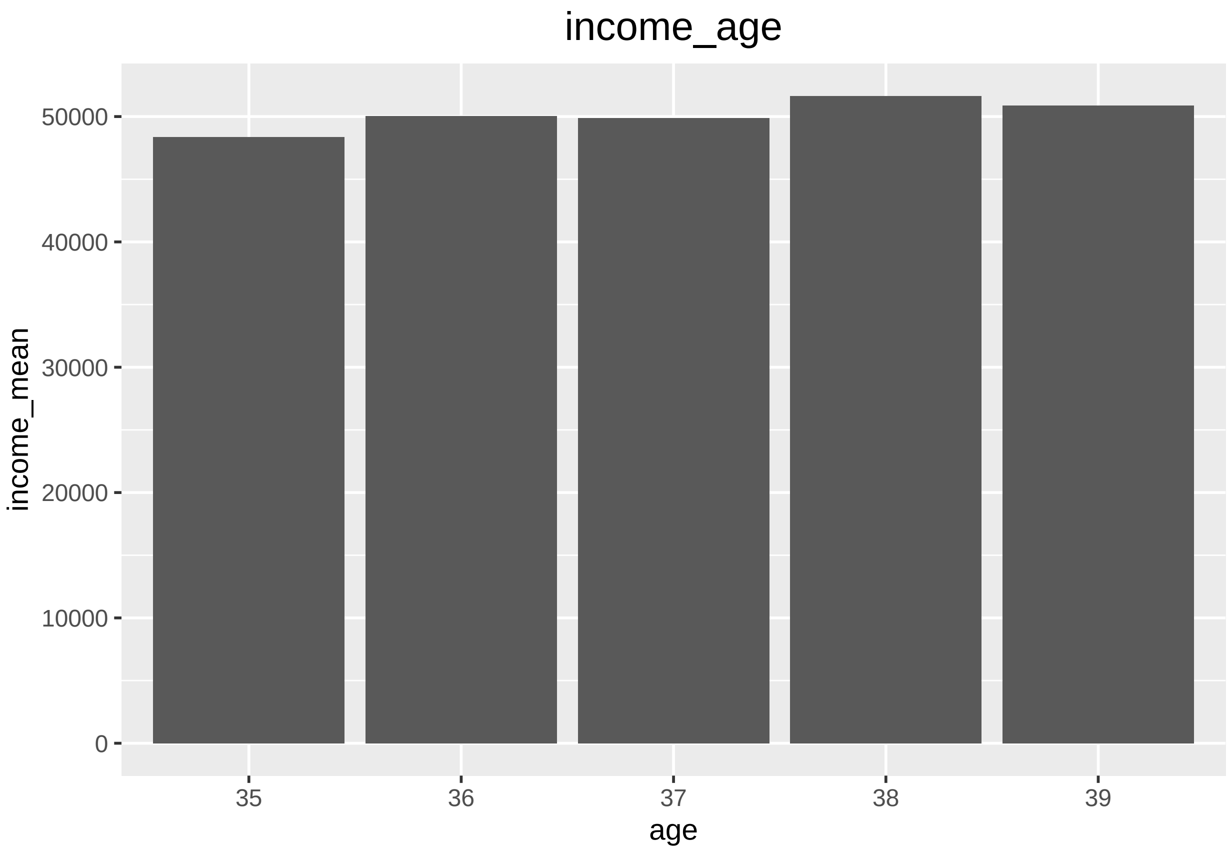
A4

Peilin Wang

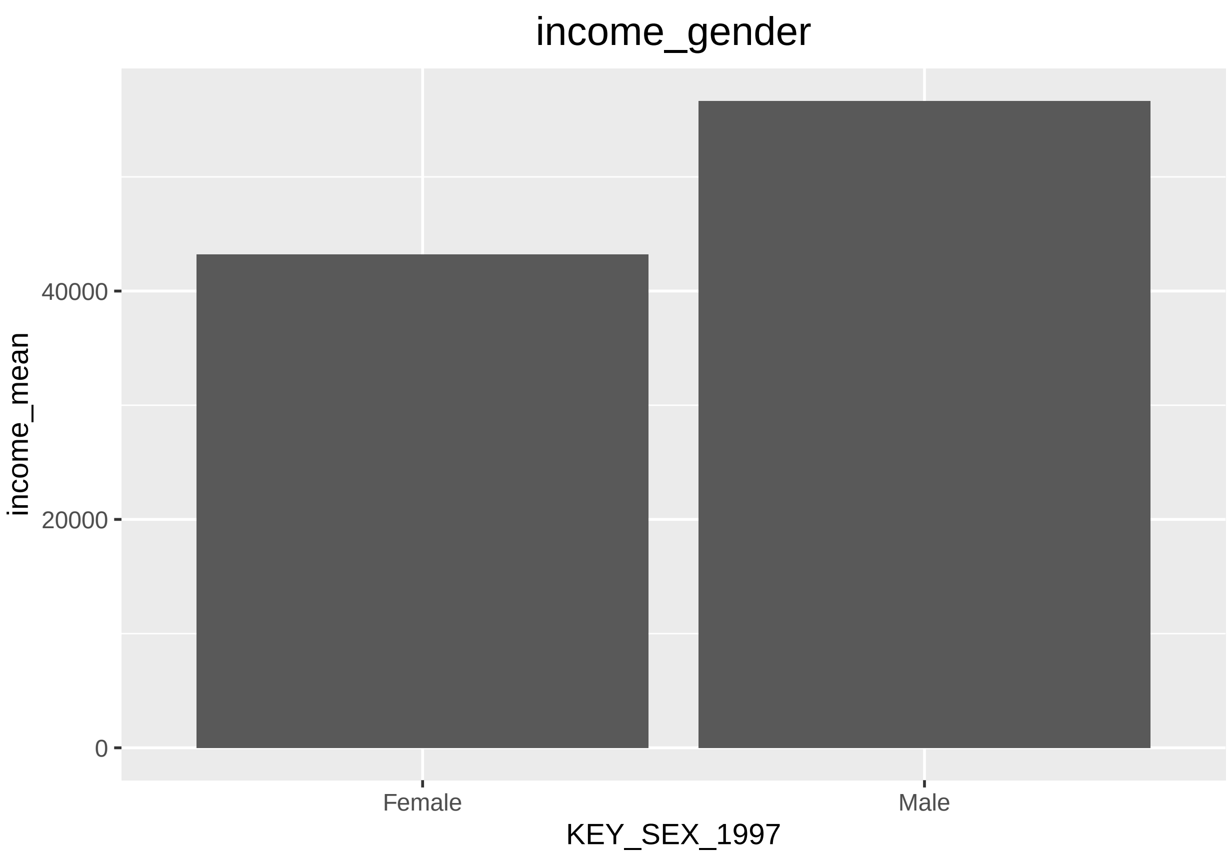
4/19/2022

Exercise 1

```
#1-1
dat_A4$age = 2019 - dat_A4$KEY_BDATE_Y_1997
dat_A4$work_exp = rowSums(dat_A4[,18:28],na.rm = "TRUE")/52

  #1-2
dat_A4$c1 = dat_A4$YSCH.3113_2019
dat_A4$c1[dat_A4$c1 == 1] = 0
dat_A4$c1[dat_A4$c1 == 2] = 4
dat_A4$c1[dat_A4$c1 == 3] = 12
dat_A4$c1[dat_A4$c1 == 4] = 14
dat_A4$c1[dat_A4$c1 == 5] = 16
dat_A4$c1[dat_A4$c1 == 6] = 18
dat_A4$c1[dat_A4$c1 == 7] = 23
dat_A4$c1[dat_A4$c1 == 8] = 21
dat_A4$CV_HGC_BIO_DAD_1997[dat_A4$CV_HGC_BIO_DAD_1997 == 95] = 0
dat_A4$CV_HGC_BIO_MOM_1997[dat_A4$CV_HGC_BIO_MOM_1997 == 95] = 0
dat_A4$CV_HGC_RES_DAD_1997[dat_A4$CV_HGC_RES_DAD_1997 == 95] = 0
dat_A4$CV_HGC_RES_MOM_1997[dat_A4$CV_HGC_RES_MOM_1997 == 95] = 0
dat_A4$edu_bio = rowSums(dat_A4[,8:9,33], na.rm = "TRUE")
dat_A4$edu_res = rowSums(dat_A4[,10:11,33], na.rm = "TRUE")

  #1-3
    #1-3-1
dat_income = subset(dat_A4, dat_A4$YINC_1700_2019 > 0)
income_age = dat_income %>% group_by(age) %>% summarise(income = mean(YINC_1700_2019))
