# A1.R

tinchan

2022-01-21

```
## Author: Tian Chen
## Purpose: Econ613 Assignment 01
## Date: 01/16/2022
rm(list = ls())
require(tidyverse)
```

```
## Loading required package: tidyverse
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.5     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.0.1     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
##################################
##  Exercise 1 Basic Statistics ##
##################################
#dathh: household survey
#idind: individual survey
## Number of household surveyed in 2007
dathh2007 <- read.csv("~/Desktop/Duke study/Econ613/A1/Data/dathh2007.csv")
length(dathh2007$idmen)
```

```
## [1] 10498
```

```
## Number of household with marital status "Couple with kids" in 2005
dathh2005 <- read.csv("~/Desktop/Duke study/Econ613/A1/Data/dathh2005.csv")
length(dathh2005[dathh2005$mstatus == "Couple, with Kids",]$idmen)
```

```
## [1] 3374
```

```
## Number of individuals surveyed in 2008
datind2008 <- read.csv("~/Desktop/Duke study/Econ613/A1/Data/datind2008.csv")
length(datind2008$idind)
```

```
## [1] 25510
```

```
## Number of individuals aged between 25 and 35 in 2016
datind2016 <- read.csv("~/Desktop/Duke study/Econ613/A1/Data/datind2016.csv")
length(datind2016[between(datind2016$age, 25, 35),]$idind)
```

```
## [1] 2765
```

```
## Cross-table gender/profession in 2007
datind2007 <- read.csv("~/Desktop/Duke study/Econ613/A1/Data/datind2007.csv")
crosstable_gender_profession <- table(datind2007$gender, datind2007$profession)
crosstable_gender_profession
```

```
##
##             0  11  12  13  21  22  23  31  33  34  35  37  38  42  43  44  45
##   Female    1  41  11  21  57  77   9  62  73 189  47 172  71 271 414   0 151
##   Male      1  77  14  73 212 118  56 105 100 153  51 253 380 123 100   2  87
##
##            46  47  48  52  53  54  55  56  62  63  64  65  67  68  69
##   Female  355  69  24 848  25 606 371 676  87  54  22  26 168 102  27
##   Male    330 416 241 206 207 107  99  65 485 537 255 175 281 185  75
```

