

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



CẤU TRÚC RỜI RẠC CHO KHMT

Báo cáo Bài tập lớn

Ứng dụng thống kê
và khảo sát kết quả của bài tập online cho phép nộp bài nhiều lần

GVHD:	Huỳnh Tường Nguyên Trần Tuấn Anh Nguyễn Ngọc Lẽ	
Lớp:	L03	Nhóm: 18
Sinh viên:	Võ Hồng Phúc Trương Hồng Hoa Nguyễn Duy Uyên Lương Thị Quỳnh Hương Dinh Lê Trần Đương Ngọc Yến	1911881 1911185 1912410 1911314 1912267 1912498



Mục lục

1 Cơ sở lý thuyết	2
2 Tổng quan về R	2
3 Đề bài	3
3.1 Động cơ nghiên cứu	3
3.2 Mục tiêu	3
3.3 Mô tả dữ liệu	3
3.4 Nội dung bài tập	3
3.5 Nhiệm vụ	4
3.6 Yêu cầu	4
4 Kết quả phân tích	4
4.1 Câu 1: Xác định số lượng sinh viên trong tập mẫu	4
4.2 Câu 2: Nhóm câu hỏi liên quan đến điểm số của các sinh viên	6
4.3 Câu 3: Nhóm câu hỏi liên quan đến số lần nộp bài	32
4.4 Câu 4: Nhóm câu hỏi liên quan đến thời gian, tần suất nộp bài của các sinh viên	66
4.5 Câu 5: Nhóm câu hỏi liên quan điểm tổng hợp, điểm trung bình của sinh viên	90
4.6 Câu 7: Nhóm câu hỏi liên quan sinh viên học đối phó	103
4.7 Câu 9: Nhóm câu hỏi liên quan sinh viên thông minh	117
4.8 Câu 10: Nhóm câu hỏi liên quan sinh viên chủ động	131
4.9 Câu 11: Phần giao của các loại sinh viên được đánh giá	143
Tài liệu tham khảo	163



1 Cơ sở lý thuyết

Một số chỉ số thống kê:

- Số trung bình (mean) là giá trị trung bình của dấu hiệu trong mẫu

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$$

- Số trung vị (median) là số chính giữa của phân bố. Nếu số lượng phần tử là lẻ thì trung vị chính là phần tử nằm ở chính giữa. Nếu số lượng phần tử là chẵn thì trung vị được tính bằng trung bình cộng của 2 phần tử chính giữa.

- Tần số (frequency) là số lần xuất hiện của mỗi giá trị trong mẫu số liệu, ký hiệu f_i

- Tần suất là tỉ số giữa tần số và kích thước mẫu

$$p_i = \frac{f_i}{n}$$

- Tần suất tích lũy là tổng các tần số từ trên xuống

$$S_i = \sum f_i$$

- Phương sai (variance) của một biến ngẫu nhiên để đo sự phân tán thống kê của biến đó, nó hàm ý các giá trị của biến đó thường ở cách giá trị kỳ vọng bao xa.

$$S^2 = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2$$

- Dộ lệch chuẩn (standard deviation) dùng để đo mức độ phân tán của một tập dữ liệu đã được lập thành bảng tần số. Có thể tính độ lệch chuẩn bằng cách lấy căn bậc hai của phương sai.

$$S = \sqrt{S^2}$$

- Dộ méo lệch (skewness) đo lường mức độ đối xứng của một phân bố xác suất. Giá trị tuyệt đối của độ lệch càng cao thì phân phối đó càng bất đối xứng. Một phân phối đối xứng có độ lệch bằng 0.

$$Skew = \frac{\frac{1}{n} \sum_{i=1}^n (X_i - \bar{X})^3}{\sqrt{\frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X})^2}^3}$$

- Dộ nhọn (kurtosis) đo mức độ tập trung của quan sát quanh trung tâm của phân phối so với hai bên.

$$Kurt = \frac{\frac{1}{n} \sum_{i=1}^n (X_i - \bar{X})^4}{(\frac{1}{n} \sum_{i=1}^n (X_i - \bar{X})^2)^2}$$

- Tứ phân vị là đại lượng mô tả sự phân bố và sự phân tán của tập dữ liệu.

Tứ phân vị có 3 giá trị, đó là tứ phân vị thứ nhất, thứ nhì, và thứ ba. Ba giá trị này chia một tập hợp dữ liệu (đã sắp xếp dữ liệu theo trật tự từ bé đến lớn) thành 4 phần có số lượng quan sát đều nhau.

Giả sử số quan sát là n.

Giá trị tứ phân vị thứ hai Q2 chính bằng giá trị trung vị

Giá trị tứ phân vị thứ nhất Q1 bằng trung vị phần dưới: $Q_1 = 25(n+1)/100$ Giá trị tứ phân vị thứ ba Q3 bằng trung vị phần trên: $Q_3 = 75(n+1)/100$

2 Tổng quan về R

- R là một công cụ thống kê và phân tích dữ liệu. Nó là một ngôn ngữ lập trình. Ngôn ngữ R có thể sử dụng nó cho bất kỳ hệ điều hành nào một cách miễn phí.
- Ngôn ngữ R là mã nguồn mở nên ai trong chúng ta cũng có thể phân tích source code để hiểu được chính xác cách R vận hành. Đồng thời, R có thể tích hợp được với ngôn ngữ khác (C,C++), cho phép chúng ta tương tác với nhiều nguồn dữ liệu khác.
- R có một kho package phong phú và rộng rãi.
- Tính toán phân tán có hiệu năng cao cho ngôn ngữ R, chia công việc thành nhiều task nhỏ cho nhiều node để chạy đồng thời, nhờ đó giảm thời gian để chạy phân tích một tập dữ liệu lớn.



3 Đề bài

3.1 Động cơ nghiên cứu

Phân tích & thống kê dữ liệu qua các lần nộp bài của sinh viên nhằm giúp giáo viên đánh giá toàn diện hiệu quả dạy và học online của môn học trong thời gian qua:

- Đánh giá điểm mạnh, điểm yếu trong hệ thống kiến thức của sinh viên.
- Xây dựng kế hoạch cải thiện môn học trong thời gian còn lại của môn học.
- Định hướng và điều chỉnh phương pháp giảng dạy.
- Bổ sung học liệu cho đề cương trong tương lai.

3.2 Mục tiêu

- a Khai phá dữ liệu từ hệ thống kiểm tra online.
- b Bắt đầu với các bài toán thống kê đơn giản từ những dữ liệu được cung cấp.
- c Tìm hiểu ý nghĩa đối với các dữ liệu thực tế.
- d Nâng cao kỹ năng lập trình và giải quyết vấn đề, tích hợp với các hệ thống quản lý và cải thiện chất lượng dạy và học.

3.3 Mô tả dữ liệu

Mỗi nhóm nhận được 1 *MD* là mã đề riêng cho mỗi nhóm (gồm 4 ký số) không trùng nhau, và 24 files chứa thông tin điểm số qua các lần nộp. Các sinh viên sẽ thực hiện yêu cầu đề bài dựa trên các thông tin được cung cấp: Nội dung file:

1. *tid* là mã số bài tập.
2. **Mã số ID** ta gọi là *uid* là mã số định danh sinh viên nộp bài.
3. **Tình trạng**: Đề hoàn thành hoặc chưa bao giờ giải.
4. **Đã bắt đầu vào lúc, Đã hoàn thành**: Thời gian theo dạng “d B Y I:M p” là thời gian bắt đầu và kết thúc làm bài.
5. **Thời gian thực hiện**: Khoảng thời gian làm bài
6. **Diểm/10**: Tổng số điểm của các quiz.
7. **Q.i/1** là điểm số của bài quiz chỉ 0 hoặc 1.

3.4 Nội dung bài tập

- Xác định số lượng sinh viên trong tập mẫu
- Nhóm câu hỏi liên quan đến điểm số của các sinh viên
- Nhóm câu hỏi liên quan đến số lần nộp bài
- Nhóm câu hỏi liên quan đến thời gian, tần suất nộp bài của các sinh viên.
- Nhóm câu hỏi phân loại sinh viên.



- Sinh viên **siêng năng**
 - Sinh viên học **đối phó**
 - Sinh viên **giỏi**
 - Sinh viên **thông minh**
 - Sinh viên **chủ động**
- Xác định phần giao của các loại sinh viên đánh giá ở trên và vẽ biểu đồ thống kê minh họa.

3.5 Nhiệm vụ

1. Cài đặt R, RStudio
2. Soạn thảo Latex
3. Tạo tài nguyên dùng chung
4. Lên kế hoạch nhóm
5. Quản lí tiến độ

3.6 Yêu cầu

1. Báo cáo Bài tập lớn
 - (a) Lời giải
 - (b) Kết quả thực nghiệm
 - (c) Source code R, kết quả tính toán
2. File logs đánh giá tiến độ, trao đổi

4 Kết quả phân tích

4.1 Câu 1: Xác định số lượng sinh viên trong tập mẫu

Lời giải:

Trước hết, chúng ta sẽ khai báo một số thư viện cần thiết cho bài tập, đồng thời import file vào môi trường R thông qua biến *c*. Để thuận tiện cho việc thực hiện các hàm cũng như tính toán chúng ta sẽ đổi tên các cột cũng như chuyển điểm số từ dạng dấu "," sang dấu ".".
Để khai báo thư viện ta dùng lệnh library và dùng gsub() để đổi từ dấu "," sang dấu "..".

```
library(xlsx)
library(readxl)
library(dplyr)
library(lubridate)
library(utf8)
library(e1071)
library(ggplot2)
c <- read_xlsx("filename.xlsx")
colnames(c) <- c("ID", "Status", "Stime", "Etime", "Time", "Diem", "Q1", "Q2",
"Q3", "Q4", "Q5", "Q6", "Q7", "Q8", "Q9", "Q10")
c$Diem=as.numeric(gsub(",",".", gsub("\\.", "", c$Diem)))
c$Q1=as.numeric(gsub(",",".", gsub("\\.", "", c$Q1)))
c$Q2=as.numeric(gsub(",",".", gsub("\\.", "", c$Q2)))
c$Q3=as.numeric(gsub(",",".", gsub("\\.", "", c$Q3)))
c$Q4=as.numeric(gsub(",",".", gsub("\\.", "", c$Q4)))
c$Q5=as.numeric(gsub(",",".", gsub("\\.", "", c$Q5)))
c$Q6=as.numeric(gsub(",",".", gsub("\\.", "", c$Q6)))
c$Q7=as.numeric(gsub(",",".", gsub("\\.", "", c$Q7)))
c$Q8=as.numeric(gsub(",",".", gsub("\\.", "", c$Q8)))
c$Q9=as.numeric(gsub(",",".", gsub("\\.", "", c$Q9)))
c$Q10=as.numeric(gsub(",",".", gsub("\\.", "", c$Q10)))
```

Kết quả sau khi đổi tên như sau:

ID	Status	Stime	Etime	Time	Diem	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
1	Đã hoàn thành	29 March 2020 9:03 AM	29 March 2020 9:05 AM	2 phút 15 giây	9.0	1	1	1.0	1.0	1.0	1	1.0	1.0	1	0
2	Đã hoàn thành	29 March 2020 9:25 AM	29 March 2020 9:28 AM	2 phút 52 giây	10.0	1	1	1.0	1.0	1.0	1	1.0	1.0	1	1
3	Đã hoàn thành	29 March 2020 9:35 AM	29 March 2020 9:40 AM	4 phút 53 giây	10.0	1	1	1.0	1.0	1.0	1	1.0	1.0	1	1
4	Đã hoàn thành	29 March 2020 10:10 AM	29 March 2020 10:13 AM	3 phút	10.0	1	1	1.0	1.0	1.0	1	1.0	1.0	1	1
5	Đã hoàn thành	29 March 2020 11:02 AM	29 March 2020 11:17 AM	14 phút 57 giây	10.0	1	1	1.0	1.0	1.0	1	1.0	1.0	1	1
6	Đã hoàn thành	29 March 2020 11:26 AM	29 March 2020 11:29 AM	3 phút 10 giây	10.0	1	1	1.0	1.0	1.0	1	1.0	1.0	1	1
7	Đã hoàn thành	29 March 2020 11:51 AM	29 March 2020 11:54 AM	3 phút 4 giây	10.0	1	1	1.0	1.0	1.0	1	1.0	1.0	1	1
8	Đã hoàn thành	29 March 2020 2:14 PM	29 March 2020 2:18 PM	3 phút 32 giây	10.0	1	1	1.0	1.0	1.0	1	1.0	1.0	1	1
9	Đã hoàn thành	29 March 2020 3:39 PM	29 March 2020 3:44 PM	5 phút 29 giây	10.0	1	1	1.0	1.0	1.0	1	1.0	1.0	1	1
10	Đã hoàn thành	29 March 2020 5:07 PM	29 March 2020 5:15 PM	8 phút 22 giây	9.0	1	1	1.0	1.0	1.0	1	1.0	1.0	1	0
11	Đã hoàn thành	29 March 2020 5:24 PM	29 March 2020 5:24 PM	42 giây	10.0	1	1	1.0	1.0	1.0	1	1.0	1.0	1	1
12	Đã hoàn thành	29 March 2020 5:36 PM	29 March 2020 5:39 PM	2 phút 43 giây	10.0	1	1	1.0	1.0	1.0	1	1.0	1.0	1	1
13	Đã hoàn thành	29 March 2020 7:39 PM	29 March 2020 7:45 PM	5 phút 12 giây	10.0	1	1	1.0	1.0	1.0	1	1.0	1.0	1	1
14	Đã hoàn thành	29 March 2020 11:18 PM	29 March 2020 11:23 PM	5 phút 20 giây	9.0	1	1	1.0	1.0	1.0	1	1.0	1.0	1	0
15	Đã hoàn thành	30 March 2020 8:23 AM	30 March 2020 8:30 AM	7 phút 15 giây	8.5	0	1	1.0	1.0	1.0	1	0.5	1.0	1	1

Để xác định số lượng sinh viên trong tập mẫu, trước hết ta sẽ tạo một dataframe sắp xếp ID của sinh viên theo thứ tự tăng dần (hoặc giảm dần). Lúc này thông tin những lần nộp bài của cùng một sinh viên sẽ nằm kề nhau. Sau đó dùng một vòng lặp để liệt kê từng mã số ID trong tập mẫu. Nếu nhiều hơn 1 lần nộp bài thì ta chỉ tính ID một lần và bỏ qua các lần còn lại. Ta sẽ dùng 1 vector để lưu ID của sinh viên. Độ dài của vector chính là số lượng sinh viên trong tập mẫu.



```
sapxep_ID=arrange(c, ID) #Tạo một dataframe sắp xếp dữ liệu ID theo thứ tự tăng dần
n=nrow(c) #số hàng = số lần nộp + 1
IDlist=c()
IDlist <- c(IDlist, sapxep_ID[["ID"]][1])
for( i in 1:(n-2)){
  if(sapxep_ID[["ID"]][i] != sapxep_ID[["ID"]][i+1])
    IDlist <- c(IDlist, sapxep_ID[["ID"]][i+1])
}
View(IDlist)
n_sv=length(IDlist) #Số SV nộp bài
n_sv #Xuất ra kết quả trên màn hình
```

Kết quả xuất ra màn hình:

Output file 6:

```
[1] 322
```

Output file 9:

```
[1] 307
```

Output file 10:

```
[1] 289
```

Output file 12:

```
[1] 280
```

4.2 Câu 2: Nhóm câu hỏi liên quan đến điểm số của các sinh viên

Để thuận tiện cho việc giải quyết các yêu cầu đề bài, ta thực hiện một số việc sau. Đầu tiên, tạo một dataframe mới không chứa dòng cuối giá trị trung bình của file gốc (trong bài làm này mẫu mới được tạo tên là New). Tiếp đó, tao một dataframe chứa ID (ID), số lần nộp (NoS), điểm số tổng kết (Mark) của sinh viên của từng sinh viên là Submissions sử dụng sapxep_ID là tập con của New chỉ chứa điểm những sinh viên đã nộp bài và được sắp xếp ID theo thứ tự tăng dần. Với những lần nộp bài chưa gửi, ta cho điểm bằng 0, số lần nộp bằng 0. Tiếp theo, ta tạo một dataframe tên sapxep_Diem chứa thông tin từng lần nộp bài của sinh viên được sắp xếp lại theo điểm số theo thứ tự tăng dần. Ta tạo vector Marklist chứa điểm lần nộp bài đã hoàn thành.

```
n=nrow(c)
New<-c # tạo New
New<-New[-c(n),] # bỏ dòng chứa giá trị trung bình
sapxep_ID=arrange(New, ID)# tạo sapxep_ID
sapxep_ID=subset(sapxep_ID,!is.na(sapxep_ID[["Diem"]]))
ID<-c(sapxep_ID[["ID"]][1])
NoS<-c(1)
Mark<-c(sapxep_ID[["Diem"]][1])
f=1
f0=1
for(i in 2:(nrow(sapxep_ID))){
  if(sapxep_ID[["ID"]][i]!=sapxep_ID[["ID"]][i-1]) {
    ID<-c(ID,sapxep_ID[["ID"]][i])
    NoS<-c(NoS,f)
    if(f0!=0){Mark<-c(Mark,0)}
    f=1
    f0=f0+1
  }
  else {
    f=f+1
  }
  if(is.na(sapxep_ID[["Diem"]][i])) {
    f=0
    sapxep_ID[["Diem"]][i]=0
  }
  else {
    if(Mark[f0]<=sapxep_ID[["Diem"]][i]){
      Mark[f0]<-sapxep_ID[["Diem"]][i]
    }
  }
}
Submissions<-data.frame(ID,NoS,Mark) # tạo Submissions
for(i in 1:(n-1)){
  if(is.na(New[["Diem"]][i])){
    New[["Diem"]][i]=0
  }
}
sapxep_Diem=arrange(New,Diem) # tạo sapxep_Diem
Marklist<-c() # tạo Marklist
for(i in 1:(n-1)) {
  if(!is.na(c[["Diem"]][i])) {
    Marklist<-c(Marklist,c[["Diem"]][i])
  }
}
```

- a) Xác định điểm số là điểm tổng của các bài làm với mỗi câu hỏi đơn vị đều có điểm tối đa là 1 điểm.
Điểm số là các phần tử trong cột điểm của New.

```
print(New[["Diem"]])
```

Output file 6:

```
[1] 9.0 10.0 10.0 10.0 10.0 10.0 10.0 10.0 10.0 9.0 10.0 10.0 10.0 10.0 9.0 8.5 9.5 10.0 10.0
[19] 10.0 10.0 10.0 10.0 9.5 10.0 9.5 10.0 9.0 10.0 9.0 10.0 9.5 10.0 10.0 10.0 10.0 10.0 10.0 9.0
[37] 10.0 10.0 10.0 10.0 10.0 10.0 3.0 10.0 9.5 10.0 10.0 10.0 9.0 10.0 9.5 10.0 10.0 10.0 10.0 10.0
[55] 9.5 10.0 10.0 9.0 10.0 9.0 10.0 10.0 7.5 9.0 10.0 10.0 10.0 10.0 10.0 9.5 10.0 10.0 8.0 10.0
[73] 10.0 10.0 10.0 10.0 9.0 10.0 9.0 9.5 10.0 9.5 10.0 9.5 10.0 10.0 9.5 10.0 10.0 9.5 9.5 10.0 9.0
[91] 9.0 10.0 10.0 7.0 10.0 8.5 9.5 10.0 9.0 10.0 7.0 9.0 9.5 9.5 10.0 10.0 10.0 9.0 10.0 9.0 10.0
[109] 10.0 10.0 10.0 10.0 10.0 10.0 10.0 10.0 10.0 9.0 10.0 10.0 9.5 10.0 10.0 7.0 8.5 10.0
[127] 9.0 9.5 10.0 10.0 10.0 10.0 10.0 10.0 10.0 10.0 10.0 10.0 10.0 10.0 10.0 10.0 10.0 10.0 10.0 9.5
[145] 9.5 10.0 9.5 10.0 8.5 9.5 9.5 10.0 10.0 7.5 9.5 9.0 10.0 10.0 10.0 10.0 10.0 10.0 10.0 10.0
[163] 10.0 9.5 10.0 10.0 10.0 10.0 9.5 7.5 10.0 10.0 9.5 10.0 9.5 9.0 10.0 10.0 10.0 8.5 9.0
[181] 10.0 5.5 10.0 9.5 10.0 10.0 10.0 9.5 10.0 8.5 9.5 10.0 8.5 10.0 10.0 10.0 10.0 5.0 8.0
[199] 9.0 10.0 7.5 10.0 9.0 9.5 10.0 9.0 10.0 10.0 10.0 10.0 9.0 10.0 9.5 10.0 10.0 10.0 10.0 10.0
[217] 10.0 10.0 10.0 9.0 10.0 10.0 8.5 10.0 9.5 8.5 10.0 10.0 10.0 9.0 8.0 10.0 9.0 10.0 9.0 10.0
[235] 9.0 10.0 10.0 10.0 10.0 10.0 10.0 9.5 9.5 10.0 10.0 10.0 9.5 9.5 10.0 9.5 10.0 9.5 10.0 9.5
[253] 10.0 10.0 9.5 10.0 10.0 9.5 10.0 10.0 9.5 10.0 9.5 10.0 9.5 10.0 10.0 7.5 9.5 10.0
[271] 9.5 9.5 10.0 10.0 8.5 10.0 10.0 9.0 10.0 8.5 10.0 10.0 10.0 10.0 9.0 10.0 10.0 10.0 10.0 10.0
[289] 10.0 10.0 8.5 10.0 10.0 10.0 7.5 9.0 9.0 10.0 10.0 10.0 8.5 10.0 10.0 10.0 9.0 10.0 10.0
[307] 10.0 8.0 10.0 10.0 10.0 10.0 10.0 9.0 10.0 10.0 9.5 10.0 10.0 8.5 10.0 9.5 10.0 9.5 10.0 9.5
[325] 10.0 9.5 10.0 10.0 10.0 10.0 9.0 10.0 9.0 10.0 9.5 10.0 9.0 10.0 10.0 10.0 10.0 10.0 10.0 10.0
[343] 10.0 10.0 10.0 10.0 10.0 10.0 7.5 9.0 10.0 10.0 10.0 10.0 9.5 10.0 10.0 10.0 10.0 10.0 9.5
[361] 10.0 8.0 10.0 10.0 6.5 9.5 10.0 9.5 10.0 10.0 10.0 10.0 10.0 9.5 10.0 10.0 0.0 10.0 10.0
[379] 10.0 10.0 10.0 10.0 10.0 10.0 9.5 10.0 10.0 10.0 9.5 10.0 10.0 10.0 10.0 10.0 10.0 10.0 9.0 10.0
[397] 10.0 9.0 10.0 8.5 10.0 9.0 10.0 9.5 9.5 10.0 10.0 10.0 3.0 10.0 10.0 10.0 10.0 9.5 9.5
[415] 10.0 10.0 10.0 10.0 8.5 9.5 10.0 4.5 9.0 9.0 9.5 10.0 9.0 10.0 10.0 10.0 10.0 10.0 10.0 10.0
[433] 10.0 10.0 10.0 10.0 10.0 10.0 10.0 10.0 0.0
```

Output file 9:

```
[1] 9.0 10.0 10.0 9.0 10.0 10.0 10.0 10.0 9.0 8.0 10.0 10.0 8.0 10.0 7.0 10.0 5.5 9.5
[19] 8.0 10.0 10.0 9.0 8.0 10.0 8.0 9.0 10.0 10.0 10.0 10.0 8.0 10.0 10.0 9.0 10.0 10.0
[37] 9.0 10.0 8.0 10.0 9.0 10.0 10.0 9.0 10.0 8.0 9.0 10.0 9.0 10.0 9.0 10.0 9.0 10.0 10.0
[55] 10.0 9.0 10.0 8.0 10.0 10.0 8.0 10.0 9.0 10.0 10.0 8.0 10.0 9.0 10.0 10.0 8.0 10.0
[73] 9.0 10.0 9.0 10.0 10.0 9.0 10.0 9.0 10.0 8.0 10.0 7.5 10.0 10.0 9.0 10.0 8.0 10.0 9.0
[91] 9.0 9.0 10.0 7.0 10.0 10.0 8.0 9.0 10.0 9.0 10.0 9.0 10.0 9.0 10.0 9.0 10.0 10.0 10.0
[109] 10.0 10.0 10.0 9.0 10.0 8.0 8.0 10.0 10.0 7.0 10.0 9.0 10.0 10.0 10.0 9.0 10.0 10.0 8.0 10.0
[127] 8.0 10.0 10.0 9.0 10.0 10.0 9.0 9.0 9.0 10.0 10.0 10.0 10.0 9.0 9.0 10.0 10.0 10.0 9.0
[145] 10.0 10.0 10.0 9.0 10.0 9.0 8.0 10.0 10.0 9.0 10.0 8.0 10.0 10.0 10.0 9.0 10.0 10.0 10.0
[163] 9.5 10.0 10.0 10.0 9.0 10.0 10.0 10.0 10.0 10.0 10.0 10.0 10.0 9.0 10.0 8.0 10.0 9.0
[181] 10.0 8.0 10.0 10.0 10.0 10.0 10.0 10.0 8.0 10.0 10.0 7.0 10.0 10.0 2.0 8.0 8.0 10.0
[199] 10.0 8.0 9.0 10.0 10.0 9.0 10.0 9.0 9.0 10.0 9.0 10.0 9.0 10.0 8.0 10.0 9.0
[217] 10.0 10.0 10.0 9.0 10.0 8.0 10.0 10.0 10.0 9.0 10.0 10.0 10.0 7.0 10.0 10.0 9.0 10.0 9.5
[235] 8.5 9.5 9.0 10.0 10.0 9.0 10.0 9.0 10.0 10.0 10.0 9.0 10.0 10.0 7.5 9.0 9.0 10.0
[253] 8.0 10.0 10.0 10.0 10.0 10.0 9.0 10.0 9.0 10.0 8.0 10.0 9.0 10.0 9.0 10.0 9.0 10.0 8.0
[271] 10.0 8.5 10.0 10.0 0.0 8.0 9.0 9.0 9.0 10.0 10.0 9.0 10.0 8.0 10.0 10.0 10.0 10.0 9.0
[289] 9.0 10.0 9.0 10.0 9.0 10.0 8.0 9.5 10.0 0.0 10.0 7.0 10.0 10.0 9.0 10.0 9.0 10.0 10.0
[307] 9.5 10.0 10.0 10.0 10.0 8.0 10.0 9.0 10.0 9.0 10.0 9.0 10.0 9.0 10.0 8.0 10.0 9.0
[325] 6.5 10.0 7.0 10.0 10.0 9.0 10.0 10.0 6.0 10.0 10.0 9.0 9.0 9.0 10.0 10.0 9.0 10.0 9.0
[343] 8.0 9.0 10.0 8.0 10.0 9.0 10.0 10.0 9.0 10.0 9.0 8.0 10.0 8.5 10.0 9.0 10.0 9.0 10.0 10.0
[361] 10.0 10.0 9.0 10.0 9.0 10.0 9.0 9.5 10.0 10.0 10.0 9.0 9.0 10.0 9.0 9.0 10.0 9.0 10.0 9.0
[379] 10.0 8.0 10.0 9.0 10.0 7.0 9.0 10.0 10.0 7.0 9.0 10.0 8.0 10.0 10.0 10.0 10.0 7.0 10.0
[397] 10.0 10.0 7.0 10.0 10.0 9.0 10.0 10.0 9.0 9.0 8.0 10.0 9.0 10.0 9.0 9.5 10.0 10.0 9.0
[415] 10.0 10.0 0.0 10.0 10.0 8.0 10.0 9.0 8.0 10.0 10.0 9.0 10.0 9.0 9.0 10.0 8.0 10.0 9.0
[433] 0.0 8.0 10.0 9.0 10.0 9.0 10.0 9.0 8.0 10.0 10.0 10.0 8.0 10.0 10.0 9.0 10.0 9.0 10.0 9.0
[451] 9.0 10.0 10.0 10.0 9.0 10.0 9.0 10.0 9.0 10.0 8.0 10.0 8.5 10.0 9.0
```

Output file 10:

```
[1] 8.5 9.5 10.0 10.0 9.5 10.0 9.5 10.0 9.5 9.5 10.0 9.5 9.5 10.0 10.0 9.5 10.0 9.5 10.0 9.5 10.0 9.5 10.0 9.5
[19] 10.0 10.0 10.0 9.5 9.5 9.5 10.0 9.5 10.0 9.5 9.5 10.0 9.5 10.0 9.5 10.0 9.5 10.0 9.5 10.0 10.0 8.5
[37] 9.0 9.0 9.5 9.5 10.0 8.5 9.0 9.0 9.0 9.5 10.0 9.5 9.0 9.0 9.5 10.0 8.5 9.5 10.0 9.5 10.0 9.5
[55] 10.0 9.5 9.5 9.5 9.5 9.5 10.0 9.0 9.5 10.0 7.5 9.5 10.0 9.5 9.5 9.5 9.5 10.0 9.5 10.0 9.5
[73] 10.0 9.0 10.0 10.0 9.5 9.5 10.0 9.5 10.0 9.5 10.0 9.5 10.0 9.5 9.5 10.0 9.5 10.0 9.5 9.5 9.5
[91] 9.5 9.0 9.5 9.5 10.0 9.0 9.0 10.0 9.5 9.5 10.0 9.0 10.0 9.5 9.5 10.0 9.0 10.0 9.5 10.0 10.0
[109] 10.0 8.5 9.5 10.0 8.5 9.5 10.0 9.5 10.0 9.5 9.5 10.0 9.5 10.0 9.5 10.0 9.5 10.0 9.5 10.0 9.5
[127] 9.5 10.0 10.0 10.0 9.0 10.0 9.0 10.0 9.5 9.5 9.5 9.5 9.5 9.5 9.5 10.0 9.5 10.0 10.0 9.5
[145] 9.5 10.0 9.0 8.5 9.5 10.0 9.0 9.5 9.5 10.0 9.5 9.5 10.0 8.5 10.0 9.5 10.0 9.5 9.5 9.5
[163] 10.0 10.0 9.5 9.5 9.5 9.5 10.0 9.5 10.0 8.0 9.5 10.0 10.0 9.5 10.0 10.0 9.5 9.5 9.5
[181] 10.0 10.0 9.5 8.5 10.0 9.5 10.0 8.5 10.0 10.0 9.5 9.5 10.0 9.5 8.5 10.0 7.5 9.5
[199] 10.0 10.0 9.5 9.5 10.0 9.5 9.5 10.0 9.5 9.5 10.0 9.5 9.5 10.0 9.5 9.5 10.0 10.0 9.5
[217] 9.5 9.5 9.0 10.0 10.0 9.5 9.5 10.0 9.0 9.5 10.0 9.5 9.5 10.0 9.5 9.5 10.0 10.0 9.5
[235] 9.5 10.0 9.5 10.0 9.5 10.0 9.5 10.0 9.5 10.0 9.5 9.5 10.0 9.5 9.5 10.0 9.5 10.0 9.5
[253] 9.5 10.0 9.0 10.0 10.0 9.5 9.0 9.0 9.0 10.0 9.5 10.0 9.5 10.0 9.5 10.0 10.0 9.5
[271] 9.5 10.0 9.5 10.0 9.5 10.0 9.5 9.5 10.0 9.5 9.5 9.0 9.0 9.5 9.0 9.5 9.0 10.0
[289] 9.5 10.0 8.0 9.0 9.5 8.5 10.0 9.5 10.0 9.5 10.0 9.5 9.5 10.0 9.0 9.5 9.5 9.5 9.5
[307] 10.0 10.0 8.0 9.0 9.5 9.5 10.0 10.0 9.5 10.0 9.5 9.5 10.0 9.5 8.0 10.0 9.5
[325] 9.0 10.0 10.0 9.5 10.0 10.0 10.0 8.5 9.0 9.0 10.0 9.5 9.5 9.5 9.5 9.5 10.0 10.0
[343] 9.5 10.0 8.5 9.5 10.0 9.5 10.0 8.0 10.0 9.5 9.5 10.0 9.5 9.5 9.0 9.5 9.5 9.5
[361] 10.0 10.0 9.0 10.0 9.0 10.0 10.0 9.5 9.5 10.0 10.0 10.0 8.5 10.0 9.5 10.0 9.5
[379] 10.0 9.5 9.5 9.5 10.0 9.5 8.0 10.0 9.5 10.0 9.5 8.5 9.0 9.5 9.5 9.0 10.0 8.5
[397] 9.5 10.0 10.0 8.5 8.5 9.0 10.0 9.5 10.0 9.5 9.5 10.0 10.0 8.5 9.0 9.5 8.5 9.5
[415] 10.0 9.5 10.0 9.5 9.5 9.5 10.0 8.0 10.0 9.5 7.0 10.0 9.5 9.5 10.0 10.0 9.5 10.0
[433] 10.0 9.5 9.0 10.0 9.0 10.0 10.0 9.5 9.5 10.0 9.5 10.0 8.0 9.5 9.5 10.0 9.5
[451] 9.5 10.0 8.0 10.0 9.5 10.0 9.5 10.0 9.5 9.5 10.0 8.5 8.5 9.5 10.0 8.5 9.0 10.0
[469] 0.0 10.0 9.5 10.0 9.5 9.5 10.0 10.0 9.5 9.5 10.0 9.5 9.5 10.0 9.5 10.0 8.5 9.0
[487] 9.5 10.0 9.5 9.5 9.0 9.5 10.0 9.5 10.0 8.5 9.5 10.0 9.5 9.5 10.0 9.0 10.0 9.0
[505] 9.0 10.0 9.0 0.0 9.5 10.0 9.5 9.5 8.5 10.0 9.5 9.5 10.0 9.5 10.0 9.5 10.0 9.5
[523] 9.0 9.0 9.5 9.5 9.5 9.5 10.0 9.5 10.0 9.5 10.0 9.5 9.5 10.0 9.5 9.5 9.5 9.5
[541] 10.0 10.0 9.5 10.0 9.5 9.5 10.0 10.0 9.5 9.0 10.0
```

Output file 12:

```
[1] 9 10 9 10 9 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10
[32] 9 10 10 9 10 10 10 10 10 10 9 9 10 10 9 10 10 10 9 10 10 10 10 10 10 10 9 10 10 10 10 10 10 10 10 10 10 10 10 10
[63] 9 10 8 10 10 9 9 10 10 10 9 9 10 10 10 9 10 10 9 10 10 10 9 10 10 10 9 9 10 9 10 10 10 10 10 10 10 10 10 10
[94] 10 10 8 10 10 10 10 10 10 10 9 10 9 10 10 10 9 10 10 10 9 10 10 10 9 10 10 9 10 10 9 10 10 9 10 10 9 10 10 6 10
[125] 10 10 10 8 9 10 10 10 10 9 10 9 10 10 10 10 10 10 10 9 10 10 10 10 9 10 10 10 9 9 10 10 10 10 10 10 10 10 10
[156] 10 10 10 10 10 10 10 10 4 10 8 10 10 10 10 9 10 9 10 7 9 10 9 10 10 10 9 9 10 10 10 9 9 10 10 7
[187] 10 10 10 10 9 10 10 10 10 10 9 10 10 10 10 9 10 10 10 9 10 10 10 9 10 9 10 10 9 10 10 9 10 10 9 10 10 9 10 10
[218] 10 10 10 9 10 10 10 10 9 10 10 10 10 9 9 10 10 9 9 10 10 10 10 10 10 10 10 10 10 10 10 9 10 10 9 10 10 9 10
[249] 10 9 10 10 9 10 10 10 9 10 10 9 10 10 9 9 10 10 9 9 10 10 9 10 10 9 10 10 9 10 10 9 10 10 9 10 10 9 10 10 9
[280] 10 10 10 9 10 9 10 10 10 8 10 10 0 10 10 9 10 10 9 10 10 9 10 10 10 10 9 10 10 9 10 10 10 9 10 10 10 10 10
[311] 9 10 10 9 10 10 10 10 9 10 10 10 10 9 10 10 8 10 10 9 10 10 9 10 10 10 10 9 10 10 10 9 10 10 9 10 10 9 10
[342] 9 9 10 9 10 10 9 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10
```

b) Xác định điểm số thấp nhất (có nộp bài)

Điểm số thấp nhất là phần tử có giá trị nhỏ nhất của vector chứa điểm - Marklist những sinh viên đã nộp bài. Ta tạo vector Marklist và tìm giá trị nhỏ nhất của Marklist như dưới đây.

```
print(min(Marklist))
```

Output file 6:

```
[1] 3
```

Output file 9:

```
[1] 0
```

Output file 10:

```
[1] 0
```

Output file 12:

```
[1] 0
```



- c) Xác định danh sách các sinh viên có ít nhất một bài có số điểm thấp nhất Danh sách các sinh viên có ít nhất một bài có số điểm thấp nhất là danh sách ID những sinh viên có ít nhất một điểm số bằng điểm số thấp nhất. Ta tạo dataframe df_2c là tập con của c chứa phần tử thỏa điều kiện có điểm bằng điểm số thấp nhất như dưới đây. Danh sách ID của df_2c là danh sách cần xác định.

```
df_2c=subset(c,Diem==min(Marklist)) print(df_2c[["ID"]])
```

Output file 6:

```
[1] "1937019" "1914661"
```

Output file 9:

```
[1] "1914661"
```

Output file 10:

```
[1] "1914661"
```

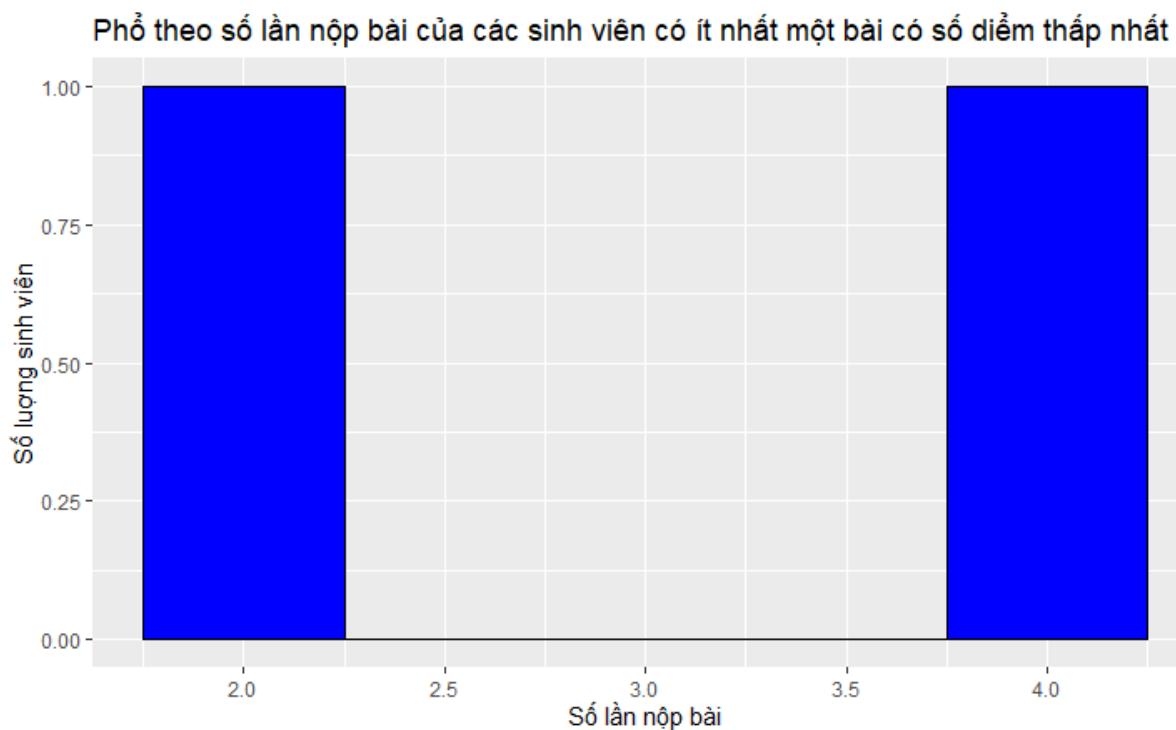
Output file 12:

```
[1] "1914661"
```

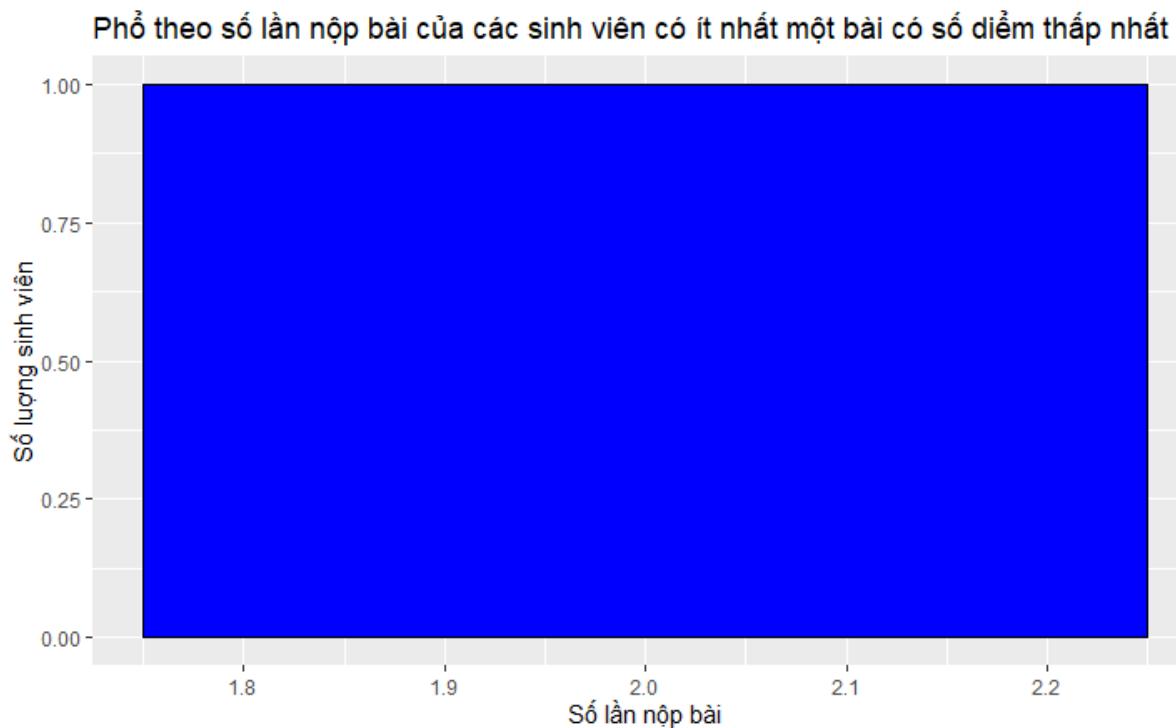
- d) Xác định phổ theo số lần nộp bài của các sinh viên có ít nhất một bài có số điểm thấp nhất
- Dầu tiên, ta tìm số lượng học sinh có ít nhất một bài có số điểm thấp nhất để truy cập các phần tử của df_2d. Ta dùng vòng lặp for và hàm lọc filter để tạo dataframe df_2d chứa thông tin về các sinh viên có ít nhất một bài có số điểm thấp nhất trong dataframe Submissions. Ta dùng các lệnh trong thư viện ggplot2 vẽ phổ theo số lần nộp bài của các sinh viên có ít nhất một bài có số điểm thấp nhất với trực dọc là số lượng sinh viên, trực ngang là số lần nộp bài như sau.

```
n2c=sum(!is.na(df_2c[["ID"]])) # n2c là số lượng phần tử của df_2c
df_2d<-data.frame() # tạo df_2d
for (i in 1:n2c){
  df_2d<-rbind(df_2d,c(filter(Submissions,ID==df_2c[["ID"]][i]))) #gán vào df_2d
  các giá trị thỏa điều kiện có ít nhất một bài có số điểm thấp nhất
}
ggplot(df_2d,aes(NoS)) +
  geom_histogram(binwidth = 0.5, fill = "blue", col = "black") +
  xlab("Số lần nộp bài") +
  ylab("Số lượng sinh viên") +
  ggtitle("Phổ theo số lần nộp bài của các sinh viên có ít nhất một bài có số điểm thấp nhất") #dùng histogram để vẽ phổ
```

