ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC & KỸ THUẬT MÁY TÍNH

CẤU TRÚC RỜI RẠC CHO KHMT (CO1007)

Thống kê khảo sát kết quả
Top 200 nhà đầu tư chứng khoán Việt Nam

GVHD: Nguyễn Ngọc Lễ
SV thực hiện: Nguyễn Trường Thản – 2114798
Nguyễn Trần Quang Vũ – 2115325
Thái Anh Khương – 2113806
Lê Phan Quốc Vũ – 2115321
Phạm Ngọc Khai – 2113650
Nguyễn Hữu Thông – 2114917

Tp. Hồ Chí Minh, Tháng 11/2022

Mục lục

1 Động cơ nghiên cứu

Thị trường chứng khoán đề cập đến các thị trường công khai tồn tại để phát hành, mua và bán cổ phiếu giao dịch trên sàn giao dịch chứng khoán hoặc mua bán qua quầy.

Một trong các mục đích mà thị trường chứng khoán phục vụ là cung cấp cho các nhà đầu tư - những người mua cổ phiếu - cơ hội chia sẻ lợi nhuận của các công ty được giao dịch công khai.

Một trong những nguyên tắc cơ bản khi đầu tư chứng khoán của nhà đầu tư là - rủi ro và cơ hội đi đôi với nhau. Chúng tăng hoặc giảm kết hợp với nhau. Các khoản đầu tư mang lại lợi nhuận tiềm năng cao hơn sẽ mang lại mức độ rủi ro tương ứng cao hơn. Tương tự như vậy, các khoản đầu tư mang lại lợi tức đầu tư tiềm năng thấp hơn thường mang lại sự an toàn cao hơn và ít rủi ro hơn.

Chiến tranh, đại dịch và thị trường chứng khoán lao dốc đã ảnh hưởng đến vận mệnh của nhiều người giàu nhất hành tinh trong năm nay. Với dữ liệu thu thập 200 nhà đầu tư đứng đầu danh sách thị trường chứng khoán Việt Nam về giá trị tài sản qua các năm sẽ được dùng để phân tích dự đoán xu hướng thu nhập và vị trí của nhà đầu tư.

2 Mục tiêu

Trong bài tập lớn này, các sinh viên sẽ bắt đầu với các bài toán thống kê đơn giản từ những dữ liệu được cung cấp. Qua đó, các em sẽ tìm ra những con số thú vị, có ý nghĩa đối với các dữ liệu thực tế từ ứng dụng IoT trong nông nghiệp. Những kết quả mà các em tìm ra sẽ là bước khởi đầu cho việc khai phá nguồn dữ liệu của hệ thống sau này, nhằm đạt tới mục tiêu nâng cao kỹ năng lập trình, kỹ năng giải quyết vấn đề cho người học, kỹ năng làm việc nhóm cũng như hướng tới mục tiêu cao hơn là đam mê trong làm việc, học tập và nghiên cứu.

3 Mô tả dữ liệu

Dữ liệu qua các năm là danh sách 200 nhà đầu tư có giá trị tài sản đứng đầu thị trường chứng khoán gồm các thuộc tính chính **"Vitri, Ten, Tuoi, Noisinh, Congtac, SoHuu, GiatriTaiSan"** được lưu tron file **xlsx**.

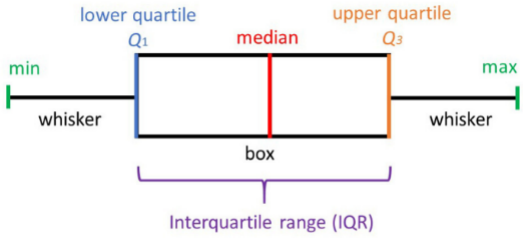
1. *Vitri*: Vị trí nhà đầu tư
2. *Ten*: Tên nhà đầu tư
3. *Tuoi* Tuổi nhà đầu tư
4. *Noisinh*: Nơi sinh nhà đầu tư
5. *Congtac*: Công tác nhà đầu tư
6. *SoHuu*: Sở hữu nhà đầu tư
7. *GiatriTaiSan*: Giá trị tài sản nhà đầu tư đơn vị tỷ đồng.

4 Cơ sơ lý thuyết

- Điểm tứ phân vị (quartile) là giá trị bằng số phân chia một nhóm các kết quả quan sát bằng số thành bốn phần, mỗi phần có số liệu quan sát bằng nhau (=25% số kết quả quan sát). Tứ phân vị có 3 giá trị, đó là tứ phân vị thứ nhất (Q1), thứ nhì (Q2) và thứ ba (Q3). Ba giá trị này chia một tập hợp dữ liệu (đã sắp xếp dữ liệu theo trật từ từ bé đến lớn) thành 4 phần có số lượng quan sát đều nhau.
  + Tứ phân vị thứ hai (Q2) (trung vị, median) là số nằm ở giữa trong một danh sách các số được sắp xếp tăng dần hoặc giảm dần (Nếu tập dữ liệu có số lượng điểm dữ liệu là số lẻ, giá trị trung vị là số nằm ở giữa có cùng một số lượng điểm dữ liệu bên dưới và bên trên. Nếu tập dữ liệu có số lượng điểm dữ liệu là số chẵn, cặp điểm dữ liệu ở giữa được cộng lại và chia hai để xác định giá trị trung bình)
  + Tứ phân vị thứ nhất (Q1) bằng trung vị phần dưới
  + Tứ phân vị thứ nhất (Q3) bằng trung vị phần trên
- Outliers (dữ liệu ngoại lai/ dữ liệu bất thường) là một quan sát nằm cách xa bất thường so với các giá trị khác trong tập dữ liệu

  Cách xác định điểm ngoại lại (outliers):
  + Đặt $IQR = Q3 - Q1$
  + Outliers là những điểm có giá trị nhỏ hơn $Q1 - 1.5 * IQR$ hoặc lớn hơn $Q3 + 1.5 * IQR$

    Trong đó, $Q3$ và $Q1$ là tứ phân vị 3 và tứ phân vị 1 trong dữ liệu
- Biểu đồ boxplot là biểu đồ diễn tả 5 vị trí phân bố của dữ liệu, đó là: giá trị nhỏ nhất (min), tứ phân vị thứ nhất (Q1), trung vị (median), tứ phân vị thứ 3 (Q3) và giá trị lớn nhất (max).



Hình 1: *Ví dụ về biểu đồ boxplot*

- Phương sai (Variance) là phép đo mức chênh lệch giữa các số liệu trong một tập dữ liệu trong thống kê. Nó đo khoảng cách giữa mỗi số liệu với nhau và đến giá trị trung bình của tập dữ liệu

  Công thức tính phương sai: $\sigma^2 = \frac{\Sigma(\chi - \mu)^2}{N}$
- Độ lệch chuẩn (Standard deviation) là thước đo độ phân tán của một tập hợp các giá trị so với giá trị trung bình của chúng. Độ lệch chuẩn của 1 giá trị càng thấp nghĩa là giá trị đó càng gần với giá trị trung bình của tập hợp.

  Công thức tính độ lệch chuẩn: $\sigma = \sqrt{\sigma^2}$

5 Xử lý dữ liệu

- Các thư viện thiết lập trong R:

```
library(readxl)
library(stringr)
library(tidyverse)
library(gridExtra)
library(ggplot2)
library(lattice)
library(grid)
```

- Gộp dữ liệu của từng năm thành một dữ liệu gồm tất cả các năm liên tiếp với nhau (từ năm 2014 đến năm 2022) với cột "Vitri" làm chuẩn
  + Đọc dữ liệu của từng năm:

```
data2022 <- read_excel("CKtop200_2022.xlsx")
data2021 <- read_excel("CKtop200_2021.xlsx")
#Tuong tu cho cac nam con lai
```

  + Gộp dữ liệu của tất cả các năm thành một dataframe (*data.new*):

```
data <- list(data2014, data2015, data2016, data2017, data2018,
             data2019, data2020, data2021, data2022)
data.new <- data %>% reduce(full_join, by='Vitri')
```

- Xử lý các số liệu bị thiếu trong dữ liệu (đưa các dữ liệu bị thiếu về "0"):

```
for(i in 2:ncol(data.new)){
  if((i-1)%%6 == 2){ #Tuoi
    data.new[i] <- replace(data.new[i], data.new[i]=="-","0")
    data.new[i] <- replace(data.new[i], is.na(data.new[i]), "0")
  }
  else if((i-1)%%6 == 3){#NoiSinh
    data.new[i] <- replace(data.new[i], data.new[i]=="--","0")
    data.new[i] <- replace(data.new[i], data.new[i]=="N/A","0")
    data.new[i] <- replace(data.new[i], is.na(data.new[i]), "0")
  }
  else{#Other
    data.new[i] <- replace(data.new[i], is.na(data.new[i]), "0")
  }
}
```

- Xử lý dữ liệu biến *TaiSan* (bỏ đi dấu "," ở các giá trị lớn hơn 1000):
  + Sử dụng hàm *gsub()* để chuyển ký tự "," về ký tự rỗng ""

```
data.matrix <- as.matrix(data.new)
for(i in 1:ncol(data.matrix)){
  if((i-1)%%6 == 0){
    for(j in 1:200){
      data.matrix[j,i] <- gsub(",","",data.matrix[j,i])
    }
  }
}
data.new <- as.data.frame(data.matrix)
```

1. Vị trí các biến trong *data.new*

   + Vector *index.tuoi* chứa index của các biến *Tuoi* của tất cả các năm trong *data.new*

```
index.tuoi <- c()
for(i in 0:8){
  index.tuoi <- append(index.tuoi, 3+6*i)
}
```

   + Vector *index.ten* chứa index của các biết *Ten* của tất cả các năm trong *data.new*

```
index.ten <- c()
for(i in 0:8){
  index.ten <- append(index.ten, 2+6*i)
}
```

   + Vector *index.noiSinh* chứa index của các biến *Noisinh* của tất cả các năm trong *data.new*

```
index.noiSinh <- c()
for(i in 0:8){
    index.noiSinh <- append(index.noiSinh, 4+6*i)
}
```

   + Vector *index.soHuu* chứa index của các biến *SoHuu* của tất cả các năm trong *data.new*

```
index.soHuu <- c()
for(i in 0:8){
  index.soHuu <- append(index.soHuu, 6+6*i)
}
```

   + Vector *index.taiSan* chứa index của các biến *GiatriTaiSan* của tất cả các năm trong *data.new*

```
index.taiSan <- c()
for(i in 0:8){
  index.taiSan <- append(index.taiSan, 7+6*i)
}
```

6 Bài làm

Mã đề: 9367

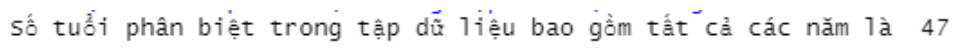
i) Nhóm câu hỏi liên quan đến tổng quát dữ liệu

1) **Cho biết có bao nhiêu tuổi phân biệt trong tập dữ liệu bao gồm tất cả các năm**

- Trình bày cách làm:
  - Dùng hàm *append()* để gộp biến *Tuoi* của tất cả các năm thành một vector *tuoi*
  - Dùng hàm *unique()* cho vector *tuoi* để tạo vector gồm các giá trị phân biệt và dùng hàm *length()* để tìm độ dài vector đó
- Source code:

```
tuoi <- as.numeric(tuoi)
data.result.i.1 <- length(unique(tuoi))
cat("So tuoi phan biet trong tap du lieu bao gom tat ca cac nam la: "
    , data.result.i.1)
```

- Kết quả chạy code:



Hình 2: *Số tuổi phân biệt trong tập dữ liệu bao gồm tất cả các năm*

2) **Cho biết tuổi và số lượng người với tuổi đó theo mỗi năm.**

- Trình bày cách làm:
  - Xây dựng hàm *i.2.countAge(tuoi.year, tuoiPhanBiet)*
    Trong đó: *tuoi.year* là biến *Tuoi* của dữ liệu trong năm, *tuoiPhanBiet* là vector gồm các tuổi khác nhau của tất cả các năm được sắp xếp tăng dần
    Kết quả trả về của hàm là vector thống kê tuổi cho năm đang xét
    * Tạo vector *result.i.2* là vector trả về của hàm có độ dài là số lượng tuổi phân biệt
    * Dùng hàm *table(tuoi.year)* để biết được số lần xuất hiện của từng độ tuổi trong vector *tuoi.year*
    * So sánh giá trị của *tuoiPhanBiet* với độ tuổi của hàm *table()*, nếu bằng nhau thì gán giá trị số lần xuất hiện của độ tuổi đó vào vector *result.i.2* với index tương ứng của *tuoiPhanBiet*
  - Sử dụng hàm *i.2.countAge(tuoi.year, tuoiPhanBiet)* cho biến *Tuoi* của từng năm rồi gán vector trả về vào dataframe *data.result.i.2* (dataframe kết quả)
- Source code:

```
i.2.countAge <- function(tuoi.year, tuoiPhanBiet){
  result.i.2 <- c(rep(NA, length(tuoiPhanBiet)))
  countTuoi.year <- as.data.frame(table(tuoi.year))
  for(i in 1:length(tuoiPhanBiet)){
    for(j in 1:length(countTuoi.year[,1])){
      if(as.numeric(tuoiPhanBiet[i]) == countTuoi.year[j,1]){
        result.i.2[i] <- countTuoi.year[j,2]
      }
      else if(as.numeric(tuoiPhanBiet[i]) < as.numeric(countTuoi.year[j,1]))
        break
      else next
    }
  }
  return(result.i.2)
}

tuoiPhanBiet <- sort(unique(tuoi))
```

```
data.result.i.2 <- as.data.frame(matrix(nrow=length(unique(tuoi)), ncol=10))
data.result.i.2[,1] <- tuoiPhanBiet
colnames(data.result.i.2) <-c("Year","2014", "2015", "2016", "2017", "2018", "2019",
    "2020", "2021", "2022")

for(i in 1:9){
  data.result.i.2[,i+1] <- i.2.countAge(data.new[,index.tuoi[i]], tuoiPhanBiet)
}
for(i in 2:10){
  data.result.i.2[,i] <- replace(data.result.i.2[,i],is.na(data.result.i.2[,i]),"")
}
data.result.i.2
```

- Kết quả chạy code:

	Year	2014	2015	2016	2017	2018	2019	2020	2021	2022
1	0	67	71	62	75	78	72	76	84	77
2	30								1	1
3	31				1			1		
4	33			1	1	2	1	1		
5	34	1	1	1	1	1	1	1	2	2
6	36				1	1				
7	37		1	1						
8	38	3	3	3	2	1	1	1	1	1
9	39	1		1	1	2	3	3	4	6
10	40	1	1					2	1	1
11	41	1	2	2	1	1	2	3	1	1
12	42	3	3	3	4	5	4	3	3	3
13	43							1	1	2
14	44	1	1	1	1	2	4	4	2	2
15	45	1							1	1
16	46	4	6	5	2	2	2	4	3	3
17	47	1	1	1	1	2	2	2	4	4
18	48	4	3	2	5	3	3	1	2	1
19	49	5	3	5	5	3	4	4	3	5
20	50	6	8	5	3	3	3	2	4	4
21	51	2	5	4	3	3	4	3	3	4
22	52	7	6	6	5	8	9	9	7	8
23	53	4	4	5	3	3	3	2	2	2
24	54	3	4	5	8	8	8	7	6	6
25	55	3	3	3	3	4	4	4	3	5

Hình 3: *Bảng thể hiện tuổi và số lượng người ứng với tuổi đó theo mỗi năm (có tổng cộng 47 hàng)*

3) **Theo mỗi năm, cho biết cập tuổi liên tiếp nào mà số người của tuổi sau cao hơn số người của tuổi trước là nhiều nhất.**

- Trình bày cách làm:
  - Xây dựng hàm $i.3.findIndexMaxDiffentAge(tuoi.year)$
    Trong đó: $tuoi.year$ là biến $Tuoi$ của dữ liệu một năm nào đó
    Kết quả trả về của hàm là một vector chứa các giá trị index mà tại đó số người tuổi sau lớn hơn só người tuổi trước là nhiều nhất
    * Tìm giá trị lớn nhất của số người tuổi sau trừ cho số người tuổi trước
    * Tìm index nào mà số người tuổi sau trừ cho số người tuổi trước bằng giá trị lớn nhất đã tìm ở trên rồi gán vào vector trả về của hàm

- Xây dựng hàm $i.3.result(tuoi.year, indexMaxDiffentAge)$
  Trong đó: $tuoi.year$ là biến $Tuoi$ của dữ liệu một năm nào đó và $indexMaxDiffentAge$ là giá trị trả về của hàm $i.3.findIndexMaxDiffentAge$ ứng với $tuoi.year$ đó.
  Kết quả trả về là dataframe có cột 1 là tuổi trước và cột 2 là tuổi sau của một năm thỏa mãn yêu cầu của đề
  * Gán cột thứ nhất và cột thức hai của dataframe trả về với giá trị tuổi của $tuoi.year$ ứng với index của $indexMaxDiffentAge$ và $indexMaxDiffentAge + 1$
- Sử dụng hai hàm đã khai báo ở trên để sử dụng cho biến $Tuoi$ của từng năm

- Source code:

```
i.3.findIndexDiffentAge <- function(tuoi.year){
  maxDifferentAge <- 0
  countTuoi.year <- as.data.frame(table(tuoi.year))
  for(i in 2:(length(countTuoi.year[,1])-1)){
    if((countTuoi.year[i+1,2] - countTuoi.year[i,2]) > maxDifferentAge)
      maxDifferentAge <- countTuoi.year[i+1,2] - countTuoi.year[i,2]
  }

  indexDiffentAge <- c()
  for(i in 2:(length(countTuoi.year[,1])-1)){
    if((countTuoi.year[i+1,2] - countTuoi.year[i,2]) == maxDifferentAge)
      indexDiffentAge <- append(indexDiffentAge, i)
  }
  return(indexDiffentAge)
}

i.3.result <- function(tuoi.year, indexDiffentAge){
  countTuoi.year <- as.data.frame(table(tuoi.year))
  i.3.data.year.result <- matrix(nrow = length(indexDiffentAge), ncol = 2)
  colnames(i.3.data.year.result) <- c("Tuoi dau", "Tuoi sau")
  for(i in 1:length(indexDiffentAge)){
    i.3.data.year.result[i,1] <- as.matrix(countTuoi.year[indexDiffentAge[i],1])
    i.3.data.year.result[i,2] <- as.matrix(countTuoi.year[indexDiffentAge[i]+1,1])
  }
  return(as.data.frame(i.3.data.year.result))
}

for(i in 1:9){
  indexDiffentAge <- i.3.findIndexDiffentAge(data.new[,index.tuoi[i]])
  i.3.data.year.result <- i.3.result(data.new[,index.tuoi[i]], indexDiffentAge)
  print(2013+i)
  print(i.3.data.year.result)
  cat("\n")
}
```

- Kết quả chạy code:

```
[1] 2014
  Tuoi dau Tuoi sau
1       51       52

[1] 2015
  Tuoi dau Tuoi sau
1       44       46
2       49       50
3       61       62

[1] 2016
  Tuoi dau Tuoi sau
1       57       58

[1] 2017
  Tuoi dau Tuoi sau
1       53       54
2       57       58

[1] 2018
  Tuoi dau Tuoi sau
1       63       64

[1] 2019
  Tuoi dau Tuoi sau
1       63       64

[1] 2020
  Tuoi dau Tuoi sau
1       51       52

[1] 2021
  Tuoi dau Tuoi sau
1       51       52
2       53       54
3       57       58

[1] 2022
  Tuoi dau Tuoi sau
1       38       39
2       63       64
```

Hình 4: *Cập tuổi liên tiếp mà số người tuổi sau cao hơn số người của tuổi trước nhiều nhất theo từng năm*

4) **Cho biết năm mà field tuổi bị bỏ trống (thiếu dữ liệu) là ít nhất.**

- Trình bày cách làm:
  - Năm mà field tuổi bị bỏ trống là ít nhất tương đương với việc năm mà field tuổi có dữ liệu 0 là ít nhất
  - Dùng hàm *min()* để tìm giá trị nhỏ nhất của hàng 1 dataframe *data.result.i.2* (từ cột 2 đến 10)
  - Dùng hàm *which()* để tìm index mà giá trị min đó xuất hiện trong vector trên
  - In ra màn hình *colnames(data.result.i.2)* với index là index đã tìm được ở trên cộng 1
- Source code:

```
tuoiWithNA <- as.numeric(data.result.i.2[1,(2:10)])
tuoiWithNAMin <- min(tuoiWithNA)
indexTuoiWithNAMin <- which(tuoiWithNA == tuoiWithNAMin)
print("Nam ma field tuoi bi bo trong it nha la: ")
colnames(data.result.i.2)[indexTuoiWithNAMin + 1]
```

- Kết quả chạy code:

```
[1] "Nam ma field tuoi bi bo trong it nha la: "
> colnames(data.result.i.2)[indexTuoiWithNAMin + 1]
[1] "2016"
```

Hình 5: *Năm mà field tuổi bị bỏ trống là ít nhất*

5) **Cho biết có bao nhiêu nơi sinh phân biệt trong tập dữ liệu bao gồm tất cả các năm.**

- Trình bày cách làm:
  - Tạo một vector *noiSinh* gồm 9 biến *Noisinh* của 9 năm gộp lại
  - Chuyển dữ liệu trong các ô nơi sinh về định dạng:
    - Ví dụ: "Tỉnh Nam Định" thành "Nam Định", "TP HCM" thành "Hồ Chí Minh"
    - Dùng hàm *str_to_title(string)* để chuyển các chữ đầu tiên sau dấu cách của ô nơi sinh đều viết hoa
  - Tạo vector *diaDanhLapLai* là dữ liệu chỉ gồm tỉnh, quốc gia hoặc dữ liệu "0" của vector *noiSinh*
  - Dùng hàm *str_count()* để tìm số lượng địa danh trong 1 ô *NoiSinh* bằng cách đếm số ký tự "," trong ô đó cộng thêm 1 (gán vào biến *soLuongDiaDanh*)
  - Dùng hàm *str_split_fixed* để tách ô *NoiSinh* thành *soLuongDiaDanh* chuỗi ký tự (là từng địa danh một) (gán vào vector *noiSinhTem*)
  - Sau cùng, ta gán chuỗi ký tự cuối cùng (có index là *soLuongDiaDanh*) trong *noiSinhTem* vào vector *diaDanhLapLai*
  - Dùng hàm *sort(unique(diaDanhLapLai))*, ta thu được vector gồm các tỉnh, quốc gia nơi sinh khác nhau sắp xếp theo thứ tự alpha rồi gán vào vector *diaDanhPhanBiet*
  - Dùng hàm *length()* để tìm độ dài của vector *diaDanhPhanBiet*, ta thu được kết quả là số nơi sinh phân biệt trong tập dữ liệu gồm tất cả các năm
- Source code:

```
noiSinh <- c()
for(i in index.noiSinh){
    noiSinh <- append(noiSinh, data.matrix[,i])
}

diaDanhLapLai <- c()
for(i in 1:length(noiSinh)){
  noiSinh[i] <- gsub("-", ",", noiSinh[i])
  noiSinh[i] <- gsub("–", ",", noiSinh[i])
  noiSinh[i] <- gsub("tỉnh ","", noiSinh[i])
  noiSinh[i] <- gsub("Tỉnh ", "", noiSinh[i])
  noiSinh[i] <- gsub("TP ","", noiSinh[i])
  noiSinh[i] <- gsub("TP. ","", noiSinh[i])
  noiSinh[i] <- gsub("TP.","", noiSinh[i])
  noiSinh[i] <- gsub("T.P ","", noiSinh[i])
  noiSinh[i] <- gsub("Tp. ","", noiSinh[i])
  noiSinh[i] <- gsub("Hoá","Hóa", noiSinh[i])
  noiSinh[i] <- gsub("HCM","Hồ Chí Minh", noiSinh[i])
  noiSinh[i] <- str_to_title(noiSinh[i])
  soLuongDiaDanh <- str_count(noiSinh[i], ",") + 1
  noiSinhTem <- str_split_fixed(noiSinh[i], ", ", soLuongDiaDanh)
  diaDanhLapLai <- append(diaDanhLapLai, noiSinhTem[soLuongDiaDanh])
}

diaDanhPhanBiet <- sort(unique(diaDanhLapLai))
data.result.i.5 <- length(diaDanhPhanBiet)
cat("Số nơi sinh phân biệt trong tập dữ liệu gồm tất cả các năm là: "
    , data.result.i.5)
```

Hình 6: *Source của câu i)5)*

- Kết quả chạy code:

```
Số nơi sinh phân biệt trong tập dữ liệu gồm tất cả các năm là:  59
```

