HW1 Answers

Yuzhe Wang

2022/1/17

```
library(ggplot2)
library(patchwork)
library(bit64)
```

```
##      bit
##
##    'bit'
## The following object is masked from 'package:base':
##
##     xor
## Attaching package bit64
## package:bit64 (c) 2011-2017 Jens Oehlschlaegel
## creators: integer64 runif64 seq :
## coercion: as.integer64 as.vector as.logical as.integer as.double as.character as.bitstring
## logical operator: ! & | xor != == < <= >= >
## arithmetic operator: + - * / %/% %% ^
## math: sign abs sqrt log log2 log10
## math: floor ceiling trunc round
## querying: is.integer64 is.vector [is.atomic} [length] format print str
## values: is.na is.nan is.finite is.infinite
## aggregation: any all min max range sum prod
## cumulation: diff cummin cummax cumsum cumprod
## access: length<- [ [<- [[ [[<-
## combine: c rep cbind rbind as.data.frame
## WARNING don't use as subscripts
## WARNING semantics differ from integer
## for more help type ?bit64
##
##    'bit64'
## The following objects are masked from 'package:base':
##
##     %in%, :, is.double, match, order, rank
```

```
library(data.table)
```

```
##
##     'data.table'
```

```
## The following object is masked from 'package:bit':
##
##     setattr
```

```
setwd("C:\\Users\\wyz_m\\Desktop\\DUKE\\courses\\ECON 613\\assignment\\A1")
getwd()
```

Exercise 1 Basic Statistics

Number of households surveyed in 2007

```
dathh2007 = fread('./data/dathh2007.csv')
length(unique(na.omit(dathh2007$idmen))) # 10498
```

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

Number of households with a marital status "Couple with kids" in 2005

```
dathh2005 = fread('./data/dathh2005.csv')
length(dathh2005$mstatus[dathh2005$mstatus=="Couple, with Kids"]) # 3374
```

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

Number of individuals surveyed in 2008

```
datind2008 = fread('./data/datind2008.csv')
length(unique(na.omit(datind2008$idind))) # 25510
```

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

Number of individuals aged between 25 and 35 in 2016

```
datind2016 = fread('./data/datind2016.csv')
length(which(datind2016$age>=25 & datind2016$age<=35))
```

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

Cross-table gender/profession in 2009

```
datind2009 = fread('./data/datind2009.csv')
table(datind2009$gender, datind2009$profession)
```

```
##
##              0  11  12  13  21  22  23  31  33  34  35  37  38  42  43  44  45
##   Female    11  30   8  29  63  65   8  68  85 184  50 179  78 258 437   1 153
##   Male      19  57  19  78 213 114  48  98 107 142  59 260 368 110 117   2  95
##
##             46  47  48  52  53  54  55  56  62  63  64  65  67  68  69
```

```
##   Female 410  82  22 782  27 584 353 696  64  35  29  19 147 120  40
##   Male   340 429 215 169 182  98 101  74 443 520 246 159 237 177  82
```

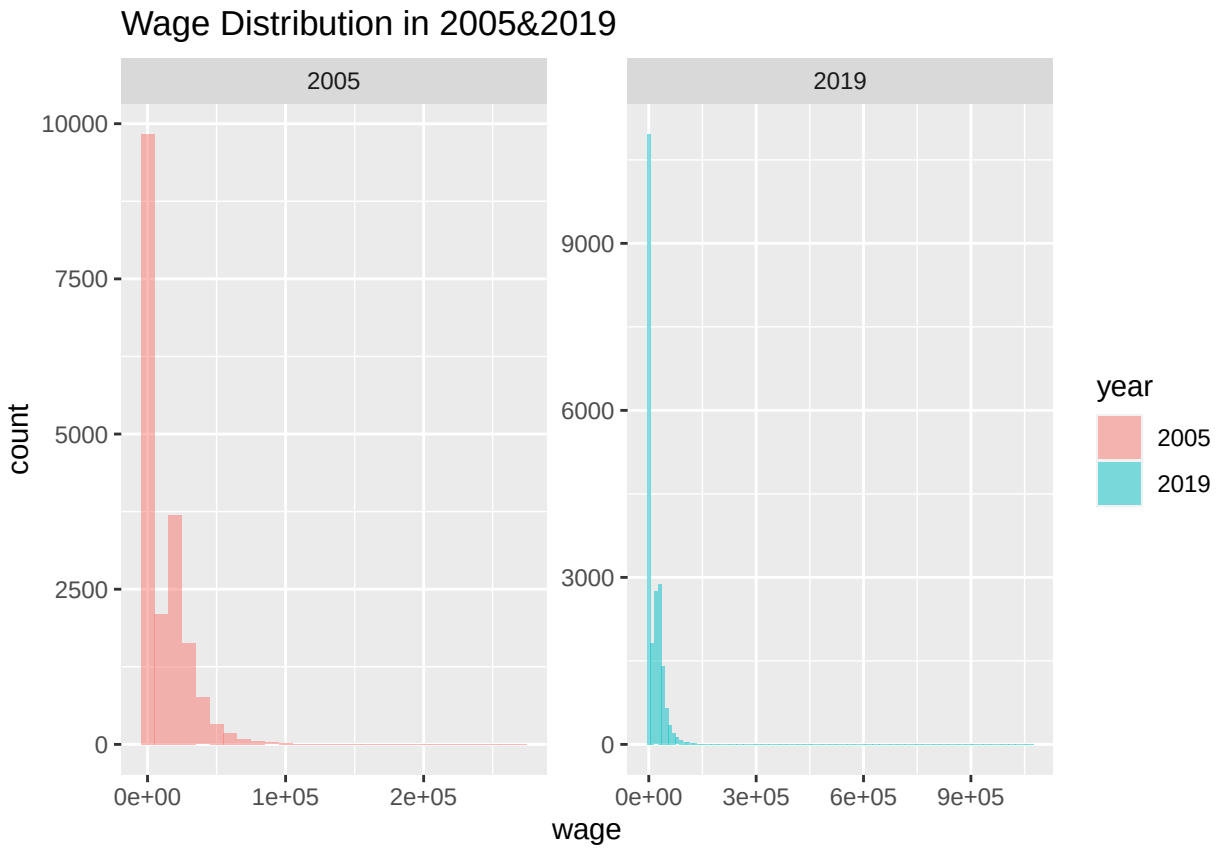
Distribution of wages in 2005 and 2019

If we don't drop wage=0

```
datind2005 = na.omit(fread('./data/datind2005.csv')[,c('idind','wage','year')])
datind2019 = na.omit(fread('./data/datind2019.csv')[,c('idind','wage','year')])
datind2019$idind = as.integer64(datind2019$idind)

dat = rbind(datind2005,datind2019)
dat$year = factor(dat$year)

fun = function(x){
  gini = sum(abs(outer(x,x,"-")))/(2*length(x)^2*mean(x))
  vec=round(c(mean(x),sd(x),quantile(x,0.9)/quantile(x,0.1),gini,max(x)),4)
  names(vec)=c("mean","sd","D9/D1","Gini_coefficient","max")
  return(vec)
  }
by(dat$wage,dat$year,fun)
```

```
## dat$year: 2005
##             mean               sd            D9/D1 Gini_coefficient
##       11992.2575       17318.5602              Inf           0.6672
##              max
##      271962.0000
## ------------------------------------------------------------
## dat$year: 2019
##             mean               sd            D9/D1 Gini_coefficient
##       15350.4739       23207.1850              Inf           0.6655
##              max
##     1068556.0000
```

```
ggplot(data=dat,aes(wage,fill=year))+
  geom_histogram(binwidth=10000,alpha=0.5)+
  facet_wrap(.~year,1,2, scales="free")+
  ggtitle(label='Wage Distribution in 2005&2019')
```

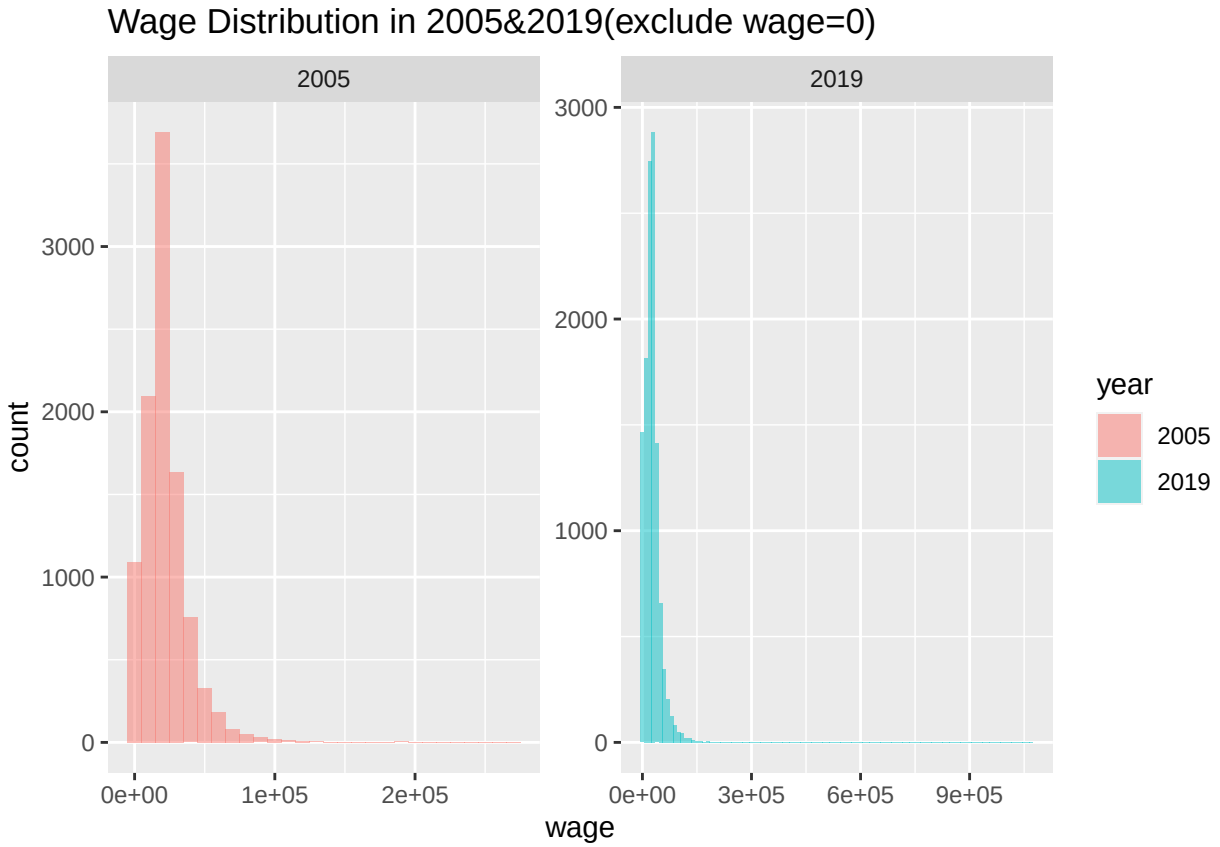


If we drop wage=0

```
dat = dat[dat$wage!=0,]
by(dat$wage,dat$year,fun)
```

```
## dat$year: 2005
##              mean                sd            D9/D1 Gini_coefficient
##       22443.0291       18076.7089           8.8965            0.3771
##               max
##      271962.0000
## ------------------------------------------------------------
## dat$year: 2019
##              mean                sd            D9/D1 Gini_coefficient
##       27578.8393       25107.1872          13.8623            0.3991
##               max
##     1068556.0000
```

```
ggplot(data=dat,aes(wage,fill=year))+
  geom_histogram(binwidth=10000,alpha=0.5)+
  facet_wrap(.~year,1,2, scales="free")+
  ggtitle(label='Wage Distribution in 2005&2019(exclude wage=0)')
```