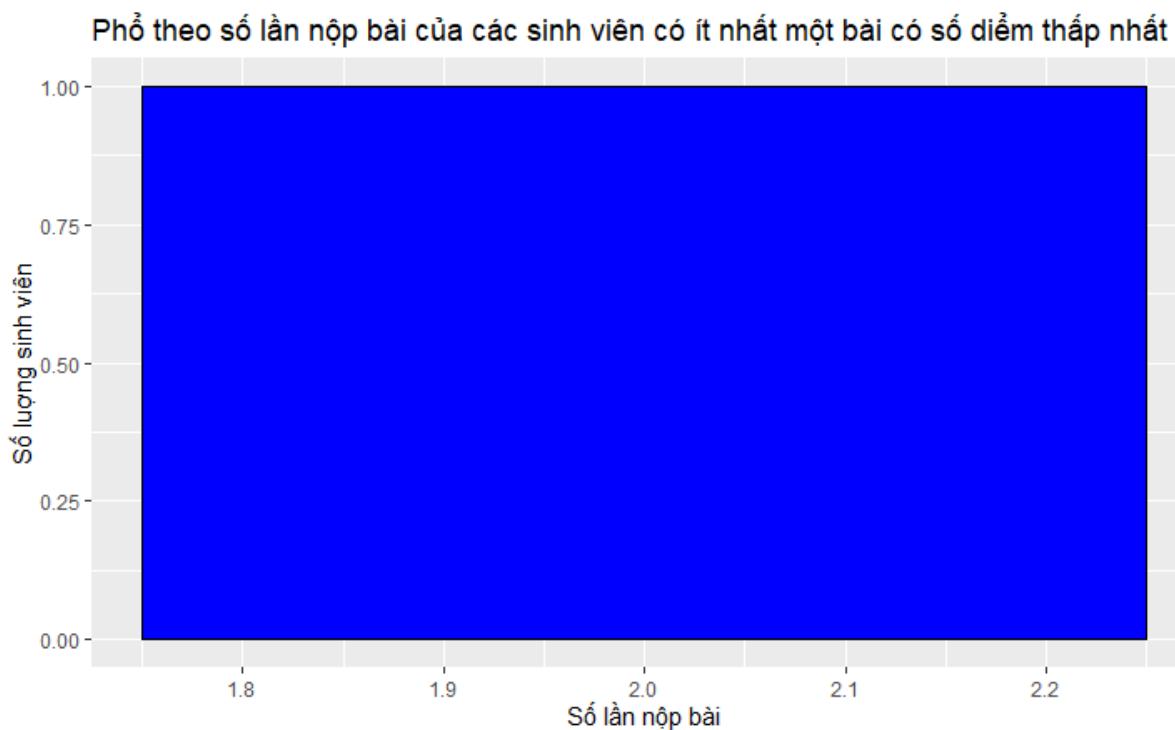
Output file 6:



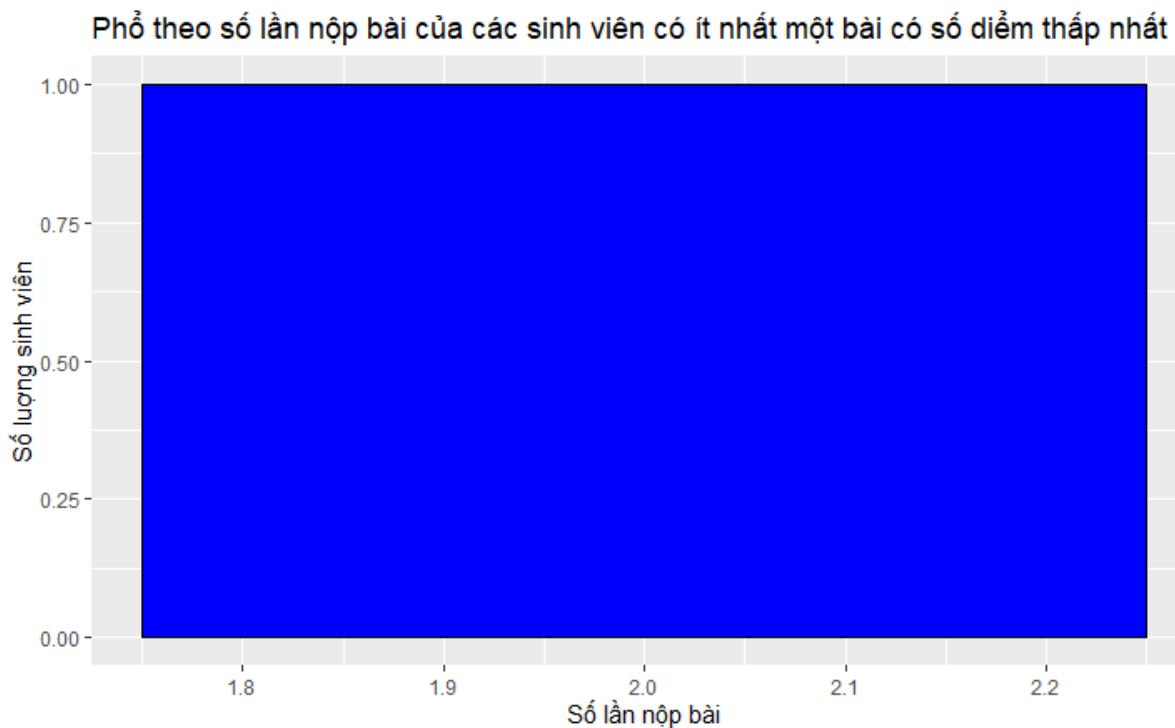
Output file 9:



Output file 10:



Output file 12:



e) Xác định điểm số tổng kết thấp nhất

Điểm số tổng kết thấp nhất là giá trị nhỏ nhất của cột điểm số tổng kết trong Submissions.

```
print(min(Submissions[["Mark"]]))
```

Output file 6:

[1] 7.5



Output file 9:

```
[1] 7
```

Output file 10:

```
[1] 8
```

Output file 12:

```
[1] 8
```

f) Xác định danh sách các sinh viên có điểm số tổng kết thấp nhất

Danh sách các sinh viên có ít nhất một bài có số điểm số tổng kết thấp nhất là danh sách ID những sinh viên có điểm số tổng kết bằng điểm số tổng kết thấp nhất. Ta tạo dataframe df_2f là tập con của Submissions chứa phần tử thỏa điều kiện có điểm bằng điểm số tổng kết thấp nhất như dưới đây. Danh sách ID của df_2f là danh sách cần xác định.

```
df_2f=subset(Submissions,Mark==min(Submissions[["Mark"]]))  
print(df_2f[["ID"]])
```

Output file 6:

```
[1] "1911283"
```

Output file 9:

```
[1] "1910984"
```

Output file 10:

```
[1] "1913218" "1914651"
```

Output file 12:

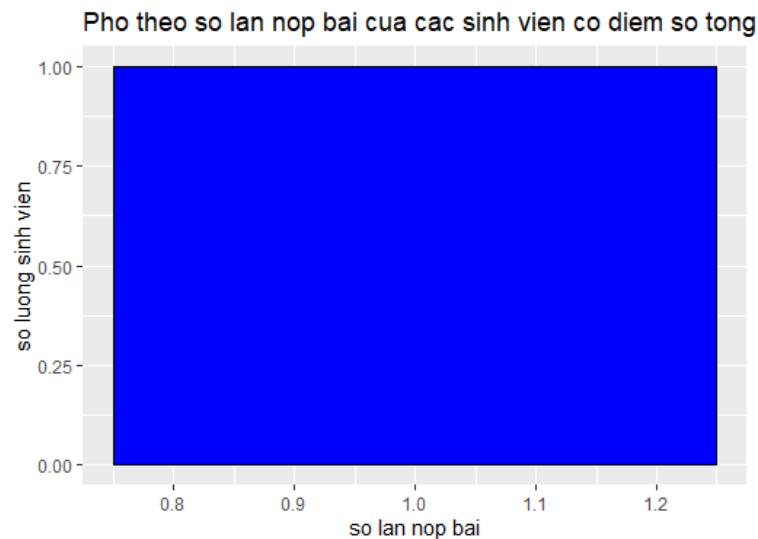
```
[1] "1912523" "1912966" "1913218"
```

g) Xác định phổ theo số lần nộp bài của các sinh viên có điểm số tổng kết thấp nhất

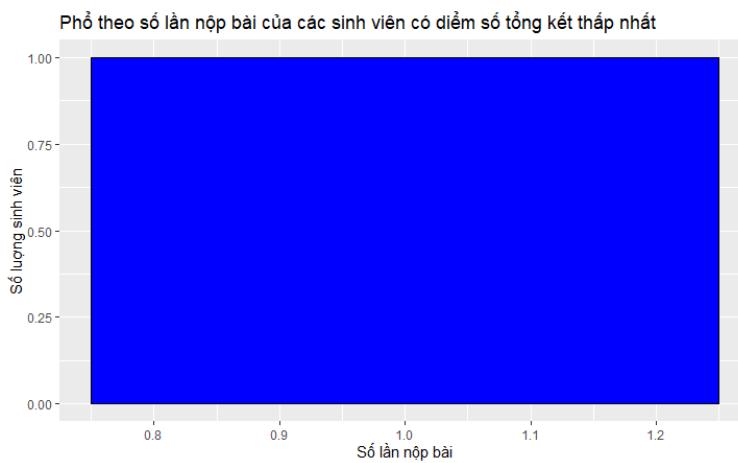
Ta vẽ phổ theo số lần nộp bài của các sinh viên có ít nhất một bài có số điểm thấp nhất với trực dọc là số lượng sinh viên, trực ngang là số lần nộp bài của dataframe df_2f.

```
ggplot(df_2f,aes(NoS)) +  
  geom_histogram(binwidth = 0.5, fill = "blue", col = "black") +  
  xlab("Số lần nộp bài") +  
  ylab("Số lượng sinh viên") +  
  ggtitle("Phổ theo số lần nộp bài của các sinh viên có điểm số tổng kết thấp nhất")
```

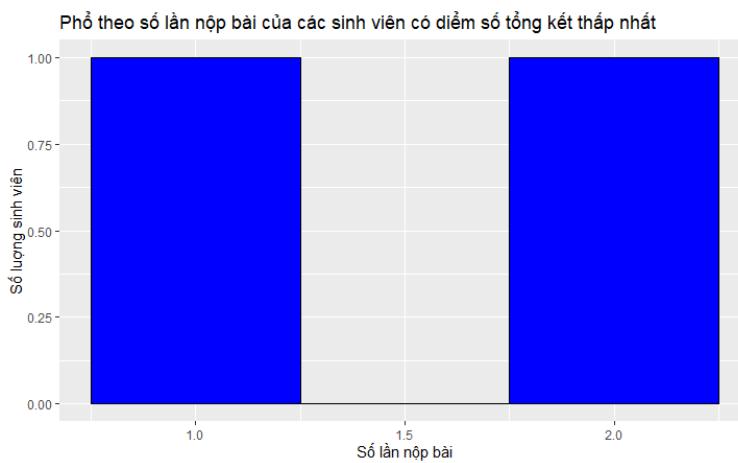
Output file 6:



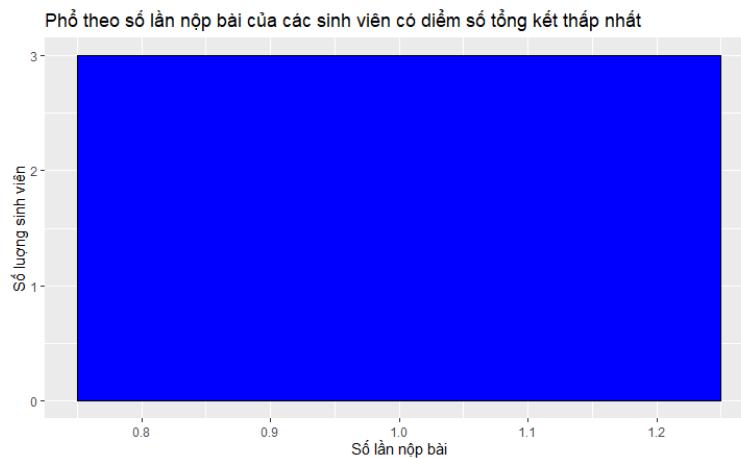
Output file 9:



Output file 10:



Output file 12:



h) Xác định điểm số cao nhất

Điểm số cao nhất là phần tử có giá trị lớn nhất của vector chứa điểm - Marklist.

```
print(max(Marklist))
```

Output file 6:

```
[1] 10
```

Output file 9:

```
[1] 10
```

Output file 10:

```
[1] 10
```

Output file 12:

```
[1] 10
```

i) Xác định danh sách các sinh viên có tối thiểu một bài nộp có số điểm số cao nhất

Sinh viên có tối thiểu một bài nộp có số điểm số cao nhất thì điểm số tổng kết cũng là điểm số tổng kết cao nhất. Do đó, ta tạo dataframe df_2i là tập con của Submissions chứa phần tử thỏa điều kiện có điểm bằng điểm số cao nhất như dưới đây. Danh sách ID của df_2i là danh sách cần xác định.

```
df_2i=subset(Submissions,Mark==max(Marklist))
print(df_2i[["ID"]])
```

Output file 6:



[1] "1511191" "1812257" "1812477" "1813096" "1813528" "1814096" "1814518" "1814611" "1820028"
[10] "1852443" "1910006" "1910032" "1910038" "1910060" "1910076" "1910101" "1910110" "1910113"
[19] "1910123" "1910137" "1910198" "1910202" "1910224" "1910238" "1910265" "1910276" "1910298"
[28] "1910339" "1910346" "1910347" "1910351" "1910402" "1910409" "1910473" "1910563" "1910565"
[37] "1910620" "1910643" "1910644" "1910650" "1910663" "1910666" "1910735" "1910865" "1910892"
[46] "1910916" "1910984" "1911000" "1911015" "1911044" "1911056" "1911058" "1911066" "1911105"
[55] "1911110" "1911136" "1911185" "1911186" "1911207" "1911217" "1911262" "1911285" "1911296"
[64] "1911314" "1911363" "1911441" "1911478" "1911520" "1911530" "1911561" "1911565" "1911569"
[73] "1911591" "1911594" "1911650" "1911704" "1911736" "1911796" "1911837" "1911841" "1911881"
[82] "1911900" "1911907" "1912041" "1912046" "1912056" "1912084" "1912123" "1912184" "1912190"
[91] "1912237" "1912267" "1912288" "1912371" "1912384" "1912386" "1912410" "1912457" "1912463"
[100] "1912522" "1912523" "1912526" "1912539" "1912579" "1912594" "1912602" "1912675" "1912677"
[109] "1912683" "1912700" "1912705" "1912713" "1912715" "1912749" "1912761" "1912798" "1912811"
[118] "1912817" "1912954" "1912958" "1912980" "1913014" "1913032" "1913040" "1913045" "1913048"
[127] "1913075" "1913094" "1913114" "1913123" "1913167" "1913186" "1913218" "1913228" "1913241"
[136] "1913254" "1913260" "1913261" "1913268" "1913306" "1913336" "1913341" "1913354" "1913355"
[145] "1913380" "1913386" "1913396" "1913418" "1913424" "1913430" "1913433" "1913457" "1913464"
[154] "1913467" "1913560" "1913566" "1913599" "1913609" "1913621" "1913629" "1913652" "1913678"
[163] "1913695" "1913713" "1913729" "1913756" "1913763" "1913764" "1913775" "1913828"
[172] "1913832" "1913844" "1913917" "1913918" "1913944" "1913949" "1913990" "1914003" "1914011"
[181] "1914022" "1914038" "1914047" "1914052" "1914054" "1914055" "1914064" "1914078" "1914084"
[190] "1914093" "1914126" "1914210" "1914220" "1914227" "1914232" "1914291" "1914310" "1914316"
[199] "1914352" "1914384" "1914405" "1914424" "1914472" "1914474" "1914477" "1914641" "1914659"
[208] "1914661" "1914677" "1914685" "1914697" "1914698" "1914704" "1914713" "1914720" "1914738"
[217] "1914763" "1914768" "1914802" "1914807" "1914830" "1914845" "1914864" "1914878" "1914880"
[226] "1914881" "1914900" "1914914" "1914979" "1915016" "1915040" "1915063" "1915071" "1915076"
[235] "1915083" "1915130" "1915133" "1915161" "1915251" "1915268" "1915294" "1915323" "1915329"
[244] "1915350" "1915378" "1915439" "1915442" "1915470" "1915473" "1915474" "1915482" "1915486"
[253] "1915520" "1915540" "1915541" "1915557" "1915562" "1915570" "1915598" "1915650" "1915651"
[262] "1915656" "1915667" "1915745" "1915775" "1915787" "1915795" "1915822" "1915865" "1915866"
[271] "1915882" "1915903" "1915919" "1915931" "1915939" "1915940" "1915982" "1915983" "1915988"
[280] "1915991" "1916022" "1916061" "1927007" "1936024" "1937019"

Output file 9:

[1] "1511191" "1613010" "1812257" "1812477" "1813528" "1814096" "1814518" "1852443" "1910006"
[10] "1910032" "1910038" "1910060" "1910101" "1910110" "1910123" "1910137" "1910198"
[19] "1910202" "1910224" "1910238" "1910265" "1910276" "1910298" "1910339" "1910346" "1910347"
[28] "1910351" "1910402" "1910409" "1910473" "1910565" "1910620" "1910643" "1910644" "1910650"
[37] "1910663" "1910666" "1910735" "1910865" "1910892" "1910916" "1911000" "1911015" "1911044"
[46] "1911056" "1911058" "1911066" "1911105" "1911110" "1911136" "1911185" "1911207"
[55] "1911217" "1911262" "1911283" "1911285" "1911314" "1911363" "1911441" "1911456" "1911478"
[64] "1911520" "1911530" "1911561" "1911565" "1911569" "1911591" "1911594" "1911650" "1911704"
[73] "1911736" "1911796" "1911837" "1911841" "1911881" "1911900" "1911907" "1911931" "1912046"
[82] "1912056" "1912084" "1912184" "1912190" "1912237" "1912288" "1912371" "1912410"
[91] "1912457" "1912463" "1912522" "1912523" "1912526" "1912539" "1912579" "1912594" "1912602"
[100] "1912675" "1912683" "1912700" "1912705" "1912713" "1912715" "1912749" "1912761" "1912798"
[109] "1912811" "1912817" "1912916" "1912954" "1912980" "1913014" "1913021" "1913026" "1913040"
[118] "1913045" "1913048" "1913075" "1913094" "1913114" "1913123" "1913167" "1913186" "1913241"
[127] "1913254" "1913260" "1913261" "1913268" "1913306" "1913334" "191336" "1913341" "1913355"
[136] "1913356" "1913396" "1913418" "1913419" "1913424" "1913430" "1913433" "1913457" "1913464"
[145] "1913467" "1913560" "1913566" "1913609" "1913621" "1913629" "1913651" "1913652" "1913678"
[154] "1913695" "1913713" "1913763" "1913764" "1913775" "1913828" "1913844" "1913918"
[163] "1913944" "1913949" "1913990" "1914003" "1914011" "1914022" "1914038" "1914047" "1914054"
[172] "1914055" "1914064" "1914084" "1914093" "1914210" "1914220" "1914227" "1914291" "1914310"
[181] "1914316" "1914352" "1914384" "1914405" "1914405" "1914424" "1914474" "1914477" "1914641"
[190] "1914661" "1914674" "1914677" "1914685" "1914698" "1914704" "1914713" "1914720" "1914738"
[199] "1914763" "1914768" "1914802" "1914807" "1914830" "1914845" "1914864" "1914878" "1914880"
[208] "1914881" "1914914" "1914979" "1915016" "1915040" "1915063" "1915071" "1915083" "1915130"
[217] "1915133" "1915146" "1915251" "1915268" "1915294" "1915323" "1915329" "1915350" "1915351"
[226] "1915378" "1915439" "1915442" "1915470" "1915473" "1915482" "1915520" "1915540" "1915541"
[235] "1915551" "1915557" "1915562" "1915570" "1915598" "1915651" "1915656" "1915745" "1915775"
[244] "1915787" "1915822" "1915865" "1915866" "1915882" "1915903" "1915919" "1915931" "1915939"
[253] "1915940" "1915982" "1915983" "1915988" "1915991" "1916022" "1916061" "1927007" "1936024"
[262] "1937019" "1937064"

Output file 10:

```
[1] "1511191" "1613010" "1812477" "1813096" "1813681" "1814518" "1820028" "1852443" "1910006"
[10] "1910032" "1910038" "1910060" "1910101" "1910110" "1910113" "1910123" "1910137" "1910202"
[19] "1910238" "1910265" "1910276" "1910298" "1910339" "1910347" "1910351" "1910402" "1910409"
[28] "1910620" "1910643" "1910644" "1910666" "1910865" "1910892" "1910916" "1911015" "1911044"
[37] "1911058" "1911066" "1911105" "1911110" "1911136" "1911185" "1911186" "1911217" "1911262"
[46] "1911314" "1911363" "1911441" "1911478" "1911520" "1911561" "1911565" "1911591" "1911594"
[55] "1911704" "1911736" "1911796" "1911837" "1911841" "1911881" "1911900" "1911907" "1912046"
[64] "1912056" "1912084" "1912184" "1912190" "1912237" "1912267" "1912288" "1912371" "1912384"
[73] "1912410" "1912457" "1912463" "1912526" "1912539" "1912594" "1912602" "1912675" "1912676"
[82] "1912700" "1912713" "1912715" "1912749" "1912761" "1912798" "1912817" "1912916" "1912954"
[91] "1912980" "1913014" "1913021" "1913040" "1913048" "1913075" "1913094" "1913114" "1913123"
[100] "1913186" "1913254" "1913306" "1913341" "1913355" "1913396" "1913418" "1913424" "1913430"
[109] "1913464" "1913467" "1913560" "1913609" "1913621" "1913652" "1913678" "1913695" "1913713"
[118] "1913763" "1913764" "1913775" "1913828" "1913832" "1913844" "1913918" "1913944" "1913949"
[127] "1914003" "1914011" "1914022" "1914038" "1914047" "1914052" "1914054" "1914055" "1914093"
[136] "1914126" "1914210" "1914227" "1914291" "1914310" "1914316" "1914352" "1914384" "1914405"
[145] "1914641" "1914659" "1914661" "1914674" "1914685" "1914698" "1914704" "1914713" "1914720"
[154] "1914738" "1914763" "1914768" "1914830" "1914845" "1914864" "1914881" "1914914" "1914979"
[163] "1915016" "1915040" "1915071" "1915083" "1915130" "1915133" "1915251" "1915268" "1915294"
[172] "1915323" "1915329" "1915350" "1915378" "1915439" "1915442" "1915470" "1915473" "1915482"
[181] "1915540" "1915541" "1915570" "1915598" "1915651" "1915656" "1915745" "1915775" "1915787"
[190] "1915822" "1915865" "1915882" "1915919" "1915940" "1915953" "1915982" "1915991" "1916022"
[199] "1916061" "1927007" "1936024"
```

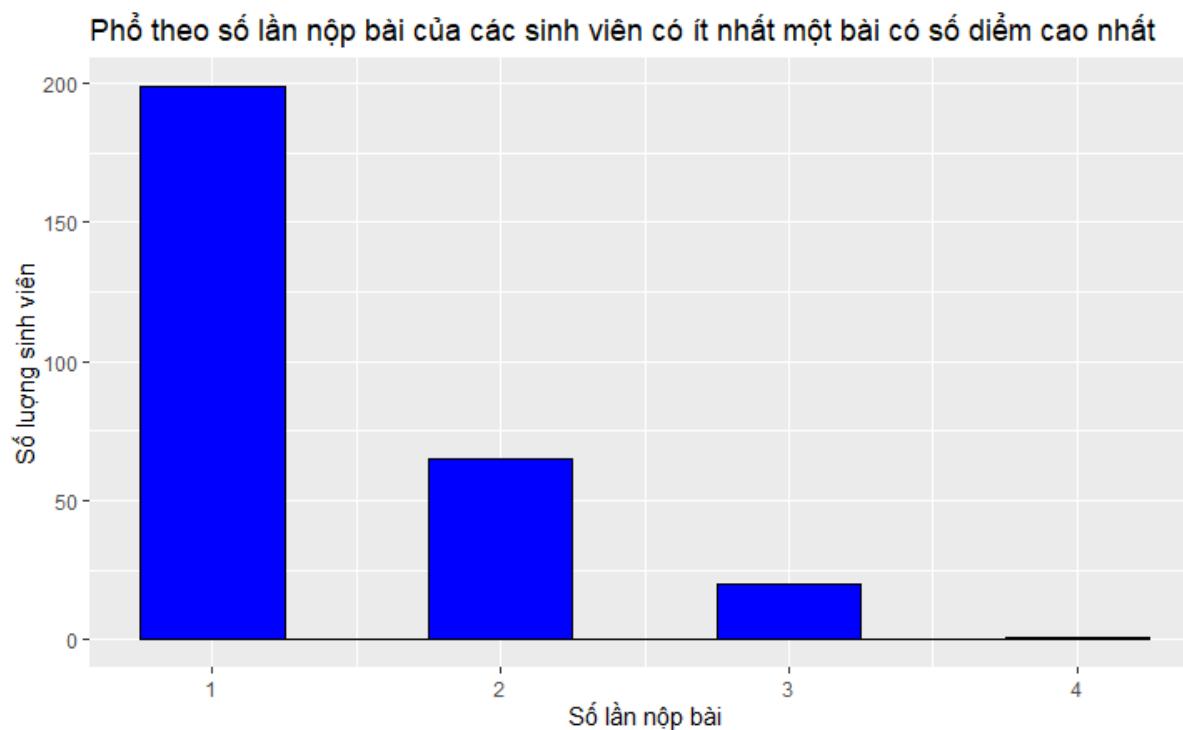
Output file 12:

```
[1] "1511191" "1613010" "1812257" "1812477" "1812478" "1813096" "1814096" "1814518" "1820028"
[10] "1852443" "1910006" "1910032" "1910038" "1910060" "1910076" "1910101" "1910110" "1910113"
[19] "1910123" "1910137" "1910202" "1910224" "1910238" "1910265" "1910276" "1910298" "1910339"
[28] "1910346" "1910347" "1910351" "1910402" "1910409" "1910473" "1910565" "1910620" "1910643"
[37] "1910644" "1910650" "1910663" "1910666" "1910865" "1910892" "1910916" "1910984" "1911000"
[46] "1911015" "1911056" "1911058" "1911066" "1911110" "1911136" "1911185" "1911186" "1911207"
[55] "1911217" "1911262" "1911283" "1911285" "1911296" "1911314" "1911363" "1911441" "1911478"
[64] "1911520" "1911530" "1911565" "1911569" "1911591" "1911594" "1911650" "1911704" "1911736"
[73] "1911796" "1911837" "1911841" "1911881" "1911900" "1911907" "1911931" "1912046" "1912056"
[82] "1912084" "1912123" "1912184" "1912190" "1912237" "1912267" "1912288" "1912371" "1912384"
[91] "1912410" "1912457" "1912463" "1912539" "1912594" "1912602" "1912675" "1912676" "1912683"
[100] "1912700" "1912713" "1912715" "1912749" "1912761" "1912798" "1912811" "1912817" "1912912"
[109] "1912980" "1913014" "1913021" "1913040" "1913048" "1913075" "1913094" "1913102"
[118] "1913114" "1913123" "1913167" "1913186" "1913241" "1913254" "1913261" "1913268" "1913306"
[127] "1913334" "1913336" "1913341" "1913354" "1913355" "1913356" "1913386" "1913396" "1913418"
[136] "1913419" "1913424" "1913430" "1913433" "1913457" "1913464" "1913467" "1913468" "1913566"
[145] "1913599" "1913609" "1913621" "1913651" "1913678" "1913695" "1913713" "1913729" "1913763"
[154] "1913764" "1913775" "1913828" "1913832" "1913844" "1913918" "1913944" "1913949" "1914003"
[163] "1914011" "1914022" "1914038" "1914047" "1914052" "1914054" "1914055" "1914078" "1914084"
[172] "1914093" "1914121" "1914126" "1914210" "1914220" "1914227" "1914232" "1914291" "1914310"
[181] "1914316" "1914352" "1914384" "1914405" "1914424" "1914472" "1914474" "1914477" "1914641"
[190] "1914659" "1914661" "1914674" "1914677" "1914685" "1914698" "1914704" "1914713" "1914720"
[199] "1914738" "1914763" "1914768" "1914830" "1914845" "1914864" "1914880" "1914881" "1914914"
[208] "1914979" "1915016" "1915040" "1915063" "1915071" "1915083" "1915130" "1915133" "1915146"
[217] "1915251" "1915268" "1915294" "1915323" "1915329" "1915350" "1915351" "1915378" "1915439"
[226] "1915442" "1915473" "1915474" "1915482" "1915486" "1915520" "1915540" "1915541" "1915551"
[235] "1915562" "1915570" "1915598" "1915651" "1915656" "1915745" "1915775" "1915787" "1915795"
[244] "1915822" "1915865" "1915866" "1915882" "1915903" "1915919" "1915953" "1915982" "1915983"
[253] "1915988" "1915991" "1916022" "1916061" "1927007" "1936024" "1937024" "1937064"
```

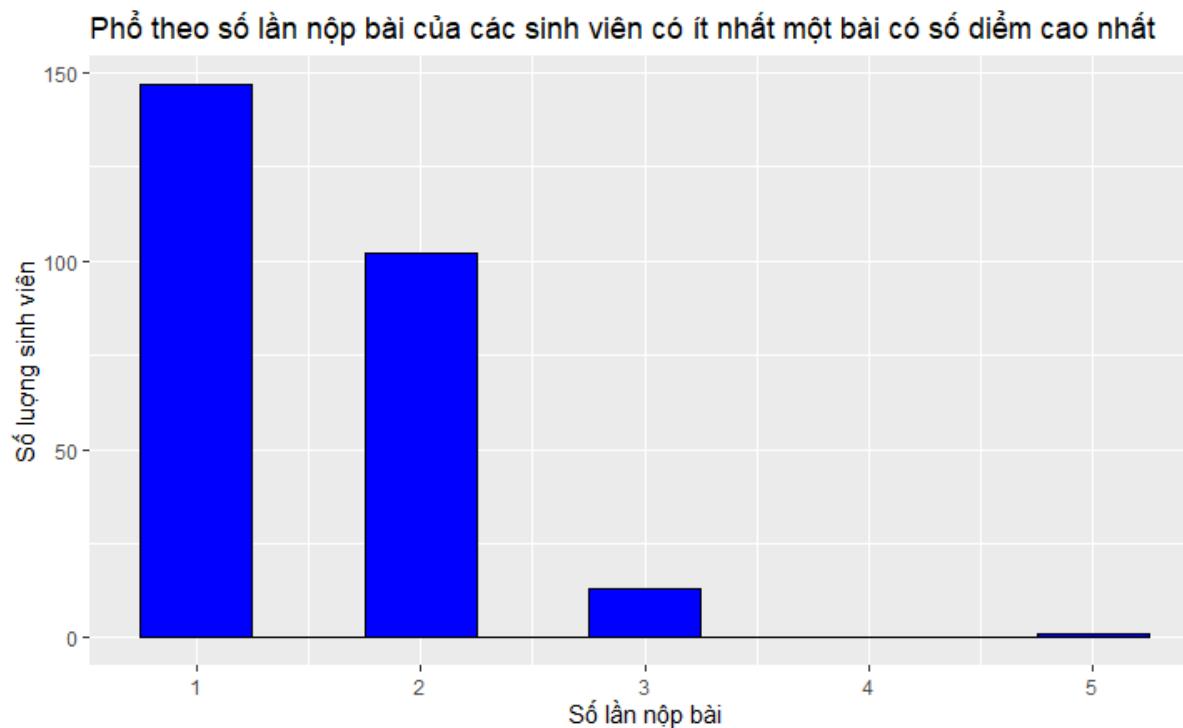
- j) Xác định phổ theo số lần nộp bài của các sinh viên có tối thiểu một bài nộp có điểm số cao nhất
 Ta vẽ phổ theo số lần nộp bài của các sinh viên có tối thiểu một bài nộp có điểm số cao nhất với trục dọc là số lượng sinh viên, trục ngang là số lần nộp bài của dataframe df_2i.

```
ggplot(df_2i,aes(NoS)) +
  geom_histogram(binwidth = 0.5, fill = "blue", col = "black") +
  xlab("Số lần nộp bài") +
  ylab("Số lượng sinh viên") +
  ggtitle("Phổ theo số lần nộp bài của các sinh viên có ít nhất một bài có số điểm cao nhất")
```

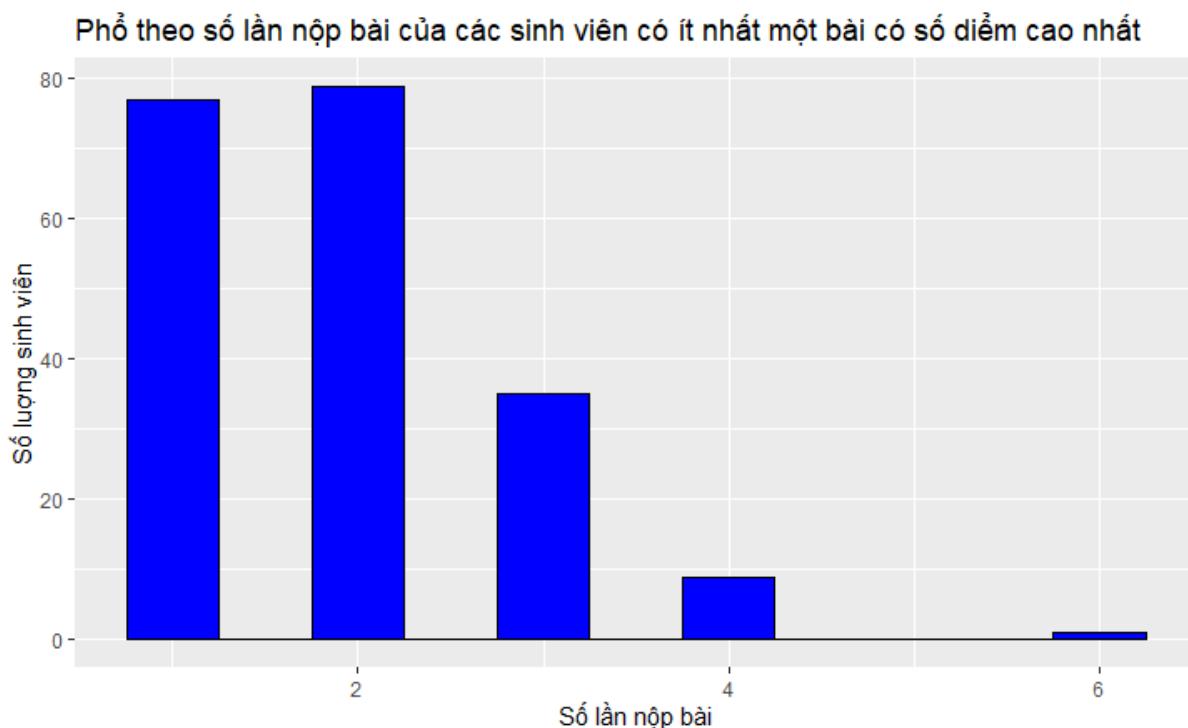
Output file 6:



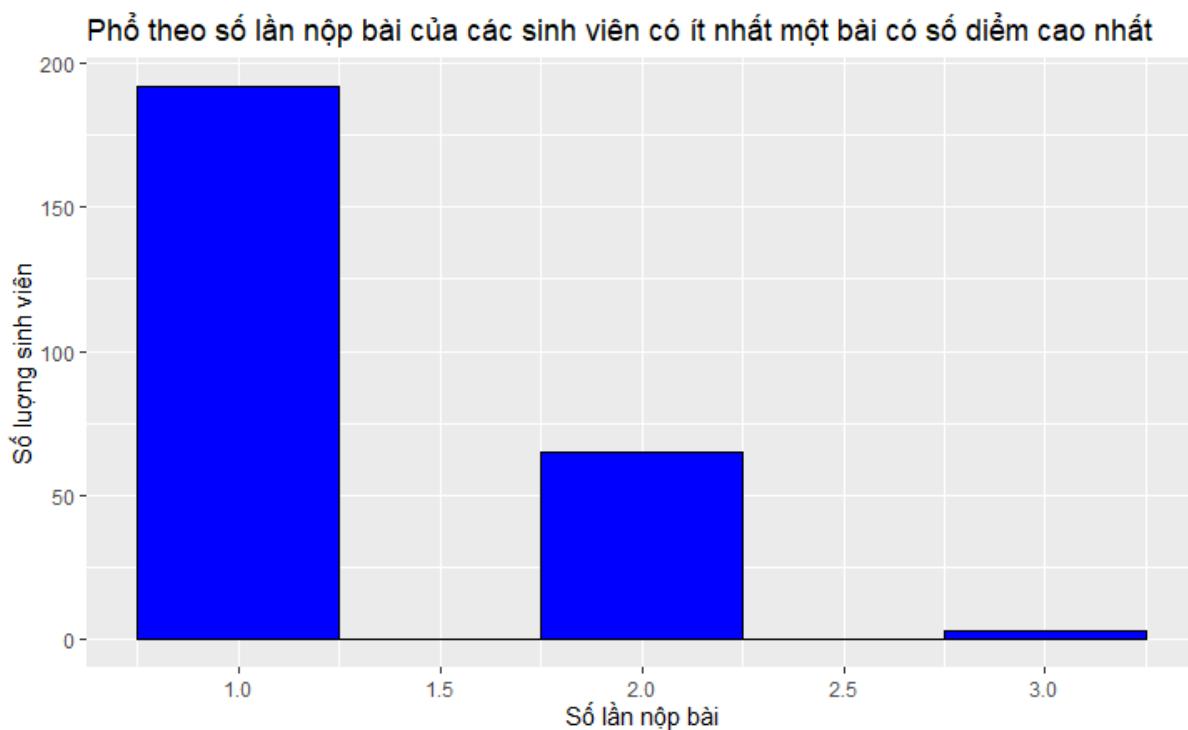
Output file 9:



Output file 10:



Output file 12:



k) Xác định điểm số tổng kết cao nhất

Điểm số cao nhất là phần tử có giá trị lớn nhất của cột điểm số tổng kết của Submissions.

```
print(max(Submissions[["Mark"]]))
```

Output file 6:

```
[1] 10
```



Output file 9:

[1] 10

Output file 10:

[1] 10

Output file 12:

[1] 10

- l) Xác định danh sách các sinh viên có điểm số tổng kết cao nhất

Ta tạo dataframe df_2l là tập con của Submissions chứa phần tử thỏa điều kiện có điểm bằng điểm số tổng kết cao nhất như dưới đây. Danh sách ID của df_2l là danh sách cần xác định.

```
df_2l=subset(Submissions,Mark==max(Submissions[["Mark"]]))  
print(df_2l[["ID"]])
```

Output file 6:

```
[1] "1511191" "1812257" "1812477" "1813096" "1813528" "1814096" "1814518" "1814611" "1820028"  
[10] "1852443" "1910006" "1910032" "1910038" "1910060" "1910076" "1910101" "1910110" "1910113"  
[19] "1910123" "1910137" "1910198" "1910202" "1910224" "1910238" "1910265" "1910276" "1910298"  
[28] "1910339" "1910346" "1910347" "1910351" "1910402" "1910409" "1910473" "1910563" "1910565"  
[37] "1910620" "1910643" "1910644" "1910650" "1910663" "1910666" "1910735" "1910865" "1910892"  
[46] "1910916" "1910984" "1911000" "1911015" "1911044" "1911056" "1911058" "1911066" "1911105"  
[55] "1911110" "1911136" "1911185" "1911186" "1911207" "1911217" "1911262" "1911285" "1911296"  
[64] "1911314" "1911363" "1911441" "1911478" "1911520" "1911530" "1911561" "1911565" "1911569"  
[73] "1911591" "1911594" "1911650" "1911704" "1911736" "1911796" "1911837" "1911841" "1911881"  
[82] "1911900" "1911907" "1912041" "1912046" "1912056" "1912084" "1912123" "1912184" "1912190"  
[91] "1912237" "1912267" "1912288" "1912371" "1912384" "1912386" "1912410" "1912457" "1912463"  
[100] "1912522" "1912523" "1912526" "1912539" "1912579" "1912594" "1912602" "1912675" "1912677"  
[109] "1912683" "1912700" "1912705" "1912713" "1912715" "1912749" "1912761" "1912798" "1912811"  
[118] "1912817" "1912954" "1912958" "1912980" "1913014" "1913032" "1913040" "1913045" "1913048"  
[127] "1913075" "1913094" "1913114" "1913123" "1913167" "1913186" "1913218" "1913228" "1913241"  
[136] "1913254" "1913260" "1913261" "1913268" "1913306" "1913336" "1913341" "1913354" "1913355"  
[145] "1913380" "1913386" "1913396" "1913418" "1913424" "1913430" "1913433" "1913457" "1913464"  
[154] "1913467" "1913560" "1913566" "1913599" "1913609" "1913621" "1913629" "1913652" "1913678"  
[163] "1913695" "1913713" "1913729" "1913756" "1913758" "1913763" "1913764" "1913775" "1913828"  
[172] "1913832" "1913844" "1913917" "1913918" "1913944" "1913949" "1913990" "1914003" "1914011"  
[181] "1914022" "1914038" "1914047" "1914052" "1914054" "1914055" "1914064" "1914078" "1914084"  
[190] "1914093" "1914126" "1914210" "1914220" "1914227" "1914232" "1914291" "1914310" "1914316"  
[199] "1914352" "1914384" "1914405" "1914424" "1914472" "1914474" "1914477" "1914641" "1914659"  
[208] "1914661" "1914677" "1914685" "1914697" "1914698" "1914704" "1914713" "1914720" "1914738"  
[217] "1914763" "1914768" "1914802" "1914807" "1914830" "1914845" "1914864" "1914878" "1914880"  
[226] "1914881" "1914900" "1914914" "1914979" "1915016" "1915040" "1915063" "1915071" "1915076"  
[235] "1915083" "1915130" "1915133" "1915161" "1915251" "1915268" "1915294" "1915323" "1915329"  
[244] "1915350" "1915378" "1915439" "1915442" "1915470" "1915473" "1915474" "1915482" "1915486"  
[253] "1915520" "1915540" "1915541" "1915557" "1915562" "1915570" "1915598" "1915650" "1915651"  
[262] "1915656" "1915667" "1915745" "1915775" "1915787" "1915795" "1915822" "1915865" "1915866"  
[271] "1915882" "1915903" "1915919" "1915931" "1915939" "1915940" "1915982" "1915983" "1915988"  
[280] "1915991" "1916022" "1916061" "1927007" "1936024" "1937019"
```

Output file 9:



```
[1] "1511191" "1613010" "1812257" "1812477" "1813528" "1814096" "1814518" "1852443" "1910006"  
[10] "1910032" "1910038" "1910060" "1910101" "1910110" "1910113" "1910123" "1910137" "1910198"  
[19] "1910202" "1910224" "1910238" "1910265" "1910276" "1910298" "1910339" "1910346" "1910347"  
[28] "1910351" "1910402" "1910409" "1910473" "1910565" "1910620" "1910643" "1910644" "1910650"  
[37] "1910663" "1910666" "1910735" "1910865" "1910892" "1910916" "1911000" "1911015" "1911044"  
[46] "1911056" "1911058" "1911066" "1911105" "1911110" "1911136" "1911185" "1911186" "1911207"  
[55] "1911217" "1911262" "1911283" "1911285" "1911314" "1911363" "1911441" "1911456" "1911478"  
[64] "1911520" "1911530" "1911561" "1911565" "1911569" "1911591" "1911594" "1911650" "1911704"  
[73] "1911736" "1911796" "1911837" "1911841" "1911881" "1911900" "1911907" "1911931" "1912046"  
[82] "1912056" "1912084" "1912184" "1912190" "1912237" "1912267" "1912288" "1912371" "1912410"  
[91] "1912457" "1912463" "1912522" "1912523" "1912526" "1912539" "1912579" "1912594" "1912602"  
[100] "1912675" "1912683" "1912700" "1912705" "1912713" "1912749" "1912761" "1912798"  
[109] "1912811" "1912817" "1912916" "1912954" "1912980" "1913014" "1913021" "1913026" "1913040"  
[118] "1913045" "1913048" "1913075" "1913094" "1913114" "1913123" "1913167" "1913186" "1913241"  
[127] "1913254" "1913260" "1913261" "1913268" "1913306" "1913334" "1913336" "1913341" "1913355"  
[136] "1913356" "1913396" "1913418" "1913419" "1913424" "1913430" "1913433" "1913457" "1913464"  
[145] "1913467" "1913560" "1913566" "1913609" "1913621" "1913629" "1913651" "1913652" "1913678"  
[154] "1913695" "1913713" "1913763" "1913764" "1913775" "1913828" "1913832" "1913844" "1913918"  
[163] "1913944" "1913949" "1913990" "1914003" "1914011" "1914022" "1914038" "1914047" "1914054"  
[172] "1914055" "1914064" "1914084" "1914093" "1914210" "1914220" "1914227" "1914291" "1914310"  
[181] "1914316" "1914352" "1914384" "1914405" "1914424" "1914474" "1914477" "1914641" "1914659"  
[190] "1914461" "1914674" "1914677" "1914685" "1914698" "1914704" "1914713" "1914720" "1914738"  
[199] "1914763" "1914768" "1914802" "1914807" "1914830" "1914845" "1914864" "1914878" "1914880"  
[208] "1914881" "1914914" "1914979" "1915016" "1915040" "1915063" "1915071" "1915083" "1915130"  
[217] "1915133" "1915146" "1915251" "1915268" "1915294" "1915323" "1915329" "1915350" "1915351"  
[226] "1915378" "1915439" "1915442" "1915470" "1915473" "1915482" "1915520" "1915540" "1915541"  
[235] "1915551" "1915557" "1915562" "1915570" "1915598" "1915651" "1915656" "1915745" "1915775"  
[244] "1915787" "1915822" "1915865" "1915866" "1915882" "1915903" "1915919" "1915931" "1915939"  
[253] "1915940" "1915982" "1915983" "1915988" "1915991" "1916022" "1916061" "1927007" "1936024"  
[262] "1937019" "1937064"
```