Hình 7: *Số nơi sinh phân biệt trong tập dữ liệu gồm tất cả các năm*

6) **Cho biết nơi sinh và số lượng người ở nơi đó theo mỗi năm.**

- Trình bày cách làm:
  - Tạo một dataframe *data.diaDanh* gán bằng dataframe *data.new* gồm dữ liệu của tất cả các năm (xem lại ở phần "Xử lý dữ liệu"). Thay thế các cột *Noisinh* thành các các địa danh chỉ gồm tỉnh, quốc gia hoặc giá trị "0" (Lấy từ vector *diaDanhLapLai*)
  - Các bước còn lại tương tự câu *i*)2), thay thế vector *tuoi.year* thành *noiSinh.year* (lấy từ dataframe *data.diaDanh*, thay vector *tuoiPhanBiet* thành vector *diaDanhPhanBiet*
- Source code:

```
data.diaDanh <- data.new
data.diaDanh[,4+6*0] <- diaDanhLapLai[1:200]
data.diaDanh[,4+6*1] <- diaDanhLapLai[201:400]
#tuong tu cho cac index tu 2 den 8
i.6.countNoiSinh <- function(noiSinh.year, diaDanhPhanBiet){
  result.i.6 <- c(rep(NA, length(diaDanhPhanBiet)))
  countNoiSinh.year <- as.data.frame(table(noiSinh.year))
  for(i in 1:length(diaDanhPhanBiet)){
    for(j in 1:length(countNoiSinh.year[,1])){
      if(diaDanhPhanBiet[i] == countNoiSinh.year[j,1]){
        result.i.6[i] <- countNoiSinh.year[j,2]
      }
    }
  }
  return(result.i.6)
}
data.result.i.6 <- as.data.frame(matrix(nrow=length(diaDanhPhanBiet), ncol=10))
data.result.i.6[,1] <- diaDanhPhanBiet
colnames(data.result.i.6) <-c("Year","2014", "2015", "2016", "2017", "2018", "2019",
    "2020", "2021", "2022")
for(i in 1:9){
  data.result.i.6[,i+1] <- i.6.countNoiSinh(data.diaDanh[,index.noiSinh[i]],
      diaDanhPhanBiet)
}
for(i in 2:10){
  data.result.i.6[,i] <- replace(data.result.i.6[,i],is.na(data.result.i.6[,i]),"")
}
data.result.i.6
```

- Kết quả chạy code:

	Year	2014	2015	2016	2017	2018	2019	2020	2021	2022
1	0	84	91	94	103	115	117	119	123	122
2	Ấn Độ	1	1							
3	An Giang	1	1	1	1	1	1	1	1	1
4	Anh Quốc		1	1	1		1	1		
5	Bắc Giang	1	1	1		1				
6	Bắc Ninh	2	3	2	4	3	3	2	4	3
7	Bến Tre	1					1			
8	Bình Định	4	6	5	5	4	4	3	3	3
9	Bình Dương	1	1			1				
10	Bình Thuận	1	1	1	1	1	1	1	1	1
11	Cà Mau	1			1	1		1	1	1
12	Cần Thơ						1			
13	Đà Nẵng	3	2	4	3	2	2	2	2	2
14	Đài Loan	1	1	1	1		1	1	1	1
15	Đồng Nai	1		1						
16	Đồng Tháp	2	2	3	1	1				
17	Gia Lai	1								
18	Hà Bắc	1	1	1	1	1	1	1	1	1
19	Hà Nam	4	3	1	1	1	1	2	2	3
20	Hà Nam Ninh	1	1	1	1	1	1	1	1	1
21	Hà Nội	12	12	9	11	9	9	9	10	10
22	Hà Tây			1						
23	Hà Tĩnh	4	6	5	4	4	4	4	5	4
24	Hải Dương	2	1	2	2	1	1	1	3	2
25	Hải Hưng	1	1	1	1	1	1	1	1	1

Hình 8: *Nơi sinh và số lượng người ở nơi đó theo mỗi năm (có tất cả 59 hàng)*

7) **Cho biết các năm nào mà số liệu nơi sinh bị trống (thiếu dữ liệu) nhiều nhất và cho biết số lượng đó.**

- Trình bày cách làm:
  - Sử dụng hàm *which.max()* cho hàng đầu tiên, từ cột 2 đến 10 trong dataframe *data.result.i.6* để tìm giá trị MAX trong hàng này
  - Dùng hàm *which()* lần nữa để tìm index những giá trị MAX này, sẽ trả về một vector nếu trong hàng này có nhiều giá trị MAX
  - In ra màn hình giá trị và *colnames(data.result.i.6)* với index tương ứng cộng thêm 1
- Source code:

```
noiSinhNA <- as.numeric(data.result.i.6[1,(2:10)])
maxNA <- noiSinhNA[which.max(noiSinhNA)]
colnames(data.result.i.6)[which(noiSinhNA == maxNA) + 1]
   data.result.i.6[1, which(noiSinhNA == maxNA) + 1]
```

- Kết quả chạy code:

```
[1] "2021"
> data.result.i.6[1, which(noiSinhNA == maxNA) + 1]
[1] "123"
```

Hình 9: *Năm mà số liệu nơi sinh bị trống nhiều nhất và số lượng đó là*

8) **Cho biết nơi sinh nào mà tổng số lượng người qua tất cả các năm là đông nhất.**

- Trình bày cách làm:
  - Tạo một dataframe *tenNoiSinhLapLai* gồm cột một là tên của tất cả các nhà đầu tư của tất cả các năm, cột hai là nơi sinh của các nhà đầu tư tương ứng (đã chuyển về tỉnh, quốc gia)
  - Tạo một dataframe *tenNoiSinhPhanBiet* gồm cột một là tên của tất cả các nhà đầu tư của các nhà đầu tư phân biệt của tất cả các năm, cột hai là dữ liệu *NA*, ta cần gán cột thứ hai này với nơi sinh tương ứng của các nhà đầu tư
  - So sánh $tenNoiSinhPhanBiet[,1]$ (tên) với $tenNoiSinhLapLai[,1]$ (tên), nếu bằng nhau thì ta gán $tenNoiSinhPhanBiet[,2]$ (nơi sinh) bằng $tenNoiSinhLapLai[,2]$ (nơi sinh) với index tương ứng
  - Ta thu được dataframe *tenNoiSinhPhanBiet* có cột 2 là nơi sinh của các nhà đầu tư phân biệt
  - Dùng hàm *table()* để tìm số lần xuất hiện của các giá trị nơi sinh trong cột 2 của *tenNoiSinhPhanBiet*.
  - Tìm max của số lần xuất hiện đó và in ra màn hình nơi sinh tương ứng với giá trị max

- Source code:

```
ten <- c()
for(i in index.ten){
  ten <- append(ten, data.new[,i])
}
tenPhanBiet <- unique(ten)

tenNoiSinhLapLai <- as.matrix(data.frame(ten,diaDanhLapLai))
tenNoiSinhPhanBiet <- as.matrix(data.frame(tenPhanBiet,
                                           rep(NA, length(tenPhanBiet))))

for(i in 1:nrow(tenNoiSinhPhanBiet)){
  for(j in 1:nrow(tenNoiSinhLapLai)){
    if(tenNoiSinhPhanBiet[i, 1] == tenNoiSinhLapLai[j,1]){
      if(tenNoiSinhLapLai[j,2] == "0") next
      else tenNoiSinhPhanBiet[i,2] <- tenNoiSinhLapLai[j,2]
    }
  }
}

tableNoiSinhTheoTen <- data.frame(table(tenNoiSinhPhanBiet[,2]))
data.result.i.8 <- tableNoiSinhTheoTen[which.max(tableNoiSinhTheoTen$Freq),1]
print("Noi sinh ma tong luong nguoi qua tat ca cac nam la dong nhat la: ")
print(data.result.i.8)
```

- Kết quả chạy code:

```
[1] "Nơi sinh mà tổng lượng người qua tất cả các năm là đông nhất là: "
> print(data.result.i.8)
[1] Hà Nội
```

Hình 10: *Nơi sinh mà tổng lượng người qua tất cả các năm là đông nhất*

9) **Cho biết có bao nhiêu mã cổ phiếu phân biệt trong tập dữ liệu bao gồm tất cả các năm.**

- Trình bày cách làm:
  - Tạo vector *soHuu* gồm 9 cột *Sohuu* trong tập dữ liệu của từng năm gộp lại (theo thứ tự từ năm 2014 đến 2022)
  - Loại bỏ khoảng trắng ở các ô dữ liệu trong vector *soHuu*
  - Dùng hàm *str_count()* để đếm số lượng cổ phiếu của từng ô bằng cách đếm ký tự "cp"
  - Nếu số lượng cổ phiếu trong ô đó bằng 0 thì ta gán thêm dữ liệu "0" vào vector kết quả *maCoPhieu* (thể hiện mã cổ phiếu của tất cả các năm)
  - Nếu số lượng cổ phiếu trong ô đó lớn hơn 0:
    - Dùng cấu trúc $unlist(gregexpr('character', my\_string))$ để tìm vị trí ký tự "-" và ký tự ":" trong từng chuỗi ký tự
    - Dùng hàm *substr()* để cắt từng chuỗi ký tự này từ vị trí ký tự "-" đến ký tự ":". Sau đó thêm kết quả này vào vector *maCoPhieu*
  - Ta tìm số lượng cổ phiếu phân biệt trong vector *maCoPhieu* bằng hàm *unique()* và hàm *length()*

- Source code:

```
soHuu <- c()
for(i in index.soHuu){
  soHuu <- append(soHuu, data.new[,i])
}

maCoPhieu <- c()
for(i in 1:length(soHuu)){
  if(soHuu[i] == "0") maCoPhieu <- append(maCoPhieu, "0")
  else {
    soLuongCoPhieu <- str_count(soHuu[i], "cp")
    for(j in 1:soLuongCoPhieu){
      soHuu[i] <- gsub(" ","", soHuu[i])
      indexStart <- unlist(gregexpr('-', soHuu[i])) + 1
      indexEnd <- unlist(gregexpr(':', soHuu[i])) - 1
      maCoPhieu <- append(maCoPhieu, substr(soHuu[i], indexStart[j], indexEnd[j]))
    }
  }
}
coPhieuPhanBiet <- sort(unique(maCoPhieu))
data.result.i.9 <- length(coPhieuPhanBiet)
cat("So ma co phieu phan biet trong tap du lieu gom tat ca cac nam la: ",
    data.result.i.9)
```

- Kết quả chạy code

```
So ma co phieu phan biet trong tap du lieu gom tat ca cac nam la:  297
```

Hình 11: *Số mã cổ phiếu phân biệt trong tập dữ liệu gồm tất cả các năm (tính cả dữ liệu 0)*

10) **Cho biết mã cổ phiếu và số lượng người có giữ mã cổ phiếu đó theo mỗi năm.**

- Trình bày cách làm:
  - Tạo một dataframe *coPhieu.matrix* có 10 cột, cột 1 là danh sách các mã cổ phiếu phân biệt (đã sắp xếp theo thứ tự alpha, hàng đầu tiên thể hiện quan sát không có dữ liệu). Các cột từ 2 đến 10 thể hiện cho năm từ 2014 đến 2022. Khởi tạo tất cả các cột còn lại giá trị 0.
  - Đếm các quan sát không có dữ liệu:
    Duyệt cột *soHuu* của từng năm, nếu xuất hiện dữ liệu "0" thì cộng giá trị hàng một, cột ứng với năm tương ứng của *coPhieu.matrix* một đơn vị
  - Đếm các quan sát có dữ liệu

    Ứng với mỗi mã cổ phiếu phân biệt, ta dùng hàm *str_detect()* để kiểm tra xem mã cổ phiếu này có trong ô dữ liệu của cột *Taisan* năm tương ứng không. Nếu có thì ta tăng giá trị điểm của mã cổ phiếu tương ứng ở năm tương ứng lên 1

- Source code:

```
coPhieu.matrix <- as.data.frame(matrix(nrow=length(coPhieuPhanBiet), ncol=10))
coPhieu.matrix[,1] <- sort(coPhieuPhanBiet, decreasing = FALSE)
for(i in 2:10){
  coPhieu.matrix[,i] <- c(rep(as.numeric(0),length(coPhieuPhanBiet)))
}
colnames(coPhieu.matrix) <-c("Year","2014", "2015", "2016", "2017", "2018", "2019",
    "2020", "2021", "2022")
#Dem quan sat khong co du lieu
for(a in 0:8){
  for(i in 1:200){
    if(data.matrix[i,6+6*a]=="0") coPhieu.matrix[1,a+2] <-
        as.numeric(coPhieu.matrix[1,a+2]) + 1
  }
}

#Dem quan sat co du lieu
for(a in 0:8){
  for(i in 2:nrow(coPhieu.matrix)){
    for(j in 1:200){
      if(str_detect(data.matrix[j,6+6*a], coPhieu.matrix[i,1])){
        coPhieu.matrix[i,a+2] <- as.numeric(coPhieu.matrix[i,a+2]) + 1
      }
    }
  }
}
data.result.i.10 <- coPhieu.matrix
for(i in 2:10){
  data.result.i.10[,i] <- replace(coPhieu.matrix[,i],coPhieu.matrix[,i]==0,"")
}
data.result.i.10
```

- Kết quả chạy code

	Year	2014	2015	2016	2017	2018	2019	2020	2021	2022
1	0	48	1	6	3	1				
2	AAA			3						
3	AAM		1	1	1	1	1	1	1	1
4	ABB			1						
5	ABT	1	1	2	1	1	1	1	1	1
6	ACB	11	11	11	11	11	8	5	6	6
7	ACL					1	1			1
8	ADC			1						
9	ADG							1		
10	ALP	2								
11	ANV	3	1	1	1	3	3	3	1	2
12	APC				1					
13	APH							1		
14	API	1			1				1	
15	APS	1			1				1	
16	ART		1	2	3	2	1	1	1	1
17	ASM	2	11	8	1	1	2	2	2	2
18	BAB					3	3	3		
19	BCE	1	1							
20	BCG							1	1	1
21	BCI	2	3	2	2	2	2	2	2	2
22	BDC						1			
23	BHS	1	1	1	1	1	2	1	1	1
24	BHT		1	1	1	1	1	1	1	1
25	BII		1							

Hình 12: *Mã cổ phiếu và số lượng người giữ mã cổ phiếu đó theo mỗi năm (có tất cả 297 hàng)*

11) **Cho biết mã cổ phiếu nào mà lượng người mua là cao nhất theo từng năm.**

- Trình bày cách làm:
  - Dùng dữ liệu *data.matrix* của câu *i*)10) với dữ liệu "0" chuyển thành dữ liệu *NA* (đặt tên là *coPhieu.matrixWithNA*)
  - Dùng hàm *which.max()* để tìm giá trị MAX của từng năm trong data trên (từ cột 2 đến cột 10)
  - In ra màn hình mã cổ phiếu với index và năm tương ứng
- Source code:

```
coPhieu.matrixWithNA <- coPhieu.matrix
for(i in 2:10){
  coPhieu.matrixWithNA[,i] <- replace(coPhieu.matrix[,i],coPhieu.matrix[,i]==0,NA)
}
maMaxCoPhieu.arr <- c()
for(i in 2:10){
  indexMaxCoPhieu <- which.max(coPhieu.matrixWithNA[2:length(coPhieuPhanBiet),i])
  maMaxCoPhieu.arr <- append(maMaxCoPhieu.arr,
                             coPhieu.matrixWithNA[indexMaxCoPhieu+1,1])
}

data.result.i.11 <- data.frame(c("2014", "2015", "2016", "2017", "2018", "2019",
    "2020", "2021", "2022"),
                           maMaxCoPhieu.arr)
colnames(data.result.i.11) <- c("Year","MaCoPhieu")
data.result.i.11
```

- Kết quả chạy code:

	Year	MaCoPhieu
1	2014	ACB
2	2015	ACB
3	2016	ACB
4	2017	ACB
5	2018	VPB
6	2019	VPB
7	2020	VPB
8	2021	VPB
9	2022	VPB

Hình 13: *Mã cổ phiếu mà lượng người mua cao nhất theo từng năm*

12) **Cho biết mã cổ phiếu nào mà lượng người mua là cao nhất tất cả các năm.**

- Trình bày cách làm:
  - Dùng hàm *table()* để xác định số lần xuất hiện từng mã cổ phiếu phân biệt trong vector gồm tất cả các cổ phiếu trong các năm (*maCoPhieu*) không tính dữ liệu "0"
  - Dùng hàm *which.max()* để tìm số lần xuất hiện lớn nhất của tất cả các cổ phiếu phân biệt
  - Dùng hàm *which()* để xác định các vị trí index có tần số xuất hiện giá trị MAX ở trên rồi in ra màn hình mã cổ phiếu với index tương ứng
- Source code:

```
table.coPhieuWihtout0 <- data.frame(table(maCoPhieu[maCoPhieu != "0"]))
var1CoPhieuWithout0 <- table.coPhieuWihtout0$Var1
freqCoPhieuWithout0 <- table.coPhieuWihtout0$Freq
valueMaxCoPhieuAllYear <- max(freqCoPhieuWithout0)
data.result.i.12 <-var1CoPhieuWithout0[which(freqCoPhieuWithout0 ==
    valueMaxCoPhieuAllYear)]
print("Ma co phieu ma luong nguoi mua cao nhat tat ca cac nam la: ")
data.result.i.12
```

- Kết quả chạy code:

```
[1] "Mã cổ phiếu mà lượng người mua cao nhất tất cả các năm là: "
> data.result.i.12
[1] VPB
```

Hình 14: *Mã cổ phiếu mà lượng người mua cao nhất tất cả các năm*

ii) Nhóm câu hỏi liên quan đến mô tả thống kê cơ bản dữ liệu

1) **Lập bảng mô tả số liệu thống kê tuổi cho từng năm những người trong top 200.**

- Trình bày cách làm:
  - Tạo dataframe *data.tuoi* gồm 9 cột, mỗi cột là một biến *Tuoi* của dữ liệu mỗi năm
  - Dùng lần lượt các hàm sau:
    - $apply(data.tuoi, 2, min)$ để xác định min cho từng cột trong *data.tuoi*
    - $apply(data.tuoi, 2, max)$ để xác định max cho từng cột trong *data.tuoi*
    - $apply(data.tuoi, 2, mean)$ để xác định Avg cho từng cột trong *data.tuoi*
    - $apply(data.tuoi, 2, sd)$ để xác định Std cho từng cột trong *data.tuoi*
    - $apply(data.tuoi, 2, quantile, 0.25)$ để xác định q1 cho từng cột trong *data.tuoi*
    - $apply(data.tuoi, 2, quantile, 0.5)$ để xác định q2 cho từng cột trong *data.tuoi*
    - $apply(data.tuoi, 2, quantile, 0.75)$ để xác định q3 cho từng cột trong *data.tuoi*
  - Tìm số outlier
    - Gán $iqr < -q3 - q1$
    - Gán $outlierMin < -q1 - 1.5 * iqr$
    - Gán $outlierMax < -q3 + 1.5 * iqr$
    - Dùng hàm $which()$ để tìm các index thỏa mãn cột tuổi của một năm nào đó lớn hơn giá trị $outlierMax$ hoặc nhỏ hơn giá trị $outlierMin$ và dùng hàm $length()$ để xác định số outliers
- Source code:

```
data.tuoi <- as.data.frame(matrix(nrow = 200, ncol = 9))
colnames(data.tuoi) <- c("2014", "2015", "2016", "2017", "2018", "2019", "2020",
    "2021", "2022")
for(i in 1:9){
  data.tuoi[,i] <- as.numeric(data.matrix[,index.tuoi[i]])
}
data.result.ii.1 <- as.data.frame(matrix(nrow = 9, ncol = 9))
colnames(data.result.ii.1) <- c("Years","Min","Q1","Q2","Q3","Max","Avg", "Std",
    "Outlier")
data.result.ii.1$Years <- c("2014", "2015", "2016", "2017", "2018", "2019", "2020",
    "2021", "2022")
data.result.ii.1$Min <- apply(data.tuoi, 2, min)
data.result.ii.1$Max <- apply(data.tuoi, 2, max)
data.result.ii.1$Avg <- apply(data.tuoi, 2, mean)
data.result.ii.1$Std <- apply(data.tuoi, 2, sd)
data.result.ii.1$Q1 <- apply(data.tuoi, 2, quantile, 0.25)
data.result.ii.1$Q2 <- apply(data.tuoi, 2, quantile, 0.5)
data.result.ii.1$Q3 <- apply(data.tuoi, 2, quantile, 0.75)
ii.1.tuoi.outlier <- c()
ii.1.iqr <- data.result.ii.1$Q3 - data.result.ii.1$Q1
ii.1.outlierMin <- data.result.ii.1$Q1 - 1.5 * ii.1.iqr
ii.1.outlierMax <- data.result.ii.1$Q3 + 1.5 * ii.1.iqr
for(i in 1:9){
  count <- length(which(data.tuoi[,i] < ii.1.outlierMin[i] | data.tuoi[,i] >
      ii.1.outlierMax[i]))
  ii.1.tuoi.outlier <- append(ii.1.tuoi.outlier, count)
}
data.result.ii.1$Outlier <- ii.1.tuoi.outlier
data.result.ii.1
```

- Kết quả chạy code:

	Years	Min	Q1	Q2	Q3	Max	Avg	Std	Outlier
1	2014	0	0	51.0	61	75	38.075	27.96605	0
2	2015	0	0	50.0	61	83	36.730	28.26609	0
3	2016	0	0	52.0	62	83	39.655	27.74391	0
4	2017	0	0	49.0	60	83	35.685	28.69114	0
5	2018	0	0	48.5	60	75	34.385	28.44048	0
6	2019	0	0	50.0	60	76	36.215	28.16514	0
7	2020	0	0	46.5	60	76	34.650	28.21859	0
8	2021	0	0	44.5	59	76	32.325	28.50389	0
9	2022	0	0	46.5	59	75	33.885	27.86529	0

Hình 15: *Bảng mô tả số liệu thống kê tuổi cho từng năm những người trong top 200*

2) **Lập bảng mô tả số liệu thống kê tuổi cho tập dữ liệu bao gồm tất cả các năm những người trong top 200.**

- Trình bày cách làm:
  - Dữ liệu sử dụng: *tuoi* (ở câu *i*)1)) là vector gồm tất cả biến *Tuoi* của tất cả các năm
  - Dùng lần lượt các hàm sau:
    - $min(tuoi)$ để xác định min của vector *tuoi*
    - $max(tuoi)$ để xác định max của vector *tuoi*
    - $mean(tuoi)$ để xác định Avg của vector *tuoi*
    - $sd(tuoi)$ để xác định Std của vector *tuoi*
    - $quantile(tuoi, 0.25)$ để xác định q1 của vector *tuoi*
    - $quantile(tuoi, 0.5)$ để xác định q2 của vector *tuoi*
    - $quantile(tuoi, 0.75)$ để xác định q3 của vector *tuoi*
  - Tìm số outlier
    - Gán $iqr < -q3 - q1$
    - Gán $outlierMin < -q1 - 1.5 * iqr$
    - Gán $outlierMax < -q3 + 1.5 * iqr$
    - Dùng hàm $which()$ để tìm các index thỏa mãn giá trị trong vector *tuoi* lớn hơn giá trị $outlierMax$ hoặc nhỏ hơn giá trị $outlierMin$
    - Dùng hàm $length()$ để xác định số outlier
- Source code:

```
data.result.ii.2 <- as.data.frame(matrix(nrow = 1, ncol = 8))
colnames(data.result.ii.2) <- c("Min","Q1","Q2","Q3","Max","Avg", "Std", "Outlier")
data.result.ii.2$Min <- min(tuoi)
data.result.ii.2$Max <- max(tuoi)
data.result.ii.2$Q1 <- quantile(tuoi, 0.25)
data.result.ii.2$Q2 <- quantile(tuoi, 0.5)
data.result.ii.2$Q3 <- quantile(tuoi, 0.75)
data.result.ii.2$Avg <- mean(tuoi)
data.result.ii.2$Std <- sd(tuoi)

ii.2.iqr <- data.result.ii.2$Q3 - data.result.ii.2$Q1
ii.2.outlierMin <- data.result.ii.2$Q1 - 1.5 * ii.2.iqr
ii.2.outlierMax <- data.result.ii.2$Q3 + 1.5 * ii.2.iqr
data.result.ii.2$Outlier <- length(which(tuoi < ii.2.outlierMin | tuoi >
    ii.2.outlierMax))
data.result.ii.2
```

- Kết quả chạy code:

	Min	Q1	Q2	Q3	Max	Avg	Std	Outlier
1	0	0	49	60	83	35.73389	28.22458	0

Hình 16: *Bảng mô tả số liệu thống kê tuổi cho tập dữ liệu bao gồm tất cả các năm những người trong top 200*

3) **Lập bảng mô tả số liệu thống kê giá trị tài sản cho từng năm những người trong top 200.**

- Trình bày các làm: Tương tự như câu *ii*)1)
- Source code:

```
index.taiSan <- c()
for(i in 0:8){
  index.taiSan <- append(index.taiSan, 7+6*i)
}
data.taiSan <- as.data.frame(matrix(nrow = 200, ncol = 9))
colnames(data.taiSan) <- c("2014", "2015", "2016", "2017", "2018", "2019", "2020",
    "2021", "2022")
for(i in 1:9){
  data.taiSan[,i] <- as.numeric(data.new[,index.taiSan[i]])
}

data.result.ii.3 <- as.data.frame(matrix(nrow = 9, ncol = 9))
colnames(data.result.ii.3) <- c("Years","Min","Q1","Q2","Q3","Max","Avg", "Std",
    "Outlier")
data.result.ii.3$Years <- c("2014", "2015", "2016", "2017", "2018", "2019", "2020",
    "2021", "2022")
data.result.ii.3$Min <- apply(data.taiSan, 2, min)
data.result.ii.3$Max <- apply(data.taiSan, 2, max)
data.result.ii.3$Avg <- apply(data.taiSan, 2, mean)
data.result.ii.3$Std <- apply(data.taiSan, 2, sd)
data.result.ii.3$Q1 <- apply(data.taiSan, 2, quantile, 0.25)
data.result.ii.3$Q2 <- apply(data.taiSan, 2, quantile, 0.5)
data.result.ii.3$Q3 <- apply(data.taiSan, 2, quantile, 0.75)

ii.3.taiSan.outlier <- c()
ii.3.iqr <- data.result.ii.3$Q3 - data.result.ii.3$Q1
ii.3.outlierMin <- data.result.ii.3$Q1 - 1.5 * ii.3.iqr
ii.3.outlierMax <- data.result.ii.3$Q3 + 1.5 * ii.3.iqr
for(i in 1:9){
  count <- length(which(data.taiSan[,i] < ii.3.outlierMin[i] | data.taiSan[,i] >
      ii.3.outlierMax[i]))
}
data.result.ii.3$Outlier <- ii.3.taiSan.outlier
data.result.ii.3
```