```
## Distribution of wages in 2005 and 2019. Report the mean,
## the standard deviation, the inter-decile ratio
## D9/D1 and the Gini coefficient.
datind2005 <- read.csv("~/Desktop/Duke study/Econ613/A1/Data/datind2005.csv")
datind2019 <- read.csv("~/Desktop/Duke study/Econ613/A1/Data/datind2019.csv")

#mean
mean(datind2005$wage, na.rm = TRUE) #2005 mean 11992.26
```

```
## [1] 11992.26
```

```
mean(datind2019$wage, na.rm = TRUE) #2019 mean 15350.47
```

```
## [1] 15350.47
```

```
#standard deviation
sd(datind2005$wage, na.rm = TRUE) #2005 sd 17318.56
```

```
## [1] 17318.56
```

```
sd(datind2019$wage, na.rm = TRUE) #2019 sd 23207.18
```

```
## [1] 23207.18
```

```
#inter-decile ratio D9/D1
quantile(datind2005$wage, probs = 0.9, na.rm = TRUE)/quantile(datind2005$wage, probs = 0.1, na.rm = TRU
```

```
## 90%
## Inf
```

```
quantile(datind2019$wage, probs = 0.9, na.rm = TRUE)/quantile(datind2019$wage, probs = 0.1, na.rm = TRU
```

```
## 90%
## Inf
```

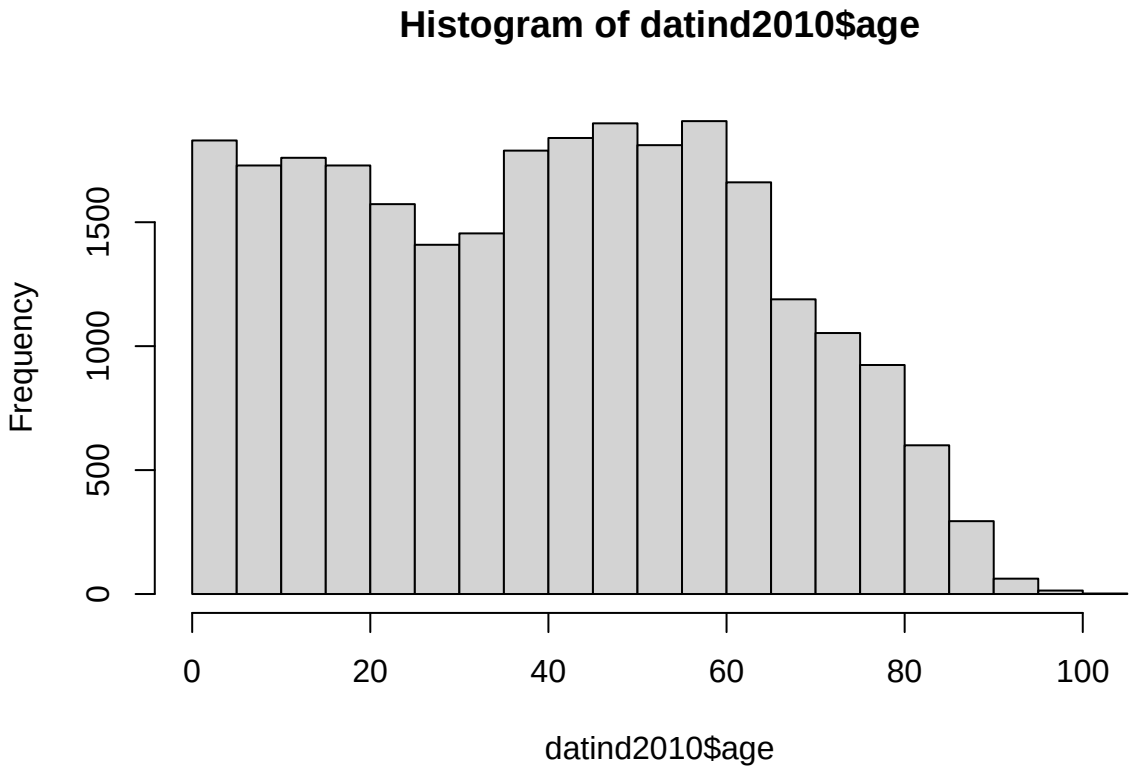
```
#Gini Coefficient
gini_2005 <- datind2005 %>% as_tibble() %>% select(wage) %>% filter(!is.na(wage)) %>%
  arrange(wage) %>% mutate(gini = sum(2*(rank(wage)/n() - cumsum(wage)/sum(wage)))/n()) %>%
  select(gini) %>% distinct()
gini_2005 #Gini 2005: 0.667
```

```
## # A tibble: 1 x 1
##    gini
##   <dbl>
## 1 0.667
```

```
gini_2019 <- datind2019 %>% as_tibble() %>% select(wage) %>% filter(!is.na(wage)) %>%
  arrange(wage) %>% mutate(gini = sum(2*(rank(wage)/n() - cumsum(wage)/sum(wage)))/n()) %>%
  select(gini) %>% distinct()
gini_2019 #Gini 2019: 0.666
```

```
## # A tibble: 1 x 1
##    gini
##   <dbl>
## 1 0.666
```

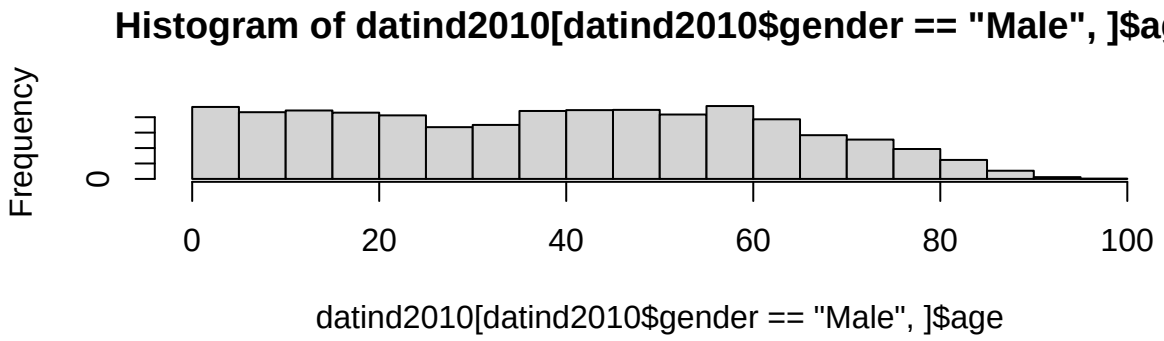
```
##Distribution of age in 2010. Plot an histogram. Is there any difference between men and women?
