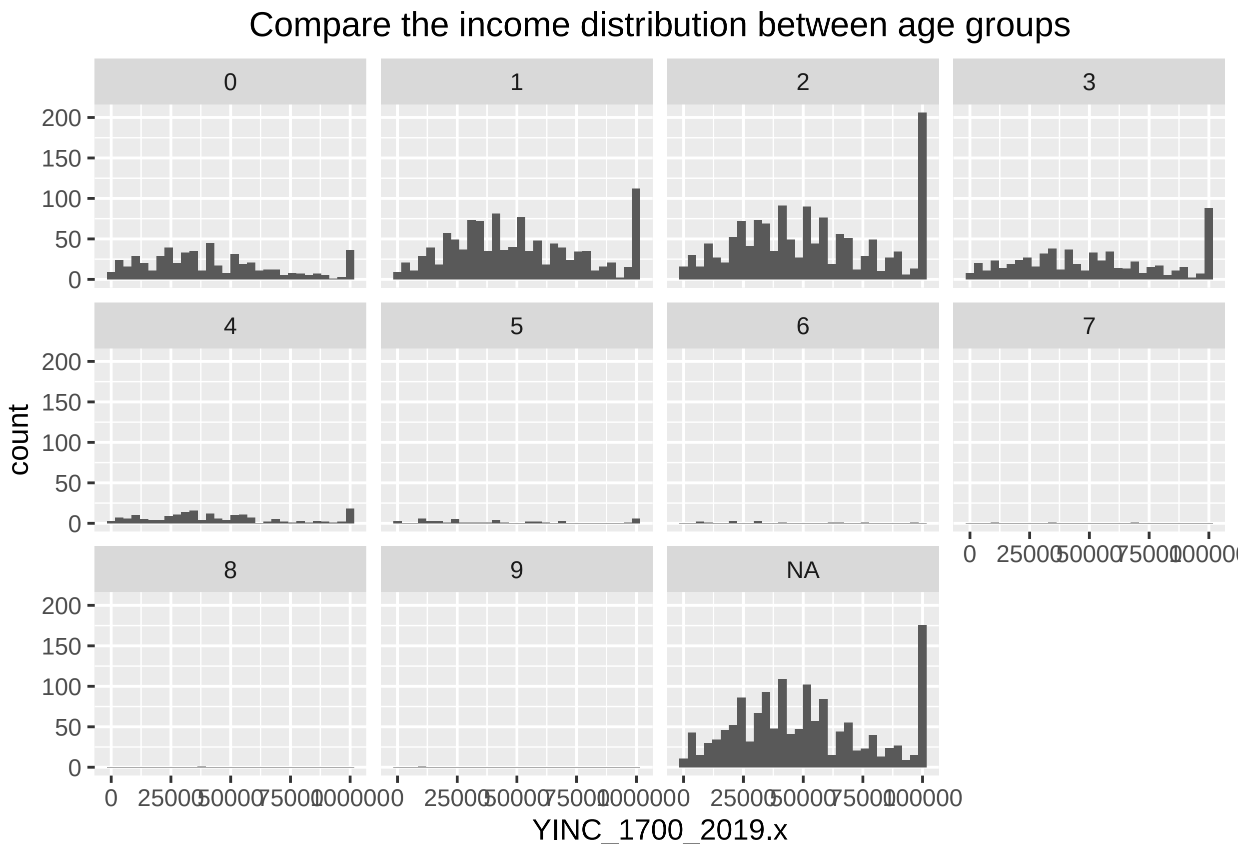



```
#1.3.1 iii) by number of children
#bar chart
ggplot(c1, aes(x = child.num, y = YINC_1700_2019.x)) +
  geom_boxplot() +
  scale_y_continuous(labels = label_comma(), limits = c(NA, 200000)) +
  ggtitle("the income of each age groups")+
  theme(plot.title=element_text(hjust=0.5))
```




```r
#1.3.1 iii) by number of children
#histogram
ggplot(c1, aes(x=YINC_1700_2019.x)) +
  geom_histogram()+
  facet_wrap( ~child.num)+
  ggtitle("Compare the income distribution between age groups")+
  theme(plot.title=element_text(hjust=0.5))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
#1.3.2 Table the share of "0" in the income data by
c1.1=filter(c,YINC_1700_2019.x==0)
#i) age groups
(count(group_by(c1.1,age_2019)))
```

```
## # A tibble: 5 x 2
## # Groups:   age_2019 [5]
##   age_2019     n
##      <dbl> <int>
## 1       35    10
## 2       36     7
## 3       37     6
## 4       38    10
## 5       39     3
```

```
#ii) gender groups
(count(group_by(c1.1,KEY_SEX_1997)))
```

```
## # A tibble: 2 x 2
## # Groups:   KEY_SEX_1997 [2]
##   KEY_SEX_1997     n
##          <int> <int>
## 1            1    21
## 2            2    15
```

```
#iii) number of children and marital status
(count(group_by(c1.1,CV_MARSTAT_COLLAPSED_2019,CV_BIO_CHILD_HH_U18_2019)))
```

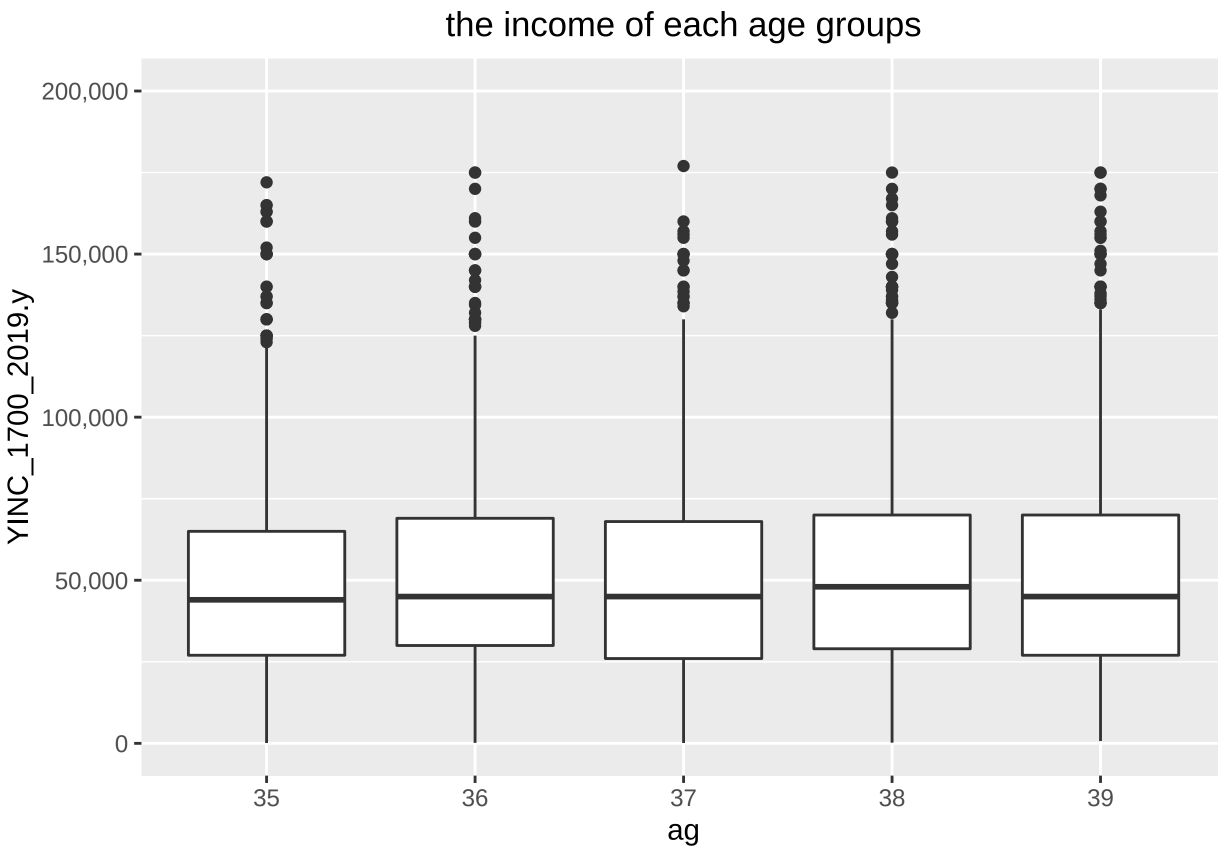
```
## # A tibble: 11 x 3
```

```
## # Groups:   CV_MARSTAT_COLLAPSED_2019, CV_BIO_CHILD_HH_U18_2019 [11]
##    CV_MARSTAT_COLLAPSED_2019 CV_BIO_CHILD_HH_U18_2019     n
##                        <int>                    <int> <int>
##  1                         0                        1     4
##  2                         0                        3     2
##  3                         0                       NA     5
##  4                         1                        0     4
##  5                         1                        1     5
##  6                         1                        2     8
##  7                         1                        3     2
##  8                         1                       NA     1
##  9                         2                        0     3
## 10                         2                        3     1
## 11                         3                        0     1
```

#Part 1 in 1.3: then plot with income variable in panel data

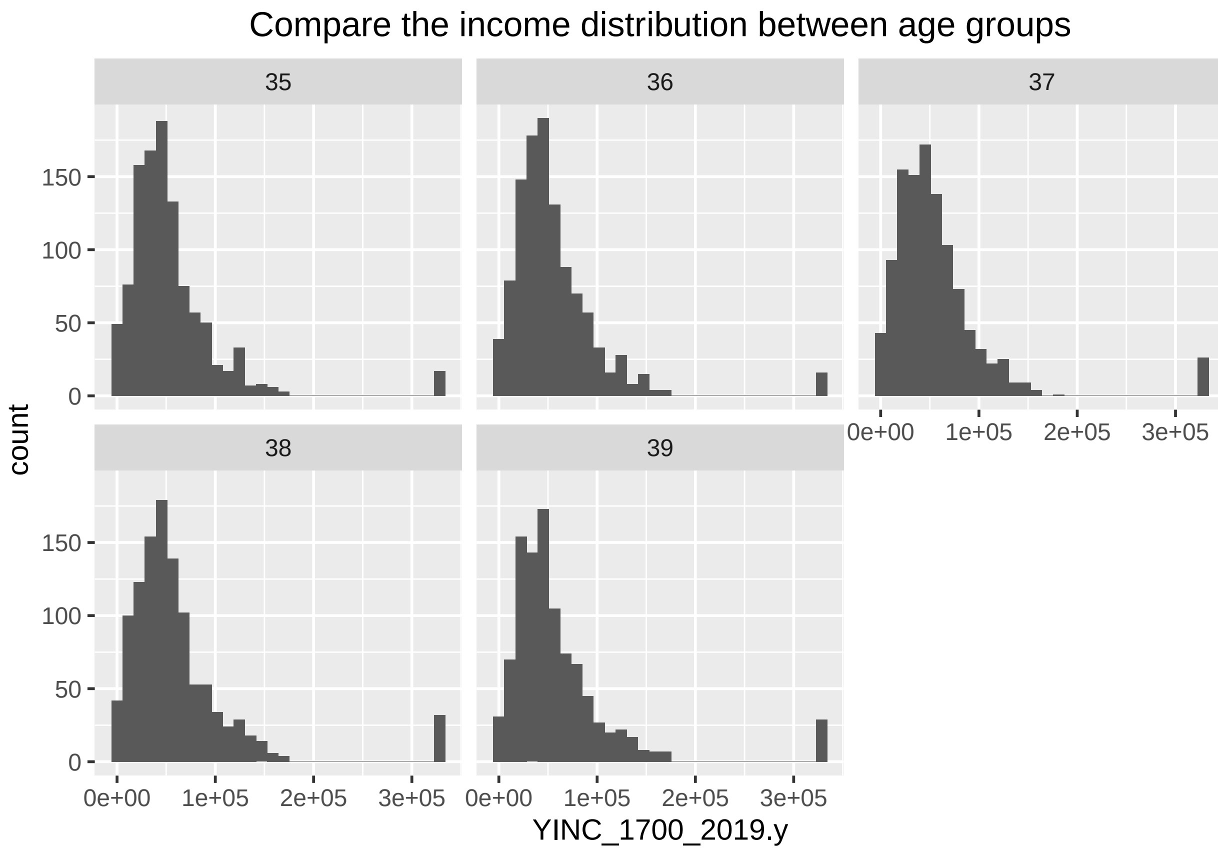
```
#1.3.1 Plot the income data (where income is positive)
c2=filter(c,YINC_1700_2019.y>0)

#i) age groups
#bar chart
ggplot(c2, aes(x = ag, y = YINC_1700_2019.y)) +
  geom_boxplot() +
  scale_y_continuous(labels = label_comma(), limits = c(NA, 200000)) +
  ggtitle("the income of each age groups")+
  theme(plot.title=element_text(hjust=0.5))
```

```
## Warning: Removed 120 rows containing non-finite values (stat_boxplot).
```