Output file 10:

```
[1] "1511191" "1613010" "1812477" "1813096" "1813681" "1814518" "1820028" "1852443" "1910006"  
[10] "1910032" "1910038" "1910060" "1910101" "1910110" "1910113" "1910123" "1910137" "1910202"  
[19] "1910238" "1910265" "1910276" "1910298" "1910339" "1910347" "1910351" "1910402" "1910409"  
[28] "1910620" "1910643" "1910644" "1910666" "1910865" "1910892" "1910916" "1911015" "1911044"  
[37] "1911058" "1911066" "1911105" "1911110" "1911136" "1911185" "1911186" "1911217" "1911262"  
[46] "1911314" "1911363" "1911441" "1911478" "1911520" "1911561" "1911565" "1911591" "1911594"  
[55] "1911704" "1911736" "1911796" "1911837" "1911841" "1911881" "1911900" "1911907" "1912046"  
[64] "1912056" "1912084" "1912184" "1912190" "1912237" "1912267" "1912288" "1912371" "1912384"  
[73] "1912410" "1912457" "1912463" "1912526" "1912539" "1912602" "1912675" "1912676"  
[82] "1912700" "1912713" "1912715" "1912749" "1912761" "1912798" "1912817" "1912916" "1912954"  
[91] "1912980" "1913014" "1913021" "1913040" "1913048" "1913075" "1913094" "1913114" "1913123"  
[100] "1913186" "1913254" "1913306" "1913341" "1913355" "1913396" "1913418" "1913424" "1913430"  
[109] "1913464" "1913467" "1913560" "1913609" "1913621" "1913652" "1913678" "1913695" "1913713"  
[118] "1913763" "1913764" "1913775" "1913828" "1913832" "1913844" "1913918" "1913944" "1913949"  
[127] "1914003" "1914011" "1914022" "1914038" "1914047" "1914052" "1914054" "1914055" "1914093"  
[136] "1914126" "1914210" "1914227" "1914291" "1914310" "1914316" "1914352" "1914384" "1914405"  
[145] "1914641" "1914659" "1914661" "1914674" "1914685" "1914698" "1914704" "1914713" "1914720"  
[154] "1914738" "1914763" "1914768" "1914830" "1914845" "1914864" "1914881" "1914914" "1914979"  
[163] "1915016" "1915040" "1915071" "1915083" "1915130" "1915133" "1915251" "1915268" "1915294"  
[172] "1915323" "1915329" "1915350" "1915378" "1915439" "1915442" "1915470" "1915473" "1915482"  
[181] "1915540" "1915541" "1915570" "1915598" "1915651" "1915656" "1915745" "1915775" "1915787"  
[190] "1915822" "1915865" "1915882" "1915919" "1915940" "1915953" "1915982" "1915991" "1916022"  
[199] "1916061" "1927007" "1936024"
```

Output file 12:

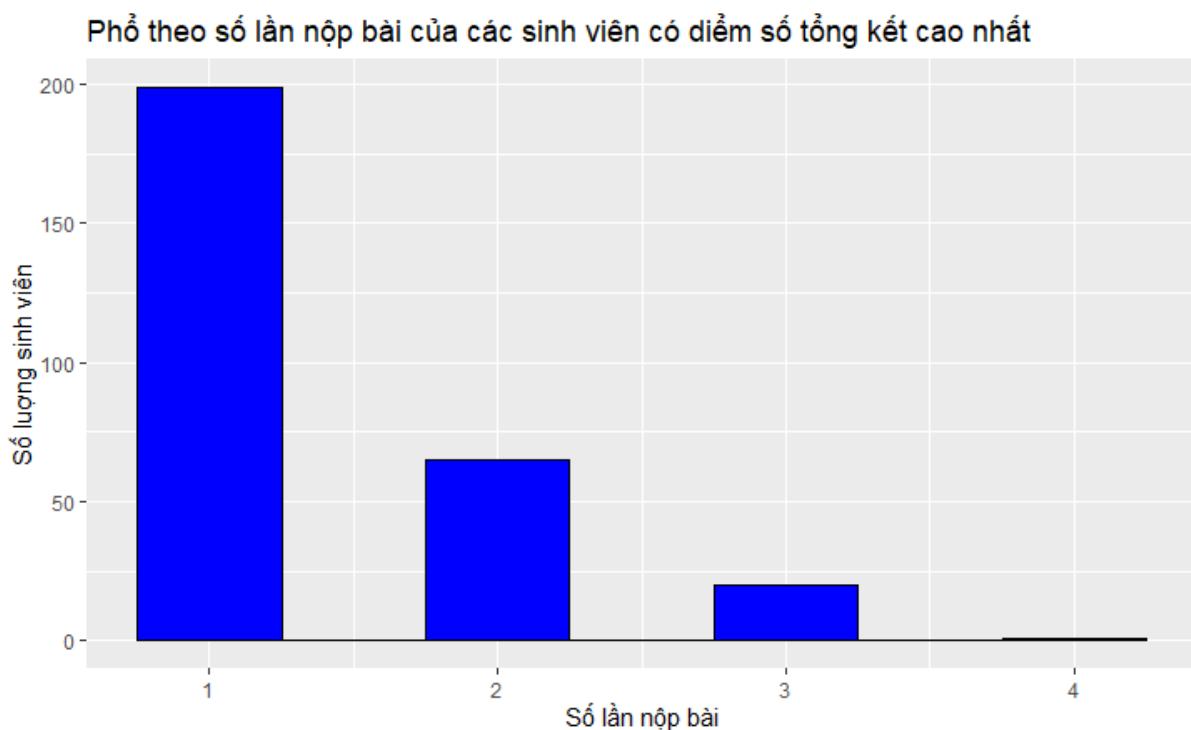
```
[1] "1511191" "1613010" "1812257" "1812477" "1812478" "1813096" "1814096" "1814518" "1820028"
[10] "1852443" "1910006" "1910032" "1910038" "1910060" "1910076" "1910101" "1910110" "1910113"
[19] "1910123" "1910137" "1910202" "1910224" "1910238" "1910265" "1910276" "1910298" "1910339"
[28] "1910346" "1910347" "1910351" "1910402" "1910409" "1910473" "1910565" "1910620" "1910643"
[37] "1910644" "1910650" "1910663" "1910666" "1910865" "1910892" "1910916" "1910984" "1911000"
[46] "1911015" "1911056" "1911058" "1911066" "1911110" "1911136" "1911185" "1911186" "1911207"
[55] "1911217" "1911262" "1911283" "1911285" "1911296" "1911314" "1911363" "1911441" "1911478"
[64] "1911520" "1911530" "1911565" "1911569" "1911591" "1911594" "1911650" "1911704" "1911736"
[73] "1911796" "1911837" "1911841" "1911881" "1911900" "1911907" "1911931" "1912046" "1912056"
[82] "1912084" "1912123" "1912184" "1912190" "1912237" "1912267" "1912288" "1912371" "1912384"
[91] "1912410" "1912457" "1912463" "1912539" "1912594" "1912602" "1912675" "1912676" "1912683"
[100] "1912700" "1912713" "1912715" "1912749" "1912761" "1912798" "1912811" "1912817" "1912912"
[109] "1912980" "1913014" "1913021" "1913040" "1913045" "1913048" "1913075" "1913094" "1913102"
[118] "1913114" "1913123" "1913167" "1913186" "1913241" "1913254" "1913261" "1913268" "1913306"
[127] "1913334" "1913336" "1913341" "1913354" "1913355" "1913356" "1913386" "1913396" "1913418"
[136] "1913419" "1913424" "1913430" "1913433" "1913457" "1913464" "1913467" "1913560" "1913566"
[145] "1913599" "1913609" "1913621" "1913651" "1913678" "1913695" "1913713" "1913729" "1913763"
[154] "1913764" "1913775" "1913828" "1913832" "1913844" "1913918" "1913944" "1913949" "1914003"
[163] "1914011" "1914022" "1914038" "1914047" "1914052" "1914054" "1914055" "1914078" "1914084"
[172] "1914093" "1914121" "1914126" "1914210" "1914220" "1914227" "1914232" "1914291" "1914310"
[181] "1914316" "1914352" "1914384" "1914405" "1914424" "1914472" "1914474" "1914477" "1914641"
[190] "1914659" "1914661" "1914674" "1914677" "1914685" "1914698" "1914704" "1914713" "1914720"
[199] "1914738" "1914763" "1914768" "1914830" "1914845" "1914864" "1914880" "1914881" "1914914"
[208] "1914979" "1915016" "1915040" "1915063" "1915071" "1915083" "1915130" "1915133" "1915146"
[217] "1915251" "1915268" "1915294" "1915323" "1915329" "1915350" "1915351" "1915378" "1915439"
[226] "1915442" "1915473" "1915474" "1915482" "1915486" "1915520" "1915540" "1915541" "1915551"
[235] "1915562" "1915570" "1915598" "1915651" "1915656" "1915745" "1915775" "1915787" "1915795"
[244] "1915822" "1915865" "1915866" "1915882" "1915903" "1915919" "1915953" "1915982" "1915983"
[253] "1915988" "1915991" "1916022" "1916061" "1927007" "1936024" "1937024" "1937064"
```

m) Xác định phô theo số lần nộp bài của các sinh viên có điểm số tổng kết cao nhất

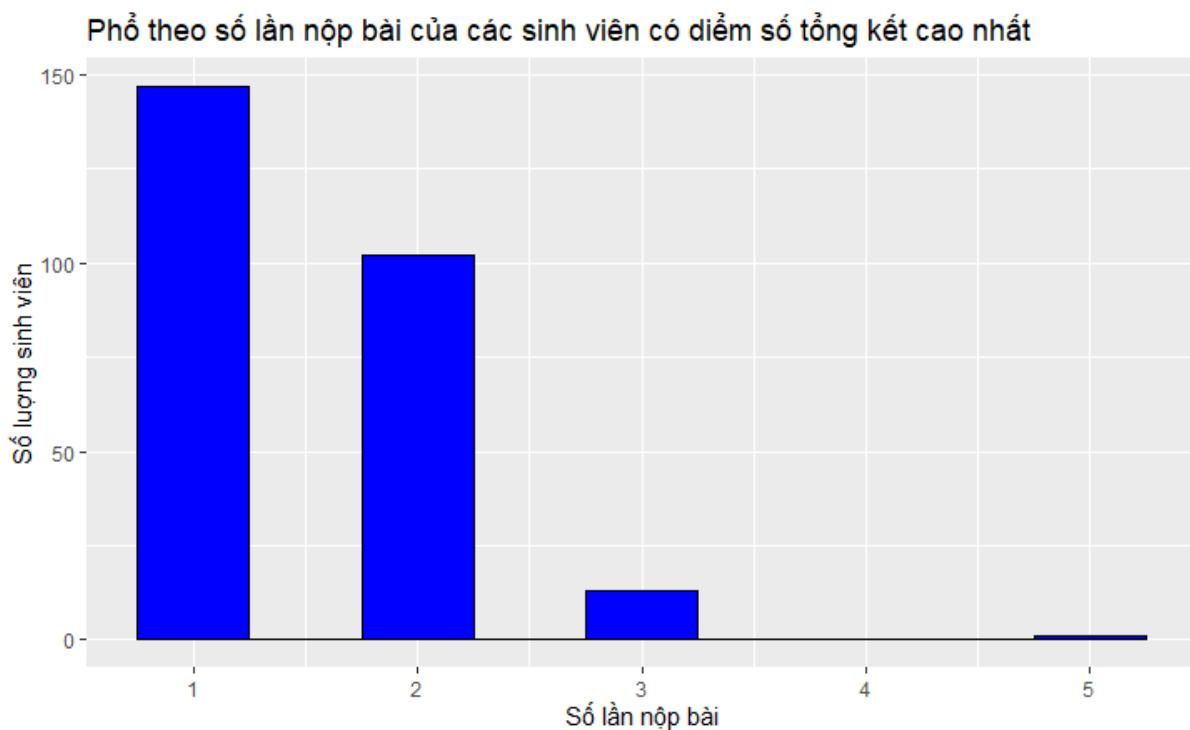
Ta vẽ phô theo số lần nộp bài của các sinh viên có tối thiểu một bài nộp có điểm số cao nhất với trực dọc là số lượng sinh viên, trực ngang là số lần nộp bài của dataframe df_2l.

```
ggplot(df_2l,aes(NoS)) +
  geom_histogram(binwidth = 0.5, fill = "blue", col = "black") +
  xlab("Số lần nộp bài") +
  ylab("Số lượng sinh viên") +
  ggtitle("Phô theo số lần nộp bài của các sinh viên có điểm số tổng kết cao nhất")
```

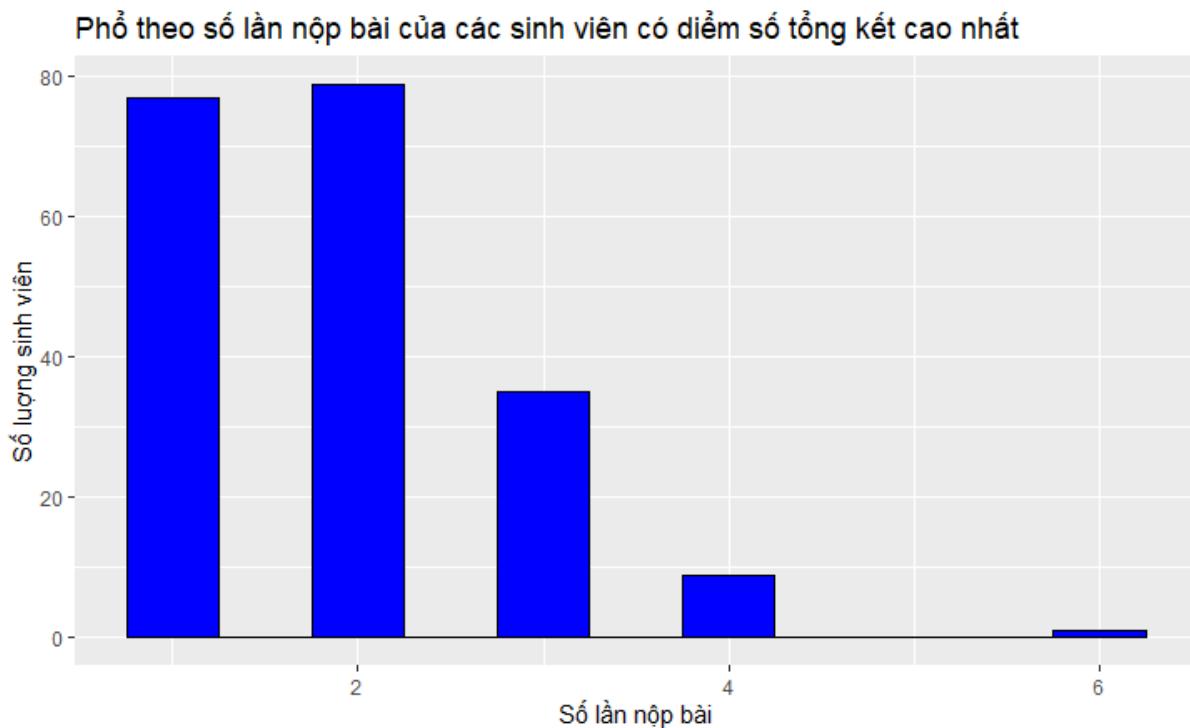
Output file 6:



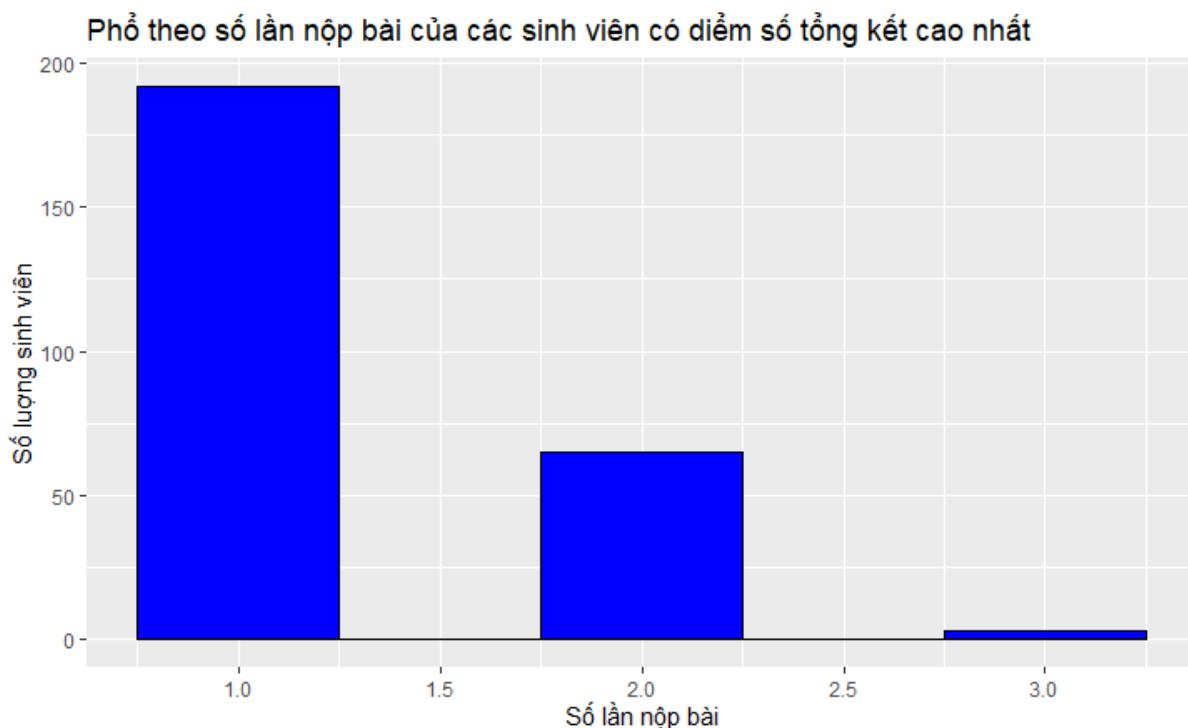
Output file 9:



Output file 10:



Output file 12:



n) Xác định điểm số trung bình của các sinh viên trong mẫu

Ta tính điểm trung bình theo công thức: $\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$

Với n là số lượng sinh viên trong mẫu, x_i là điểm của các sinh viên Ta dùng hàm mean để tính điểm số trung bình và hàm round để làm tròn.

```
print(round(mean(New[["Diem"]]), digits = 1))
```

Output file 6:

```
[1] 9.6
```

Output file 9:

```
[1] 9.3
```

Output file 10:

```
[1] 9.5
```

Output file 12:

```
[1] 9.7
```

o) Xác định số lượng sinh viên có điểm số trung bình

Ta dùng vòng lặp for để kiểm tra điểm của từng sinh viên trong từng lần nộp bài có bằng giá trị trung bình hay không, nếu có tăng giá trị biến đếm f lên 1 đơn vị. Biến đếm f cho ta biết số lượng sinh viên có điểm số trung bình.

Source:

```
f=0 # biến đếm số lượng f
for(i in 1:(n-1)){
  if(New[["Diem"]][i]==round(mean(New[["Diem"]]), digits = 1)) {
    f=f+1
  }
}
print(f)
```

Output file 6:

```
[1] 0
```

Output file 9:

```
[1] 0
```

Output file 10:

```
[1] 256
```

Output file 12:

```
[1] 0
```

p) Tính trung vị mẫu, cực đại mẫu, cực tiểu mẫu của trên

Ta dùng hàm median để tìm trung vị, max, min để tìm cực đại, cực tiểu của mẫu.

```
print(median(New[["Diem"]]))  
print(max(New[["Diem"]]))  
print(min(New[["Diem"]]))
```

Output file 6:

```
[1] 10  
[1] 10  
[1] 0
```

Output file 9:

```
[1] 10  
[1] 10  
[1] 0
```

Output file 10:

```
[1] 9.5  
[1] 10  
[1] 0
```

Output file 12:

```
[1] 10  
[1] 10  
[1] 0
```

q) Hãy đo mức độ phân tán của điểm số (xung quanh giá trị trung bình) của mẫu

Để đo mức độ phân tán của điểm số (xung quanh giá trị trung bình của mẫu) ta sẽ tính phương sai mẫu và độ lệch chuẩn. Ta dùng hàm var() để tính phương sai và dùng hàm sd() để tính độ lệch chuẩn.

```
print(var(New[["Diem"]])) #Phương sai  
print(sd(New[["Diem"]])) #Độ lệch chuẩn
```

Output file 6:

```
[1] 1.115989  
[1] 1.056404
```

Output file 9:



```
[1] 1.643098  
[1] 1.281834
```

Output file 10:

```
[1] 0.5551246  
[1] 0.7450668
```

Output file 12:

```
[1] 0.6682011  
[1] 0.8174357
```

r) Tính độ méo lệch (skewness), và độ nhọn (kurtosis) của dữ liệu trong mẫu trên.

Ta áp dụng công thức tính độ méo lệch và độ nhọn: $Skew = \frac{\frac{1}{n} \sum_{i=1}^n (X_i - \bar{X})^3}{\sqrt{\frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X})^2}^3}$ và $Kurt = \frac{\frac{1}{n} \sum_{i=1}^n (X_i - \bar{X})^4}{(\frac{1}{n} \sum_{i=1}^n (X_i - \bar{X})^2)^2}$

Ta tính độ méo lệch bằng hàm skewness, độ nhọn bằng hàm kurtosis trong thư viện e1071.

```
print(skewness(New[["Diem"]]))  
print(kurtosis(New[["Diem"]]))
```

Output file 6:

```
[1] -5.325134  
[1] 37.42494
```

Output file 9:

```
[1] -4.183249  
[1] 24.83796
```

Output file 10:

```
[1] -7.904371  
[1] 94.64424
```

Output file 12:

```
[1] -6.276879  
[1] 60.94515
```

s) Tính tứ phân vị (quartile) thứ nhất (Q_1) và thứ ba (Q_3) của mẫu.

Ta có công thức tính tứ phân vị thứ nhất và thứ ba:

$$Q_1 = 25(n+1)/100$$

$$Q_3 = 75(n+1)/100$$

Ta tính tứ phân vị thứ nhất (Q_1) và thứ ba (Q_3) của mẫu bằng hàm quantile với tứ phân vị thứ nhất là phần tử thứ hai và tứ phân vị thứ ba là phần tử thứ tư của kết quả.

```
print(quantile(New[["Diem"]])[2])  
print(quantile(New[["Diem"]])[4])
```

Output file 6:

```
25%  
9.5  
75%  
10
```

Output file 9:

```
25%  
9  
75%  
10
```

Output file 10:

25%
9.5
75%
10

Output file 12:

25%
9
75%
10

t) Xác định số lượng sinh viên có điểm số nằm trong 2 mức điểm cao nhất

Ta tạo vector arrangedMark chứa các loại điểm theo thứ tự tăng dần và tìm số phần tử nMark của vector này. Sinh viên có điểm số nằm trong 2 mức điểm cao nhất là sinh viên có điểm số nằm bằng giá trị arrangedMark[nMark] hoặc arrangedMark[nMark-1]. Vì đây là 2 mức điểm cao nhất nên sinh viên có điểm số nằm trong 2 mức này cũng là số sinh viên có điểm số tổng kết nằm trong 2 mức này. Số lượng sinh viên có điểm số nằm trong 2 mức điểm cao nhất chính là số hàng của Submissions sau khi lọc theo điều kiện trên.

```
arrangedMark<-c(sapxep_Diem[["Diem"]][1])
for(i in 2:(n-1)){
  if(sapxep_Diem[["Diem"]][i] != sapxep_Diem[["Diem"]][i-1]){
    arrangedMark<-c(arrangedMark,sapxep_Diem[["Diem"]][i]) }
  }# tạo arrangedMark
nMark=sum(!is.na(arrangedMark))# tìm số phần tử của arrangedMark
print(nrow(filter(Submissions,(Mark == arrangedMark[nMark]) | (Mark ==
arrangedMark[nMark-1])))) # in ra số lượng sinh viên
```

Output file 6:

[1] 310

Output file 9:

[1] 266

Output file 10:

[1] 267

Output file 12:

[1] 277

u) Xác định phổ theo số lần nộp bài của các sinh viên có điểm số tổng kết ở 2 mức điểm cao nhất

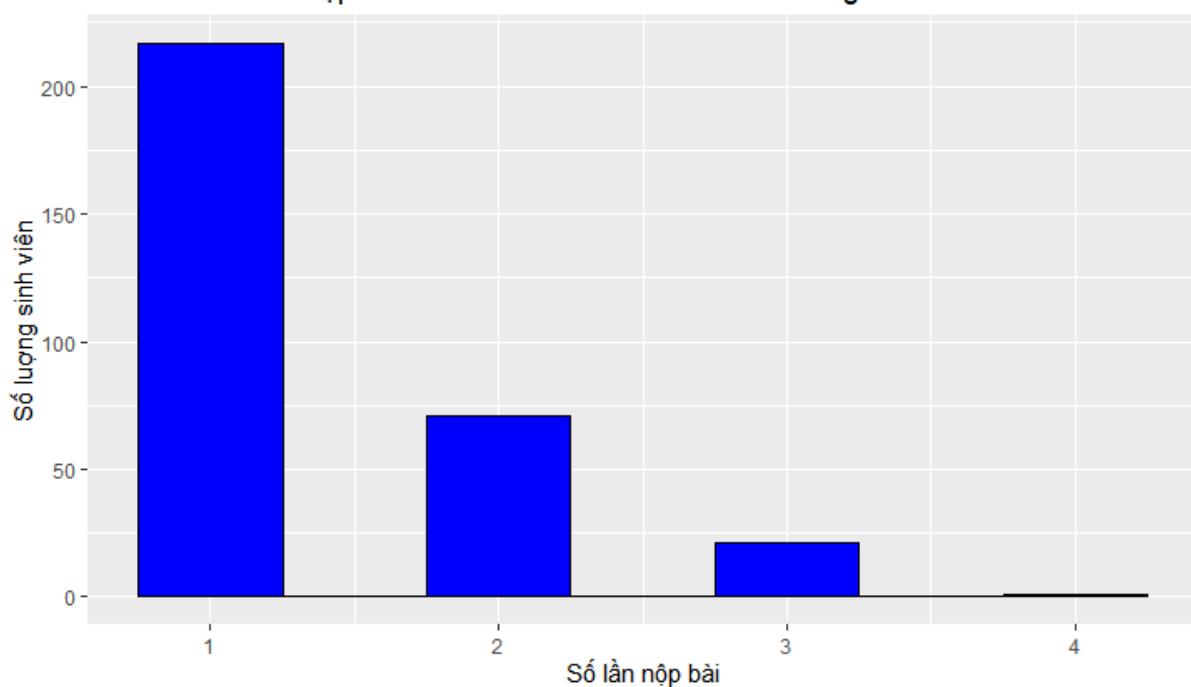
Ta vẽ phổ theo số lần nộp bài của các sinh viên có điểm số tổng kết ở 2 mức điểm cao nhất với trực dọc là số lượng sinh viên, trực ngang là số lần làm bài.

Source:

```
ggplot(filter(Submissions,(Mark == arrangedMark[nMark]) | (Mark ==
arrangedMark[nMark-1])), aes(NoS)) +
  geom_histogram(binwidth = 0.5, fill = "blue", col = "black") +
  xlab("Số lần nộp bài") +
  ylab("Số lượng sinh viên") +
  ggtitle("Phổ theo số lần nộp bài của các sinh viên có điểm số tổng kết ở 2 mức
điểm cao nhất")
```

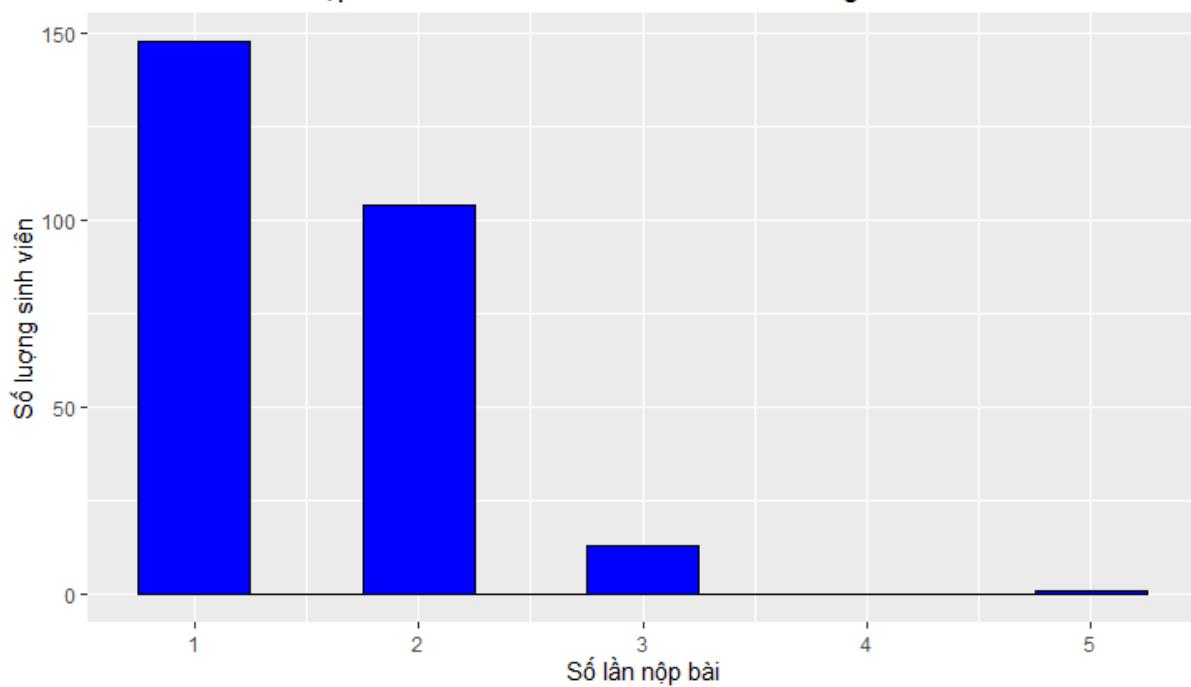
Output file 6:

Phổ theo số lần nộp bài của các sinh viên có điểm số tổng kết ở 2 mức điểm cao nhất

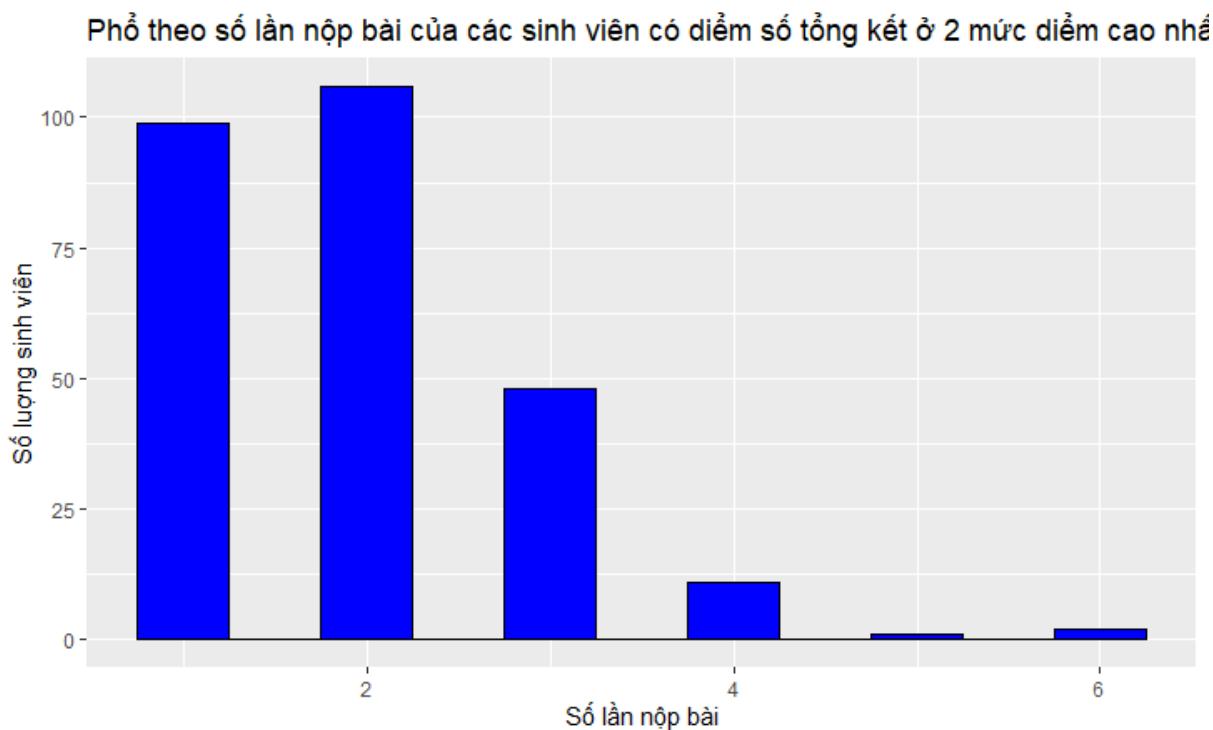


Output file 9:

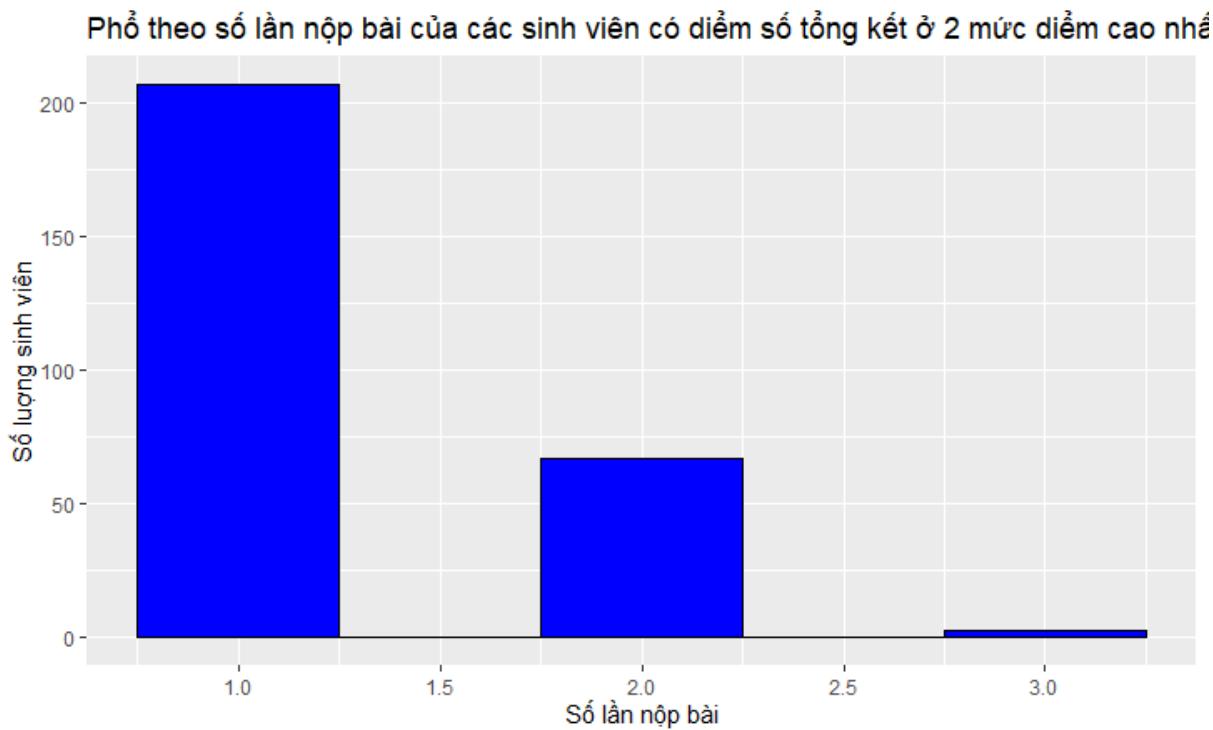
Phổ theo số lần nộp bài của các sinh viên có điểm số tổng kết ở 2 mức điểm cao nhất



Output file 10:



Output file 12:



- v) Xác định số lượng sinh viên có điểm số tổng kết ở mức điểm cao thứ k với k cho trước
Số lượng sinh viên có điểm số tổng kết ở mức điểm cao thứ k với k cho trước chính là số hàng của Submissions sau khi lọc theo điều kiện trên.
Lấy ví dụ với $k=3$.

```
k=3  
print(nrow(filter(Submissions,Mark==arrangedMark[nMark-k+1])))
```

Output file 6:

[1] 8

Output file 9:

[1] 27

Output file 10:

[1] 14

Output file 12:

[1] 3

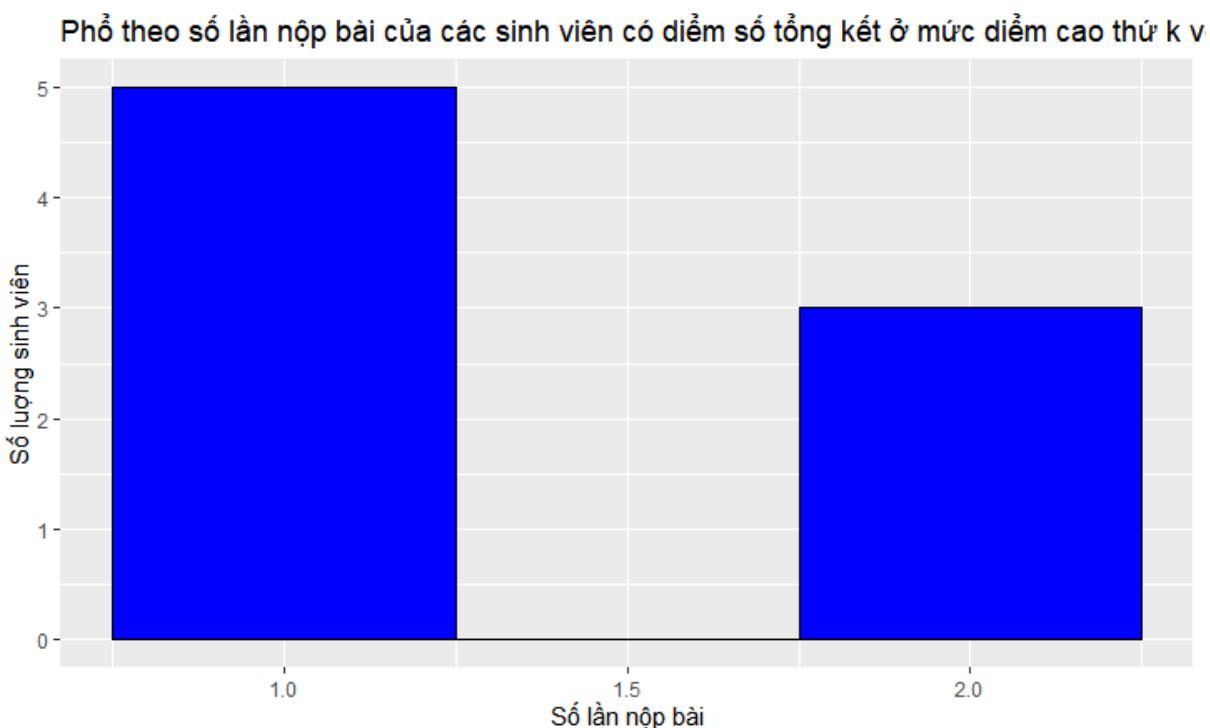
- w) Xác định phổ theo số lần nộp bài của các sinh viên có điểm số tổng kết ở mức điểm cao thứ k với k cho trước

Ta vẽ phổ theo số lần nộp bài của các sinh viên có điểm số tổng kết ở mức điểm cao thứ k với k cho trước với trục dọc là số lượng sinh viên, trục ngang là số lần làm bài.