datind2010 <- read.csv("~/Desktop/Duke study/Econ613/A1/Data/datind2010.csv")
hist(datind2010$age) #all respondents
```

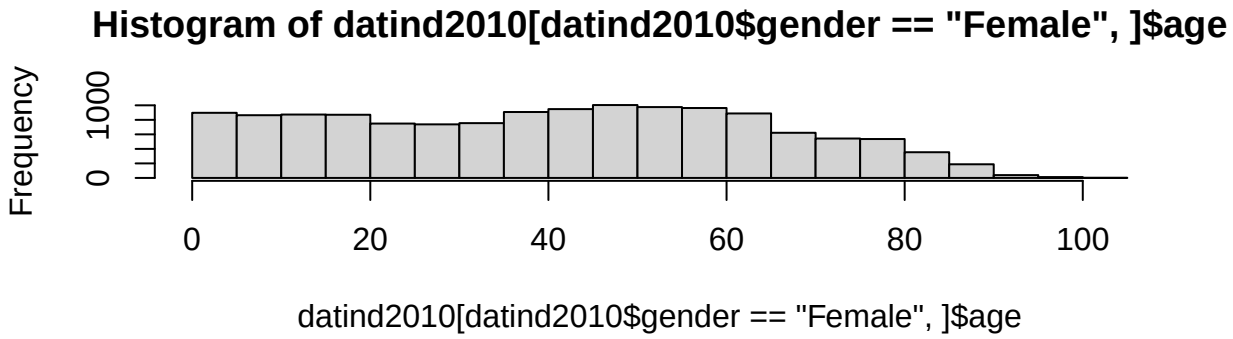



```
split.screen(c(2,1)) #split the screen for comparision
```

```
## [1] 1 2
```

```
screen(1)
hist(datind2010[datind2010$gender == "Male", ]$age) #male
screen(2)
hist(datind2010[datind2010$gender == "Female", ]$age) #female
```




```
##Number of individuals in Paris in 2011
datind2011 <- read.csv("~/Desktop/Duke study/Econ613/A1/Data/datind2011.csv")
dathh2011 <- read.csv("~/Desktop/Duke study/Econ613/A1/Data/dathh2011.csv")
da_combined <- left_join(datind2011, dathh2011, by = "idmen")
length(da_combined[da_combined$location == "Paris",]$idind)
```

```
## [1] 3531
```

```
#####################################
##  Exercise 2 Merge Data sets     ##
#####################################
rm(list = ls()) #clear the environment
path <- "~/Desktop/Duke study/Econ613/A1/Data/"
setwd(path)

## Read all individual datasets from 2004 to 2019. Append all these dataset.
datind_file <- list.files(path,pattern = "^datind")
datind_combined <- read.csv(datind_file[1])
datind_file <- datind_file[-1]
for(file in datind_file){
  csv <- read.csv(file)
  datind_combined <- rbind(datind_combined, csv)
}

## Read all household datasets from 2004 to 2019. Append all these dataset.
dathh_file <- list.files(path,pattern = "^dathh")
dathh_combined <- read.csv(dathh_file[1])
dathh_file <- dathh_file[-1]
for(file in dathh_file){
  csv <- read.csv(file)
```

```
  dathh_combined <- rbind(dathh_combined, csv)
}
rm(csv)

## List the variables that are simultaneously present in the individual and household datasets.
datind_names <- names(datind_combined)
dathh_names <- names(dathh_combined)

# Variables that are simultaneously present: X, idmen, year,

## Merge the appended individual and household datasets.
combined <- full_join(datind_combined, dathh_combined, by = c("idmen", "year"))
combined_tbl <- combined %>% as_tibble()

## Number of households in which there are more than four family members
combined_household <- combined_tbl %>% select(idmen, idind, year) %>% group_by(idmen,year) %>%
                      summarise(householdmember = n()) %>% filter(householdmember > 4) %>%
                      select(idmen) %>% unique()
```