income_age$age = as.factor(income_age$age)

income_gender = dat_income %>% group_by(KEY_SEX_1997) %>% summarise(income = mean(YINC_1700_2019))
income_gender$KEY_SEX_1997[income_gender$KEY_SEX_1997 == 1] = "Male"
income_gender$KEY_SEX_1997[income_gender$KEY_SEX_1997 == 2] = "Female"

income_chil = dat_income %>% group_by(CV_BIO_CHILD_HH_U18_2019) %>% summarise(income = mean(YINC_1700_20

ggplot(income_age,aes(x = age,y = income)) + geom_bar(stat='identity') + ylab("income_mean") +
  ggtitle("income_age") + theme(plot.title = element_text(size = 15L, hjust = 0.5))
```

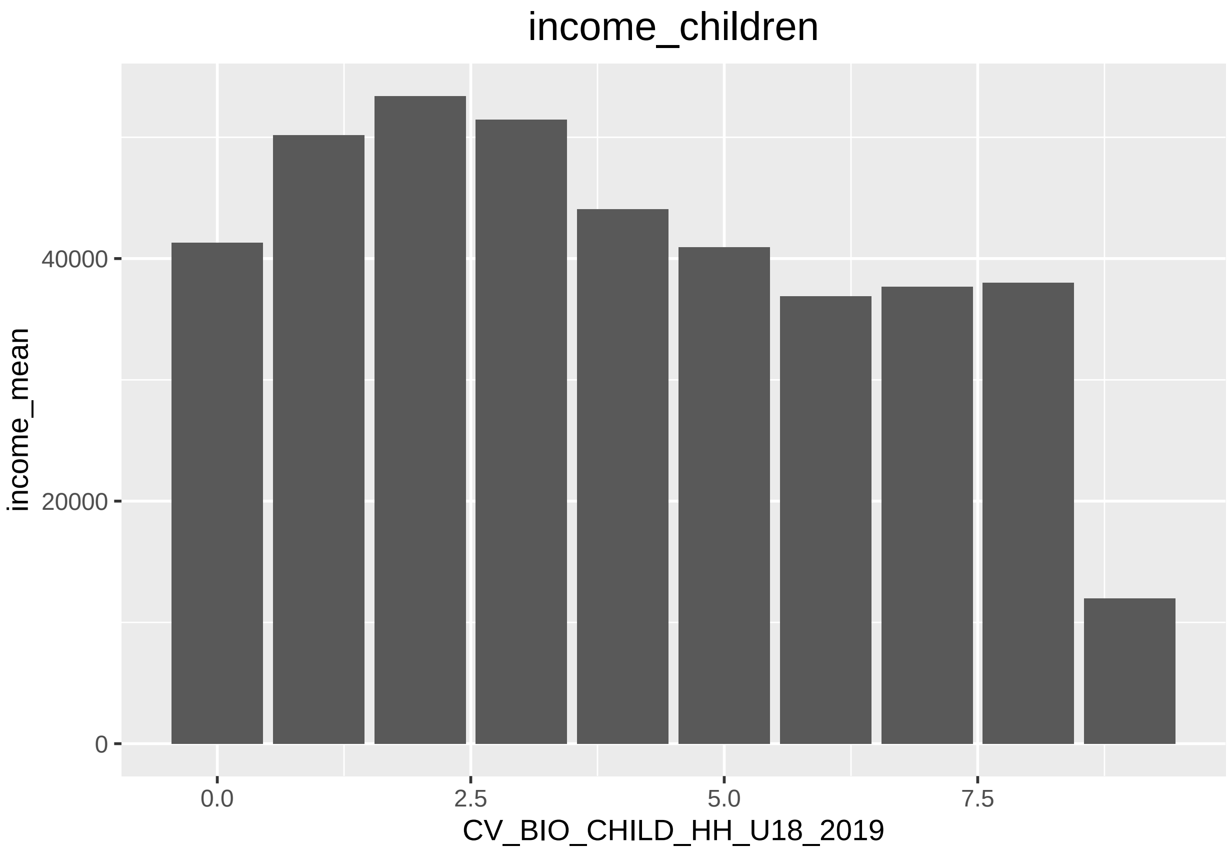



```
ggplot(income_gender,aes(x = KEY_SEX_1997,y = income)) + geom_bar(stat='identity') + ylab("income_mean")
  ggtitle("income_gender") + theme(plot.title = element_text(size = 15L, hjust = 0.5))
```

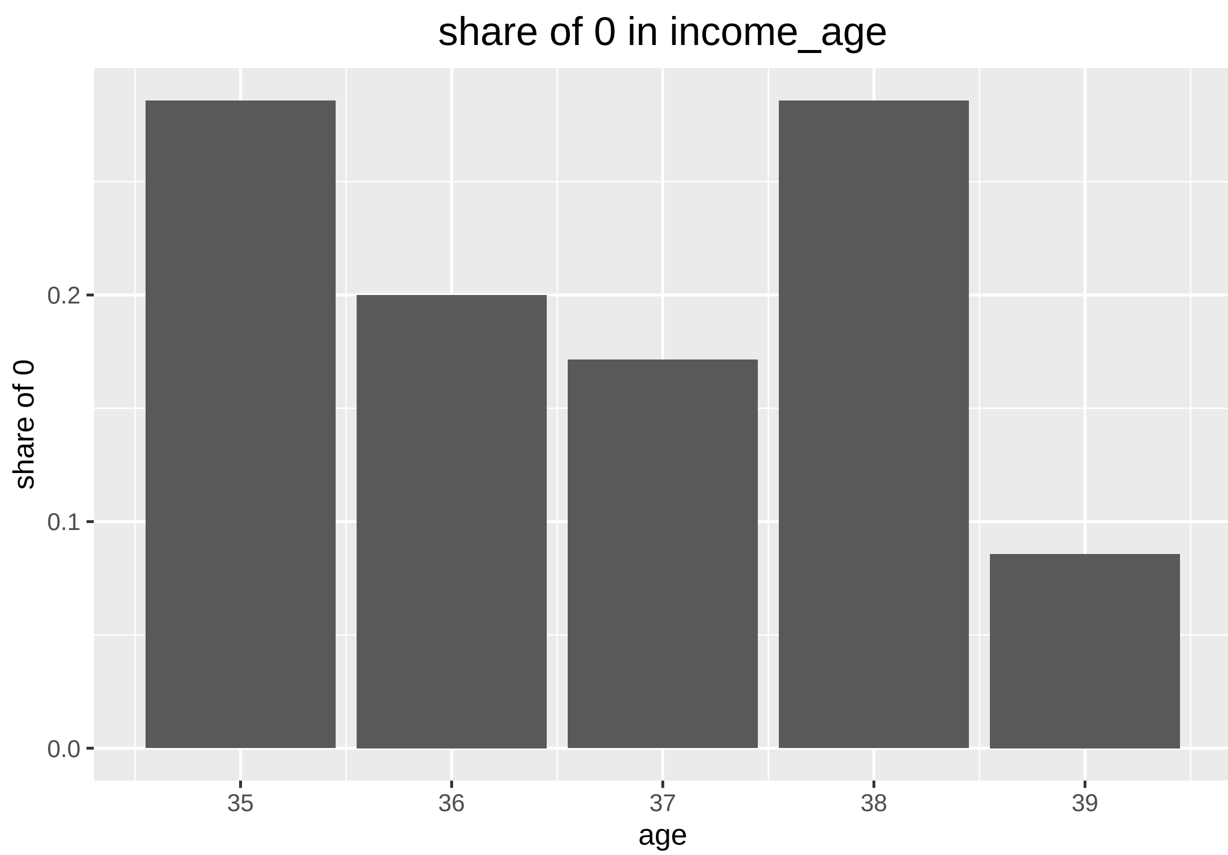



```
ggplot(income_chil,aes(x = CV_BIO_CHILD_HH_U18_2019,y = income)) + geom_bar(stat='identity') + ylab("in
  ggtitle("income_children") + theme(plot.title = element_text(size = 15L, hjust = 0.5))
```