Distribution of age in 2010

```
# Distribution of age in 2010
datind2010 = fread('./data/datind2010.csv')
datind2010_age = na.omit(datind2010[,c('idind','age','gender')])
summary(datind2010_age)
```

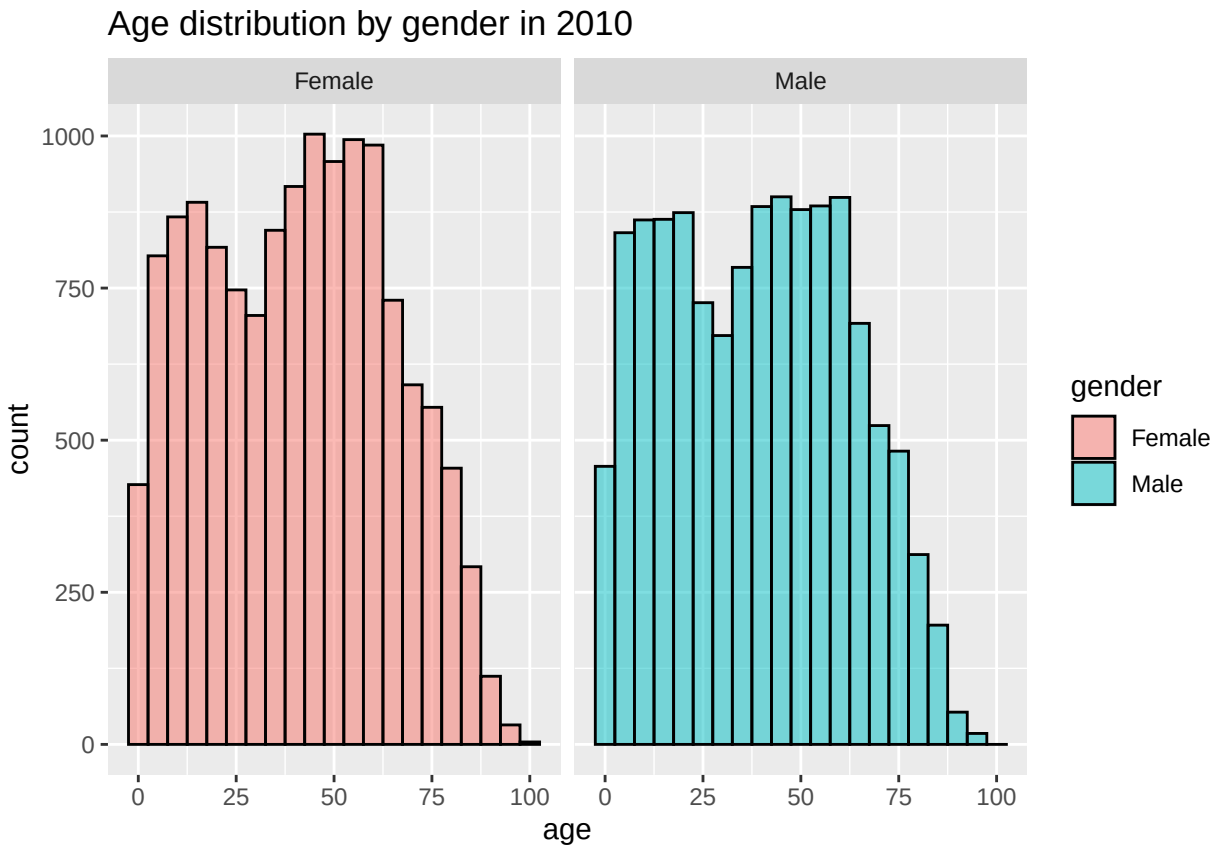
```
##      idind                          age            gender
##  Min.   :1170001001739010001   Min.   :  0.00   Length:26531
##  1st Qu.:1210059706786010001   1st Qu.: 19.00   Class :character
##  Median :1230091301707010003   Median : 40.00   Mode  :character
##  Mean   :1265722996120815890   Mean   : 39.88
##  3rd Qu.:1240920108750010003   3rd Qu.: 58.00
##  Max.   :2241095811330010002   Max.   :102.00
```

```
datind2010_age$gender = factor(datind2010_age$gender)
by(datind2010_age$age,datind2010_age$gender,summary)
```

```
## datind2010_age$gender: Female
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00   20.00   42.00   40.82   59.00  102.00
## ------------------------------------------------------------
## datind2010_age$gender: Male
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00   19.00   39.00   38.87   57.00   96.00
```

```
ggplot(data=datind2010_age,aes(age))+
  geom_histogram(aes(y=..density..),fill="white",color="black",binwidth=5)+
  geom_density(fill="red",color="black",alpha=.2)+
  ggtitle(label='Total age distribution')
```


```
ggplot(data=datind2010_age,aes(age,fill=gender))+
  geom_histogram(binwidth=5,color="black",alpha=0.5)+
  facet_grid(.~gender)+
  ggtitle(label='Age distribution by gender in 2010')
```



Number of individuals in Paris in 2011.

```
datind2011 = fread('./data/datind2011.csv')
dathh2011 = fread('./data/dathh2011.csv')
dat2011 = merge(datind2011,dathh2011,by=c("idmen", "year" ),all=T)
dat2011$idind = factor(dat2011$idind)
sum(by(dat2011$location, dat2011$idind, function(x) "Paris" %in% x)>0)
```

```
## [1] 3514
```

Exercise 2 Merge Datasets

Read all individual datasets from 2004 to 2019. Append all these datasets

```
require(dplyr)
```

```
##      dplyr
##
##    'dplyr'
## The following objects are masked from 'package:data.table':
##
##     between, first, last
## The following objects are masked from 'package:stats':
```

```
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
datind_dirs = paste("./data/datind",2004:2019,sep = "")%>%paste(".csv",sep = "")
datind_total = fread(datind_dirs[1])
for (i in c(2:length(datind_dirs))){
  dat_temp = fread(datind_dirs[i])
  dat_temp$idind = as.integer64(dat_temp$idind)
  datind_total = rbind(datind_total, dat_temp)
}
datind_total = datind_total[,-1]
```

Read all household datasets from 2004 to 2019. Append all these datasets

```
dathh_dirs = paste("./data/dathh",2004:2019,sep = "")%>%paste(".csv",sep = "")
dathh_total = fread(dathh_dirs[1])
for (i in c(2:length(dathh_dirs))){
  dat_temp = fread(dathh_dirs[i])
  dathh_total = rbind(dathh_total, dat_temp)
}
dathh_total = dathh_total[,-1]
```

List the variables that are simultaneously present in the individual and household datasets

```
same_var = c()
for (datind_name in colnames(datind_total)){
  for (dathh_name in colnames(dathh_total)){
    if (datind_name == dathh_name){
      same_var = append(same_var,datind_name)
    }
  }
}
same_var
```

```
## [1] "idmen" "year"
```

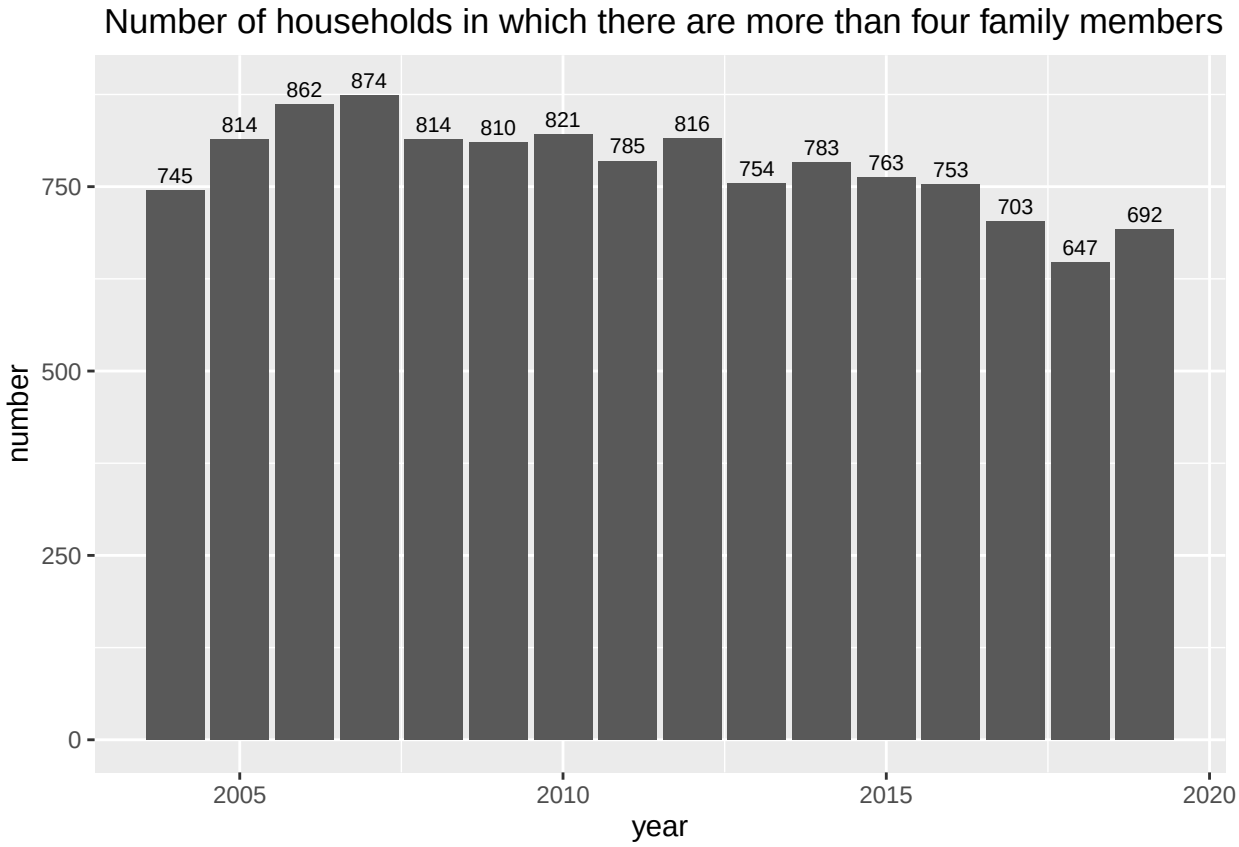
Merge the appended individual and household datasets

```
# Drop the cases that are totally the same(including year, idind, idmen, and...)
# If individul's idmen is not in all dathh, we still keep it,
# but with NAs in all dathh characteristics
dat_total = data.frame(distinct(merge(datind_total, dathh_total,
                                      by = same_var, all = T),.keep_all=T))
```