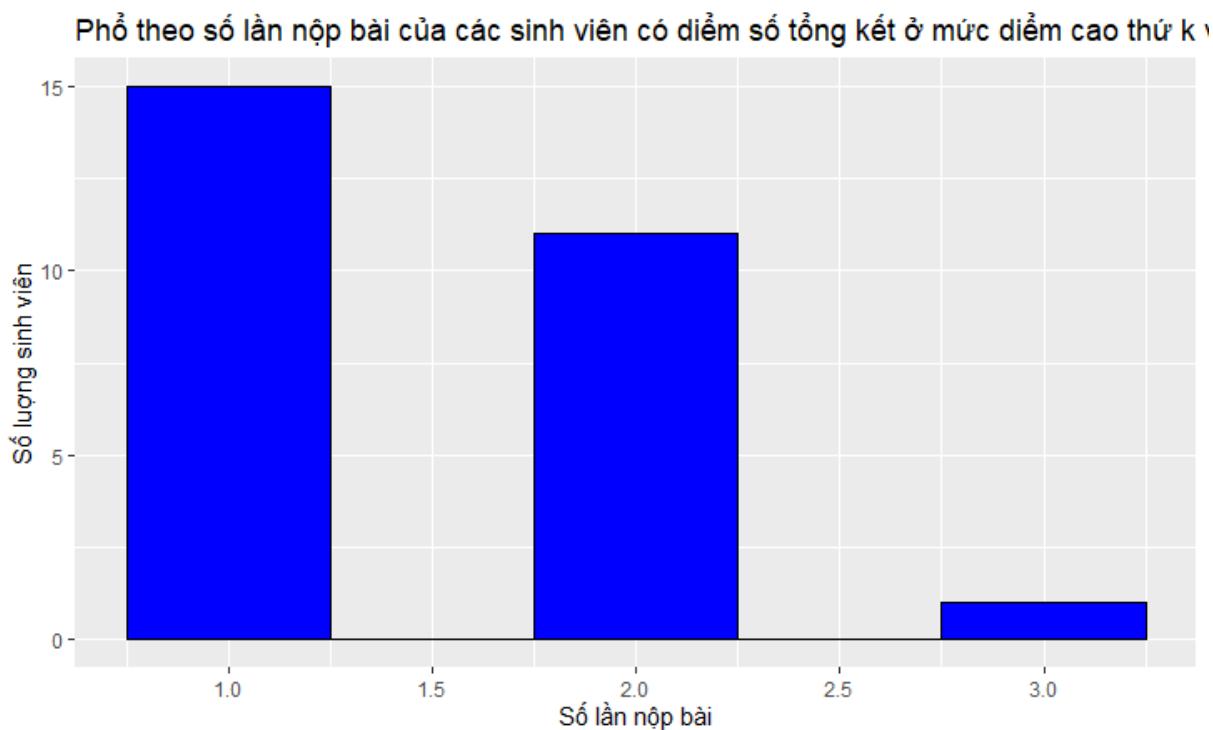
Lấy ví dụ với $k=3$.

```
k=3
Submissions % > %
filter(Mark==arrangedMark[nMark-k+1]) % > %
ggplot(aes(NoS)) +
  geom_histogram(binwidth = 0.5, fill = "blue", col = "black") +
  xlab("Số lần nộp bài") +
  ylab("Số lượng sinh viên") +
  ggtitle("Phổ theo số lần nộp bài của các sinh viên có điểm số tổng kết ở mức điểm cao thứ k với k cho trước")
```

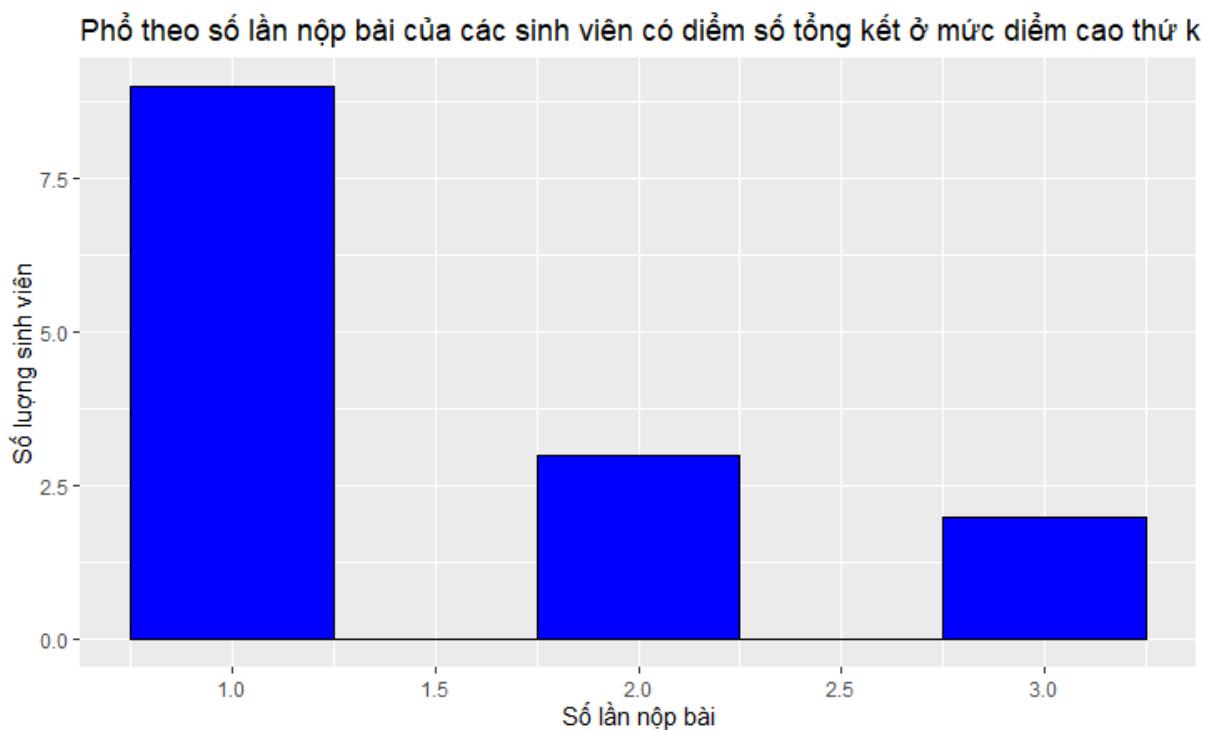
Output file 6:



Output file 9:

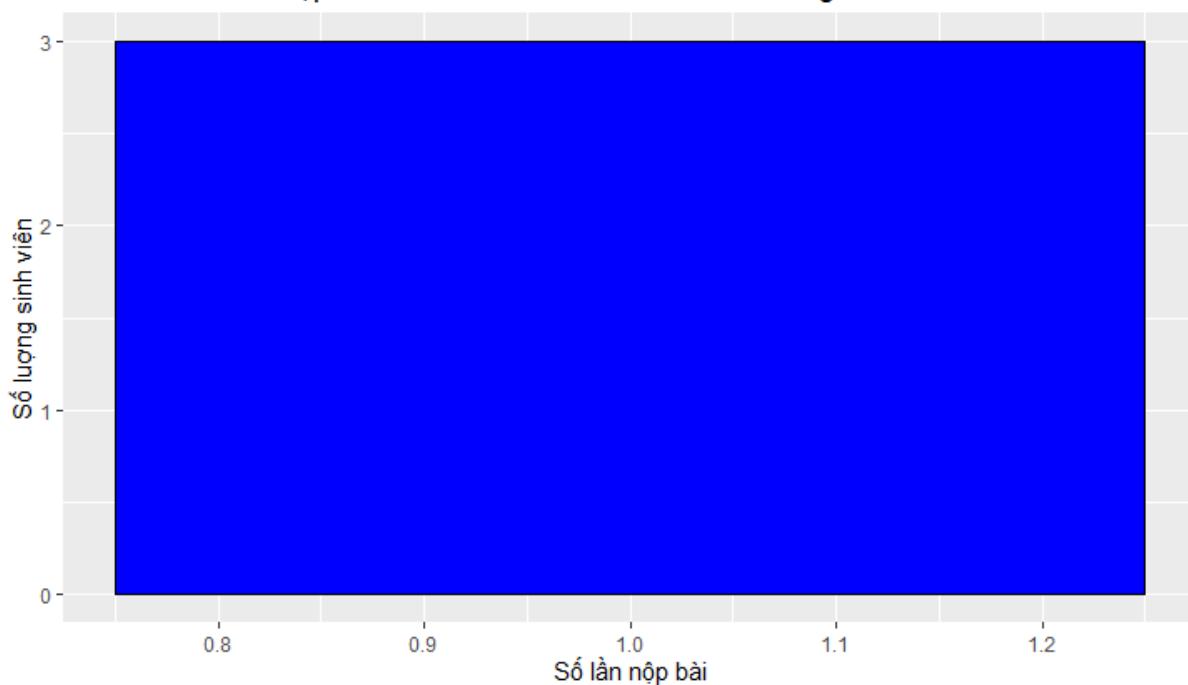


Output file 10:



Output file 12:

Phân theo số lần nộp bài của các sinh viên có điểm số tổng kết ở mức điểm cao thứ k v



4.3 Câu 3: Nhóm câu hỏi liên quan đến số lần nộp bài

a) Xác định số lần nộp bài ít nhất

Để xác định số lần nộp bài ít nhất ta dùng vòng lặp for. Kết quả số lần nộp bài của mỗi sinh viên sẽ được truyền vào một vector (giả sử tên là `numOfSubs`), ta sẽ tạo một dataframe mới từ vector `IDlist` và `numOfSubs`. Lưu ý 2 vector này đều được tạo từ vòng lặp for theo thứ tự ID có giá trị tăng dần nên thông số giữa 2 vector này là tương ứng. Giá trị đầu tiên của vector `numOfSubs` chính là số lần nộp bài ít nhất.

```
numOfSubs=c() #Vecto dem so lan nop bai
f=1 #Bien dem so lan nop bai
for(i in 1:(n-1)){
  if(sapxep_ID[["ID"]][i] != sapxep_ID[["ID"]][i+1]){
    numOfSubs <- c(numOfSubs, f)
    f=1
  }
  if(i==n-1){
    numOfSubs<-c(numOfSubs, f)
  }
}
else{
  f = f+1
  if(i==n-1){
    numOfSubs<-c(numOfSubs, f)
  }
}
#Tao 1 data frame chua ID va so lan nop bai
New_list <- data.frame(IDlist, numOfSubs)
#Sap xep theo thu tu so lan nop bai tang dan
New_list = arrange(New_list, numOfSubs)
#So lan nop bai it nhat la so lan nop bai o dong dau tien
numOfSubs_min = New_list[["numOfSubs"]][1]
numOfSubs_min
```

Output của file 6:

```
[1] 1
```

Output của file 9:

```
[1] 1
```

Output của file 10:

```
[1] 1  
>
```

Output của file 12:

```
> numofsubs_min  
[1] 1  
> |
```

b) Xác định danh sách các sinh viên có số lần nộp bài ít nhất

Ta xác định số sinh viên có số lần nộp bài ít nhất bằng cách tạo một dataframe là tập con của *New_list* với điều kiện là *numOfSubs=numOfSubs_min*. Danh sách sinh viên cần tìm chính là danh sách sinh viên thuộc tập con vừa tạo.

```
df_3b = subset(New_list, New_list$numOfSubs == numofsubs_min)  
df_3b[["IDlist"]]
```

Output của file 6:

```
[1] "1913014" "1812257" "1812478" "1813528" "1813681" "1814518" "1814611" "1820028" "1852443" "1910032" "1910060" "1910076" "1910094" "1910101"  
[15] "1910110" "1910113" "1910123" "1910137" "1910198" "1910202" "1910224" "1910238" "1910265" "1910298" "1910346" "1910347" "1910402" "1910409"  
[29] "1910473" "1910563" "1910565" "1910620" "1910643" "1910644" "1910663" "1910666" "1910735" "1910892" "1910916" "1910984" "1911000" "1911015"  
[43] "1911044" "1911056" "1911058" "1911066" "1911136" "1911185" "1911207" "1911217" "1911262" "1911283" "1911285" "1911296" "1911314" "1911441"  
[57] "1911456" "1911478" "1911520" "1911530" "1911565" "1911569" "1911591" "1911594" "1911650" "1911704" "1911736" "1911837" "1911841" "1911878"  
[71] "1911931" "1912041" "1912084" "1912123" "1912133" "1912184" "1912267" "1912288" "1912371" "1912384" "1912457" "1912462" "1912522" "1912523"  
[85] "1912526" "1912539" "1912594" "1912602" "1912675" "1912683" "1912705" "1912713" "1912715" "1912749" "1912798" "1912811" "1912817"  
[99] "1912912" "1912958" "1912980" "1913021" "1913026" "1913040" "1913075" "1913094" "1913102" "1913114" "1913167" "1913186" "1913218" "1913228"  
[113] "1913241" "1913254" "1913260" "1913261" "1913268" "1913334" "1913336" "1913354" "1913355" "1913356" "1913380" "1913386" "1913396" "1913419"  
[127] "1913430" "1913446" "1913457" "1913464" "1913467" "1913566" "1913599" "1913621" "1913629" "1913652" "1913678" "1913695" "1913711" "1913729"  
[141] "1913758" "1913763" "1913764" "1913828" "1913832" "1914011" "1914022" "1914038" "1914054" "1914055" "1914064" "1914078" "1914079" "1914084"  
[155] "1914093" "1914121" "1914126" "1914220" "1914235" "1914291" "1914310" "1914316" "1914405" "1914424" "1914472" "1914477" "1914641" "1914651"  
[169] "1914659" "1914677" "1914697" "1914698" "1914738" "1914763" "1914806" "1914809" "1914830" "1914845" "1914864" "1914878" "1914880" "1914881"  
[183] "1914900" "1914914" "1915016" "1915063" "1915071" "1915076" "1915094" "1915130" "1915146" "1915161" "1915251" "1915275" "1915294" "1915323"  
[197] "1915329" "1915351" "1915442" "1915473" "1915482" "1915486" "1915520" "1915540" "1915541" "1915551" "1915557" "1915650" "1915651"  
[211] "1915656" "1915667" "1915745" "1915787" "1915795" "1915822" "1915865" "1915866" "1915931" "1915953" "1915982" "1915988" "1915991" "1916061"  
[225] "1937024" "1937064"  
>
```

Output của file 9:

```
[1] "1812257" "1812478" "1813096" "1813681" "1820028" "1910032" "1910038" "1910076" "1910094" "1910101" "1910110" "1910202" "1910339"  
[14] "1910346" "1910351" "1910473" "1910565" "1910650" "1910663" "1910666" "1910892" "1910916" "1910984" "1911000" "1911015" "1911044"  
[27] "1911056" "1911058" "1911186" "1911207" "1911283" "1911296" "1911314" "1911441" "1911456" "1911478" "1911530" "1911569" "1911591"  
[40] "1911594" "1911704" "1911796" "1911837" "1911878" "1911881" "1911900" "1911975" "1912041" "1912056" "1912084" "1912123" "1912184"  
[53] "1912190" "1912267" "1912288" "1912384" "1912386" "1912410" "1912457" "1912463" "1912522" "1912523" "1912594" "1912675"  
[66] "1912676" "1912683" "1912700" "1912705" "1912715" "1912798" "1912811" "1912817" "1912912" "1912966" "1913014" "1913021" "1913026"  
[79] "1913075" "1913102" "1913114" "1913218" "1913241" "1913254" "1913261" "1913268" "1913334" "1913336" "1913354" "1913355" "1913380"  
[92] "1913386" "1913419" "1913446" "1913457" "1913464" "1913566" "1913599" "1913621" "1913629" "1913651" "1913652" "1913729" "1913763"  
[105] "1913764" "1913775" "1913817" "1913832" "1913844" "1913918" "1913944" "1913990" "1914022" "1914052" "1914054" "1914055" "1914078"  
[118] "1914079" "1914121" "1914126" "1914220" "1914232" "1914291" "1914310" "1914384" "1914404" "1914451" "1914467" "1914713" "1914768"  
[131] "1914806" "1914807" "1914845" "1914864" "1914914" "1915063" "1915071" "1915076" "1915146" "1915251" "1915275" "1915294" "1915351"  
[144] "1915439" "1915442" "1915473" "1915474" "1915482" "1915486" "1915520" "1915540" "1915541" "1915557" "1915656" "1915745" "1915787" "1915865"  
[157] "1915905" "1915919" "1915953" "1915988" "1915991" "1916022" "1937024"
```

Output của file 10:

```
[1] "1812257" "1812478" "1813681" "1910032" "1910038" "1910076" "1910094" "1910123" "1910224" "1910346" "1910473" "1910565" "1910984" "1911000"  
[15] "1911136" "1911207" "1911283" "1911441" "1911530" "1911565" "1911569" "1911594" "1911704" "1911837" "1911841" "1911931" "1912041"  
[29] "1912123" "1912190" "1912286" "1912463" "1912523" "1912539" "1912579" "1912675" "1912683" "1912705" "1912761" "1912912"  
[43] "1912966" "1913045" "1913052" "1913102" "1913218" "1913244" "1913261" "1913268" "1913305" "1913334" "1913356" "1913380"  
[57] "1913386" "1913419" "1913446" "1913457" "1913464" "1913599" "1913621" "1913652" "1913679" "1913695" "1913729" "1913811" "1913944" "1914011"  
[71] "1914022" "1914054" "1914078" "1914084" "1914121" "1914220" "1914227" "1914232" "1914291" "1914424" "1914472" "1914477" "1914641" "1914651"  
[85] "1914807" "1914880" "1915076" "1915133" "1915146" "1915251" "1915275" "1915294" "1915350" "1915351" "1915439" "1915473" "1915482"  
[99] "1915486" "1915551" "1915557" "1915562" "1915787" "1915795" "1915865" "1915903" "1915983" "1915988" "1916022" "1937024" "1937064"  
>
```

Output của file 12:

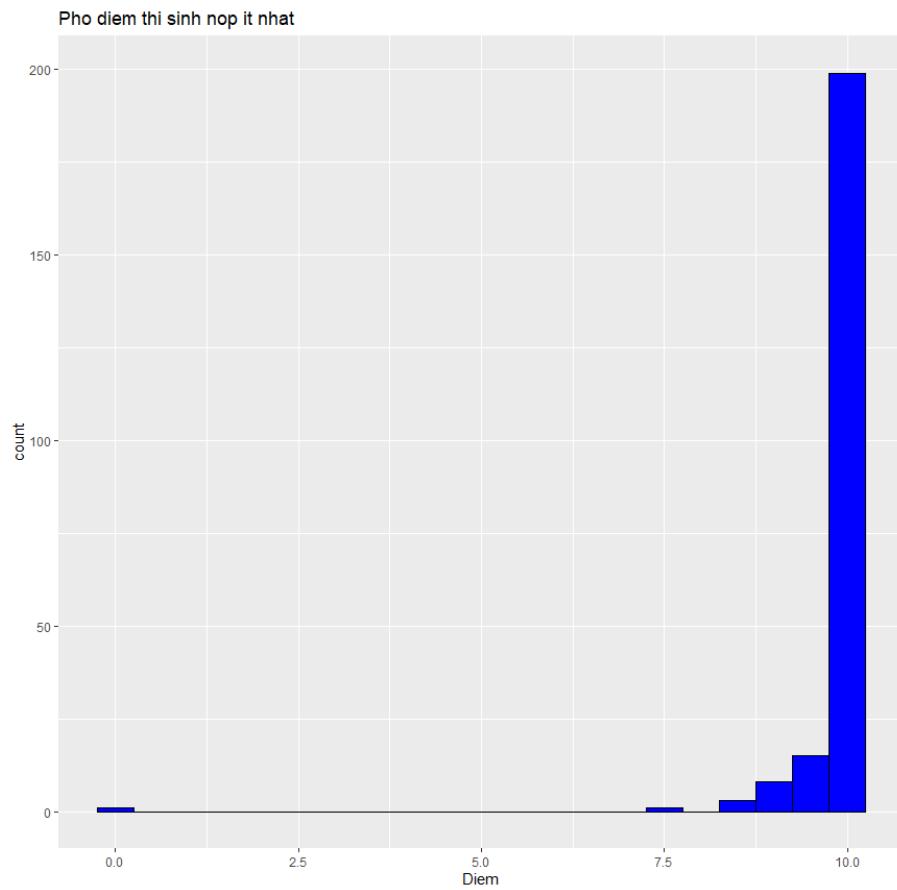
```
[1] "1613010" "1812257" "1812478" "1813096" "1813681" "1814096" "1814518" "1820028" "1910006" "1910032" "1910038" "1910060" "1910076" "1910094"  
[15] "1910101" "1910110" "1910113" "1910137" "1910202" "1910224" "1910238" "1910265" "1910276" "1910298" "1910339" "1910346" "1910347" "1910351"  
[29] "1910402" "1910473" "1910565" "1910620" "1910643" "1910644" "1910650" "1910663" "1910666" "1910865" "1910916" "1910984" "1911000" "1911015"  
[43] "1911056" "1911058" "1911110" "1911136" "1911185" "1911186" "1911207" "1911217" "1911283" "1911285" "1911296" "1911314" "1911478" "1911520"  
[57] "1911530" "1911565" "1911569" "1911591" "1911594" "1911650" "1911704" "1911736" "1911796" "1911837" "1911841" "1911881" "1911900" "1911907"  
[71] "1911931" "1912041" "1912046" "1912056" "1912084" "1912184" "1912190" "1912237" "1912267" "1912384" "1912386" "1912463" "1912523" "1912539"  
[85] "1912579" "1912675" "1912683" "1912700" "1912715" "1912749" "1912811" "1912817" "1912912" "1912916" "1912966" "1912980" "1913026" "1913040"  
[99] "1913102" "1913114" "1913123" "1913167" "1913186" "1913218" "1913241" "1913254" "1913261" "1913268" "1913306" "1913334" "1913336" "1913341"  
[113] "1913354" "1913355" "1913356" "1913396" "1913419" "1913433" "1913446" "1913457" "1913467" "1913566" "1913651" "1913652" "1913678"  
[127] "1913695" "1913713" "1913729" "1913763" "1913764" "1913775" "1913817" "1913828" "1913832" "1913844" "1913949" "1914003" "1914011"  
[141] "1914022" "1914038" "1914052" "1914054" "1914055" "1914078" "1914079" "1914084" "1914121" "1914210" "1914232" "1914291" "1914310" "1914316"  
[155] "1914384" "1914405" "1914424" "1914427" "1914472" "1914474" "1914641" "1914651" "1914674" "1914698" "1914720" "1914738" "1914763" "1914768" "1914807"  
[169] "1914845" "1914864" "1914880" "1914881" "1914914" "1914979" "1915063" "1915071" "1915076" "1915251" "1915275" "1915294" "1915323" "1915329"  
[183] "1915351" "1915439" "1915473" "1915479" "1915482" "1915486" "1915540" "1915551" "1915557" "1915562" "1915570" "1915598" "1915651" "1915656"  
[197] "1915745" "1915787" "1915822" "1915863" "1915866" "1915882" "1915903" "1915953" "1915982" "1915991" "1916022" "1916061" "1937024" "1937064"
```

c) Xác định phô diễm của các sinh viên có số lần nộp bài ít nhất

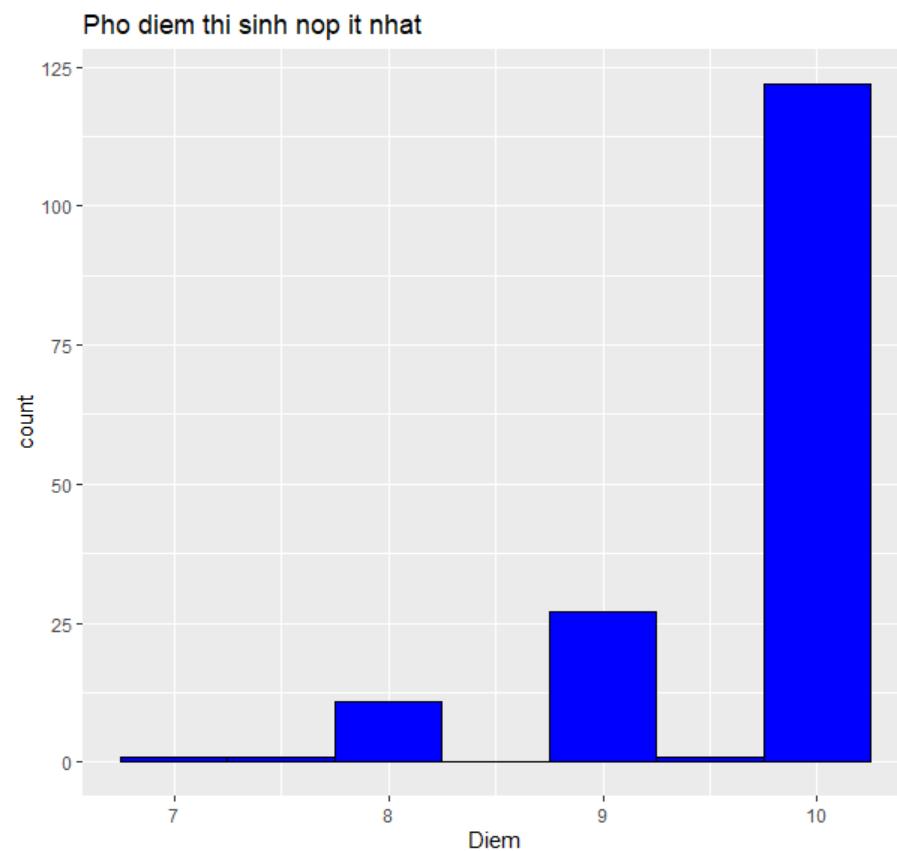
Dầu tiên ta tạo một vecto đặt tên là *New_list1* là vecto con của *New_list* với điều kiện là số lần nộp bài của sinh viên bằng với số lần nộp bài ít nhất của sinh viên đã tìm được ở trên. Sau đó ta dùng vòng *for*, với mỗi hàng dữ liệu trong *New_list1* ta sẽ dò tất cả sinh viên trong *mydata* để tìm ra sinh viên tương ứng. Đầu tiên ta tạo một data frame tên là *min_spec* rỗng. Nếu tìm thấy sinh viên nào ta lưu vào vecto *add* sau đó nối cho data frame ta đã tạo. Để vẽ phô diễm, ta dùng hàm *ggplot* với loại biểu đồ là *histogram*, hay còn gọi là biểu đồ phân bố.

```
New_list1 = c()  
New_list1 = subset(New_list, num0fSubs == num0fSubs_min)  
min_spec = data.frame()  
for (i in 1:nrow(New_list1))  
{  
  for (j in 1:nrow(mydata))  
  {  
    if (mydata[[1]][j] == New_list1[[1]][i])  
    {  
      add = c(mydata[["Diem"]][j])  
      min_spec = rbind(min_spec, add)  
    }  
  }  
}  
colnames(min_spec) = c("Diem")  
ggplot(min_spec, aes(Diem))+geom_histogram(binwidth = 0.5, fill ="blue", col = "black")+ ggtitle("Phô diem thi sinh nopol it nhat")
```

Output của file 6:

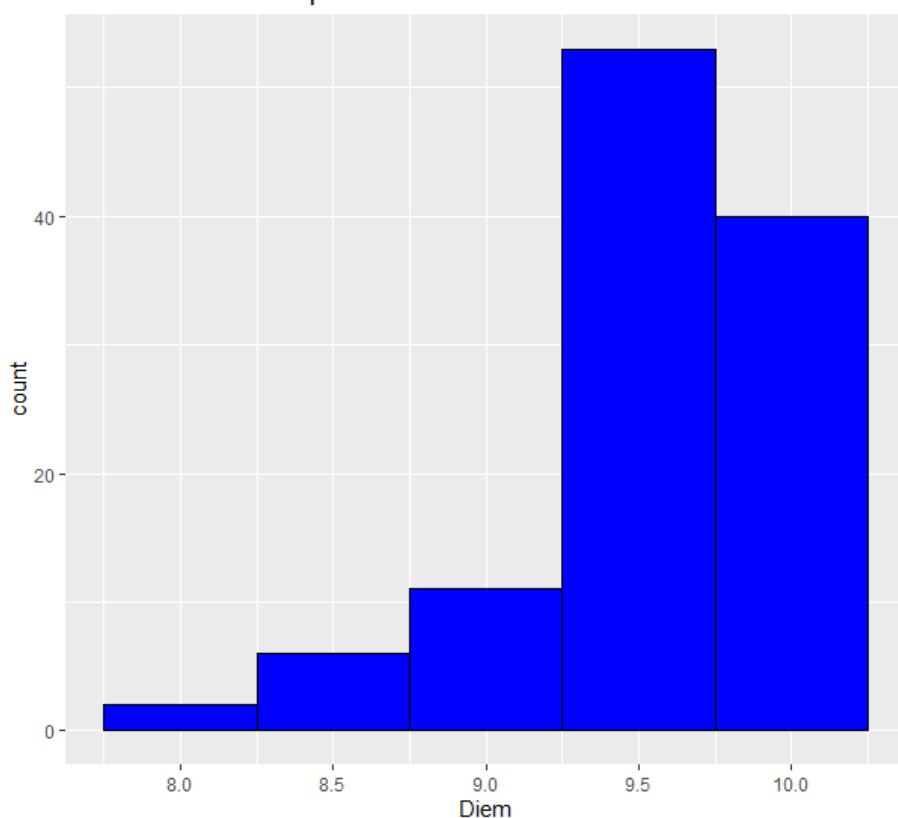


Output của file 9:

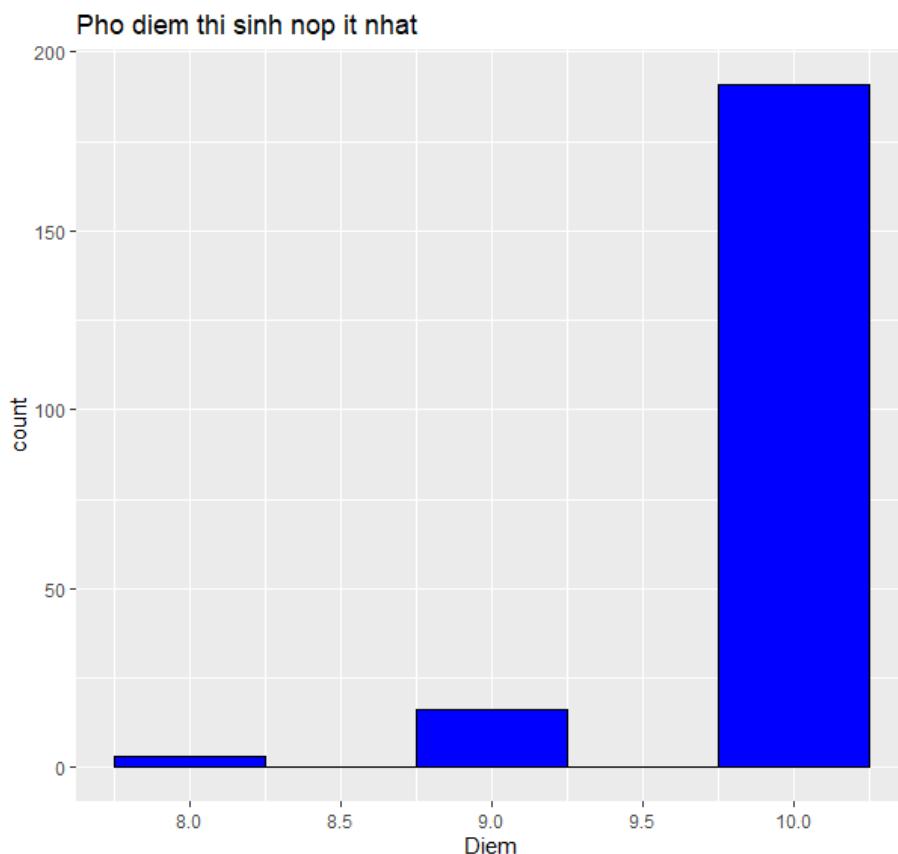


Output của file 10:

Phân佈 điểm thi sinh nộp ít nhất



Output của file 12:



- d) Xác định số lần nộp bài nhiều nhất

Ta đã có danh sách *New_list* là danh sách các bạn sinh viên theo thứ tự số lần nộp bài tăng dần và đã



được tìm được ở các câu trên. Vậy ta chỉ cần truy xuất đến số lần nộp bài của bạn sinh viên cuối danh sách sẽ tìm được số lần nộp bài nhiều nhất.

```
numOfSubs_max = New_list[["numOfSubs"]][nrow(New_list)]  
numOfSubs_max
```

Output của file 6:

```
[1] 4
```

Output của file 9:

```
| [1] 5
```

Output của file 10:

```
[1] 6  
> |
```

Output của file 12:

```
> numOfSubs_max  
[1] 3  
> |
```

- e) Xác định các sinh viên có số lần nộp bài nhiều nhất
Cách làm tương tự với câu 3b.

```
df_3e = subset(New_list, New_list$numOfSubs == numOfSubs_max)  
df_3e[["IDlist"]]
```

Output của file 6:

```
[1] "1937019"
```

Output của file 9:

```
| [1] "1914210" |
```

Output của file 10:

```
| [1] "1913021" "1913167"  
> |
```

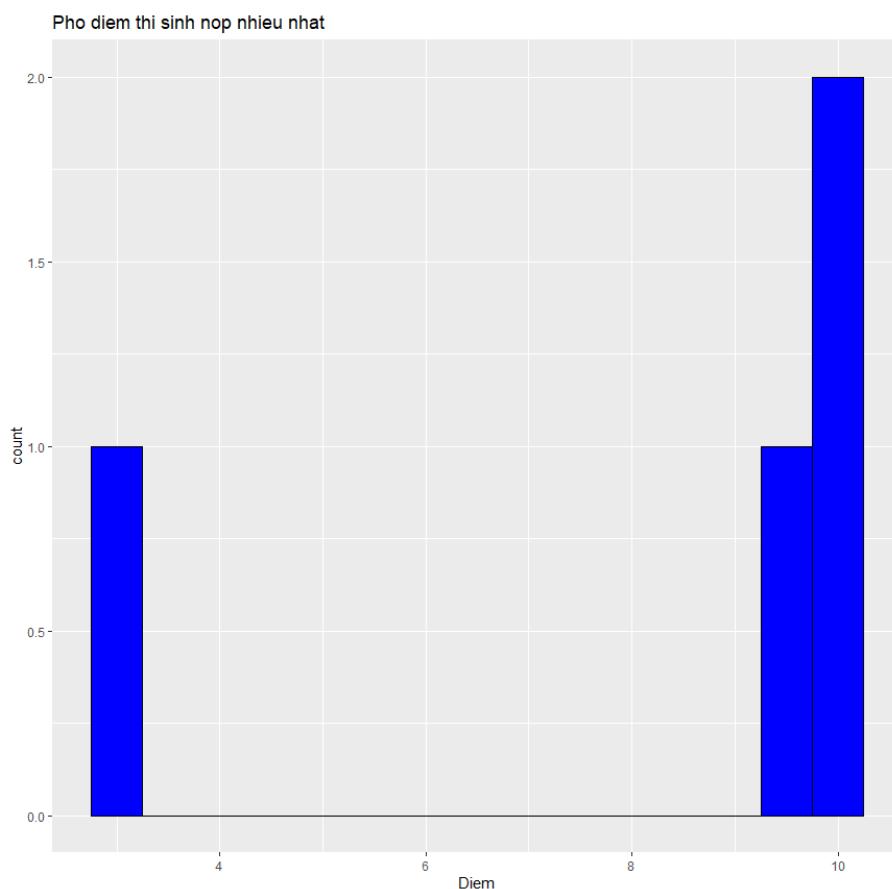
Output của file 12:

```
> df_3e[["IDlist"]]  
[1] "1913045" "1915520" "1927007  
> |
```

- f) Xác định phổ điểm của các sinh viên có số lần nộp bài nhiều nhất
Ta làm tương tự câu 3c

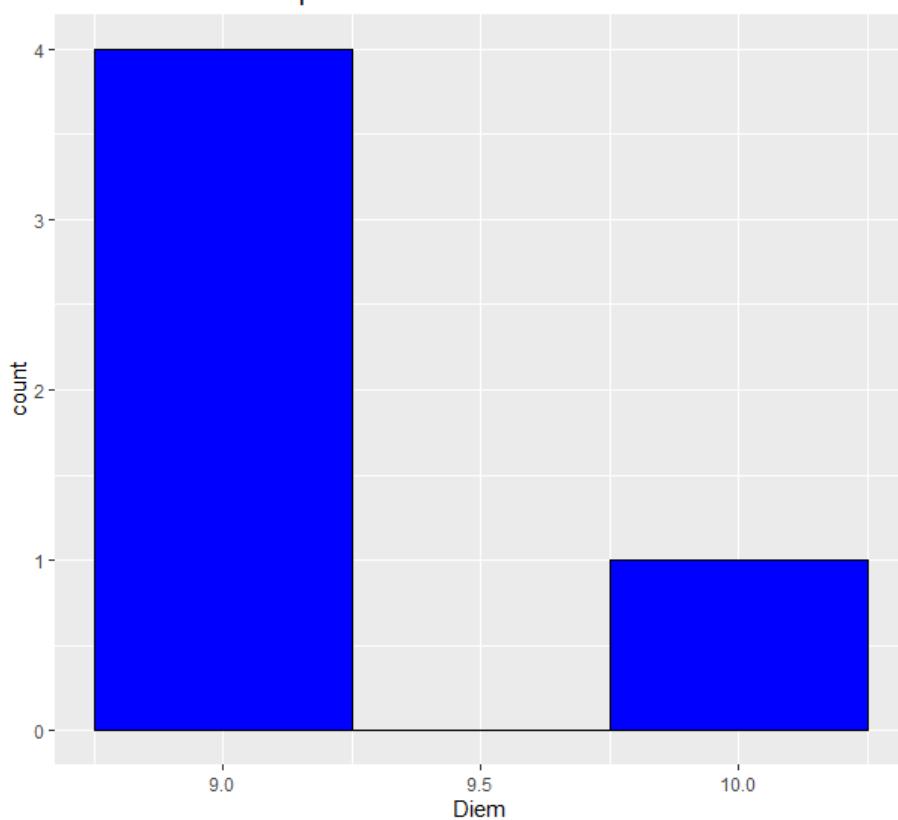
```
New_list4 = c()  
New_list4 = subset(New_list, numOfSubs == numOfSubs_max)  
min_spec2 = data.frame()  
for (i in 1:nrow(New_list4))  
{  
  for (j in 1:nrow(mydata))  
  {  
    if (mydata[[1]][j] == New_list4[[1]][i])  
    {  
      add2 = c(mydata[["Diem"]][j])  
      min_spec2 = rbind(min_spec2, add2)  
    }  
  }  
}  
colnames(min_spec2) = c("Diem")  
ggplot(min_spec2, aes(Diem))+geom_histogram(binwidth = 0.5, fill ="blue", col = "black")+ ggtitle("Pho diem thi sinh nopol nhieu nhat")
```

Output của file 6:



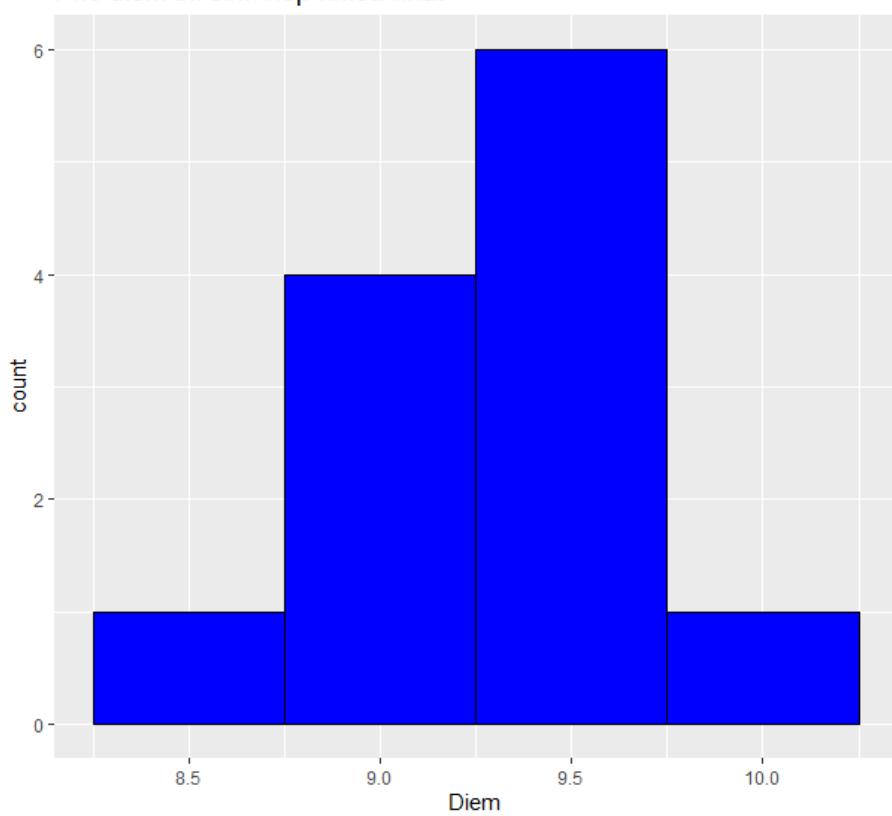
Output của file 9:

Phân佈 điểm thi sinh nộp nhiều nhất



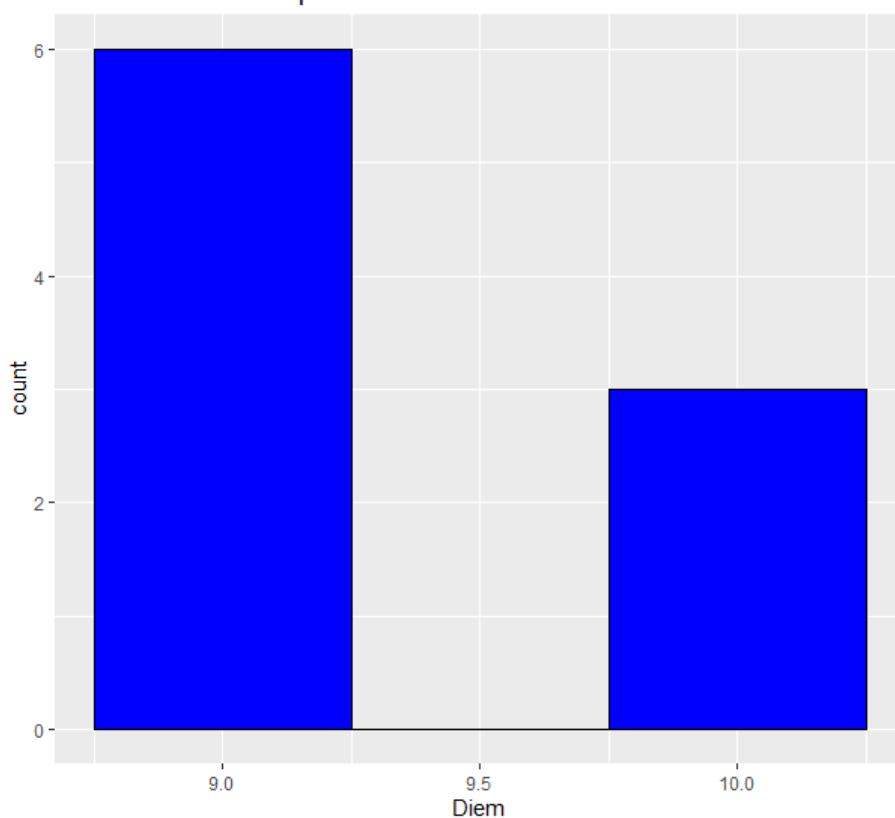
Output của file 10:

Phân佈 điểm thi sinh nộp nhiều nhất



Output của file 12:

Phân佈 điểm thi sinh nộp nhiều nhất



- g) Xác định số lần nộp bài trung bình của của các sinh viên
Ta sẽ dùng hàm mean() để xác định số lần nộp bài trung bình của sinh viên.

```
averagesubs = mean(New_list$numOfSubs)  
averagesubs
```

Output của file 6:

```
[1] 1.369565
```

Output của file 9:

```
[1] 1.522876
```

Output của file 10:

```
[1] 1.906574  
> |
```

Output của file 12:

```
> averagesubs  
[1] 1.260714  
> |
```

- h) Xác định số lượng sinh viên có số lần nộp trung bình
Đầu tiên ta tìm danh sách các bạn sinh viên có số lần nộp bài bằng với số lần nộp bài trung bình bằng cách so số lần nộp bài của các bạn trong New_list với số lần nộp bài trung bình. Sau khi có được danh sách ta



dùng hàm `length()` để tìm độ dài vecto vừa tìm được sẽ được số sinh viên có số lần nộp bài bằng số lần nộp trung bình

```
average_stus = c()
for (i in 1:numofStus){
  if (New_list[["numOfSubs"]][i] == averagesubs){
    average_stus = c(average_stus, New_list[["IDlist"]][i])
  }
}
average_num = length(average_stus)
average_num
```

Output của file 6:

```
> average_num
[1] 0
>
```

Output của file 9:

```
[1] 0
```

Output của file 10:

```
[1] 0
>
```

Output của file 12:

```
|> average_num
| [1] 0
|>
```

- i) Xác định phô theo điểm số của các sinh viên có lần nộp bài trung bình
Ta làm tương tự câu 3c

```
New_list_average = c()
New_list_average = subset(New_list, numOfSubs == averagesubs)
while(nrow(New_list_average)!=0){
  min_spec3 = data.frame()
  for (i in 1:nrow(New_list_average))
  {
    for (j in 1:nrow(mydata))
    {
      if (mydata[[1]][j] == New_list_average[[1]][i])
      {
        add3 = c(mydata[["Diem"]][j])
        min_spec3 = rbind(min_spec3, add3)
      }
    }
  }
  colnames(min_spec3) = c("Diem")
  ggplot(min_spec3, aes(Diem))+geom_histogram(binwidth = 0.5, fill ="blue", col =
"black")+ ggttitle("Phô diem thi sinh co so lan nop trung binh") }
```

Output của file 6: Do không có sinh viên nào có số lần nộp bài bằng số lần nộp bài trung bình nên ta không thể vẽ được phô điểm

Output của file 9: Do không có sinh viên nào có số lần nộp bài bằng số lần nộp bài trung bình nên ta không thể vẽ được phô diễm

Output của file 10: Do không có sinh viên nào có số lần nộp bài bằng số lần nộp bài trung bình nên ta không thể vẽ được phô diễm

Output của file 12: Do không có sinh viên nào có số lần nộp bài bằng số lần nộp bài trung bình nên ta không thể vẽ được phô diễm

- j) Tính trung vị mẫu, cực đại mẫu, cực tiểu mẫu của trên.

Ta dùng hàm `median()` để tìm trung vị mẫu, hàm `max()` để tìm cực đại mẫu và hàm `min()` để tìm cực tiểu mẫu

```
median(New_list$numOfSubs)  
min(New_list$numOfSubs)  
max(New_list$numOfSubs)
```

Output của file 6:

```
> median(New_list$numofsubs)  
[1] 1  
> min(New_list$numofsubs)  
[1] 1  
> max(New_list$numofsubs)  
[1] 4  
>
```

Output của file 9:

```
> median(New_list$numofsubs)  
[1] 1  
> min(New_list$numofsubs)  
[1] 1  
> max(New_list$numofsubs)  
[1] 5  
>
```

Output của file 10:

```
> median(New_list$numOfSubs)  
[1] 2  
> min(New_list$numOfSubs)  
[1] 1  
> max(New_list$numOfSubs)  
[1] 6  
>
```

Output của file 12:

```
> median(New_list$numofsubs)  
[1] 1  
> min(New_list$numofsubs)  
[1] 1  
> max(New_list$numofsubs)  
[1] 3  
>
```

- k) Hãy đo mức độ phân tán của điểm số (xung quanh giá trị trung bình) của mẫu.

Để đo mức độ phân tán, ta tìm độ lệch mẫu của số lần nộp quanh giá trị trung bình bằng hàm `sd()`

```
sd(New_list$numOfSubs)
```

Output của file 6:

```
[1] 0.6190681
```

Output của file 9:

```
[1] 0.6176101
```

Output của file 10:

```
> sd(New_list$numofsubs)
[1] 0.9251079
> |
```

Output của file 12:

```
> sd(New_list$numofsubs)
[1] 0.4636147
>
```

- l) Tính độ méo lệch (skewness), và độ nhọn (kurtosis) của dữ liệu trong mẫu trên.