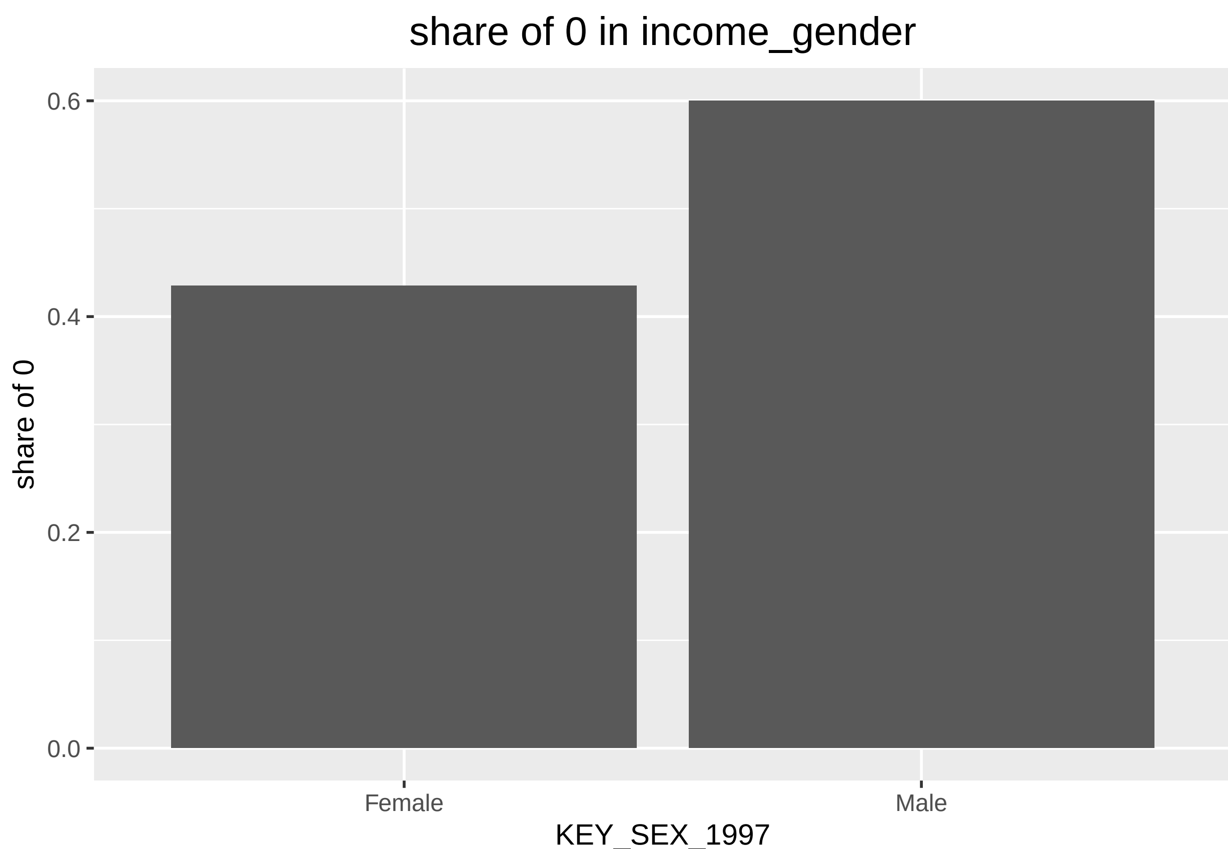
```
## Warning: Removed 1 rows containing missing values (position_stack).
```




```
    #1-3-2
income0_age = dat_A4 %>% group_by(age) %>% summarise(income0 = length(which(YINC_1700_2019 == 0))/
                                                        length(dat_A4))
income0_gender = dat_A4 %>% group_by(KEY_SEX_1997) %>% summarise(income0 = length(which(YINC_1700_2019 =
                                                        length(dat_A4))
income0_gender$KEY_SEX_1997[income0_gender$KEY_SEX_1997 == 1] = "Male"
income0_gender$KEY_SEX_1997[income0_gender$KEY_SEX_1997 == 2] = "Female"
income0_chil = dat_A4 %>% group_by(CV_BIO_CHILD_HH_U18_2019) %>% summarise(income0 = length(which(YINC_
                                                        length(dat_A4))
income0_mar = dat_A4 %>% group_by(CV_MARSTAT_COLLAPSED_2019) %>% summarise(income0 = length(which(YINC_
                                                                            length(dat_A4))
income0_mar$CV_MARSTAT_COLLAPSED_2019[income0_mar$CV_MARSTAT_COLLAPSED_2019 == 0] = "Never-married"
income0_mar$CV_MARSTAT_COLLAPSED_2019[income0_mar$CV_MARSTAT_COLLAPSED_2019 == 1] = "Married"
income0_mar$CV_MARSTAT_COLLAPSED_2019[income0_mar$CV_MARSTAT_COLLAPSED_2019 == 2] = "Separated"
income0_mar$CV_MARSTAT_COLLAPSED_2019[income0_mar$CV_MARSTAT_COLLAPSED_2019 == 3] = "Divorced"
income0_mar$CV_MARSTAT_COLLAPSED_2019[income0_mar$CV_MARSTAT_COLLAPSED_2019 == 4] = "Widowed"

ggplot(income0_age,aes(x = age,y = income0)) + geom_bar(stat='identity') + ylab("share of 0") +
  ggtitle("share of 0 in income_age") + theme(plot.title = element_text(size = 15L, hjust = 0.5))
```