Number of households in which there are more than four family members

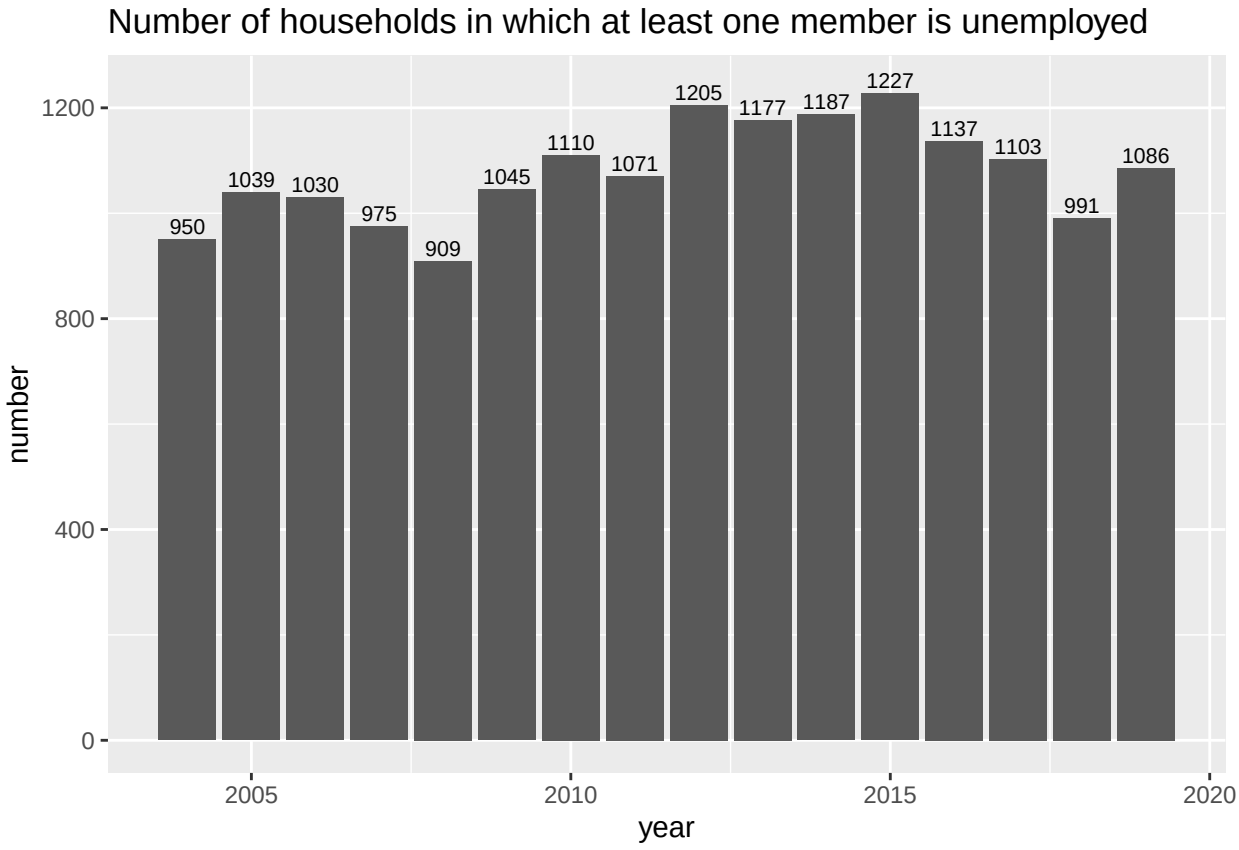
```
dat_total$idmen = factor(dat_total$idmen)
dat_total$year = factor(dat_total$year)
```

```
member = by(dat_total$idind,dat_total[c('idmen','year')],function(x) length(unique(x)))
member[is.na(member)] = 0

# find the number of household once number members of which > 4 in each year.
data.frame(year = 2004:2019, number = apply(member, MARGIN = 2,
                                            function(x) sum(x>4))) %>%
  ggplot(aes(year,number))+
  geom_bar(stat = 'identity')+
  geom_text(aes(label=number, y=number+10),
            position="dodge", vjust=0,size = 8/.pt) +
  ggtitle(label=' Number of households in which there are more than four family members')
```



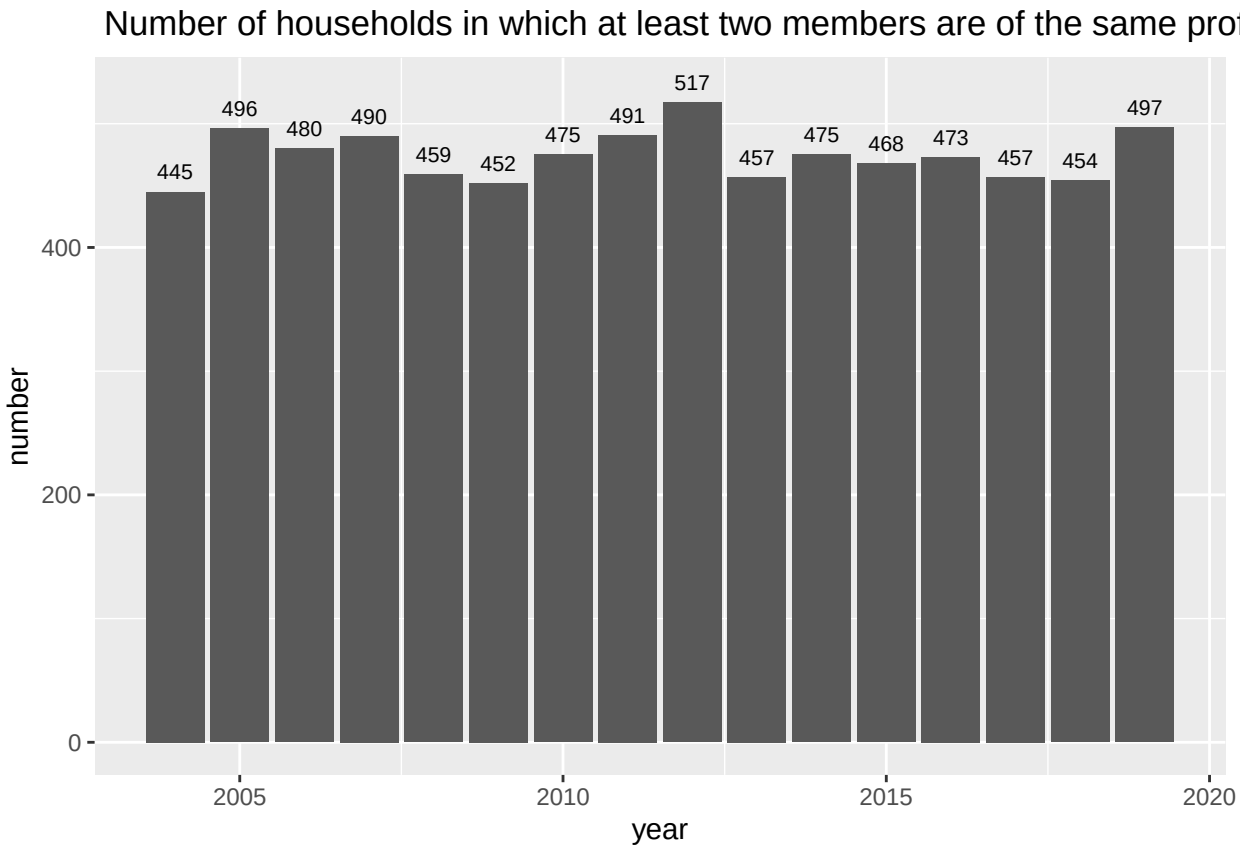
Number of households in which at least one member is unemployed

```
# number of household which at least one member is unemployed in each year.
emp = by(dat_total$empstat,dat_total[c('idmen','year')],
         function(x)'Unemployed' %in% x)
emp[is.na(emp)] = 0
data.frame(year = 2004:2019, number = apply(emp,2,sum)) %>%
  ggplot(aes(year,number))+
  geom_bar(stat = 'identity')+
  geom_text(aes(label=number, y=number+10),
            position="dodge", vjust=0,size = 8/.pt) +
  ggtitle(label='Number of households in which at least one member is unemployed')
```



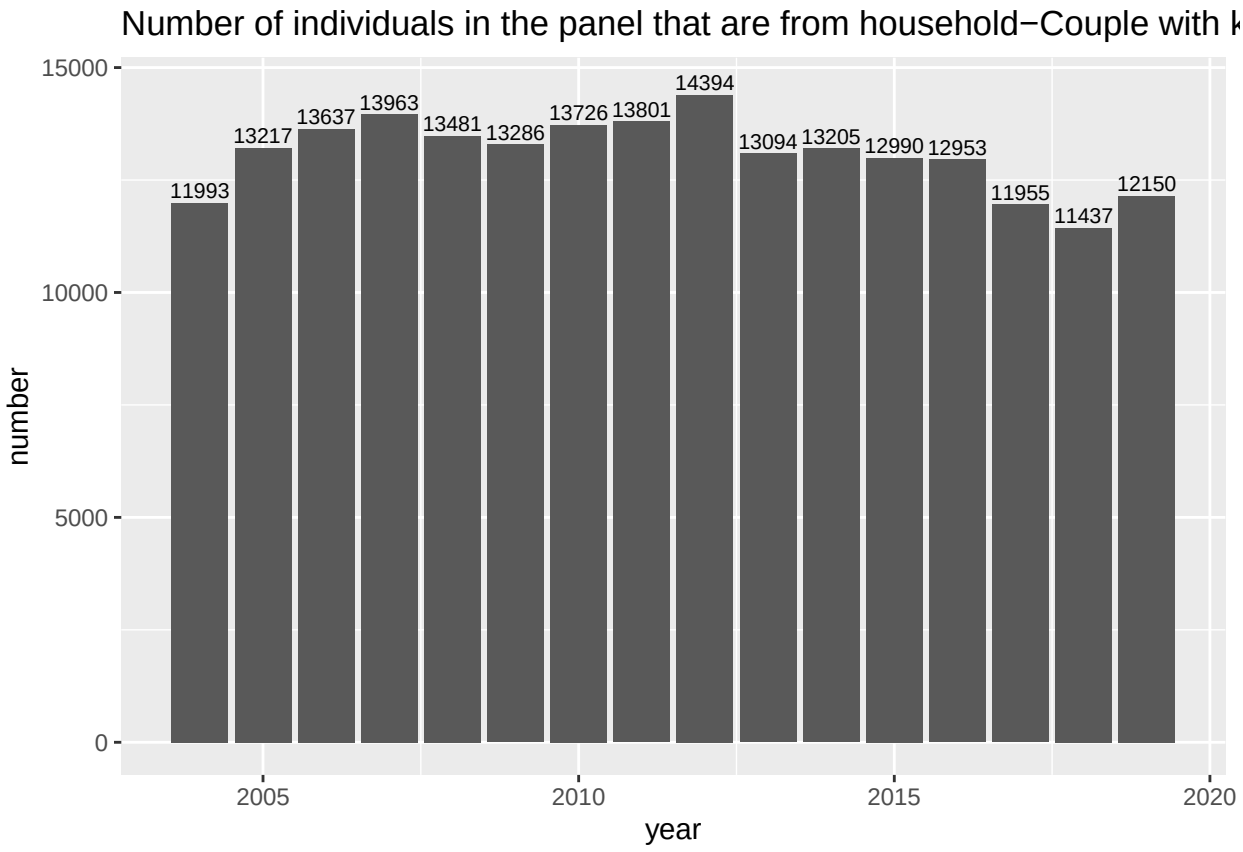
Number of households in which at least two members are of the same profession

```
prof = dat_total[!is.na(dat_total$profession) & dat_total$profession != "",]
prof = by(prof$profession,prof[c('idmen','year')],
          function(x) length(x)-length(unique(x)))>0
prof[is.na(prof)] = 0

# number of household which at least two members
# are of the same profession in each year.
data.frame(year = 2004:2019, number = apply(prof,2,function(x) sum(x>0)))%>%
   ggplot(aes(year,number))+
   geom_bar(stat = 'identity')+
   geom_text(aes(label=number, y=number+10),
             position="dodge", vjust=0,size = 8/.pt) +
   ggtitle(label=' Number of households in which at least two members are of the same profession')
```