Ta áp dụng công thức tính độ méo lệch $Skew = \frac{\frac{1}{n} \sum_{i=1}^n (X_i - \bar{X})^3}{\sqrt{\frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X})^2}^3}$ và độ nhọn $Kurt = \frac{\frac{1}{n} \sum_{i=1}^n (X_i - \bar{X})^4}{(\frac{1}{n} \sum_{i=1}^n (X_i - \bar{X})^2)^2}$

Trong ngôn ngữ R, ta dùng hàm *skewness()* để tìm độ méo lệch và hàm *kurtosis()* để tìm độ nhọn

```
skewness(New_list$numOfSubs)  
kurtosis(New_list$numOfSubs)
```

Output của file 6:

```
> skewness(New_List$numofsubs)
[1] 1.52303
> kurtosis(New_List$numofsubs)
[1] 1.486368
>
```

Output của file 9:

```
> skewness(New_list$numOfSubs)
[1] 1.078709
> kurtosis(New_list$numOfSubs)
[1] 2.202863
```

Output của file 10:

```
> skewness(New_list$numofsubs)
[1] 1.102738
> kurtosis(New_list$numofsubs)
[1] 1.792094
>
```

Output của file 12:

```
> skewness(New_list$numOfSubs)
[1] 1.402593
> kurtosis(New_list$numOfSubs)
[1] 0.7445292
>
```

- m) Tính tứ phân vị (quartile) thứ nhất (Q_1) và thứ ba (Q_3) của mẫu

Ta có công thức tính tứ phân vị thứ nhất và thứ ba:

$$Q_1 = 25(n + 1)/100$$

$$Q_3 = 75(n + 1)/100$$

Trong ngôn ngữ R, ta dùng hàm `quantile()` để tìm tứ phân vị như sau

```
quantile(New_list$numOfSubs)
```

Output của file 6:

```
> quantile(New_list$numOfSubs)
 0% 25% 50% 75% 100%
 1   1   1   2   4
> |
```

Output của file 9:

```
> quantile(New_list$numofsubs)
 0% 25% 50% 75% 100%
 1   1   1   2   5
> |
```

Output của file 10:

```
> quantile(New_list$numofsubs)
 0% 25% 50% 75% 100%
 1   1   2   2   6
> |
```

Output của file 12:

```
> quantile(New_list$numofsubs)
 0% 25% 50% 75% 100%
1.00 1.00 1.00 1.25 3.00
> |
```

- n) Xác định danh sách các sinh viên nằm trong nhóm có số lần nộp bài nhiều nhì

Dầu tiên ta lập một danh sách các lần nộp bài, đặt tên là `substime`. Vì ta lập `substime` từ `New_list` nên `substime` sẽ được xếp theo thứ tự tăng dần. Ta chỉ cần truy xuất đến phần tử kế cuối thì sẽ được số lần nộp nhiều nhì. Từ số lần nộp nhiều nhì ta sẽ tìm được sinh danh sách sinh viên nộp nhiều nhì tương tự như cách ta tìm danh sách sinh viên nộp nhiều nhất và ít nhất.

```
#Lap danh sach so cac lan nop bai
substime = c(New_list[["numOfSubs"]][1])
for(i in 1:numofStus){
  if(New_list[["numOfSubs"]][i] != New_list[["numOfSubs"]][i+1]){
    substime = c(substime, New_list[["numOfSubs"]][i+1])
  }
}
#Tim so lan nop bai nhieu thu nhì
second = substime[length(substime)-1]
#Lap danh sach sinh vien nop nhieu nhat
second_stus = c()
second_stus = subset(New_list, New_list$numOfSubs == second)
View(second_stus)
```

Output của file 6:

IDlist	numOfSubs	
301	1911110	3
302	1911561	3
303	1911881	3
304	1912056	3
305	1912916	3
306	1913418	3
307	1913609	3
308	1913775	3
309	1913844	3
310	1913918	3

Output của file 9:

IDlist	numOfSubs	
292	1910620	3
293	1814096	3
294	1910265	3
295	1910644	3
296	1911262	3
297	1911736	3
298	1911907	3
299	1912749	3
300	1912761	3
301	1913045	3
302	1913949	3
303	1914227	3
304	1914316	3
305	1915350	3

Output của file 10:

IDlist	numOfSubs	
287	1911907	5

Output của file 12:

IDlist	numOfSubs	
211	1912041	2
212	1812477	2
213	1852443	2
214	1910123	2
215	1910409	2
216	1910892	2
217	1911066	2
218	1911262	2
219	1911363	2
220	1911441	2
221	1912123	2
222	1912288	2
223	1912371	2
224	1912410	2
225	1912457	2
226	1912594	2
227	1912602	2
228	1912676	2
229	1912713	2
230	1912761	2
231	1912798	2
232	1912814	2

- o) Xác định danh sách các sinh viên nằm trong nhóm có số lần nộp bài nhiều nhất hoặc nhiều nhì
Tương tự như cách tìm danh sách sinh viên nộp nhiều nhì, ta tìm danh sách sinh viên nộp nhiều nhất rồi nối hai danh sách với nhau bằng hàm *rbind*.

```
first = substime[length(substime)]
first_stus = c()
first_stus = subset(New_list, New_list$numOfSubs == first)
firstandsecond_stus = c()
firstandsecond_stus = rbind(second_stus, first_stus)
View(firstandsecond_stus)
```

Output của file 6:

IDlist	numOfSubs	
304	1912056	3
305	1912916	3
306	1913418	3
307	1913609	3
308	1913775	3
309	1913844	3
310	1913918	3
311	1913949	3
312	1913990	3
313	1914052	3
314	1914210	3
315	1914802	3
316	1914979	3
317	1915268	3
318	1915570	3
319	1915775	3
320	1915983	3
321	1927007	3
322	1937019	4

Output của file 9:

	IDlist	numOfSubs
292	1910620	3
293	1814096	3
294	1910265	3
295	1910644	3
296	1911262	3
297	1911736	3
298	1911907	3
299	1912749	3
300	1912761	3
301	1913045	3
302	1913949	3
303	1914227	3
304	1914316	3
305	1915350	3
306	1914210	5

Output của file 10:

	IDlist	numOfSubs
287	1911907	5
288	1913021	6
289	1913167	6

Output của file 12:

IDlist	numOfSubs	
211	1912041	2
212	1812477	2
213	1852443	2
214	1910123	2
215	1910409	2
216	1910892	2
217	1911066	2
218	1911262	2
219	1911363	2
220	1911441	2
221	1912123	2
222	1912288	2
223	1912371	2
224	1912410	2
225	1912457	2
226	1912594	2
227	1912602	2
228	1912676	2
229	1912713	2

- p) Xác định số lượng sinh viên nằm trong nhóm có số lần nộp bài nhiều nhất hoặc nhiều nhì
Khi đã có danh sách, ta tìm độ dài danh sách sẽ tìm được số sinh viên.

```
numof_firstandsecond = nrow(firstandsecond_stus)  
numof_firstandsecond
```

Output của file 6:

```
[1] 22
```

Output của file 9:

```
[1] 15
```

Output của file 10:

```
> numof_firstandsecond  
[1] 3  
>
```

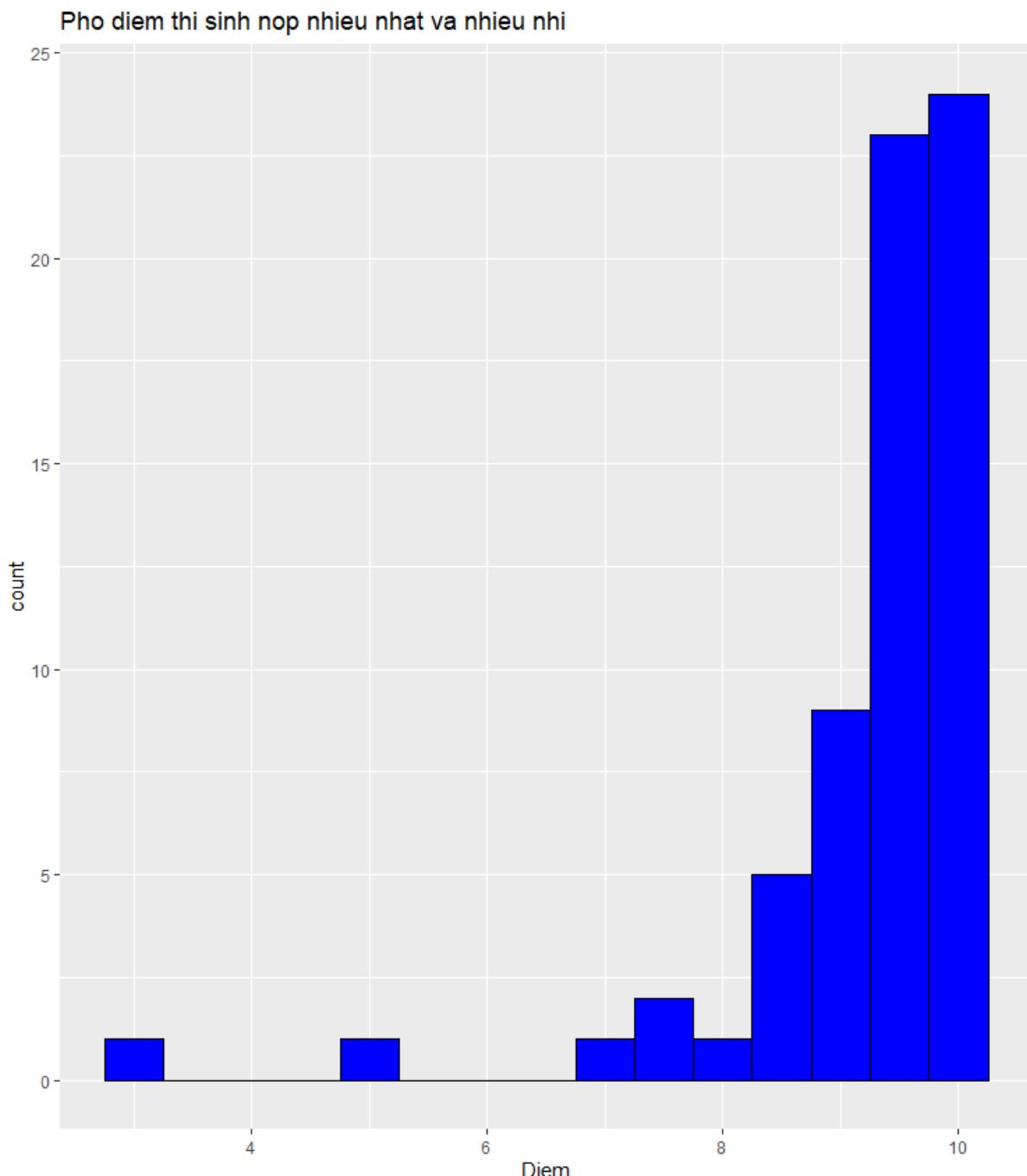
Output của file 12:

```
> numof_firstandsecond
[1] 70
>
```

- q) Xác định phổ theo điểm số của các sinh viên có lần nộp bài nhiều nhất hoặc nhiều nhì
Ta làm tương tự câu 3c

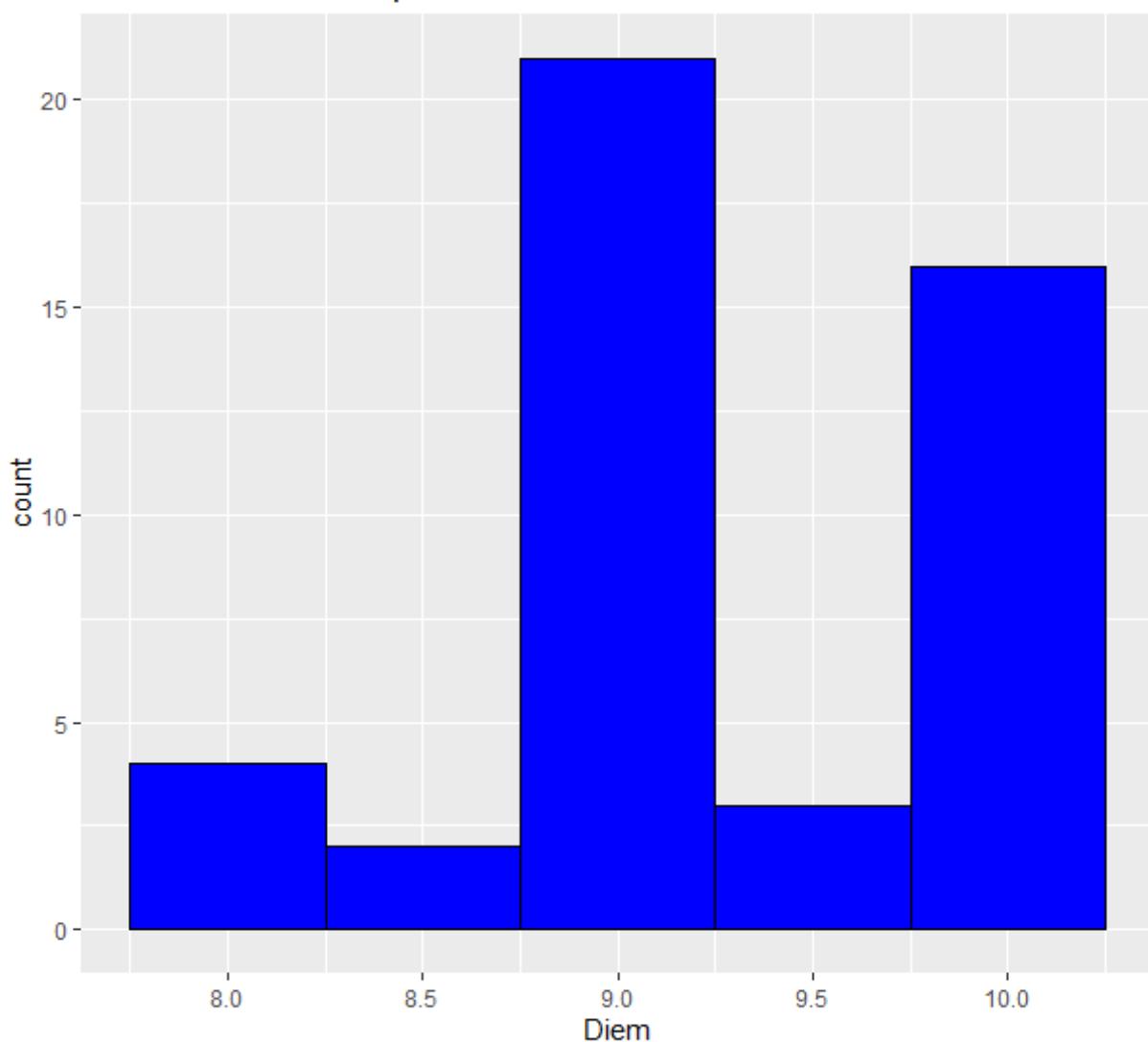
```
New_list_secfi = c()
New_list_secfi = subset(New_list, num0fSubs == second | num0fSubs == first)
min_spec5 = data.frame()
for (i in 1:nrow(New_list_secfi))
{
  for (j in 1:nrow(mydata))
  {
    if (mydata[[1]][j] == New_list_secfi[[1]][i])
    {
      add5 = c(mydata[["Diem"]][j])
      min_spec5 = rbind(min_spec5, add5)
    }
  }
}
colnames(min_spec5) = c("Diem")
ggplot(min_spec5, aes(Diem))+geom_histogram(binwidth = 0.5, fill ="blue", col =
"black")+ ggtitle("Pho diem thi sinh nop nhieu nhat va nhieu nhi")
```

Output của file 6:



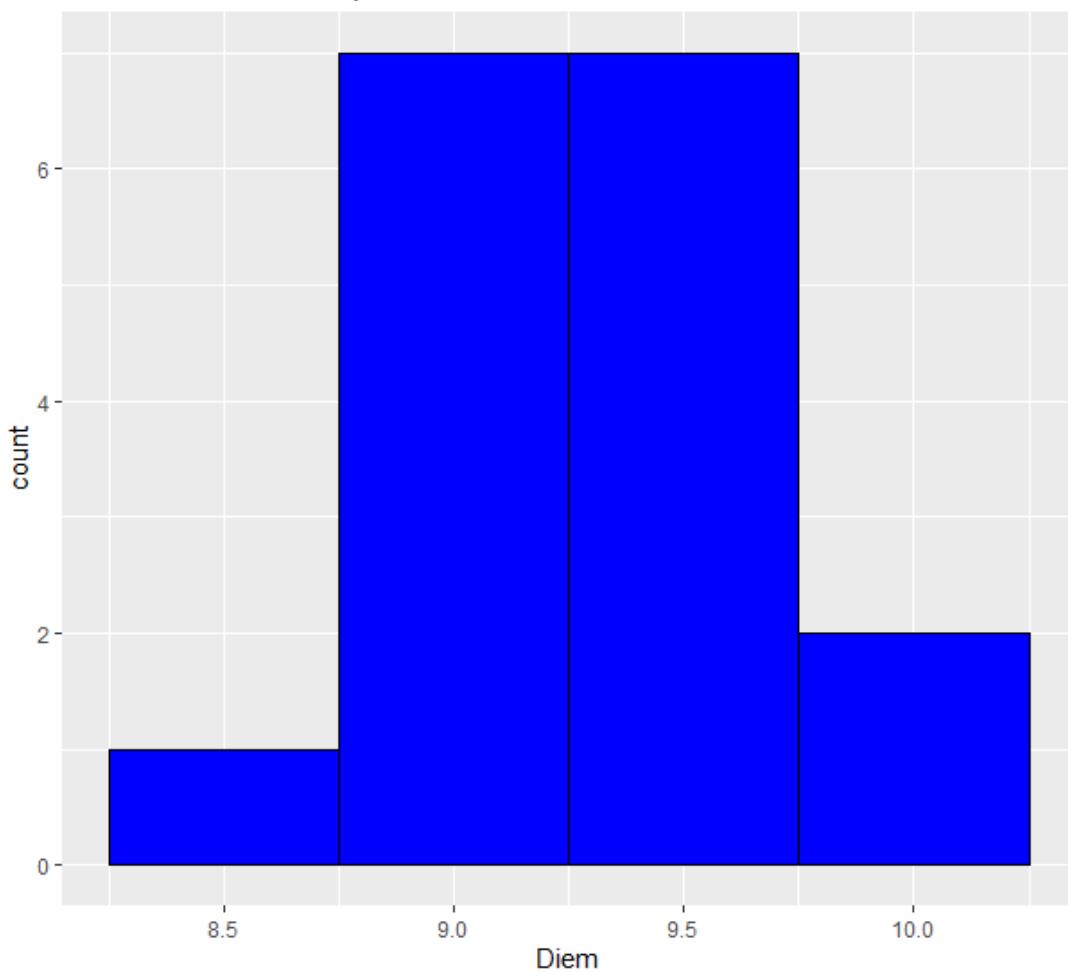
Output của file 9:

Phân佈 điểm thi sinh nộp nhiều nhất và nhiều nhì



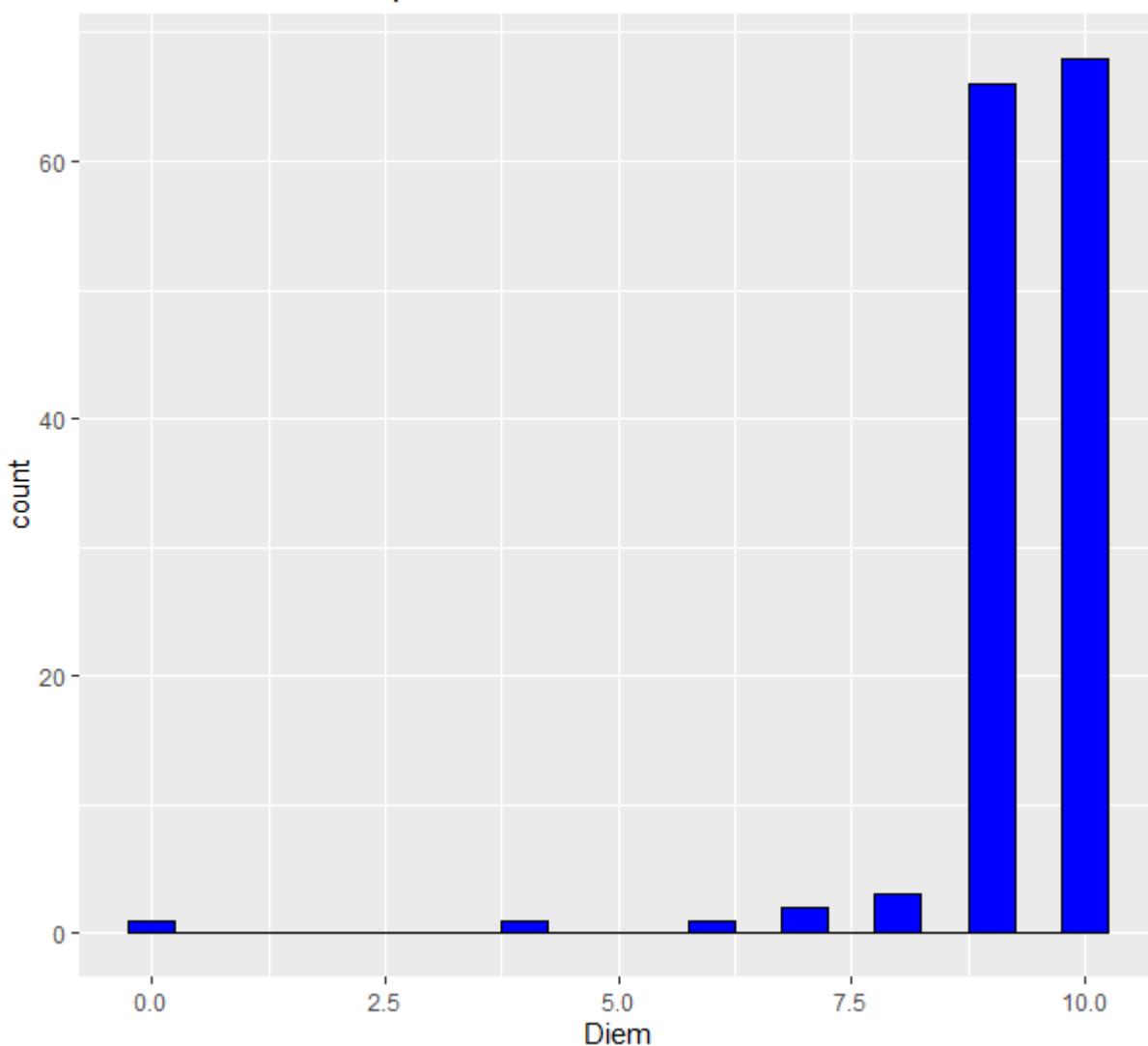
Output của file 10:

Phân佈 điểm thi sinh nộp nhiều nhất và nhiều nh



Output của file 12:

Phân佈 điểm thi sinh nộp nhiều nhất và nhiều nhất



- r) Xác định danh sách các sinh viên nằm trong nhóm một phần ba đầu theo thứ tự số lần nộp bài giảm dần
Dầu tiên ta sắp xếp danh sách *New_list* theo thứ tự giảm dần rồi lập một vecto các số ID của các bạn sinh viên thuộc một phần ba đầu.

```
New_list_dec = c() New_list_dec = arrange(New_list, desc(numOfSubs))
onethree_stus = c(New_list_dec[["IDlist"]][1])
tam = 2
while(tam <= numofStus/3){
  onethree_stus <- c(onethree_stus, New_list_dec[["IDlist"]][tam])
  tam = tam+1
}
View(onethree_stus)
```

Output của file 6:

V1
1 1937019
2 1911110
3 1911561
4 1911881
5 1912056
6 1912916
7 1913418
8 1913609
9 1913775
10 1913844
11 1913918
12 1913949
13 1913990
14 1914052
15 1914210
16 1914802
17 1914979
18 1915268
19 1915570
20 1915775

Output của file 9:

V1
1 1914210
2 1910620
3 1814096
4 1910265
5 1910644
6 1911262
7 1911736
8 1911907
9 1912749
10 1912761
11 1913045
12 1913949
13 1914227
14 1914316
15 1915350

Output của file 10:

V1
1 1913021
2 1913167
3 1911907
4 1910238
5 1910666
6 1912457
7 1913396
8 1913918
9 1914126
10 1914352
11 1914405
12 1914738
13 1915470
14 1915882
15 1912980
16 1613010
17 1910006
18 1910060
19 1910101
20 1910110
21 1910265
22 1910220

Output của file 12:

V1
1 1913045
2 1915520
3 1927007
4 1912041
5 1812477
6 1852443
7 1910123
8 1910409
9 1910892
10 1911066
11 1911262
12 1911363
13 1911441
14 1912123
15 1912288
16 1912371
17 1912410
18 1912457
19 1912594
20 1912602
21 1912676

- s) Xác định số lượng các sinh viên nằm trong nhóm một phần ba đầu theo thứ tự số lần nộp bài giảm dần
Tương tự các câu trên, ta dùng hàm *length()* để tìm độ dài của danh sách.

```
onethree_count = length(onethree_stus)  
onethree_count
```

Output của file 6:

```
> onethree_count  
[1] 107  
> |
```

Output của file 9:

```
| [1] 107
```

Output của file 10:

```
> onethree_count  
[1] 107  
> |
```

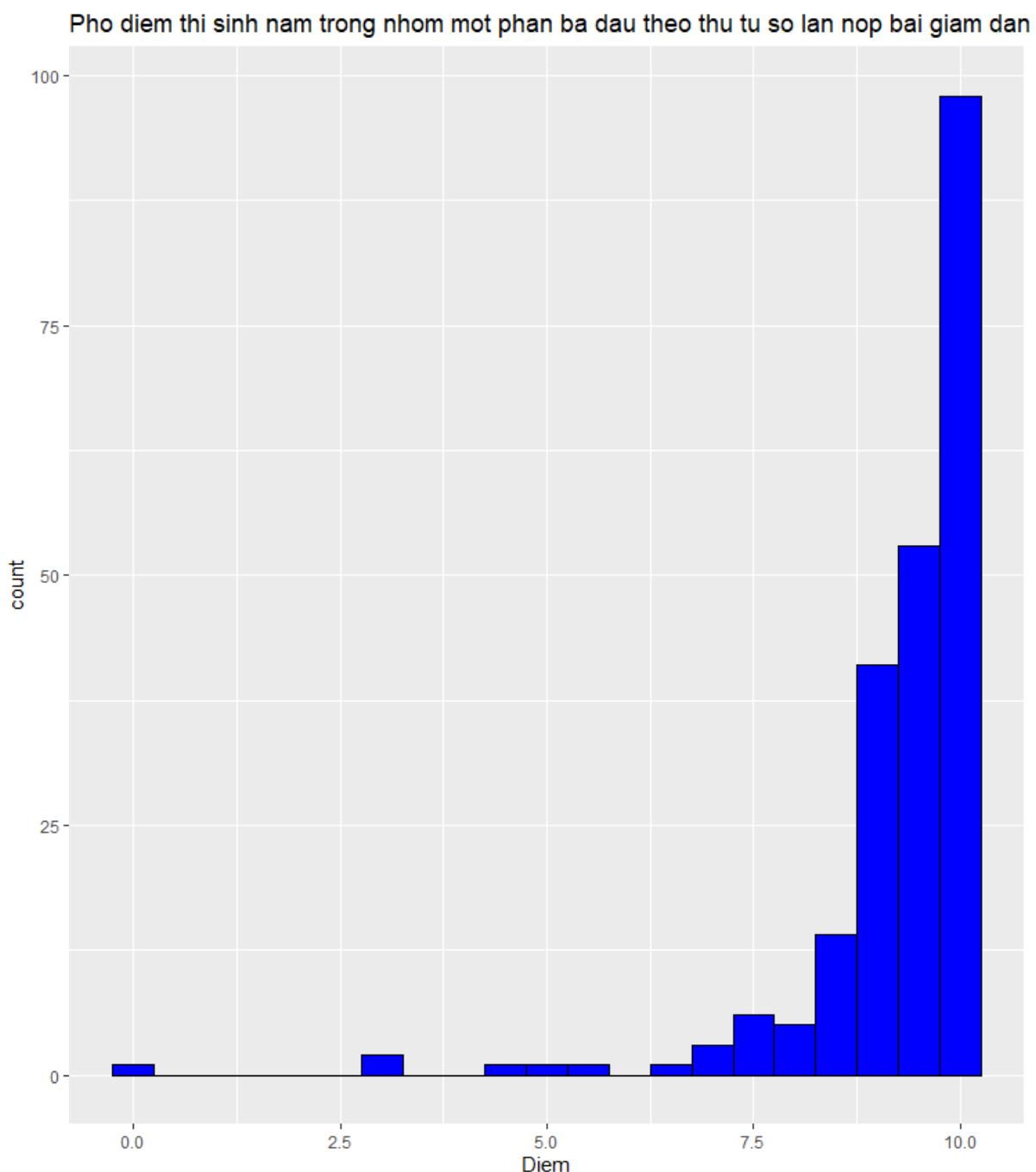
Output của file 12:

```
> onethree_count  
[1] 107  
> |
```

- t) Xác định phổ theo điểm số của các sinh viên nằm trong nhóm một phần ba đầu theo thứ tự số lần nộp bài giảm dần
Ta làm tương tự câu 3c

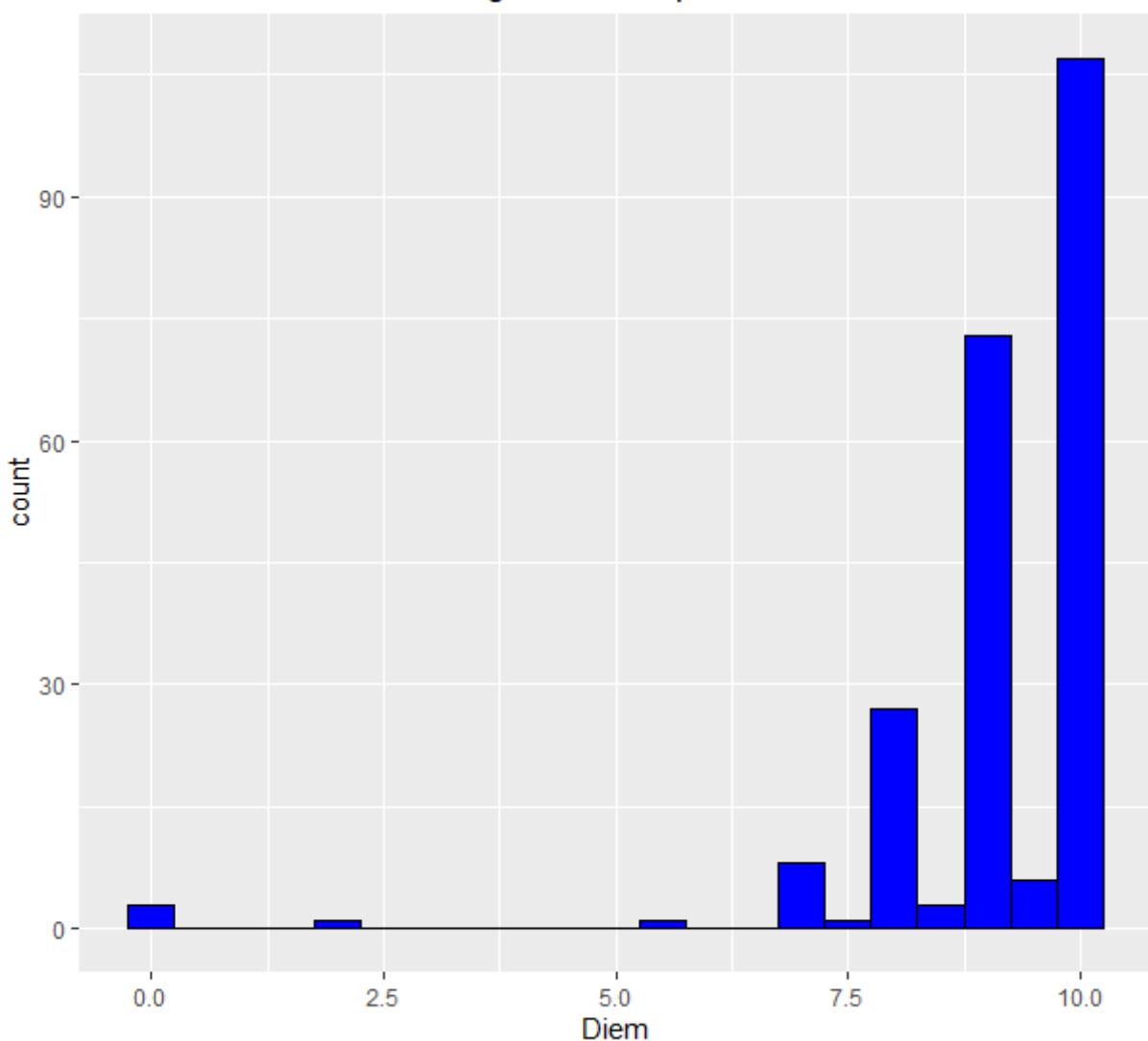
```
min_spec6 = data.frame()
for (i in 1:length(onethree_stus))
{
  for (j in 1:nrow(mydata))
  {
    if (mydata[[1]][j] == onethree_stus[i])
    {
      add6 = c(mydata[["Diem"]][j])
      min_spec6 = rbind(min_spec6, add6)
    }
  }
}
colnames(min_spec6) = c("Diem")
ggplot(min_spec6, aes(Diem))+geom_histogram(binwidth = 0.5, fill ="blue", col =
"black")+ ggtitle("Pho diem thi sinh nam trong nhom mot phan ba dau theo thu tu
so lan nop bai giam dan")
```

Output của file 6:

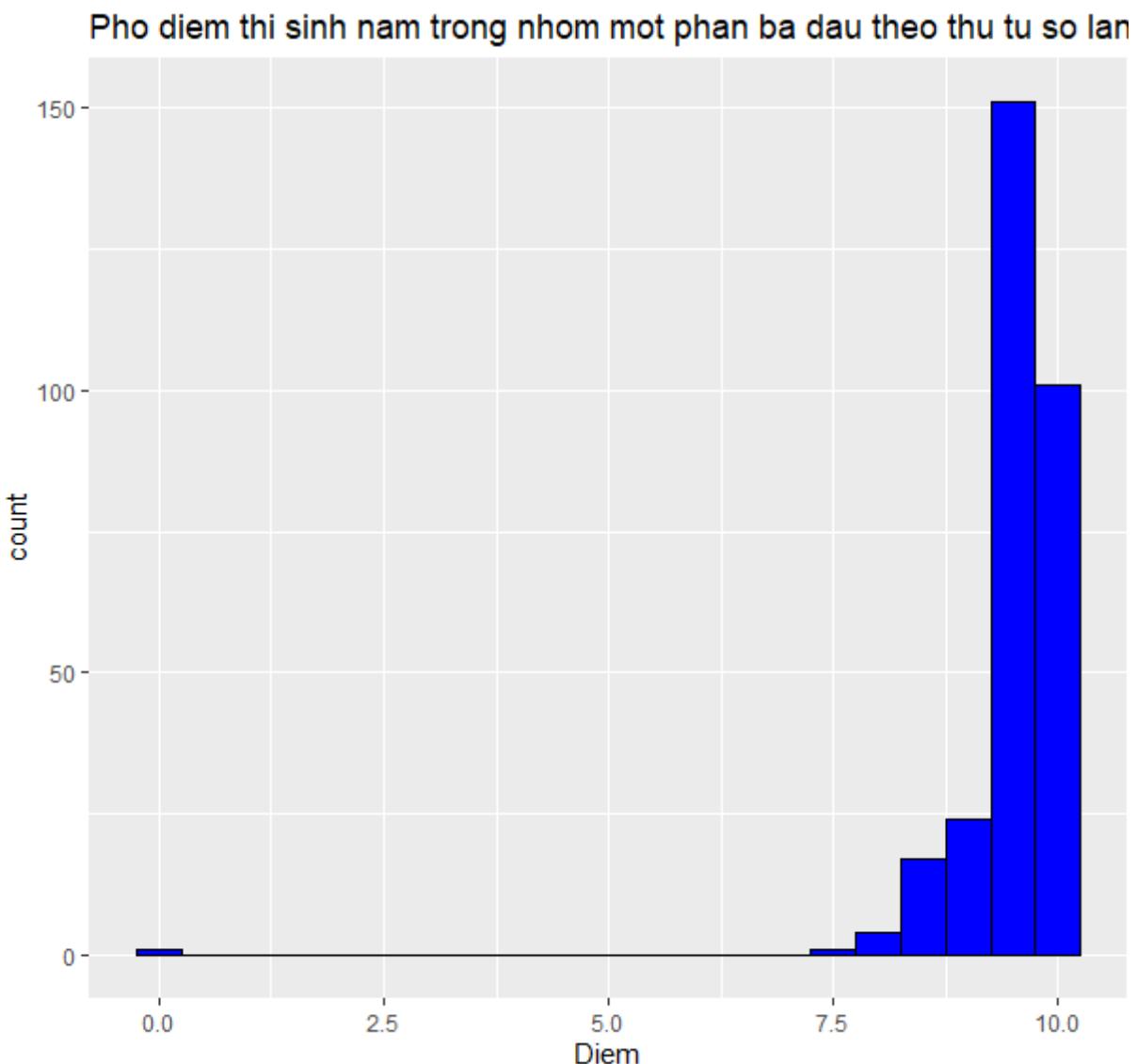


Output của file 9:

Phân佈 điểm thi sinh nam trong nhóm một phần ba đầu theo thứ tự lần

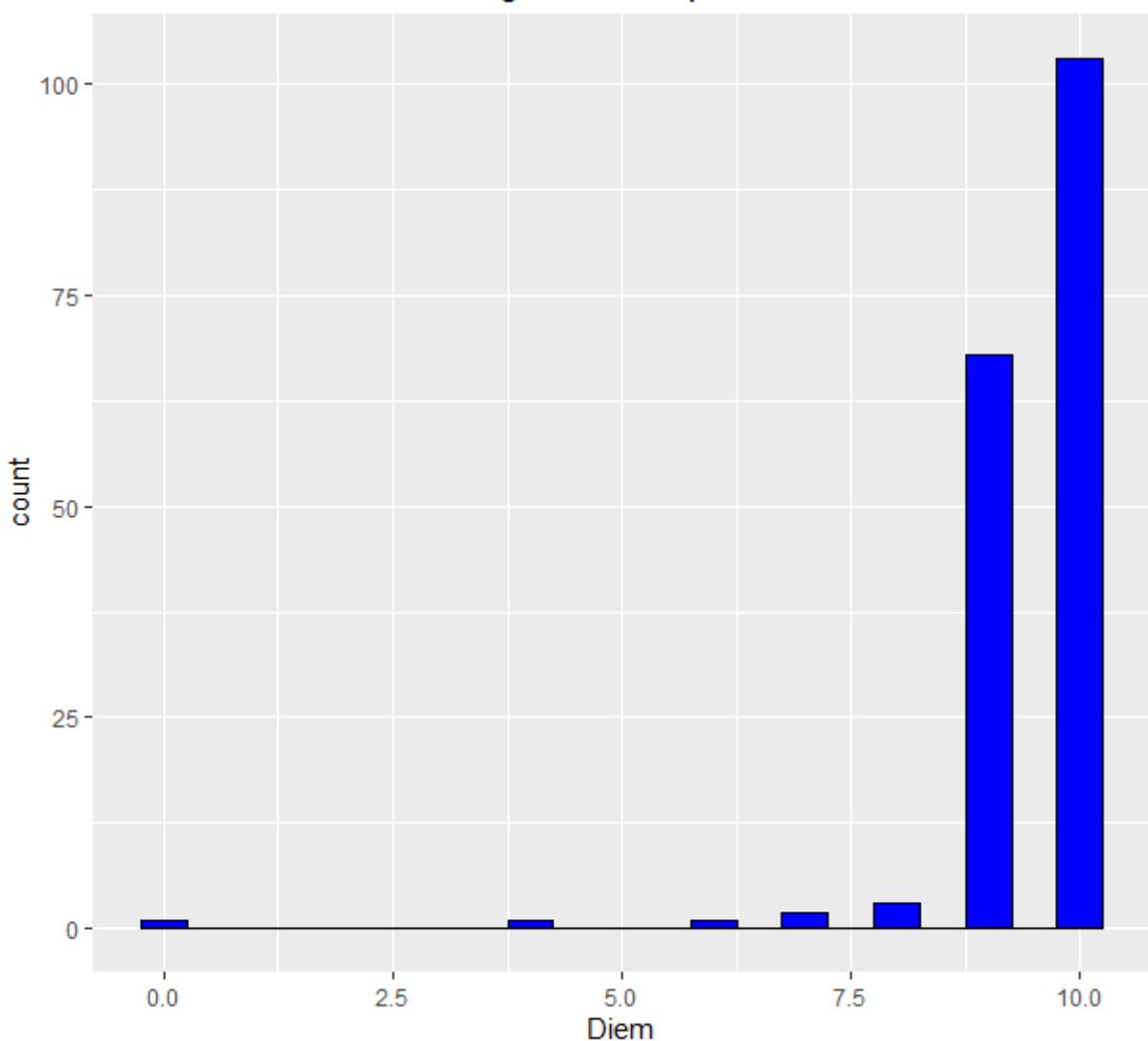


Output của file 10:



Output của file 12:

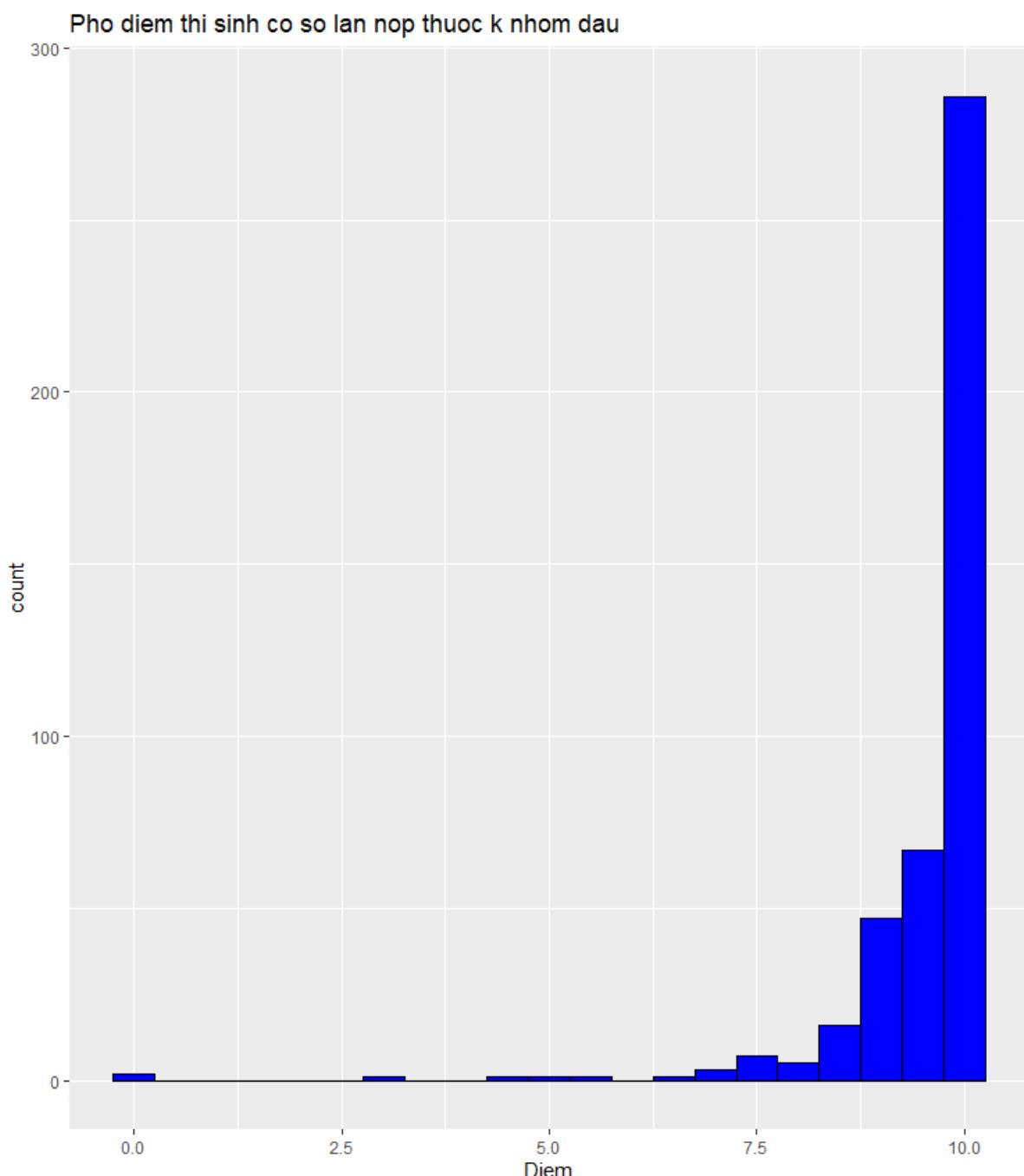
Phân佈 điểm thi sinh nam trong nhóm một phần ba bài đầu theo thứ tự lần



- u) Xác định phổ theo điểm số của các sinh viên nằm trong k nhóm đầu mà mỗi nhóm chứa các sinh viên có cùng số lần nộp bài và các nhóm được sắp xếp theo thứ tự giảm dần của có số lần nộp bài (với k cho trước). Đầu tiên, ta cho người dùng nhập k từ bàn phím bằng hàm `scan()`. Ta lập danh sách các sinh viên với số lần nộp bài giảm dần với điểm của từng sinh viên tương ứng. Sau đó ta thực hiện vẽ biểu đồ tương tự câu 3c