- Kết quả chạy code:

	Years	Min	Q1	Q2	Q3	Max	Avg	Std	Outlier
1	2014	68	98.25	158.5	352.00	20400	492.025	1623.734	23
2	2015	87	117.00	190.5	334.75	24332	495.325	1805.983	26
3	2016	112	148.75	208.5	473.50	33806	852.985	3319.054	22
4	2017	169	243.00	384.0	927.75	56703	1517.120	5548.912	22
5	2018	243	346.00	532.0	1300.00	177752	2451.795	12880.313	21
6	2019	226	341.75	578.0	1346.25	214496	2634.455	15419.315	20
7	2020	293	425.25	819.0	1806.75	207926	3100.605	15159.705	17
8	2021	710	1041.00	1668.0	3654.00	205027	5106.745	16157.675	23
9	2022	647	914.50	1456.0	3116.50	167515	4455.485	13454.085	27

Hình 17: *Bảng mô tả số liệu thống kê giá trị tài sản cho từng năm những người trong top 200*

4) **Lập bảng mô tả số liệu thống kê giá trị tài sản cho tập dữ liệu bao gồm tất cả các năm những người trong top 200.**

- Trình bày cách làm: Tương tự câu *ii*)2)
- Source code:

```
taiSan <- as.numeric(taiSan)

data.result.ii.4 <- as.data.frame(matrix(nrow = 1, ncol = 8))
colnames(data.result.ii.4) <- c("Min","Q1","Q2","Q3","Max","Avg", "Std", "Outlier")
data.result.ii.4$Min <- min(taiSan)
data.result.ii.4$Max <- max(taiSan)
data.result.ii.4$Q1 <- quantile(taiSan, 0.25)
data.result.ii.4$Q2 <- quantile(taiSan, 0.5)
data.result.ii.4$Q3 <- quantile(taiSan, 0.75)
data.result.ii.4$Avg <- mean(taiSan)
data.result.ii.4$Std <- sd(taiSan)

ii.4.iqr <- data.result.ii.4$Q3 - data.result.ii.4$Q1
ii.4.outlierMin <- data.result.ii.4$Q1 - 1.5 * ii.4.iqr
ii.4.outlierMax <- data.result.ii.4$Q3 + 1.5 * ii.4.iqr
data.result.ii.4$Outlier <- length(which(taiSan < ii.4.outlierMin | taiSan >
    ii.4.outlierMax))
data.result.ii.4
```

- Kết quả chạy code:

```
  Min  Q1  Q2      Q3    Max      Avg      Std Outlier
1  68 253 567 1404.75 214496 2345.171 11258.54     201
```

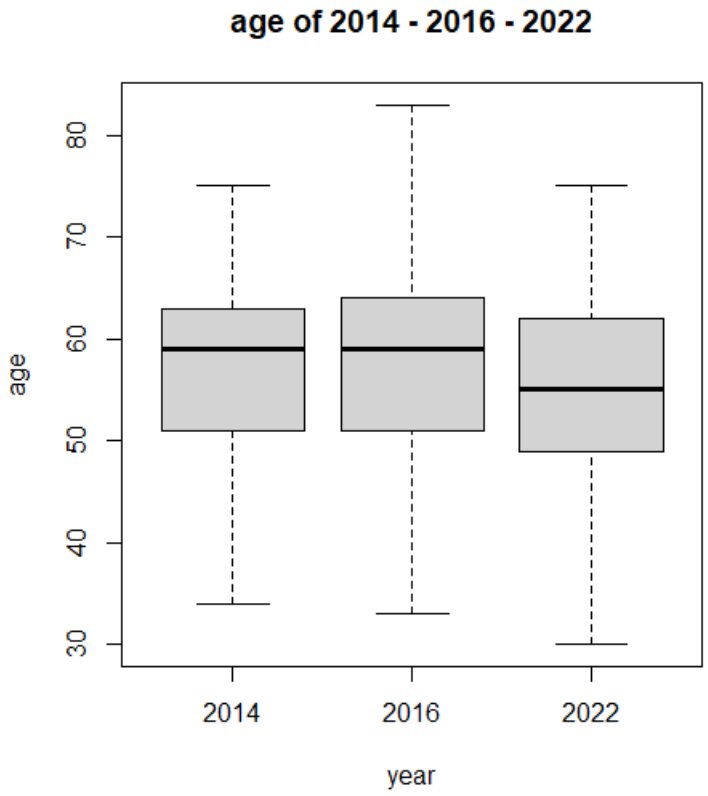
Hình 18: *Bảng mô tả số liệu thống kê giá trị tài sản cho dữ liệu gồm tất cả các năm những người trong top 200*

5) **Vẽ biểu đồ boxplot hay còn được gọi là box-and-whisker cho các năm 1, 3, 9 của tuổi.**

- Trình bày cách làm:
  - Từ khung dữ liệu data.tuoi, ta muốn tạo một list có tên là AGE chứa tất cả các vector của data.tuoi mà trong đó không có vector nào chứa tuổi 0. Để làm điều này ta phải xử lý lần lượt từng vector
  - Sau khi có được AGE, gọi lệnh boxplot() để vẽ biểu đồ hộp.
- Source code:

```
age.2014 <- data.tuoi[which(data.tuoi$`2014` > 0), 1]
age.2015 <- data.tuoi[which(data.tuoi$`2015` > 0), 2]
#Tuong tu cho cac nam con lai

AGE <- list(
  "2014" = age.2014, "2015" = age.2015, "2016" = age.2016,
  "2017" = age.2017, "2018" = age.2018, "2019" = age.2019,
  "2020" = age.2020, "2021" = age.2021, "2022" = age.2022
)
boxplot(c(AGE[1], AGE[3], AGE[9]), main = "age of 2014 - 2016 - 2022",
          ylab = "age", xlab = "year")
```

- Kết quả chạy code:



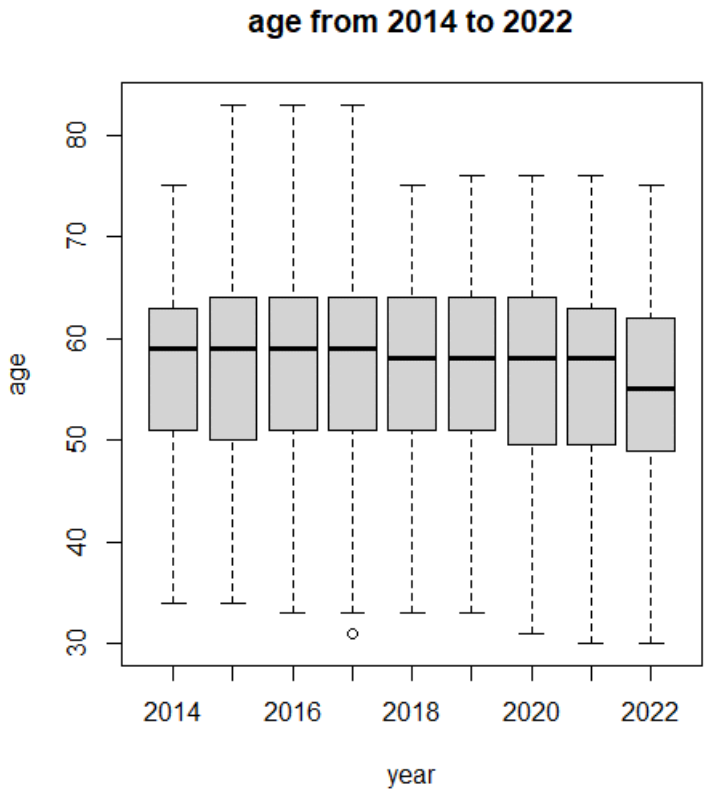
Hình 19: *Biểu đồ boxplot cho các năm 2014, 2016, 2022 của tuổi*

6) **Vẽ biểu đồ boxplot hay còn được gọi là box-and-whisker cho tất cả các năm của tuổi.**

- Trình bày cách làm:
  - Dùng hàm boxplot() và dựa vào dữ liệu AGE, ta có thể vẽ biểu đồ cột cho tất cả các năm từ 2014 đến 2022
- Source code:

```
boxplot(AGE, main = "age from 2014 to 2022",
          xlab = "year", ylab = "age")
```

- Kết quả chạy code:



Hình 20: *Biểu đồ boxplot cho tất cả các năm của tuổi*

- Nhận xét:
  - Quan sát tất cả các năm, ta thấy rằng tuổi trung bình của nhà đầu tư giảm rất nhẹ và tập trung ở độ tuổi 59, độ dao động của tuổi (khoảng cách giữa 2 điểm tứ phân vị) gần như tương đồng nhau từ 50 đến 63 tuổi.
  - Tuy nhiên, độ tuổi nhỏ nhất và độ tuổi lớn nhất của các nhà đầu tư có sự thay đổi tương đối rõ rệt. Càng về sau, độ tuổi nhỏ nhất và lớn nhất càng ngày càng giảm cho thấy số lượng các nhà đầu tư trẻ đang có dấu hiệu tăng. Riêng năm 2022 là năm có nhiều nhà đầu tư trẻ nhất (từ 30 đến 75 tuổi) và trẻ hơn hẳn các năm từ 2017 trở về trước (từ 35 đến hơn 80 tuổi), chính vì có nhiều nhà đầu tư trẻ khiến cho độ tuổi trung bình của năm 2022 giảm xuống khoảng 55 tuổi và là cột mốc đánh dấu sự trẻ hóa của các nhà đầu tư
  - Dù vậy, phần lớn trong 200 người giàu nhất thị trường chứng khoán, các nhà đầu tư đều tập trung ở tuổi 59 và phần ít các nhà đầu tư dưới tuổi 40. Sở dĩ điều này xảy ra là vì các nhà đầu tư trung niên là những người lão làng, có nhiều năm kinh nghiệm trong thị trường chứng khoán, họ có đủ kiến thức để bảo toàn tài sản khi khủng hoảng, hoặc gia tăng tài sản khi có cơ hội. Ngược lại đối với những nhà đầu tư trẻ tuổi họ đều rất non nớt, họ không biết cách đối phó với các biến động, dẫn đến phần lớn đều thua lỗ và có số

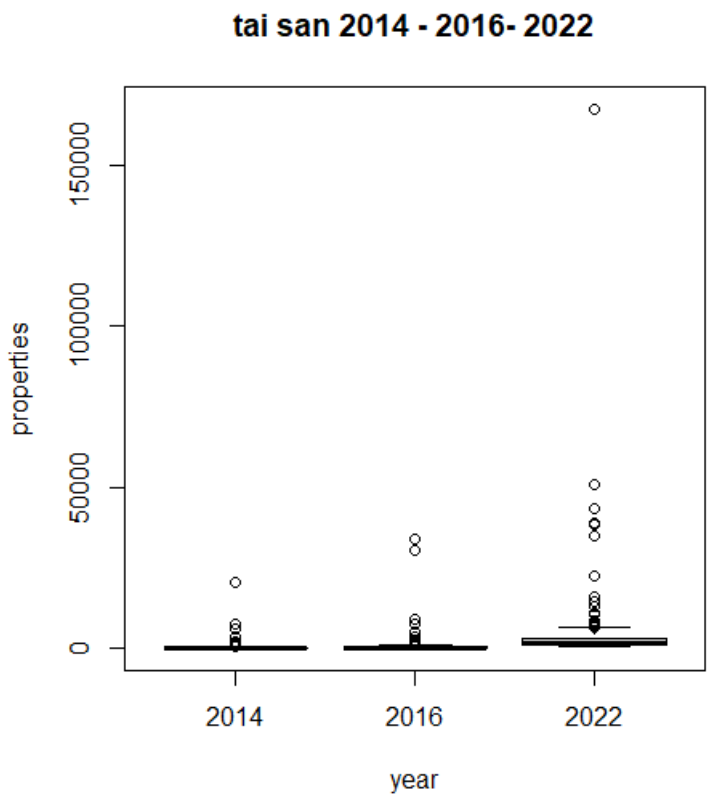
ít đủ sự thông minh để vượt lên trở thành những nhà đầu tư giàu có nhất. Nhưng, các năm về sau, các nhà đầu tư trẻ dần khôn ngoan hơn và dần thế chân các bô lão trong danh sách những nhà đầu tư giàu nhất nước ta. Minh chứng rõ ràng nhất chính là năm 2022.

7) **Vẽ biểu đồ boxplot hay còn được gọi là box-and-whisker cho các năm 1, 3, 9 của giá trị tài sản.**

- Trình bày cách làm:
  - Dựa vào data.taisan của phần trên kết hợp hàm boxplot(), ta sẽ vẽ được biểu đồ hộp của 3 năm 2014 – 2016 – 2022.
- Source code:

```
boxplot (c(data.taiSan[1], data.taiSan[3], data.taiSan[9]),
          main = "tai san 2014 - 2016- 2022",
          xlab = "year" , ylab = "properties")
```

- Kết quả chạy code:



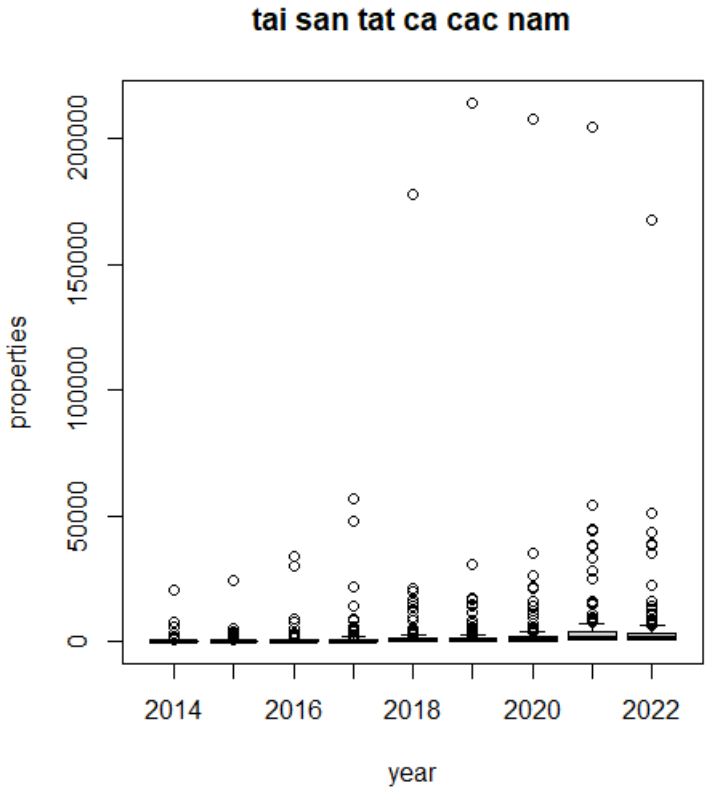
Hình 21: *Biểu đồ boxplot cho các năm 2014, 2016, 2022 của giá trị tài sản*

8) **Vẽ biểu đồ boxplot hay còn được gọi là box-and-whisker cho tất cả các năm của giá trị tài sản.**

- Trình bày cách làm:
  - Dựa vào data.taisan của phần trên kết hợp hàm boxplot(), ta sẽ vẽ được biểu đồ hộp của tất cả các năm từ 2014 đến 2022.
- Source code:

```
boxplot(data.taiSan, main = "tai san tat ca cac nam" ,
          xlab = "year" , ylab = "properties")
```

- Kết quả chạy code:



Hình 22: *Biểu đồ boxplot cho tất cả các năm của giá trị tài sản*

iii) Nhóm câu hỏi liên quan đến trực quan dữ liệu tuổi

Mã ký số: 9367

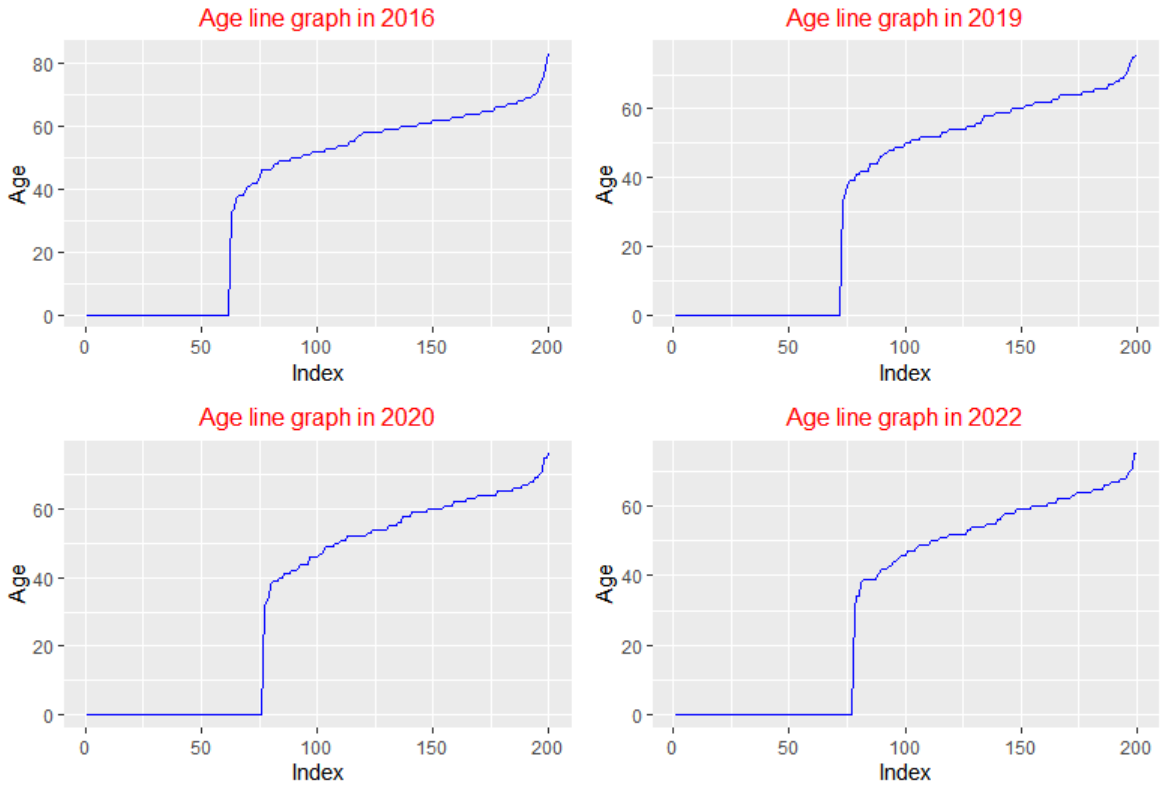
Trong bài này chúng ta sẽ tính toán và vẽ cái đồ thị liên quan đến trực quan dữ liệu tuổi.

1) **Vẽ 4 biểu đồ thể hiện đường tuổi cho 4 năm.**

- Trình bày cách làm:
  - Lấy dữ liệu tuổi từ dataframe chung mà chúng ta đã tính toán ở những câu trước.
  - Ta nhận thấy các dữ liệu tuổi chúng ta vừa lấy vào thuộc dạng kí tự. Vì vậy chúng ta cần chuyển những dữ liệu này thành dạng số bằng hàm *as.numeric()*
  - Sắp xếp các dữ liệu tuổi theo giá trị từ bé đến lớn bằng hàm *sort()* để quan sát đồ thị dễ dàng hơn.
  - Sử dụng thư viện *ggplot2* để vẽ đồ thị.
- Source code:

```
data2014['Tuoi'] <- data.new[index.tuoi[1]] #2014 //1
data2015['Tuoi'] <- data.new[index.tuoi[2]] #2015 //2
#Tuong tu cho cac nam 2016, 2017, ...

data2014$Tuoi <- as.numeric(data2014$Tuoi)
data2015$Tuoi <- as.numeric(data2015$Tuoi)
#Tuong tu cho cac nam 2016, 2017

#Draw age line graph for 4 years
plot_line = function(dataframe,a,b){
  ggplot(data = dataframe,aes(x= a,y= col,group = 1)) + geom_line(aes(y =
      sort(b)),color="blue") +labs(x="Index", y = "Age")
}
line_2016 <- plot_line(data2016,data2016$Vitri,data2016$Tuoi) +
  ggtitle("Age line graph in 2016") + theme(plot.title = element_text(hjust = 0.5,
      color = "red",size = 12))
line_2019 <- plot_line(data2019,data2019$Vitri,data2019$Tuoi) +
  ggtitle("Age line graph in 2019") + theme(plot.title = element_text(hjust = 0.5,
      color = "red",size = 12))
# Tuong tu voi nam 2020, 2022.
grid.arrange(line_2016,line_2019,line_2020,line_2022)
```

- Biểu đồ:



Hình 23: *Biểu đồ thể hiện đường tuổi của 4 năm 2016, 2019, 2020 ,2022.*

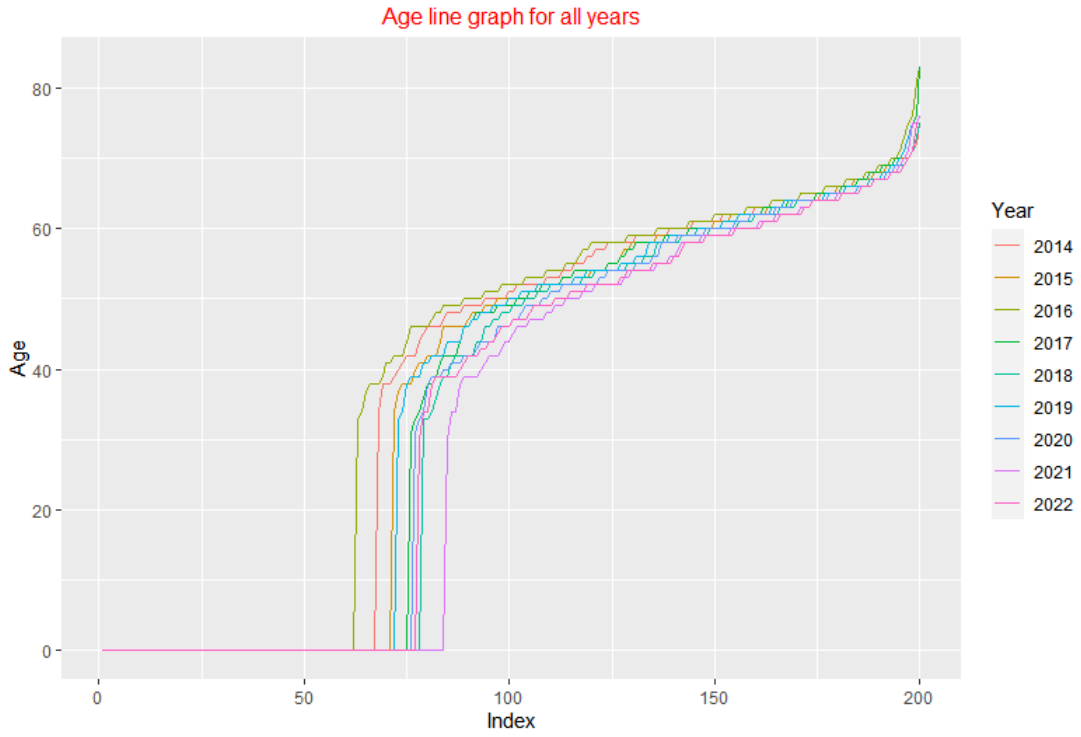
- Nhận xét biểu đồ : Trong biểu đồ có một lượng lớn giá trị tuổi chúng ta chưa biết ('0').

2) **Biểu đồ đường tuổi gồm tất cả các năm.**

- Trình bày cách làm:
  - Để biểu đồ dễ quan sát, chúng ta sắp xếp các giá trị tuổi theo thứ tự từ bé đến lớn bằng hàm *sort()* rồi vẽ biểu đồ.
- Source code:

```
ggplot(data = data2014,aes(x=Vitri,group = 1)) +
geom_line(aes(y=sort(Tuoi),color="2014")) +
geom_line(aes(y=sort(data2015$Tuoi), color="2015")) +
geom_line(aes(y= sort(data2016$Tuoi),color="2016")) +
geom_line(aes(y=sort(data2017$Tuoi),color="2017")) +
geom_line(aes(y=sort(data2018$Tuoi),color="2018")) +
geom_line(aes(y=sort(data2019$Tuoi), color="2019")) +
geom_line(aes(y= sort(data2020$Tuoi),color="2020")) +
geom_line(aes(y=sort(data2021$Tuoi),color="2021")) +
geom_line(aes(y=sort(data2022$Tuoi),color="2022")) +
labs(x="Index", y= "Age" , colour="Year" ) +
ggtitle("Age line graph for all years") + theme(plot.title = element_text(hjust =
    0.5, color = "red",size = 12))
```

- Biểu đồ:



Hình 24: *Biểu đồ đường tuổi tất các các năm*

- **Nhận xét:** Chúng ta có thể thấy được năm 2017 là năm chứa giá trị tuổi lớn nhất.