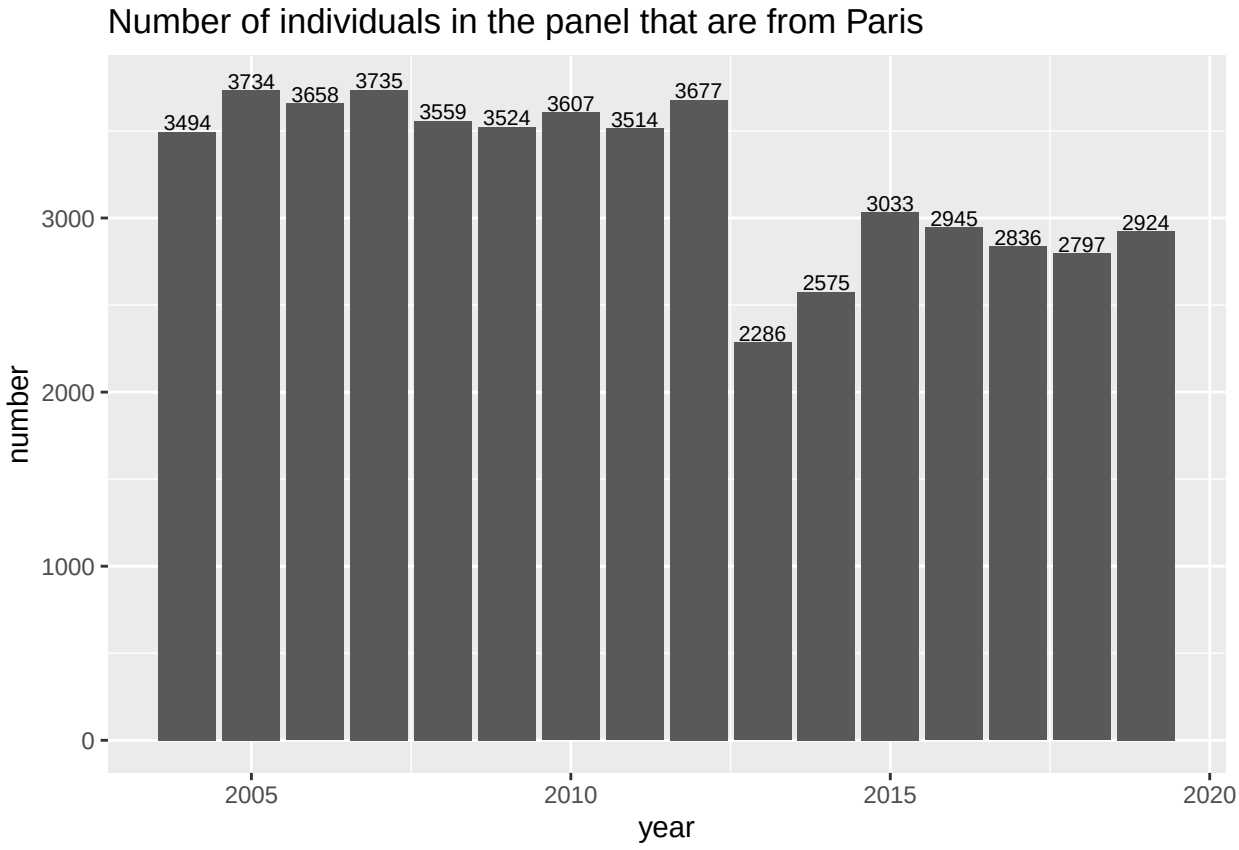
Number of individuals in the panel that are from household-Couple with kids

```
dat_total$idind = factor(dat_total$idind)

# find the individuals once who are from household-Couple with kids in each year.
mstatus_ind = by(dat_total$mstatus,dat_total[c("idind","year")],
                 function(x) "Couple, with Kids" %in% x)
mstatus_ind[is.na(mstatus_ind)] = 0
data.frame(year = 2004:2019, number = apply(mstatus_ind,2,sum))%>%
  ggplot(aes(year,number))+
  geom_bar(stat = 'identity')+
  geom_text(aes(label=number, y=number+100),
            position="dodge", vjust=0,size = 8/.pt) +
  ggtitle(label='Number of individuals in the panel that are from household-Couple with kids')
```



Number of individuals in the panel that are from Paris

```
# number of individuals that are from Paris in each year.
location_ind = by(dat_total$location,dat_total[c("idind","year")],
                  function(x) "Paris" %in% x)
location_ind[is.na(location_ind)] = 0
data.frame(year = 2004:2019, number = apply(location_ind,2,sum))%>%
  ggplot(aes(year,number))+
  geom_bar(stat = 'identity')+
  geom_text(aes(label=number, y=number+10),
            position="dodge", vjust=0,size = 8/.pt) +
  ggtitle(label='Number of individuals in the panel that are from Paris')
```



Find the household with the most number of family members. Report its idmen

```
# Find the household with the most number of family members in each year.
max_member = apply(member,2,max)
max_member_list = list()
for (i in 1:length(colnames(member))){
  for (j in 1:length(rownames(member))){
    if (member[j,i] == max_member[i]){
      if (i-1==length(max_member_list)){
        max_member_list[[i]] = paste(rownames(member)[j],":",
                                     max_member[i],"family members")
      }
      else{
        max_member_list[[i]] = c(max_member_list[[i]],
                                 paste(rownames(member)[j],
                                       ":",max_member[i],"family members"))
      }
    }
  }
}
names(max_member_list) = 2004:2019
max_member_list
```

```
## $`2004`
## [1] "1208045118450100 : 10 family members"
## [2] "1607839058220100 : 10 family members"
```

```
## [3] "1610263040580100 : 10 family members"
## [4] "1804363114960100 : 10 family members"
##
## $`2005`
## [1] "1607839058220100 : 11 family members"
##
## $`2006`
## [1] "1607839058220100 : 10 family members"
## [2] "1811109095380100 : 10 family members"
##
## $`2007`
## [1] "2207811124040100 : 14 family members"
##
## $`2008`
## [1] "1700707001000100 : 10 family members"
## [2] "1811109095380100 : 10 family members"
## [3] "2006865025180100 : 10 family members"
##
## $`2009`
## [1] "1700707001000100 : 11 family members"
##
## $`2010`
## [1] "2510263102990100 : 14 family members"
##
## $`2011`
## [1] "1905191114960100 : 10 family members"
## [2] "2202243098040100 : 10 family members"
##
## $`2012`
## [1] "1905191114960100 : 10 family members"
## [2] "2202243098040100 : 10 family members"
##
## $`2013`
## [1] "2202243098040100 : 10 family members"
##
## $`2014`
## [1] "2106457101960100 : 9 family members" "2200896118640100 : 9 family members"
## [3] "2209201025180100 : 9 family members" "2701042078730100 : 9 family members"
## [5] "2707811117610100 : 9 family members" "2710263020060100 : 9 family members"
## [7] "2905191059550100 : 9 family members" "2905459051770100 : 9 family members"
##
## $`2015`
## [1] "3000896115750100 : 12 family members"
##
## $`2016`
## [1] "3000896115750100 : 12 family members"
##
## $`2017`
## [1] "3000896115750100 : 12 family members"
##
## $`2018`
## [1] "3000896115750100 : 11 family members"
##
## $`2019`
```

```
## [1] "2806477001000100 : 9 family members" "3200528124040100 : 9 family members"
## [3] "3300896124060100 : 9 family members" "3402178051020100 : 9 family members"
```

Number of households present in 2010 and 2011.

```
dathh2010 = fread('./data/dathh2010.csv')
dathh2011 = fread('./data/dathh2011.csv')
length(intersect(na.omit(dathh2010$idmen), na.omit(dathh2011$idmen)))
```

```
## [1] 8984
```

Exercise3 Migration

Find out the year each household enters and exit the panel. Report the length of years each household stay in the panel