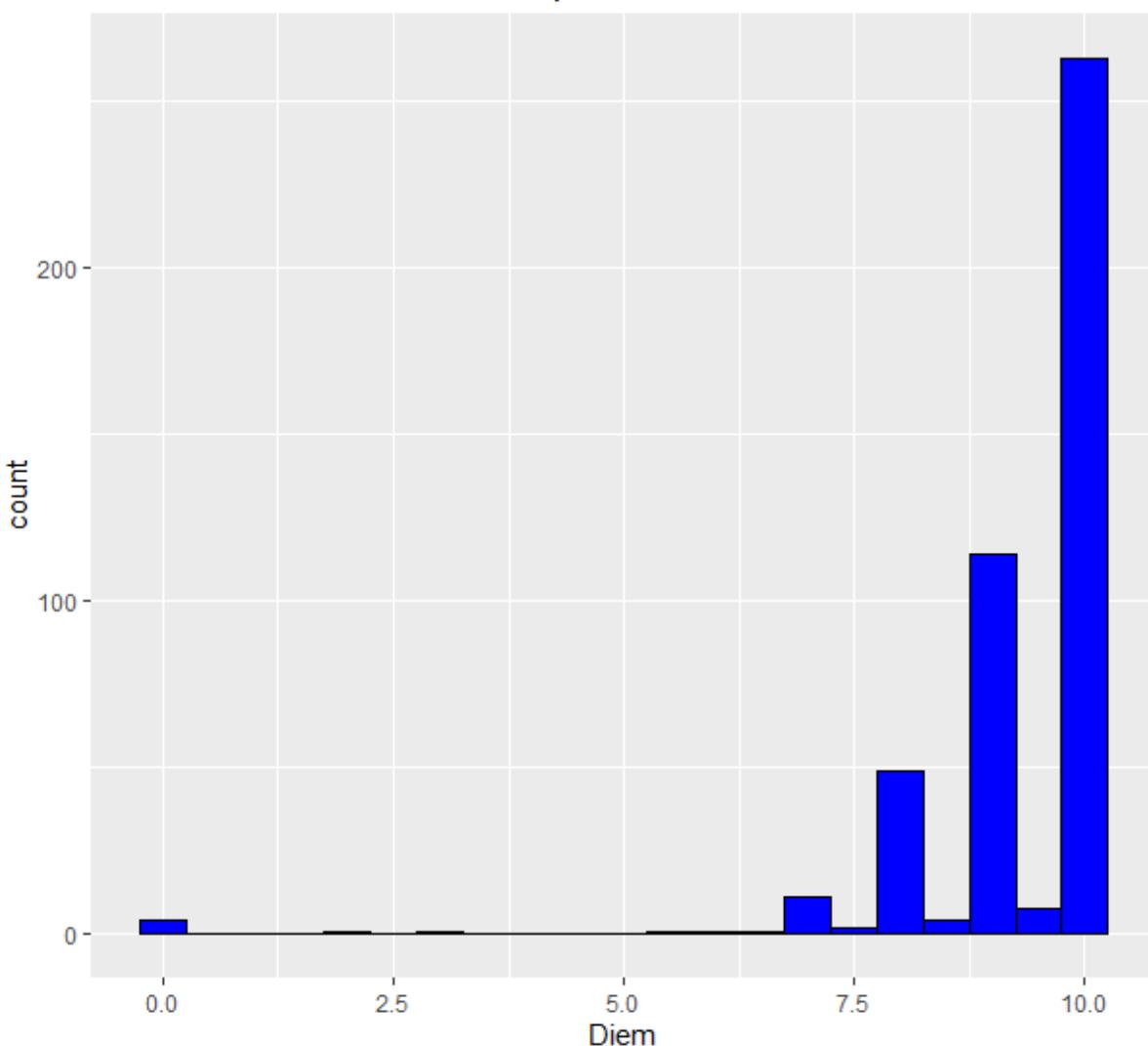
```
k = scan()
substime = sort(substime, decreasing = TRUE)
New_list_k = c()
for (i in 1:k){
  for (j in 1:nrow(New_list)){
    if (New_list[[2]][j]==substime[i]){
      add_k = c(New_list[["IDlist"]][j])
      New_list_k = c(New_list_k, add_k)
    }}}
min_spec7 = data.frame()
for (i in 1:length(New_list_k)){
  for (j in 1:nrow(mydata)){
    if (mydata[[1]][j] == New_list_k[i]){
      add7 = c(mydata[["Diem"]][j])
      min_spec7 = rbind(min_spec7, add7)
    }}}
colnames(min_spec7) = c("Diem")
ggplot(min_spec7, aes(Diem))+geom_histogram(binwidth = 0.5, fill ="blue", col =
"black")+ ggtitle("Pho diem thi sinh co so lan nop thuoc k nhom dau")
```

Output của file 6:



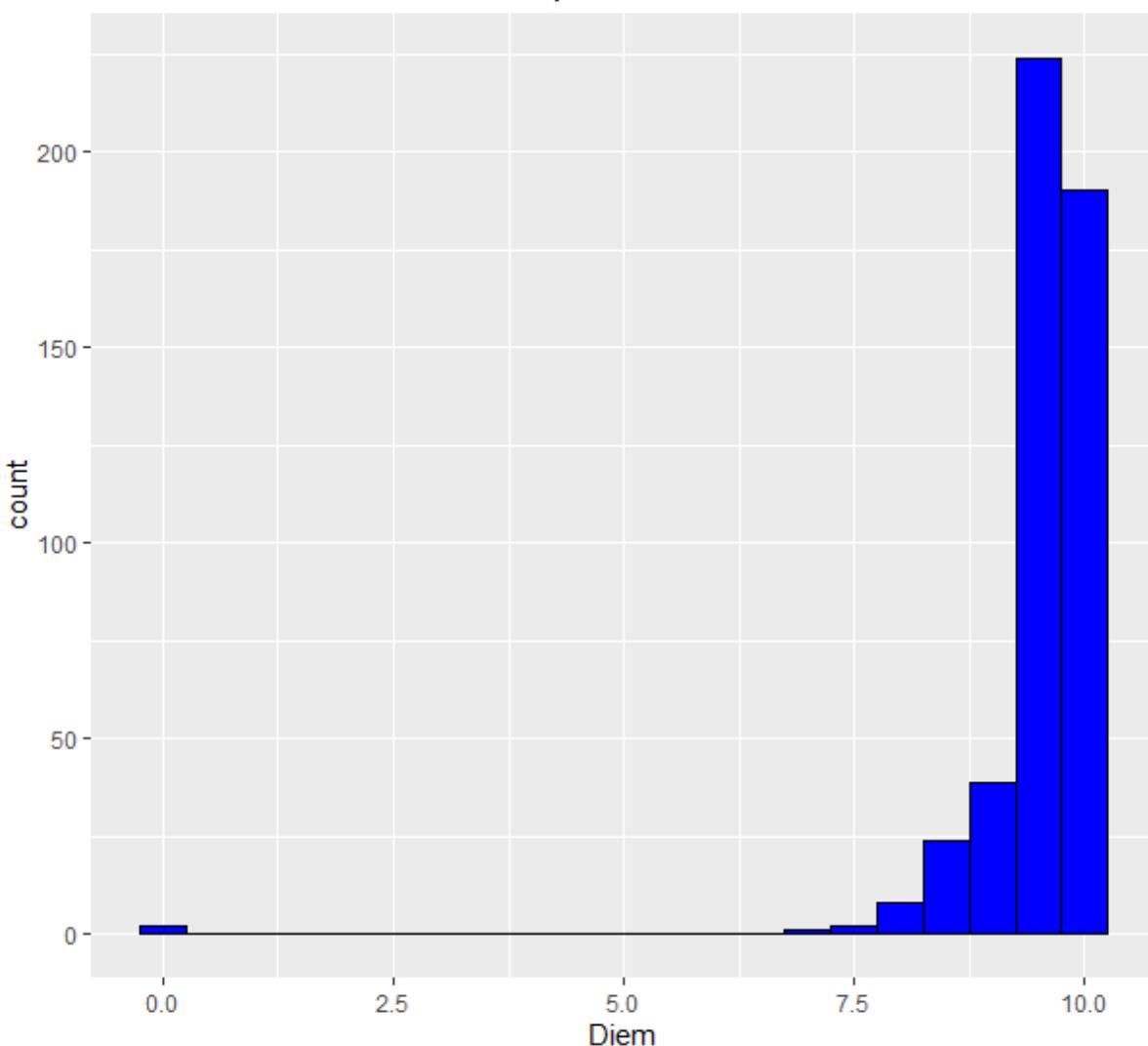
Output của file 9:

Phân佈 điểm thi sinh có số lần nộp thuộc k nhom dau



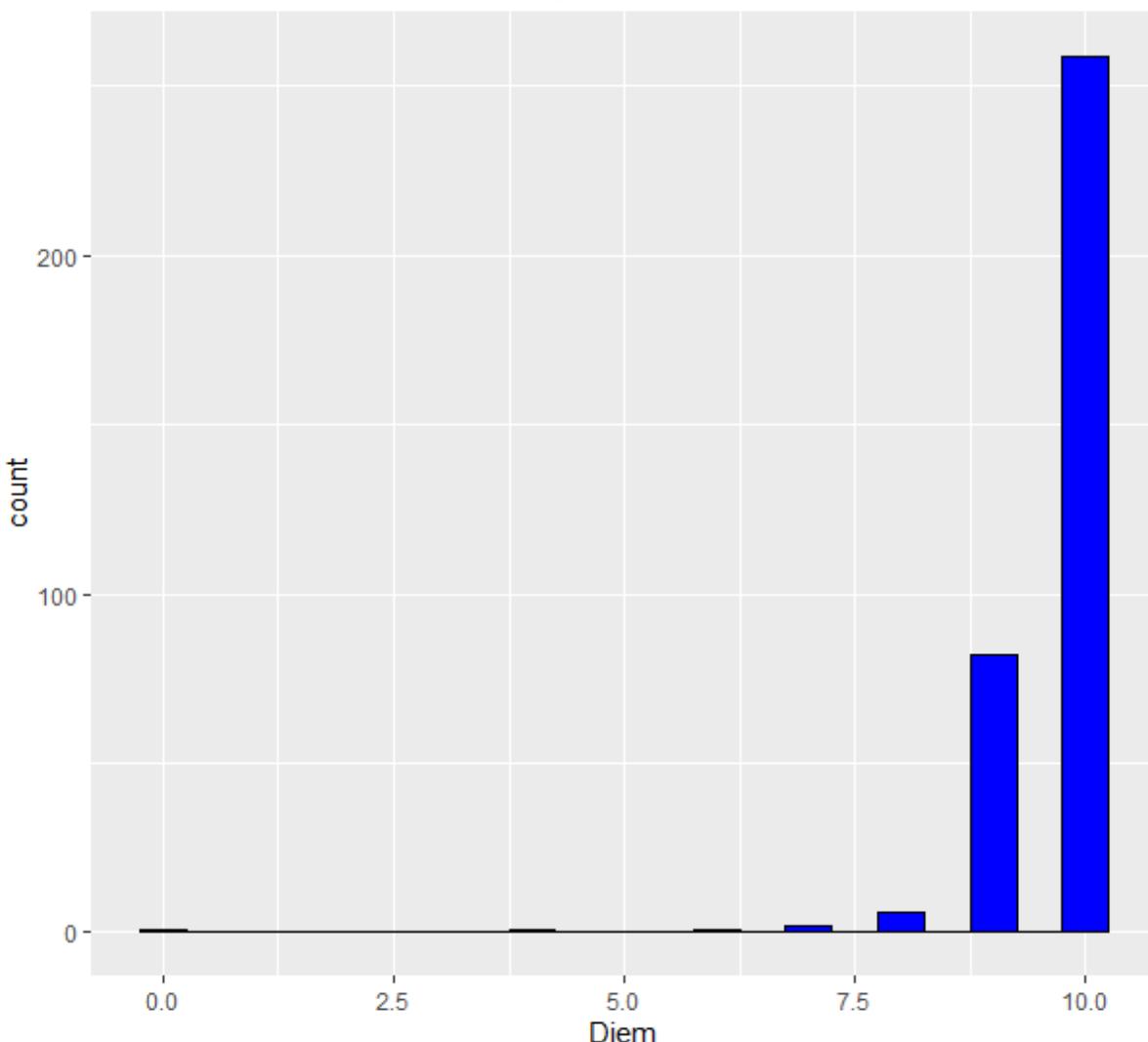
Output của file 10:

Phân điểm thi sinh có số lần nộp thuộc k nhóm đầu



Output của file 12:

Phân佈 điểm thi sinh có số lần nộp thuộc k nhóm đầu



4.4 Câu 4: Nhóm câu hỏi liên quan đến thời gian, tần suất nộp bài của các sinh viên

- a) Với mỗi sinh viên, xác định thời gian dài nhất tính từ lần nộp bài đầu tiên đến lần nộp cuối.
Trước hết, tạo vector `time1` và `time2` để lưu giữ thời gian nộp bài lần đầu và lần cuối của mỗi sinh viên. Ta tiếp tục dùng vòng lặp đối với data frame `sapxep_ID` để tìm các giá trị này. Sau đó chúng ta sẽ chuyển đổi dạng của 2 vector này sang dạng `POSIXct` bằng hàm `dmy_hm()`. Cuối cùng ta sẽ tạo một data frame mới `New_list_4` để các dữ liệu thời gian có thể tương ứng với ID của các sinh viên.

```
time1=c() # Vector thoi gian nopol lan dau
time2=c() # Vector thoi gian lan cuoi
time1<-c(time1, sapxep_ID[["Etime"]][1])
for( i in 1:(n-2)){
if(sapxep_ID[["ID"]][i]!=sapxep_ID[["ID"]][i+1]){
time1 <- c(time1, sapxep_ID[["Etime"]][i+1])
time2 <- c(time2, sapxep_ID[["Etime"]][i])
if(i==n-2) {
time2 <- c(time2, sapxep_ID[["Etime"]][i+1])
}}
else{
if(i==n-2) {
time2 <- c(time2, sapxep_ID[["Etime"]][i+1])
}}}
# chuyen doi dang thoi gian cua time1 va time2
time1 <- dmy_hm(time1)
time2 <- dmy_hm(time2)
T=c() # Vector tinh thoi gian chenh lech tu lan nopol dau va lan nopol cuoi
# Ham difftime thuoc thu vien "lubridate", do chenh lech tinh bang giay
T=difftime(time2, time1, units="secs")
New_list_4<-data.frame(IDlist, numOfSubs, time1, time2, T)
View(New_list_4)
```

Ta có thể thấy thời gian nộp lần đầu và lần cuối cũng như độ chênh lệch tính bằng giây của chúng của các sinh viên tương ứng với mã số ID như sau (trích một vài sinh viên đầu danh sách):

Output của file 6:

	IDlist	numOfSubs	time1	time2	T
1	1511191	1	2020-04-04 15:19:00	2020-04-04 15:19:00	0
2	1613010	2	2020-04-19 23:32:00	2020-04-19 23:33:00	60
3	1812257	1	2020-04-20 13:19:00	2020-04-20 13:19:00	0
4	1812477	2	2020-04-23 07:44:00	2020-04-23 07:46:00	120
5	1812478	1	2020-04-06 16:59:00	2020-04-06 16:59:00	0
6	1813096	2	2020-04-12 15:17:00	2020-04-12 15:19:00	120

Output của file 9:

	IDlist	numOfSubs	time1	time2	T
1	1511191	3	2020-04-04 20:16:00	2020-04-04 20:19:00	180
2	1613010	2	2020-04-19 23:38:00	2020-04-19 23:40:00	120
3	1812257	1	2020-04-20 13:52:00	2020-04-20 13:52:00	0
4	1812477	2	2020-04-24 14:17:00	2020-04-24 14:19:00	120
5	1812478	1	2020-04-06 17:41:00	2020-04-06 17:41:00	0
6	1813096	1	2020-04-13 10:08:00	2020-04-13 10:08:00	0

Output của file 10:

	IDlist	numOfSubs	time1	time2	T
1	1511191	3	2020-04-20 13:06:00	2020-04-20 13:08:00	120
2	1613010	3	2020-04-27 17:09:00	2020-04-27 17:11:00	120
3	1812257	1	2020-04-20 17:32:00	2020-04-20 17:32:00	0
4	1812477	2	2020-04-24 18:51:00	2020-04-24 18:54:00	180
5	1812478	1	2020-04-06 17:43:00	2020-04-06 17:43:00	0
6	1813096	2	2020-05-06 09:50:00	2020-05-06 09:51:00	60

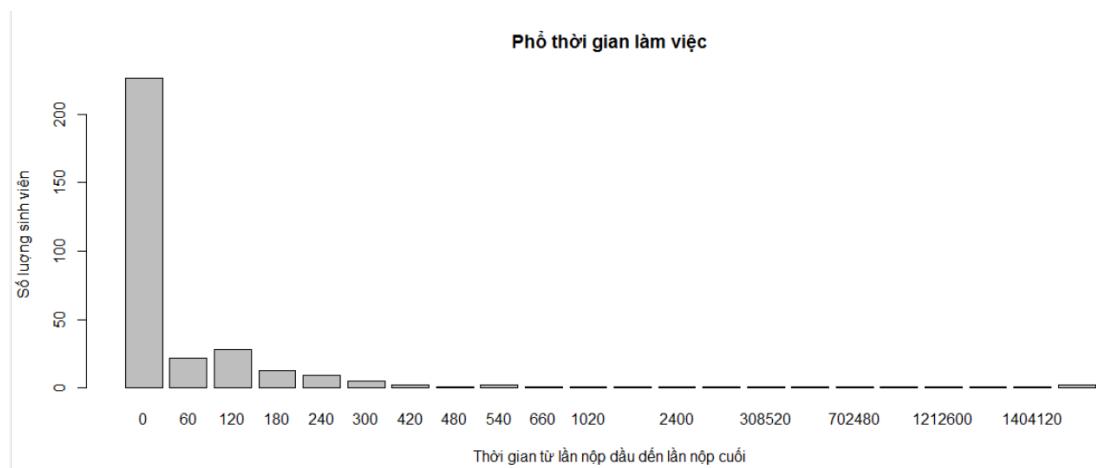
Output của file 12:

	IDlist	numOfSubs	time1	time2	T
1	1511191	2	2020-04-20 13:25:00	2020-04-20 13:26:00	60
2	1613010	1	2020-04-27 17:18:00	2020-04-27 17:18:00	0
3	1812257	1	2020-04-20 17:45:00	2020-04-20 17:45:00	0
4	1812477	2	2020-04-24 20:05:00	2020-04-25 15:53:00	71280
5	1812478	1	2020-04-13 12:33:00	2020-04-13 12:33:00	0
6	1813096	1	2020-05-06 11:38:00	2020-05-06 11:38:00	0

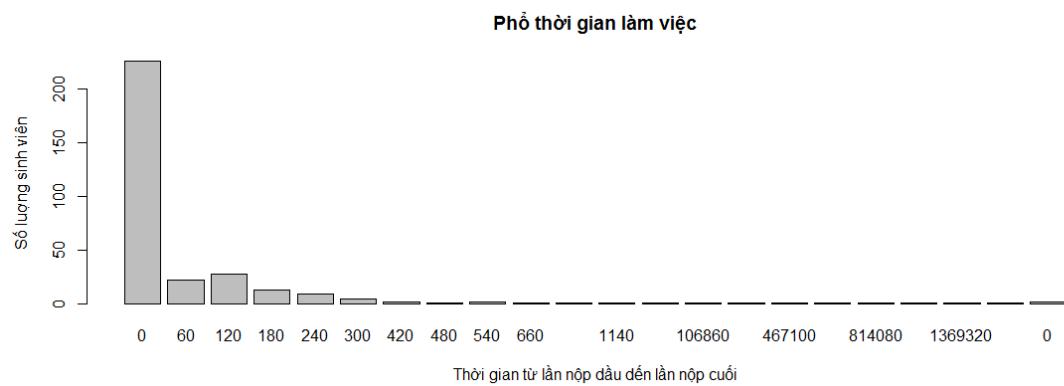
- b) Xác định phổ thời gian làm việc (được tính từ lần nộp bài đầu tiên đến lần nộp cuối) của các sinh viên.
Ta sẽ vẽ phổ điểm dựa vào số lượng sinh viên có khoảng thời gian chênh lệch từ lần nộp đầu đến lần nộp cuối theo nhiều mức khác nhau.

```
df_4b=arrange(New_list_4, T)
df_4b$T[is.na(df_4b$T)] <- 0
T_4b=c()
t_4b=1
numT_4b=c()
T_4b <- c(T_4b, df_4b$T[1])
for(i in 1:(nrow(New_list_4)-1)){
if(df_4b[["T"]][i] != df_4b[["T"]][i+1]){
T_4b <- c(T_4b, df_4b[["T"]][i+1])
numT_4b <- c(numT_4b, t_4b)
t_4b=1
if(i==(nrow(New_list_4)-1)) {
numT_4b <- c(numT_4b, t_4b)
}}
else{
t_4b=t_4b+1
if(i==(nrow(New_list_4)-1)) {
numT_4b <- c(numT_4b, t_4b)
}}}
barplot(numT_4b, T_4b, width=1, space=0.2,
xlab="Thời gian từ lần nộp đầu đến lần nộp cuối",
ylab="Số lượng sinh viên",
main="Phổ thời gian làm việc")
```

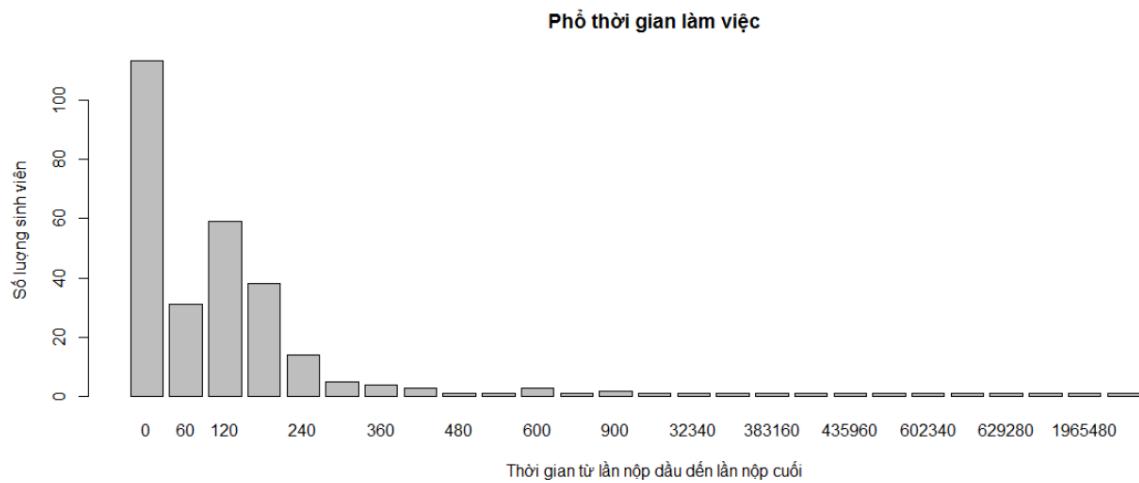
Output của file 6:



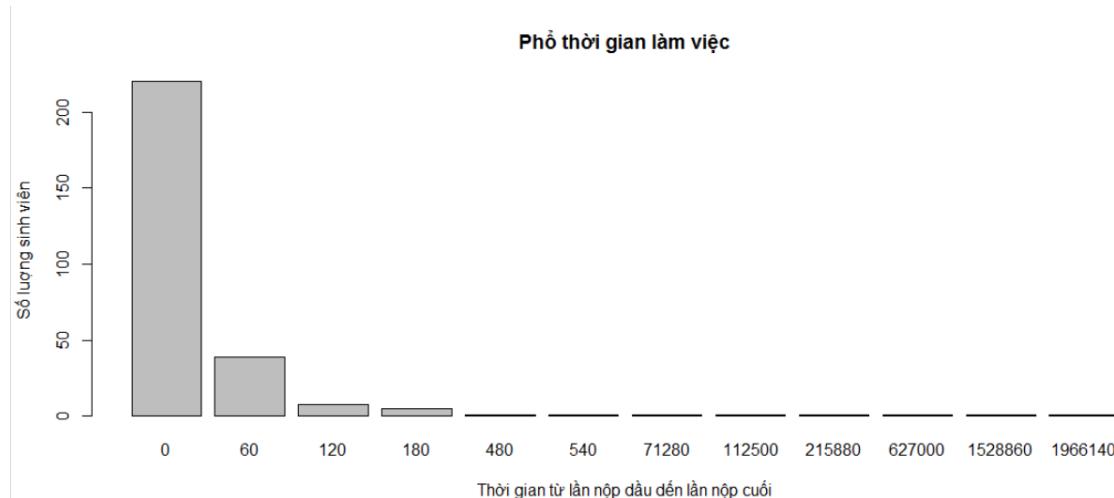
Output của file 9:



Output của file 10:



Output của file 12:



- c) Tần suất nộp bài được tính bằng phân số giữa khoảng thời gian tính từ lần nộp bài đầu tiên đến lần nộp cuối và số lần nộp bài.

Tạo vector *fre* lưu tần suất nộp bài của các sinh viên. Tần suất nộp bài *fre=as.numeric(T)/numOfSubs*. Lưu ý tử số phải ở dạng numeric.

```
fre=c() # Vector tần suất
fre=as.numeric(T)/numOfSubs
New_list_4=cbind(New_list_4, fre)
View(New_list_4)
```

Output của file 6 (trích một vài sinh viên đầu danh sách):

IDlist	numOfSubs	time1	time2	T	fre
1 1511191	1	2020-04-04 15:19:00	2020-04-04 15:19:00	0	0
2 1613010	2	2020-04-19 23:32:00	2020-04-19 23:33:00	60	30
3 1812257	1	2020-04-20 13:19:00	2020-04-20 13:19:00	0	0
4 1812477	2	2020-04-23 07:44:00	2020-04-23 07:46:00	120	60
5 1812478	1	2020-04-06 16:59:00	2020-04-06 16:59:00	0	0
6 1813096	2	2020-04-12 15:17:00	2020-04-12 15:19:00	120	60

Output của file 9 (trích một vài sinh viên đầu danh sách):

IDlist	numOfSubs	time1	time2	T	fre
1 1511191	3	2020-04-04 20:16:00	2020-04-04 20:19:00	180	60
2 1613010	2	2020-04-19 23:38:00	2020-04-19 23:40:00	120	60
3 1812257	1	2020-04-20 13:52:00	2020-04-20 13:52:00	0	0
4 1812477	2	2020-04-24 14:17:00	2020-04-24 14:19:00	120	60
5 1812478	1	2020-04-06 17:41:00	2020-04-06 17:41:00	0	0
6 1813096	1	2020-04-13 10:08:00	2020-04-13 10:08:00	0	0

Output của file 10 (trích một vài sinh viên đầu danh sách):

	IDlist	numOfSubs	time1	time2	T	fre
1	1511191	3	2020-04-20 13:06:00	2020-04-20 13:08:00	120	40
2	1613010	3	2020-04-27 17:09:00	2020-04-27 17:11:00	120	40
3	1812257	1	2020-04-20 17:32:00	2020-04-20 17:32:00	0	0
4	1812477	2	2020-04-24 18:51:00	2020-04-24 18:54:00	180	90
5	1812478	1	2020-04-06 17:43:00	2020-04-06 17:43:00	0	0
6	1813096	2	2020-05-06 09:50:00	2020-05-06 09:51:00	60	30

Output của file 12 (trích một vài sinh viên đầu danh sách):

	IDlist	numOfSubs	time1	time2	T	fre
1	1511191	2	2020-04-20 13:25:00	2020-04-20 13:26:00	60	30
2	1613010	1	2020-04-27 17:18:00	2020-04-27 17:18:00	0	0
3	1812257	1	2020-04-20 17:45:00	2020-04-20 17:45:00	0	0
4	1812477	2	2020-04-24 20:05:00	2020-04-25 15:53:00	71280	35640
5	1812478	1	2020-04-13 12:33:00	2020-04-13 12:33:00	0	0
6	1813096	1	2020-05-06 11:38:00	2020-05-06 11:38:00	0	0

d) Xác định danh sách các sinh viên có tần suất nộp bài ít nhất

Sắp xếp dataframe *New_list_4* theo thứ tự *fre* tăng dần, phần tử *fre* đầu tiên lúc này chính là tần suất nộp bài ít nhất. Ta tạo một dataframe là tập con của *New_list_4* có tần suất bằng với tần suất nộp bài ít nhất. Cột *IDlist* của dataframe mới này chính là danh sách sinh viên có tần suất nộp bài ít nhất.

```
New_list_4=arrange(New_list_4, fre)
df_4d=subset(New_list_4, fre==New_list_4[["fre"]][1])
df_4d[["IDlist"]]
```

Output của file 6:

```
[1] "1511191" "1812257" "1812478" "1813096" "1813681" "1814518" "1820028" "1852443" "1910032" "1910060" "1910076" "1910094"
[14] "1910101" "1910110" "1910113" "1910123" "1910137" "1910198" "1910202" "1910224" "1910238" "1910265" "1910298" "1910346" "1910347"
[27] "1910402" "1910409" "1910473" "1910563" "1910565" "1910620" "1910643" "1910644" "1910663" "1910666" "1910735" "1910892" "1910916"
[40] "1910984" "1911000" "1911015" "1911044" "1911056" "1911058" "1911066" "1911136" "1911185" "1911207" "1911217" "1911262" "1911283"
[53] "1911285" "1911296" "1911314" "1911441" "1911456" "1911478" "1911520" "1911530" "1911565" "1911569" "1911591" "1911594" "1911650"
[66] "1911704" "1911736" "1911837" "1911841" "1911878" "1911907" "1911931" "1912041" "1912084" "1912123" "1912184" "1912190" "1912267"
[79] "1912288" "1912371" "1912384" "1912457" "1912463" "1912522" "1912523" "1912526" "1912539" "1912594" "1912602" "1912675" "1912677"
[92] "1912683" "1912705" "1912713" "1912715" "1912749" "1912798" "1912811" "1912817" "1912912" "1912958" "1912980" "1913021" "1913026"
[105] "1913040" "1913075" "1913094" "1913102" "1913114" "1913167" "1913186" "1913218" "1913228" "1913241" "1913254" "1913260" "1913261"
[118] "1913268" "1913334" "1913354" "1913355" "1913356" "1913380" "1913386" "1913396" "1913419" "1913430" "1913446" "1913457"
[131] "1913464" "1913467" "1913566" "1913599" "1913621" "1913629" "1913652" "1913678" "1913695" "1913713" "1913729" "1913758" "1913763"
[144] "1913764" "1913828" "1913832" "1914011" "1914022" "1914038" "1914054" "1914064" "1914078" "1914079" "1914084" "1914093"
[157] "1914121" "1914126" "1914220" "1914232" "1914291" "1914310" "1914316" "1914405" "1914424" "1914472" "1914477" "1914641" "1914651"
[170] "1914659" "1914677" "1914697" "1914698" "1914738" "1914763" "1914806" "1914807" "1914830" "1914845" "1914864" "1914878" "1914880"
[183] "1914881" "1914900" "1914914" "1915016" "1915063" "1915071" "1915076" "1915130" "1915146" "1915161" "1915251" "1915275" "1915294"
[196] "1915323" "1915329" "1915351" "1915442" "1915473" "1915474" "1915482" "1915486" "1915520" "1915540" "1915541" "1915551" "1915557"
[209] "1915650" "1915651" "1915656" "1915667" "1915745" "1915787" "1915795" "1915822" "1915865" "1915866" "1915931" "1915953" "1915982"
[222] "1915988" "1915991" "1916061" "1937024" "1937064"
```

Output của file 9:

```
[1] "1812257" "1812478" "1813096" "1813681" "1820028" "1910032" "1910038" "1910076" "1910094" "1910101" "1910110"
[12] "1910202" "1910339" "1910346" "1910351" "1910473" "1910565" "1910650" "1910663" "1910666" "1910892" "1910916"
[23] "1910984" "1911000" "1911015" "1911044" "1911056" "1911058" "1911186" "1911207" "1911283" "1911296" "1911314"
[34] "1911441" "1911456" "1911478" "1911530" "1911569" "1911591" "1911594" "1911796" "1911873" "1911878"
[45] "1911881" "1911900" "1911975" "1912041" "1912056" "1912084" "1912123" "1912184" "1912190" "1912267" "1912288"
[56] "1912384" "1912386" "1912410" "1912457" "1912463" "1912522" "1912523" "1912526" "1912594" "1912675" "1912676"
[67] "1912683" "1912700" "1912705" "1912715" "1912798" "1912811" "1912817" "1912912" "1912958" "1913021" "1913026"
[78] "1913026" "1913075" "1913102" "1913114" "1913167" "1913218" "1913228" "1913241" "1913261" "1913268" "1913343"
[89] "1913336" "1913354" "1913355" "1913380" "1913386" "1913419" "1913446" "1913457" "1913464" "1913466" "1913599"
[100] "1913621" "1913629" "1913651" "1913652" "1913695" "1913729" "1913763" "1913764" "1913775" "1913817" "1913832"
[111] "1913844" "1913918" "1913944" "1913990" "1914022" "1914052" "1914054" "1914055" "1914078" "1914079" "1914093"
[122] "1914121" "1914126" "1914220" "1914232" "1914291" "1914310" "1914384" "1914474" "1914651" "1914677" "1914713"
[133] "1914720" "1914763" "1914768" "1914806" "1914807" "1914845" "1914864" "1914914" "1915063" "1915071" "1915076"
[144] "1915146" "1915251" "1915275" "1915294" "1915351" "1915439" "1915442" "1915473" "1915474" "1915482" "1915486"
[155] "1915520" "1915540" "1915541" "1915557" "1915656" "1915745" "1915787" "1915865" "1915905" "1915919" "1915953"
[166] "1915988" "1915991" "1916022" "1937024"
```

Output của file 10:

```
[1] "1812257" "1812478" "1813681" "1910032" "1910038" "1910076" "1910094" "1910123"
[9] "1910224" "1910346" "1910473" "1910565" "1910984" "1911000" "1911136" "1911207"
[17] "1911283" "1911285" "1911441" "1911530" "1911565" "1911569" "1911594" "1911704"
[25] "1911837" "1911841" "1911931" "1912041" "1912056" "1912123" "1912190" "1912386"
[33] "1912463" "1912523" "1912526" "1912539" "1912579" "1912675" "1912683" "1912705"
[41] "1912761" "1912817" "1912912" "1912966" "1913026" "1913045" "1913102" "1913218"
[49] "1913241" "1913261" "1913268" "1913306" "1913334" "1913336" "1913354" "1913356"
[57] "1913380" "1913386" "1913419" "1913446" "1913457" "1913467" "1913599" "1913629"
[65] "1913652" "1913678" "1913695" "1913729" "1913817" "1913944" "1914011" "1914022"
[73] "1914054" "1914078" "1914084" "1914121" "1914220" "1914227" "1914232" "1914291"
[81] "1914424" "1914472" "1914477" "1914641" "1914651" "1914807" "1914880" "1915076"
[89] "1915133" "1915146" "1915251" "1915275" "1915294" "1915350" "1915351" "1915439"
[97] "1915473" "1915474" "1915482" "1915486" "1915551" "1915557" "1915562" "1915787"
[105] "1915795" "1915866" "1915903" "1915953" "1915983" "1915988" "1916022" "1937024"
[113] "1937064"
```

Output của file 12:

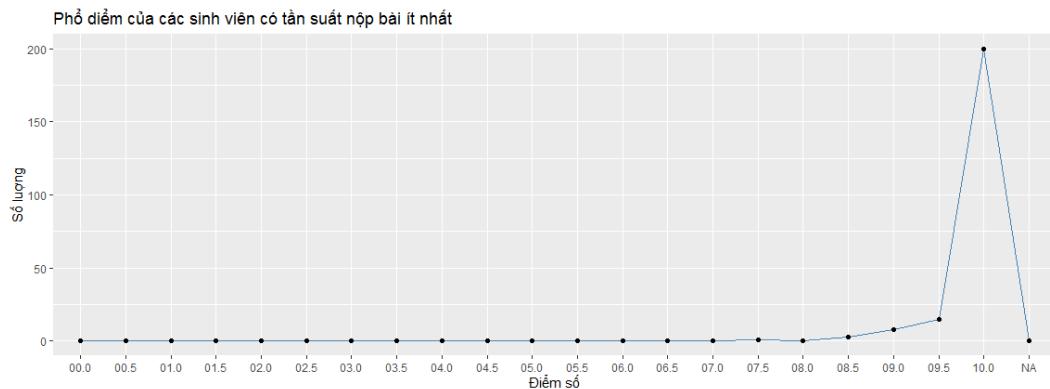
```
[1] "1613010" "1812257" "1812478" "1813096" "1813681" "1814096" "1814518" "1820028" "1910006" "1910032" "1910038"
[12] "1910060" "1910076" "1910094" "1910101" "1910110" "1910113" "1910137" "1910202" "1910224" "1910238" "1910265"
[23] "1910276" "1910298" "1910339" "1910346" "1910347" "1910351" "1910402" "1910473" "1910565" "1910620" "1910643"
[34] "1910644" "1910650" "1910663" "1910666" "1910865" "1910916" "1910984" "1911000" "1911015" "1911056" "1911058"
[45] "1911066" "1911110" "1911130" "1911185" "1911186" "1911207" "1911217" "1911283" "1911285" "1911296" "1911314"
[56] "1911478" "1911520" "1911530" "1911565" "1911569" "1911591" "1911594" "1911650" "1911704" "1911736" "1911796"
[67] "1911837" "1911841" "1911900" "1911907" "1911931" "1912041" "1912046" "1912056" "1912084" "1912184"
[78] "1912190" "1912237" "1912267" "1912384" "1912463" "1912523" "1912539" "1912579" "1912675" "1912683"
[89] "1912700" "1912715" "1912749" "1912811" "1912817" "1912912" "1912916" "1912966" "1912980" "1913021" "1913026"
[100] "1913040" "1913094" "1913102" "1913114" "1913123" "1913167" "1913186" "1913218" "1913241" "1913254" "1913261"
[111] "1913268" "1913306" "1913334" "1913336" "1913341" "1913354" "1913355" "1913356" "1913386" "1913396" "1913419"
[122] "1913433" "1913446" "1913457" "1913467" "1913566" "1913621" "1913651" "1913652" "1913678" "1913695" "1913713"
[133] "1913729" "1913763" "1913764" "1913775" "1913817" "1913828" "1913832" "1913844" "1913944" "1913949" "1914003"
[144] "1914011" "1914022" "1914038" "1914052" "1914054" "1914055" "1914078" "1914079" "1914084" "1914121" "1914210"
[155] "1914227" "1914232" "1914291" "1914310" "1914316" "1914384" "1914405" "1914424" "1914472" "1914641"
[166] "1914651" "1914674" "1914685" "1914698" "1914720" "1914738" "1914763" "1914768" "1914807" "1914830" "1914845"
[177] "1914864" "1914880" "1914881" "1914914" "1914979" "1915063" "1915071" "1915076" "1915130" "1915146" "1915251"
[188] "1915275" "1915294" "1915323" "1915329" "1915351" "1915439" "1915473" "1915474" "1915482" "1915486" "1915540"
[199] "1915551" "1915557" "1915562" "1915570" "1915598" "1915651" "1915656" "1915745" "1915787" "1915822" "1915865"
[210] "1915866" "1915882" "1915903" "1915953" "1915982" "1915988" "1915991" "1916022" "1916061" "1937024" "1937064"
```

e) Xác định phổ điểm của các sinh viên có tần suất nộp bài ít nhất

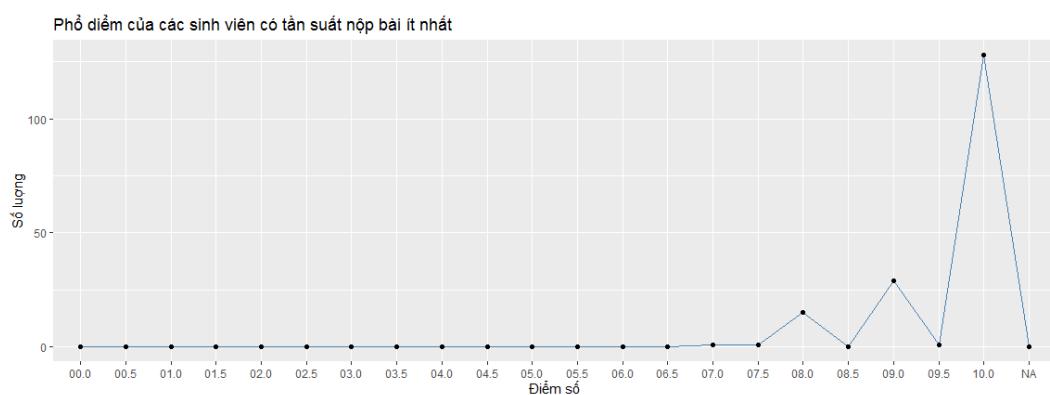
Ta xác định danh sách sinh viên có tần suất nộp bài ít nhất cũng như kết quả những lần nộp bài của họ. Từ đó biểu diễn được phổ điểm của những sinh viên này.

```
diem_4e=c("00.0", "00.5", "01.0", "01.5", "02.0", "02.5",
"03.0", "03.5", "04.0", "04.5", "05.0", "05.5", "06.0", "06.5",
"07.0", "07.5", "08.0", "08.5", "09.0", "09.5", "10.0", "NA")
New_list_4=arrange(New_list_4, fre)
df_4e=subset(New_list_4, fre==New_list_4[["fre"]][1])
df_4e1=subset(c, as.numeric(ID) %in% as.numeric(df_4e[["IDlist"]]))
f_4e=c()
for(i in 0:20){
df_4e2=subset(df_4e1, Diem==0.5*i)
f_4e <- c(f_4e, nrow(df_4e2))
}
df_4e2=subset(df_4e1, is.na(Diem))
f_4e <- c(f_4e, nrow(df_4e2))
df_4e3=data.frame(diem_4e, f_4e)
df_4e3 %>%
ggplot(aes(y=as.numeric(f_4e), x=diem_4e, group=1))+geom_line(color="steelblue") + geom_point() + labs(x="Điểm số",
y="Số lượng", title="Phổ điểm của các sinh viên có tần suất nộp bài ít nhất")
```

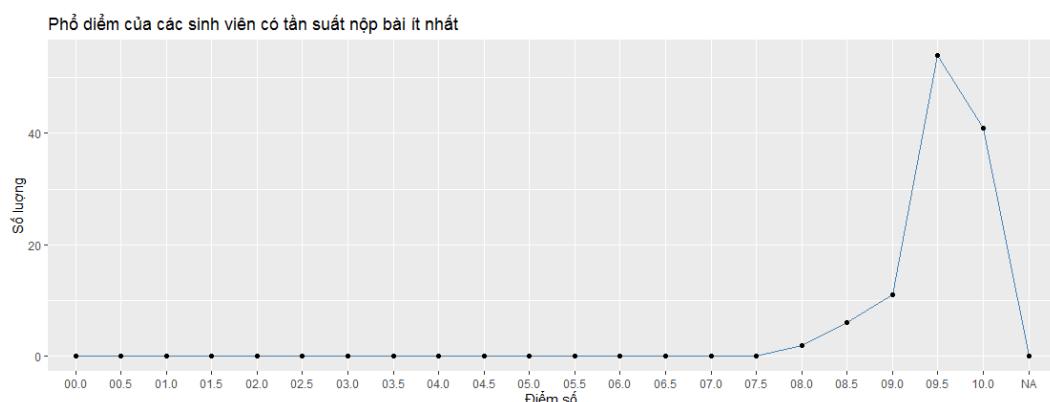
Output của file 6:



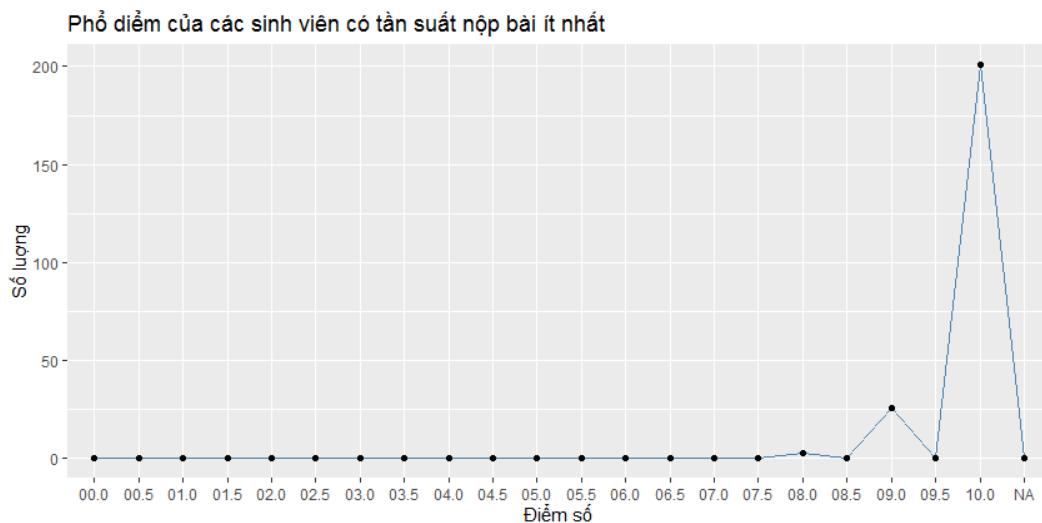
Output của file 9:



Output của file 10:



Output của file 12:



f) Xác định số lượng sinh viên có tần suất nộp bài nhiều nhất

Sắp xếp dataframe `New_list_4` theo thứ tự `fre` giảm dần, phần tử `fre` đầu tiên lúc này chính là tần suất nộp bài nhiều nhất. Ta tạo một dataframe là tập con của `New_list_4` có tần suất bằng với tần suất nộp bài nhiều nhất. Cột `IDlist` của dataframe mới này chính là danh sách sinh viên có tần suất nộp bài nhiều nhất. Ta dùng lệnh `nrow()` để tính số lượng sinh viên có tần suất nộp bài nhiều nhất.

```
New_list_4=arrange(New_list_4, desc(fre))
df_4f=subset(New_list_4, fre==New_list_4[["fre"]][1])
nrow(df_4f)
```

Output của file 6:

```
[1] 1
```

Output của file 9:

```
[1] 1
```

Output của file 10:

```
[1] 1
```

Output của file 12:

```
[1] 1
```

g) Xác định các sinh viên có tần suất nộp bài nhiều nhất

Danh sách sinh viên có tần suất nộp bài nhiều nhất:

```
df_4f[["IDlist"]]
```

Output của file 6:

```
> df_4f[["IDlist"]]
[1] "1910038"
```

Output của file 9:

```
[1] "1914405"
```

Output của file 10:

```
[1] "1915520"
```

Output của file 12:

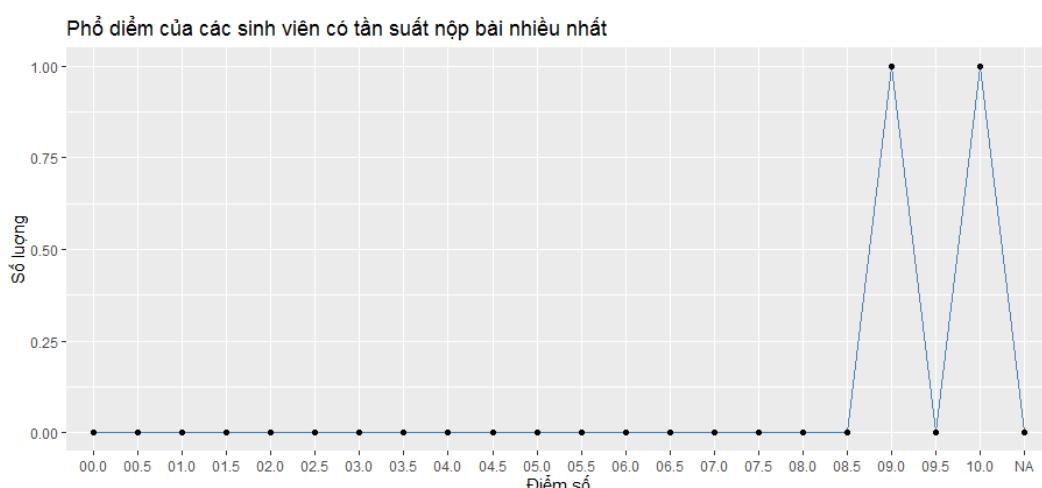
```
[1] "1915442"
```

h) Xác định phổ điểm của các sinh viên có tần suất nộp bài nhiều nhất.