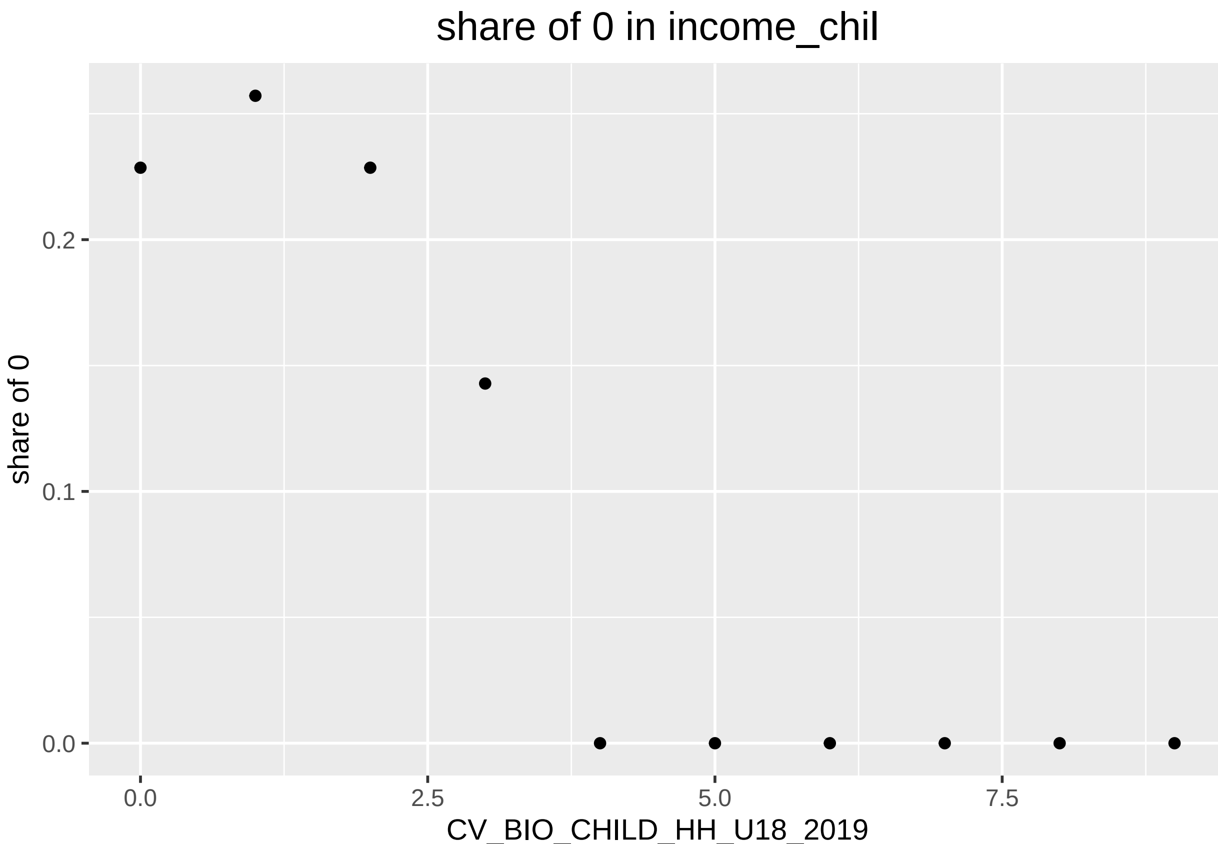



```
ggplot(income0_gender,aes(x = KEY_SEX_1997,y = income0)) + geom_bar(stat='identity') + ylab("share of 0
  ggtitle("share of 0 in income_gender") + theme(plot.title = element_text(size = 15L, hjust = 0.5))
```

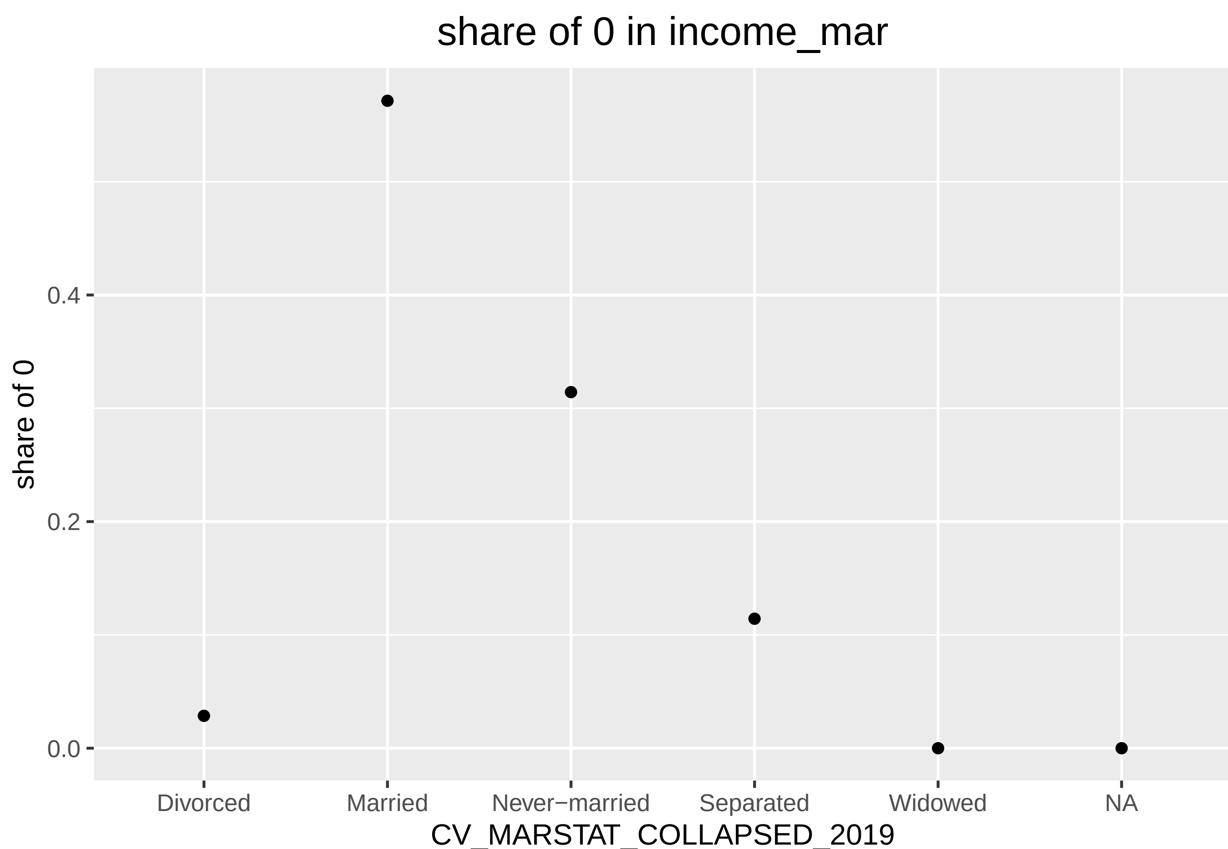



```
ggplot(income0_chil,aes(x = CV_BIO_CHILD_HH_U18_2019,y = income0)) + geom_point(stat='identity') + ylab
  ggtitle("share of 0 in income_chil") + theme(plot.title = element_text(size = 15L, hjust = 0.5))
```

```
## Warning: Removed 1 rows containing missing values (geom_point).
```




```
ggplot(income0_mar,aes(x = CV_MARSTAT_COLLAPSED_2019,y = income0)) + geom_point(stat='identity') + ylab
  ggtitle("share of 0 in income_mar") + theme(plot.title = element_text(size = 15L, hjust = 0.5))
```



##When income is positive, older people will have slightly high income. But, overall, there is no significant differences between different age groups. For gender group, male is more likely to have higher income than female. Household with 3 children will have the highest income. The income increases at first and then decreases with number of children increases.

##When analyzing the share of income is 0, age group 35 and 38 have larger proportion. Male have the higher proportion than female. Married people and household with one child are more likely to have high share of 0 income.