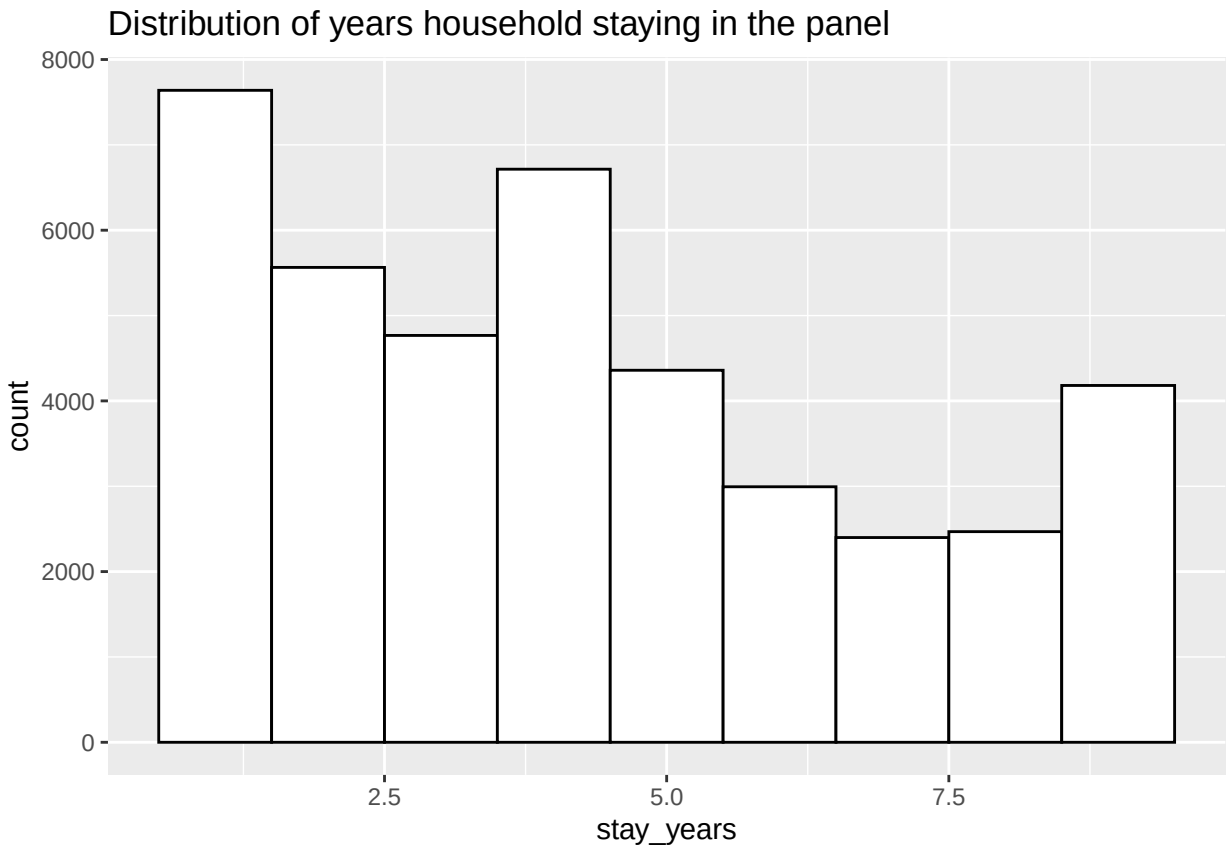
```
dat_total$year = as.numeric(as.character(dat_total$year))
# the year each household enters the panel (report first 10 houselod)
by(dat_total$year,dat_total$idmen,min)[1:10]
```

```
## dat_total$idmen
## 1200010012930100 1200010040580100 1200010066630100 1200010082450100
##             2004             2004             2004             2004
## 1200010086440100 1200010102990100 1200010118450100 1200020012930100
##             2004             2004             2004             2004
## 1200020017390100 1200020026420100
##             2004             2004
```

```
# the year each household exits the panel (report first 10 houselod)
by(dat_total$year,dat_total$idmen,max)[1:10]
```

```
## dat_total$idmen
## 1200010012930100 1200010040580100 1200010066630100 1200010082450100
##             2004             2005             2005             2005
## 1200010086440100 1200010102990100 1200010118450100 1200020012930100
##             2005             2005             2005             2005
## 1200020017390100 1200020026420100
##             2005             2005
```

```
#length of years each household stay in the panel
stay_years = by(dat_total$year,dat_total$idmen,function(x) length(unique(x)))
data.frame(stay_years = stay_years[1:length(stay_years)]) %>%
  ggplot(aes(stay_years))+
  geom_histogram(fill="white",color="black",binwidth=1)+
  ggtitle(label='Distribution of years household staying in the panel')
```



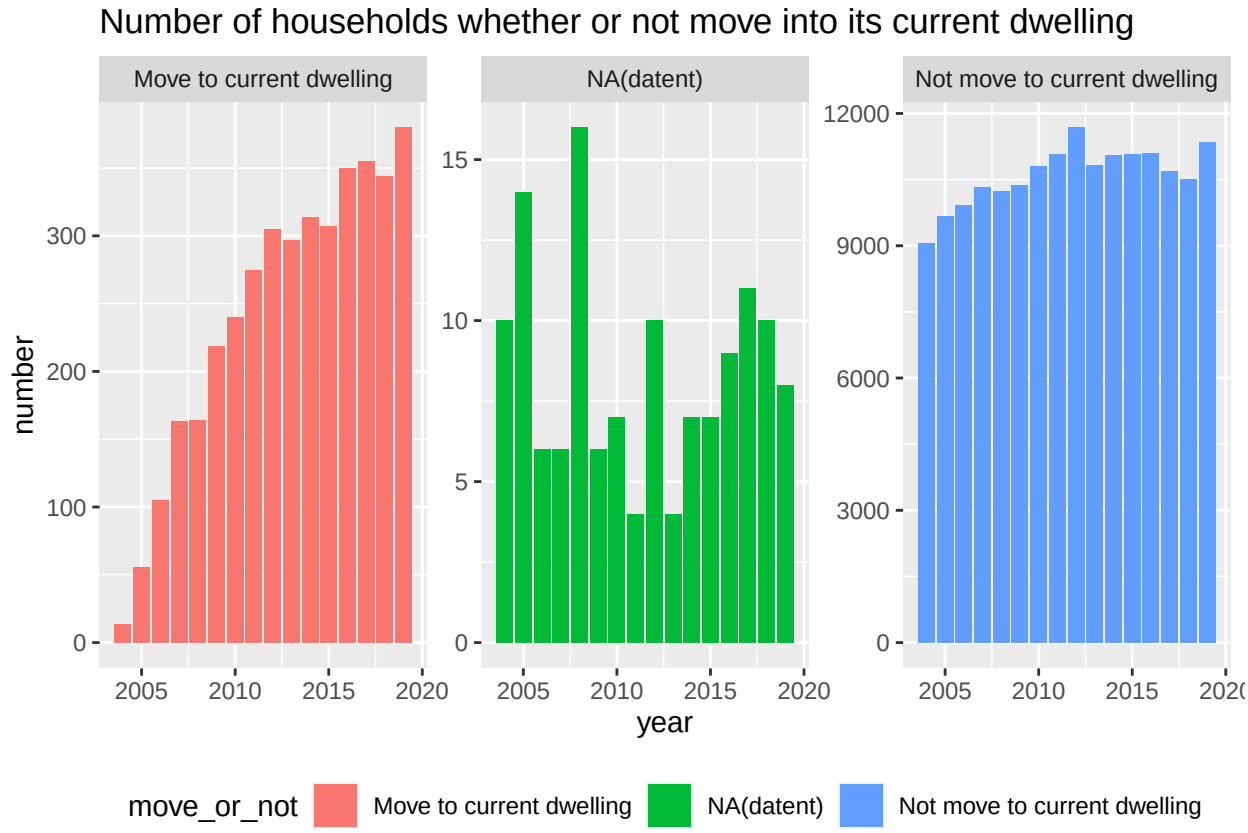
Base on datent, identify whether or not household moved into its current dwelling at the year of survey

```
dat_total$year = factor(dat_total$year)
dat_total$idind = as.integer64(as.character(dat_total$idind))
year_matrix = matrix(rep(2004:2019,length(unique(dat_total$idmen))),
                     length(unique(dat_total$idmen)),length(unique(dat_total$year)))
datent = by(dat_total$datent,dat_total[c('idmen','year')],unique) == year_matrix
# matrix shows that whether or not household moved into its current dwelling at the year of survey
datent[1:10,]
```

```
##                   year
## idmen               2004  2005 2006 2007 2008 2009 2010 2011 2012 2013 2014
##   1200010012930100 FALSE    NA   NA   NA   NA   NA   NA   NA   NA   NA   NA
##   1200010040580100 FALSE FALSE   NA   NA   NA   NA   NA   NA   NA   NA   NA
##   1200010066630100 FALSE FALSE   NA   NA   NA   NA   NA   NA   NA   NA   NA
##   1200010082450100 FALSE FALSE   NA   NA   NA   NA   NA   NA   NA   NA   NA
##   1200010086440100 FALSE FALSE   NA   NA   NA   NA   NA   NA   NA   NA   NA
##   1200010102990100 FALSE FALSE   NA   NA   NA   NA   NA   NA   NA   NA   NA
##   1200010118450100 FALSE FALSE   NA   NA   NA   NA   NA   NA   NA   NA   NA
##   1200020012930100 FALSE FALSE   NA   NA   NA   NA   NA   NA   NA   NA   NA
##   1200020017390100 FALSE FALSE   NA   NA   NA   NA   NA   NA   NA   NA   NA
##   1200020026420100 FALSE FALSE   NA   NA   NA   NA   NA   NA   NA   NA   NA
##                   year
## idmen               2015 2016 2017 2018 2019
```

```
##   1200010012930100   NA   NA   NA   NA   NA
##   1200010040580100   NA   NA   NA   NA   NA
##   1200010066630100   NA   NA   NA   NA   NA
##   1200010082450100   NA   NA   NA   NA   NA
##   1200010086440100   NA   NA   NA   NA   NA
##   1200010102990100   NA   NA   NA   NA   NA
##   1200010118450100   NA   NA   NA   NA   NA
##   1200020012930100   NA   NA   NA   NA   NA
##   1200020017390100   NA   NA   NA   NA   NA
##   1200020026420100   NA   NA   NA   NA   NA
```