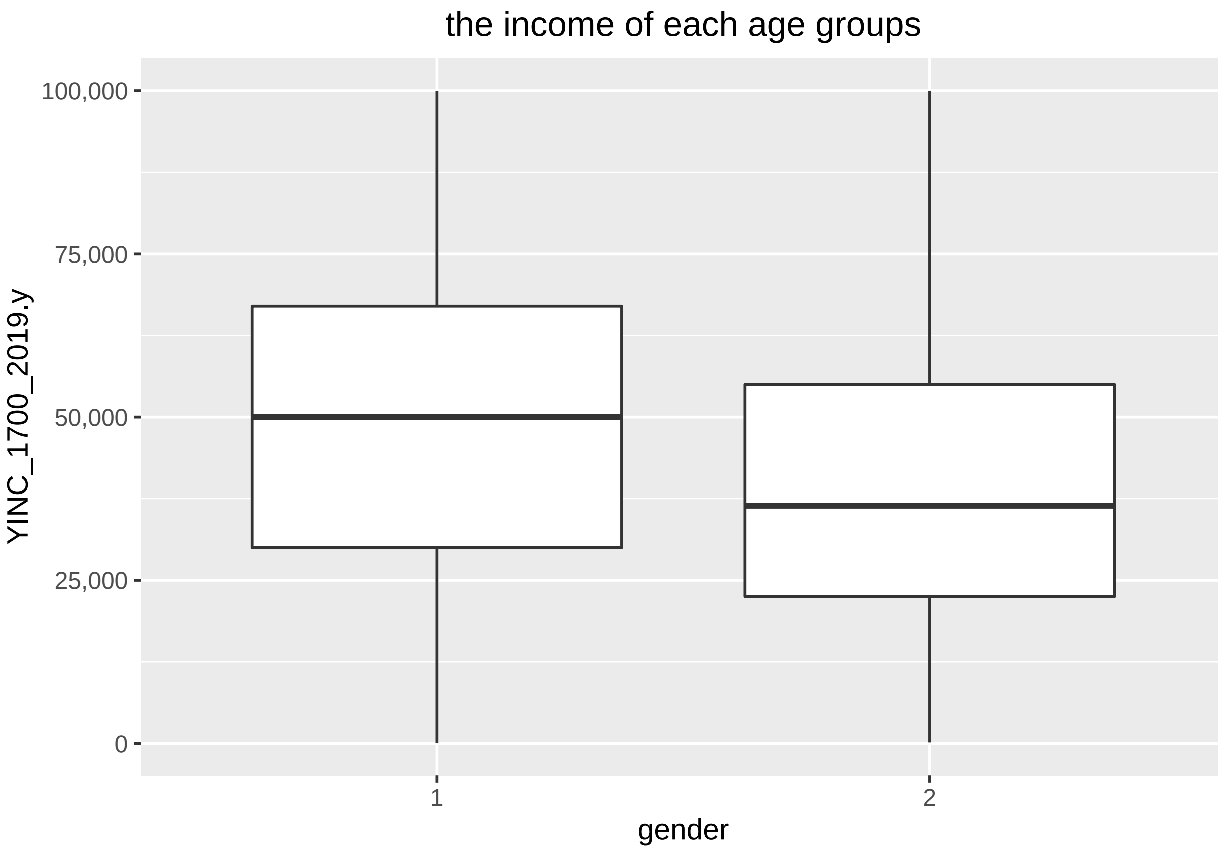

```r
#1.3.1 i) by age groups
#histogram
ggplot(c2, aes(x=YINC_1700_2019.y)) +
  geom_histogram()+
  facet_wrap( ~ag)+
  ggtitle("Compare the income distribution between age groups")+
  theme(plot.title=element_text(hjust=0.5))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

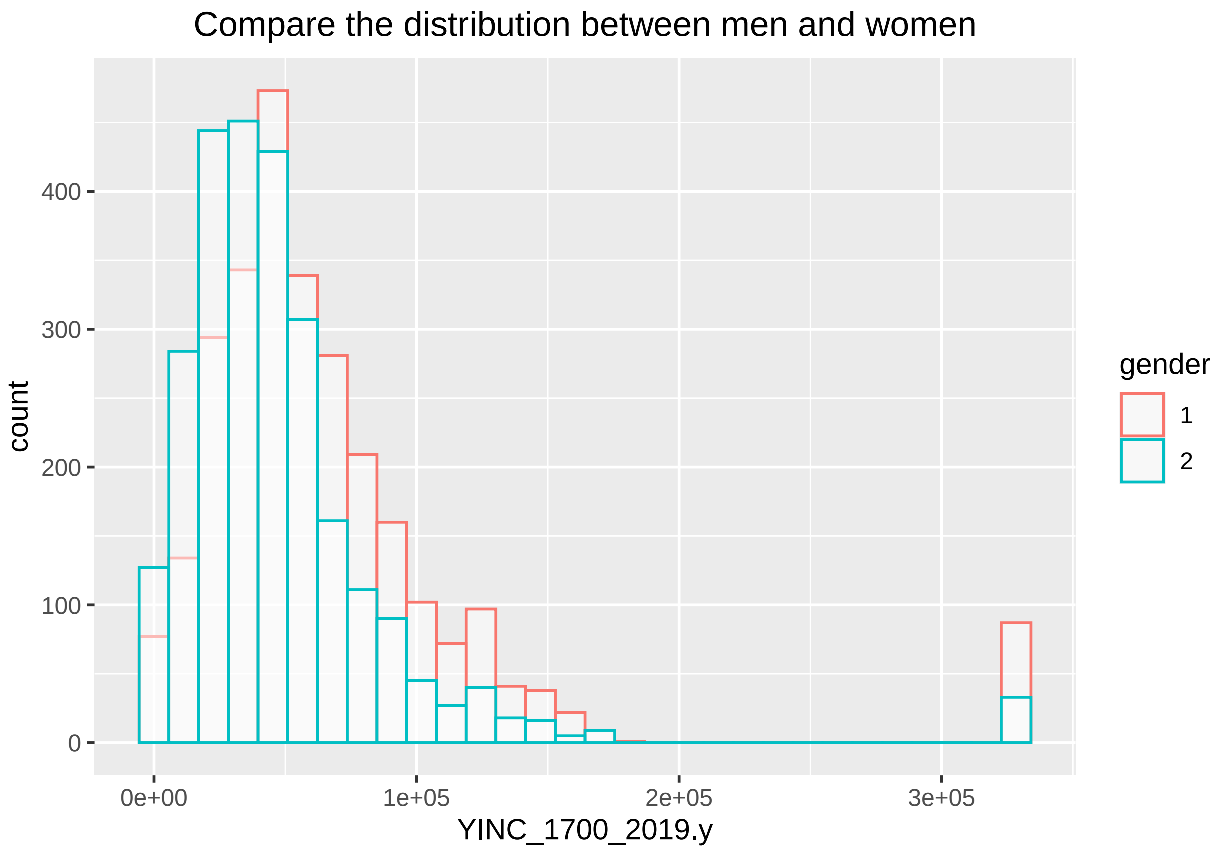



```
#1.3.1 ii) by gender groups and
#bar chart
ggplot(c2, aes(x = gender, y = YINC_1700_2019.y)) +
  geom_boxplot() +
  scale_y_continuous(labels = label_comma(), limits = c(NA, 100000)) +
  ggtitle("the income of each age groups")+
  theme(plot.title=element_text(hjust=0.5))
```

```
## Warning: Removed 562 rows containing non-finite values (stat_boxplot).
```

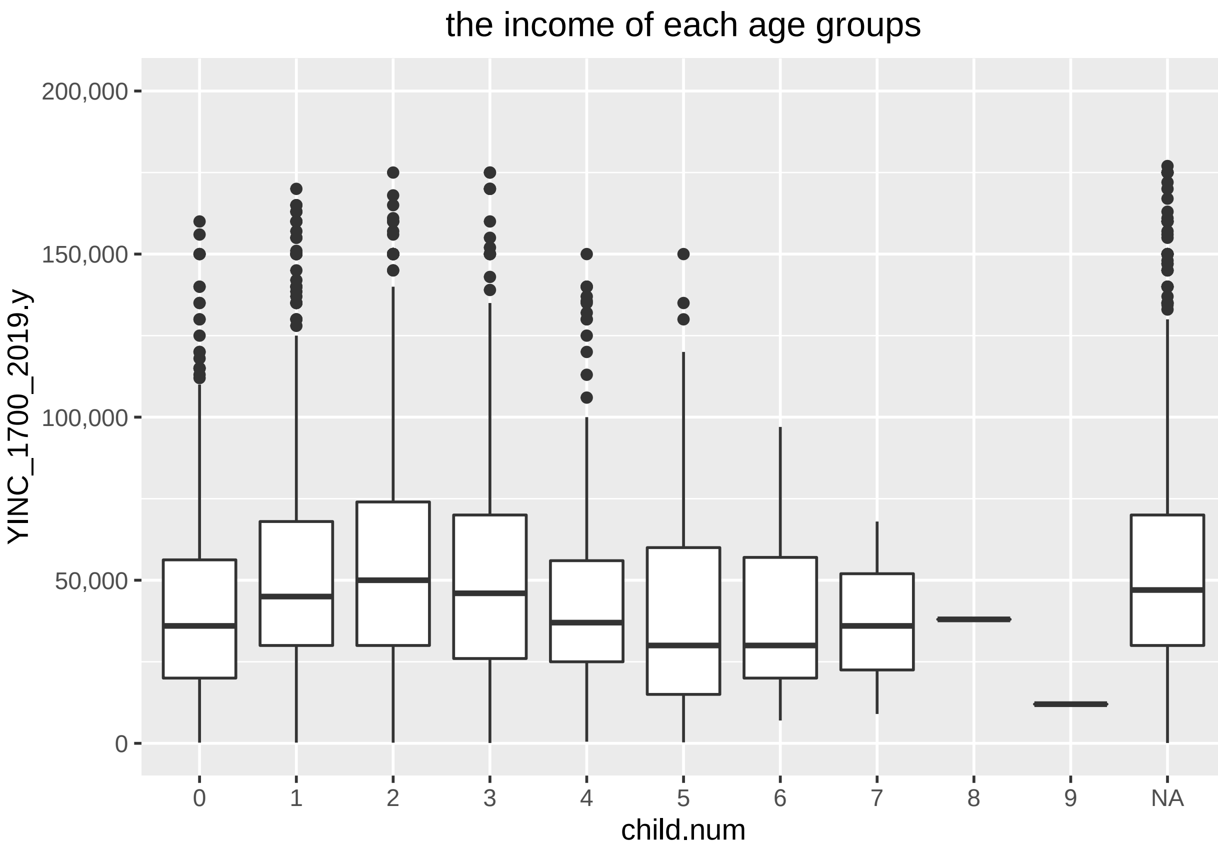



```
#1.3.1 ii) by gender groups and
#histogram
ggplot(c2, aes(x=YINC_1700_2019.y, color=gender)) +
  geom_histogram(fill="white", alpha=0.5,bins=30,
                 position="identity")+
  ggtitle("Compare the distribution between men and women")+
  theme(plot.title=element_text(hjust=0.5))
```



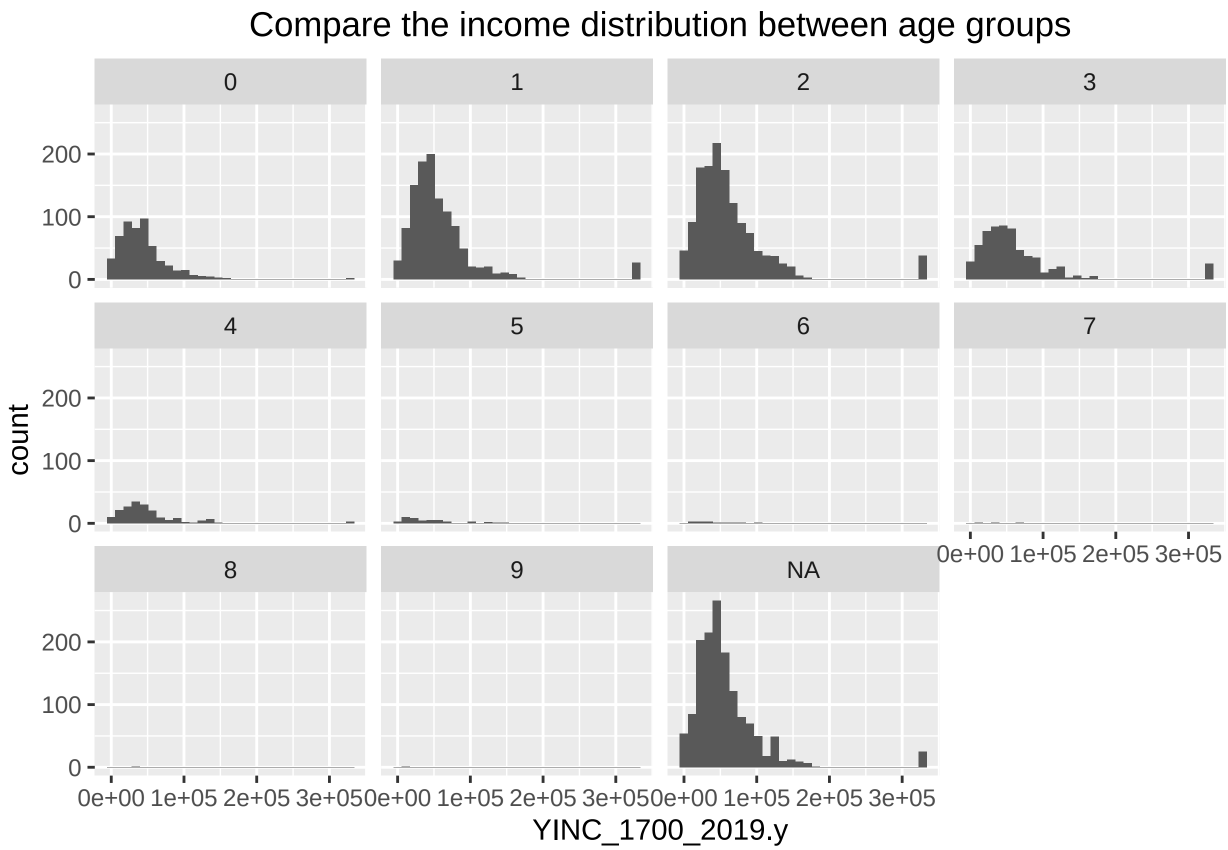