```
## `summarise()` has grouped output by 'idmen'. You can override using the `.groups` argument.
```

```
length(combined_household$idmen)
```

```
## [1] 3622
```

```
## Number of households in which at least one member is unemployed
combined_employment <- combined_tbl %>% select(idmen, idind, year, empstat) %>%
                      filter(empstat == "Unemployed") %>% group_by(idmen, year) %>%
                      summarise(household = n()) %>% select(idmen) %>% unique()
```

```
## `summarise()` has grouped output by 'idmen'. You can override using the `.groups` argument.
```

```
length(combined_employment$idmen)
```

```
## [1] 8162
```

```
## Number of households in which at least two members are of the same profession
combined_profession <- combined_tbl %>% select(idmen, idind, year, profession) %>% filter(!profession ==
                      group_by(idmen, year, profession) %>% summarise(sameprofession = n()) %>%
                      filter(sameprofession >= 2) %>% select(idmen) %>% unique()
```

```
## `summarise()` has grouped output by 'idmen', 'year'. You can override using the `.groups` argument.
## Adding missing grouping variables: `year`
```

```
length(combined_profession$idmen)
```

```
## [1] 7586
```

```
## Number of individuals in the panel that are from household-Couple with kids
combined_couple_kids <- combined_tbl %>% select(idmen, idind, year, mstatus) %>%
                        filter(mstatus == "Couple, with Kids") %>% select(idmen) %>% unique()
length(combined_couple_kids$idmen)
```

```
## [1] 13930
```

```
## Number of individuals in the panel that are from Paris.
combined_paris <- combined_tbl %>% select(idmen, idind, location) %>% filter(location == "Paris") %>%
                  select(idmen) %>% unique()
```

```
length(combined_paris$idmen)
```

```
## [1] 5838
```

```
## Find the household with the most number of family members. Report its idmen.
combined_householdmax <- combined_tbl %>% select(idmen, idind, year) %>% group_by(idmen,year) %>%
                      summarise(householdmember = n()) %>% arrange(desc(householdmember))
```

```