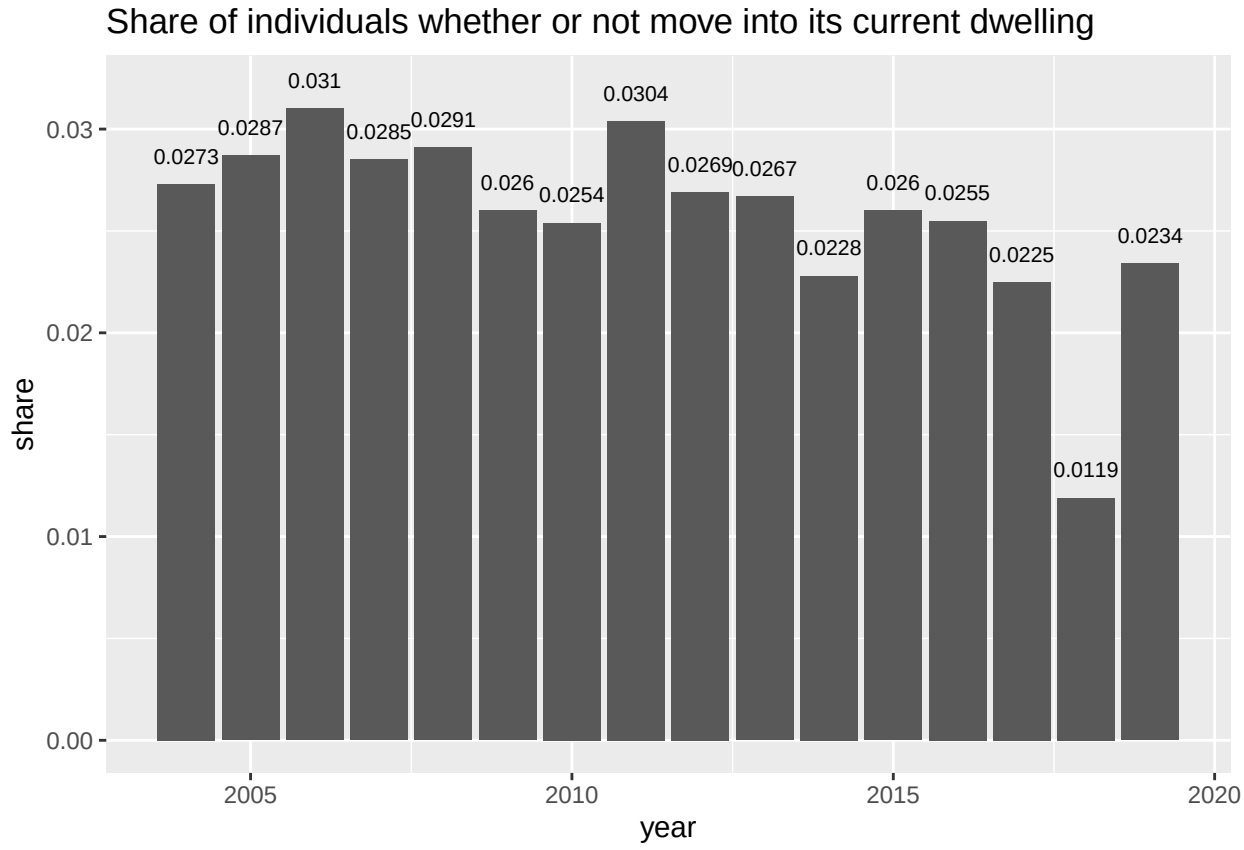
```
datent[is.na(datent)] = 0

hh_surveyed_in_years =  by(dat_total$idmen,dat_total$year,
                           function(x) length(unique(na.omit(x))))[1:16]
hh_datent_na_in_years =  by(dat_total$datent,dat_total[c('idmen','year')],
                           function(x) is.na(unique(x))) == TRUE
hh_datent_na_in_years[is.na(hh_datent_na_in_years)] = 0
hh_datent_na_in_years = apply(hh_datent_na_in_years, 2, sum)

hh_numbers_move_dw = data.frame(year = 2004:2019, number = apply(datent,2,sum),
                                move_or_not = rep("Move to current dwelling",16))
hh_numbers_not_move_dw = data.frame(year = 2004:2019,
                                    number = hh_surveyed_in_years -
                                      hh_datent_na_in_years - apply(datent,2,sum),
                                    move_or_not = rep("Not move to current dwelling",16))
hh_numbers_na_dw = data.frame(year = 2004:2019, number = hh_datent_na_in_years,
                              move_or_not = rep("NA(datent)",16))

ind_move_dw = c()
ind_unique = c()
for (y in 2004:2019){
  ind_move = length(na.omit(unique(dat_total[dat_total$datent==y &
                                               dat_total$year == y,'idind'])))
  ind_move_dw = c(ind_move_dw, ind_move)
}
for (y in 2004:2019){
  ind = length(na.omit(unique(dat_total[dat_total$year == y,'idind'])))
  ind_unique = c(ind_unique, ind)
}
ind_share_move_dw = data.frame(year = 2004:2019,
                               share = round(ind_move_dw / ind_unique,4))

datent_hist = rbind(hh_numbers_move_dw, hh_numbers_not_move_dw) %>%
  rbind(hh_numbers_na_dw)
move_dw_plot =
  ggplot(data=datent_hist,aes(x=year, y=number, fill=move_or_not))+
  geom_bar(stat = 'identity')+
  facet_wrap(.~move_or_not,1,3,scales="free_y")+
  ggtitle(label='Number of households whether or not move into its current dwelling')+
  theme(legend.position = "bottom")
move_dw_plot
```




```
ind_share_move_dw
```

```
##    year  share
## 1  2004 0.0273
## 2  2005 0.0287
## 3  2006 0.0310
## 4  2007 0.0285
## 5  2008 0.0291
## 6  2009 0.0260
## 7  2010 0.0254
## 8  2011 0.0304
## 9  2012 0.0269
## 10 2013 0.0267
## 11 2014 0.0228
## 12 2015 0.0260
## 13 2016 0.0255
## 14 2017 0.0225
## 15 2018 0.0119
## 16 2019 0.0234
```

```
move_dw_share_plot =
  ggplot(data=ind_share_move_dw,aes(x=year, y=share))+
  geom_bar(stat = 'identity')+
  geom_text(aes(label=share, y=share+0.001), position="dodge", vjust=0,size = 8/.pt) +
  ggtitle(label='Share of individuals whether or not move into its current dwelling')
move_dw_share_plot
```



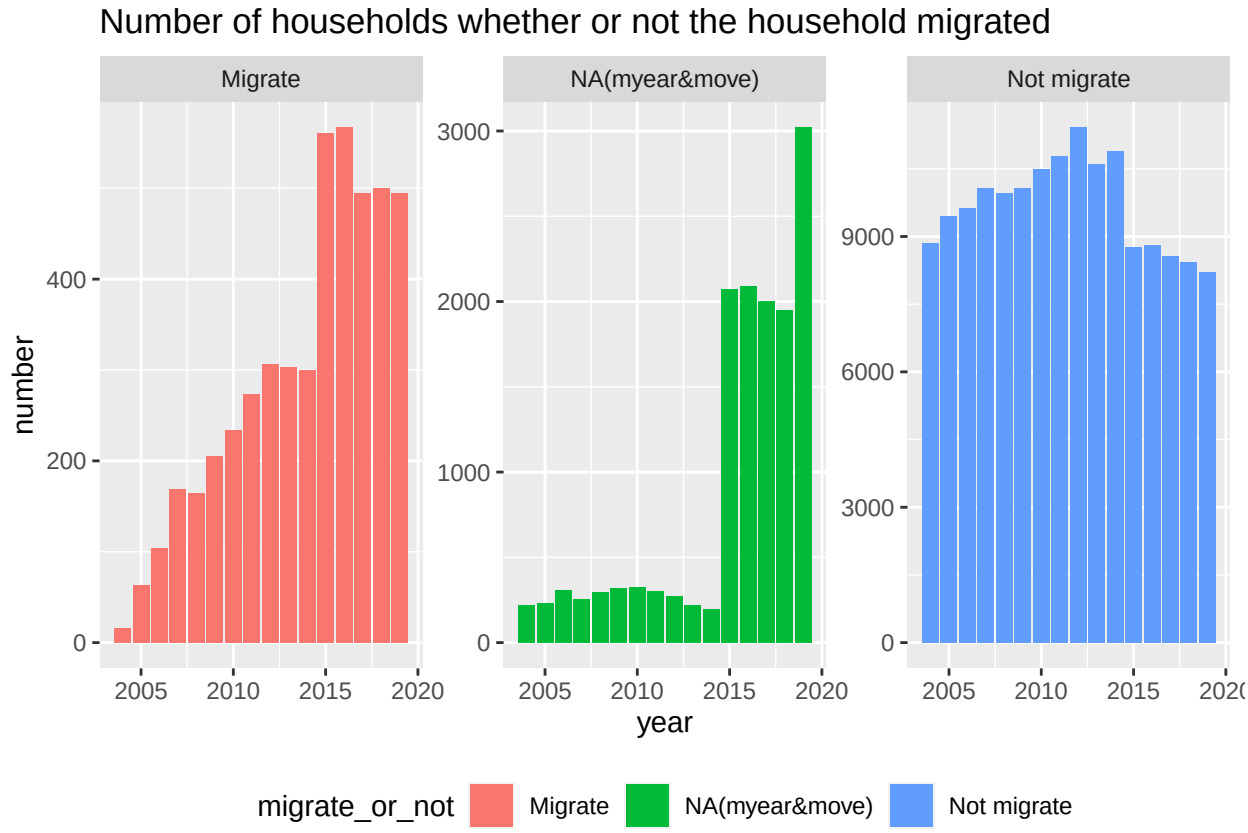
Base on myear and move, identify whether or not household migrated at the year of survey.

```
migrate1 = by(dat_total$myear,dat_total[c('idmen','year')],
              function(x) return(unique(x))) == year_matrix
migrate2 = by(dat_total$move,dat_total[c('idmen','year')],
               function(x) return(unique(x))) == 2
migrate1[is.na(migrate1)] = 0
migrate2[is.na(migrate2)] = 0
migrate = migrate1 + migrate2 > 0
migrate[1:10,]
```

```
##                   year
## idmen               2004  2005  2006  2007  2008  2009  2010  2011  2012  2013
##   1200010012930100 FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##   1200010040580100 FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##   1200010066630100 FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##   1200010082450100 FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##   1200010086440100 FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##   1200010102990100 FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##   1200010118450100 FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##   1200020012930100 FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##   1200020017390100 FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##   1200020026420100 FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##                   year
```

```
## idmen              2014  2015  2016  2017  2018  2019
##   1200010012930100 FALSE FALSE FALSE FALSE FALSE FALSE
##   1200010040580100 FALSE FALSE FALSE FALSE FALSE FALSE
##   1200010066630100 FALSE FALSE FALSE FALSE FALSE FALSE
##   1200010082450100 FALSE FALSE FALSE FALSE FALSE FALSE
##   1200010086440100 FALSE FALSE FALSE FALSE FALSE FALSE
##   1200010102990100 FALSE FALSE FALSE FALSE FALSE FALSE
##   1200010118450100 FALSE FALSE FALSE FALSE FALSE FALSE
##   1200020012930100 FALSE FALSE FALSE FALSE FALSE FALSE
##   1200020017390100 FALSE FALSE FALSE FALSE FALSE FALSE
##   1200020026420100 FALSE FALSE FALSE FALSE FALSE FALSE
```