```
#1.3.1 iii) by number of children
#bar chart
ggplot(c2, aes(x = child.num, y = YINC_1700_2019.y)) +
  geom_boxplot() +
  scale_y_continuous(labels = label_comma(), limits = c(NA, 200000)) +
  ggtitle("the income of each age groups")+
  theme(plot.title=element_text(hjust=0.5))
```

```
## Warning: Removed 120 rows containing non-finite values (stat_boxplot).
```




```
#1.3.1 iii) by number of children
#histogram
ggplot(c2, aes(x=YINC_1700_2019.y)) +
  geom_histogram()+
  facet_wrap( ~child.num)+
  ggtitle("Compare the income distribution between age groups")+
  theme(plot.title=element_text(hjust=0.5))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
#1.3.2 Table the share of "0" in the income data by
c2.1=filter(c,YINC_1700_2019.y==0)
#i) age groups
(count(group_by(c2.1,age_2019)))
```

```
## # A tibble: 5 x 2
## # Groups:   age_2019 [5]
##   age_2019     n
##      <dbl> <int>
## 1       35    10
## 2       36     7
## 3       37     6
## 4       38    10
## 5       39     3
```

```
#ii) gender groups
(count(group_by(c2.1,KEY_SEX_1997)))
```

```
## # A tibble: 2 x 2
## # Groups:   KEY_SEX_1997 [2]
##   KEY_SEX_1997     n
##          <int> <int>
## 1            1    21
## 2            2    15
```

```
#iii) number of children and marital status
(count(group_by(c2.1,CV_MARSTAT_COLLAPSED_2019,CV_BIO_CHILD_HH_U18_2019)))
```

```
## # A tibble: 11 x 3
```

```
## # Groups:   CV_MARSTAT_COLLAPSED_2019, CV_BIO_CHILD_HH_U18_2019 [11]
##    CV_MARSTAT_COLLAPSED_2019 CV_BIO_CHILD_HH_U18_2019     n
##                        <int>                    <int> <int>
##  1                         0                        1     4
##  2                         0                        3     2
##  3                         0                       NA     5
##  4                         1                        0     4
##  5                         1                        1     5
##  6                         1                        2     8
##  7                         1                        3     2
##  8                         1                       NA     1
##  9                         2                        0     3
## 10                         2                        3     1
## 11                         3                        0     1
```

#1.3.3 interpret the visualizations from above #Interpret: #For positive income: both in dat and dat.panel #Generally, the average income increases as age increases; #the income of male is higher than that of female; #the average income increases with numbers of children in hh, then decreases. The highest average income is at the 2 children hh. #With censoring, the number of people with 10000 income in male is much larger than that in female.

#For "0" income hh: #almost same numbers in different age; as well as for gender; #most "0" income hh are married with 1 children.

Exercise 2 Heckman Selection Model

#2.1 Specify and estimate an OLS model to explain the income variable (where income is positive)

```
#set up dataset in this part (include income/age/gender/exper/edu/marital statu)
d=select(dat,PUBID_1997,work_exp_years,sy.edu.parents,sy.edu.all)
d1=left_join(d,c,by="PUBID_1997")
names(d1)
```

```
##  [1] "PUBID_1997"                "work_exp_years"
##  [3] "sy.edu.parents"            "sy.edu.all"
##  [5] "YINC_1700_2019.x"          "age_1997"
##  [7] "age_2019"                  "KEY_SEX_1997"
##  [9] "CV_MARSTAT_COLLAPSED_2019" "CV_BIO_CHILD_HH_U18_2019"
## [11] "YINC_1700_2019.y"          "income.group.x"
## [13] "income.group.y"            "ag"
## [15] "gender"                    "child.num"
## [17] "marital"
```

```
summary(d1$YINC_1700_2019.x)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##       0   28000   45000   49838   70000  100000    3572
```

```
summary(d1$YINC_1700_2019.y)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##       0   28000   45000   57217   70000  328451    3572
```

```
#only positive income
d2.1=filter(d1,YINC_1700_2019.x>0)
d2.2=filter(d1,YINC_1700_2019.y>0)
#to explain income variable, we use ln(income) here, and all education variables included in sy.edu.all
```

```r
d2.1$ln.income.x = log(d2.1$YINC_1700_2019.x)
d2.2$ln.income.y = log(d2.2$YINC_1700_2019.y)

#ols with sy.edu.all with ln(YINC_1700_2019.x) where max(income) is 100000 (censoring)
olsmodel.lnincome2.1 <- lm(ln.income.x~ag+gender+marital+work_exp_years+sy.edu.all,data=d2.1)

#ols with income where max(income) is 100000 (censoring)
olsmodel.income2.1 <- lm(YINC_1700_2019.x~ag+gender+marital+work_exp_years+sy.edu.all,data=d2.1)

#ols with sy.edu.all with ln(YINC_1700_2019.y) where max(income) is 328451
olsmodel.lnincome2.2 <- lm(ln.income.y~ag+gender+marital+work_exp_years+sy.edu.all,data=d2.2)

#ols with income where max(income) is 328451
olsmodel.income2.2 <- lm(YINC_1700_2019.y~ag+gender+marital+work_exp_years+sy.edu.all,data=d2.2)
```

```r
#ols with sy.edu.all with ln(YINC_1700_2019.y) where max(income) is 328451
olsmodel.income2.2 <- lm(ln.income.y~ag+gender+marital+work_exp_years+sy.edu.all,data=d2.2)
summary(olsmodel.income2.2)
```

```
##
## Call:
## lm(formula = ln.income.y ~ ag + gender + marital + work_exp_years +
##     sy.edu.all, data = d2.2)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -5.645 -0.325  0.098  0.477  2.637
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     9.7362678  0.0480038 202.823  < 2e-16 ***
## ag36            0.0396031  0.0363624   1.089  0.27615
## ag37            0.0203243  0.0364424   0.558  0.57707
## ag38            0.0234460  0.0364864   0.643  0.52051
## ag39            0.0517902  0.0375531   1.379  0.16791
## gender2        -0.3576395  0.0231667 -15.438  < 2e-16 ***
## marital1        0.2448677  0.0257299   9.517  < 2e-16 ***
## marital2       -0.1189530  0.0918920  -1.294  0.19555
## marital3        0.1447072  0.0384553   3.763  0.00017 ***
## marital4       -0.2550810  0.1951272  -1.307  0.19118
## work_exp_years  0.0365001  0.0021599  16.899  < 2e-16 ***
## sy.edu.all      0.0114418  0.0006474  17.672  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8447 on 5344 degrees of freedom
##   (20 observations deleted due to missingness)
## Multiple R-squared:  0.1745, Adjusted R-squared:  0.1728
## F-statistic: 102.7 on 11 and 5344 DF,  p-value: < 2.2e-16
```

#2.1.1 Interpret the estimation results

```r
#=============
#ln(income.x)~ag+gender+marital+work_exp_years+sy.edu.all
#=============
```

```r
summary(olsmodel.lnincome2.1)
```

```
##
## Call:
## lm(formula = ln.income.x ~ ag + gender + marital + work_exp_years +
##     sy.edu.all, data = d2.1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.6264 -0.2773  0.1469  0.4863  1.6907
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    9.7820348  0.0446577 219.045  < 2e-16 ***
## ag36           0.0405505  0.0338278   1.199    0.231
## ag37           0.0158416  0.0339022   0.467    0.640
## ag38           0.0069372  0.0339431   0.204    0.838
## ag39           0.0324010  0.0349354   0.927    0.354
## gender2       -0.3218831  0.0215518 -14.935  < 2e-16 ***
## marital1       0.2077246  0.0239364   8.678  < 2e-16 ***
## marital2      -0.1163761  0.0854867  -1.361    0.173
## marital3       0.1441208  0.0357748   4.029 5.69e-05 ***
## marital4      -0.2503629  0.1815260  -1.379    0.168
## work_exp_years  0.0356355  0.0020094  17.735  < 2e-16 ***
## sy.edu.all     0.0100553  0.0006023  16.694  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7859 on 5344 degrees of freedom
##   (20 observations deleted due to missingness)
## Multiple R-squared:  0.1678, Adjusted R-squared:  0.1661
## F-statistic: 97.94 on 11 and 5344 DF,  p-value: < 2.2e-16
```

#Interpret with the edu variable (include own highest degree) and with ln(YINC_1700_2019.x) where max(income) is 100000 (censoring) which is ln(olsmodel2.1). #From the estimation results, we find that: #if you are female, your income will decrease 32.2% compared with male; #one more year in work experience will increase 3.56% in income; #one more year in education in whole hh will increase 1.01% in income.

```r
#=============
#income.x~ag+gender+marital+work_exp_years+sy.edu.all
#=============
summary(olsmodel.income2.1)
```

```
##
## Call:
## lm(formula = YINC_1700_2019.x ~ ag + gender + marital + work_exp_years +
##     sy.edu.all, data = d2.1)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -70721 -18239  -2503  17729  78102
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    20331.92    1437.27  14.146  < 2e-16 ***
```

```
## ag36              1501.16    1088.72   1.379 0.168005
## ag37               695.92    1091.12   0.638 0.523631
## ag38              1493.35    1092.43   1.367 0.171685
## ag39              1224.36    1124.37   1.089 0.276236
## gender2         -12962.92     693.63 -18.689  < 2e-16 ***
## marital1          9667.27     770.38  12.549  < 2e-16 ***
## marital2          -105.56    2751.32  -0.038 0.969396
## marital3          4188.79    1151.38   3.638 0.000277 ***
## marital4         -5016.02    5842.28  -0.859 0.390614
## work_exp_years     997.27      64.67  15.421  < 2e-16 ***
## sy.edu.all         409.03      19.39  21.100  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 25290 on 5344 degrees of freedom
##   (20 observations deleted due to missingness)
## Multiple R-squared:  0.2131, Adjusted R-squared:  0.2115
## F-statistic: 131.6 on 11 and 5344 DF,  p-value: < 2.2e-16
```

#Interpret with the edu variable (include own highest degree) and with YINC_1700_2019.x where max(income) is 100000 (censoring) which is olsmodel2.1. #if you are female, your income will decrease -12962.92 compared with male; #one more year in work experience will increase 997.27 in income; #one more year in education in whole hh will increase 409.03 in income.

```
#=============
#ln(income.y)~ag+gender+marital+work_exp_years+sy.edu.all
#=============
summary(olsmodel.lnincome2.2)
```

```
##
## Call:
## lm(formula = ln.income.y ~ ag + gender + marital + work_exp_years +
##     sy.edu.all, data = d2.2)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -5.645 -0.325  0.098  0.477  2.637
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     9.7362678  0.0480038 202.823  < 2e-16 ***
## ag36            0.0396031  0.0363624   1.089  0.27615
## ag37            0.0203243  0.0364424   0.558  0.57707
## ag38            0.0234460  0.0364864   0.643  0.52051
## ag39            0.0517902  0.0375531   1.379  0.16791
## gender2        -0.3576395  0.0231667 -15.438  < 2e-16 ***
## marital1        0.2448677  0.0257299   9.517  < 2e-16 ***
## marital2       -0.1189530  0.0918920  -1.294  0.19555
## marital3        0.1447072  0.0384553   3.763  0.00017 ***
## marital4       -0.2550810  0.1951272  -1.307  0.19118
## work_exp_years  0.0365001  0.0021599  16.899  < 2e-16 ***
## sy.edu.all      0.0114418  0.0006474  17.672  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 0.8447 on 5344 degrees of freedom
##   (20 observations deleted due to missingness)
## Multiple R-squared:  0.1745, Adjusted R-squared:  0.1728
## F-statistic: 102.7 on 11 and 5344 DF,  p-value: < 2.2e-16
```

#Interpret with the edu variable (include own highest degree) and with ln(YINC_1700_2019.y) where max(income) is 328451 which is ln(olsmodel2.2). #if you are female, your income will decrease 35.8% compared with male; #one more year in work experience will increase 3.65% in income; #one more year in education in whole hh will increase 1.144% in income.