## `summarise()` has grouped output by 'idmen'. You can override using the `.groups` argument.
```

```
combined_householdmax$idmen[1:2]
```

```
## [1] 2.207811e+15 2.510263e+15
```

```
## Number of household present in 2010 and 2011
combined_household2010_2011 <- combined_tbl %>% select(idmen, year) %>% filter(year == 2010 | year == 20
                               select(idmen) %>% unique()

length(combined_household2010_2011$idmen)
```

```
## [1] 13426
```

```
########################################
##      Exercise3 Migration           ##
########################################

##Find out the year each household enters and exit the panel. Report the distribution of the time spent
##in the survey for each household.
#firstly, we group the respondents by their household id, we find out the minimum year (enter year)
#and the maximum year (exit year), then we calculate the time the household spents in the survey
dathh_enter_exit <- dathh_combined %>% as_tibble()
dathh_enter_exit <- dathh_enter_exit %>% group_by(idmen) %>% summarise(enter_year = min(year, na.rm = TR
                                                                       exit_year = max(year, na.rm = TRU
                                                                    mutate(time_spend = exit_year - enter_year +
dathh_enter_exit
```

```
## # A tibble: 41,084 x 4
##      idmen enter_year exit_year time_spend
##      <dbl>      <int>     <int>      <dbl>
##  1 1.20e15       2004      2004          1
##  2 1.20e15       2004      2005          2
##  3 1.20e15       2004      2005          2
##  4 1.20e15       2004      2005          2
##  5 1.20e15       2004      2005          2
##  6 1.20e15       2004      2005          2
##  7 1.20e15       2004      2005          2
##  8 1.20e15       2004      2005          2
##  9 1.20e15       2004      2005          2
## 10 1.20e15       2004      2005          2
## # ... with 41,074 more rows
```

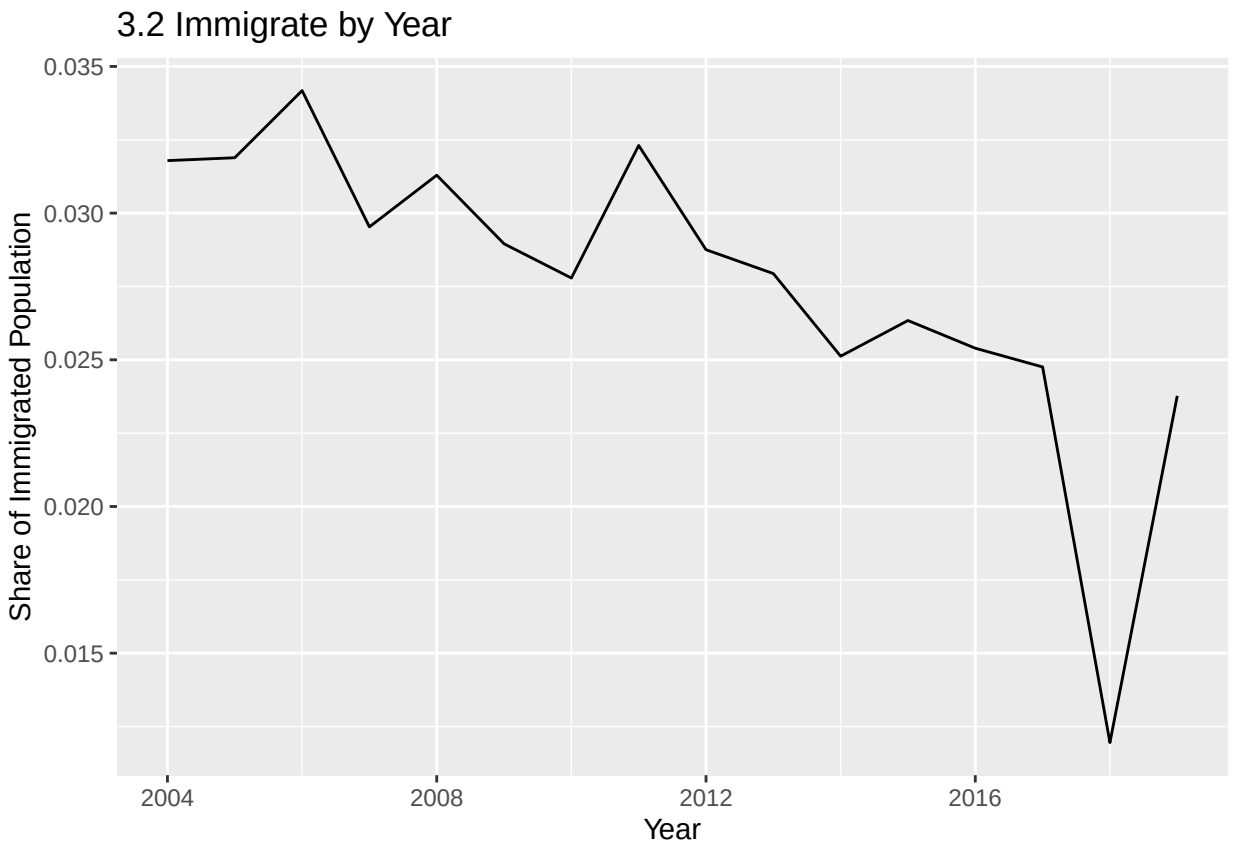
```
##Based on datent, identify whether or not a household moved into its current dwelling at the year of
#survey. Report the first 10 rows of your result and plot the share of individuals in that situation ac
#years.