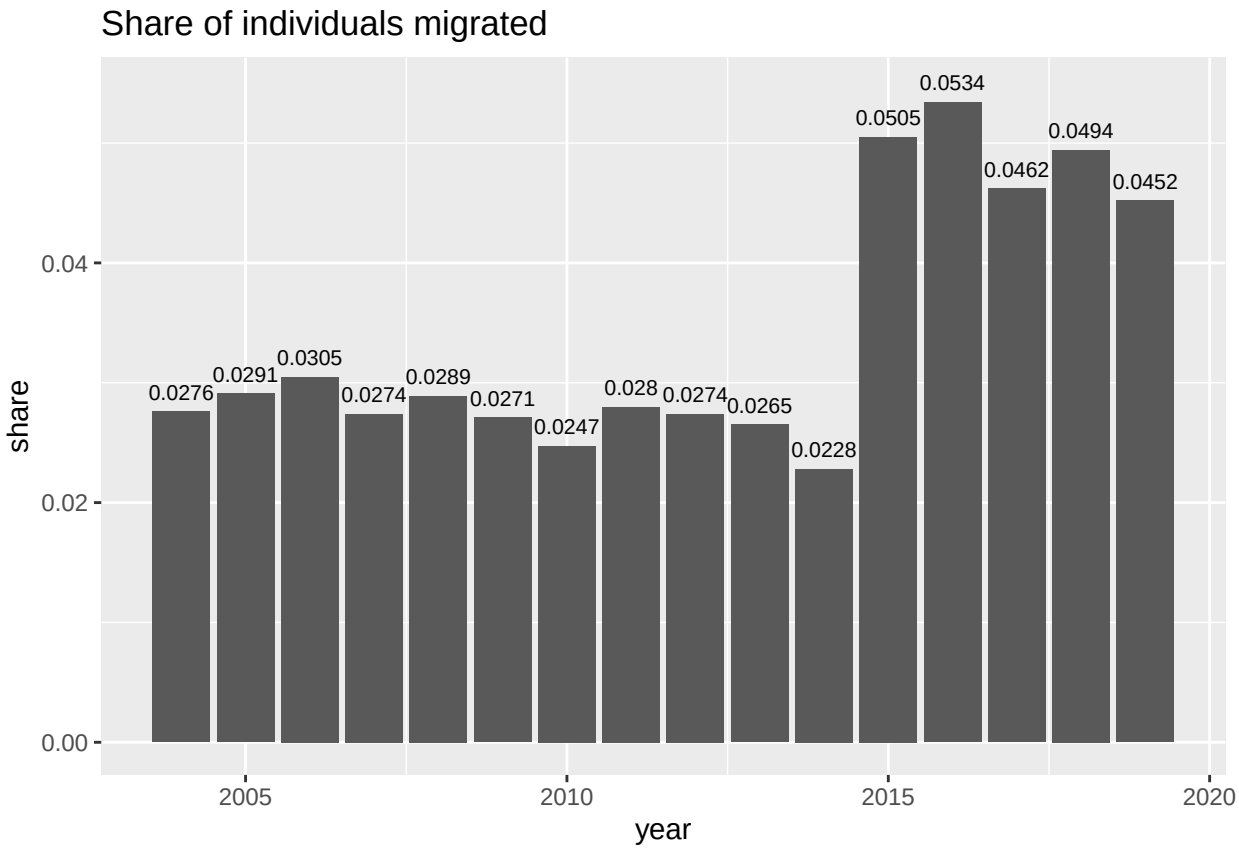
```
hh_myear_na_in_years = by(dat_total$myear,dat_total[c('idmen','year')],
                          function(x) is.na(unique(x))) == TRUE
hh_myear_na_in_years[is.na(hh_myear_na_in_years)] = 0

hh_move_na_in_years = by(dat_total$move,dat_total[c('idmen','year')],
                         function(x) is.na(unique(x))) == TRUE
hh_move_na_in_years[is.na(hh_move_na_in_years)] = 0

hh_migrate_na_in_years = apply(hh_myear_na_in_years + hh_move_na_in_years == 2,2,sum)


hh_numbers_migrate = data.frame(year = 2004:2019, number = apply(migrate, 2, sum),
                                migrate_or_not = rep("Migrate",16))
hh_numbers_not_migrate = data.frame(year = 2004:2019,
                                    number = hh_surveyed_in_years -
                                      hh_migrate_na_in_years - apply(migrate, 2, sum),
                                    migrate_or_not = rep("Not migrate",16))
hh_numbers_na_migrate = data.frame(year = 2004:2019,
                                   number = hh_migrate_na_in_years,
                                   migrate_or_not = rep("NA(myear&move)",16))
ind_migrate_dw = c()
for (y in 2004:2019){
  ind_migrate = length(na.omit(unique(dat_total[(dat_total$myear == y
                                                 | dat_total$move == 2)
                                                 & dat_total$year == y,'idind'])))
  ind_migrate_dw = c(ind_migrate_dw, ind_migrate)
}
ind_share_migrate_dw = data.frame(year = 2004:2019,
                                  share = round(ind_migrate_dw / ind_unique,4))

migrate_hist = rbind(hh_numbers_migrate, hh_numbers_not_migrate) %>%
  rbind(hh_numbers_na_migrate)
migrate_plot =
  ggplot(data=migrate_hist,aes(x=year, y=number, fill=migrate_or_not))+
  geom_bar(stat = 'identity')+
  facet_wrap(.~migrate_or_not,1,3,scales="free_y")+
  ggtitle(label='Number of households whether or not the household migrated')+
  theme(legend.position = "bottom")
migrate_plot
```




```
ind_share_migrate_dw
```

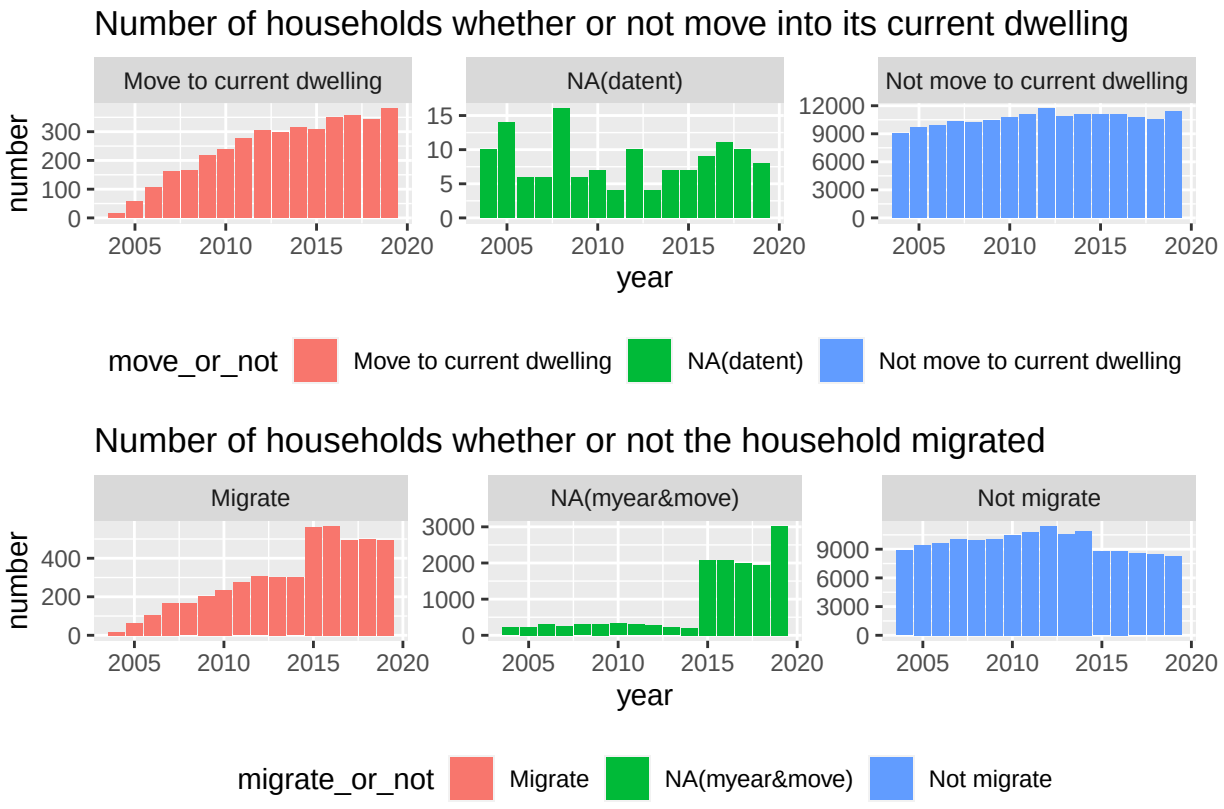
```
##    year  share
## 1  2004 0.0276
## 2  2005 0.0291
## 3  2006 0.0305
## 4  2007 0.0274
## 5  2008 0.0289
## 6  2009 0.0271
## 7  2010 0.0247
## 8  2011 0.0280
## 9  2012 0.0274
## 10 2013 0.0265
## 11 2014 0.0228
## 12 2015 0.0505
## 13 2016 0.0534
## 14 2017 0.0462
## 15 2018 0.0494
## 16 2019 0.0452
```

```
migrate_share_plot =
  ggplot(data=ind_share_migrate_dw,aes(x=year, y=share))+
  geom_bar(stat = 'identity')+
  geom_text(aes(label=share, y=share+0.001), position="dodge", vjust=0,size = 8/.pt) +
  ggtitle(label='Share of individuals migrated')
migrate_share_plot
```

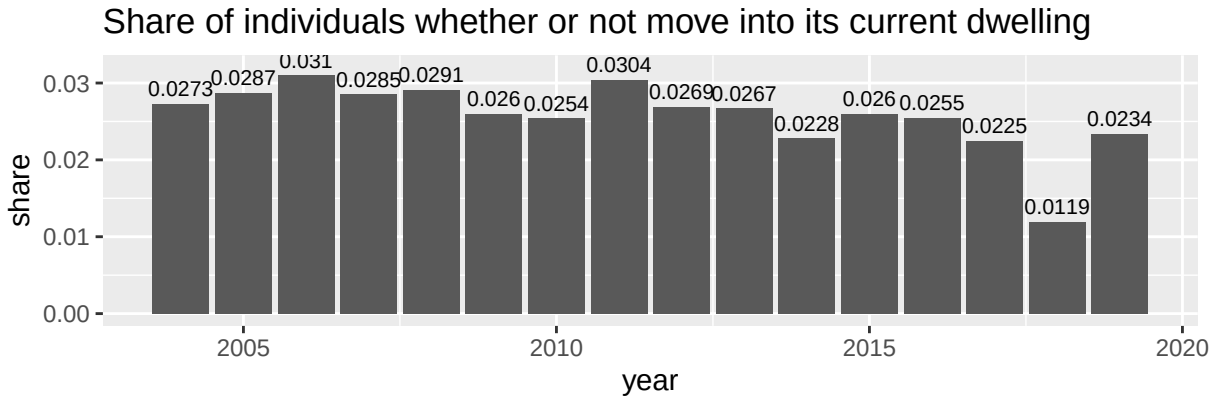


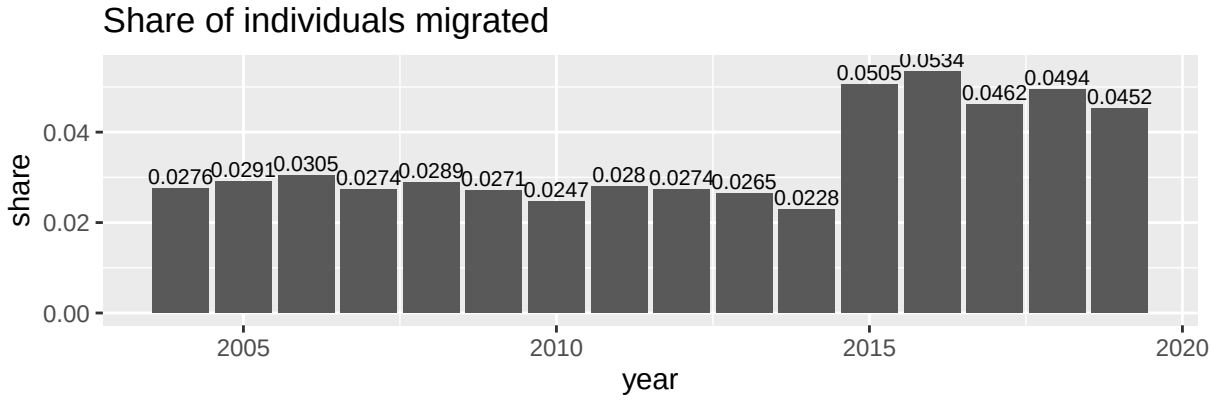
Mix the two plots you created above in one graph, clearly label the graph.

```
move_dw_plot+migrate_plot+plot_layout(ncol=1,nrow=2)
```




```
move_dw_share_plot+migrate_share_plot+plot_layout(ncol=1,nrow=2)
```