```r
#=============
#income.y~ag+gender+marital+work_exp_years+sy.edu.all
#=============
#Interpret with the edu variable (include own highest degree)
#and with ln(YINC_1700_2019.y) where max(income) is 328451 which is olsmodel2.2.
summary(olsmodel.income2.2)
```

```
##
## Call:
## lm(formula = ln.income.y ~ ag + gender + marital + work_exp_years +
##     sy.edu.all, data = d2.2)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -5.645 -0.325  0.098  0.477  2.637
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     9.7362678  0.0480038 202.823  < 2e-16 ***
## ag36            0.0396031  0.0363624   1.089  0.27615
## ag37            0.0203243  0.0364424   0.558  0.57707
## ag38            0.0234460  0.0364864   0.643  0.52051
## ag39            0.0517902  0.0375531   1.379  0.16791
## gender2        -0.3576395  0.0231667 -15.438  < 2e-16 ***
## marital1        0.2448677  0.0257299   9.517  < 2e-16 ***
## marital2       -0.1189530  0.0918920  -1.294  0.19555
## marital3        0.1447072  0.0384553   3.763  0.00017 ***
## marital4       -0.2550810  0.1951272  -1.307  0.19118
## work_exp_years  0.0365001  0.0021599  16.899  < 2e-16 ***
## sy.edu.all      0.0114418  0.0006474  17.672  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8447 on 5344 degrees of freedom
##   (20 observations deleted due to missingness)
## Multiple R-squared:  0.1745, Adjusted R-squared:  0.1728
## F-statistic: 102.7 on 11 and 5344 DF,  p-value: < 2.2e-16
```

#if you are female, your income will decrease -18725.21 compared with male; #one more year in work experience will increase 1120.48 in income; #one more year in education in whole hh will increase 629.4 in income.

#2.1.2 Explain why there might be a selection problem when estimating an OLS this way #selection problem may exist because the most efficient individuals have higher earnings and stay in school longer #only positive income is included in regression, and others with high education but not working are not included in regression sample.

#2.2 Explain why the Heckman model can deal with the selection problem.

#Use Heckman Two-Step Estimator to solve the selection problem produced by including only positive income #first, estimate a probit model, to estimate the probability that income is positive, #then calculate IMR, partly(x*beta/theta)/whole(x*beta/theta), to control the bias. #second, include IMR in OLS. #Through the Heckman model, we control the selection bias with the rate estimated in first step.

#2.3.1 Estimate a Heckman selection model. Interpret the results from the Heckman selection model

```
#(Note: You can not use a pre-programmed Heckman selection package.
#=================
#prepare data: for Heckman model
#considering OLS model: income~ag+gender+marital+work_exp_years+sy.edu.all
#and OLS model: ln(income)~ag+gender+marital+work_exp_years+sy.edu.all
#=================
d3=subset(d1,d1$CV_MARSTAT_COLLAPSED_2019!='NA'&d1$YINC_1700_2019.x!='NA')
#create indicator variable whether income equals to 0
d3=mutate(d3,ind.income.x=0,ind.income.y=0)
d3$ind.income.x[which(d3$YINC_1700_2019.x>0)] <- 1
d3$ind.income.y[which(d3$YINC_1700_2019.y>0)] <- 1
#create intersection variable
d3$intersection = 1
#create ln(income)
d3=mutate(d3,ln.income.x=log(d3$YINC_1700_2019.x),ln.income.y=log(d3$YINC_1700_2019.y))
d3$ln.income.x[which(d3$ln.income.x==-Inf)] <- 0
d3$ln.income.y[which(d3$ln.income.y==-Inf)] <- 0
#define income~ag+gender+marital+work_exp_years+sy.edu.all
income.ind.x=d3$ind.income.x

intsct=d3$intersection
age2019=as.numeric(d3$age_2019)
gender=as.numeric(d3$KEY_SEX_1997)
marital.status=d3$CV_MARSTAT_COLLAPSED_2019
wrk.exp.year=d3$work_exp_years
edu.all=d3$sy.edu.all
#=======================
#Heckman Two-Step Estimator
#=======================
#Step 1: Probit Estimation of Probability

set.seed(0)
#likelihood
probit_flike = function(par,x1,x2,x3,x4,x5,x6,y){
  yhat = par[1]*x1 + par[2]*x2 + par[3]*x3 + par[4]*x4 + par[5]*x5 + par[6]*x6
  prob = pnorm(yhat)
  like = y*log(prob) + (1-y)*log(1-prob)
  return(-sum(like))
}
#optimize
res  <- optim(runif(6,-0.1,0.1),fn=probit_flike,method="BFGS",
              control=list(trace=6,REPORT=1,maxit=1000),
              x1=intsct,x2=age2019,x3=gender,x4=marital.status,
              x5=wrk.exp.year,x6=edu.all,y=income.ind.x,hessian=TRUE)
```

```
## initial  value 68197.479943
## iter   2 value 21505.081976
```

```
## iter   3 value 21208.413257
## iter   4 value 21084.022422
## iter   5 value 21043.170530
## iter   6 value 20288.225925
## iter   7 value 497.687318
## iter   8 value 476.620734
## iter   9 value 366.737896
## iter  10 value 308.402436
## iter  11 value 292.755487
## iter  12 value 273.801548
## iter  13 value 254.127873
## iter  14 value 223.217122
## iter  15 value 216.746709
## iter  16 value 216.030568
## iter  17 value 215.704577
## iter  18 value 215.309679
## iter  19 value 215.148212
## iter  20 value 213.915596
## iter  21 value 213.769106
## iter  22 value 213.758760
## iter  22 value 213.758760
## iter  23 value 213.758547
## iter  24 value 213.757134
## iter  25 value 213.757076
## iter  25 value 213.757076
## iter  26 value 213.756585
## iter  26 value 213.756585
## iter  26 value 213.756585
## final  value 213.756585
## converged
```

```
res$par
```

```
## [1]  0.371667674  0.052784085  0.097800442  0.013646179  0.019212331
## [6] -0.002176258
```

```
#===========
#use glm()
probit.ind.income <- glm(income.ind.x~age2019+gender+
                           marital.status+wrk.exp.year+edu.all,
                         family =binomial(link = "probit"))
summary(probit.ind.income)
```

```
##
## Call:
## glm(formula = income.ind.x ~ age2019 + gender + marital.status +
##     wrk.exp.year + edu.all, family = binomial(link = "probit"))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.3356   0.0981   0.1134   0.1287   0.1796
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)      0.666882   1.623107   0.411    0.681
```

```
## age2019         0.045064   0.043595   1.034    0.301
## gender          0.097305   0.120974   0.804    0.421
## marital.status  0.013954   0.065152   0.214    0.830
## wrk.exp.year    0.019277   0.012716   1.516    0.130
## edu.all        -0.002368   0.003351  -0.706    0.480
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 432.42  on 5391  degrees of freedom
## Residual deviance: 427.48  on 5386  degrees of freedom
## AIC: 439.48
##
## Number of Fisher Scoring iterations: 8
```

```
probit.ind.income$coefficients
```

```
##    (Intercept)        age2019         gender marital.status   wrk.exp.year
##    0.666881897    0.045064333    0.097304723    0.013953652    0.019277212
##        edu.all
##   -0.002367501
```

```
#===========

#compute IMR
predict_fun = function(par,x1,x2,x3,x4,x5,x6,y){
  yhat = par[1]*x1 + par[2]*x2 + par[3]*x3 + par[4]*x4 + par[5]*x5 + par[6]*x6
  return(yhat)
}
#likelihood par
predictor.likeli <- predict_fun(res$par,intsct,age2019,gender,marital.status,
                                wrk.exp.year,edu.all,income.ind.x)
IMR.likeli <- dnorm(predictor.likeli)/pnorm(predictor.likeli)
#glm() par
predictor.glm <- predict_fun(probit.ind.income$coefficients,intsct,
                             age2019,gender,marital.status,wrk.exp.year,
                             edu.all,income.ind.x)
IMR.glm <- dnorm(predictor.glm)/pnorm(predictor.glm)


#Step 2: Include Inverse Mills Ratio as a Regressor
income.x=d3$YINC_1700_2019.x
ln.income.x=d3$ln.income.x
ols.heckman.income.x.likeli <- lm(income.x~age2019+gender+marital.status+
                                    wrk.exp.year+edu.all+IMR.likeli)
ols.heckman.income.x.glm <- lm(income.x~age2019+gender+marital.status+
                                 wrk.exp.year+edu.all+IMR.glm)
ols.heckman.lnincome.x.likeli <- lm(ln.income.x~age2019+gender+marital.status+wrk.exp.year+edu.all+IMR.l
ols.heckman.lnincome.x.glm <- lm(ln.income.x~age2019+gender+marital.status+
                                   wrk.exp.year+edu.all+IMR.glm)

#Interpret the Heckman results
summary(ols.heckman.income.x.likeli)
```

```
##
## Call:
```

```
## lm(formula = income.x ~ age2019 + gender + marital.status + wrk.exp.year +
##     edu.all + IMR.likeli)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -79213 -18717  -3115  18227  74471
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     2.107e+05  3.362e+04   6.266 3.99e-10 ***
## age2019        -3.560e+03  7.207e+02  -4.940 8.04e-07 ***
## gender         -2.067e+04  1.453e+03 -14.225  < 2e-16 ***
## marital.status  6.043e+02  4.142e+02   1.459    0.145
## wrk.exp.year   -2.419e+02  2.279e+02  -1.061    0.289
## edu.all         6.117e+02  3.362e+01  18.195  < 2e-16 ***
## IMR.likeli     -1.671e+06  2.796e+05  -5.979 2.39e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 25800 on 5385 degrees of freedom
## Multiple R-squared:  0.192,  Adjusted R-squared:  0.1911
## F-statistic: 213.2 on 6 and 5385 DF,  p-value: < 2.2e-16
```

```
summary(ols.heckman.lnincome.x.likeli)
```

```
##
## Call:
## lm(formula = ln.income.x ~ age2019 + gender + marital.status +
##     wrk.exp.year + edu.all + IMR.likeli)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.3786  -0.2394   0.1946   0.5620   1.7370
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     1.529e+01  1.519e+00  10.064  < 2e-16 ***
## age2019        -1.062e-01  3.256e-02  -3.261  0.00112 **
## gender         -5.355e-01  6.566e-02  -8.157 4.25e-16 ***
## marital.status  1.792e-02  1.872e-02   0.957  0.33841
## wrk.exp.year    7.046e-04  1.030e-02   0.068  0.94545
## edu.all         1.542e-02  1.519e-03  10.155  < 2e-16 ***
## IMR.likeli     -5.017e+01  1.263e+01  -3.973 7.20e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.165 on 5385 degrees of freedom
## Multiple R-squared:  0.08184,    Adjusted R-squared:  0.08082
## F-statistic:    80 on 6 and 5385 DF,  p-value: < 2.2e-16
```

```
summary(ols.heckman.income.x.glm)
```

```
##
## Call:
## lm(formula = income.x ~ age2019 + gender + marital.status + wrk.exp.year +
```

```
##     edu.all + IMR.glm)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -79434 -18669  -3116  18197  74652
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      197146.4    31662.9   6.226 5.13e-10 ***
## age2019           -3154.9      661.7  -4.768 1.91e-06 ***
## gender           -21032.3     1514.6 -13.887  < 2e-16 ***
## marital.status      530.1      420.3   1.261    0.207
## wrk.exp.year       -318.3      241.8  -1.317    0.188
## edu.all             635.3       37.1  17.121  < 2e-16 ***
## IMR.glm        -1759929.8   296462.5  -5.936 3.09e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 25800 on 5385 degrees of freedom
## Multiple R-squared:  0.1919, Adjusted R-squared:  0.191
## F-statistic: 213.1 on 6 and 5385 DF,  p-value: < 2.2e-16
```

```
summary(ols.heckman.lnincome.x.glm)
```

```
##
## Call:
## lm(formula = ln.income.x ~ age2019 + gender + marital.status +
##     wrk.exp.year + edu.all + IMR.glm)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.3851  -0.2398   0.1952   0.5634   1.7365
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     14.861870   1.430494  10.389  < 2e-16 ***
## age2019         -0.093602   0.029894  -3.131  0.00175 **
## gender          -0.545456   0.068427  -7.971  1.9e-15 ***
## marital.status   0.015813   0.018987   0.833  0.40498
## wrk.exp.year    -0.001437   0.010924  -0.132  0.89536
## edu.all          0.016111   0.001676   9.611  < 2e-16 ***
## IMR.glm        -52.636459  13.393861  -3.930  8.6e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.166 on 5385 degrees of freedom
## Multiple R-squared:  0.08178,    Adjusted R-squared:  0.08076
## F-statistic: 79.94 on 6 and 5385 DF,  p-value: < 2.2e-16
```

```
#almost the same while using likelihood function and glm in step 1
#only gender and edu are significantly correlated with income,
#work experience and marital are not correalated to income
```

#2.3.2 compare the results to OLS results. Why does there exist a difference?