#datend: year of moving into the dwelling
#I created a dummy variable for immigration: if year == datent, it is 1; if year != datent, it is 0; NA
dathh_immigrated <- dathh_combined %>% mutate(immigrate = ifelse(dathh_combined$year == dathh_combined$
```

```
#report the first 10 rows
slice_head(select(dathh_immigrated, idmen, year, datent, immigrate), n = 10)
```

```
##          idmen year datent immigrate
## 1  1.20001e+15 2004   2000         0
## 2  1.20001e+15 2004   2001         0
## 3  1.20001e+15 2004   2000         0
## 4  1.20001e+15 2004   1957         0
## 5  1.20001e+15 2004   2001         0
## 6  1.20001e+15 2004   1990         0
## 7  1.20001e+15 2004   2000         0
## 8  1.20002e+15 2004   1948         0
## 9  1.20002e+15 2004   1979         0
## 10 1.20002e+15 2004   1984         0
```

```
#plot the share of immigration across years
dathh_immigrated_share <- dathh_immigrated %>% group_by(year) %>%
  summarise(total_ob = n(), n_immigrate = sum(immigrate, na.rm = TRUE)) %>% mutate(share_immigrate = n_

ggplot(dathh_immigrated_share, aes(y=share_immigrate,x=year)) + geom_line() +
  ggtitle("3.2 Immigrate by Year") + xlab("Year") + ylab("Share of Immigrated Population")
```



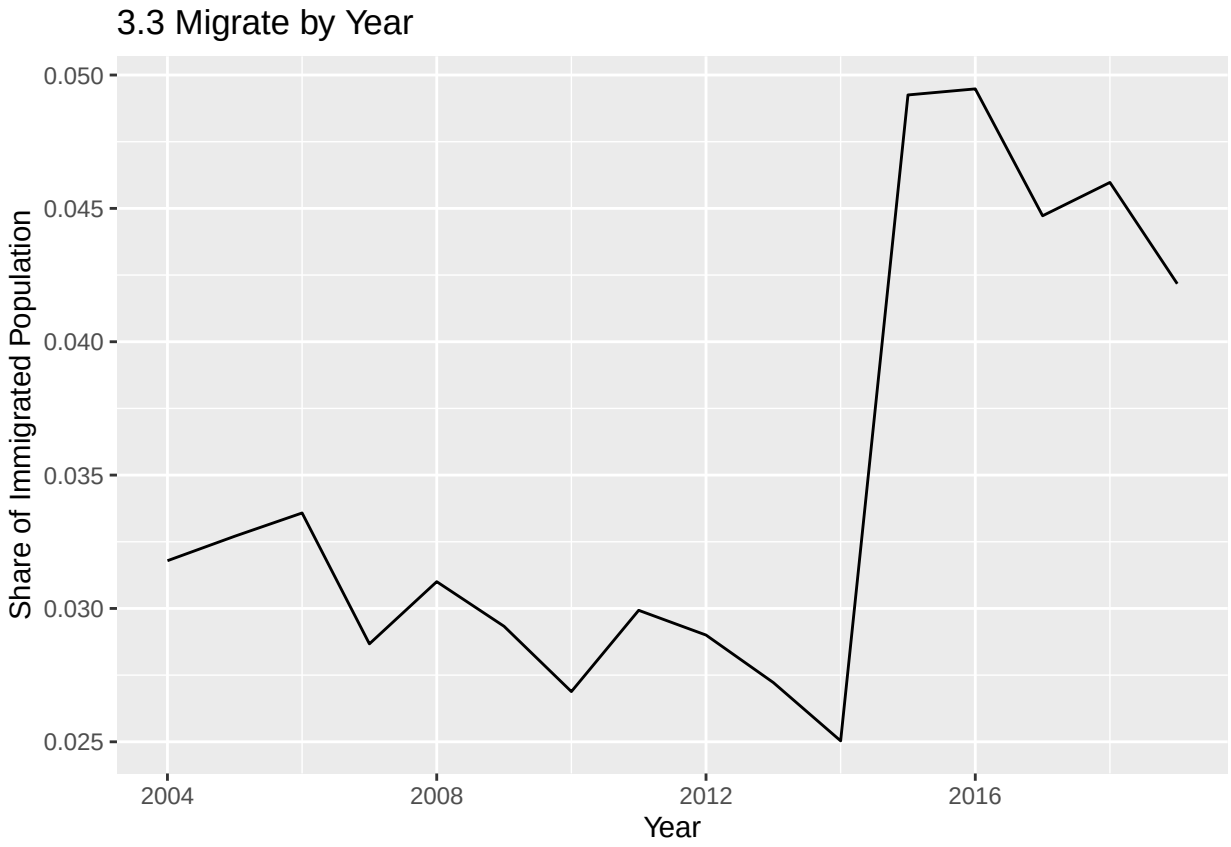

```
##Based on myear and move, identify whether or not household migrated at the year of survey. Report
##the first 10 rows of your result and plot the share of individuals in that situation across years.