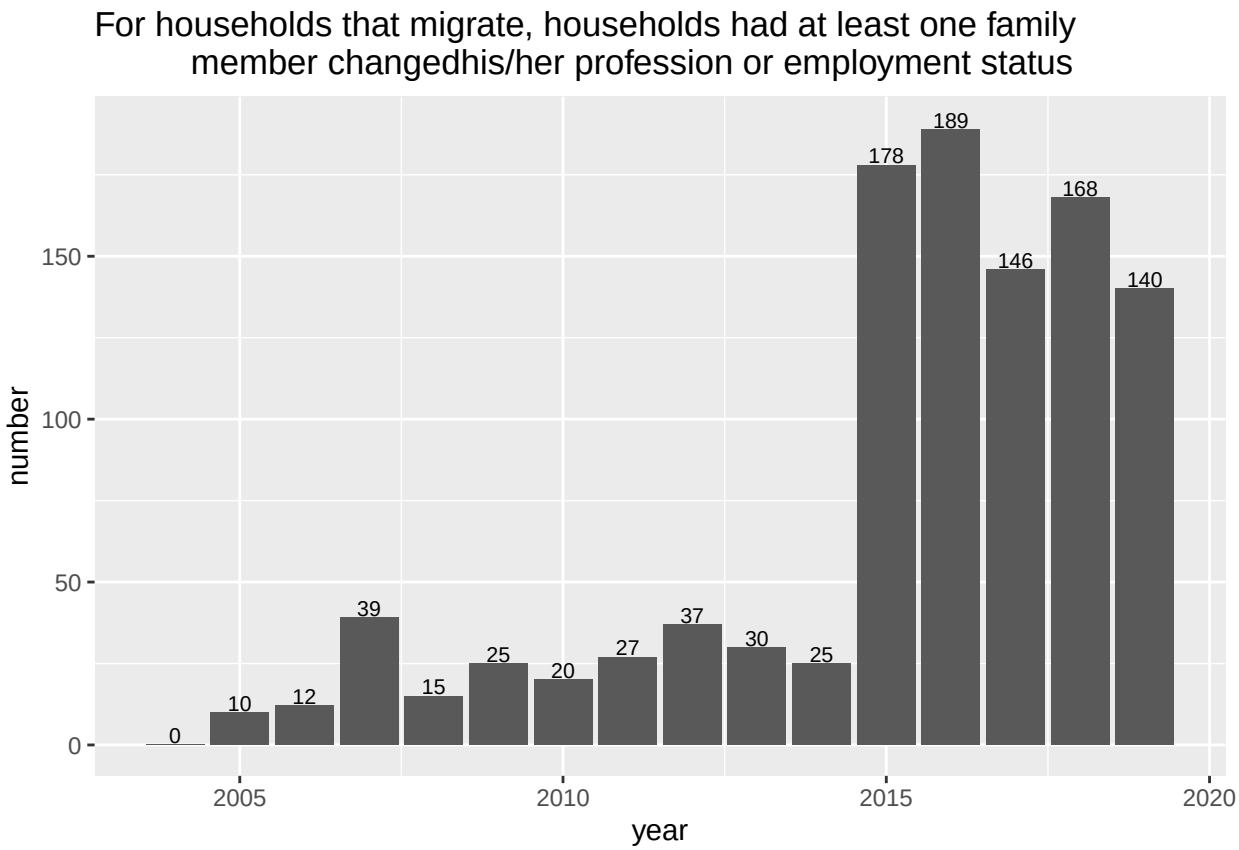


I like the first method better, because it is more smooth. The second method has a sudden increase after 2015 which seems to be unreal.

For households that migrate, find out how many households had at least one family member changed his/her profession or employment status.

```
# find the change in each year.
migrate_hh = apply(migrate,1,sum)>0
migrate_hh = names(migrate_hh[migrate_hh==TRUE])
dat_total$year = as.numeric(as.character(dat_total$year))

years = 2004:2019
change_number = c(0)
for(y in 2:16){
  year_change = 0
  # check household migrated some year in each year from 2005
  for(hh in migrate_hh){
    hh_change = 0
    temp = dat_total[dat_total$idmen==hh&dat_total$year==years[y],]
    # check whether the household is surveyed in that year
    if (dim(temp)[1]>0){
      temp[is.na(temp)] = 0
      # check whether the household migrated that year
      if (unique(temp$myear)==unique(temp$year) | unique(temp$move)==2){
        temp = dat_total[dat_total$idmen==hh&dat_total$year%in%c(years[y-1],years[y]),]
        ids_name = unique(temp$idind)
```

```
        temp[is.na(temp)] = -1
        # check whether each member this year change prof or emp
        for (id in 1:length(ids_name)){
          id_temp = temp[which(temp$idind==ids_name[id]),]
          # if the member is not in the household last year, skip
          if (dim(id_temp)[1]==2){
            # if member's prof of epm changes from NA or changes to NA, he/she is also included
            if (length(unique(id_temp$profession))>1 | length(unique(id_temp$empstat))>1){
              hh_change = hh_change + 1
            }
          }
        }
      }
    }
    if (hh_change > 0){year_change = year_change + 1}
  }
  change_number = c(change_number, year_change)
}
change_number_dat = data.frame(year=years, number=change_number)
ggplot(change_number_dat,aes(year,number))+
  geom_bar(stat = 'identity')+
  geom_text(aes(label=number, y=number+0.5), position="dodge", vjust=0,size = 8/.pt) +
  ggtitle(label='For households that migrate, households had at least one family
          member changedhis/her profession or employment status')
```



Exercise 4 Attrition

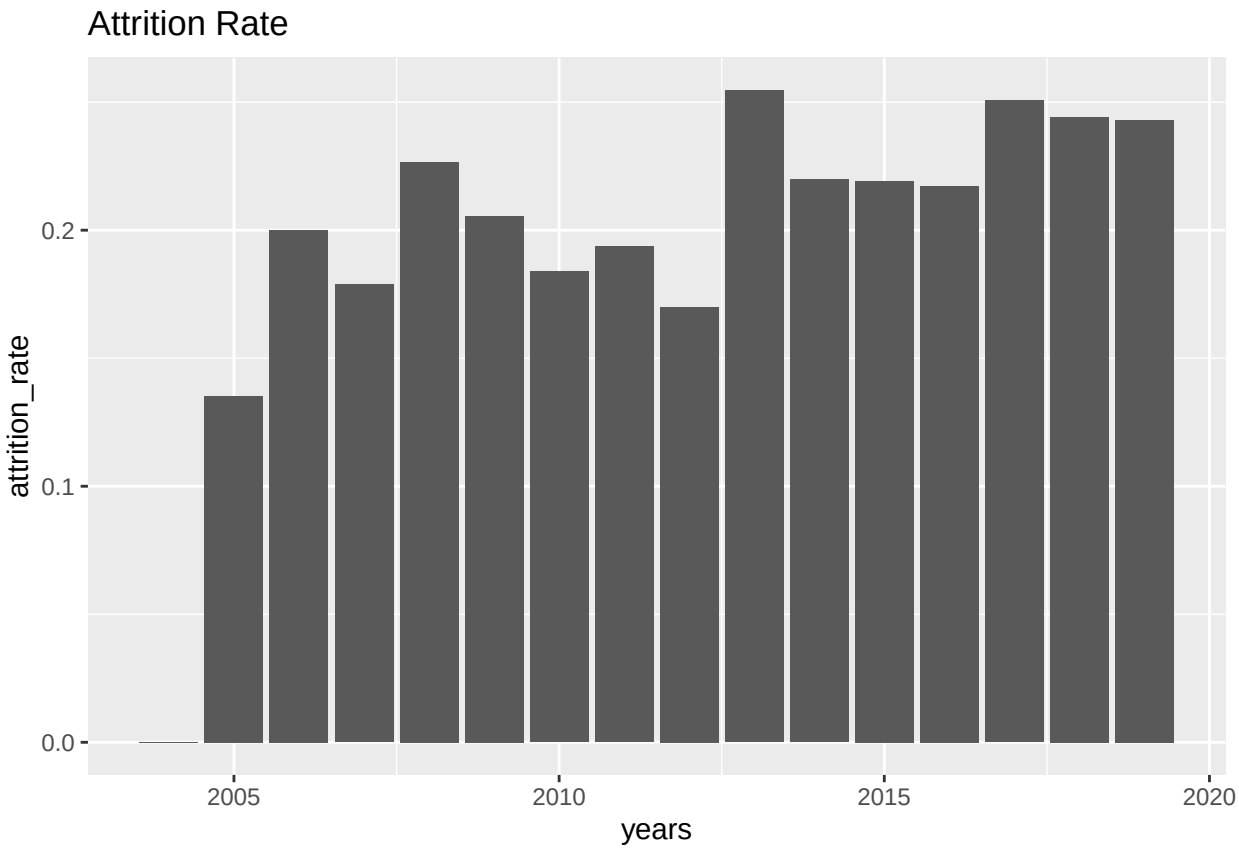
```
dat_total$idind = as.integer64(as.character(dat_total$idind))
attrition = c(0)
attrition_rate = c(0)
for (y in 2005:2019){
  number_stay = 0
  id_first = unique(dat_total[dat_total$year==y-1,'idind'])
  id_second = unique(dat_total[dat_total$year==y,'idind'])
  for (i in 1:length(id_first)){
    if (id_first[i] %in% id_second) {number_stay = number_stay + 1}
  }
  attrition = c(attrition, length(id_first) - number_stay)
  attrition_rate = c(attrition_rate, (length(id_first) - number_stay)/length(id_first))
}
attrition
```

```
##  [1]    0 2996 4850 4457 5873 5245 4707 5137 4599 7263 5782 5866 5784 6677 6201
## [16] 6003
```

```
attrition_rate
```

```
##  [1] 0.0000000 0.1352962 0.2000743 0.1787089 0.2266955 0.2056056 0.1837882
##  [8] 0.1936226 0.1698866 0.2546188 0.2198228 0.2191832 0.2172313 0.2507040
## [15] 0.2442012 0.2431250
```

```
data.frame(years = 2004:2019, attrition_rate = attrition_rate) %>%
  ggplot(aes(x = years, y = attrition_rate)) +
  geom_bar(stat = 'identity')+
  ggtitle(label='Attrition Rate')
```


Attrition Rate
attrition_rate
0.0
0.1
0.2
2005
2010
2015
2020
years