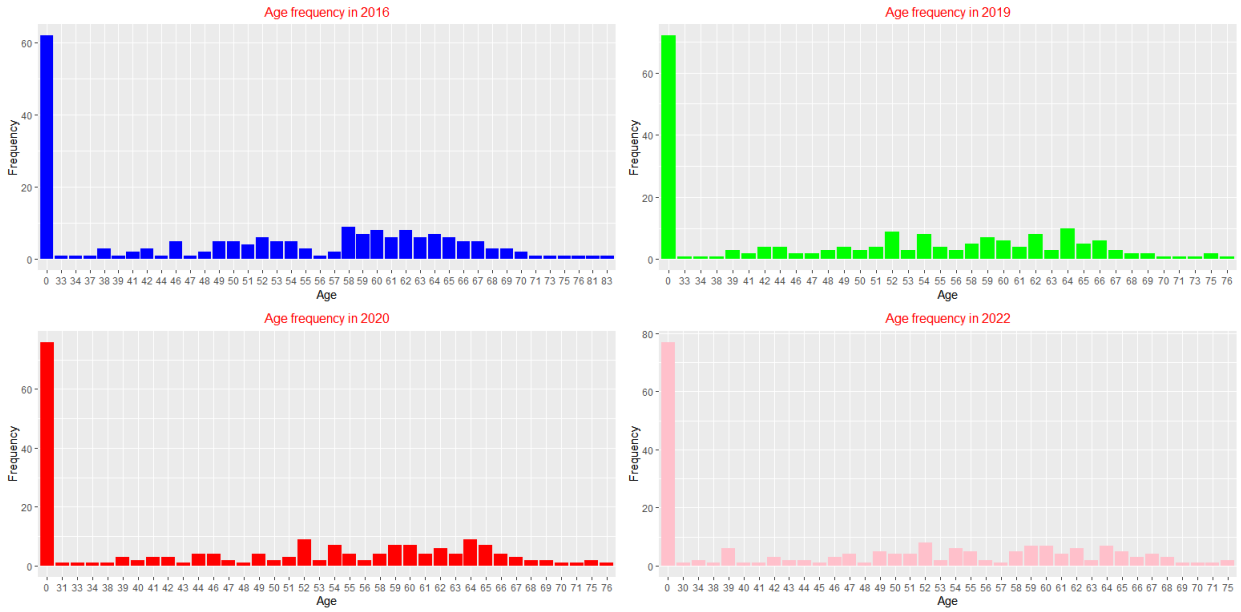
3) **Vẽ 4 biểu đồ thể hiện tần số tuổi cho 4 năm.**

- Trình bày cách làm:
  - **Tần số** là số lần xuất hiện của mỗi giá trị trong mẫu số liệu (Kí hiệu là n).
  - Để xác định tần số tuổi của mỗi năm chúng ta sử dụng hàm *table()* để lập bảng tần số.
- Soure code:

```
#Tan so cua Tuoi
#2016
freq2016 <- as.data.frame(table(data2016$Tuoi))
colnames(freq2016)[1] <- c("Tuoi")
#2019
freq2019 <- as.data.frame(table(data2019$Tuoi))
colnames(freq2019)[1] <- c("Tuoi")
#2020
# Tuong tu cho nam 2020, 2022

# bieu do tan so
#2016
chart2016 <- ggplot(freq2016,aes(x = Tuoi, y = Freq)) + geom_bar(stat = "identity",
    fill = "blue") + labs(x="Age", y = "Frequency") +
  ggtitle("Age frequency in 2016") + theme(plot.title = element_text(hjust = 0.5,
      color = "red",size = 12))
#2019
chart2019 <- ggplot(freq2019,aes(x = Tuoi, y = Freq)) + geom_bar(stat = "identity",
    fill ="green") + labs(x= "Age", y = "Frequency") +
  ggtitle("Age frequency in 2019") + theme(plot.title = element_text(hjust = 0.5,
      color = "red",size = 12))
#Tuong tu voi nam 2020, 2022

```

```
20 grid.arrange(chart2016,chart2019,chart2020,chart2022)
```

- Biểu đồ:



Hình 25: *Biểu đồ thể hiện tần số tuổi qua các năm*

- **Nhận xét:** Thông qua biểu đồ thể hiện tần số tuổi ta có thể thấy được:
  - Tần số tuổi ở năm 2016 lớn nhất ở 58 tuổi và tần số tuổi dao động từ 0 đến 10 (người / tuổi)
  - Tần số tuổi ở năm 2019 lớn nhất ở 64 tuổi và tần số tuổi dao động từ 0 đến 10 (người / tuổi)
  - Tần số tuổi ở năm 2020 lớn nhất ở 52 tuổi và tần số tuổi dao động từ 0 đến 10 (người / tuổi)
  - Tần số tuổi ở năm 2022 lớn nhất ở 52 tuổi và tần số tuổi dao động từ 0 đến 10 (người / tuổi)

4) **Vẽ 4 biểu đồ thể hiện tần số tích lũy tuổi cho 4 năm.**

- Trình bày cách làm:
  - **Tần số tích lũy:** Tần số tích lũy là tổng các tần số tuyệt đối. Kết quả cuối cùng của tần số tích lũy phải là độ rộng của mẫu mà chúng ta thống kê.
    Công thức:

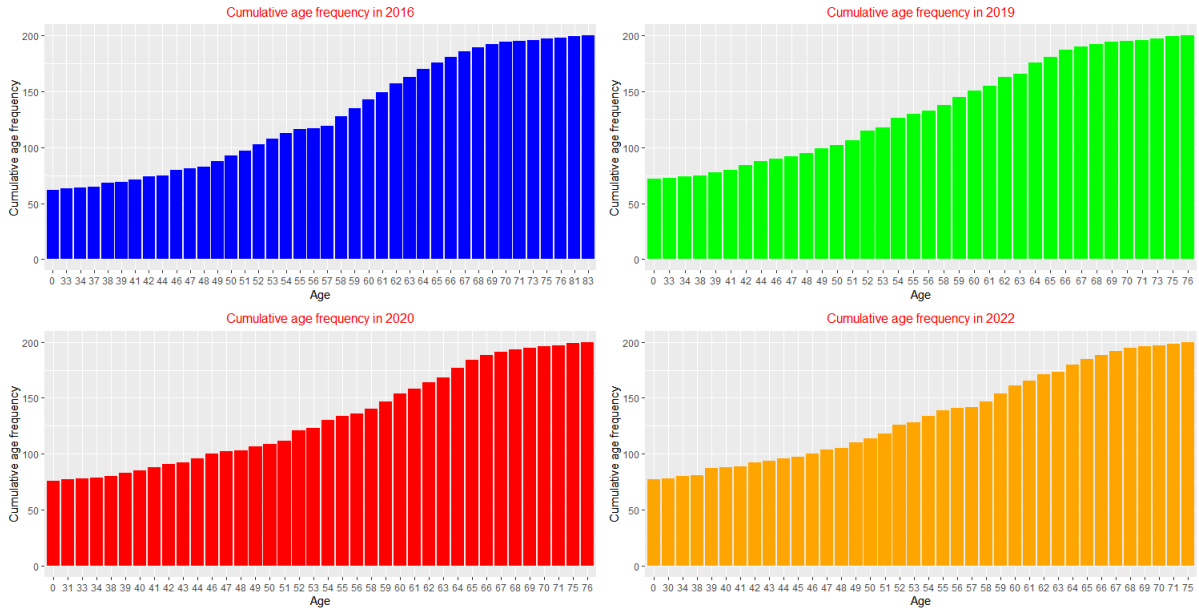
$$number = \sum_{i=1}^{N}(number)$$

  - Để tính tần số tích lũy tuổi, chúng ta sử dụng hàm $cumSum()$.
  - Sau đó thêm tần số tích lũy tuổi vừa tính toán được vào dataframe bằng hàm $tranform()$.

- Source code:

```
#Tan so tich luy tuoi
freq2016 <- transform(freq2016, cumFreq= cumsum(Freq)) #2016
freq2019 <- transform(freq2019, cumFreq= cumsum(Freq)) #2019
#Tuong tu voi nam 2020, 2022.

#Bieu do tan so tich luy

#2016
cumFreq_2016 <- ggplot(freq2016,aes(x = Tuoi, y = cumFreq)) + geom_bar(stat =
    "identity", fill = "blue") + labs(x= "Age", y = "Cumulative age frequency") +
  ggtitle("Cumulative age frequency in 2016") + theme(plot.title =
      element_text(hjust = 0.5, color = "red",size = 12))
#2019
cumFreq_2019 <- ggplot(freq2019,aes(x = Tuoi, y = cumFreq)) + geom_bar(stat =
    "identity", fill = "green") + labs(x= "Age", y = "Cumulative age frequency") +
  ggtitle("Cumulative age frequency in 2019") + theme(plot.title =
      element_text(hjust = 0.5, color = "red",size = 12))
#Tuong tu voi nam 2020, 2022

grid.arrange(cumFreq_2016,cumFreq_2019,cumFreq_2020,cumFreq_2022)
```

- Biểu đồ:



Hình 26: *Biểu đồ thể hiện tần số tích lũy tuổi trong 4 năm 2016, 2019, 2020, 2022.*

5) **Vẽ 4 biểu đồ thể hiện tần số tương đối tuổi cho 4 năm.**

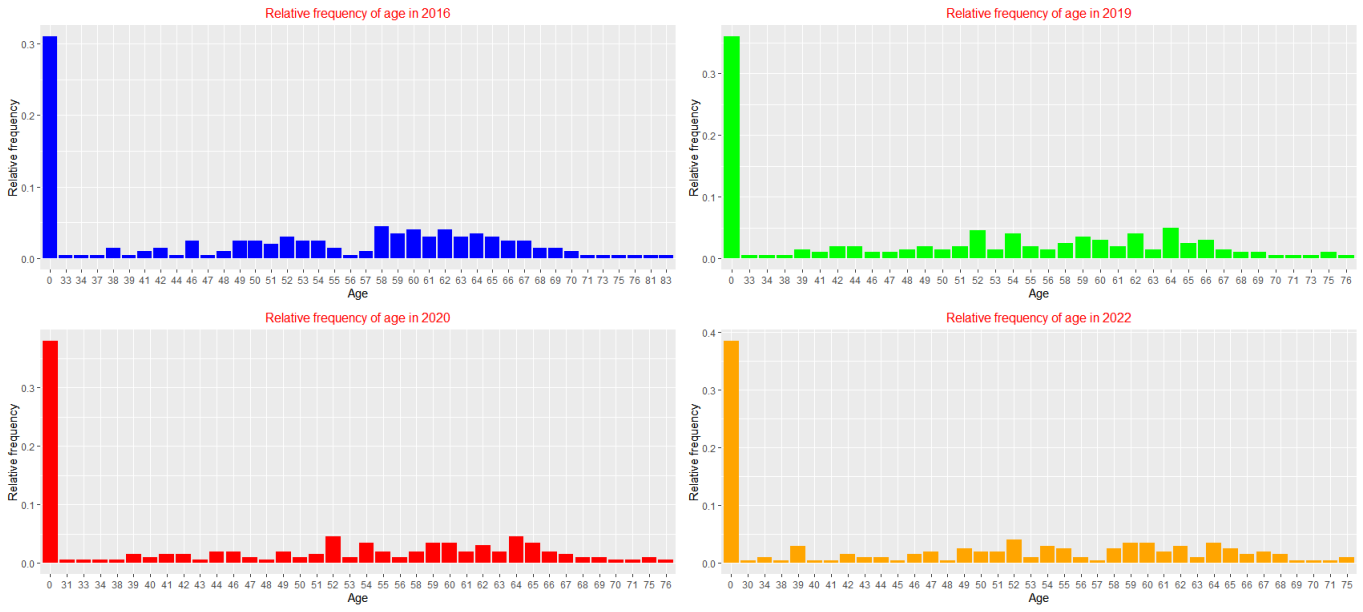
- Trình bày cách làm:
  - **Tần số tương đối:** Thu được bằng cách chia tần số tuyệt đối cho tổng dữ liệu N. Công thức:

$$f = \frac{number}{\sum_{i=1}^{N}(number)}$$

  - Để tính tần số tương đối, chúng ta sử dụng hàm *prop.table()*.
- Source code:

```
#Tan so tuong doi
freq2016 <- transform(freq2016,relative = prop.table(Freq)) #2016
freq2019 <- transform(freq2019,relative = prop.table(Freq)) #2019
#Tuong tu voi nam 2020, 2022.

#Bieu do tan so tuong doi
#2016
relative_2016 <- ggplot(freq2016,aes(x = Tuoi, y = relative)) + geom_bar(stat =
    "identity", fill = "blue") + labs(x= "Age", y = "Relative frequency") +
  ggtitle("Relative frequency of age in 2016") + theme(plot.title =
      element_text(hjust = 0.5, color = "red",size = 12))
#2019
relative_2019 <- ggplot(freq2019,aes(x = Tuoi, y = relative)) + geom_bar(stat =
    "identity", fill = "green") + labs(x= "Age", y = "Relative frequency") +
  ggtitle("Relative frequency of age in 2019") + theme(plot.title =
      element_text(hjust = 0.5, color = "red",size = 12))
#Tuong tu voi nam 2020, 2022.

grid.arrange(relative_2016,relative_2019,relative_2020,relative_2022)
```

- Biểu đồ:



Hình 27: *Biểu đồ cột thể hiện tần số tương đối của tuổi trong 4 năm 2016, 2019, 2020, 2022.*

- **Nhận xét:**
  - Tần số tương đối tuổi ở năm 2016 lớn nhất ở 58 tuổi và tần số tương đối tuổi dao động từ 0 đến 0.05.
  - Tần số tuổi ở năm 2019 lớn nhất ở 64 tuổi và tần số tuổi dao động từ 0 đến 0.05.
  - Tần số tuổi ở năm 2020 lớn nhất ở 52 tuổi và tần số tuổi dao động từ 0 đến 0.05.
  - Tần số tuổi ở năm 2022 lớn nhất ở 52 tuổi và tần số tuổi dao động từ 0 đến 0.05

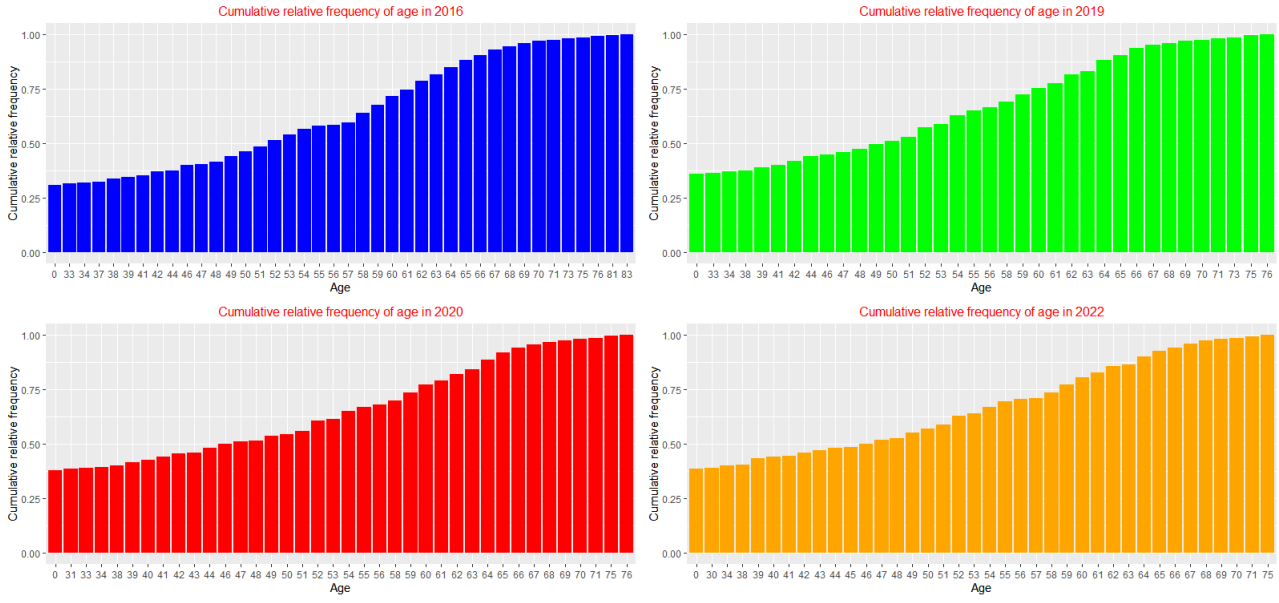
6) **Vẽ 4 biểu đồ thể hiện tần số tương đối tích lũy tuổi cho 4 năm.**

- Trình bày cách làm:
  - **Tần số tương đối tích lũy:** Là tần số tích lũy của tần số tương đối. Kết quả cuối cùng phải bằng 1.
  - Để tính tần số tương đối tích lũy tuổi. Chúng ta sử dụng hàm *cumSum()* với dữ liệu đầu vào là tần số tương đối.

- Soure code:

```
#Tan so tuong doi tich luy

freq2016 <- transform(freq2016, cumrelative = cumsum(relative))
freq2019 <- transform(freq2019, cumrelative = cumsum(relative))
#Tuong tu voi nam 2020, 2022.

#Bieu do tan so tuong doi tich luy

#2016
cumrelative_2016 <- ggplot(freq2016,aes(x = Tuoi, y = cumrelative)) + geom_bar(stat
    = "identity", fill = "blue") + labs(x= "Age", y = "Cumulative relative
    frequency") +
  ggtitle("Cumulative relative frequency of age in 2016") + theme(plot.title =
      element_text(hjust = 0.5, color = "red",size = 12))
#2019
cumrelative_2019 <- ggplot(freq2019,aes(x = Tuoi, y = cumrelative)) + geom_bar(stat
    = "identity", fill = "green") + labs(x= "Age", y = "Cumulative relative
    frequency") +
  ggtitle("Cumulative relative frequency of age in 2019") + theme(plot.title =
      element_text(hjust = 0.5, color = "red",size = 12))
# Tuong tu voi nam 2020,2022

grid.arrange(cumrelative_2016,cumrelative_2019,cumrelative_2020,cumrelative_2022)
```

- Biểu đồ:



Hình 28: *Biểu đồ thể hiện tần số tương đối tích lũy của tuổi trong 2016, 2019, 2020, 2022.*

7) **Vẽ biểu đồ thể hiện min, mean, max tuổi cho 4 năm.**

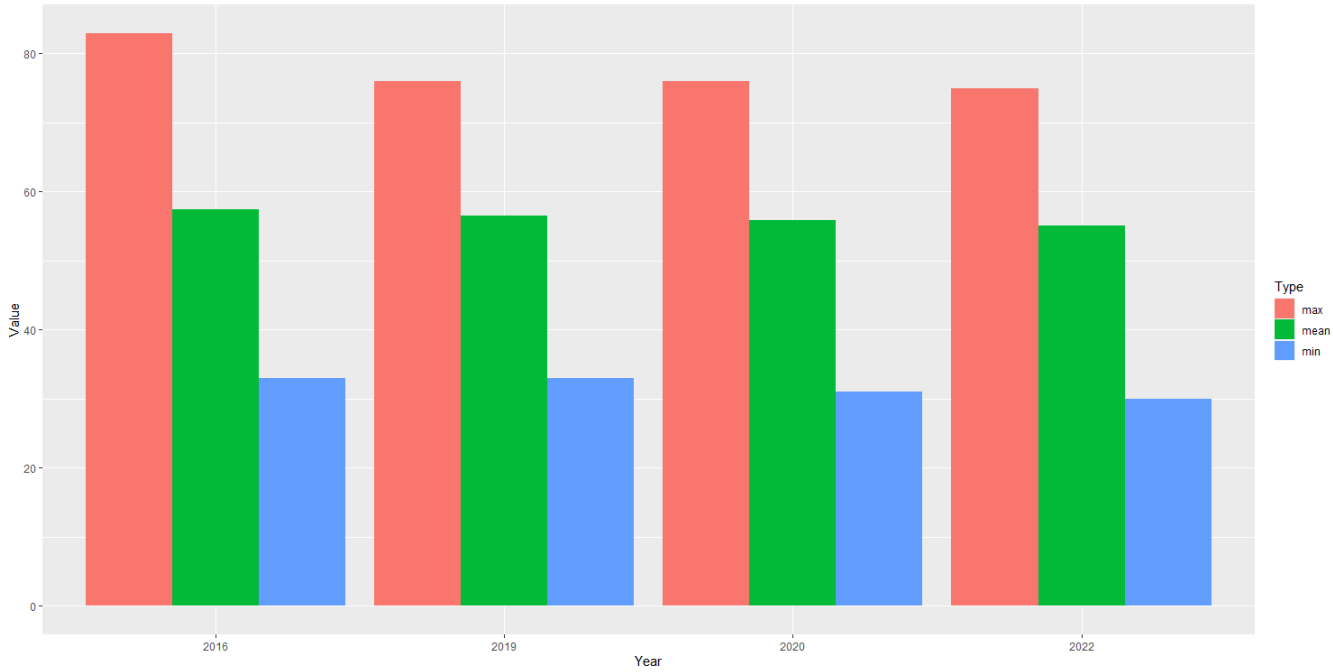
- Trình bày cách làm:
  - Min là giá trị tuổi nhỏ nhất.
  - Max là giá trị tuổi lớn nhất.
  - Mean là giá trị tuổi trung bình
  - Xử lí những giá trị không xác định mà ta đã gán bằng giá trị "0" ở những câu trước bằng cách xóa bỏ chúng.

- Tạo dataframe trong đó chứa các dữ liệu "mean", "max" , "min" của 4 năm.
- Vẽ biểu đồ dạng cột thể hiện giá trị min, mean , max trong 4 năm 2016, 2019, 2020, 2022.

- Source code :

```
#Xu li du lieu
tuoi2014 <- data2014$Tuoi[data2014$Tuoi!="0"] #delete value unknown
tuoi2015 <- data2015$Tuoi[data2015$Tuoi!="0"] #delete value unknown
#Tuong tu voi nhung nam con lai

#Tao dataframe chua nhung gia tri max min mean
chart <- data.frame("Year" = rep(c("2016","2019","2020","2022"), each=3),
                    "Type" = c("mean","min","max","mean","min",
                    "max","mean","min","max","mean","min","max"),
                    "Value" = c(mean(tuoi2016), min(tuoi2016),max(tuoi2016),
                                mean(tuoi2019), min(tuoi2019), max(tuoi2019),
                                mean(tuoi2020), min(tuoi2020),max(tuoi2020),
                                mean(tuoi2022), min(tuoi2022), max(tuoi2022)))

#Ve bieu do
ggplot(chart, aes(fill=Type, y=Value, x=Year)) +
  geom_bar(position='dodge', stat='identity')
```

- Biểu đồ:



Hình 29: *Biểu đồ thể hiện giá trị min, mean, max trong 4 năm 2016, 2019, 2020, 2022*

- **Nhận xét:** Nhìn vào biểu đồ, ta thấy:
  - Năm 2016 có giá trị tuổi lớn nhất trong 4 năm.
  - Năm 2016 có giá trị tuổi trung bình cao nhất trong 4 năm.
  - Năm 2019 có giá trị tuổi nhỏ nhất trong 4 năm.

8) **Vẽ phổ tuổi outliers xuất hiện cho tất cả các năm.**

- Trình bày cách làm:
  - Sử dụng cách làm tương tự câu *ii*)2) (tìm số outliers cho một vector), ta áp dụng cho các vector *tuoi*2014, *tuoi*2015, ..., *tuoi*2022

– Vẽ đồ thị dạng cột số outliers xuất hiện trong tất cả các năm.

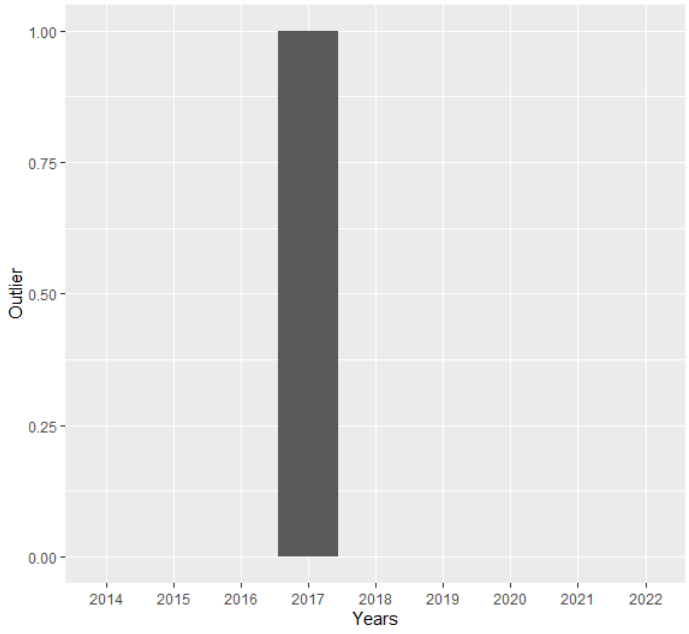
- Soure code:

```
data.outlier <- as.data.frame(matrix(nrow = 9, ncol =5))
colnames(data.outlier) <- c("Years", "Q1", "IQR","Q3", "Outlier")
data.outlier$Years <-
    c("2014","2015","2016","2017","2018","2019","2020","2021","2022")
data.outlier
#ve do thi
ggplot(data.outlier,aes(y = Outlier, x = Years)) + geom_bar(stat = "identity")
```

```
> data.outlier
  Years    Q1   IQR    Q3 Outlier
1  2014 51.00 12.00 63.00       0
2  2015 50.00 14.00 64.00       0
3  2016 51.25 12.75 64.00       0
4  2017 51.00 13.00 64.00       1
5  2018 51.25 12.50 63.75       0
6  2019 51.00 13.00 64.00       0
7  2020 49.75 14.25 64.00       0
8  2021 49.75 13.25 63.00       0
9  2022 49.00 13.00 62.00       0
```

Hình 30: *Số outliers xuất hiện trong tất cả các năm.*

- Biểu đồ:



Hình 31: *Biểu đồ thể hiện số outliers xuất hiện trong tất cả các năm.*

- Nhận xét: Nhìn vào đồ thị ta thấy được chỉ có năm 2017 xuất hiện giá trị ngoại lai với số lượng là 1.