#myear: year of last immigration (until 2014)
#move: the household lives at the same address or has moved since last survey

#I create a dummy: if migrate_year = survey year, 1; if move = 2, 1; otherwise, 0
```

```
dathh_migrate <- dathh_combined %>% mutate(migrate_year = ifelse(dathh_combined$myear == dathh_combined$
                                           mutate(migrate_move = ifelse(dathh_combined$move == 2, 1, 0
                                           mutate(migrate = ifelse(dathh_combined$year <= 2014, migrate
#report the first 10 rows
slice_head(select(dathh_migrate, idmen, year, datent, migrate), n = 10)
```

```
##          idmen year datent migrate
## 1  1.20001e+15 2004   2000       0
## 2  1.20001e+15 2004   2001       0
## 3  1.20001e+15 2004   2000       0
## 4  1.20001e+15 2004   1957       0
## 5  1.20001e+15 2004   2001       0
## 6  1.20001e+15 2004   1990       0
## 7  1.20001e+15 2004   2000       0
## 8  1.20002e+15 2004   1948       0
## 9  1.20002e+15 2004   1979       0
## 10 1.20002e+15 2004   1984       0
```

```
#plot the share of migration across year
dathh_migrate_share <- dathh_migrate %>% group_by(year) %>%
  summarise(total_ob = n(), n_migrate = sum(migrate, na.rm = TRUE)) %>% mutate(share_migrate = n_migrat
ggplot(dathh_migrate_share, aes(y=share_migrate,x=year)) + geom_line() + ggtitle("3.3 Migrate by Year")
  xlab("Year") + ylab("Share of Immigrated Population")
```




```
## Mix the two plots you created above in one graph, clearly label the graph. Do you prefer one method
#over the other? Justify.
dathh_migrate_mix <- left_join(dathh_migrate_share, dathh_immigrated_share, by = "year")
```

```
ggplot(data = dathh_migrate_mix) + geom_line(aes(y=share_migrate, x = year, color = "3.3"), size = 1) +
        geom_line(aes(y=share_immigrate,x = year, color = "3.2"), size = 1) +
        scale_color_manual(values = c('3.3'= 'green', '3.2' = 'red')) +   xlab("Year") + ylab("Share")
```


```
#The first one is better because the second one has lots of missing value


##For households who migrate, find out how many households had at least one family member changed
##his/her profession or employment status.
dathh_migrate_profession <- dathh_migrate %>% left_join(datind_combined, by= c("year", "idmen")) %>%
                            select(idmen, profession, empstat) %>% mutate(count = n()) %>%
                            mutate(change = ifelse(count!=1, 1, 0)) %>% filter(change != 1)
length(dathh_migrate_profession$idmen)
```

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

```
########################################
##      Exercise 4 Attrition       ##
########################################

##Compute the attrition across each year, where attrition is defined as
##the reduction in the number of individuals staying in the data panel.
##Report your final result as a table in proportions.

years <- 2004:2018
results <- seq(2005, 2019, by = 1)
attrition <- seq(2005, 2019, by = 1)
total_obs <- seq(2005, 2019, by = 1)
```

```r
n = 1
# in this loop, we use the set diff to find out the exit individuals
for (y in years) {
  y1 <- y
  y2 <- y + 1
  attrition[n] <- length(setdiff(datind_combined[datind_combined$year == y1,]$idind,
                                 datind_combined[datind_combined$year == y2,]$idind))
  total_obs[n] <- length(datind_combined[datind_combined$year == y1,]$idind)
  results[n] <- attrition[n]/total_obs[n]
  n = n + 1
}
table <- data.frame(year = 2005:2019, attrition = attrition,
                    total_observation = total_obs, ratio = results)
table
```

```
##    year attrition total_observation      ratio
## 1  2005      1249             22144 0.05640354
## 2  2006      1999             24241 0.08246359
## 3  2007      1799             24940 0.07213312
## 4  2008      2358             25907 0.09101787
## 5  2009      2150             25510 0.08428067
## 6  2010      1924             25611 0.07512397
## 7  2011      2151             26531 0.08107497
## 8  2012      1943             27071 0.07177422
## 9  2013      3066             28534 0.10745076
## 10 2014      2435             26353 0.09239935
## 11 2015      2531             26787 0.09448613
## 12 2016      2473             26644 0.09281639
## 13 2017      2821             26647 0.10586558
## 14 2018      2697             25402 0.10617274
## 15 2019      2634             24698 0.10664831
```