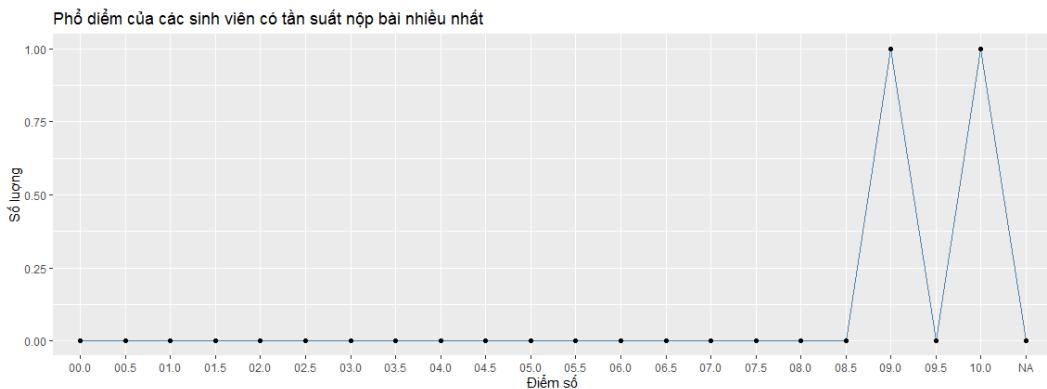
Tương tự cách làm của câu 4e chỉ khác lúc này ta sẽ tạo tập con chứa thông tin các lần nộp bài của các sinh viên có tần suất nộp bài nhiều nhất.

```
diem_4h=c("00.0", "00.5", "01.0", "01.5", "02.0", "02.5",
"03.0", "03.5", "04.0", "04.5","05.0", "05.5", "06.0", "06.5",
"07.0", "07.5", "08.0", "08.5", "09.0", "09.5", "10.0", "NA")
New_list_4=arrange(New_list_4, fre)
df_4h=subset(New_list_4, fre==New_list_4[["fre"]][1])
df_4h1=subset(c, as.numeric(ID) %in% as.numeric(df_4h[["IDlist"]]))
f_4h=c()
for(i in 0:20){
df_4h2=subset(df_4h1, Diem==0.5*i)
f_4h <- c(f_4h, nrow(df_4h2))
}
df_4h2=subset(df_4h1, is.na(Diem))
f_4h <- c(f_4h, nrow(df_4h2))
df_4h3=data.frame(diem_4h, f_4h)
df_4h3 %>%
ggplot(aes(y=as.numeric(f_4h), x=diem_4h, group=1))+geom_line(color="steelblue") +geom_point() +labs(x="Điểm số",
y="Số lượng", title="Phổ điểm của các sinh viên có tần suất nộp bài nhiều nhất")
```

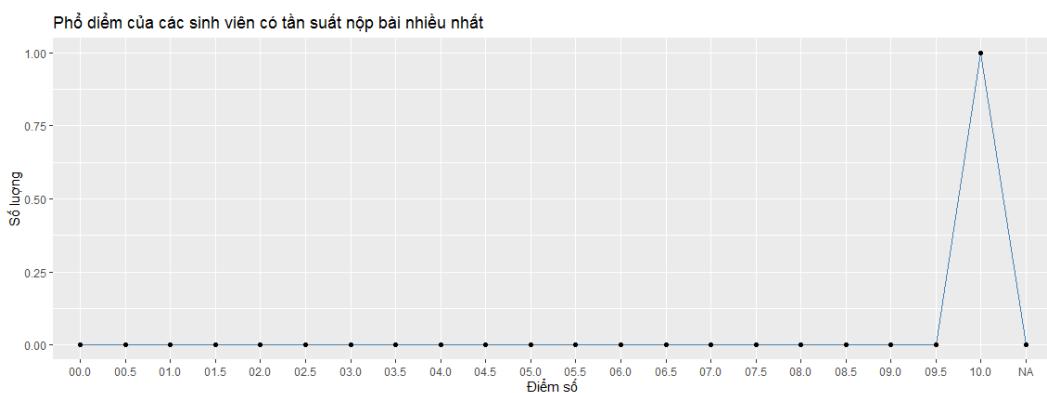
Output của file 6:



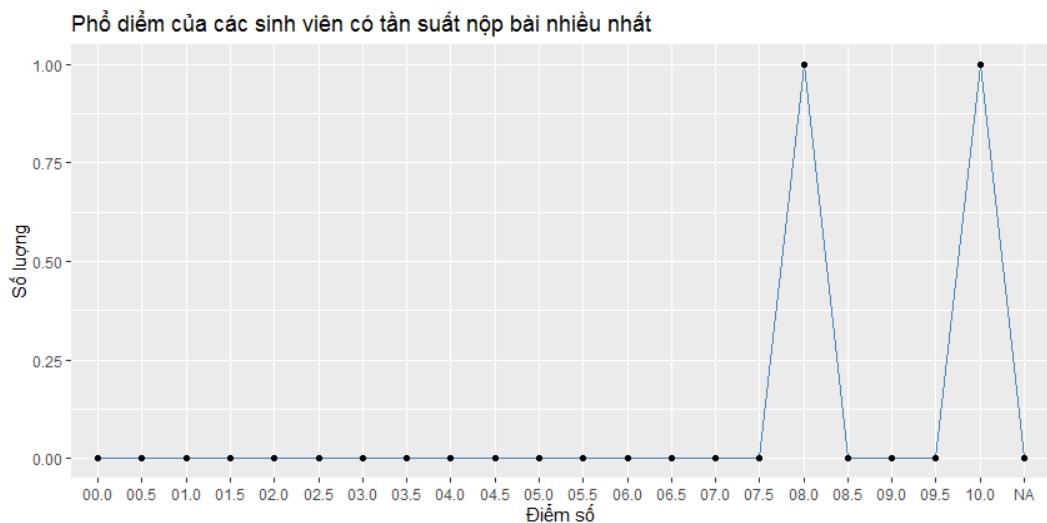
Output của file 9:



Output của file 10:



Output của file 12:



i) Xác định các sinh viên nằm trong nhóm có tần suất nộp bài nhiều nhì

Để xác định các sinh viên nằm trong nhóm có tần suất nộp bài nhiều nhì, ta sẽ sắp xếp dataframe theo thứ tự *fre* giảm dần. Tiếp theo ta dựa vào số lượng sinh viên có tần suất nộp bài lớn nhất để xác định vị trí của sinh viên đầu tiên có tần suất nộp bài cao thứ nhì (chính là *nrow(df_4f) + 1*)

```
df_4i=subset(New_list_4,  
fre==New_list_4[["fre"]][nrow(df_4f)+1])  
df_4i[["IDlist"]]
```

Output của file 6:



```
[1] "1914003"
```

Output của file 9:

```
[1] "1910620"
```

Output của file 10:

```
[1] "1914738"
```

Output của file 12:

```
[1] "1915520"
```

j) Xác định các sinh viên nằm trong nhóm có tần suất nộp bài nhiều nhất hoặc nhiều nhì.

Ta tạo 2 vector chứa danh sách sinh viên có tần suất nộp bài nhiều nhất và danh sách sinh viên có tần suất nộp bài ít nhất. Cuối cùng ta tạo một vector là vector tổng của 2 vector vừa rồi.

```
c_4j1=df_4f[["IDlist"]]
c_4j2=df_4i[["IDlist"]]
c_4j=c(c_4j1, c_4j2)
c_4j
```

Output của file 6:

```
[1] "1910038" "1914003"
```

Output của file 9:

```
[1] "1914405" "1910620"
```

Output của file 10:

```
[1] "1915520" "1914738"
```

Output của file 12:

```
[1] "1915442" "1915520"
```

k) Hãy tính thời gian trung bình (tính bằng giây) giữa hai lần nộp bài liên nhau của cùng một sinh viên trong mẫu đã chọn.

Ta tạo biến $sumT$ là tổng thời gian giữa hai lần nộp liên tiếp của cùng một sinh viên trong mẫu. Ta sẽ dùng vòng lặp để tính tổng này. Lưu ý những giá trị NA sẽ không được tính. Số lượng hai lần nộp bài liên tiếp của một sinh viên sẽ bằng số lần nộp bài trừ cho số sinh viên. Biến mT lưu trữ giá trị thời gian trung bình cần tính.

```
sumT=0
t3 <- dmy_hm(sapxep_ID$Etime)
for( i in 1:(n-2)){
  if(sapxep_ID[["ID"]][i]==sapxep_ID[["ID"]][i+1]){
    if(is.na(t3[i+1])!=TRUE && is.na(t3[i])!=TRUE)
      sumT=abs(as.numeric(difftime(t3[i+1], t3[i],
        units="secs")))+sumT
  }
  mT=as.numeric(sumT)/(n-1+n_sv)
  mT
```

Output ở file 6:



[1] 8392.372

Output ở file 9:

[1] 9698.316

Output ở file 10:

[1] 10520.43

Output ở file 12:

[1] 7151.469

- 1) Tính tần số, tần suất và tần suất tích lũy của mẫu trên

Tần số là số bài làm đạt một mức điểm nào đó. Tần suất nộp bài bằng số lần nộp bài chia cho số bài đạt một mức điểm nào đó. Còn tần suất tích lũy sẽ được tính bằng hàm cumsum()

```
diem_4l=c("00.0", "00.5", "01.0", "01.5", "02.0", "02.5",
"03.0", "03.5", "04.0", "04.5","05.0", "05.5", "06.0", "06.5",
"07.0", "07.5", "08.0", "08.5", "09.0", "09.5", "10.0", "NA")
f_4l=c() # Tần số
# Xóa hàng tính trung bình điểm
sapxep_ID<-sapxep_ID[1:(n-1), ]
for(i in 0:20){
df_4l=subset(sapxep_ID, Diem==0.5*i)
f_4l <- c(f_4l, nrow(df_4l))
}
df_4lmno=data.frame(diem_4l, f_4l)
F_4l=c() # Vector tần suất
F_4l=as.numeric(df_4lmno$f_4l)/(n-1)
df_4lmno=cbind(df_4lmno, F_4l)
cfre_4l=cumsum(df_4lmno$F_4l)
df_4lmno=cbind(df_4lmno, cfre_4l)
View(df_4lmno)
```

Output của file 6, cột *diem_4l* chứa dữ liệu số lần nộp bài, cột *f_4l* chứa dữ liệu tần số, cột *F_4l* chứa dữ liệu tần suất còn *cfre_4l* chứa dữ liệu tần suất tích lũy của mẫu: Output của file 6:

	diem_4l	f_4l	F_4l	cfre_4l
1	00.0	0	0.0000000000	0.0000000000
2	00.5	0	0.0000000000	0.0000000000
3	01.0	0	0.0000000000	0.0000000000
4	01.5	0	0.0000000000	0.0000000000
5	02.0	0	0.0000000000	0.0000000000
6	02.5	0	0.0000000000	0.0000000000
7	03.0	2	0.004535147	0.004535147
8	03.5	0	0.0000000000	0.004535147
9	04.0	0	0.0000000000	0.004535147
10	04.5	1	0.002267574	0.006802721
11	05.0	1	0.002267574	0.009070295
12	05.5	1	0.002267574	0.011337868
13	06.0	0	0.0000000000	0.011337868
14	06.5	1	0.002267574	0.013605442
15	07.0	3	0.006802721	0.020408163
16	07.5	7	0.015873016	0.036281179
17	08.0	5	0.011337868	0.047619048
18	08.5	16	0.036281179	0.083900227
19	09.0	46	0.104308390	0.188208617
20	09.5	68	0.154195011	0.342403628
21	10.0	288	0.653061224	0.995464853
22	NA	2	0.004535147	1.0000000000

Output của file 9:



	diem_4l	f_4l	F_4l	cfre_4l
1	00.0	1	0.002145923	0.002145923
2	00.5	0	0.0000000000	0.002145923
3	01.0	0	0.0000000000	0.002145923
4	01.5	0	0.0000000000	0.002145923
5	02.0	1	0.002145923	0.004291845
6	02.5	0	0.0000000000	0.004291845
7	03.0	1	0.002145923	0.006437768
8	03.5	0	0.0000000000	0.006437768
9	04.0	0	0.0000000000	0.006437768
10	04.5	0	0.0000000000	0.006437768
11	05.0	0	0.0000000000	0.006437768
12	05.5	1	0.002145923	0.008583691
13	06.0	1	0.002145923	0.010729614
14	06.5	1	0.002145923	0.012875536
15	07.0	11	0.023605150	0.036480687
16	07.5	2	0.004291845	0.040772532
17	08.0	50	0.107296137	0.148068670
18	08.5	4	0.008583691	0.156652361
19	09.0	118	0.253218884	0.409871245
20	09.5	8	0.017167382	0.427038627
21	10.0	264	0.566523605	0.993562232
22	NA	3	0.006437768	1.0000000000

Output của file 10:

	diem_4l	f_4l	F_4l	cfre_4l
1	00.0	1	0.001814882	0.001814882
2	00.5	0	0.0000000000	0.001814882
3	01.0	0	0.0000000000	0.001814882
4	01.5	0	0.0000000000	0.001814882
5	02.0	0	0.0000000000	0.001814882
6	02.5	0	0.0000000000	0.001814882
7	03.0	0	0.0000000000	0.001814882
8	03.5	0	0.0000000000	0.001814882
9	04.0	0	0.0000000000	0.001814882
10	04.5	0	0.0000000000	0.001814882
11	05.0	0	0.0000000000	0.001814882
12	05.5	0	0.0000000000	0.001814882
13	06.0	0	0.0000000000	0.001814882
14	06.5	0	0.0000000000	0.001814882
15	07.0	1	0.001814882	0.003629764
16	07.5	2	0.003629764	0.007259528
17	08.0	9	0.016333938	0.023593466
18	08.5	27	0.049001815	0.072595281
19	09.0	51	0.092558984	0.165154265
20	09.5	256	0.464609800	0.629764065
21	10.0	203	0.368421053	0.998185118
22	NA	1	0.001814882	1.0000000000

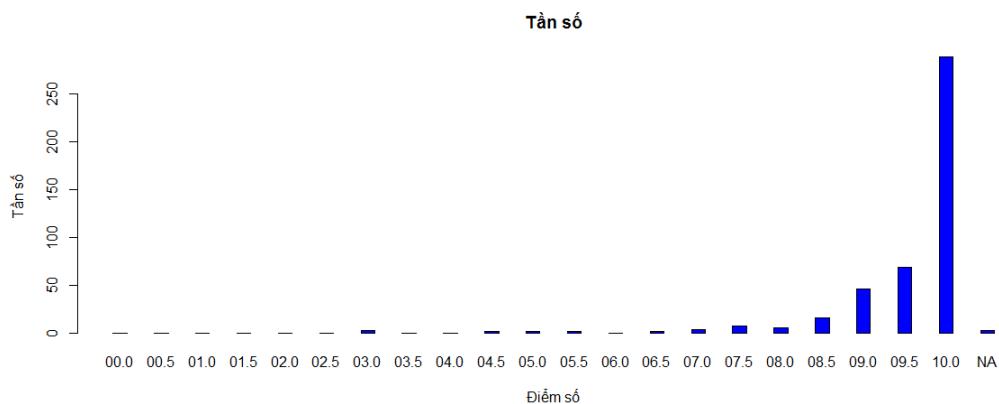
Output của file 12:

	diem_4l	f_4l	F_4l	cfre_4l
1	00.0	1	0.002832861	0.002832861
2	00.5	0	0.0000000000	0.002832861
3	01.0	0	0.0000000000	0.002832861
4	01.5	0	0.0000000000	0.002832861
5	02.0	0	0.0000000000	0.002832861
6	02.5	0	0.0000000000	0.002832861
7	03.0	0	0.0000000000	0.002832861
8	03.5	0	0.0000000000	0.002832861
9	04.0	1	0.002832861	0.005665722
10	04.5	0	0.0000000000	0.005665722
11	05.0	0	0.0000000000	0.005665722
12	05.5	0	0.0000000000	0.005665722
13	06.0	1	0.002832861	0.008498584
14	06.5	0	0.0000000000	0.008498584
15	07.0	2	0.005665722	0.014164306
16	07.5	0	0.0000000000	0.014164306
17	08.0	6	0.016997167	0.031161473
18	08.5	0	0.0000000000	0.031161473
19	09.0	82	0.232294618	0.263456091
20	09.5	0	0.0000000000	0.263456091
21	10.0	260	0.736543909	1.0000000000
22	NA	0	0.0000000000	1.0000000000

- m) Vẽ biểu đồ tần số của mẫu trên. Hãy nhận xét về biểu đồ
 Ta dùng hàm barplot() để vẽ biểu đồ hình cột minh họa cho số lượng bài làm theo từng mức điểm.

```
barplot( df_4lmno$f_4l, width = 1, space=2, main="Tần số",
names.arg=df_4lmno$diem_4l, xlab="Điểm số", ylab="Tần số", col="blue")
```

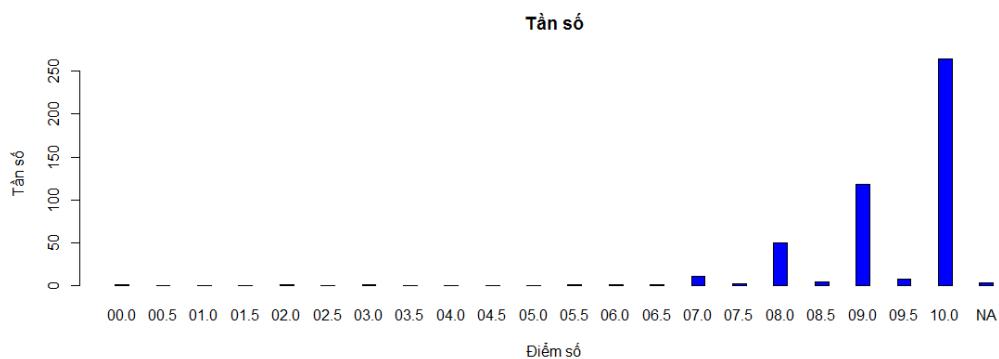
Output của file 6:



Nhận xét:

- Đa số các bài đạt 10 điểm, tiếp theo là 9.5 điểm.
- Tồn tại những lần làm bài của sinh viên tuy nhiên không được chấm điểm (cột NA).

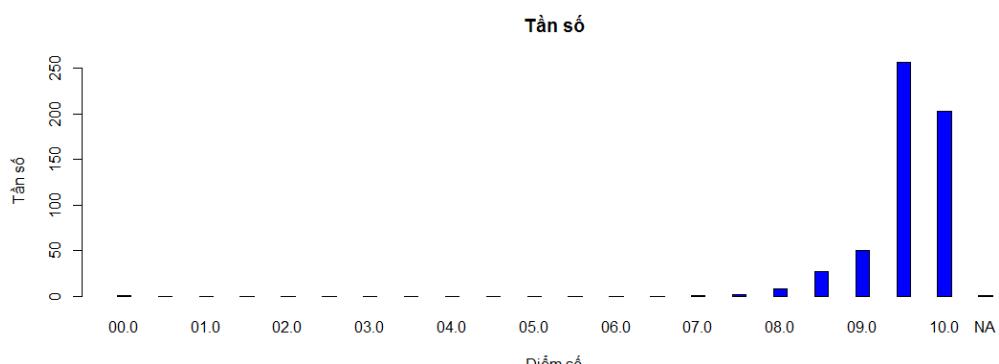
Output của file 9:



Nhận xét:

- Đa số các bài đạt 10 điểm, tiếp theo là 9 điểm.
- Tồn tại những lần làm bài của sinh viên tuy nhiên không được chấm điểm (cột NA).

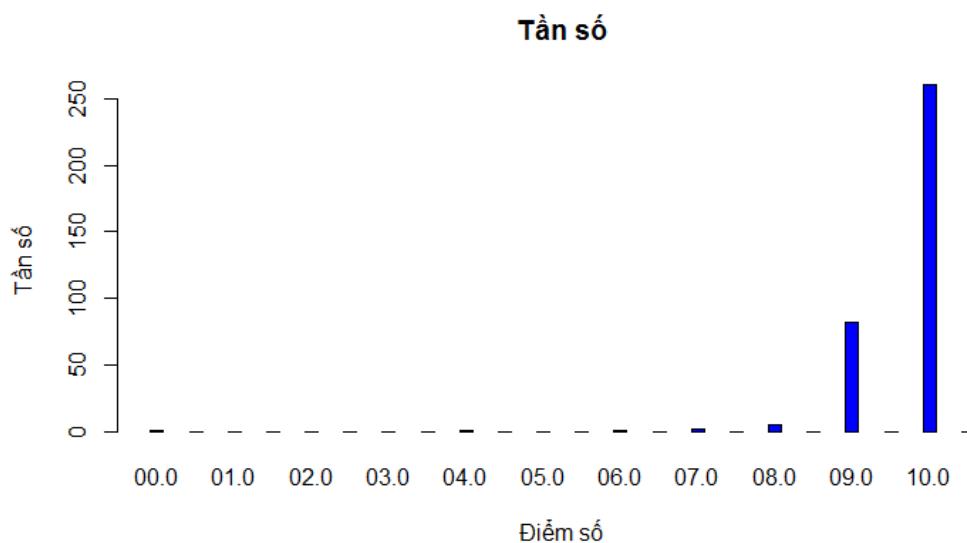
Output của file 10:



Nhận xét:

- Đa số các bài đạt 9.5 điểm, tiếp theo là 10 điểm.

Output của file 12:



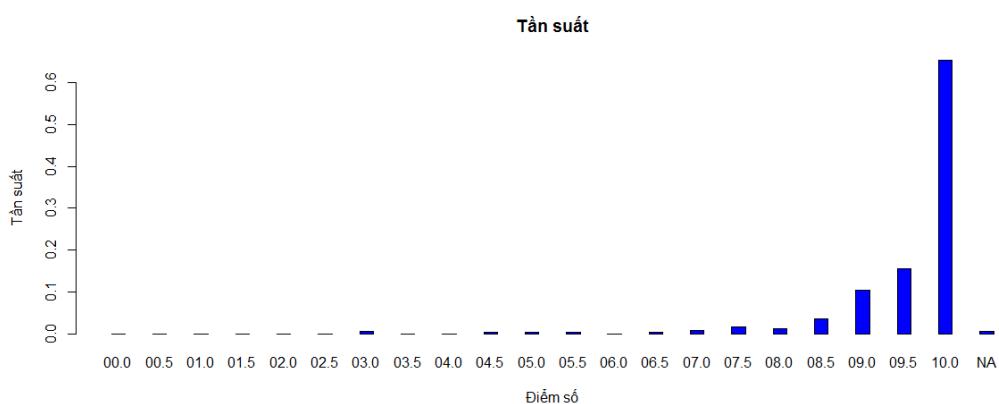
Nhận xét:

- Đa số các bài đạt 10 điểm, tiếp theo là 9 điểm.
- Tất cả bài làm đều được chấm điểm).

n) Vẽ biểu đồ tần suất của mẫu trên. Hãy nhận xét về biểu đồ
Ta tiếp tục dùng hàm barplot() để biểu diễn tần suất của từng mức điểm

```
barplot( df_4lmno$F_4l, width = 1, space=2, main="Tần suất",
names.arg=df_4lmno$diem_4l, xlab="Điểm số", ylab="Tần suất", col="blue")
```

Output của file 6:

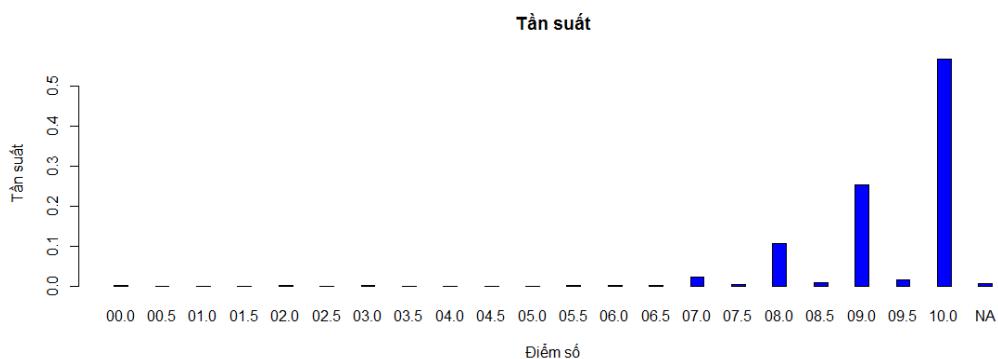


Nhận xét:

- Khoảng hơn 70% bài làm đạt 10 điểm.

- Các bài làm còn lại đa số từ 8.5-9.5 điểm.

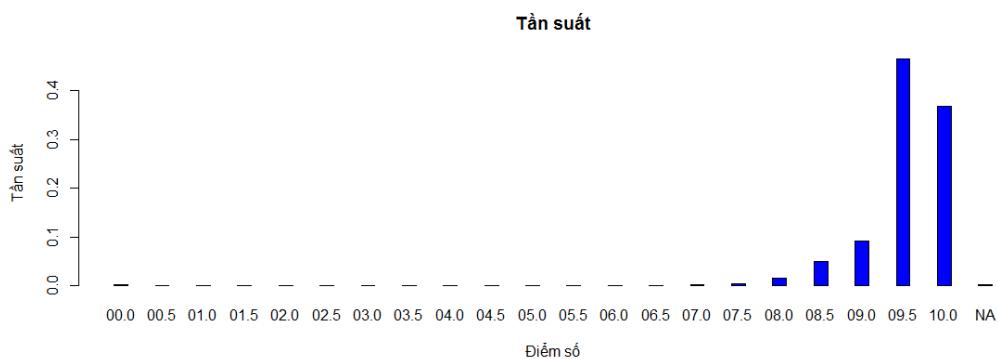
Output của file 9:



Nhận xét:

- Khoảng hơn 70% bài làm đạt 10 điểm.
- Các bài làm còn lại đa số đạt 8 và 9 điểm.

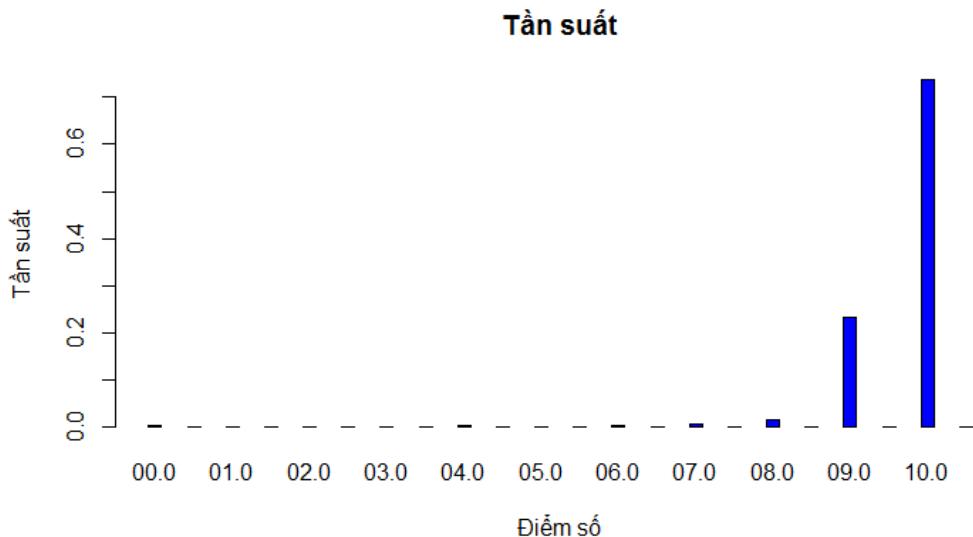
Output của file 10:



Nhận xét:

- Khoảng 50% bài làm đạt 9.5 điểm.
- Các bài làm còn lại đa số đạt 10 điểm.

Output của file 12:



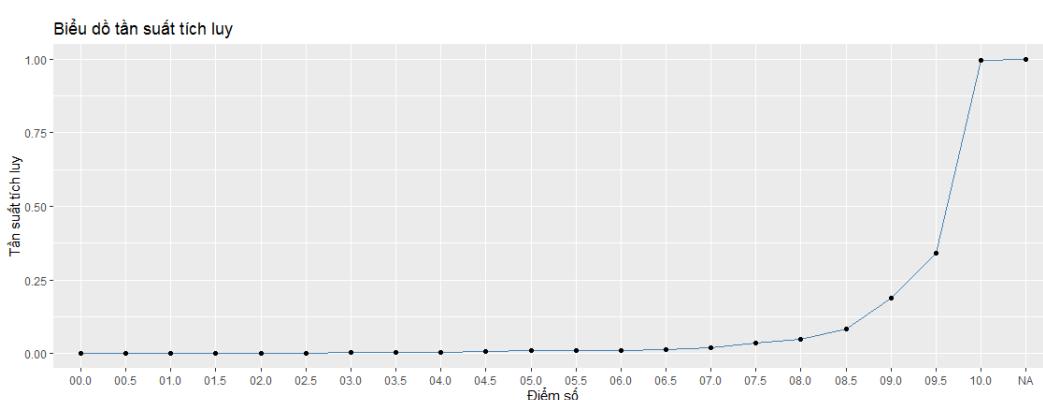
Nhận xét:

- Khoảng 80% bài làm đạt 10 điểm.
- Các bài làm còn lại đa số đạt 9 điểm.

- o) Vẽ biểu đồ tần suất tích lũy của mẫu trên. Hãy nhận xét về biểu đồ.
Ta dùng hàm ggplot() để vẽ biểu đồ đường. Trước đó ta cần khai báo thư viện "ggplot2".

```
df_4o=data.frame(df_4lmno$diem_41, df_4lmno$cfre_41)
df_4o=arrange(df_4o, cfre_41)
df_4o %>% ggplot(aes(y=as.numeric(df_4lmno.cfre_41), x=df_4lmno.diem_41,
group=1))+geom_line(color="steelblue")+geom_point()+labs(x="Điểm số",
y="Tần suất tích lũy", title="Biểu đồ tần suất tích lũy")
```

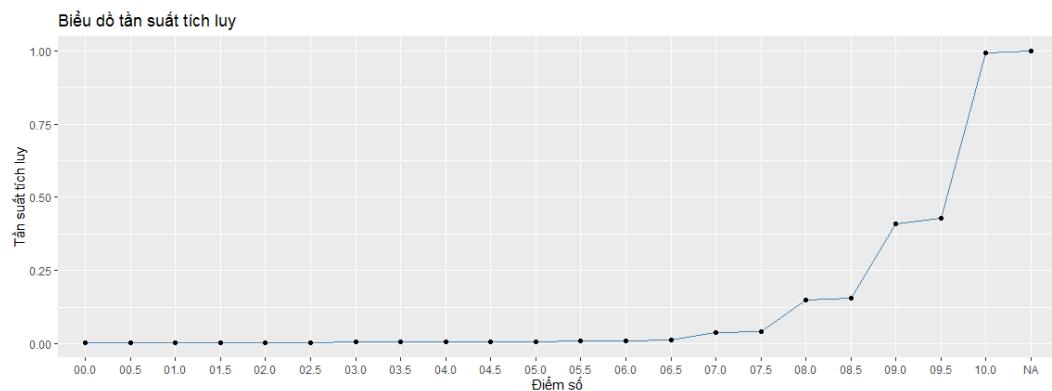
Output của file 6:



Nhận xét:

- Tần suất tích lũy nộp bài tăng dần.
- Mức độ tăng tần số tích lũy không đồng đều, tăng đột ngột trong khoảng từ 9.5 đến 10.

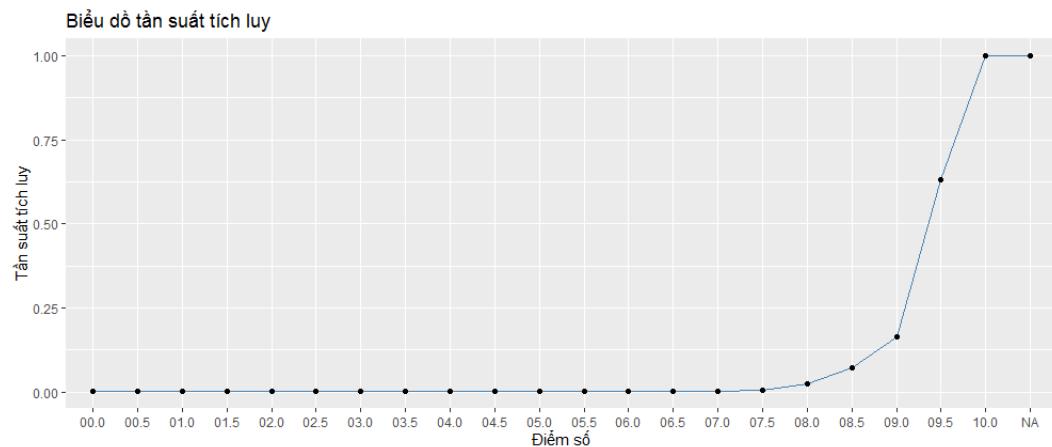
Output của file 9:



Nhận xét:

- Tần suất tích lũy nộp bài tăng dần.
- Mức độ tăng tần số tích lũy không đồng đều, tăng đột ngột trong khoảng từ 9.5 đến 10.

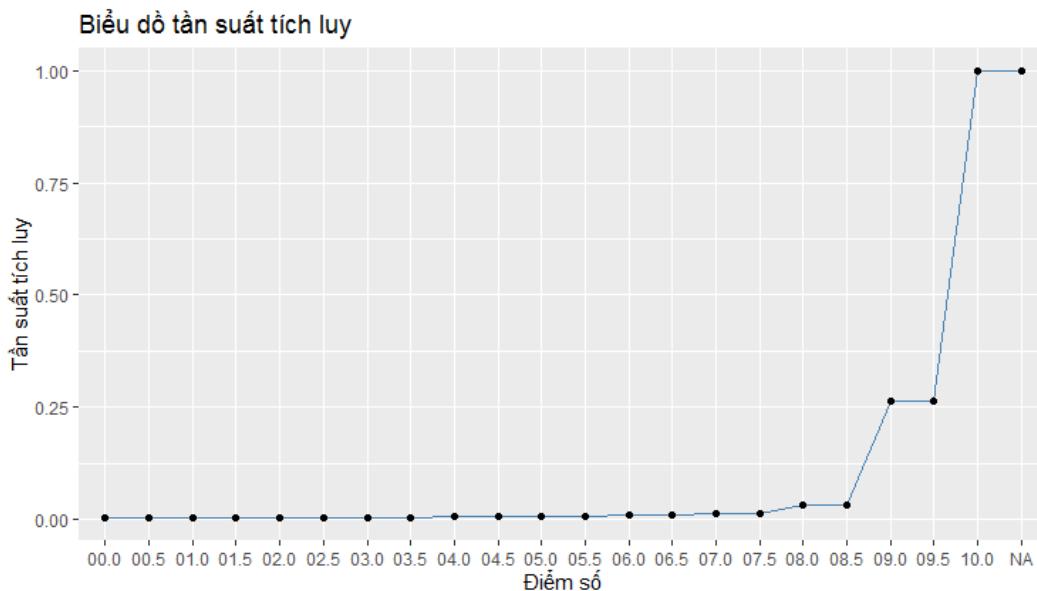
Output của file 10:



Nhận xét:

- Tần suất tích lũy nộp bài tăng dần.
- Mức độ tăng tần số tích lũy không đồng đều, tăng đột ngột trong khoảng từ 9 đến 10.

Output của file 12:



Nhận xét:

- Tần suất tích lũy nộp bài tăng dần.
- Mức độ tăng tần số tích lũy không đồng đều, tăng đột ngột trong khoảng từ 9.5 đến 10.

p) Tính trung vị mẫu, cực đại mẫu, cực tiểu mẫu của trên.

Ta sử dụng hàm summary() để tìm trung vị mẫu, cực đại mẫu và cực tiểu mẫu của mẫu trên. Đồng thời ta sẽ tìm được cả tứ phân vị thứ nhất và thứ ba của mẫu. Min: cực tiểu mẫu

1st Qu: tứ phân vị thứ nhất

Median: trung vị mẫu

Mean: trung bình

3rd Qu: tứ phân vị thứ ba

Max: cực đại mẫu

NA's: số giá trị NA

```
summary(New_list_4$fre)
```

Output của file 6:

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	NA's
0	0	0	9272	30	702060	2

Output của file 9:

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	NA's
0	0	0	9905	40	1146240	3

Output của file 10:

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
0	0	30	11667	60	982740

Output của file 12:



Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
0	0	0	6913	0	764430

- q) Hãy đo mức độ phân tán của điểm số (xung quanh giá trị trung bình) của mẫu.

Để đo mức độ phân tán của điểm số (xung quanh giá trị trung bình của mẫu) ta sẽ tính phương sai mẫu và độ lệch chuẩn. Ta dùng hàm var() để tính phương sai và dùng hàm sd() để tính độ lệch chuẩn. Tuy nhiên trước đó ta sẽ quy định những giá trị tần số NA sang 0.

```
New_list_4$fre[is.na(New_list_4$fre)] <- 0
var(New_list_4$fre) #Phương sai
sd(New_list_4$fre) #Độ lệch chuẩn
```

Output của file 6:

```
> var(New_list_4$fre) #Phương sai
[1] 4868264934
> sd(New_list_4$fre) #Độ lệch chuẩn
[1] 69772.95
```

Output của file 9:

```
> var(New_list_4$fre) #Phương sai
[1] 6078222421
> sd(New_list_4$fre) #Độ lệch chuẩn
[1] 77962.96
```

Output của file 10:

```
> var(New_list_4$fre) #Phương sai
[1] 5709093957
> sd(New_list_4$fre) #Độ lệch chuẩn
[1] 75558.55
> summary(New_list_4$fre)
```

Output của file 12:

```
> var(New_list_4$fre) #Phương sai
[1] 3995920366
> sd(New_list_4$fre) #Độ lệch chuẩn
[1] 63213.29
```

- r) Tính độ méo lệch (skewness), và độ nhọn (kurtosis) của dữ liệu trong mẫu trên.

Ta áp dụng công thức tính độ méo lệch $Skew = \frac{\frac{1}{n} \sum_{i=1}^n (X_i - \bar{X})^3}{\sqrt{\frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X})^2}^3}$ Ta áp dụng công thức tính độ nhọn $Kurt = \frac{\frac{1}{n} \sum_{i=1}^n (X_i - \bar{X})^4}{(\frac{1}{n} \sum_{i=1}^n (X_i - \bar{X})^2)^2}$

```
skewness(New_list_4$fre) kurtosis(New_list_4$fre)
```

Output của file 6:

```
> skewness(New_list_4$fre)
[1] 8.522071
> kurtosis(New_list_4$fre)
[1] 74.81918
```

Output của file 9:



```
> skewness(New_list_4$fre)
[1] 11.32464
> kurtosis(New_list_4$fre)
[1] 149.7745
```

Output của file 10:

```
> skewness(New_list_4$fre)
[1] 9.309026
> kurtosis(New_list_4$fre)
[1] 102.569
```

Output của file 12:

```
> skewness(New_list_4$fre)
[1] 10.42429
> kurtosis(New_list_4$fre)
[1] 112.203
```

s) Tính tứ phân vị (quartile) thứ nhất (Q_1) và thứ ba (Q_3) của mẫu.

Ta có công thức tính tứ phân vị thứ nhất và thứ ba:

$$Q_1 = 25(n+1)/100$$

$$Q_3 = 75(n+1)/100$$

Kết quả đã có khi sử dụng hàm summary() ở câu p.