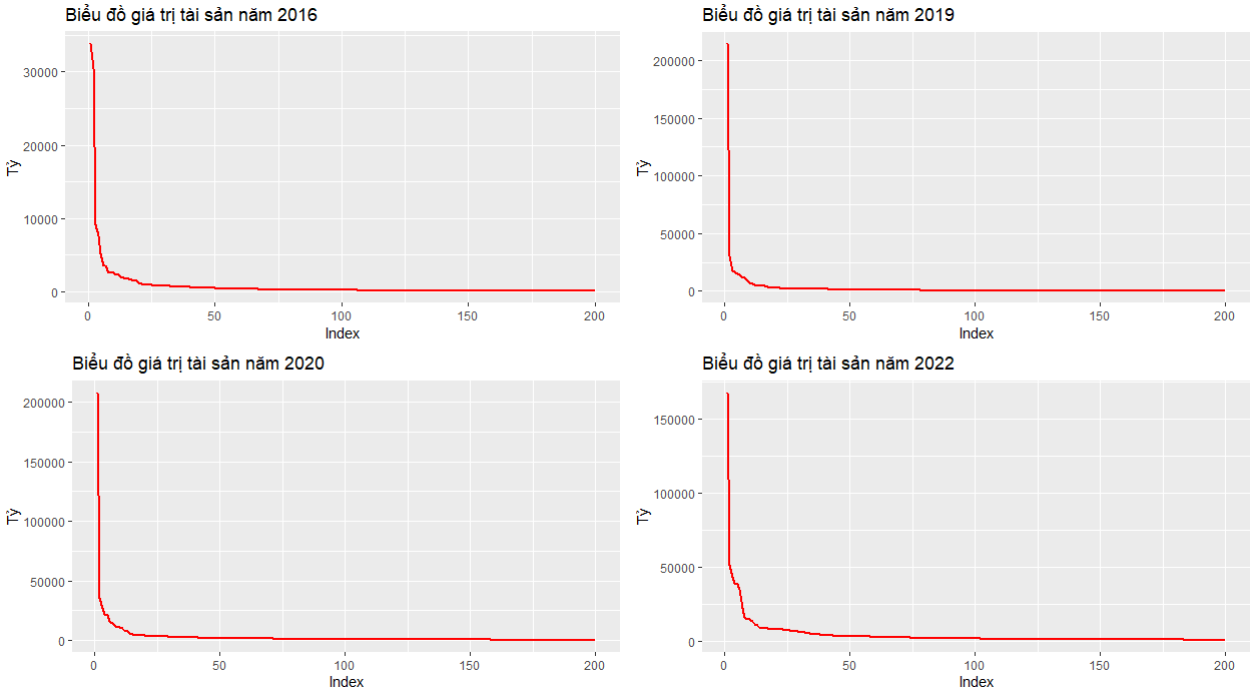
iv) Nhóm câu hỏi liên quan đến trực quan dữ liệu giá trị tài sản
Mã đề: 9367

1 **Vẽ 4 biểu đồ thể hiện đường giá trị tài sản cho 4 năm**

- Trình bày cách làm:
  - Dữ liệu tài sản sau khi xử lí đã được bỏ vào một data.frame là : *data.Taisan* (từ câu *ii*)3)). Từ bảng dữ liệu này ta lấy dữ liệu từ trong bảng này bỏ lại vào dữ liệu data mỗi năm ban đầu sau đó sử dụng dữ liệu ở chính dữ liệu ban đầu đó.
  - Sau đó sử dụng một hàm để vẽ biểu đồ tài sản cho 4 năm, dùng ggplot kiểu biểu đồ đường để thể hiện tài sản của 4 năm :2016,2019,2020,2022
  - Truyền dữ liệu vào hàm sau đó dùng grid.arrange để vẽ đồ thị tài sản của 4 năm trong cùng 1 bảng như hình vẽ.
- Source code

```
#lay du lieu trong data.taiSan dua vao du lieu ban dau
data2014['GiatriTaiSan'] <- data.taiSan[1]
data2015['GiatriTaiSan'] <- data.taiSan[2]
#Tuong tu cho cac nam con lai tu 2016, 2017, ..., 2022
# Ve bieu do

plot_line = function(dataframe,a,b){
  ggplot(data = dataframe, aes(x= a,y= b))
  }
line_2016 <- plot_line(data2016,data2016$Vitri,data2016$GiatriTaiSan) +
   labs(title ="Bieu do gia tri tai san nam 2016", x = "Index", y = "Ty") +
       geom_line(color = "red", size = 0.8 )

line_2019 <- plot_line(data2019,data2019$Vitri,data2019$GiatriTaiSan) +
  labs(title ="Bieu do gia tri tai san nam 2019", x = "Index", y = "Ty") +
      geom_line(color = "red", size = 0.8 )

line_2020 <- plot_line(data2020,data2020$Vitri,data2020$GiatriTaiSan) +
  labs(title ="Bieu do gia tri tai san nam 2020", x = "Index", y = "Ty") +
      geom_line(color = "red", size = 0.8 )

line_2022 <- plot_line(data2022,data2022$Vitri,data2022$GiatriTaiSan )+
  labs(title ="Bieu do gia tri tai san nam 2022", x = "Index", y = "Ty") +
      geom_line(color = "red", size = 0.8 )

grid.arrange(line_2016,line_2019,line_2020,line_2022)
```

- Biểu đồ giá trị tài sản 4 năm 2016, 2019, 2020, 2022

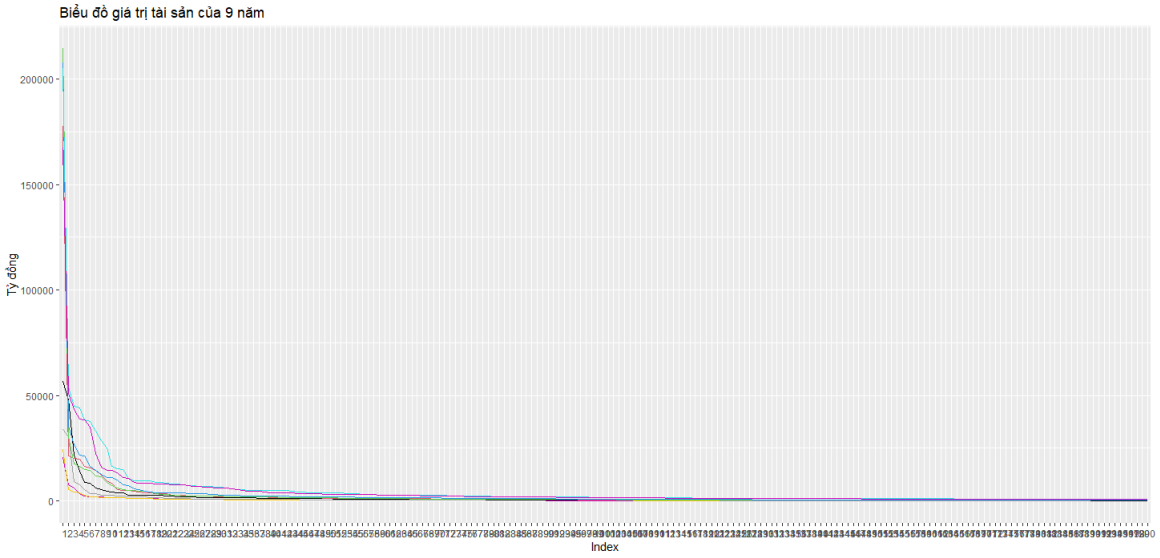


2 **Biểu đồ đường giá trị tài sản gồm tất cả các năm.**

- Trình bày cách làm
  - Ở câu này cũng lấy dữ liệu từ bảng trên nhưng ta sẽ vẽ đồ thị tài sản của 9 năm vào cùng 1 biểu đồ để có cái nhìn tổng quan về toàn bộ tài sản
  - Dùng ggplot dạng biểu đồ đường cho tài sản 9 năm
- Source code

```
          # Ve bieu do gia tri tai san cho tat ca cac nam

ggplot(data = data.new ,aes(x = Vitri, group = 1)) +
geom_line(aes(y =data2014$GiatriTaiSan), color = "2014") +
geom_line(aes(y =data2015$GiatriTaiSan), color = "2015") +
geom_line(aes(y =data2016$GiatriTaiSan), color = "2016") +
geom_line(aes(y =data2017$GiatriTaiSan), color = "2017") +
geom_line(aes(y =data2018$GiatriTaiSan), color = "2018") +
geom_line(aes(y =data2019$GiatriTaiSan), color = "2019") +
geom_line(aes(y =data2020$GiatriTaiSan), color = "2020") +
geom_line(aes(y =data2021$GiatriTaiSan), color = "2021") +
geom_line(aes(y =data2022$GiatriTaiSan), color = "2022") +
labs(title = "Bieu do gia tri tai san cua 9 nam",x="Index", y= "Ty dong" , color =
    "Year", size = 0.5)
```

- Biểu đồ giá trị tài sản trong 9 năm từ 2014-2022



Nhận xét : Do sự chênh lệch về giá trị tài sản các năm cũng như giá trị tài sản trong năm của mỗi người chênh lệch khá lớn nên đồ thị khó quan sát hơn

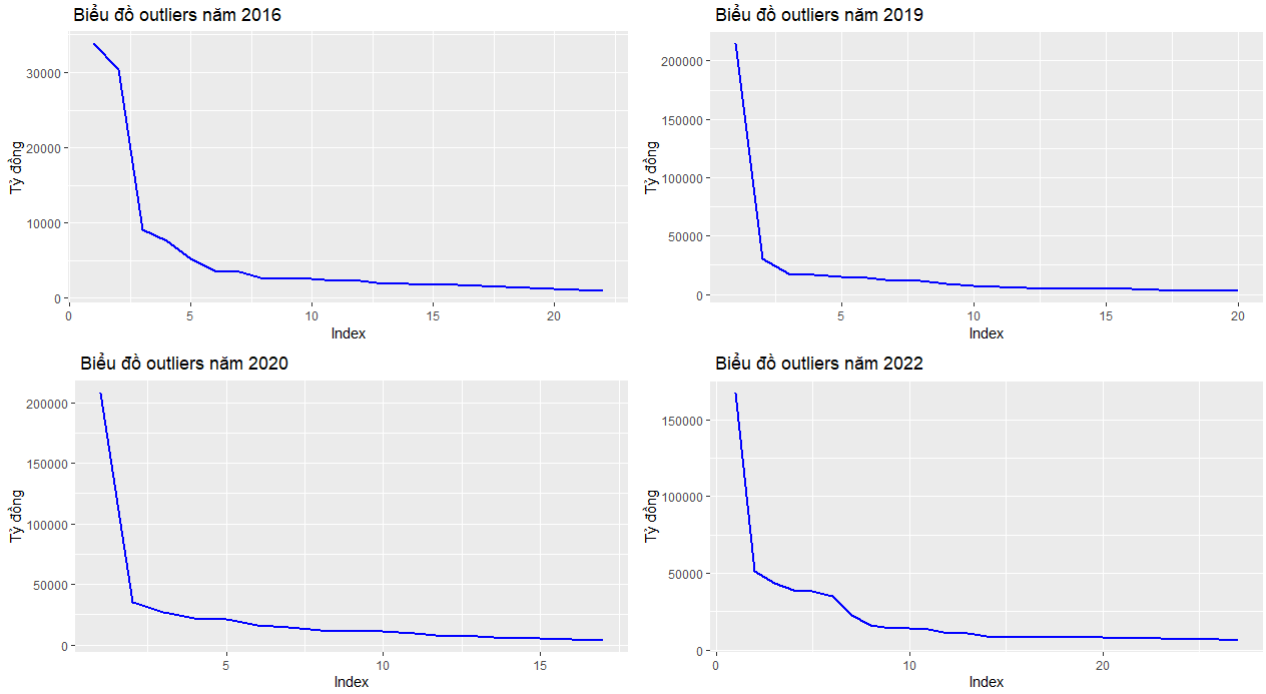
3 **Biểu đồ đường giá trị tài sản mà outliers xuất hiện cho 4 năm.**

- Trình bày cách làm
  - Từ dữ liệu câu *ii*)3) ở trên ta lập bảng giá trị để tìm ra các giá trị outlier của 4 năm theo đề bài.
  - Sau đó tìm những giá trị nằm ngoài outliers với hàm which().
  - Tiếp đến sẽ thành lập 1 bảng các giá trị outliers theo từng năm của đề bài
  - Cuối cùng tiế hành lập hàm để vẽ đồ thị rồi dùng grid.arrange() để hiển thị 4 đồ thị trong 1 khung hình
- Source code

```
#tinh gia tri cac outlier
ii.3.taiSan.outlier <- c()
ii.3.iqr <- data.result.ii.3$Q3 - data.result.ii.3$Q1
ii.3.outlierMin <- data.result.ii.3$Q1 - 1.5 * ii.3.iqr
ii.3.outlierMax <- data.result.ii.3$Q3 + 1.5 * ii.3.iqr

#lap bang gia tri outlier
outlier2016 <- data.taiSan[,3][which(data.taiSan[,3] < ii.3.outlierMin[3] |
    data.taiSan[,3] > ii.3.outlierMax[3])]
outlier2019 <- data.taiSan[,6][which(data.taiSan[,6] < ii.3.outlierMin[6] |
    data.taiSan[,6] > ii.3.outlierMax[6])]
outlier2020 <- data.taiSan[,7][which(data.taiSan[,7] < ii.3.outlierMin[7] |
    data.taiSan[,7] > ii.3.outlierMax[7])]
outlier2022 <- data.taiSan[,9][which(data.taiSan[,9] < ii.3.outlierMin[9] |
    data.taiSan[,9] > ii.3.outlierMax[9])]

outlier_line2016 <- outlier_line(out2016, out2016$Index, out2016$Property) +
                                geom_line(color = "blue" , size = 0.8) +
                                ggtitle(" Bieu do outliers nam 2016") +
                                labs(x= "Index", y = "Ty dong")
outlier_line2019 <- outlier_line(out2019, out2019$Index, out2019$Property) +
                                geom_line(color = "blue" , size = 0.8) +
                                ggtitle(" Bieu do outliers nam 2019") +
                                labs(x= "Index", y = "Ty dong")
outlier_line2020 <- outlier_line(out2020, out2020$Index, out2020$Property) +
                                geom_line(color = "blue" , size = 0.8) +
```

```
                                    ggtitle(" Bieu do outliers nam 2020") +
                                    labs(x= "Index", y = "Ty dong")
outlier_line2022 <- outlier_line(out2022, out2022$Index, out2022$Property) +
                                    geom_line(color = "blue" , size = 0.8) +
                                    ggtitle(" Bieu do outliers nam 2022") +
                                    labs(x= "Index", y = "Ty dong")
grid.arrange(outlier_line2016, outlier_line2019, outlier_line2020, outlier_line2022)
```

- Biểu đồ giá trị tài sản mà outliers xuất hiện cho 4 năm



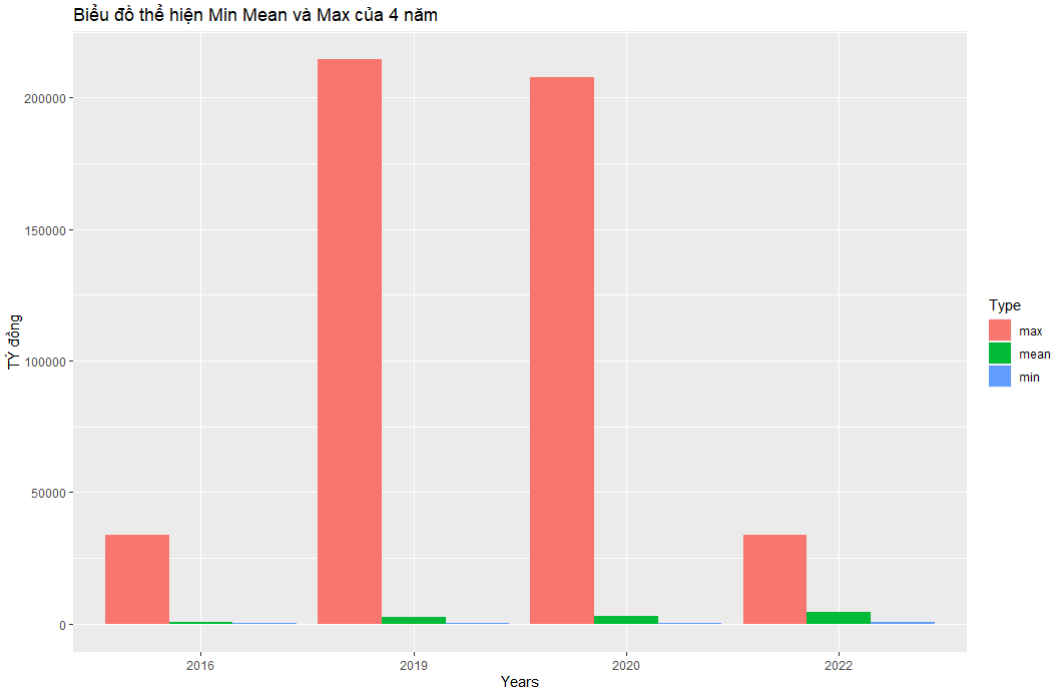
4 Vẽ biểu đồ thể hiện min, max mean giá trị tài sản cho 4 năm

- Trình bày cách làm
  - Đầu tiên chúng ta dựa vào các dữ liệu đã có dùng các hàm mean, min, max trong R để tiến hành tìm ra các giá trị của mỗi năm
  - Thành lập các bảng giá trị của mỗi năm rồi dùng *rbind()* để trở thành data.frame.
  - Dùng biến biểu đồ để thành lap 1 bảng đầy đủ các yếu tố để vẽ biểu đồ bằng cách dùng *cbind()* để nối các cột lại với nhau.
  - Sau khi đã hoàn thành bảng ta tiến hành vẽ biểu đồ sử dụng ggplot để vẽ biểu đồ bằng `geom_bar()`.
- Source code

```
#Ve bieu do min, max mean cho 4 nam
#nam 2016
mean2016 =mean(data2016$GiatriTaiSan)
min2016 = min(data2016$GiatriTaiSan)
max2016 = max(data2016$GiatriTaiSan)
nam2016 = data.frame(rbind(mean2016,min2016,max2016))
colnames(nam2016) =c("Taisan")

#Thuc hien tuong tu cho nam 2019, 2020, 2022

pp <- rbind(nam2016, nam2019,nam2020, nam2022)
Years = rep(c("2016", "2019","2020","2022"), each = 3)
```

```
Year = data.frame(Years)
Type = c("mean","min","max","mean","min","max","mean","min","max","mean","min","max")

bieudo <- cbind(Year, Type,pp)
ggplot(bieudo, aes(fill = Type, x= Years, y = Taisan)) +
                      geom_bar(position='dodge', stat='identity') +
                      ggtitle("Bieu do the hien Min, Mean va Max cua 4 nam") + labs(x
                          = "Years" , y= "Ty dong")
```

- Biểu đồ thể hiện min, mean, max giá trị tài sản cho 4 năm



v) Nhóm câu hỏi liên quan đến middle class

Mã đề: 9367

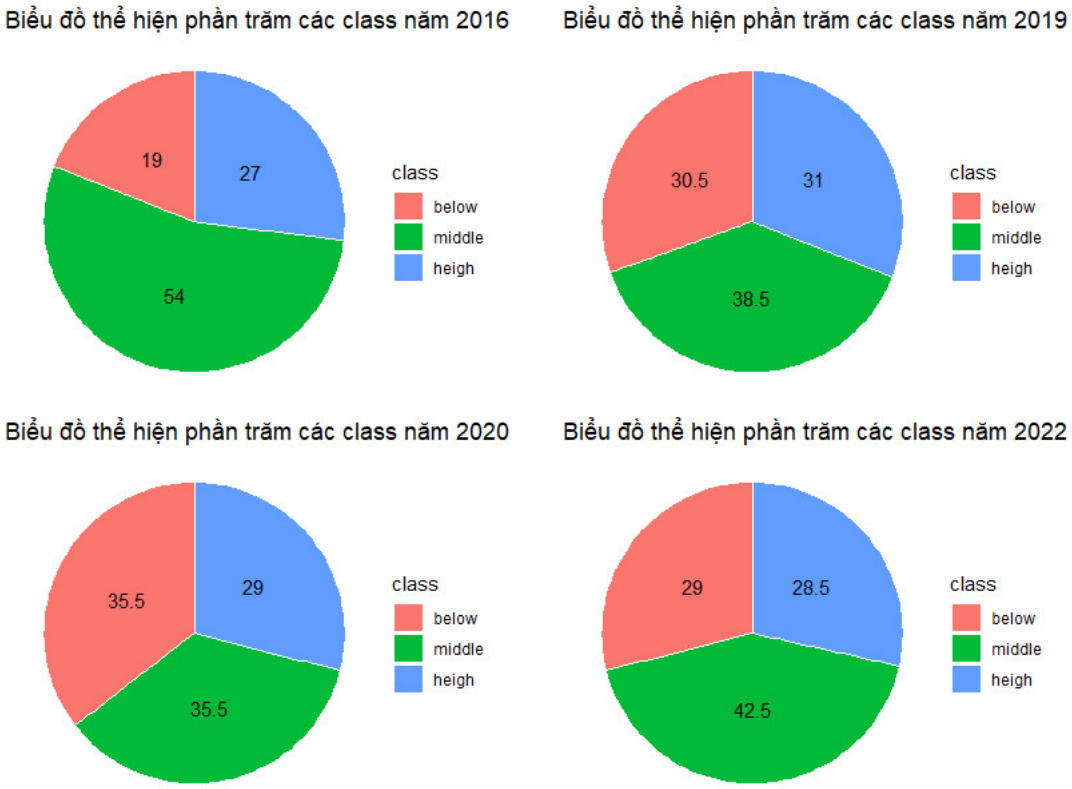
Định nghĩa "middle class" khi một người có thu nhập từ hai phần 3 đến gấp đổi thu nhập trung vị (median) trong năm.

1 **Vẽ 4 biểu đồ thể hiện phần trăm below class, middle class, heigh class cho 4 năm.**

- Trình bày cách làm
  - Dùng hàm *which* để lọc ra các giá trị tài sản thuộc mỗi class trong tất cả các năm, sau đó tính số người thuộc mỗi class trong 4 năm 2016, 2019, 2020, 2022.
  - Dùng số liệu trên để vẽ biểu đồ tròn thể hiện phần trăm của mỗi class cho 4 năm 2016, 2019, 2020, 2022.
- Source code

```
#Xu li so lieu
v.findMiddle <- function(data.taiSan.year, q2){
  index.middle <- which(data.taiSan.year >= (2/3)*q2 & data.taiSan.year <= 2*q2)
  middle <- sort(data.taiSan.year[index.middle])
  return(middle)
}
v.findBelow <- function(data.taiSan.year, q2){
  index.below <- which(data.taiSan.year < (2/3)*q2)
  below <- sort(data.taiSan.year[index.below])
  return(below)
}
v.findHeigh <- function(data.taiSan.year, q2){
  index.heigh <- which(data.taiSan.year > 2*q2)
  heigh <- sort(data.taiSan.year[index.heigh])
  return(heigh)
}
#Vi du cho nam 2016, tuong tu cho cac nam 2019, 2020, 2022
v.2016.middle <- v.findMiddle(data.taiSan$`2016`, data.result.ii.3[3,4])
v.2016.below <- v.findBelow(data.taiSan$`2016`, data.result.ii.3[3,4])
v.2016.heigh <- v.findHeigh(data.taiSan$`2016`, data.result.ii.3[3,4])
below2016 <- length(v.2016.below)
middle2016 <- length(v.2016.middle)
heigh2016 <- length(v.2016.heigh)

data2016 <- data.frame(
  class = c("below", "middle", "heigh"),
  value2016 = c(below2016, middle2016, heigh2016),
  percent2016 = c(below2016/200*100, middle2016/200*100, heigh2016/200*100)
)

data2016 <- data2016 %>%
  mutate(class = fct_relevel(class, "below", "middle", "heigh"))

data2016 <- data2016 %>%
  arrange(desc(class)) %>%
  mutate(lab.ypos = cumsum(percent2016) - 0.5*percent2016)

piechart2016 <- ggplot(data2016, aes(x = "", y = percent2016, fill = class)) +
  geom_bar(width = 1, stat = "identity", color = "white") +
  coord_polar("y", start = 0) +
  geom_text(aes(y = lab.ypos, label = percent2016), color = "black") +
  ggtitle("Bieu do the hien phan tram cac class trong nam 2016") +
  theme_void()

#Lenh gop 4 bieu do tron chung 1 khung hinh
grid.arrange(piechart2016, piechart2019, piechart2020, piechart2022)
```

- Biểu đồ thể hiện phần trăm below class, middle class, heigh class cho 4 năm



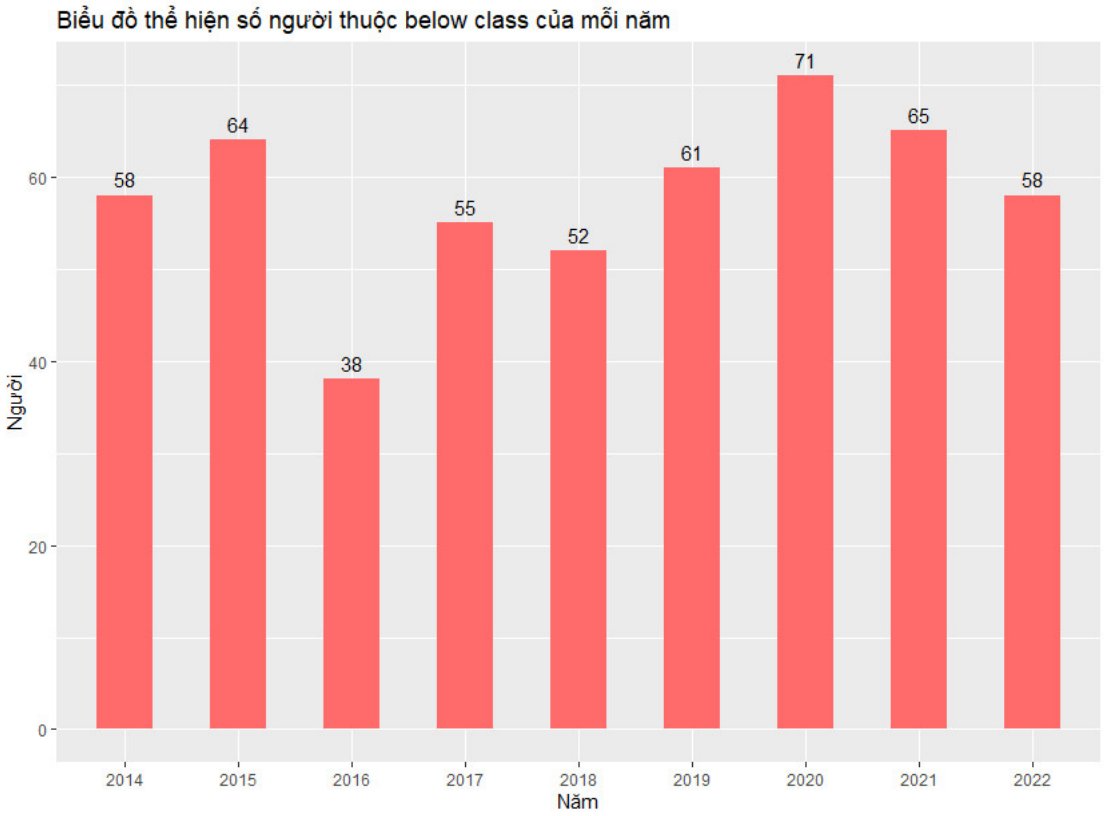
- Nhận xét
  - Qua 4 năm 2016, 2019, 2020, 2022 ta thấy được số người thuộc below class có sự tăng lên khá nhiều kể từ năm 2016 và ít dao động ở các năm 2019, 2020, 2022, trong khi đó có sự ngược lại ở middle class khi năm 2016 lại là năm có số người đông nhất, các năm còn lại có sự tụt giảm và không chênh lệch nhiều ở các năm 2019, 2020, 2022.
  - Heigh class có số người ổn định nhất khi trải qua 4 năm mà không có sự chênh lệch nhiều so với tổng thể.