```r
#Only compare with YINC_1700_2019.x with censoring problem (use glm results in Heckman)
#OLS model: income.x~ag+gender+marital+work_exp_years+sy.edu.all
summary(ols.heckman.income.x.glm)
```

```
##
## Call:
## lm(formula = income.x ~ age2019 + gender + marital.status + wrk.exp.year +
##     edu.all + IMR.glm)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -79434 -18669  -3116  18197  74652
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      197146.4    31662.9   6.226 5.13e-10 ***
## age2019           -3154.9      661.7  -4.768 1.91e-06 ***
## gender           -21032.3     1514.6 -13.887  < 2e-16 ***
## marital.status      530.1      420.3   1.261    0.207
## wrk.exp.year       -318.3      241.8  -1.317    0.188
## edu.all             635.3       37.1  17.121  < 2e-16 ***
## IMR.glm        -1759929.8   296462.5  -5.936 3.09e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 25800 on 5385 degrees of freedom
## Multiple R-squared:  0.1919, Adjusted R-squared:  0.191
## F-statistic: 213.1 on 6 and 5385 DF,  p-value: < 2.2e-16
```

```r
OLS.income.x <- lm(income.x~age2019+gender+marital.status+wrk.exp.year+edu.all)
summary(OLS.income.x)
```

```
##
## Call:
## lm(formula = income.x ~ age2019 + gender + marital.status + wrk.exp.year +
##     edu.all)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -78242 -18613  -3038  18369  76095
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     17809.09    9512.54   1.872   0.0612 .
## age2019           473.18     254.43   1.860   0.0630 .
## gender         -13073.67     706.96 -18.493  < 2e-16 ***
## marital.status   1652.24     376.54   4.388 1.17e-05 ***
## wrk.exp.year     1063.35      65.72  16.179  < 2e-16 ***
## edu.all           447.34      19.42  23.038  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 25880 on 5386 degrees of freedom
## Multiple R-squared:  0.1866, Adjusted R-squared:  0.1858
```

```
## F-statistic: 247.1 on 5 and 5386 DF,  p-value: < 2.2e-16
```

```
#OLS model: ln.income.x~ag+gender+marital+work_exp_years+sy.edu.all
summary(ols.heckman.lnincome.x.glm)
```

```
##
## Call:
## lm(formula = ln.income.x ~ age2019 + gender + marital.status +
##     wrk.exp.year + edu.all + IMR.glm)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.3851  -0.2398   0.1952   0.5634   1.7365
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     14.861870   1.430494  10.389  < 2e-16 ***
## age2019         -0.093602   0.029894  -3.131  0.00175 **
## gender          -0.545456   0.068427  -7.971  1.9e-15 ***
## marital.status   0.015813   0.018987   0.833  0.40498
## wrk.exp.year    -0.001437   0.010924  -0.132  0.89536
## edu.all          0.016111   0.001676   9.611  < 2e-16 ***
## IMR.glm        -52.636459  13.393861  -3.930  8.6e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.166 on 5385 degrees of freedom
## Multiple R-squared:  0.08178,    Adjusted R-squared:  0.08076
## F-statistic: 79.94 on 6 and 5385 DF,  p-value: < 2.2e-16
```

```
OLS.lnincome.x <- lm(ln.income.x~age2019+gender+marital.status+wrk.exp.year+edu.all)
summary(OLS.income.x)
```

```
##
## Call:
## lm(formula = income.x ~ age2019 + gender + marital.status + wrk.exp.year +
##     edu.all)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -78242 -18613  -3038  18369  76095
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     17809.09    9512.54   1.872   0.0612 .
## age2019           473.18     254.43   1.860   0.0630 .
## gender         -13073.67     706.96 -18.493  < 2e-16 ***
## marital.status   1652.24     376.54   4.388 1.17e-05 ***
## wrk.exp.year     1063.35      65.72  16.179  < 2e-16 ***
## edu.all           447.34      19.42  23.038  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 25880 on 5386 degrees of freedom
## Multiple R-squared:  0.1866, Adjusted R-squared:  0.1858
```

```
## F-statistic: 247.1 on 5 and 5386 DF,  p-value: < 2.2e-16
```

#=========compare============== #because I include the IMR to solve the selection bias, work experience and marital status are no longer correlated with income. #Some people got married and decided not to work any more, so the income is zero, which results selection bias. #also, some people decide not to work and income is 0. #IMR eliminates such selection bias,then, marital status and work experience are no longer correlated with income in regression.

Exercise 3 Censoring

#3.1 Plot a histogram to check whether the distribution of the income variable. What might be the censored value here?

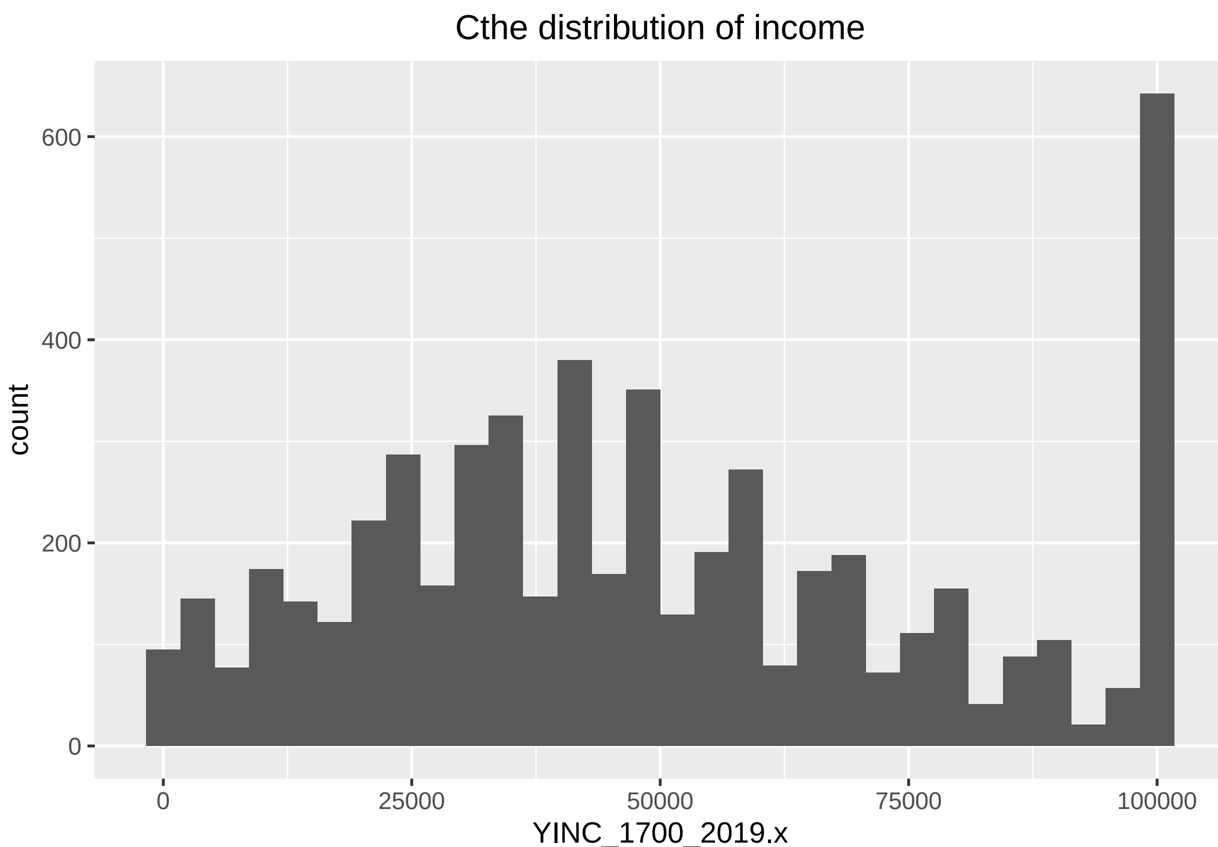
```
#income in dat_A4 top-coded as 100000
summary(d1$YINC_1700_2019.x)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
##       0   28000   45000   49838   70000  100000    3572
```

```
#plot
ggplot(d1, aes(x=YINC_1700_2019.x)) +
  geom_histogram()+
  ggtitle("Cthe distribution of income")+
  theme(plot.title=element_text(hjust=0.5))
```

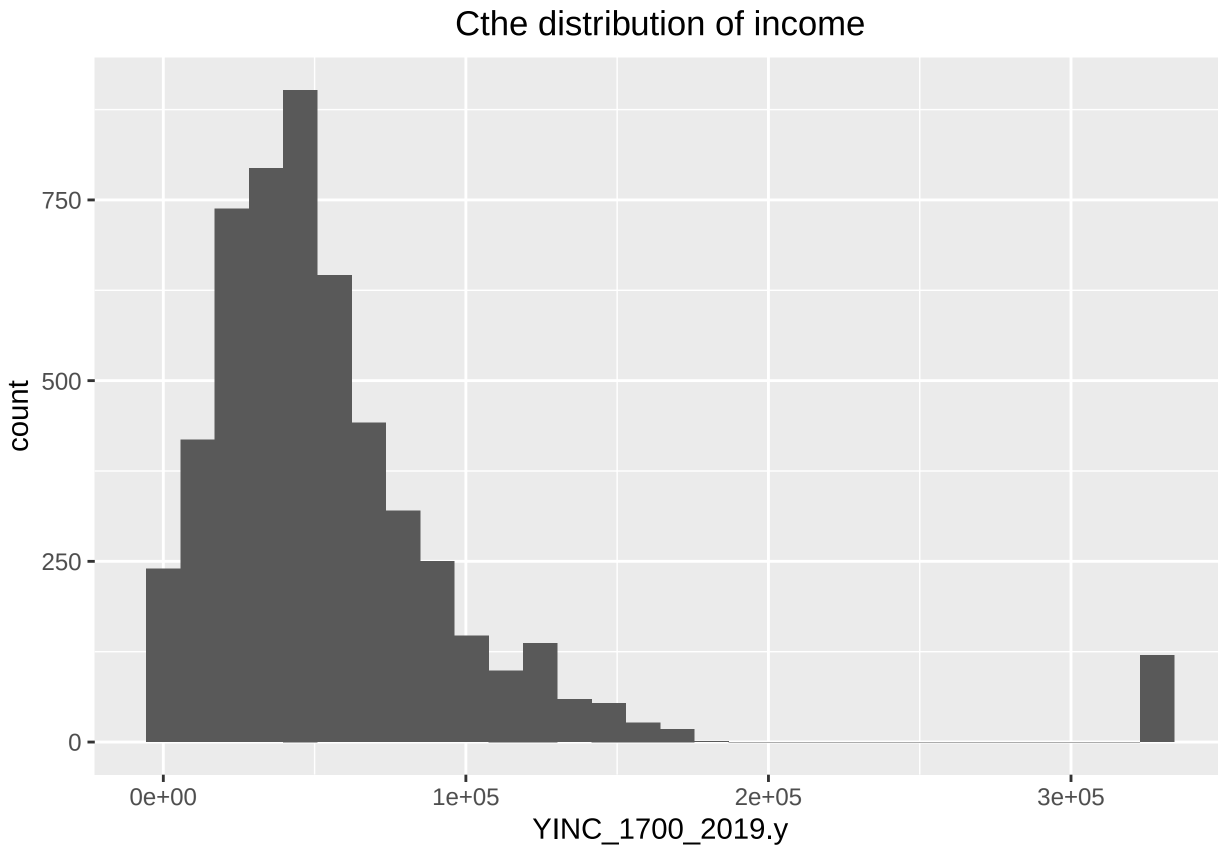
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 3572 rows containing non-finite values (stat_bin).
```



```
#compare with no censor income in panel
ggplot(d1, aes(x=YINC_1700_2019.y)) +
  geom_histogram()+
  ggtitle("Cthe distribution of income")+
  theme(plot.title=element_text(hjust=0.5))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 3572 rows containing non-finite values (stat_bin).
```



#================ #we found that top wages are coded as 10000, but the real max income is 30000+, so for part of income variable, we only observe the range, but not the exact value. #================

#3.2 Propose a model to deal with the censoring problem. #Use Tobit Model to deal with the censoring problem