Exercise 2

```
#2-1
reg = lm(YINC_1700_2019 ~ age + work_exp + KEY_SEX_1997 + c1, data = dat_income)
summary(reg)
```

```
##
## Call:
## lm(formula = YINC_1700_2019 ~ age + work_exp + KEY_SEX_1997 +
##     c1, data = dat_income)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -82757 -18002  -2558  17311  94206
##
## Coefficients:
##                 Estimate Std. Error t value            Pr(>|t|)
## (Intercept)     18196.38    9226.66   1.972              0.0486 *
## age               381.16     246.97   1.543              0.1228
## work_exp         1055.14      64.18  16.441 <0.0000000000000002 ***
```

```
## KEY_SEX_1997 -14835.46     689.23 -21.525 <0.0000000000000002 ***
## c1             2375.85      84.01  28.279 <0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 25130 on 5367 degrees of freedom
##   (4 observations deleted due to missingness)
## Multiple R-squared:  0.2222, Adjusted R-squared:  0.2216
## F-statistic: 383.4 on 4 and 5367 DF,  p-value: < 0.00000000000000022
```

###interpret### ##If increasing one year in age, income will increase 381.16. If increasing work experience by one year, income will increase 1055.14. Female will have less income (14835.46) than male. If increasing education by one year, income will increase 2375.85. ###explain### ##Since only positive income is considered, the unemployed people with high educational level and work experience are not taken into account. It will cause bias because proper randomization is not achieved.

#2-2

The heckman model can be separated into two part. First of all, we run the probit model to make estimation. Then, we include IMR in OLS which has the function to reduce bias (selection bias).

```
  #2-3
dat = dat_A4 %>% mutate(income_exist = 0)
dat = subset(dat,dat$YSCH.3113_2019!='NA')
dat$income_exist[which(dat$YINC_1700_2019 > 0)] = 1
x1 = dat$KEY_SEX_1997
x2 = dat$age
x3 = dat$work_exp
x4 = dat$c1
y = dat$income_exist
prob = glm(y ~ x1+x2+x3+x4,family = binomial(link = "probit"), data = dat)
summary(prob)
```

```
##
## Call:
## glm(formula = y ~ x1 + x2 + x3 + x4, family = binomial(link = "probit"),
##     data = dat)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -4.0463   0.1311   0.4884   0.7462   1.5595
##
## Coefficients:
##              Estimate Std. Error z value             Pr(>|z|)
## (Intercept)  0.118197   0.484624   0.244                0.807
## x1          -0.227638   0.036447  -6.246       0.000000000422 ***
## x2          -0.005202   0.012973  -0.401                0.688
## x3           0.112625   0.004701  23.958 < 0.0000000000000002 ***
## x4           0.049519   0.003769  13.139 < 0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
```

```
##     Null deviance: 7404.4  on 6935  degrees of freedom
## Residual deviance: 6168.1  on 6931  degrees of freedom
## AIC: 6178.1
##
## Number of Fisher Scoring iterations: 7
```

```
dat$intercept = 1
intercept = dat$intercept
set.seed(123)
start = runif(7,-10,10)

prob_like = function(par, intercept, x1, x2, x3, x4,y) {
  yhat = par[1] * intercept + par[2] * x1 + par[3] * x2 + par[4] * x3 + par[5] * x4
  prob = pnorm(yhat)
  prob[prob > 0.999999] <- 0.999999
  prob[prob < 0.000001] <- 0.000001
  like = y * log(prob) + (1 - y) * log(1 - prob)
  return(-sum(like))
}
res  = optim(start,fn=prob_like,method="BFGS",control=list(trace=6,REPORT=1,maxit=1000),intercept = int
```

```
## initial  value 21220.938442
## iter   2 value 19952.084679
## iter   3 value 19551.204452
## iter   4 value 19044.084596
## iter   5 value 13103.691497
## iter   6 value 13040.756971
## iter   7 value 12913.703895
## iter   8 value 12628.858016
## iter   9 value 12496.560196
## iter  10 value 12442.333159
## iter  11 value 12392.585858
## iter  12 value 12364.700048
## iter  13 value 12353.206159
## iter  14 value 12339.288338
## iter  15 value 12333.050804
## iter  16 value 12325.866513
## iter  17 value 12321.946552
## iter  18 value 12320.563665
## iter  19 value 12316.345437
## iter  20 value 12314.929249
## iter  21 value 12314.573878
## iter  22 value 12313.723695
## iter  23 value 12313.460252
## iter  24 value 12313.381974
## iter  25 value 12312.710908
## iter  26 value 12311.533527
## iter  27 value 12303.014299
## iter  28 value 12302.434645
## iter  29 value 12302.229072
## iter  30 value 12299.745619
## iter  31 value 12298.814513
## iter  32 value 12298.243427
## iter  33 value 12295.236935
## iter  34 value 12295.214964
```

```
## iter  35 value 12293.083958
## iter  36 value 12289.634195
## iter  37 value 12289.261234
## iter  38 value 12289.258612
## iter  39 value 12288.948068
## iter  40 value 12286.139894
## iter  41 value 12282.432691
## iter  42 value 12183.615095
## iter  43 value 12181.415034
## iter  44 value 12181.363320
## iter  45 value 12153.179097
## iter  46 value 12091.054701
## iter  47 value 12090.270244
## iter  48 value 12078.538681
## iter  49 value 12074.933197
## iter  50 value 11986.888505
## iter  51 value 11675.450610
## iter  52 value 11294.244692
## iter  53 value 11271.810794
## iter  54 value 11249.649533
## iter  55 value 10982.855780
## iter  56 value 10592.275995
## iter  57 value 10587.768954
## iter  58 value 10491.752396
## iter  59 value 10488.281258
## iter  60 value 10433.260571
## iter  61 value 9638.890909
## iter  62 value 9392.759098
## iter  63 value 9232.616309
## iter  64 value 8800.054451
## iter  65 value 8417.851638
## iter  66 value 8289.484441
## iter  67 value 7627.115221
## iter  68 value 7455.522657
## iter  69 value 7230.241496
## iter  70 value 6457.876886
## iter  71 value 6440.205225
## iter  72 value 6233.393080
## iter  73 value 5431.350825
## iter  74 value 5428.258575
## iter  75 value 4720.872332
## iter  76 value 4711.797905
## iter  77 value 4664.704900
## iter  78 value 4377.966005
## iter  79 value 4316.167624
## iter  80 value 3478.228418
## iter  81 value 3179.239038
## iter  82 value 3091.827443
## iter  83 value 3084.072275
## iter  84 value 3084.060290
## iter  84 value 3084.060290
## iter  84 value 3084.060288
## final  value 3084.060288
## converged
```

```r
res$par
```

```
## [1]  0.061119313 -0.228085973 -0.003645621  0.112575029  0.049553430
## [6] -9.088870012  0.562109761
```