2 **Biểu đồ thể hiện số người thuộc below class trong mỗi năm, gồm tất cả các năm.**

- Trình bày cách làm
  - Sử dụng kết quả ở câu 1 tính được số lượng người thuộc below class của mỗi năm.
  - Vẽ biểu đồ cột thể hiện số người thuộc below class trong mỗi năm, gồm tất cả các năm, với trục Ox là Năm, trục Oy là Người.
- Source code

```
dataBelow <- data.frame(
  year = c("2014", "2015", "2016", "2017", "2018", "2019", "2020", "2021", "2022"),
  y = c(below2014, below2015, below2016, below2017, below2018, below2019, below2020,
      below2021, below2022)
)
barchartBelow <- ggplot(dataBelow, aes(x = year, y = y)) +
  geom_bar(stat = "identity", width = 0.5, fill = "indianred1") +
  geom_text(
    aes(label = y, y = y + 0.05),
    position = position_dodge(0.9),
    vjust = -0.5
  ) +
  xlab("Nam") + ylab("Nguoi") +
  ggtitle("Bieu do the hien so nguoi thuoc below class cua moi nam")

barchartBelow
```

- Biểu đồ thể hiện số người thuộc below class trong mỗi năm, gồm tất cả các năm



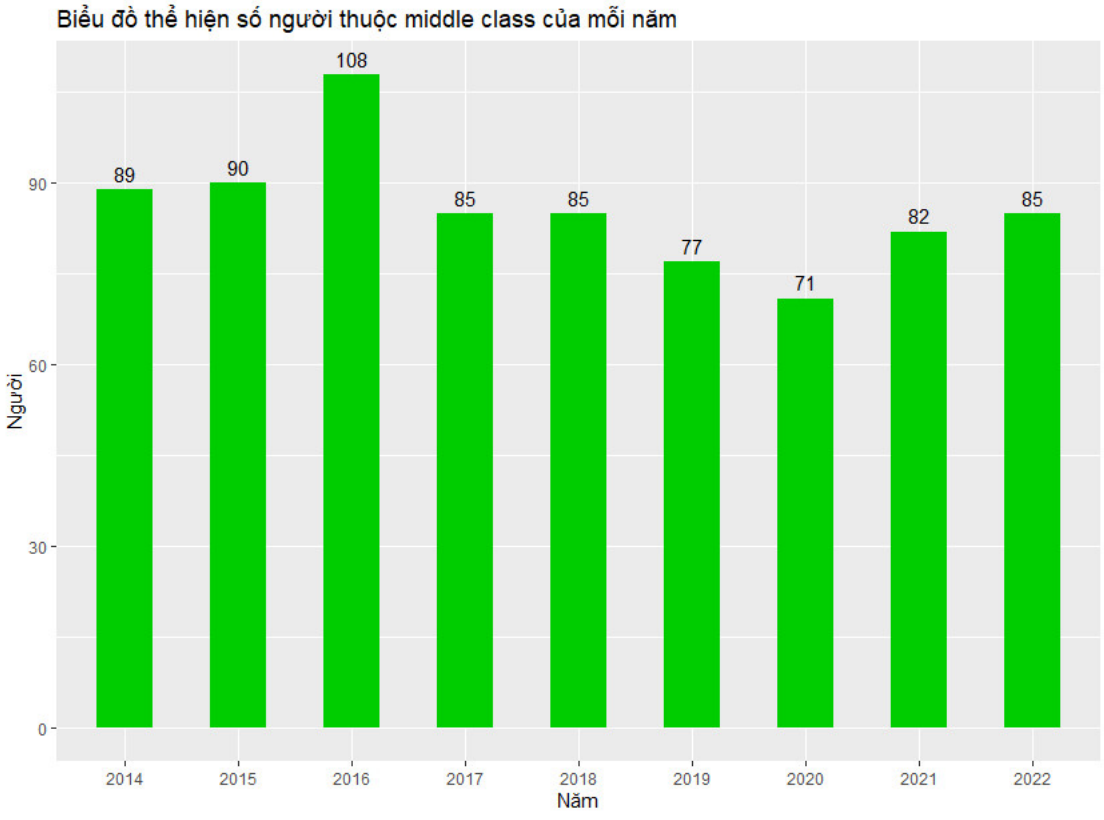
- Nhận xét
  Qua 9 năm, ta thấy rõ số người thuộc below class thấp nhất ở năm 2016 – đây là năm số lượng người thuộc below class khá thấp do với các năm còn lại, và cao nhất ở năm 2020, các năm còn lại dao động không đáng kể.

3 **Biểu đồ thể hiện số người thuộc middle class trong mỗi năm, gồm tất cả các năm.**

- Các bước làm hoàn toàn tương tự câu 2 phần below class, chỉ đổi lại các tên biến từ below thành middle.
- Source code

```
dataMiddle <- data.frame(
  year = c("2014", "2015", "2016", "2017", "2018", "2019", "2020", "2021", "2022"),
  y = c(middle2014, middle2015, middle2016, middle2017, middle2018, middle2019,
      middle2020, middle2021, middle2022)
)
barchartMiddle <- ggplot(dataMiddle, aes(x = year, y = y)) +
  geom_bar(stat = "identity", width = 0.5, fill = "green3") +
  geom_text(
    aes(label = y, y = y + 0.05),
    position = position_dodge(0.9),
    vjust = -0.5
  ) +
  xlab("Nam") + ylab("Nguoi") +
  ggtitle("Bieu do the hien so nguoi thuoc middle class cua moi nam")

barchartMiddle
```

- Biểu đồ thể hiện số người thuộc middle class trong mỗi năm, gồm tất cả các năm



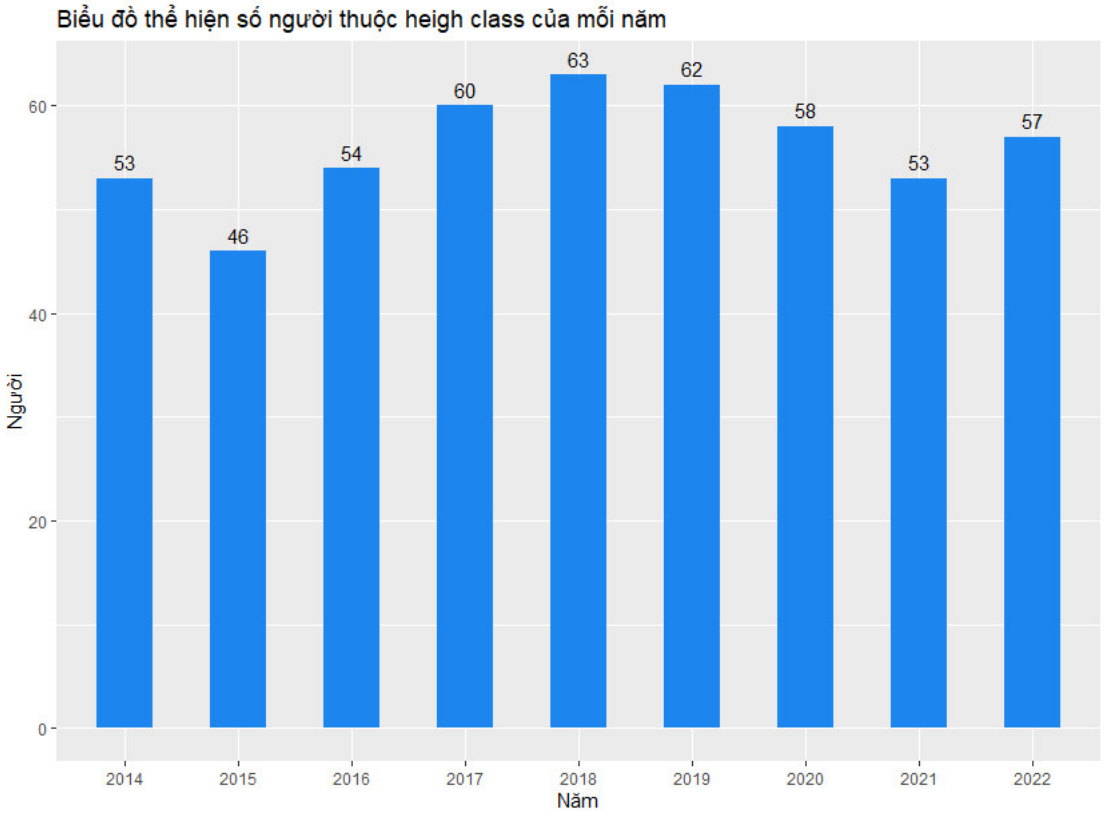
- Nhận xét
  Qua 9 năm, ta thấy rõ số người thuộc middle class cao nhất ở năm 2016 – năm này có số người khá cao so với các năm khác, và thấp nhất ở năm 2020, các năm còn lại dao động không đáng kể.

4 **Biểu đồ thể hiện số người thuộc heigh class trong mỗi năm, gồm tất cả các năm.**

- Các bước làm hoàn toàn tương tự câu 2 phần below class, chỉ đổi lại các tên biến từ below thành heigh.
- Source code

```
dataHeigh <- data.frame(
  year = c("2014", "2015", "2016", "2017", "2018", "2019", "2020", "2021", "2022"),
  y = c(heigh2014, heigh2015, heigh2016, heigh2017, heigh2018, heigh2019, heigh2020,
      heigh2021, heigh2022)
)
barchartHeigh <- ggplot(dataHeigh, aes(x = year, y = y)) +
  geom_bar(stat = "identity", width = 0.5, fill = "dodgerblue2") +
  geom_text(
    aes(label = y, y = y + 0.05),
    position = position_dodge(0.9),
    vjust = -0.5
  ) +
  xlab("Nam") + ylab("Nguoi") +
  ggtitle("Bieu do the hien so nguoi thuoc heigh class cua moi nam")

barchartHeigh
```

- Biểu đồ thể hiện số người thuộc heigh class trong mỗi năm, gồm tất cả các năm



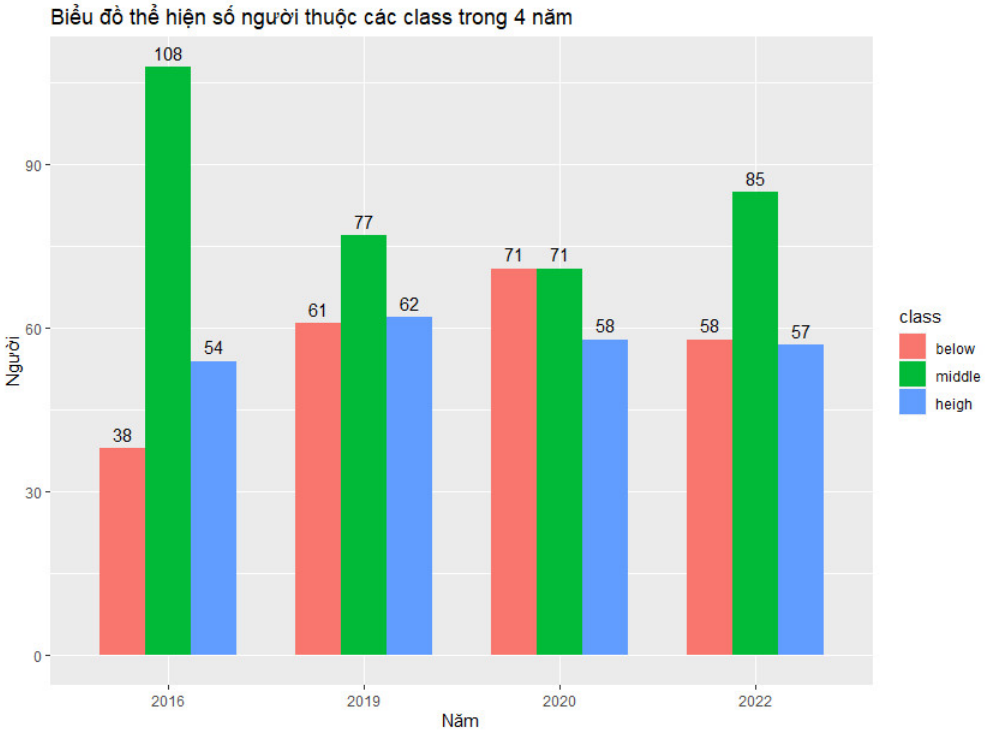
- Nhận xét
  Nhìn qua biểu đồ, ta thấy heigh class có số người ổn định nhất qua các năm, trong đó năm 2015 có phần thấp so với mặt bằng chung nhưng không đáng kể

5 **Biểu đồ thể hiện 4 năm (4 cluster). Trong mỗi năm (mỗi cluster) sẽ gồm 3 cột, cột một là số người below class, cột 2 là số người middle class, cột 3 là số người heigh class.**

- Trình bày cách làm
  - Sử dụng kết quả ở câu 1 tính được số lượng người thuộc mỗi class của 4 năm 2016, 2019, 2020, 2022.
  - Vẽ biểu đồ cột thể hiện số người thuộc mỗi class của 4 năm 2016, 2019, 2020, 2022, với trục Ox là Năm, trục Oy là Người.
- Source code

```
data.clustered <- data.frame(
  year = factor(c(2016, 2016, 2016, 2019, 2019, 2019, 2020, 2020, 2020, 2022, 2022,
      2022)),
  y = c(below2016, middle2016, heigh2016, below2019, middle2019, heigh2019,
      below2020, middle2020, heigh2020, below2022, middle2022, heigh2022),
  class = c("below", "middle", "heigh", "below", "middle", "heigh", "below",
      "middle", "heigh", "below", "middle", "heigh")
)
data.clustered <- data.clustered %>%
  mutate(class = fct_relevel(class, "below", "middle", "heigh"))
clusteredBarchart <- ggplot(data = data.clustered, aes(year, y, group = class)) +
  geom_col(aes(fill = class), width = 0.7, position = "dodge") +
  geom_text(
    aes(label = y, y = y + 0.05),
    position = position_dodge(0.7),
    vjust = -0.5
  ) +
  xlab("Nam") + ylab("Nguoi") +
  ggtitle("Bieu do the hien so nguoi thuoc cac class trong 4 nam")

clusteredBarchart
```

- Biểu đồ thể hiện số người thuộc mỗi class trong 4 năm



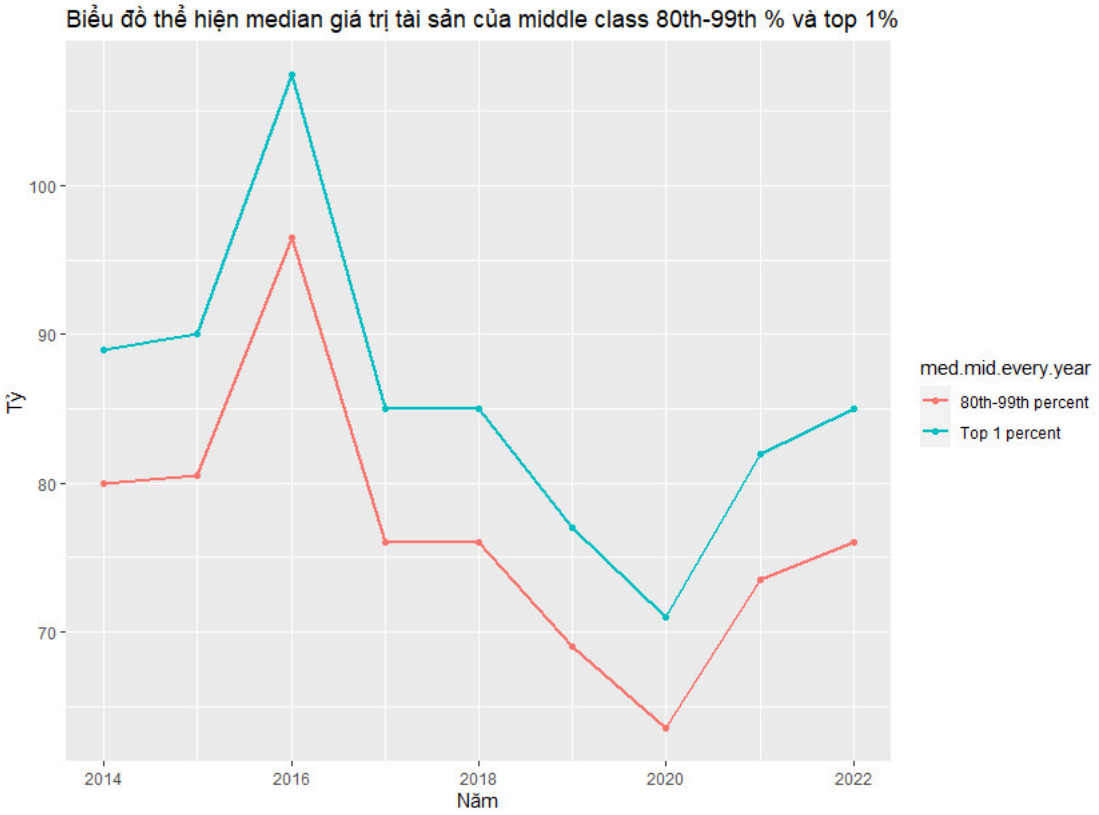
- Nhận xét
  - Số người thuộc below class tăng đều đến năm 2020, sau đó có sự suy giảm nhẹ.
  - Số người thuộc middle class giảm mạnh từ năm 2016, sau đó tăng giảm không đáng kể.
  - Số người thuộc heigh class nhìn chung khá ổn định qua 4 năm.

6 **Vẽ biểu đồ thể hiện median giá trị tài sản của middle class 80th - 99th percent và top 1 percent**

- Trình bày cách làm
  - Tìm các giá trị tài sản của middle class 80th – 99th percent và top 1 percent của tất cả các năm.
  - Tiếp theo, sắp xếp và tính median của middle class 80th – 99th percent và top 1 percent của tất cả các năm.
  - Cuối cùng, vẽ biểu đồ đường thể hiện median giá trị tài sản của middle class 80th - 99th percent và top 1 percent với trục Ox là năm, trục Oy là giá trị tài sản (đơn vị: Tỷ).
- Source code

```
v.6.2014.80.99 <- c(ceiling(length(v.2014.middle) * 0.8): floor(length(v.2014.middle)
    * 0.99))
v.6.2015.80.99 <- c(ceiling(length(v.2015.middle) * 0.8): floor(length(v.2015.middle)
    * 0.99))
#Tuong tu cho cac nam con lai

v.6.2014.99.100 <- c(ceiling(length(v.2014.middle) * 0.99): length(v.2014.middle))
v.6.2015.99.100 <- c(ceiling(length(v.2015.middle) * 0.99): length(v.2015.middle))
#Tuong tu cho cac nam con lai
v.6.taiSan.80.99.median <- c(quantile(v.6.2014.80.99, 0.5, names = FALSE),
                             quantile(v.6.2015.80.99, 0.5, names = FALSE),
                             quantile(v.6.2016.80.99, 0.5, names = FALSE),
                             quantile(v.6.2017.80.99, 0.5, names = FALSE),
                             quantile(v.6.2018.80.99, 0.5, names = FALSE),
                             quantile(v.6.2019.80.99, 0.5, names = FALSE),
                             quantile(v.6.2020.80.99, 0.5, names = FALSE),
                             quantile(v.6.2021.80.99, 0.5, names = FALSE),
                             quantile(v.6.2022.80.99, 0.5, names = FALSE))

#Tuong tu cho v.6.taiSan.99.100.median
v.6.taiSan.median <- data.frame(append(rep("80th-99th percent", 9), rep("Top 1
    percent", 9)),
                                rep(c(2014,2015,2016,2017,2018,2019,2020,2021,2022),2),
                                append(v.6.taiSan.80.99.median,
                                    v.6.taiSan.99.100.median))
colnames(v.6.taiSan.median) <- c("med.mid.every.year", "Year", "Value")

groupLinechart <- ggplot(data = v.6.taiSan.median,
                         aes(x = Year, y = Value, color = med.mid.every.year)) +
                         geom_line(linewidth = 1) +
                         xlab("Nam") + ylab("Ty") +
                         geom_point() +
                         ggtitle("Bieu do the hien median gia tri tai san cua middle
                             clase 80th-99th % va top 1% "
groupLinechart
```

- Biểu đồ thể hiện median giá trị tài sản của middle class 80th - 99th percent và top 1 percent



- Nhận xét
  Ta thấy 2 đồ thị trên có hình dáng khá tương tự nhau, chứng tỏ top 1% vẫn hơn 80th-99th percent một khoảng ổn định về giá trị tài sản qua các năm, không có sự thu hẹp hay giãn cách quá lớn.

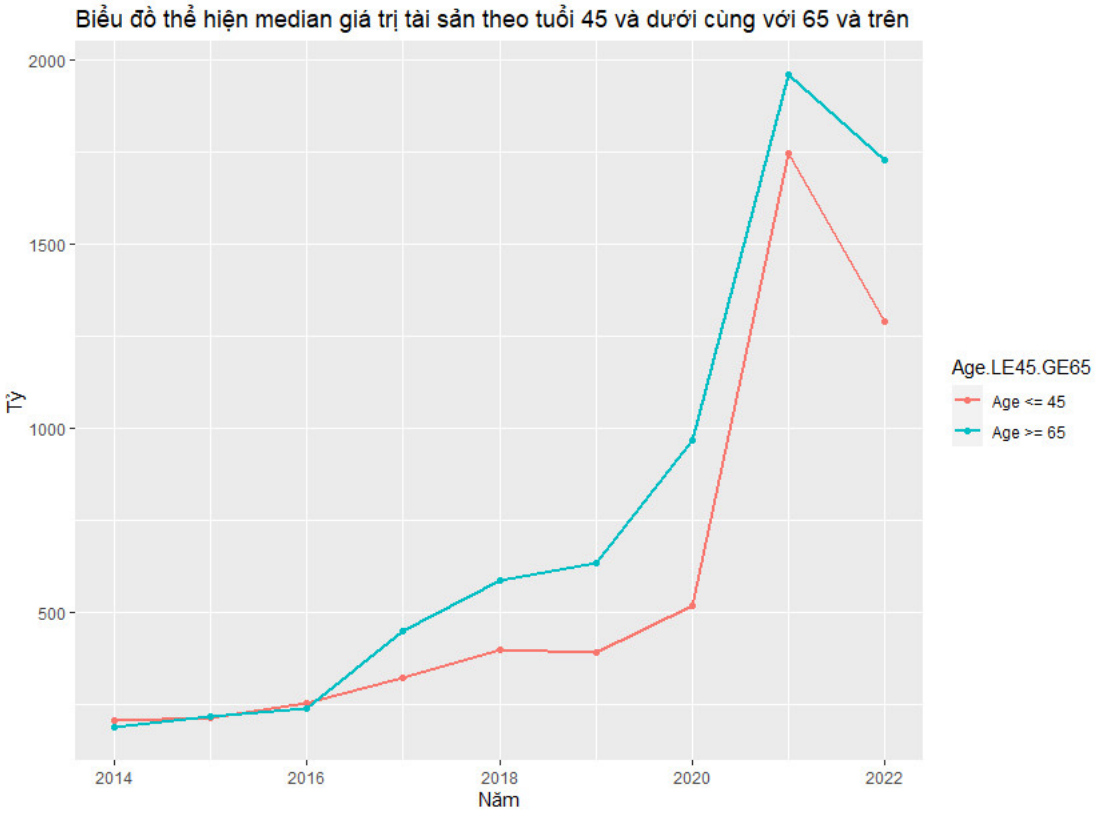
7 Biểu đồ thể hiện median giá trị tài sản theo tuổi 45 và dưới cùng với 65 và trên.