#3.3 Estimate the appropriate model with the censored data (please write down the likelihood function and optimize yourself without using the pre-programmed package)

```
#Write the likelihood function and optimize for Tobit Model
#=========prepare dataset======
#create indicator variables whether the income is top-coded. 100000+
d3$top.coded.income.x = 0
d3$top.coded.income.x[which(d3$YINC_1700_2019.x<100000)] = 1
top.coded.income.x = d3$top.coded.income.x

#=========model==========
#income.x ~ age2019+ gender + marital.status + wrk.exp.year + edu.all
#========================
```

```r
set.seed(77)
#likelihood function
tobit_likeli = function(par,x1,x2,x3,x4,x5,x6,indct,y){
  yhat = par[1]*x1 + par[2]*x2 + par[3]*x3 + par[4]*x4 + par[5]*x5 + par[6]*x6
  res = y - yhat
  standard = (100000-yhat)/exp(par[7])
  like = indct*log(dnorm(res/exp(par[7]))/exp(par[7])) + (1-indct)*log(1 - pnorm(standard))
  return(-sum(like))
}

#optimize with income
tobit.income <- optim(runif(7,-20,20),fn=tobit_likeli,method="BFGS",
                      control=list(trace=6,REPORT=1,maxit=1000),
               x1=intsct,x2=age2019,x3=gender,x4=marital.status,x5=wrk.exp.year,x6=edu.all,
               indct=top.coded.income.x,y=income.x,hessian=TRUE)
```

```
## initial  value 71711.386690
## iter   2 value 64780.097867
## iter   3 value 61547.817421
## iter   4 value 60266.092644
## iter   5 value 60228.133364
## iter   6 value 60135.646355
## iter   7 value 60105.240769
## iter   8 value 60104.853182
## iter   9 value 60080.156989
## iter  10 value 58003.962815
## iter  11 value 57410.787611
## iter  12 value 56939.161842
## iter  13 value 56668.517831
## iter  14 value 56629.317450
## iter  15 value 56624.679949
## iter  16 value 56624.413378
## iter  17 value 56624.327215
## iter  18 value 56624.162027
## iter  19 value 56623.814734
## iter  20 value 56622.878475
## iter  21 value 56620.530587
## iter  22 value 56614.725721
## iter  23 value 56610.496603
## iter  24 value 56609.500213
## iter  25 value 56609.318193
## iter  26 value 56608.031370
## iter  27 value 56606.782515
## iter  28 value 56605.748080
## iter  29 value 56605.453644
## iter  30 value 56605.346756
## iter  31 value 56605.212946
## iter  32 value 56604.843636
## iter  33 value 56603.919941
## iter  34 value 56601.536318
## iter  35 value 56595.812821
## iter  36 value 56583.774184
## iter  37 value 56565.211715
## iter  38 value 56553.095379
```

```
## iter  39 value 56550.814522
## iter  40 value 56550.727759
## iter  41 value 56550.338914
## iter  42 value 56549.852246
## iter  43 value 56549.337716
## iter  44 value 56549.138671
## iter  45 value 56549.090118
## iter  46 value 56549.062904
## iter  47 value 56548.992617
## iter  48 value 56548.818904
## iter  49 value 56548.359625
## iter  50 value 56547.197739
## iter  51 value 56544.368101
## iter  52 value 56538.309107
## iter  53 value 56533.348793
## iter  54 value 56532.313416
## iter  55 value 56532.250888
## iter  56 value 56531.780660
## iter  57 value 56531.409237
## iter  58 value 56531.166171
## iter  59 value 56531.118322
## iter  60 value 56531.104911
## iter  61 value 56531.085655
## iter  62 value 56531.030761
## iter  63 value 56530.893152
## iter  64 value 56530.532254
## iter  65 value 56529.626877
## iter  66 value 56527.477650
## iter  67 value 56523.132592
## iter  68 value 56519.664889
## iter  69 value 56518.927767
## iter  70 value 56518.887084
## iter  71 value 56518.550209
## iter  72 value 56518.326354
## iter  73 value 56518.204959
## iter  74 value 56518.186774
## iter  75 value 56518.182091
## iter  76 value 56518.174202
## iter  77 value 56518.152191
## iter  78 value 56518.096396
## iter  79 value 56517.950555
## iter  80 value 56517.582881
## iter  81 value 56516.701520
## iter  82 value 56514.868671
## iter  83 value 56513.454320
## iter  84 value 56513.134567
## iter  85 value 56513.110904
## iter  86 value 56512.876908
## iter  87 value 56512.761968
## iter  88 value 56512.715360
## iter  89 value 56512.710248
## iter  90 value 56512.708308
## iter  91 value 56512.703335
## iter  92 value 56512.690841
```

```
## iter  93 value 56512.657800
## iter  94 value 56512.572760
## iter  95 value 56512.356553
## iter  96 value 56511.836130
## iter  97 value 56510.734280
## iter  98 value 56509.852348
## iter  99 value 56509.650729
## iter 100 value 56509.636721
## iter 101 value 56509.486119
## iter 102 value 56509.423653
## iter 103 value 56509.401753
## iter 104 value 56509.399649
## iter 105 value 56509.398602
## iter 106 value 56509.395551
## iter 107 value 56509.388082
## iter 108 value 56509.368054
## iter 109 value 56509.316349
## iter 110 value 56509.182117
## iter 111 value 56508.842603
## iter 112 value 56508.028234
## iter 113 value 56507.264780
## iter 114 value 56507.090824
## iter 115 value 56507.079988
## iter 116 value 56506.958317
## iter 117 value 56506.911869
## iter 118 value 56506.896598
## iter 119 value 56506.895155
## iter 120 value 56506.894265
## iter 121 value 56506.891611
## iter 122 value 56506.885111
## iter 123 value 56506.867648
## iter 124 value 56506.822374
## iter 125 value 56506.703471
## iter 126 value 56506.393510
## iter 127 value 56505.589332
## iter 128 value 56504.747637
## iter 129 value 56504.558088
## iter 130 value 56504.547953
## iter 131 value 56504.432684
## iter 132 value 56504.389404
## iter 133 value 56504.375321
## iter 134 value 56504.373980
## iter 135 value 56504.373100
## iter 136 value 56504.370483
## iter 137 value 56504.364060
## iter 138 value 56504.346815
## iter 139 value 56504.302073
## iter 140 value 56504.184451
## iter 141 value 56503.876900
## iter 142 value 56503.072638
## iter 143 value 56502.215020
## iter 144 value 56502.022627
## iter 145 value 56502.012968
## iter 146 value 56501.902402
```

```
## iter 147 value 56501.861272
## iter 148 value 56501.847958
## iter 149 value 56501.846676
## iter 150 value 56501.845797
## iter 151 value 56501.843188
## iter 152 value 56501.836773
## iter 153 value 56501.819562
## iter 154 value 56501.774897
## iter 155 value 56501.657491
## iter 156 value 56501.350503
## iter 157 value 56500.547797
## iter 158 value 56499.685292
## iter 159 value 56499.492292
## iter 160 value 56499.483060
## iter 161 value 56499.376782
## iter 162 value 56499.337596
## iter 163 value 56499.324975
## iter 164 value 56499.323747
## iter 165 value 56499.322867
## iter 166 value 56499.320268
## iter 167 value 56499.313862
## iter 168 value 56499.296690
## iter 169 value 56499.252114
## iter 170 value 56499.134974
## iter 171 value 56498.828778
## iter 172 value 56498.028926
## iter 173 value 56497.164350
## iter 174 value 56496.971319
## iter 175 value 56496.962490
## iter 176 value 56496.860316
## iter 177 value 56496.822945
## iter 178 value 56496.810960
## iter 179 value 56496.809779
## iter 180 value 56496.808897
## iter 181 value 56496.806303
## iter 182 value 56496.799895
## iter 183 value 56496.782736
## iter 184 value 56496.738191
## iter 185 value 56496.621225
## iter 186 value 56496.316015
## iter 187 value 56495.522507
## iter 188 value 56494.663195
## iter 189 value 56494.471667
## iter 190 value 56494.463288
## iter 191 value 56494.365649
## iter 192 value 56494.330265
## iter 193 value 56494.318965
## iter 194 value 56494.317828
## iter 195 value 56494.316927
## iter 196 value 56494.314297
## iter 197 value 56494.307781
## iter 198 value 56494.290366
## iter 199 value 56494.245191
## iter 200 value 56494.127032
```

```
## iter 201 value 56493.821622
## iter 202 value 56493.047350
## iter 203 value 56492.222530
## iter 204 value 56492.039000
## iter 205 value 56492.031444
## iter 206 value 56491.941715
## iter 207 value 56491.909747
## iter 208 value 56491.899572
## iter 209 value 56491.898467
## iter 210 value 56491.897446
## iter 211 value 56491.894548
## iter 212 value 56491.887291
## iter 213 value 56491.868034
## iter 214 value 56491.818358
## iter 215 value 56491.690934
## iter 216 value 56491.377259
## iter 217 value 56490.675036
## iter 218 value 56490.003073
## iter 219 value 56489.853517
## iter 220 value 56489.848225
## iter 221 value 56489.779860
## iter 222 value 56489.756595
## iter 223 value 56489.749047
## iter 224 value 56489.747878
## iter 225 value 56489.746103
## iter 226 value 56489.741596
## iter 227 value 56489.729840
## iter 228 value 56489.699547
## iter 229 value 56489.623267
## iter 230 value 56489.442937
## iter 231 value 56489.077016
## iter 232 value 56488.544578
## iter 233 value 56488.216467
## iter 234 value 56488.143229
## iter 235 value 56488.141445
## iter 236 value 56488.104413
## iter 237 value 56488.088426
## iter 238 value 56488.080541
## iter 239 value 56488.077797
## iter 240 value 56488.072172
## iter 241 value 56488.059000
## iter 242 value 56488.024352
## iter 243 value 56487.939700
## iter 244 value 56487.747875
## iter 245 value 56487.398935
## iter 246 value 56486.986729
## iter 247 value 56486.752541
## iter 248 value 56486.715596
## iter 249 value 56486.706410
## iter 250 value 56486.705387
## iter 251 value 56486.679987
## iter 252 value 56486.636771
## iter 253 value 56486.512956
## iter 254 value 56486.266522
```

```
## iter 255 value 56485.863346
## iter 256 value 56485.503741
## iter 257 value 56485.365739
## iter 258 value 56485.344152
## iter 259 value 56485.340741
## iter 260 value 56485.335653
## iter 261 value 56485.324041
## iter 262 value 56485.305925
## iter 263 value 56485.303360
## iter 264 value 56485.302135
## iter 265 value 56485.300938
## iter 266 value 56485.295741
## iter 267 value 56485.284307
## iter 268 value 56485.253231
## iter 269 value 56485.179323
## iter 270 value 56485.015308
## iter 271 value 56484.731790
## iter 272 value 56484.424574
## iter 273 value 56484.261587
## iter 274 value 56484.210044
## iter 275 value 56484.179255
## iter 276 value 56484.117891
## iter 277 value 56484.007493
## iter 278 value 56483.959256
## iter 279 value 56483.944863
## iter 280 value 56483.940497
## iter 281 value 56483.910329
## iter 282 value 56483.885562
## iter 283 value 56483.868539
## iter 284 value 56483.864603
## iter 285 value 56483.863088
## iter 286 value 56483.860315
## iter 287 value 56483.852997
## iter 288 value 56483.834274
## iter 289 value 56483.786522
## iter 290 value 56483.671182
## iter 291 value 56483.423789
## iter 292 value 56483.020832
## iter 293 value 56482.755061
## iter 294 value 56482.694846
## iter 295 value 56482.692739
## iter 296 value 56482.658286
## iter 297 value 56482.648565
## iter 298 value 56482.645322
## iter 299 value 56482.644318
## iter 300 value 56482.641420
## iter 301 value 56482.634853
## iter 302 value 56482.616852
## iter 303 value 56482.571417
## iter 304 value 56482.456727
## iter 305 value 56482.189736
## iter 306 value 56481.663318
## iter 307 value 56480.941117
## iter 308 value 56480.503709
```

```
## iter 309 value 56480.407387
## iter 310 value 56480.405943
## iter 311 value 56480.356890
## iter 312 value 56480.276961
## iter 313 value 56480.059389
## iter 314 value 56479.685383
## iter 315 value 56479.227345
## iter 316 value 56478.973278
## iter 317 value 56478.912458
## iter 318 value 56478.901152
## iter 319 value 56478.890505
## iter 320 value 56478.864960
## iter 321 value 56478.829684
## iter 322 value 56478.799834
## iter 323 value 56478.796447
## iter 324 value 56478.795141
## iter 324 value 56478.794307
## iter 324 value 56478.794307
## final  value 56478.794307
## converged
```

```
tobit.income$par
```

```
## [1]    -1.538114   451.650697 -1640.807702   477.081213  1162.639580
## [6]   512.206797    10.294090
```

```
#reg.tobit <- tobit(income.x ~ age2019+ gender + marital.status + wrk.exp.year + edu.all,left=-Inf,righ
#reg.tobit2 <- tobit(ln.income.x~ age2019+ gender + marital.status + wrk.exp.year + edu.all,left=-Inf,r
#summary(reg.tobit)
#summary(reg.tobit2)
```

#3.4 Interpret the results above and compare to those when not correcting for the censored data

```
tobit.income$par
```

```
## [1]    -1.538114   451.650697 -1640.807702   477.081213  1162.639580
## [6]   512.206797    10.294090
```

```
#===============interpret=============
#income increases as age increase and work experience increase, as well as education years.
#if you are female, your income will be lower than male.

#===============compare=============
#the OLS model with the censored data (ols)
OLS.income.x$coefficients
```

```
##    (Intercept)        age2019         gender marital.status   wrk.exp.year
##     17809.0930       473.1817    -13073.6667      1652.2372      1063.3472
##        edu.all
##       447.3382
```

```
#compare with these results:
#the significant change is the value of coefficient of gender
#because the existence of censoring problem, the gender differences between income is much greater.
```