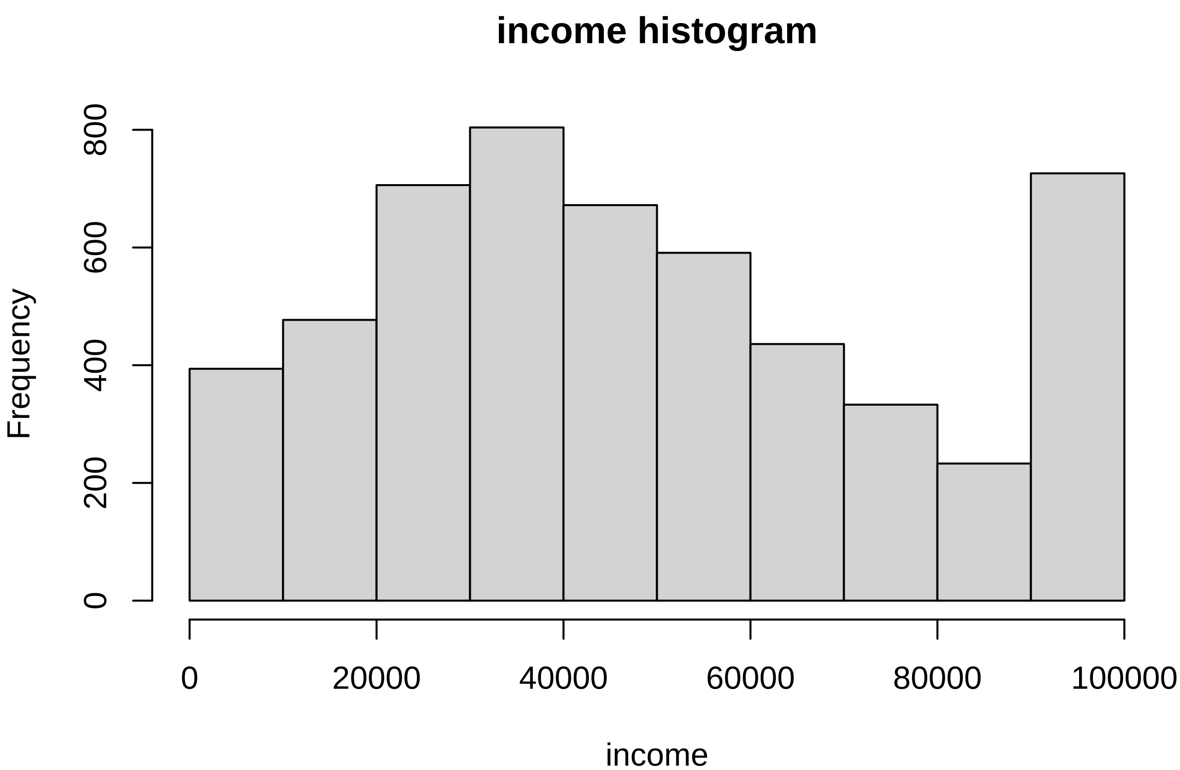
```r
predictor = function(par, intercept, x1, x2, x3, x4) {
  yhat = par[1] * intercept + par[2] * x1 + par[3] * x2 + par[4] * x3 + par[5] * x4

  return(yhat)
}
pred = predictor(res$par, intercept, x1, x2, x3, x4)
IMR = dnorm(pred)/pnorm(pred)
reg_heckman = lm(dat$YINC_1700_2019 ~ x1 + x2 + x3 + x4 + IMR)
summary(reg_heckman)
```

```
##
## Call:
## lm(formula = dat$YINC_1700_2019 ~ x1 + x2 + x3 + x4 + IMR)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -66060 -18215  -2715  17203  99343
##
## Coefficients:
##             Estimate Std. Error t value             Pr(>|t|)
## (Intercept)  38814.0     9696.5   4.003   0.0000634119851967 ***
## x1          -11432.7      811.1 -14.095 < 0.0000000000000002 ***
## x2             553.3      247.6   2.234               0.0255 *
## x3            -142.3      170.5  -0.835               0.4038
## x4            1531.3      139.9  10.946 < 0.0000000000000002 ***
## IMR         -38236.1     5020.4  -7.616   0.0000000000000307 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 25220 on 5402 degrees of freedom
##   (1528 observations deleted due to missingness)
## Multiple R-squared:  0.2276, Adjusted R-squared:  0.2269
## F-statistic: 318.3 on 5 and 5402 DF,  p-value: < 0.00000000000000022
```

##If increasing one year in age, income will increase 553.3. If increasing work experience by one year, income will idecrease 142.3. Female will have less income (11432.7) than male. If increasing education by one year, income will increase 1531.3. Comparing to the results from OLS, the work experience is not significant in heckman model.

Exercise 3

```r
#3-1
dat_income = subset(dat_income,dat_income$YSCH.3113_2019!='NA')
hist(dat_income$YINC_1700_2019,main = "income histogram", xlab = "income")
```