- Trình bày cách làm
  - Lọc tuổi tất cả các năm, từ đó chọn ra các giá trị tuổi thỏa mãn nhỏ hơn hoặc bằng 45 cùng với lớn hơn hoặc bằng 65.
  - Sắp xếp và tính median của các giá trị tuổi thỏa mãn nhỏ hơn hoặc bằng 45 cùng với lớn hơn hoặc bằng 65 của tất cả các năm.
  - Vẽ biểu đồ đường thể hiện median giá trị tài sản theo tuổi 45 và dưới cùng với 65 và trên với trục Ox là năm, trục Oy là giá trị tài sản (đơn vị: Tỷ).

- Source code

```
index.tuoi <- c()
for(i in 0:8){
  index.tuoi <- append(index.tuoi, 3+6*i)
}
tuoi.1 <- replace(data.new[,index.tuoi[1]], data.new[,index.tuoi[1]] == "0", NA)
tuoi.2 <- replace(data.new[,index.tuoi[2]], data.new[,index.tuoi[2]] == "0", NA)
#Tuong tu cho tuoi.3, tuoi.4, ...

tuoi.1.below45 <- which(as.numeric(tuoi.1) <= 45)
tuoi.2.below45 <- which(as.numeric(tuoi.2) <= 45)
#Tuong tu cho tuoi.3.blow45, tuoi.4.blow45, ...

index.taiSan <- c()
for(i in 0:8){
  index.taiSan <- append(index.taiSan, 7+6*i)
}
taiSan.1.below45 <- as.numeric(data.new[tuoi.1.below45, index.taiSan[1]])
taiSan.2.below45 <- as.numeric(data.new[tuoi.2.below45, index.taiSan[2]])
#Tuong tu cho taiSan.3.below45, taiSan.4.below45, ...

taiSan.below45.median <- c(median(taiSan.1.below45),
                           median(taiSan.2.below45),
                           median(taiSan.3.below45),
                           median(taiSan.4.below45),
                           median(taiSan.5.below45),
                           median(taiSan.6.below45),
                           median(taiSan.7.below45),
                           median(taiSan.8.below45),
                           median(taiSan.9.below45))


#Thuc hien tuong tu 3 buoc tren cho taiSan.higher65.median

med.le45.ge65<-data.frame(append(rep("Age <= 45", 9), rep("Age >= 65", 9)),
                          rep(c(2014,2015,2016,2017,2018,2019,2020,2021,2022),2),
                          append(taiSan.below45.median, taiSan.higher65.median))
colnames(med.le45.ge65) <- c("Age.LE45.GE65", "Year", "Value")

medianV7Linechart <- ggplot(data = med.le45.ge65, aes(x = Year, y = Value, color =
    Age.LE45.GE65)) +
  geom_line(linewidth = 1) +
  xlab("Nam") + ylab("Ty") +
  geom_point() +
  ggtitle("Bieu do the hien median gia tri tai san theo tuoi 45 va duoi cung voi 65
      va tren")

medianV7Linechart
```

- Biểu đồ thể hiện median giá trị tài sản theo tuổi 45 và dưới cùng với 65 và trên



- Nhận xét
  Qua biểu đồ ta thấy median giá trị tài sản của 2 nhóm tuổi <=45 và >=65 gần như xấp xỉ ở các năm 2014-2016, nhưng sau đó nhóm tuổi >=65 đã vượt lên so với nhóm <=45 từ năm 2016-2020, sau đó khoảng cách đã được rút ngắn lại ở các năm 2021 và cả 2 đều sụt giảm ở năm 2022

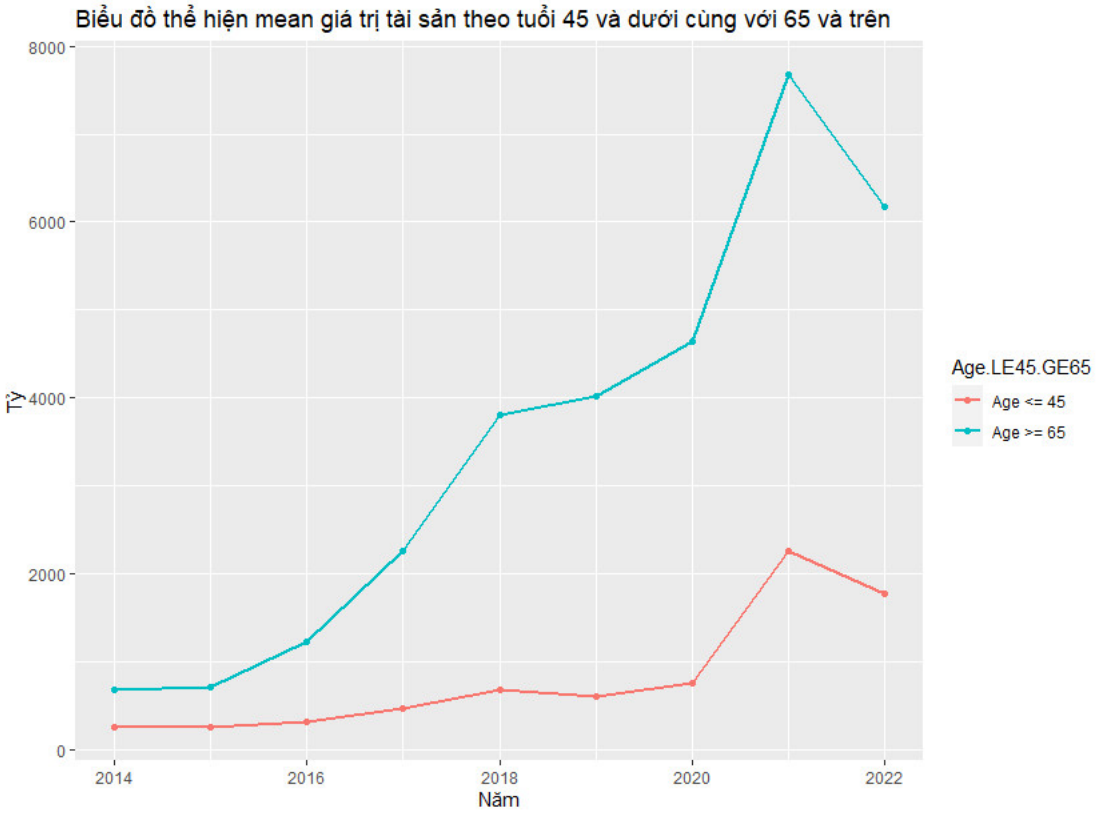
8 **Biểu đồ thể hiện mean giá trị tài sản theo tuổi 45 và dưới cùng với 65 và trên.**

- Trình bày cách làm
  - Dựa vào tuổi đã lọc ở câu 7, tính được mean của các giá trị tuổi thỏa mãn nhỏ hơn hoặc bằng 45 cùng với lớn hơn hoặc bằng 65 của tất cả các năm.
  - Vẽ biểu đồ đường thể hiện mean giá trị tài sản theo tuổi 45 và dưới cùng với 65 và trên với trục Ox là năm, trục Oy là giá trị tài sản (đơn vị: Tỷ).

- Source code

```
taiSan.below45.mean <- c(mean(taiSan.1.below45),
                         mean(taiSan.2.below45),
                         mean(taiSan.3.below45),
                         mean(taiSan.4.below45),
                         mean(taiSan.5.below45),
                         mean(taiSan.6.below45),
                         mean(taiSan.7.below45),
                         mean(taiSan.8.below45),
                         mean(taiSan.9.below45))
#Tuong tu cho taiSan.higher65.mean
mean.le45.ge65 <- data.frame(append(rep("Age <= 45", 9), rep("Age >= 65", 9)),
                             rep(c(2014,2015,2016,2017,2018,2019,2020,2021,2022),2),
                             append(taiSan.below45.mean, taiSan.higher65.mean))
colnames(mean.le45.ge65) <- c("Age.LE45.GE65", "Year", "Value")
meanV8Linechart <- ggplot(data = mean.le45.ge65,
                          aes(x = Year, y = Value, color = Age.LE45.GE65)) +
                          geom_line(linewidth = 1) +
                          xlab("Nam") + ylab("Ty") +
                          geom_point() +
                          ggtitle("Bieu do the hien mean gia tri tai san theo tuoi 45
                              va duoi cung voi 65 va tren")
meanV8Linechart
```

- Biểu đồ thể hiện mean giá trị tài sản theo tuổi 45 và dưới cùng với 65 và trên



- Nhận xét
  Qua biểu đồ ta thấy mean giá trị tài sản của nhóm tuổi >=65 tăng mạnh qua từng năm và đã bỏ xa so với nhóm tuổi <=45, trong khi nhóm tuổi <=45 chỉ duy trì ổn định và không có bước đột phá lớn trong mean giá trị tài sản.

vi) Nhóm câu hỏi riêng

3) **Cho biết tỉnh nào có giá trị tài sản thay đổi giảm mạnh nhất.**

- Trình bày cách làm
  - Dựa vào dữ liệu thống kê đã được xử lý thông tin về nơi sinh ở phần trước – data.diaDanh, ta tiến hành lọc dữ liệu tổng giá trị tài sản của tất cả các tỉnh qua các năm từ 2014 đến 2022 và lưu vào biến province.proper.
  - Tuy nhiên sau khi lọc dữ liệu, bảng thống kê còn thiếu một trường là tập hợp tất cả các năm, là vector dữ liệu cần thiết để xử lý dữ liệu và vẽ biểu đồ sau này, do đó ta tiến hành thêm một cột năm vào bảng thống kê.
  - Tiếp theo, duyệt hết tất cả các tỉnh, với mỗi tỉnh ta tiến hành tính toán mức giảm tài sản trong 2 năm liên tiếp, tiến hành liên tục từ năm 2014 đến năm 2022, sau đó dùng hàm max() tìm mức giảm mạnh nhất của tỉnh đó qua 9 năm, lưu trữ lại mức giảm lớn nhất của tỉnh đó vào một biến decrease và tên tỉnh vào một biến province, mỗi tỉnh ứng với một chỉ số tính từ 1 và được lưu vào biến index.
  - Cuối cùng ta tạo một dataframe là most.decrease là tập hợp của độ giảm tài sản – decrease, tên tỉnh – province, chỉ số của tỉnh – index. Vẽ đồ thị tương quan giữa index và decrease, quan sát và tìm ra tỉnh có tài sản giảm mạnh nhất, bên cạnh đó ta dùng hàm max() để kiểm chứng kết quả được tìm thông qua đồ thị.
- Source code

```
i = 7
while(i <= 55){
  result <- tapply(as.numeric(data.diaDanh[,i]),data.diaDanh[,4],sum)
  if(i==7){
    province.proper <- matrix(result)
  }
  else {
    province.proper <- cbind(province.proper,result)
  }
  i = i + 6
}


colnames(province.proper) <- c(1:9)
province.proper <- t(province.proper)
province.proper <- as.data.frame(province.proper)

year <- c(2014 : 2022)
province.proper <- cbind.data.frame(province.proper, year)

province <- ""
decrease <- 0
for(i in 1:47){
  for(j in 1:8){
    cal <- province.proper[j,i] - province.proper[j+1,i]
    if (j == 1){
      temp <- as.vector(cal)
    }
    else{
      temp <- c(temp,cal)
    }
  }

  if(i == 1){
    province <- as.vector(colnames(province.proper[i]))
    decrease <- as.vector(max(temp))
  }
  else {
```

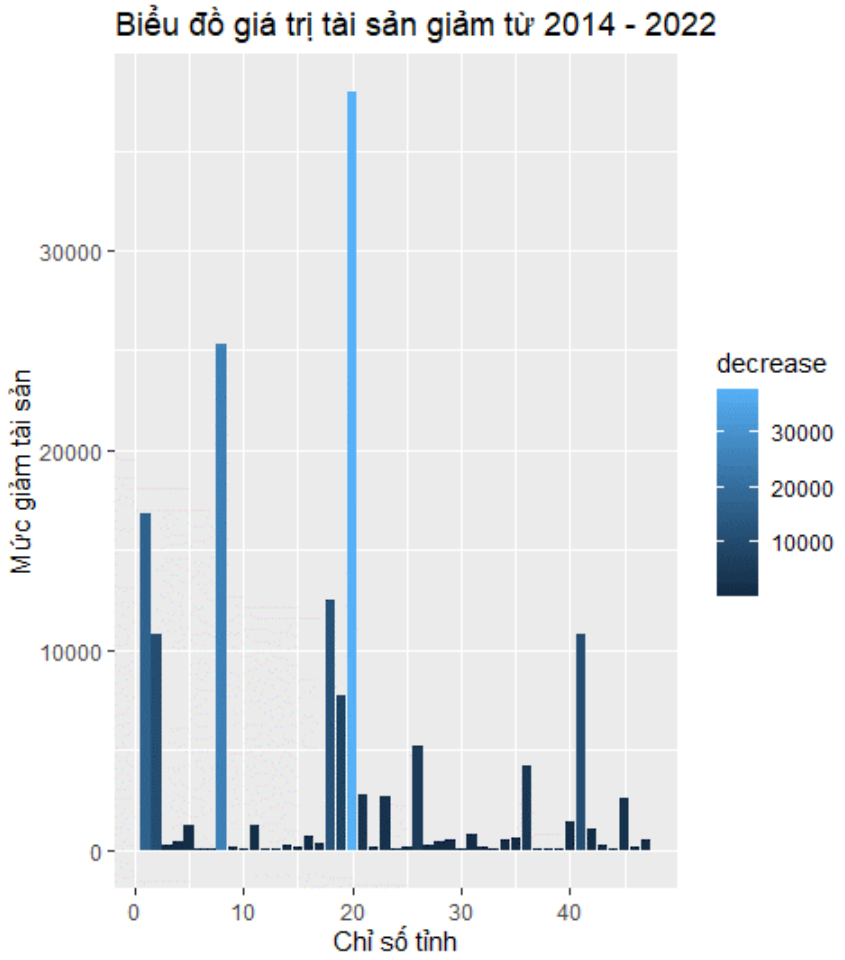
```
    province <- c(province, colnames(province.proper[i]))
    decrease <- c(decrease, max(temp))
  }

}
most.decrease <- data.frame(province,decrease,index = c(1:47))
ggplot(data = most.decrease, binwith = 3) + geom_col(mapping = aes(x = index, y =
    decrease, fill = decrease)) +
labs(
  x = "Chi so tinh",
  y = "Muc giam tai san",
  title = "Bieu do gia tri tai san giam tu 2014-2022"
)
max(most.decrease$decrease)
which(most.decrease$decrease == 37915)
most.decrease$province[20]
```

- Kết quả chạy code Bảng kết quả province.proper :

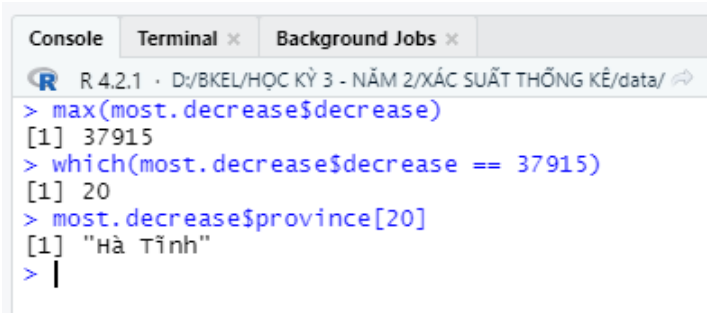
	0	An Giang	Ấn Độ	Bắc Giang	Bắc Ninh	Bến Tre	Bình Dương	Bình Định	Bình Thuận	Cà Mau	Đà Nẵng	Đài Loan	Đồng Nai	Đồng Tháp	Gia Lai	Hà Bắc	Hà Nam	Hà Nam Ninh	Hà Nội	Hà Tĩnh	Hải Dương	Hải Hưng	Hải Phòng
1	18077	1768	534	723	511	94	82	9167	264	83	1607	188	82	818	147	486	412	1764	8663	21451	6235	606	2245
2	19172	1779	504	666	514	114	99	6651	273	102	1477	206	100	745	172	445	495	1641	9205	25353	4360	563	2154
3	25835	3501	719	917	681	142	130	32158	338	131	2415	243	130	1019	197	663	613	2649	16553	35148	9329	765	3335
4	49568	5986	1415	1841	1295	233	199	51160	657	201	3748	485	199	2051	351	1231	1025	5503	30217	59348	21939	1512	5847
5	72243	14396	1800	2323	1843	330	292	25872	969	302	5746	612	293	2614	488	1729	1466	12255	45766	181227	20650	1964	8039
6	72142	11601	1835	2420	1889	324	272	35178	896	280	6000	690	277	2810	509	1745	1443	11405	41067	217988	17844	2090	8920
7	93727	14485	2460	3376	2403	412	342	42164	1430	345	7645	1039	343	3841	722	2316	1819	12381	56847	212991	27201	2831	13811
8	200486	33217	5238	7341	5560	967	883	68479	2601	895	14036	1914	885	8255	1552	4957	4266	28340	110838	215722	46185	6279	22795
9	183600	22368	4989	6922	4341	854	778	64698	2456	784	12764	1802	779	8034	1392	4281	3874	15816	103102	177807	44550	6084	20054

Hưng Yên	Lào Cai	Lâm Đồng	Malaysia	Nam Định	Nghệ An	Quảng Bình	Quảng Nam	Quảng Ninh	Quảng Ngãi	Quảng Trị	Sơn La	Tây Ninh	Tiền Giang	Thái Bình	Thanh Hóa	Thừa Thiên Huế	Trà Vinh	Trung Quốc	Vĩnh Long	Vĩnh Phúc	Năm
234	407	526	189	1961	161	85	1271	415	1823	136	107	301	996	2031	1372	1508	84	3026	238	348	2014
282	379	498	209	2071	199	104	1249	388	1837	170	125	302	1040	2041	1331	1434	103	2856	247	334	2015
324	530	719	244	2616	256	136	1627	548	2931	188	159	381	1373	3219	1934	2179	134	4695	315	462	2016
550	1010	1414	498	4953	386	214	3182	1069	4972	335	272	820	2503	5554	3700	3370	206	7901	604	921	2017
791	1387	1796	613	6800	581	316	4419	1407	6792	470	372	1178	3574	11049	5315	5562	312	10920	830	1300	2018
827	1462	1821	692	7051	540	294	4183	1506	7098	482	368	1248	3599	10161	4956	5795	286	11559	837	1332	2019
1111	1841	2441	1039	9589	687	374	6044	1916	9864	628	465	1655	4803	13033	6456	8524	356	13767	1240	1804	2020
2445	3731	5063	1944	19644	1712	916	12514	3929	19863	1401	1089	3066	10346	29201	13973	13053	909	28874	2407	3650	2021
2157	3272	4532	1838	18873	1525	804	11988	3294	15594	1296	985	2960	8884	18390	12907	12751	795	26245	2203	3110	2022

Xem xét biểu đồ thống kê mức giảm tài sản mạnh nhất của từng tỉnh:



Sau khi quan sát biểu đồ, ta nhận thấy tỉnh có chỉ số 20 là tỉnh có mức giảm tài sản mạnh nhất, ước tính trên 35 nghìn tỉ đồng. Xem xét lại bảng thống kê most.decrease thì tỉnh đó chính là tỉnh Hà Tĩnh. Một lần nữa kiểm tra lại bằng hàm max() để khẳng định kết quả :



Vậy kết quả ta khẳng định là chính xác, ở đây `max(most.decrease$decrease)` cho ta biết tổng tài sản giảm mạnh nhất là bao nhiêu tiền, cụ thể là 37915 tỉ đồng, tương tự như ta ước tính.
`Which(most.decrease$decrease == 37915)` cho ta biết chỉ số của tỉnh tương ứng có số tài sản bị giảm là 37915 tỉ đồng, hàm trả về chỉ số 20. Và tương ứng với chỉ số 20 là tỉnh Hà Tĩnh.

6) **Cho biết nhà đầu tư chứng khoán nào sẽ có tài sản giảm trong năm 2023**

- Trình bày cách làm :

- Dựa vào dữ liệu thống kê đã được xử lý thông tin về giá trị tài sản ở phần trước – data.diaDanh, ta tiến hành lọc dữ liệu tổng giá trị tài sản của tất cả các nhà đầu tư qua các năm từ 2014 đến 2022 và lưu vào biến investors.
- Tuy nhiên sau khi lọc dữ liệu, bảng thống kê còn thiếu một trường là tập hợp tất cả các năm, là vector dữ liệu cần thiết để xử lý dữ liệu và vẽ biểu đồ sau này, do đó ta tiến hành thêm một cột năm vào bảng thống kê.
- Duyệt hết tất cả các nhà đầu tư, với mỗi nhà đầu tư dùng hàm lm() tạo một mô hình hồi quy tuyến tính đơn giản tương quan giữa giá trị tài sản của nhà đầu tư với năm tương ứng. Tiếp theo dùng hàm predict() dựa vào mô hình hồi quy được tạo ở trên để dự đoán tài sản của nhà đầu tư ở năm tiếp theo 2023 và tính hiệu tài sản của năm 2022 và năm 2023. Nếu hiệu tài sản là âm chứng tỏ nhà đầu tư đang có dấu hiệu tăng giá trị tài sản, ngược lại chứng tỏ giá trị tài sản của nhà đầu tư đang sụt giảm.
- Lưu tất cả các nhà đầu tư đang có dấu hiệu giảm tài sản vào một vector dữ liệu là decrease.investor và là kết quả của bài toán.

- Source code :

```
i = 7
count = 0
while(i <= 55){
  money <- tapply(as.numeric(data.diaDanh[,i]),data.diaDanh[,2],sum)
  if(i==7){
    investors <- matrix(money)
  }
  else {
    investors <- cbind(investors,money)
  }

  i = i + 6
  count = count + 1
}
colnames(investors) <- c(1:count)
investors <- t(investors)
investors <- as.data.frame(investors)
investors <- cbind.data.frame(year, investors)

#du doan ket qua
decrease.investor = ""
for(i in 2 : 199){
  model <- lm(investors[[i]] ~ year, data = investors)
  int.predict <- predict(model, newdata = data.frame(year = 2023))
  print(int.predict)
  print(investors[9,i])
  int.predict <- investors[9,i] - int.predict
  if(int.predict > 0){
    decrease.investor <- c(decrease.investor, colnames(investors[i]))
  }
}
```

- Kết quả chạy code : Bảng kết quả investors : 11 hàng đầu tiên.

btl.R × investors ×

Filter Cols: « < 1 - 50 > »

	year	Bùi Minh Tuấn	Bùi Pháp	Bùi Quang Ngọc	Bùi Thị Hương	Bùi Văn Hữu	Cao Thị Ngọc Dung	Cô Gia Thọ	Chu Thị Bình	Chu Văn An	David Cam Hao Ong
1	2014	100	611	606	81	158	306	92	1608	102	68
2	2015	120	570	563	99	187	304	114	1559	121	87
3	2016	152	769	765	128	205	411	142	2634	155	112
4	2017	247	1516	1512	194	368	836	232	4466	263	169
5	2018	358	2057	1964	292	526	1267	329	9614	363	243
6	2019	352	2110	2090	272	571	1283	323	8647	359	226
7	2020	437	3141	2831	341	780	1699	407	10946	452	293
8	2021	1050	6529	6279	872	1658	3239	967	24976	1054	710
9	2022	945	6230	6084	759	1441	3032	850	14372	960	647

Kết quả vector decrease.investor :

```
> decrease.investor
[1] ""              "Đặng Ngọc Lan"
>
```

Vậy trong 200 nhà đầu tư, chỉ duy có Đặng Ngọc Lan là được dự đoán có tài sản giảm trong năm 2023, tuy nhiên mô hình vẫn mang tính tương đối, ta xem xét hệ số tương quan giữa giá trị tài sản của bà Đặng Ngọc Lan và năm tương ứng.
Vì vậy bà Đặng Ngọc Lan vẫn sẽ là người có tài sản giảm trong 2023.