Exercise 4 Panel Data

```
#===========data prepare===============
names(dat_A4_panel)
```

```
##   [1] "PUBID_1997"                  "YINC_1700_1997"
##   [3] "KEY_SEX_1997"                "KEY_BDATE_M_1997"
##   [5] "KEY_BDATE_Y_1997"            "CV_MARSTAT_COLLAPSED_1997"
##   [7] "CV_WKSWK_JOB_DLI.01_1997"    "CV_WKSWK_JOB_DLI.02_1997"
##   [9] "CV_WKSWK_JOB_DLI.03_1997"    "CV_WKSWK_JOB_DLI.04_1997"
##  [11] "CV_WKSWK_JOB_DLI.05_1997"    "CV_WKSWK_JOB_DLI.06_1997"
##  [13] "CV_WKSWK_JOB_DLI.07_1997"    "CV_SAMPLE_TYPE_1997"
##  [15] "KEY_RACE_ETHNICITY_1997"     "YINC-1700_1998"
##  [17] "CV_HIGHEST_DEGREE_9899_1998" "CV_MARSTAT_COLLAPSED_1998"
##  [19] "CV_WKSWK_JOB_DLI.01_1998"    "CV_WKSWK_JOB_DLI.02_1998"
##  [21] "CV_WKSWK_JOB_DLI.03_1998"    "CV_WKSWK_JOB_DLI.04_1998"
##  [23] "CV_WKSWK_JOB_DLI.05_1998"    "CV_WKSWK_JOB_DLI.06_1998"
##  [25] "CV_WKSWK_JOB_DLI.07_1998"    "CV_WKSWK_JOB_DLI.08_1998"
##  [27] "CV_WKSWK_JOB_DLI.09_1998"    "YINC-1700_1999"
##  [29] "CV_HIGHEST_DEGREE_9900_1999" "CV_MARSTAT_COLLAPSED_1999"
##  [31] "CV_WKSWK_JOB_DLI.01_1999"    "CV_WKSWK_JOB_DLI.02_1999"
##  [33] "CV_WKSWK_JOB_DLI.03_1999"    "CV_WKSWK_JOB_DLI.04_1999"
##  [35] "CV_WKSWK_JOB_DLI.05_1999"    "CV_WKSWK_JOB_DLI.06_1999"
##  [37] "CV_WKSWK_JOB_DLI.07_1999"    "CV_WKSWK_JOB_DLI.08_1999"
##  [39] "CV_WKSWK_JOB_DLI.09_1999"    "YINC-1700_2000"
##  [41] "CV_HIGHEST_DEGREE_0001_2000" "CV_MARSTAT_COLLAPSED_2000"
##  [43] "CV_WKSWK_JOB_DLI.01_2000"    "CV_WKSWK_JOB_DLI.02_2000"
##  [45] "CV_WKSWK_JOB_DLI.03_2000"    "CV_WKSWK_JOB_DLI.04_2000"
##  [47] "CV_WKSWK_JOB_DLI.05_2000"    "CV_WKSWK_JOB_DLI.06_2000"
##  [49] "CV_WKSWK_JOB_DLI.07_2000"    "CV_WKSWK_JOB_DLI.08_2000"
##  [51] "CV_WKSWK_JOB_DLI.09_2000"    "YINC-1700_2001"
##  [53] "CV_HIGHEST_DEGREE_0102_2001" "CV_MARSTAT_COLLAPSED_2001"
##  [55] "CV_WKSWK_JOB_DLI.01_2001"    "CV_WKSWK_JOB_DLI.02_2001"
##  [57] "CV_WKSWK_JOB_DLI.03_2001"    "CV_WKSWK_JOB_DLI.04_2001"
##  [59] "CV_WKSWK_JOB_DLI.05_2001"    "CV_WKSWK_JOB_DLI.06_2001"
##  [61] "CV_WKSWK_JOB_DLI.07_2001"    "CV_WKSWK_JOB_DLI.08_2001"
##  [63] "YINC-1700_2002"              "CV_HIGHEST_DEGREE_0203_2002"
##  [65] "CV_MARSTAT_COLLAPSED_2002"   "CV_WKSWK_JOB_DLI.01_2002"
##  [67] "CV_WKSWK_JOB_DLI.02_2002"    "CV_WKSWK_JOB_DLI.03_2002"
##  [69] "CV_WKSWK_JOB_DLI.04_2002"    "CV_WKSWK_JOB_DLI.05_2002"
##  [71] "CV_WKSWK_JOB_DLI.06_2002"    "CV_WKSWK_JOB_DLI.07_2002"
##  [73] "CV_WKSWK_JOB_DLI.08_2002"    "CV_WKSWK_JOB_DLI.09_2002"
##  [75] "CV_WKSWK_JOB_DLI.10_2002"    "CV_WKSWK_JOB_DLI.11_2002"
##  [77] "CV_HIGHEST_DEGREE_0304_2003" "CV_MARSTAT_COLLAPSED_2003"
##  [79] "CV_WKSWK_JOB_DLI.01_2003"    "CV_WKSWK_JOB_DLI.02_2003"
##  [81] "CV_WKSWK_JOB_DLI.03_2003"    "CV_WKSWK_JOB_DLI.04_2003"
##  [83] "CV_WKSWK_JOB_DLI.05_2003"    "CV_WKSWK_JOB_DLI.06_2003"
##  [85] "CV_WKSWK_JOB_DLI.07_2003"    "CV_WKSWK_JOB_DLI.08_2003"
##  [87] "CV_WKSWK_JOB_DLI.09_2003"    "CV_WKSWK_JOB_DLI.10_2003"
##  [89] "YINC-1700_2003"              "CV_HIGHEST_DEGREE_0405_2004"
##  [91] "CV_MARSTAT_COLLAPSED_2004"   "CV_WKSWK_JOB_DLI.01_2004"
##  [93] "CV_WKSWK_JOB_DLI.02_2004"    "CV_WKSWK_JOB_DLI.03_2004"
##  [95] "CV_WKSWK_JOB_DLI.04_2004"    "CV_WKSWK_JOB_DLI.05_2004"
##  [97] "CV_WKSWK_JOB_DLI.06_2004"    "CV_WKSWK_JOB_DLI.07_2004"
```

```
##  [99] "YINC-1700_2004"                  "CV_HIGHEST_DEGREE_0506_2005"
## [101] "CV_MARSTAT_COLLAPSED_2005"       "CV_WKSWK_JOB_DLI.01_2005"
## [103] "CV_WKSWK_JOB_DLI.02_2005"        "CV_WKSWK_JOB_DLI.03_2005"
## [105] "CV_WKSWK_JOB_DLI.04_2005"        "CV_WKSWK_JOB_DLI.05_2005"
## [107] "CV_WKSWK_JOB_DLI.06_2005"        "CV_WKSWK_JOB_DLI.07_2005"
## [109] "CV_WKSWK_JOB_DLI.08_2005"        "CV_WKSWK_JOB_DLI.09_2005"
## [111] "YINC-1700_2005"                  "CV_HIGHEST_DEGREE_0607_2006"
## [113] "CV_MARSTAT_COLLAPSED_2006"       "CV_WKSWK_JOB_DLI.01_2006"
## [115] "CV_WKSWK_JOB_DLI.02_2006"        "CV_WKSWK_JOB_DLI.03_2006"
## [117] "CV_WKSWK_JOB_DLI.04_2006"        "CV_WKSWK_JOB_DLI.05_2006"
## [119] "CV_WKSWK_JOB_DLI.06_2006"        "CV_WKSWK_JOB_DLI.07_2006"
## [121] "CV_WKSWK_JOB_DLI.08_2006"        "CV_WKSWK_JOB_DLI.09_2006"
## [123] "YINC-1700_2006"                  "CV_HIGHEST_DEGREE_0708_2007"
## [125] "CV_MARSTAT_COLLAPSED_2007"       "CV_WKSWK_JOB_DLI.01_2007"
## [127] "CV_WKSWK_JOB_DLI.02_2007"        "CV_WKSWK_JOB_DLI.03_2007"
## [129] "CV_WKSWK_JOB_DLI.04_2007"        "CV_WKSWK_JOB_DLI.05_2007"
## [131] "CV_WKSWK_JOB_DLI.06_2007"        "CV_WKSWK_JOB_DLI.07_2007"
## [133] "CV_WKSWK_JOB_DLI.08_2007"        "YINC-1700_2007"
## [135] "CV_HIGHEST_DEGREE_0809_2008"     "CV_MARSTAT_COLLAPSED_2008"
## [137] "CV_WKSWK_JOB_DLI.01_2008"        "CV_WKSWK_JOB_DLI.02_2008"
## [139] "CV_WKSWK_JOB_DLI.03_2008"        "CV_WKSWK_JOB_DLI.04_2008"
## [141] "CV_WKSWK_JOB_DLI.05_2008"        "CV_WKSWK_JOB_DLI.06_2008"
## [143] "CV_WKSWK_JOB_DLI.07_2008"        "CV_WKSWK_JOB_DLI.08_2008"
## [145] "YINC-1700_2008"                  "CV_HIGHEST_DEGREE_0910_2009"
## [147] "CV_MARSTAT_COLLAPSED_2009"       "CV_WKSWK_JOB_DLI.01_2009"
## [149] "CV_WKSWK_JOB_DLI.02_2009"        "CV_WKSWK_JOB_DLI.03_2009"
## [151] "CV_WKSWK_JOB_DLI.04_2009"        "CV_WKSWK_JOB_DLI.05_2009"
## [153] "CV_WKSWK_JOB_DLI.06_2009"        "CV_WKSWK_JOB_DLI.07_2009"
## [155] "CV_WKSWK_JOB_DLI.08_2009"        "CV_WKSWK_JOB_DLI.09_2009"
## [157] "YINC-1700_2009"                  "CV_HIGHEST_DEGREE_EVER_EDT_2010"
## [159] "CV_HIGHEST_DEGREE_1011_2010"     "CV_MARSTAT_COLLAPSED_2010"
## [161] "CV_WKSWK_JOB_DLI.01_2010"        "CV_WKSWK_JOB_DLI.02_2010"
## [163] "CV_WKSWK_JOB_DLI.03_2010"        "CV_WKSWK_JOB_DLI.04_2010"
## [165] "CV_WKSWK_JOB_DLI.05_2010"        "CV_WKSWK_JOB_DLI.06_2010"
## [167] "CV_WKSWK_JOB_DLI.07_2010"        "CV_WKSWK_JOB_DLI.08_2010"
## [169] "CV_WKSWK_JOB_DLI.09_2010"        "YINC-1700_2010"
## [171] "CV_HIGHEST_DEGREE_EVER_EDT_2011" "CV_HIGHEST_DEGREE_1112_2011"
## [173] "CV_MARSTAT_COLLAPSED_2011"       "CV_WKSWK_JOB_DLI.01_2011"
## [175] "CV_WKSWK_JOB_DLI.02_2011"        "CV_WKSWK_JOB_DLI.03_2011"
## [177] "CV_WKSWK_JOB_DLI.04_2011"        "CV_WKSWK_JOB_DLI.05_2011"
## [179] "CV_WKSWK_JOB_DLI.06_2011"        "CV_WKSWK_JOB_DLI.07_2011"
## [181] "CV_WKSWK_JOB_DLI.08_2011"        "CV_WKSWK_JOB_DLI.09_2011"
## [183] "CV_WKSWK_JOB_DLI.10_2011"        "CV_WKSWK_JOB_DLI.11_2011"
## [185] "CV_WKSWK_JOB_DLI.12_2011"        "CV_WKSWK_JOB_DLI.13_2011"
## [187] "YINC-1700_2011"                  "CV_HIGHEST_DEGREE_EVER_EDT_2013"
## [189] "CV_HIGHEST_DEGREE_1314_2013"     "CV_MARSTAT_COLLAPSED_2013"
## [191] "CV_WKSWK_JOB_DLI.01_2013"        "CV_WKSWK_JOB_DLI.02_2013"
## [193] "CV_WKSWK_JOB_DLI.03_2013"        "CV_WKSWK_JOB_DLI.04_2013"
## [195] "CV_WKSWK_JOB_DLI.05_2013"        "CV_WKSWK_JOB_DLI.06_2013"
## [197] "CV_WKSWK_JOB_DLI.07_2013"        "CV_WKSWK_JOB_DLI.08_2013"
## [199] "CV_WKSWK_JOB_DLI.09_2013"        "CV_WKSWK_JOB_DLI.10_2013"
## [201] "YINC-1700_2013"                  "CV_HIGHEST_DEGREE_EVER_EDT_2015"
## [203] "CV_MARSTAT_COLLAPSED_2015"       "CV_WKSWK_JOB_DLI.01_2015"
## [205] "CV_WKSWK_JOB_DLI.02_2015"        "CV_WKSWK_JOB_DLI.03_2015"
```

```
## [207] "CV_WKSWK_JOB_DLI.04_2015"        "CV_WKSWK_JOB_DLI.05_2015"
## [209] "CV_WKSWK_JOB_DLI.06_2015"        "CV_WKSWK_JOB_DLI.07_2015"
## [211] "CV_WKSWK_JOB_DLI.08_2015"        "CV_WKSWK_JOB_DLI.09_2015"
## [213] "CV_WKSWK_JOB_DLI.10_2015"        "CV_WKSWK_JOB_DLI.11_2015"
## [215] "CV_WKSWK_JOB_DLI.12_2015"        "YINC-1700_2015"
## [217] "CV_HIGHEST_DEGREE_EVER_EDT_2017" "CV_MARSTAT_COLLAPSED_2017"
## [219] "CV_WKSWK_JOB_DLI.01_2017"        "CV_WKSWK_JOB_DLI.02_2017"
## [221] "CV_WKSWK_JOB_DLI.03_2017"        "CV_WKSWK_JOB_DLI.04_2017"
## [223] "CV_WKSWK_JOB_DLI.05_2017"        "CV_WKSWK_JOB_DLI.06_2017"
## [225] "CV_WKSWK_JOB_DLI.07_2017"        "CV_WKSWK_JOB_DLI.08_2017"
## [227] "CV_WKSWK_JOB_DLI.09_2017"        "CV_WKSWK_JOB_DLI.10_2017"
## [229] "CV_WKSWK_JOB_DLI.11_2017"        "CV_WKSWK_JOB_DLI.12_2017"
## [231] "CV_WKSWK_JOB_DLI.13_2017"        "CV_WKSWK_JOB_DLI.14_2017"
## [233] "CV_WKSWK_JOB_DLI.15_2017"        "YINC-1700_2017"
## [235] "CV_HIGHEST_DEGREE_EVER_EDT_2019" "CV_MARSTAT_COLLAPSED_2019"
## [237] "CV_WKSWK_JOB_DLI.01_2019"        "CV_WKSWK_JOB_DLI.02_2019"
## [239] "CV_WKSWK_JOB_DLI.03_2019"        "CV_WKSWK_JOB_DLI.04_2019"
## [241] "CV_WKSWK_JOB_DLI.05_2019"        "CV_WKSWK_JOB_DLI.06_2019"
## [243] "CV_WKSWK_JOB_DLI.07_2019"        "CV_WKSWK_JOB_DLI.08_2019"
## [245] "CV_WKSWK_JOB_DLI.09_2019"        "CV_WKSWK_JOB_DLI.10_2019"
## [247] "CV_WKSWK_JOB_DLI.11_2019"        "YINC_1700_2019"
```

#4.1 Explain the potential ability bias when trying to explain to understand the determinants of wages #the theory of human capital and signaling theory both predict that the most productive individuals have an interest in studying for the longest period, entailing the possibility of the so called ability bias