##the highest income is $100000.

```r
#3-2
reg_tobit = tobit(YINC_1700_2019 ~ KEY_SEX_1997 + age + work_exp + c1, left = -Inf, right = 100000,data
summary(reg_tobit)
```

```
##
## Call:
## tobit(formula = YINC_1700_2019 ~ KEY_SEX_1997 + age + work_exp +
##     c1, left = -Inf, right = 100000, data = dat_income)
##
## Observations:
##          Total  Left-censored     Uncensored Right-censored
##           5372              0           4735            637
##
## Coefficients:
##                 Estimate   Std. Error z value            Pr(>|z|)
## (Intercept)   13655.30231 10386.11748   1.315              0.1886
## KEY_SEX_1997 -16446.87907   776.42061 -21.183 <0.0000000000000002 ***
## age             506.98788   278.06565   1.823              0.0683 .
## work_exp       1125.99357    72.27063  15.580 <0.0000000000000002 ***
## c1             2621.20977    94.72729  27.671 <0.0000000000000002 ***
## Log(scale)       10.24029     0.01064 962.868 <0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Scale: 28009
##
## Gaussian distribution
## Number of Newton-Raphson Iterations: 4
## Log-likelihood: -5.597e+04 on 6 Df
```

```
## Wald-statistic:  1444 on 4 Df, p-value: < 0.0000000000000000222
```

```r
  #3-3
dat_income$intercept = 1
dat_income$indictor = 0
dat_income$indictor[which(dat_income$YINC_1700_2019 < 100000)] = 1
tobit_like = function(par, intercept, x1, x2, x3, x4,x5,y){
  yhat = par[1]*intercept + par[2]*x1 + par[3]*x2 + par[4]*x3 + par[5]*x4
  res = y - yhat
  standard = (100000-yhat)/exp(par[6])
  like = x5*log(dnorm(res/exp(par[6]))/exp(par[6])) + (1-x5)*log(1 - pnorm(standard))
  return(-sum(like))
}
start_1 = runif(6,-1000,1000)
res_2 = optim(start_1,fn=tobit_like,method="BFGS",control=list(trace=6,REPORT=1,maxit=3000),intercept =
               x1=dat_income$KEY_SEX_1997,x2=dat_income$age,x3=dat_income$work_exp,x4=dat_income$c1,x5=
```

```
## initial  value 1686386.626447
## iter   2 value 789577.626472
## iter   3 value 610215.826470
## iter   4 value 430854.026469
## iter   5 value 239613.594552
## iter   6 value 158051.293552
## iter   7 value 129998.408301
## iter   8 value 112962.629609
## iter   9 value 76672.866431
## iter  10 value 59851.243668
## iter  11 value 59734.971376
## iter  12 value 59709.356495
## iter  13 value 59694.828667
## iter  14 value 59659.825752
## iter  15 value 59547.851761
## iter  16 value 59219.573919
## iter  17 value 58902.610099
## iter  18 value 58871.291875
## iter  19 value 58851.292740
## iter  20 value 58810.614637
## iter  21 value 58705.817434
## iter  22 value 58441.857807
## iter  23 value 57839.931197
## iter  24 value 57268.591934
## iter  25 value 57012.033168
## iter  26 value 56635.764550
## iter  27 value 56573.272157
## iter  28 value 56449.458067
## iter  29 value 56442.891198
## iter  30 value 56421.684706
## iter  31 value 56415.105209
## iter  32 value 56414.931776
## iter  33 value 56414.428507
## iter  34 value 56413.221948
## iter  35 value 56410.845197
## iter  36 value 56407.521906
## iter  37 value 56405.217341
## iter  38 value 56404.549726
```

```
## iter  39 value 56404.403243
## iter  40 value 56404.311974
## iter  41 value 56404.034873
## iter  42 value 56403.361924
## iter  43 value 56403.070128
## iter  44 value 56402.965393
## iter  45 value 56402.908100
## iter  46 value 56402.550293
## iter  47 value 56401.835927
## iter  48 value 56399.888682
## iter  49 value 56395.654413
## iter  50 value 56387.761177
## iter  51 value 56378.743153
## iter  52 value 56373.824076
## iter  53 value 56372.013745
## iter  54 value 56371.119577
## iter  55 value 56369.695064
## iter  56 value 56369.142979
## iter  57 value 56368.634551
## iter  58 value 56367.385619
## iter  59 value 56364.525007
## iter  60 value 56359.001340
## iter  61 value 56351.861252
## iter  62 value 56347.663066
## iter  63 value 56346.653861
## iter  64 value 56346.492258
## iter  65 value 56346.449105
## iter  66 value 56346.289840
## iter  67 value 56345.927780
## iter  68 value 56344.925320
## iter  69 value 56344.722311
## iter  70 value 56344.642979
## iter  71 value 56344.591094
## iter  72 value 56344.307861
## iter  73 value 56343.744133
## iter  74 value 56342.321858
## iter  75 value 56339.671318
## iter  76 value 56336.156160
## iter  77 value 56333.891885
## iter  78 value 56333.120401
## iter  79 value 56332.783567
## iter  80 value 56332.319393
## iter  81 value 56331.027306
## iter  82 value 56330.371209
## iter  83 value 56330.108338
## iter  84 value 56329.917114
## iter  85 value 56329.047908
## iter  86 value 56327.546941
## iter  87 value 56324.985357
## iter  88 value 56322.866157
## iter  89 value 56322.087076
## iter  90 value 56321.934119
## iter  91 value 56321.883666
## iter  92 value 56321.764332
```

```
## iter  93 value 56321.457716
## iter  94 value 56320.650644
## iter  95 value 56320.300458
## iter  96 value 56320.197806
## iter  97 value 56320.167587
## iter  98 value 56319.864787
## iter  99 value 56319.291975
## iter 100 value 56317.601181
## iter 101 value 56313.526921
## iter 102 value 56303.486550
## iter 103 value 56281.897781
## iter 104 value 56246.743480
## iter 105 value 56217.997183
## iter 106 value 56206.950517
## iter 107 value 56203.195783
## iter 108 value 56203.186340
## iter 109 value 56202.754686
## iter 110 value 56183.760335
## iter 111 value 56181.652020
## iter 112 value 56181.492282
## iter 113 value 56181.483770
## iter 114 value 56181.482180
## iter 115 value 56181.476275
## iter 116 value 56181.462761
## iter 117 value 56181.425497
## iter 118 value 56181.330393
## iter 119 value 56181.083260
## iter 120 value 56180.464666
## iter 121 value 56180.242168
## iter 122 value 56180.180002
## iter 123 value 56180.165523
## iter 124 value 56180.018369
## iter 125 value 56179.840417
## iter 126 value 56179.618525
## iter 127 value 56179.524449
## iter 128 value 56179.506942
## iter 129 value 56179.504686
## iter 130 value 56179.502386
## iter 131 value 56179.495116
## iter 132 value 56179.477452
## iter 133 value 56179.429954
## iter 134 value 56179.407847
## iter 135 value 56179.401613
## iter 136 value 56179.400204
## iter 137 value 56179.381501
## iter 138 value 56179.350075
## iter 139 value 56179.272498
## iter 140 value 56179.156232
## iter 141 value 56179.044876
## iter 142 value 56178.996058
## iter 143 value 56178.980378
## iter 144 value 56178.966026
## iter 145 value 56178.928265
## iter 146 value 56178.834096
```

```
## iter 147 value 56178.769411
## iter 148 value 56178.750566
## iter 149 value 56178.745498
## iter 150 value 56178.695570
## iter 151 value 56178.629964
## iter 152 value 56178.536496
## iter 153 value 56178.486068
## iter 154 value 56178.473441
## iter 155 value 56178.471032
## iter 156 value 56178.468199
## iter 157 value 56178.459996
## iter 158 value 56178.439528
## iter 159 value 56178.385099
## iter 160 value 56178.350018
## iter 161 value 56178.341100
## iter 162 value 56178.340065
## iter 163 value 56178.315037
## iter 164 value 56178.271661
## iter 165 value 56178.142308
## iter 166 value 56177.856342
## iter 167 value 56177.262915
## iter 168 value 56176.410882
## iter 169 value 56175.735944
## iter 170 value 56175.475141
## iter 171 value 56175.380762
## iter 172 value 56175.271137
## iter 173 value 56175.232336
## iter 174 value 56175.182367
## iter 175 value 56175.008083
## iter 176 value 56174.670143
## iter 177 value 56174.059942
## iter 178 value 56173.441776
## iter 179 value 56173.163728
## iter 180 value 56173.117609
## iter 181 value 56173.114173
## iter 182 value 56173.113258
## iter 183 value 56173.109079
## iter 184 value 56173.100059
## iter 185 value 56173.074515
## iter 186 value 56173.071371
## iter 187 value 56173.069660
## iter 188 value 56173.067665
## iter 189 value 56173.060245
## iter 190 value 56173.044841
## iter 191 value 56173.011575
## iter 192 value 56172.963079
## iter 193 value 56172.922293
## iter 194 value 56172.906588
## iter 195 value 56172.901850
## iter 196 value 56172.896835
## iter 197 value 56172.883170
## iter 198 value 56172.848868
## iter 199 value 56172.825931
## iter 200 value 56172.819324
```

```
## iter 201 value 56172.817669
## iter 202 value 56172.799113
## iter 203 value 56172.772607
## iter 204 value 56172.727169
## iter 205 value 56172.693521
## iter 206 value 56172.681171
## iter 207 value 56172.678201
## iter 208 value 56172.675420
## iter 209 value 56172.667641
## iter 210 value 56172.648274
## iter 211 value 56172.596644
## iter 212 value 56172.557450
## iter 213 value 56172.547815
## iter 213 value 56172.547025
## iter 213 value 56172.547025
## final  value 56172.547025
## converged
```

```
res_2$par
```

```
## [1]  796.43729 -410.55865  290.94411 1180.35612 2375.75118   10.28029
```

#3-4 ##Female will have lower income than male. If age, work experience,or educational level increases, the income will increase.