7) **Tuổi của nhà đầu tư chứng khoán từ 45 trở xuống sẽ tăng hay giảm trong tương lai**

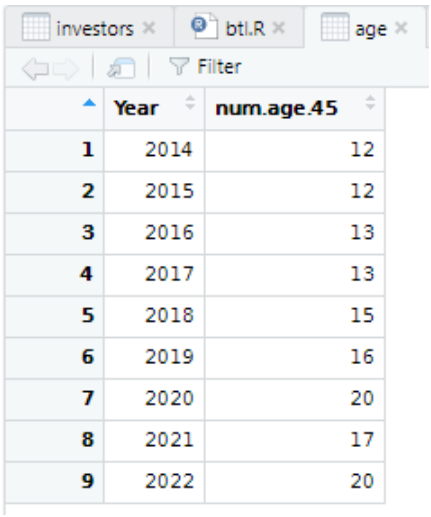
- Trình bày cách làm:
  - Dựa vào dataframe data.tuoi đã được xử lý ở bài trên - là tập hợp tuổi của các nhà đầu tư qua từng năm – ta tiến hành lọc số người có tuổi từ 45 trở xuống trong 9 năm từ 2014 đến 2022, sau đó lưu kết quả vào một dataframe tên là age. Ta có bảng thống kê như sau :

investors × btl.R × age × data.tuoi ×

Filter

	2014	2015	2016	2017	2018	2019	2020	2021	2022
age <= 45	12	12	13	13	15	16	20	17	20

- Nếu dùng bảng thống kê trên, ta hoàn toàn không thể tìm được mô hình hồi quy bởi ta đang cần sự tương quan giữa năm và số lượng người có tuổi từ 45 trở xuống, tức ta chỉ cần 2 vector (2 cột) : một cột đại diện cho năm và một cột đại diện cho số người có tuổi <= 45. Tuy nhiên trong bảng này mỗi vector (mỗi cột) là một năm có nghĩa ta chỉ có thể tìm mô hình hồi quy cho số người có tuổi <= 45 của năm này với năm khác. Trái với ý định ban đầu của ta. Vì thế ta tiến hành xoay trục dữ liệu bằng hàm pivot-longer() để đưa age về định dạng mong muốn.



	Year	num.age.45
1	2014	12
2	2015	12
3	2016	13
4	2017	13
5	2018	15
6	2019	16
7	2020	20
8	2021	17
9	2022	20

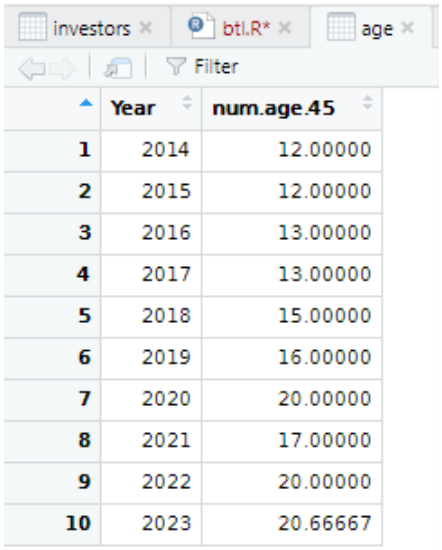
- Cuối cùng, dùng hàm lm() tạo mô hình hồi quy tuyến tính line.mode, dùng mô hình này kết hợp predict() để dự đoán số người có tuổi <= 45 trong năm 2023. Vẽ đồ thị cột biểu thị số lượng nhà đầu tư trẻ qua 9 năm kết hợp năm 2023. Đưa ra nhận xét tổng quan về sự tăng giảm số lượng nhà đầu tư có tuổi từ 45 trở xuống.

- Source code :

```r
#loc so nguoi co tuoi tu 45 tro xuong
age <- c()
for(i in 1:9){
  age <- cbind(age,length(which(data.tuoi[i] <= 45 & data.tuoi[i] != 0)))
}


colnames(age) <- c(2014:2022)
rownames(age) <- "age <= 45"
age <- as.data.frame(age)
#xoay truc di lieu

age <- age %>% pivot_longer(
  cols = 1:9,
  names_to = "Year",
  values_to = "num.age.45"
  )
age$Year <- as.numeric(age$Year)


#du doan ket qua :

correlation <- cor(age$Year, age$num.age.45)
line.mode <- lm(num.age.45 ~ Year, data = age)
line.mode
correlation
pre <- predict(line.mode, newdata = data.frame(Year = 2023))

```

```
#them 1 nam 2023 de ve bieu do
age <- rbind(age,c(2023,pre))

#ve bieu do nhan xet :
ggplot(data = age, mapping = aes( x = Year, y = num.age.45, fill = num.age.45))+
geom_col() +
  labs( title = "bieu do tuoi duoi 45", x = "Nam" , y = "So luong")+
  theme_minimal()+
  scale_fill_continuous(name = "<= 45")
```

- Kết quả chạy code : Khung dữ liệu age :



investors | btl.R* | age

Filter

	Year	num.age.45
1	2014	12.00000
2	2015	12.00000
3	2016	13.00000
4	2017	13.00000
5	2018	15.00000
6	2019	16.00000
7	2020	20.00000
8	2021	17.00000
9	2022	20.00000
10	2023	20.66667

Hệ số tương quan correlation: Ta có : correlation : $0.9237604 > 0.8$

```
> corvariant
[1] 0.9237604
```

Điều này cho ta thấy giữa số lượng người có tuổi 45 trở xuống và năm tương ứng có mối quan hệ gần như tuyến tính. Vậy 2 đại lượng này có thể triển khai dưới dạng một phương trình đường thẳng. Ta tìm hệ số của đường thẳng này thông qua mô hình line.mode.
Mô hình line.mode :

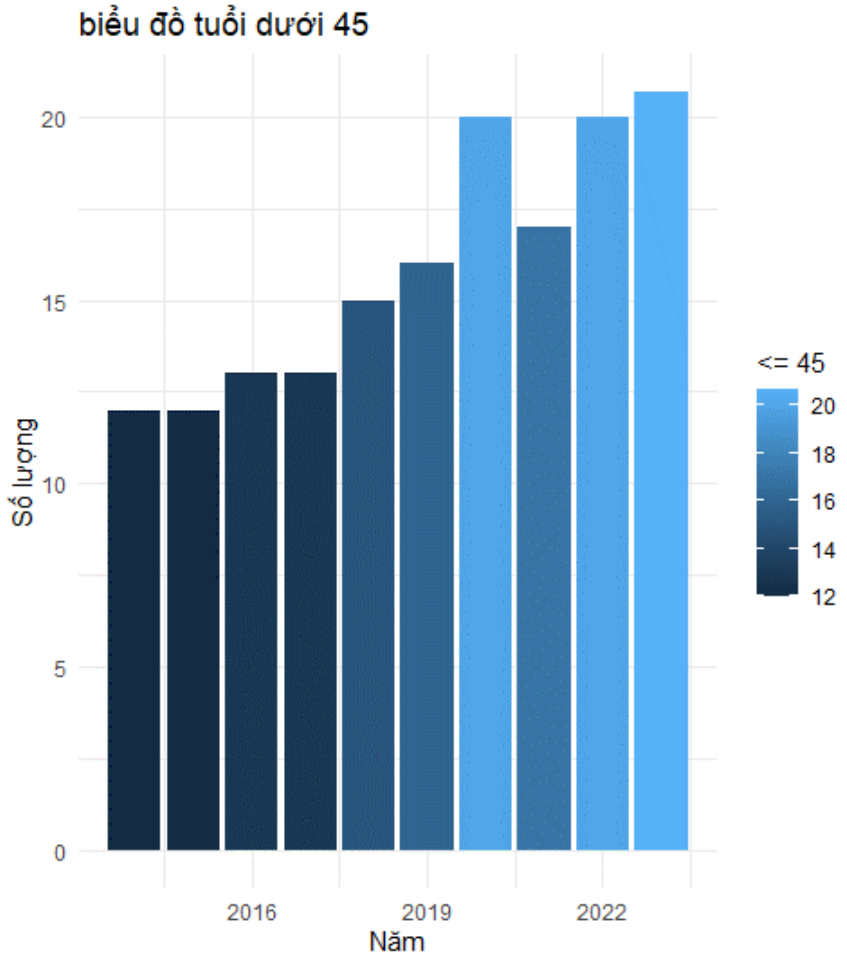
```
> line.mode

Call:
lm(formula = num.age.45 ~ Year, data = age)

Coefficients:
(Intercept)         Year
  -2137.200        1.067
```

Vậy sau cùng, hai đại lượng năm và số người dưới tuổi 45 được đặc trưng qua phương trình :

$$y = -2137.2 + 1.067x$$

Với : y : Số lượng doanh nhân dưới 45 tuổi
x : Năm tương ứng.

Biểu đồ :


biểu đồ tuổi dưới 45
Số lượng
0
5
10
15
20
2016
2019
2022
Năm
<= 45
20
18
16
14
12

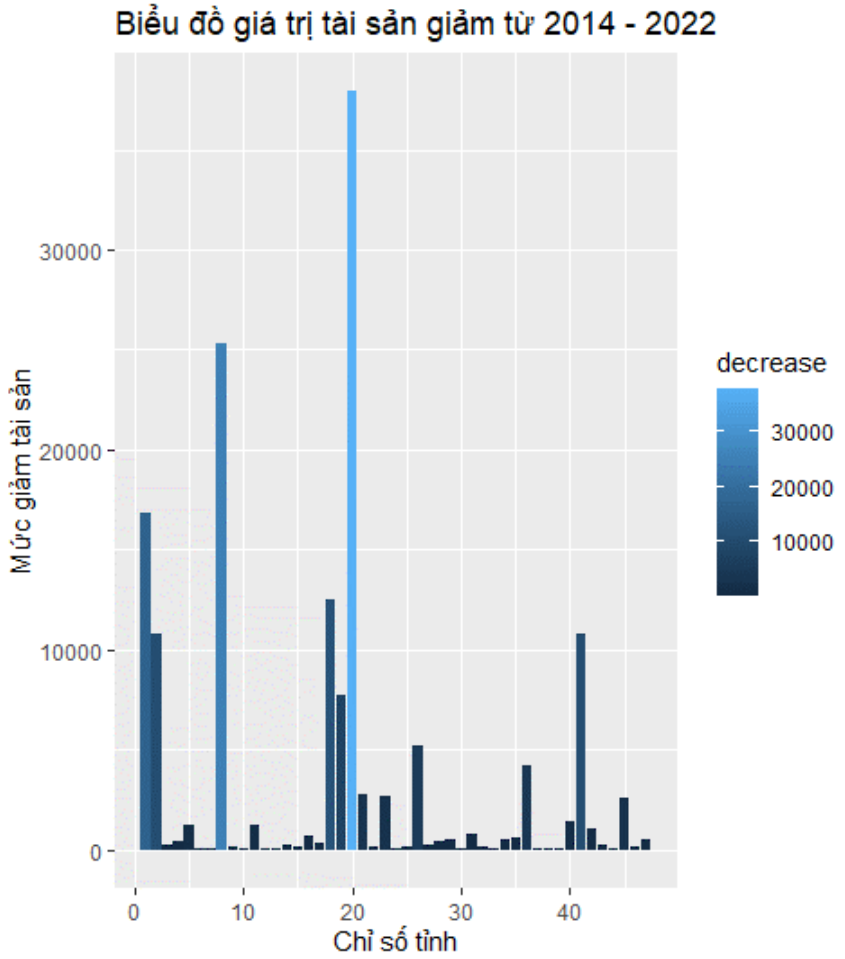
9) **Cho nhận xét của các bạn dựa vào các tập mẫu mà nhóm đã phân tích**

- Nhận xét về tỉnh có giá trị tài sản thay đổi giảm mạnh nhất Từ câu 3 ta có được bảng số

	provice	decrease	index
20	Hà Tĩnh	37915	20
21	Hải Dương	2806	21
22	Hải Hưng	195	22
23	Hải Phòng	2741	23
24	Hàn Quốc	48	24
25	Hậu Giang	129	25
26	Hồ Chí Minh	5186	26
27	Hưng Yên	288	27
28	Lâm Đồng	531	28
29	Lào Cai	459	29
30	Malaysia	106	30
31	Nam Định	771	31
32	Nghệ An	187	32
33	Quảng Bình	112	33
34	Quảng Nam	526	34
35	Quảng Ngãi	4269	35
36	Quảng Ninh	635	36
37	Quảng Trị	105	37
38	Sơn La	104	38
39	Tây Ninh	106	39
40	Thái Bình	10811	40
41	Thanh Hóa	1066	41
42	Thừa Thiên Huế	302	42
43	Tiền Giang	1462	43
44	Trà Vinh	114	44
45	Trung Quốc	2629	45
46	Vĩnh Long	204	46
47	Vĩnh Phúc	540	47

	provice	decrease	index
1	0	16886	1
2	Ấn Độ	249	2
3	An Giang	10849	3
4	Bắc Giang	419	4
5	Bắc Ninh	1219	5
6	Bến Tre	113	6
7	Bình Định	25288	7
8	Bình Dương	105	8
9	Bình Thuận	145	9
10	Cà Mau	111	10
11	Đà Nẵng	1272	11
12	Đài Loan	112	12
13	Đồng Nai	106	13
14	Đồng Tháp	221	14
15	Gia Lai	160	15
16	Hà Bắc	676	16
17	Hà Nam	392	17
18	Hà Nam Ninh	12524	18
19	Hà Nội	7736	19

liệu thống kê về số tài sản các tỉnh giảm qua các năm từ năm 2014-2022.
Từ bảng trên ta có được biểu đồ cột biểu diễn số tài sản các tỉnh giảm qua các năm



Qua biểu đồ trên ta có thể thấy được hầu hết ở các tỉnh trên cả nước thì lượng tài sản giảm là khá thấp và khá đồng đều nhau.Nhưng đặc biệt những tỉnh như là An Giang, Bình Định, Hà Nam Ninh, Hà Tĩnh, Thái Bình đều có lượng tài sản giảm đi qua từng năm là khá lớn, Trong đó lớn nhất là tỉnh Hà Tĩnh với lượng giảm là 37915, vượt trội hẳn so với các tỉnh còn lại trong nước. Một số nguyên nhân dẫn đến tình trạng trên là vì:

- Thị trường bất động sản những năm gần đây sụt giảm mạnh
- Thị trường chứng khoán ngày một lao dốc
- Do ảnh hưởng của đại dịch Covid-19 trong những năm trở lại đây dẫn đến việc kinh doanh trao đổi mua bán của nhiều nhà đầu tư bị ảnh hưởng

- Nhận xét và dự đoán nhà đầu tư sẽ có tài sản giảm trong năm 2023
  Như đã phân tích ở trên, ta dự đoán được trong năm 2023 người có tài sản giảm sẽ là bà Đặng Ngọc Lan.

```
> cor(investors$`Đặng Ngọc Lan`, investors$year)
[1] 0.9071731
>
```

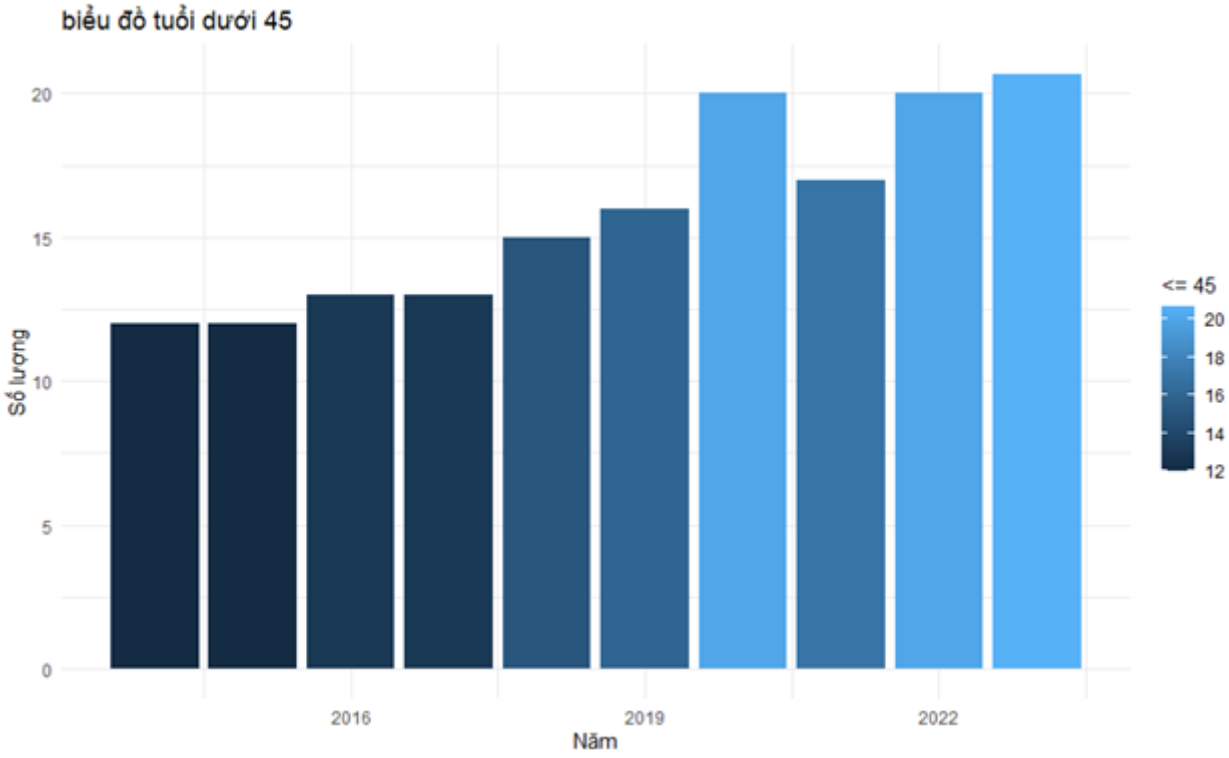
Hệ số tương quan cho thấy giá trị tài sản của bà Lan có quan hệ mật thuyết gần như tuyến tính theo từng năm, dù vậy vẫn không thể kết luận điều gì bởi mức tăng hay giảm tài sản không liên quan tới năm tương ứng. Vì vậy, để biết chính xác rằng liệu có hay không sự giảm tài sản của bà Lan theo năm ta phải tiến hành kiểm định một số giả thuyết. Nhưng ở phần này, ta sẽ không kiểm định bất kì giả thuyết nào để đơn giản hóa bài toán, xem như tất cả các giả thuyết đều thỏa mãn.
Một số nguyên nhân chính dẫn đến việc tài sản bà Đặng Ngọc Lan giảm:

- Mua bán cổ phiếu chứng khoán không hợp lí gây thua lỗ
- Chồng bà là ông Nguyễn Đức Kiên (Bầu Kiên) hiện đang phải thụ án tù nên tài sản chung của 2 vợ chồng bị niêm phong khá đáng kể

- Nhận xét tuổi của nhà đầu tư chứng khoán từ 45 trở xuống sẽ tăng hay giảm trong tương lai

	Year	num.age.45
1	2014	12.00000
2	2015	12.00000
3	2016	13.00000
4	2017	13.00000
5	2018	15.00000
6	2019	16.00000
7	2020	20.00000
8	2021	17.00000
9	2022	20.00000
10	2023	20.66667

Qua tập dữ liệu đã có, ta thống kê được số lượng nhà đầu tư dưới 45 tuổi qua các năm từ năm 2014-2022. Từ đó ta sử dụng hồi quy tuyến tính để dự đoán số nhà đầu tư dưới 45 tuổi trong năm 2023.
Từ bảng thống kê trên ta có được biểu đồ cột thể hiện số nhà đầu tư dưới 45 tuổi qua các năm.



Qua biểu đồ trên, ta thấy số nhà đầu tư dưới 45 tuổi có xu hướng tăng dần qua từng năm, vào năm 2021 có chút sụt giảm nhưng nhanh chóng phục hồi ngay năm kế tiếp và tiếp tục tăng vào năm 2023 dựa vào hàm dự đoán đã tính toán được. Trong thời đại công nghệ 4.0 và dần tiến đến 5.0, khi mà công nghệ đang dần trở thành một thứ tất yếu trong hoạt động mua bán, trao đổi giữa các nhà đầu tư, giới trẻ ngày càng dễ dàng để tiếp cận, tìm hiểu và học hỏi về các hình thức đầu tư, những phương thức trao đổi, kinh doanh khác nhau

và bằng sự nhanh nhạy, sự nhiệt huyết của tuổi trẻ mà giờ đây không khó để chúng ta bắt gặp những nhà đầu tư trẻ tuổi đầy tài năng và đầy triển vọng. Bên cạnh đó, so sự tác động của dịch bệnh Covid-19, mọi hoạt động mua bán trao đổi đã phải dừng lại trong một thời gian dài, nhiều nhà đầu tư lớn tuổi do không quen thuộc với việc mua bán trao đổi qua những chiếc smartphone hay laptop mà họ dần tụt lại phía sau và đó là cơ hội để các nhà đầu tư trẻ vươn lên với thứ mà họ đã quá đỗi ưa thích và quen thuộc đó là mua bán trao đổi online như bán hàng qua các trang bán hàng online.

7 Hướng dẫn và yêu cầu

7.1 Hướng dẫn

- Cài đặt đồng thời cả R và Rstudio.
- Đọc kĩ và xử lý lại tất cả những thí dụ đã có trong file mẫu.
- Tìm hiểu kĩ cách soạn thảo văn bản bằng LaTeX và cách sử dụng phần mềm R trong các file hướng dẫn và tìm hiểu thêm trong các tài liệu khác.
- Tạo một folder chung chứa mọi thứ cần thiết để share giữa các thành viên trong nhóm trên các cloud services như Google Drive hay Dropbox,...
- Dùng Doodle để lên kế hoạch họp nhóm.
- Dùng Trello để quản lý project.

7.2 Yêu cầu

Mỗi nhóm, từ 3 đến 6 sinh viên, đề xuất giải pháp. Nhóm cần nộp báo cáo trình bày về lời giải cho các câu hỏi và kết quả thực nghiệm. Đồng thời, nhóm cũng cần nộp source code, và trình bày các kết quả của nhóm trong khoảng 5 minutes.

Báo cáo và slide trình bày cần được viết dưới dạng LaTeX.

- Thời gian làm bài: Từ ngày 14/11/2021 – 18g00 ngày 03/12/2022.

  Đối với mỗi bài toán, yêu cầu sinh viên trình bày lời giải theo lối truyền thống, sử dụng các công thức, kết quả lý thuyết trong phần kiến thức chuẩn bị. Đồng thời, sau đó trình bày kết quả tính toán và biểu đồ minh họa bằng R.
- Trình bày cả code R và kết quả tính toán trong R giống như file mẫu.
- Viết báo cáo theo đúng **bố cục như trong file mẫu** bằng LaTeX.
- Mỗi nhóm khi nộp bài **cần phải nộp theo file log (nhật ký)** ghi rõ: tiến độ công việc, phân công nhiệm vụ, trao đổi của các thành viên,...
- Mỗi nhóm nộp 1 **video** thể hiện phần trình bày của nhóm (15-20 phút)

7.3 Nộp bài

- SV chỉ nộp bài qua hệ thống BKEL: nén tất cả các file cần thiết (file .tex, file .R, ...) thành một file tên là "*NHOM-MADE.zip*": 1-3456.zip và nộp trong mục Assignment.
- Lưu ý: mỗi nhóm **chỉ cần một thành viên là nhóm trưởng nộp bài.**

8 Cách đánh giá và xử lý gian lận

8.1 Đánh giá

Mỗi bài làm sẽ được đánh giá như sau.

Nội dung	Tỉ lệ điểm (%)
Giải đúng các bài toán bằng công thức và lập luận	30%
Các lệnh (hàm) R được sử dụng đúng đắn và hợp lý	30%
Trình bày kiến thức chuẩn bị rõ ràng, phù hợp	20%
Trình bày văn bản đẹp, đúng chuẩn	20%

8.2 Xử lý gian lận

Bài tập lớn phải được sinh viên (nhóm) TỰ LÀM. Sinh viên (nhóm) sẽ bị coi là gian lận nếu:

- Có sự giống nhau bất thường giữa các bài thu hoạch (nhất là phần kiến thức chuẩn bị). Trong trường hợp này, TẤT CẢ các bài nộp có sự giống nhau đều bị coi là gian lận. Do vậy sinh viên (nhóm) phải bảo vệ bài làm của mình.
- Sinh viên (nhóm) không hiểu bài làm do chính mình viết. Sinh viên (nhóm) có thể tham khảo từ bất kỳ nguồn tài liệu nào, tuy nhiên phải đảm bảo rằng mình hiểu rõ ý nghĩa của tất cả những gì mình viết.

Bài bị phát hiện gian lận thì sinh viên sẽ bị xử ý theo quy định của nhà trường.

9 Phân công công việc

9.1 Phân công

Họ và tên	MSSV	Phân công công việc
Thái Anh Khương	2113806	Bài vi) (câu 9), xử lí và viết code latex
Nguyễn Trần Quang Vũ	2115325	Bài vi) (câu 3, 6, 7)
Nguyễn Trường Thản	2114798	Bài i), ii) (câu 1 đến 4), v (xử lý số liệu)
Lê Phan Quốc Vũ	2115321	Bài iv
Nguyễn Hữu Thông	2114917	Bài iii
Phạm Ngọc Khai	2113650	Bài v

9.2 Nhật kí công việc

Thời gian	Nội dung cuộc họp
15/11/2022	Họp bàn về bài tập lớn và chia việc giữa các thành viên
22/11/2022	Họp trao đổi tiến độ công việc giữa các thành viên
29/11/2022	Tổng hợp số liệu và điểm lại phần việc chưa làm xong
2/12/2022	Tổng hợp bài làm của các thành viên

Tài liệu

[Dal] Dalgaard, P. *Introductory Statistics with R.* Springer 2008.

[K-Z] Kenett, R. S. and Zacks, S. *Modern Industrial Statistics: with applications in R, MINITAB and JMP,* 2nd ed., John Wiley and Sons, 2014.

[Ker] Kerns, G. J. *Introduction to Probability and Statistics Using R,* 2nd ed., CRC 2015.