#4.2 Exploit the panel dimension of the data to propose a model to correct for the ability bias. Estimate the model using the following strategy.

```r
#=========prepare data (edu/marital status/work experience on income)====================
#income in last year
colnames(dat_A4_panel)[c(2,16,28,40,52,63,89,
                         99,111,123,134,145,
                         157,170,187,201,216,234,248)]=c("income.1997","income.1998","income.1999",
                                                          "income.2000","income.2001","income.2002",
                                                          "income.2003","income.2004","income.2005",
                                                          "income.2006","income.2007","income.2008",
                                                          "income.2009","income.2010","income.2011",
                                                          "income.2013","income.2015","income.2017",
                                                          "income.2019")

#marital at the survey date
colnames(dat_A4_panel)[c(6,18,30,42,54,65,78,
                         91,101,113,125,136,
                         147,160,173,190,203,218,236)]=c("mar.1997","mar.1998","mar.1999","mar.2000",
                                                          "mar.2001","mar.2002","mar.2003","mar.2004",
                                                          "mar.2005","mar.2006","mar.2007","mar.2008",
                                                          "mar.2009","mar.2010","mar.2011","mar.2013",
                                                          "mar.2015","mar.2017","mar.2019")


#edu
#there are two variables representing "highest degree ever received":
#1998-2009: only "HIGHEST DEGREE RECEIVED PRIOR TO THE ACAD YEAR"
#2010-2013: one is "The highest degree received as of the survey date";
```

```
#"HIGHEST DEGREE RECEIVED PRIOR TO THE ACAD YEAR"
#2015-2019: only "The highest degree received as of the survey date"
#we use "THE ACAD YEAR" from 1998-2013, and "of the survey date" from 2015-2019
#there are no significant differences in these two variables.
colnames(dat_A4_panel)[c(17,29,41,53,64,77,
                         90,100,112,124,135,146,
                         159,172,189,202,217,235)]=c("edu.1998","edu.1999","edu.2000","edu.2001","edu.2
                                                     "edu.2003","edu.2004","edu.2005","edu.2006","edu.2
                                                     "edu.2008","edu.2009","edu.2010","edu.2011","edu.2
                                                     "edu.2015","edu.2017","edu.2019")


#work experience total (up to survey date), then, translate it into years (assume that there are 52 wee
dat.exp=dat_A4_panel[,c(7:13,19:27,31:39,43:51,55:62,66:76,79:88,
                        92:98,102:110,114:122,126:133,137:144,148:156,
                        161:169,174:186,191:200,204:215,219:233,237:247,1)]
dat.exp[is.na(dat.exp)]<-0
dat.exp = mutate(dat.exp,
                 wrk.exp.1997 = rowSums(dat.exp[,1:7])/52,
                 wrk.exp.1998 = rowSums(dat.exp[,8:16])/52,
                 wrk.exp.1999 = rowSums(dat.exp[,17:25])/52,
                 wrk.exp.2000 = rowSums(dat.exp[,26:34])/52,
                 wrk.exp.2001 = rowSums(dat.exp[,35:42])/52,
                 wrk.exp.2002 = rowSums(dat.exp[,43:53])/52,
                 wrk.exp.2003 = rowSums(dat.exp[,54:63])/52,
                 wrk.exp.2004 = rowSums(dat.exp[,64:70])/52,
                 wrk.exp.2005 = rowSums(dat.exp[,71:79])/52,
                 wrk.exp.2006 = rowSums(dat.exp[,80:88])/52,
                 wrk.exp.2007 = rowSums(dat.exp[,89:96])/52,
                 wrk.exp.2008 = rowSums(dat.exp[,97:104])/52,
                 wrk.exp.2009 = rowSums(dat.exp[,105:113])/52,
                 wrk.exp.2010 = rowSums(dat.exp[,114:122])/52,
                 wrk.exp.2011 = rowSums(dat.exp[,123:135])/52,
                 wrk.exp.2013 = rowSums(dat.exp[,136:145])/52,
                 wrk.exp.2015 = rowSums(dat.exp[,146:157])/52,
                 wrk.exp.2017 = rowSums(dat.exp[,158:172])/52,
                 wrk.exp.2019 = rowSums(dat.exp[,173:183])/52)
dat.exp.year=dat.exp[,184:203]


#The panel data used in this problem:
dat.panel = select(dat_A4_panel,
                   PUBID_1997,KEY_BDATE_Y_1997,KEY_BDATE_M_1997,KEY_SEX_1997,KEY_RACE_ETHNICITY_1997,
                   income.1997,income.1998,income.1999,income.2000,income.2001,income.2002,
                   income.2003,income.2004,income.2005,income.2006,income.2007,income.2008,
                   income.2009,income.2010,income.2011,income.2013,income.2015,income.2017,income.2019,
                   edu.1998,edu.1999,edu.2000,edu.2001,edu.2002,edu.2003,edu.2004,edu.2005,edu.2006,
                   edu.2007,edu.2008,edu.2009,edu.2010,edu.2011,edu.2013,edu.2015,edu.2017,edu.2019,
                   mar.1997,mar.1998,mar.1999,mar.2000,mar.2001,mar.2002,mar.2003,mar.2004,mar.2005,
                   mar.2006,mar.2007,mar.2008,mar.2009,mar.2010,mar.2011,mar.2013,mar.2015,mar.2017,mar
dat.panel = left_join(dat.panel,dat.exp.year,by="PUBID_1997")
colnames(dat.panel)[2:5]=c("Birth.year","Birth.month","Sex","Race")
```

```r
#=======convert to long ============
dat.panel.long = long_panel(dat.panel, prefix='.', begin  = 1997, end = 2019, label_location = "end")
dat.panel.long = subset(dat.panel.long,wave!='2012' & wave!='2014' & wave!='2016' & wave!='2018')
#=======age==========
dat.panel.long$age=dat.panel.long$wave-dat.panel.long$Birth.year

#======data used below======
e = as.data.frame(dat.panel.long)
e$id = as.numeric(e$id)
e$income = as.numeric(e$income)
e$age = as.numeric(e$age)
e$Sex = as.numeric(e$Sex)
e$wrk.exp = as.numeric(e$wrk.exp)
e$edu <- as.numeric(e$edu)
e$mar <- as.numeric(e$mar)
```

```r
#4.2.1 Within Estimator
e1 = e

e1$meanincome <- ave(e1$income, e1$id, FUN=function(x)mean(x, na.rm=T))
e1$meanedu <- ave(e1$edu, e1$id, FUN=function(x)mean(x, na.rm=T))
e1$meanwrkex <- ave(e1$wrk.exp, e1$id, FUN=function(x)mean(x, na.rm=T))
e1$meanmar<- ave(e1$mar, e1$id, FUN=function(x)mean(x, na.rm=T))

e1$d.income <- e1$income - e1$meanincome
e1$d.edu <- e1$edu - e1$meanedu
e1$d.wrkex <- e1$wrk.exp - e1$meanwrkex
e1$d.mar <- e1$mar - e1$meanmar

panel.within.estimator <- lm(d.income~ 0+d.edu  + d.mar+ d.wrkex,e1)
summary(panel.within.estimator)
```

```
##
## Call:
## lm(formula = d.income ~ 0 + d.edu + d.mar + d.wrkex, data = e1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -132972  -11214   -3075    4834  277172
##
## Coefficients:
##         Estimate Std. Error t value Pr(>|t|)
## d.edu    8435.41      84.01  100.41   <2e-16 ***
## d.mar    7622.94     139.27   54.74   <2e-16 ***
## d.wrkex  2088.54      24.53   85.14   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 20010 on 81959 degrees of freedom
##   (88734 observations deleted due to missingness)
## Multiple R-squared:  0.3276, Adjusted R-squared:  0.3276
## F-statistic: 1.331e+04 on 3 and 81959 DF,  p-value: < 2.2e-16
```

```
#==use package====
within = plm(income ~  edu + mar + wrk.exp, e1, model = "within")
summary(within)
```

```
## Oneway (individual) effect Within Model
##
## Call:
## plm(formula = income ~ edu + mar + wrk.exp, data = e1, model = "within")
##
## Unbalanced Panel: n = 8599, T = 1-18, N = 81962
##
## Residuals:
##       Min.    1st Qu.     Median    3rd Qu.       Max.
## -139433.98   -8435.51    -386.96    7171.99  276879.59
##
## Coefficients:
##         Estimate Std. Error t-value  Pr(>|t|)
## edu     9819.271     92.467 106.192 < 2.2e-16 ***
## mar     7266.890    147.935  49.122 < 2.2e-16 ***
## wrk.exp 2515.973     27.290  92.193 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    4.7977e+13
## Residual Sum of Squares: 3.0643e+13
## R-Squared:      0.36129
## Adj. R-Squared: 0.28641
## F-statistic: 13832.4 on 3 and 73360 DF, p-value: < 2.22e-16
```

```
#4.2.2 Between Estimator
e2 = e

m.inc=summarise(group_by(e2,id),income.mean=mean(income,na.rm = TRUE))
m.age=summarise(group_by(e2,id),age.mean=mean(age,na.rm = TRUE))
m.gender=summarise(group_by(e2,id),gender.mean=mean(Sex,na.rm = TRUE))
m.wrkex=summarise(group_by(e2,id),wrkex.mean=mean(wrk.exp,na.rm = TRUE))
m.edu=summarise(group_by(e2,id),edu.mean=mean(edu,na.rm = TRUE))
m.mar=summarise(group_by(e2,id),mar.mean=mean(mar,na.rm = TRUE))


panel.between.estimator <- lm(m.inc$income.mean~m.edu$edu.mean+
                                m.mar$mar.mean+m.wrkex$wrkex.mean)
summary(panel.between.estimator)
```

```
##
## Call:
## lm(formula = m.inc$income.mean ~ m.edu$edu.mean + m.mar$mar.mean +
##     m.wrkex$wrkex.mean)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -39240  -8713  -2576   5506 156981
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)         6121.02     344.90  17.747  < 2e-16 ***
## m.edu$edu.mean      5498.14     161.71  34.001  < 2e-16 ***
## m.mar$mar.mean      2275.30     317.39   7.169 8.18e-13 ***
## m.wrkex$wrkex.mean  2333.50      94.55  24.681  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14150 on 8693 degrees of freedom
##   (287 observations deleted due to missingness)
## Multiple R-squared:  0.2306, Adjusted R-squared:  0.2303
## F-statistic: 868.5 on 3 and 8693 DF,  p-value: < 2.2e-16
```

```
#==use package====
between = plm(income ~  edu + mar + wrk.exp, e2, model = "between")
summary(between)
```

```
## Oneway (individual) effect Between Model
##
## Call:
## plm(formula = income ~ edu + mar + wrk.exp, data = e2, model = "between")
##
## Unbalanced Panel: n = 8599, T = 1-18, N = 81962
## Observations used in estimation: 8599
##
## Residuals:
##     Min.   1st Qu.   Median  3rd Qu.     Max.
## -50910.5  -8982.0  -2535.3   5766.6 272411.8
##
## Coefficients:
##             Estimate Std. Error t-value  Pr(>|t|)
## (Intercept) 3904.354    392.396   9.950 < 2.2e-16 ***
## edu         5822.949    151.850  38.347 < 2.2e-16 ***
## mar         3263.937    303.623  10.750 < 2.2e-16 ***
## wrk.exp     2060.835     74.068  27.823 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    2.7051e+12
## Residual Sum of Squares: 1.9929e+12
## R-Squared:      0.26328
## Adj. R-Squared: 0.26302
## F-statistic: 1023.85 on 3 and 8595 DF, p-value: < 2.22e-16
```

```
#4.2.3 Difference (any) Estimator
e3 = select(e,id,wave,income,edu,mar,wrk.exp)

e3$fir.inc = ave(e3$income,e3$id,FUN=function(x)x[1])
e3$fir.edu = ave(e3$edu,e3$id,FUN=function(x)x[1])
e3$fir.mar = ave(e3$mar,e3$id,FUN=function(x)x[1])
e3$fir.wrk.exp = ave(e3$wrk.exp,e3$id,FUN=function(x)x[1])

e3$fd.inc = e3$income - e3$fir.inc
e3$fd.edu = e3$edu - e3$fir.edu
e3$fd.mar = e3$mar - e3$fir.mar
e3$fd.wrk.exp = e3$wrk.exp - e3$fir.wrk.exp
```

```r
#panel.fd.estimator <- lm(fd.inc~fd.edu+fd.mar+fd.wrk.exp,e3)
#==use package====
fd = plm(income ~  edu + mar + wrk.exp, e2, model = "fd")
summary(fd)
```

```
## Oneway (individual) effect First-Difference Model
##
## Call:
## plm(formula = income ~ edu + mar + wrk.exp, data = e2, model = "fd")
##
## Unbalanced Panel: n = 8599, T = 1-18, N = 81962
## Observations used in estimation: 73363
##
## Residuals:
##      Min.   1st Qu.    Median   3rd Qu.      Max.
## -210826.2   -5959.1   -2166.4    4330.4  321870.0
##
## Coefficients:
##             Estimate Std. Error t-value  Pr(>|t|)
## (Intercept) 3849.249     69.380  55.481 < 2.2e-16 ***
## edu         1366.470    109.107  12.524 < 2.2e-16 ***
## mar         1674.430    159.429  10.503 < 2.2e-16 ***
## wrk.exp      947.909     29.596  32.028 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    2.1799e+13
## Residual Sum of Squares: 2.1409e+13
## R-Squared:      0.017857
## Adj. R-Squared: 0.017817
## F-statistic: 444.602 on 3 and 73359 DF, p-value: < 2.22e-16
```

#4.3 Interpret the results from each model and explain why different models yield different parameter estimates

```r
within.co = as.vector(c(NaN,within$coefficients))
between.co=as.vector(between$coefficients)
fd.co=as.factor(fd$coefficients)
result=data.frame(within.co,between.co,fd.co)
result
```

```
##             within.co between.co            fd.co
## (Intercept)       NaN   3904.354 3849.24945579281
## edu          9819.271   5822.949 1366.47021463334
## mar          7266.890   3263.937 1674.43012347055
## wrk.exp      2515.973   2060.835 947.909171980052
```

#the result in fd model has the smallest coefficient, while the within model has the largest coefficients. #the differences are due to the differences in different groups #within estimators indicate the differences on individual level; #between estimators indicate the differences between different individual; #fd estimators control the individual heterogeneity.