Exercise 4

#4-1 ##The ability bias indicates the relation between educational level and innate skills. People with innate skills are more likely to go to school. Also, there exists the casual relationship between educational level and income.

```
  #4-2
dat_A4_panel = read_csv("Data/dat_A4_panel.csv")
```

```
## New names:
## * `` -> ...1
```

```
## Warning: One or more parsing issues, see `problems()` for details
```

```
## Rows: 8984 Columns: 249
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (203): ...1, PUBID_1997, YINC-1700_1997, KEY_SEX_1997, KEY_BDATE_M_1997,...
## lgl  (46): CV_WKSWK_JOB_DLI.06_1997, CV_WKSWK_JOB_DLI.07_1997, CV_WKSWK_JOB_...
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
dat_A4_panel = dat_A4_panel %>% rename(CV_HIGHEST_DEGREE_EVER_EDT_1998=CV_HIGHEST_DEGREE_9899_1998) %>%
  rename(CV_HIGHEST_DEGREE_EVER_EDT_2000=CV_HIGHEST_DEGREE_0001_2000) %>% rename(CV_HIGHEST_DEGREE_EVER_
  rename(CV_HIGHEST_DEGREE_EVER_EDT_2002=CV_HIGHEST_DEGREE_0203_2002) %>% rename(CV_HIGHEST_DEGREE_EVER_
  rename(CV_HIGHEST_DEGREE_EVER_EDT_2004=CV_HIGHEST_DEGREE_0405_2004) %>% rename(CV_HIGHEST_DEGREE_EVER_
  rename(CV_HIGHEST_DEGREE_EVER_EDT_2006=CV_HIGHEST_DEGREE_0607_2006) %>% rename(CV_HIGHEST_DEGREE_EVER_
  rename(CV_HIGHEST_DEGREE_EVER_EDT_2008=CV_HIGHEST_DEGREE_0809_2008) %>% rename(CV_HIGHEST_DEGREE_EVER_

dat_A4_panel_long = long_panel(dat_A4_panel,prefix='_',begin  = 1997, end = 2019,label_location = "end")
dat_A4_panel_long = dat_A4_panel_long %>% rename(edu = CV_HIGHEST_DEGREE_EVER_EDT) %>% rename(year = wa
```

```r
                  rename(marital = CV_MARSTAT_COLLAPSED)
colnames(dat_A4_panel_long)[5] = "income"

dat_A4_panel_long$age = dat_A4_panel_long$year - dat_A4_panel_long$KEY_BDATE_Y
dat_A4_panel_long$work_exp= rowSums(dat_A4_panel_long[,10:16], na.rm = "TRUE")/52 +
                            rowSums(dat_A4_panel_long[,23:30], na.rm = "TRUE")/52

dat_A4_panel_long$edu[dat_A4_panel_long$edu == 0] = 0
dat_A4_panel_long$edu[dat_A4_panel_long$edu == 1] = 4
dat_A4_panel_long$edu[dat_A4_panel_long$edu == 2] = 12
dat_A4_panel_long$edu[dat_A4_panel_long$edu == 3] = 14
dat_A4_panel_long$edu[dat_A4_panel_long$edu == 4] = 16
dat_A4_panel_long$edu[dat_A4_panel_long$edu == 5] = 18
dat_A4_panel_long$edu[dat_A4_panel_long$edu == 6] = 23
dat_A4_panel_long$edu[dat_A4_panel_long$edu == 7] = 21
#===============================================================================
#Within Estimator: work_exp/education/marital status
#===============================================================================
dat_A4_panel_long$mean_income = ave(dat_A4_panel_long$income, dat_A4_panel_long$id, FUN = function(x)mea
dat_A4_panel_long$mean_exp = ave(dat_A4_panel_long$work_exp, dat_A4_panel_long$id, FUN = function(x)mean
dat_A4_panel_long$mean_edu = ave(dat_A4_panel_long$edu, dat_A4_panel_long$id, FUN = function(x)mean(x,na
dat_A4_panel_long$mean_mar = ave(dat_A4_panel_long$marital, dat_A4_panel_long$id, FUN = function(x)mean

dat_A4_panel_long$income_diff = dat_A4_panel_long$income - dat_A4_panel_long$mean_income
dat_A4_panel_long$exp_diff = dat_A4_panel_long$work_exp - dat_A4_panel_long$mean_exp
dat_A4_panel_long$edu_diff = dat_A4_panel_long$edu - dat_A4_panel_long$mean_edu
dat_A4_panel_long$mar_diff = dat_A4_panel_long$marital - dat_A4_panel_long$mean_mar

within = lm(income_diff ~ exp_diff + edu_diff + mar_diff, dat_A4_panel_long)
summary(within)
```

```
##
## Call:
## lm(formula = income_diff ~ exp_diff + edu_diff + mar_diff, data = dat_A4_panel_long)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -129398   -9599    -887    7787  278863
##
## Coefficients:
##              Estimate Std. Error t value            Pr(>|t|)
## (Intercept) -3604.88      80.58  -44.74 <0.0000000000000002 ***
## exp_diff     2562.91      25.48  100.57 <0.0000000000000002 ***
## edu_diff     1299.58      19.90   65.31 <0.0000000000000002 ***
## mar_diff     8942.25     142.91   62.57 <0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 20660 on 82004 degrees of freedom
##   (124624 observations deleted due to missingness)
## Multiple R-squared:  0.2794, Adjusted R-squared:  0.2794
## F-statistic: 1.06e+04 on 3 and 82004 DF,  p-value: < 0.00000000000000022
```

```r
#==============================================================================
#Between Estimator: work_exp/education/marital status
#==============================================================================
y.1 = dat_A4_panel_long %>% group_by(id) %>% select(mean_income)
```

```
## Adding missing grouping variables: `id`
```

```r
y.1 = y.1[!duplicated(y.1$id),]
x.exp = dat_A4_panel_long %>% group_by(id) %>% select(mean_exp)
```

```
## Adding missing grouping variables: `id`
```

```r
x.exp = x.exp[!duplicated(x.exp$id),]
x.edu = dat_A4_panel_long %>% group_by(id) %>% select(mean_edu)
```

```
## Adding missing grouping variables: `id`
```

```r
x.edu = x.edu[!duplicated(x.edu$id),]
x.mar = dat_A4_panel_long %>% group_by(id) %>% select(mean_mar)
```

```
## Adding missing grouping variables: `id`
```

```r
x.mar = x.mar[!duplicated(x.mar$id),]

between = y.1 %>% left_join(x.exp) %>% left_join(x.edu) %>% left_join(x.mar)
```

```
## Joining, by = "id"
```

```
## Joining, by = "id"
## Joining, by = "id"
```

```r
between_reg = lm(mean_income ~ mean_edu + mean_mar + mean_exp, data = between)
summary(between_reg)
```

```
##
## Call:
## lm(formula = mean_income ~ mean_edu + mean_mar + mean_exp, data = between)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -43440  -8968  -2824   5496 171214
##
## Coefficients:
##             Estimate Std. Error t value             Pr(>|t|)
## (Intercept)  5125.81     441.40  11.613 < 0.0000000000000002 ***
## mean_edu     1014.13      41.56  24.403 < 0.0000000000000002 ***
## mean_mar     1927.29     327.15   5.891        0.00000000398 ***
## mean_exp     3222.53     116.80  27.591 < 0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14570 on 8693 degrees of freedom
##   (287 observations deleted due to missingness)
## Multiple R-squared:  0.1841, Adjusted R-squared:  0.1838
## F-statistic: 653.8 on 3 and 8693 DF,  p-value: < 0.00000000000000022
```

```r
#==============================================================================
#Difference Estimator: work_exp/education/marital status
```

```r
#==========================================================================
difference = dat_A4_panel_long %>% group_by(id) %>% mutate(income_diff= income-lag(income)) %>% mutate(
diff_reg = lm(income_diff~ edu_diff + mar_diff + work_diff, data = difference)

fd = plm(income ~  edu + marital + work_exp, dat_A4_panel_long, model = "fd")
summary(fd)
```

```
## Oneway (individual) effect First-Difference Model
##
## Call:
## plm(formula = income ~ edu + marital + work_exp, data = dat_A4_panel_long,
##     model = "fd")
##
## Unbalanced Panel: n = 8600, T = 1-18, N = 82008
## Observations used in estimation: 73408
##
## Residuals:
##      Min.   1st Qu.    Median   3rd Qu.      Max.
## -210994.8   -5871.0   -2148.4    4258.5  321674.9
##
## Coefficients:
##             Estimate Std. Error t-value              Pr(>|t|)
## (Intercept) 4035.838     68.787 58.6711 < 0.00000000000000022 ***
## edu           78.928     20.599  3.8317             0.0001274 ***
## marital     1697.743    159.531 10.6421 < 0.00000000000000022 ***
## work_exp     952.566     29.684 32.0898 < 0.00000000000000022 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    21799000000000
## Residual Sum of Squares: 21454000000000
## R-Squared:      0.015837
## Adj. R-Squared: 0.015796
## F-statistic: 393.726 on 3 and 73404 DF, p-value: < 0.000000000000000222
```

```r
  #4-3
##Coefficients have the same sign in these three models, but difference estimation gives the smallest c
```