4.5 Câu 5: Nhóm câu hỏi liên quan điểm tổng hợp, điểm trung bình của sinh viên

Gọi điểm số lần nộp bài thứ k của sinh viên i với i là uid và $k \in (1, 2, 3, \dots)$. Điểm tổng hợp của sinh viên tính tới lần nộp thứ k là điểm lớn nhất cho bài tập đó mà sinh viên đạt được cho tới lần nộp thứ k , tức là:

$$score_{ik} = \max(s_{i1}, s_{i2}, \dots, s_{ik})$$

Đối với sinh viên nộp ít hơn k lần thì vẫn tính theo công thức với giá trị khuyết xem như là 0.

a) Tính và vẽ biểu đồ sự phân bố về điểm đạt được của sinh viên sau $k = 6$ lần nộp bài

Ta sẽ tạo một ma trận để lưu điểm mỗi lần nộp bài của sinh viên. Tìm điểm tổng hợp của mỗi sinh viên bằng hàm max.

```
k=6
LanNop=matrix('NULL', nrow=n_sv, ncol=k)
i=0
for(l in 1:n_sv){
  for(j in 1:numOfSubs[l]){
    i=i+1
    if(is.na(sapxep_ID[["Diem"]][i]) == TRUE{ LanNop[1,j]=0 }
    else{ LanNop[1,j]<-sapxep_ID[["Diem"]][i] }
  }
  for(j in (numOfSubs[l]+1):k){ LanNop[1,j]=0 }
}
colnames(LanNop)<-c("L1","L2","L3","L4","L5","L6")
LanNop=data.frame(LanNop) LanNop$L1=as.numeric(LanNop$L1)
LanNop$L2=as.numeric(LanNop$L2)
LanNop$L3=as.numeric(LanNop$L3)
LanNop$L4=as.numeric(LanNop$L4)
LanNop$L5=as.numeric(LanNop$L5)
LanNop$L6=as.numeric(LanNop$L6)
Diemcuocung=c()
for(i in 1:n_sv){
  DiemCuoiCung<-c(DiemCuoiCung,max(LanNop[i,]))
}
LanNop=cbind(LanNop,DiemCuoiCung)
```

Output file 6:

	L1	L2	L3	L4	L5	L6	DiemCuoiCung
1	9.61	0.0	0	0	0	0	9.61
2	10.00	0.0	0	0	0	0	10.00
3	7.50	9.0	0	0	0	0	9.00
4	10.00	0.0	0	0	0	0	10.00
5	9.00	10.0	0	0	0	0	10.00
6	8.50	0.0	0	0	0	0	8.50
7	9.50	10.0	0	0	0	0	10.00
8	10.00	0.0	0	0	0	0	10.00
9	9.00	0.0	0	0	0	0	9.00
10	9.50	10.0	0	0	0	0	10.00
11	10.00	0.0	0	0	0	0	10.00
12	10.00	0.0	0	0	0	0	10.00
13	10.00	0.0	0	0	0	0	10.00
14	10.00	0.0	0	0	0	0	10.00
15	9.50	10.0	0	0	0	0	10.00
16	10.00	0.0	0	0	0	0	10.00
17	9.00	10.0	0	0	0	0	10.00
18	10.00	0.0	0	0	0	0	10.00

Output file 9:

	L1	L2	L3	L4	L5	L6	DiemCuoiCung
1	9.35	0.0	0	0	0	0	9.35
2	8.00	9.0	10	0	0	0	10.00
3	7.00	10.0	0	0	0	0	10.00
4	10.00	0.0	0	0	0	0	10.00
5	7.00	10.0	0	0	0	0	10.00
6	7.50	0.0	0	0	0	0	7.50
7	9.00	0.0	0	0	0	0	9.00
8	7.00	10.0	0	0	0	0	10.00
9	8.00	0.0	0	0	0	0	8.00
10	9.00	9.0	10	0	0	0	10.00
11	8.00	10.0	0	0	0	0	10.00
12	9.00	0.0	0	0	0	0	9.00
13	9.00	10.0	0	0	0	0	10.00
14	9.00	10.0	0	0	0	0	10.00
15	10.00	0.0	0	0	0	0	10.00
16	10.00	0.0	0	0	0	0	10.00
17	9.00	10.0	0	0	0	0	10.00
18	9.00	0.0	0	0	0	0	9.00

Output file 10:

	L1	L2	L3	L4	L5	L6	DiemCuoiCung
1	9.54	0.0	0.0	0.0	0.0	0	9.54
2	9.00	9.5	10.0	0.0	0.0	0	10.00
3	9.00	9.5	10.0	0.0	0.0	0	10.00
4	9.00	0.0	0.0	0.0	0.0	0	9.00
5	8.00	10.0	0.0	0.0	0.0	0	10.00
6	8.50	0.0	0.0	0.0	0.0	0	8.50
7	9.00	10.0	0.0	0.0	0.0	0	10.00
8	10.00	0.0	0.0	0.0	0.0	0	10.00
9	9.50	9.5	0.0	0.0	0.0	0	9.50
10	9.50	10.0	0.0	0.0	0.0	0	10.00
11	8.00	10.0	0.0	0.0	0.0	0	10.00
12	9.50	10.0	0.0	0.0	0.0	0	10.00
13	8.50	9.5	10.0	0.0	0.0	0	10.00
14	10.00	0.0	0.0	0.0	0.0	0	10.00
15	10.00	0.0	0.0	0.0	0.0	0	10.00
16	9.50	9.5	10.0	0.0	0.0	0	10.00
17	9.50	0.0	0.0	0.0	0.0	0	9.50
18	9.50	0.0	0.0	0.0	0.0	0	9.50

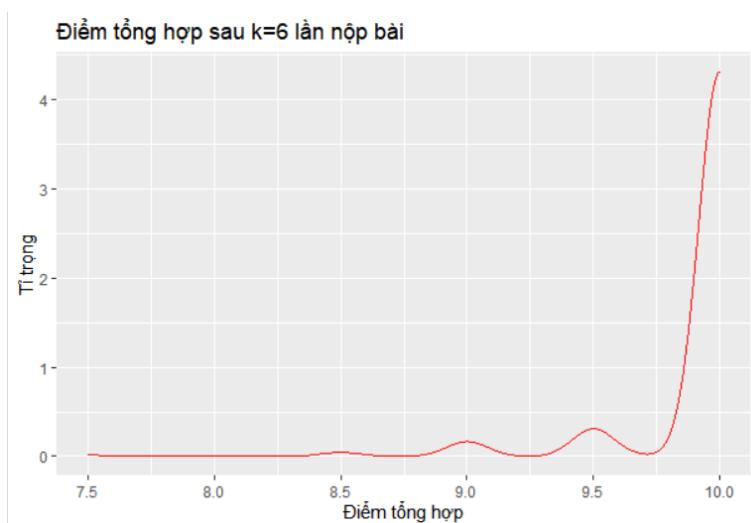
Output file 12:

	L1	L2	L3	L4	L5	L6	DiemCuoiCung
1	9.66	0	0	0	0	0	9.66
2	9.00	10	0	0	0	0	10.00
3	10.00	0	0	0	0	0	10.00
4	10.00	0	0	0	0	0	10.00
5	7.00	10	0	0	0	0	10.00
6	10.00	0	0	0	0	0	10.00
7	10.00	0	0	0	0	0	10.00
8	9.00	0	0	0	0	0	9.00
9	10.00	0	0	0	0	0	10.00
10	10.00	0	0	0	0	0	10.00
11	10.00	0	0	0	0	0	10.00
12	9.00	10	0	0	0	0	10.00
13	10.00	0	0	0	0	0	10.00
14	10.00	0	0	0	0	0	10.00
15	10.00	0	0	0	0	0	10.00
16	10.00	0	0	0	0	0	10.00
17	10.00	0	0	0	0	0	10.00
18	9.00	0	0	0	0	0	9.00

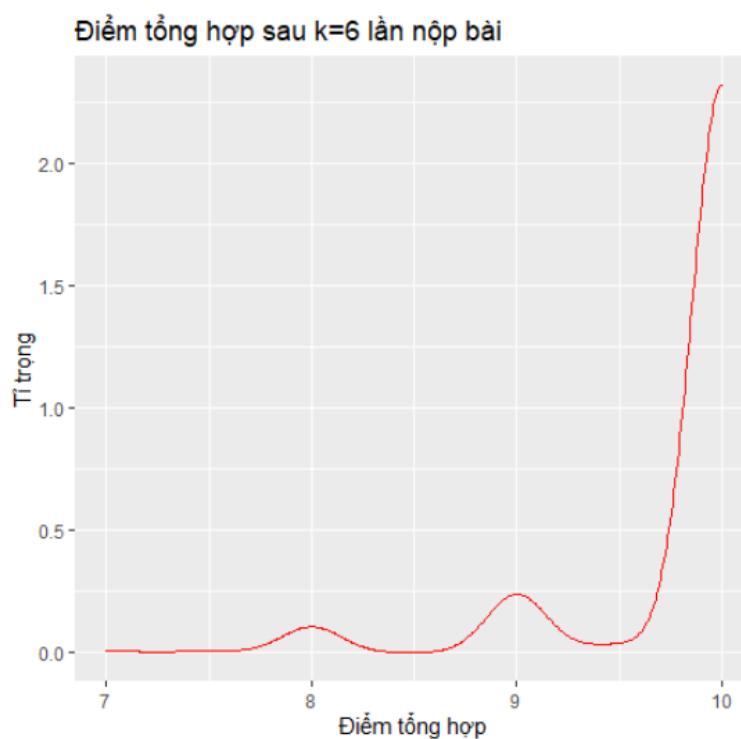
Ta dùng hàm ggplot để vẽ hàm phân bố điểm

```
p=ggplot(LanNop,aes(x=DiemCuoiCung)) + geom_density(alpha=0.5,col="red") +
  labs(tittle="Điểm tổng hợp sau k=6 lần nộp bài", x="Điểm tổng hợp", y="Tỉ trọng")
```

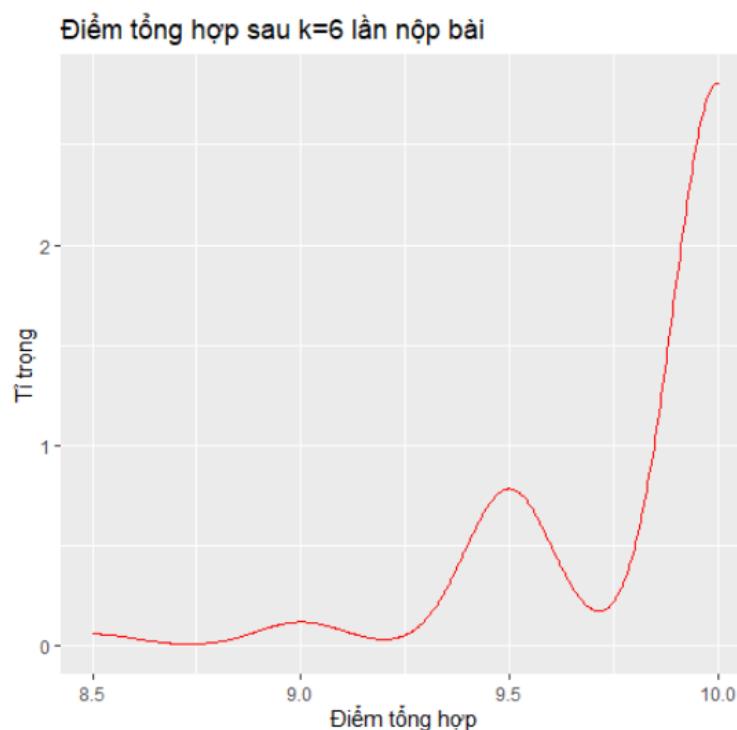
Output của file 6:



Output file 9:

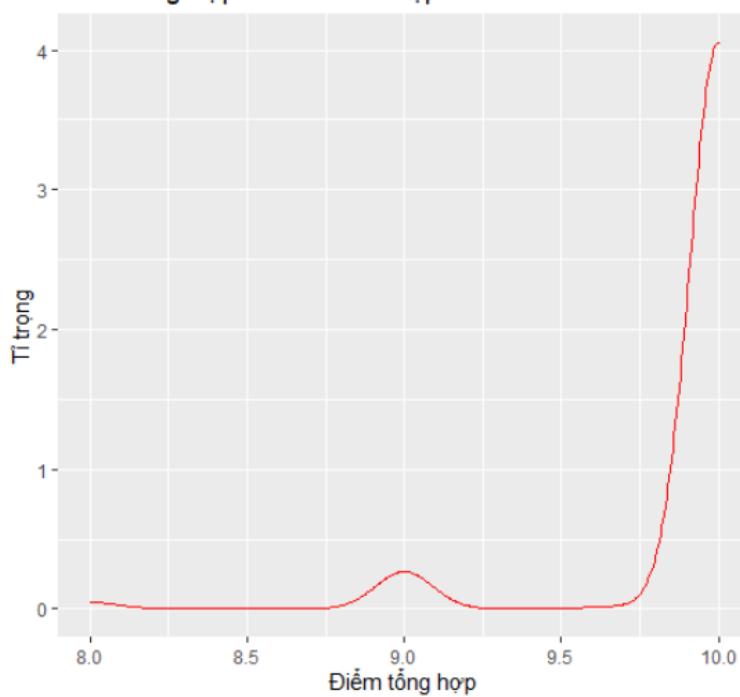


Output của file 10:



Output của file 12:

Điểm tổng hợp sau k=6 lần nộp bài



b) Với $k = 3$, ta sẽ lấy điểm từng phần của ba lần nộp bài đầu tiên.

```
k=3
LanNop2=data.frame(LanNop2$L1,LanNop2$L2,LanNop2$L3)
DiemCuoiCung2=c()
for(i in 1:n_sv){
  DiemCuoiCung2 <- c(DiemCuoiCung2, max(LanNop2[i,]))
}
LanNop2=data.frame(LanNop2,DiemCuoiCung2)
```

Output file 6:

	LanNop.L1	LanNop.L2	LanNop.L3	DiemCuoiCung2
1	9.61	0.0	0	9.61
2	10.00	0.0	0	10.00
3	7.50	9.0	0	9.00
4	10.00	0.0	0	10.00
5	9.00	10.0	0	10.00
6	8.50	0.0	0	8.50
7	9.50	10.0	0	10.00
8	10.00	0.0	0	10.00
9	9.00	0.0	0	9.00
10	9.50	10.0	0	10.00
11	10.00	0.0	0	10.00
12	10.00	0.0	0	10.00
13	10.00	0.0	0	10.00
14	10.00	0.0	0	10.00
15	9.50	10.0	0	10.00
16	10.00	0.0	0	10.00
17	9.00	10.0	0	10.00
18	10.00	0.0	0	10.00

Output file 9:

	LanNop.L1	LanNop.L2	LanNop.L3	DiemCuoiCung2
1	9.35	0.0	0	9.35
2	8.00	9.0	10	10.00
3	7.00	10.0	0	10.00
4	10.00	0.0	0	10.00
5	7.00	10.0	0	10.00
6	7.50	0.0	0	7.50
7	9.00	0.0	0	9.00
8	7.00	10.0	0	10.00
9	8.00	0.0	0	8.00
10	9.00	9.0	10	10.00
11	8.00	10.0	0	10.00
12	9.00	0.0	0	9.00
13	9.00	10.0	0	10.00
14	9.00	10.0	0	10.00
15	10.00	0.0	0	10.00
16	10.00	0.0	0	10.00
17	9.00	10.0	0	10.00
18	9.00	0.0	0	9.00

Output file 10:

	LanNop.L1	LanNop.L2	LanNop.L3	DiemCuoiCung2
1	9.54	0.0	0.0	9.54
2	9.00	9.5	10.0	10.00
3	9.00	9.5	10.0	10.00
4	9.00	0.0	0.0	9.00
5	8.00	10.0	0.0	10.00
6	8.50	0.0	0.0	8.50
7	9.00	10.0	0.0	10.00
8	10.00	0.0	0.0	10.00
9	9.50	9.5	0.0	9.50
10	9.50	10.0	0.0	10.00
11	8.00	10.0	0.0	10.00
12	9.50	10.0	0.0	10.00
13	8.50	9.5	10.0	10.00
14	10.00	0.0	0.0	10.00
15	10.00	0.0	0.0	10.00
16	9.50	9.5	10.0	10.00
17	9.50	0.0	0.0	9.50
18	9.50	0.0	0.0	9.50

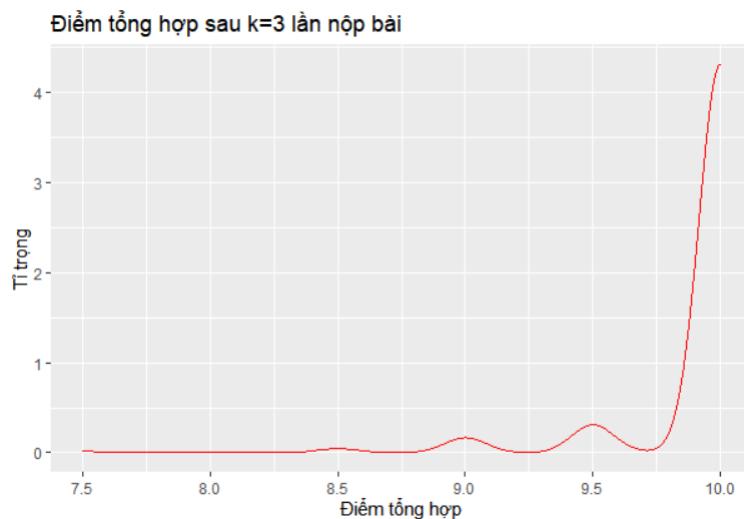
Output file 12:

	LanNop.L1	LanNop.L2	LanNop.L3	DiemCuoiCung2
1	9.66	0	0	9.66
2	9.00	10	0	10.00
3	10.00	0	0	10.00
4	10.00	0	0	10.00
5	7.00	10	0	10.00
6	10.00	0	0	10.00
7	10.00	0	0	10.00
8	9.00	0	0	9.00
9	10.00	0	0	10.00
10	10.00	0	0	10.00
11	10.00	0	0	10.00
12	9.00	10	0	10.00
13	10.00	0	0	10.00
14	10.00	0	0	10.00
15	10.00	0	0	10.00
16	10.00	0	0	10.00
17	10.00	0	0	10.00
18	9.00	0	0	9.00

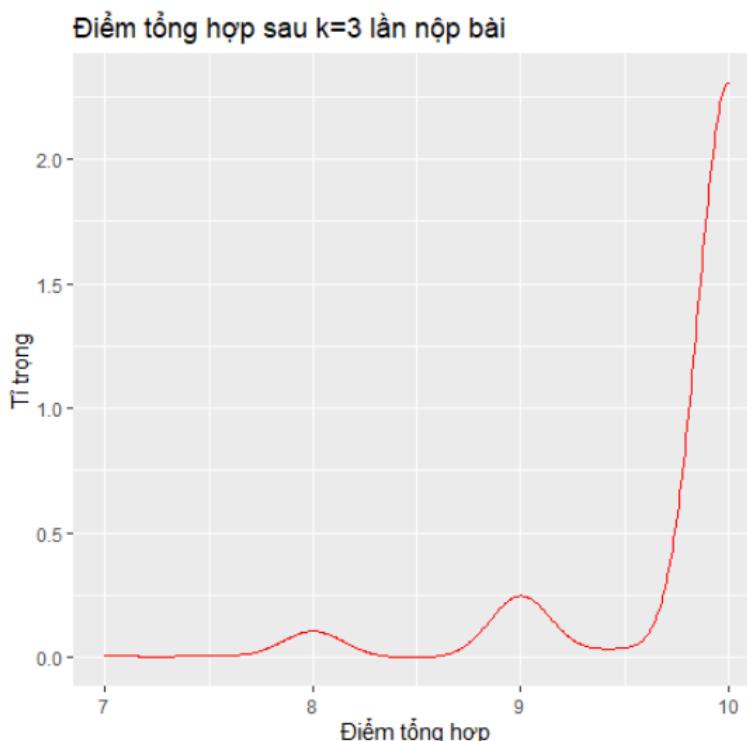
Ta dùng hàm ggplot để vẽ hàm phân bố điểm

```
p=ggplot(LanNop, aes(x=DiemCuoiCung)) + geom_density(alpha=0.5,col="red")
+ labs(title="Điểm tổng hợp sau k=3 lần nộp bài", x="Điểm tổng hợp", y="Tỉ
trọng")
```

Output file 6:

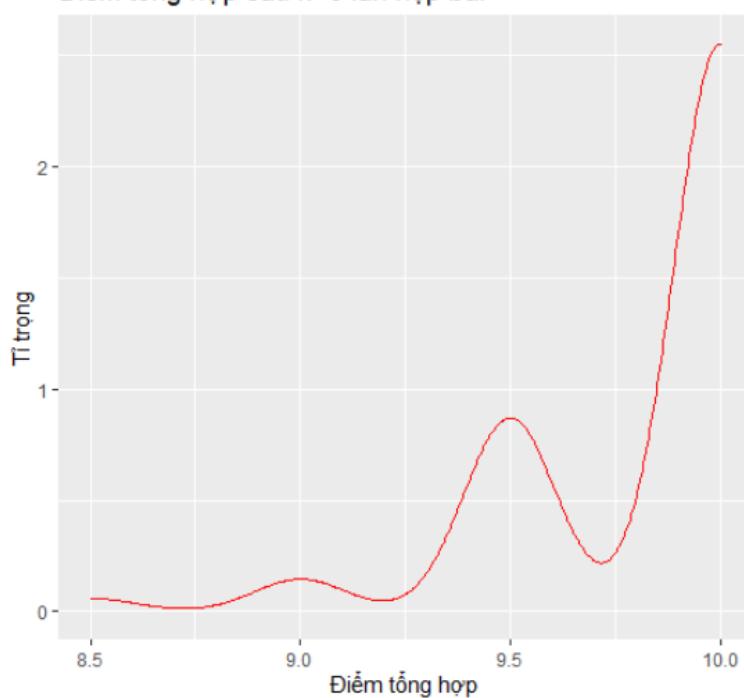


Output file 9:



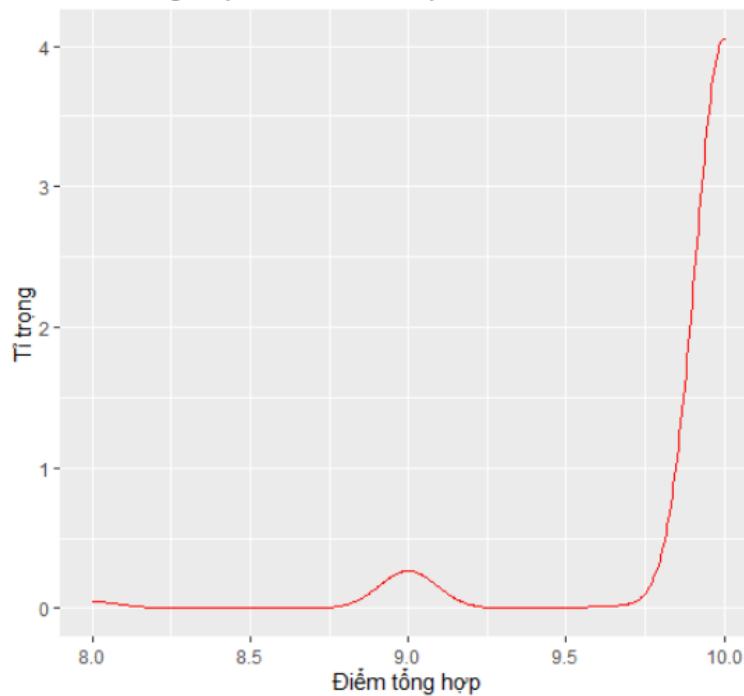
Output file 10:

Điểm tổng hợp sau k=3 lần nộp bài



Output file 12:

Điểm tổng hợp sau k=3 lần nộp bài



- c) Tính các giá trị TB_k vẽ biểu đồ thể hiện sự thay đổi của các giá trị trung bình này với sự thay đổi của k .
Chọn $k = 6$, ta lưu điểm trung bình sau mỗi lần nộp của tất cả sinh viên trên ma trận tbk. Dùng hàm mean để tính trung bình điểm mà các sinh viên đạt được

```
L=c(1,2,3)
tb1=mean(LanNop$L1)
tb2=mean(LanNop$L2)
tb3=mean(LanNop$L3)
tb4=mean(LanNop$L4)
tb5=mean(LanNop$L5)
tb6=mean(LanNop$L6)
tbk=c(tb1,tb2,tb3,tb4,tb5,tb6)
tbk=data.frame(L,tbk)
```

Output file 6:

	L	tb
1	1	9.4708385
2	2	2.9161491
3	3	0.6785714
4	4	0.0310559
5	5	0.0000000
6	6	0.0000000

Output file 9:

	L	tb
1	1	8.97019544
2	2	4.57491857
3	3	0.48534202
4	4	0.02931596
5	5	0.03257329
6	6	0.00000000

Output file 10:

	L	tb
1	1	9.276
2	2	6.400
3	3	2.376
4	4	0.388
5	5	0.156
6	6	0.080

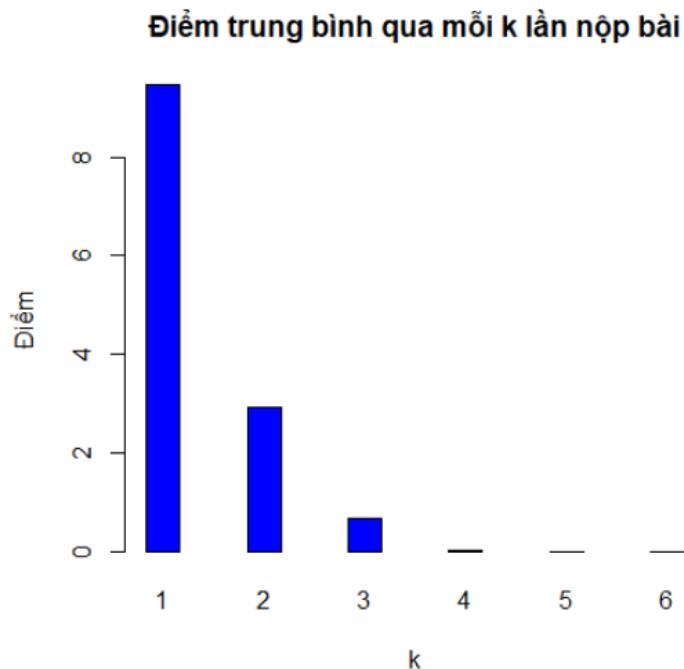
Output file 12:

L	tb
1	9.5845000
2	2.4892857
3	0.1035714
4	0.0000000
5	0.0000000
6	0.0000000

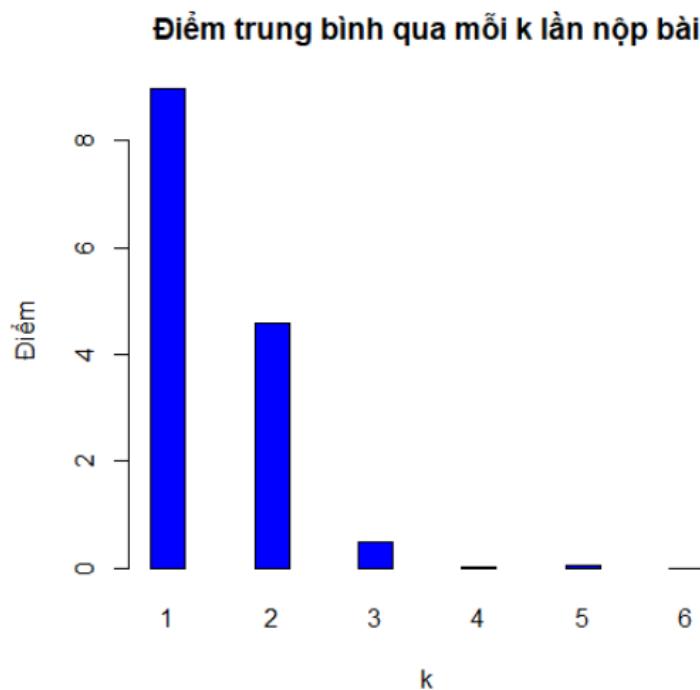
Vẽ biểu đồ

```
barplot(tbk$tb,width = 1,space = 2,main = "Điểm trung bình qua mỗi k lần nộp bài",names.arg = tbk$L, xlab = "k",ylab = "Điểm",col = "blue")
```

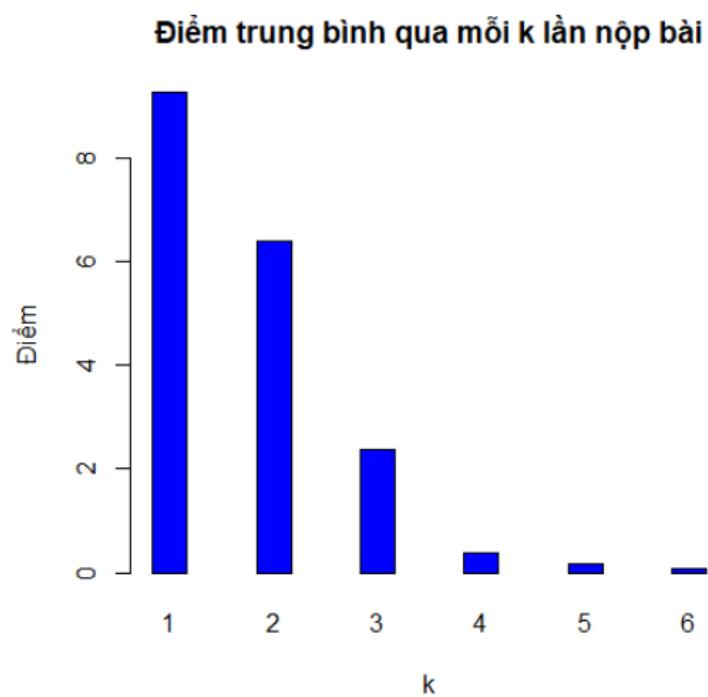
Output file 6:



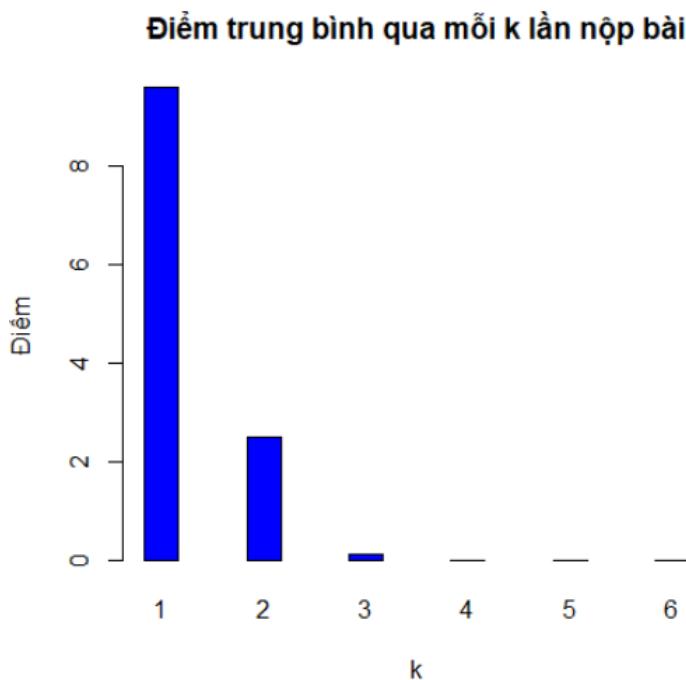
Output file 9:



Output file 10:



Output file 12:



Nhận xét: Điểm trung bình lần nộp đầu tiên cao nhất(khoảng 8.5 đến 9.5). Điểm trung bình các lần nộp còn lại khá thấp. Điểm trung bình của lần 6 gần bằng 0.

- d) Hãy cho biết trung bình điểm số mà các sinh viên đạt được qua bài tập tid_n này là bao nhiêu.

```
TrungBinhDiem=mean(LanNop$DiemCuoiCung)
```

Output file 6:

```
> TrungBinhDiem  
[1] 9.880776
```

Output file 9:

```
> TrungBinhDiem  
[1] 9.808958
```

Output file 10:

```
> TrungBinhDiem  
[1] 9.84
```

Output file 12:

```
> TrungBinhDiem  
[1] 9.917857
```

4.6 Câu 7: Nhóm câu hỏi liên quan sinh viên học đối phó

- a) Thời gian t_2 phù hợp để xác định sinh viên đối phó là sinh viên có thời gian nộp lần đầu sau $2/3$ khoảng thời gian giữa thời điểm sinh viên nộp lần đầu tiên và thời điểm sinh viên nộp lần cuối nộp bài.
- (a) Đầu tiên ta tạo một data frame là c để chứa dữ liệu từ file xlsx:

```

library(readxl)
library(utf8)
library(dplyr)
library(lubridate)
library("zoo")
a<-read_xlsx("filename.xlsx")
c=data.frame(a)
names(c)[1]<-"ID"
names(c)[2]<-"Status"
names(c)[3]<-"Stime"
names(c)[4]<-"Etime"
names(c)[5]<-"Time"
names(c)[6]<-"Diem"
names(c)[7]<-"Q1"
names(c)[8]<-"Q2"
names(c)[9]<-"Q3"
names(c)[10]<-"Q4"
names(c)[11]<-"Q5"
names(c)[12]<-"Q6"
names(c)[13]<-"Q7"
names(c)[14]<-"Q8"
names(c)[15]<-"Q9"
names(c)[16]<-"Q10"
c=subset(c,!is.na(c$ID))
c$ID=as.numeric(c$ID)
c$Etime=as.POSIXct(c$Etime, format = "%d %B %Y %I:%M %p")
c$Stime=as.POSIXct(c$Stime, format = "%d %B %Y %I:%M %p")

```

Kết quả ta có bảng c như sau: File 6:

ID	Status	Stime	Etime	Time	Diem	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
1	Đã hoàn thành	2020-03-29 09:30:00	2020-03-29 09:30:00	0 phút 0 giây	9,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	0,00
2	Đã hoàn thành	2020-03-29 09:25:00	2020-03-29 09:28:00	2 phút 52 giây	10,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
3	Đã hoàn thành	2020-03-29 09:35:00	2020-03-29 09:40:00	4 phút 53 giây	10,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
4	Đã hoàn thành	2020-03-29 10:10:00	2020-03-29 10:13:00	3 phút	10,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
5	Đã hoàn thành	2020-03-29 11:02:00	2020-03-29 11:17:00	14 phút 57 giây	10,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
6	Đã hoàn thành	2020-03-29 11:26:00	2020-03-29 11:29:00	3 phút 10 giây	10,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
7	Đã hoàn thành	2020-03-29 11:51:00	2020-03-29 11:54:00	3 phút 4 giây	10,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
8	Đã hoàn thành	2020-03-29 14:14:00	2020-03-29 15:44:00	3 phút 32 giây	10,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
9	Đã hoàn thành	2020-03-29 15:39:00	2020-03-29 15:44:00	5 phút 29 giây	10,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
10	Đã hoàn thành	2020-03-29 17:07:00	2020-03-29 17:50:00	8 phút 22 giây	9,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	0,00

Showing 1 to 12 of 441 entries, 16 total columns

File 9:

ID	Status	Stime	Etime	Time	Diem	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
1	Đã hoàn thành	2020-04-04 09:00:00	2020-04-04 09:07:00	6 phút 14 giây	9,00	1,00	0,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
2	Đã hoàn thành	2020-04-04 09:09:00	2020-04-04 09:17:00	8 phút 32 giây	10,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
3	Đã hoàn thành	2020-04-04 09:24:00	2020-04-04 09:32:00	8 phút 31 giây	10,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
4	Đã hoàn thành	2020-04-04 09:32:00	2020-04-04 09:40:00	7 phút 57 giây	9,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	0,00	1,00
5	Đã hoàn thành	2020-04-04 09:41:00	2020-04-04 09:42:00	1 phút 4 giây	10,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
6	Đã hoàn thành	2020-04-04 09:55:00	2020-04-04 10:02:00	7 phút 11 giây	10,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00

Showing 1 to 8 of 466 entries, 16 total columns

File 10:

ID	Status	Stime	Etime	Time	Diem	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
1	Đã hoàn thành	2020-04-06 09:36:00	2020-04-06 09:38:00	2 phút 16 giây	8,50	1,00	1,00	1,00	1,00	1,00	1,00	1,00	0,50	1,00	0,00
2	Đã hoàn thành	2020-04-06 09:39:00	2020-04-06 09:40:00	45 giây	9,50	1,00	1,00	1,00	1,00	1,00	1,00	1,00	0,50	1,00	1,00
3	Đã hoàn thành	2020-04-06 09:40:00	2020-04-06 09:44:00	4 phút 10 giây	10,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
4	Đã hoàn thành	2020-04-06 09:40:00	2020-04-06 09:41:00	42 giây	10,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
5	Đã hoàn thành	2020-04-06 09:54:00	2020-04-06 09:58:00	3 phút 35 giây	9,50	1,00	1,00	1,00	1,00	1,00	1,00	1,00	0,50	1,00	1,00
6	Đã hoàn thành	2020-04-06 09:59:00	2020-04-06 10:01:00	2 phút 48 giây	10,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
7	Đã hoàn thành	2020-04-06 10:34:00	2020-04-06 10:39:00	4 phút 16 giây	9,50	1,00	1,00	1,00	1,00	1,00	1,00	1,00	0,50	1,00	1,00
8	Đã hoàn thành	2020-04-06 10:39:00	2020-04-06 10:40:00	38 giây	10,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00

Showing 1 to 10 of 551 entries, 16 total columns

File 12:

ID	Status	Stime	Time	Time	Diem	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
1	Đã hoàn thành	2020-04-12 08:38:00	2020-04-12 08:42:00	4 phút 38 giây	9,00	1,00	1,00	1,00	0,00	1,00	1,00	1,00	1,00	1,00	1,00
2	Đã hoàn thành	2020-04-12 08:44:00	2020-04-12 08:45:00	1 phút	9,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
3	Đã hoàn thành	2020-04-12 08:52:00	2020-04-12 08:53:00	1 phút 4 giây	9,00	1,00	1,00	1,00	0,00	1,00	1,00	1,00	1,00	1,00	1,00
4	Đã hoàn thành	2020-04-12 09:20:00	2020-04-12 09:23:00	2 phút 34 giây	10,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
5	Đã hoàn thành	2020-04-12 09:31:00	2020-04-12 09:33:00	1 phút 47 giây	9,00	1,00	1,00	1,00	0,00	1,00	1,00	1,00	1,00	1,00	1,00
6	Đã hoàn thành	2020-04-12 09:38:00	2020-04-12 09:39:00	10 giây	10,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
7	Đã hoàn thành	2020-04-12 10:12:00	2020-04-12 10:13:00	1 phút 27 giây	10,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
8	Đã hoàn thành	2020-04-12 10:46:00	2020-04-12 10:50:00	4 phút	10,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00

Showing 1 to 10 of 353 entries, 16 total columns

(b) Ta tạo vector ID để lưu lại ID của tất cả các sinh viên có trong c:

```
ID=as.factor(c$ID)
ID <- ID[!is.na(ID)]
ID=levels(ID)
ID=as.numeric(ID)
```

Kết quả thu được như sau:

File 6:

```
> ID
[1] 1511191 1613010 1812257 1812477 1812478 1813096 1813528 1813681 1814096 1814518 1814611 1820028 1852443 1910006
[15] 1910032 1910038 1910060 1910076 1910094 1910101 1910110 1910113 1910123 1910137 1910198 1910202 1910224 1910238
[29] 1910265 1910276 1910276 1910298 1910339 1910346 1910347 1910351 1910402 1910409 1910473 1910563 1910565 1910620 1910643
[43] 1910644 1910650 1910663 1910666 1910735 1910865 1910892 1910916 1910984 1911000 1911015 1911044 1911056 1911058
[57] 1911066 1911105 1911110 1911134 1911185 1911186 1911207 1911217 1911262 1911283 1911285 1911296 1911314 1911363
[71] 1911441 1911451 1911478 1911520 1911530 1911561 1911569 1911591 1911594 1911650 1911704 1911736 1911796
[85] 1911837 1911841 1911878 1911881 1911900 1911907 1911931 1911975 1912041 1912046 1912054 1912084 1912123 1912184
[99] 1912190 1912237 1912267 1912284 1912371 1912384 1912386 1912410 1912457 1912463 1912522 1912523 1912526 1912539
[113] 1912579 1912591 1912602 1912675 1912676 1912683 1912700 1912705 1912713 1912715 1912749 1912761 1912798
[127] 1912811 1912817 1912912 1912916 1912954 1912958 1912966 1912980 1913014 1913021 1913026 1913040 1913045
[141] 1913048 1913075 1913094 1913102 1913114 1913123 1913167 1913186 1913218 1913241 1913254 1913260 1913261
[155] 1913268 1913303 1913334 1913336 1913341 1913354 1913356 1913358 1913380 1913386 1913391 1913418 1913419 1913424
[169] 1913430 1913433 1913446 1913447 1913467 1913468 1913469 1913471 1913472 1913473 1913476 1913477 1913478 1913479
[183] 1913678 1913695 1913713 1913729 1913756 1913758 1913763 1913764 1913775 1913817 1913828 1913832 1913844 1913917
[197] 1913918 1913948 1913949 1913950 1914003 1914011 1914022 1914038 1914047 1914052 1914054 1914055 1914064 1914078
[211] 1914079 1914084 1914093 1914121 1914126 1914210 1914220 1914227 1914232 1914239 1914310 1914316 1914352 1914384
[225] 1914405 1914424 1914427 1914447 1914461 1914465 1914469 1914476 1914477 1914481 1914485 1914486 1914487 1914488
[239] 1914704 1914713 1914720 1914738 1914763 1914802 1914806 1914807 1914808 1914810 1914812 1914813 1914814
[253] 1914881 1914900 1914914 1914917 1914921 1914924 1914927 1914930 1914933 1914936 1914939 1914942 1914945 1914946
[267] 1915161 1915251 1915256 1915275 1915294 1915304 1915329 1915350 1915351 1915352 1915353 1915354 1915355 1915356
[281] 1915474 1915482 1915486 1915520 1915540 1915541 1915551 1915557 1915562 1915567 1915598 1915650 1915651 1915656
[295] 1915667 1915745 1915775 1915787 1915795 1915822 1915865 1915866 1915882 1915903 1915909 1915919 1915928 1915931
[309] 1915939 1915940 1915953 1915982 1915983 1915988 1915991 1916022 1916061 1927007 1936024 1937019 1937024 1937064
```

File 9:

```
> ID
[1] 1511191 1613010 1812257 1812477 1812478 1813096 1813528 1813681 1814096 1814518 1814611 1820028 1852443 1910006
[15] 1910060 1910076 1910094 1910101 1910113 1910123 1910137 1910198 1910202 1910224 1910238
[31] 1910339 1910346 1910347 1910351 1910402 1910409 1910473 1910563 1910643 1910650 1910663 1910666 1910735
[46] 1910865 1910892 1910916 1910984 1911000 1911015 1911040 1911056 1911058 1911066 1911105 1911110 1911136 1911185
[61] 1911207 1911217 1911262 1911283 1911296 1911314 1911363 1911441 1911446 1911456 1911478 1911520 1911530 1911561 1911565
[76] 1911569 1911591 1911594 1911650 1911704 1911736 1911796 1911837 1911841 1911878 1911881 1911900 1911907 1911931 1911975
[91] 1912041 1912046 1912056 1912084 1912123 1912184 1912237 1912267 1912288 1912384 1912410 1912457
[106] 1912463 1912522 1912523 1912526 1912539 1912579 1912594 1912602 1912675 1912683 1912700 1912705 1912713 1912715
[121] 1912749 1912761 1912798 1912811 1912817 1912912 1912916 1912954 1912980 1913014 1913021 1913026 1913040 1913045
[136] 1913048 1913075 1913094 1913102 1913114 1913123 1913136 1913167 1913186 1913218 1913241 1913254 1913260 1913261 1913268
[151] 1913334 1913336 1913341 1913354 1913355 1913356 1913380 1913386 1913398 1913418 1913419 1913424 1913433 1913446
[166] 1913457 1913464 1913467 1913468 1913469 1913470 1913471 1913472 1913473 1913474 1913475 1913476 1913477 1913478 1913479
[181] 1913763 1913764 1913775 1913817 1913828 1913832 1913844 1913918 1913949 1913949 1913950 1914003 1914011 1914022 1914038
[196] 1914047 1914052 1914054 1914055 1914056 1914057 1914058 1914059 1914060 1914061 1914062 1914063 1914064 1914065 1914066
[211] 1914291 1914310 1914316 1914352 1914384 1914405 1914424 1914472 1914477 1914481 1914485 1914486 1914487 1914488 1914489
[226] 1914898 1914904 1914914 1914917 1914921 1914924 1914927 1914930 1914933 1914936 1914939 1914942 1914945 1914946 1914947
[241] 1915128 1915148 1915150 1915151 1915152 1915153 1915154 1915155 1915156 1915157 1915158 1915159 1915160 1915161 1915162
[256] 1915268 1915275 1915294 1915304 1915329 1915350 1915357 1915378 1915394 1915401 1915404 1915407 1915409 1915412 1915415
[271] 1915352 1915354 1915355 1915356 1915357 1915358 1915359 1915360 1915361 1915362 1915363 1915364 1915365 1915366 1915367
[286] 1915865 1915866 1915882 1915903 1915919 1915931 1915932 1915933 1915934 1915935 1915936 1915937 1915938 1915939 1915940
[301] 1916061 1927007 1936024 1937019 1937024 1937064
```

File 10:

```
> ID
[1] 1511191 1613010 1812257 1812477 1812478 1813096 1813528 1813681 1814096 1814518 1814611 1820028 1852443 1910006
[17] 1910094 1910101 1910113 1910123 1910137 1910198 1910202 1910224 1910238
[33] 1910402 1910409 1910413 1910563 1910620 1910643 1910644 1910650 1910663 1910685 1910892 1910916 1910926 1911000
[49] 1911044 1911056 1911058 1911066 1911105 1911110 1911136 1911185 1911186 1911207 1911217 1911262 1911283 1911285 1911296 1911314
[65] 1911363 1911441 1911478 1911520 1911530 1911561 1911565 1911569 1911591 1911594 1911650 1911704 1911736 1911796 1911837 1911841
[81] 1911846 1911851 1911852 1911853 1911854 1911855 1911856 1911857 1911858 1911859 1911860 1911861 1911862 1911863 1911864 1911865
[97] 1912386 1912410 1912457 1912463 1912523 1912539 1912579 1912594 1912602 1912675 1912683 1912700 1912705 1912713 1912715
[113] 1912715 1912749 1912761 1912798 1912811 1912817 1912912 1912916 1912954 1912980 1913014 1913022 1913026 1913040 1913045
[129] 1913048 1913075 1913094 1913102 1913114 1913123 1913167 1913241 1913242 1913254 1913261 1913268 1913306 1913334 1913336
[145] 1913341 1913354 1913355 1913356 1913380 1913386 1913396 1913418 1913419 1913424 1913430 1913433 1913446 1913457 1913464
[161] 1913560 1913566 1913599 1913609 1913621 1913629 1913651 1913652 1913678 1913693 1913713 1913729 1913763 1913764 1913775 1913817
[177] 1913824 1913832 1913844 1913918 1913944 1913949 1914003 1914012 1914022 1914038 1914047 1914052 1914054 1914078 1914079
[193] 1914084 1914093 1914121 1914126 1914210 1914227 1914233 1914291 1914310 1914316 1914352 1914384 1914405 1914424 1914472
[209] 1914474 1914477 1914481 1914484 1914487 1914491 1914494 1914497 1914500 1914503 1914506 1914509 1914512 1914515 1914518 1914521
[225] 1914524 1914528 1914532 1914536 1914539 1914542 1914545 1914548 1914551 1914554 1914557 1914560 1914563 1914566 1914569 1914572
[241] 1915146 1915251 1915258 1915264 1915263 1915269 1915329 1915330 1915331 1915332 1915333 1915334 1915335 1915336 1915337 1915338
[257] 1915486 1915520 1915540 1915541 1915551 1915557 1915562 1915570 1915578 1915598 1915651 1915656 1915745 1915775 1915787 1915795 1915822
[273] 1915865 1915866 1915882 1915903 1915919 1915931 1915932 1915933 1915934 1915935 1915936 1915937 1915938 1915939 1915940 1915941
[289] 1915904 1915905 1915906 1915907 1915908 1915909 1915910 1915911 1915912 1915913 1915914 1915915 1915916 1915917 1915918 1915919
```

File 12:

```
> ID
[1] 1511191 1613010 1812257 1812478 1813096 1813681 1814096 1814518 1820028 1852443 1910009 1910032 1910038
[15] 1910060 1910076 1910094 1910101 1910113 1910123 1910137 1910203 1910224 1910238 1910265 1910276 1910298
[29] 1910339 1910346 1910347 1910351 1910409 1910473 1910565 1910620 1910643 1910644 1910650 1910663 1910666
[43] 1910865 1910892 1910916 1910984 1911000 1911056 1911058 1911061 1911100 1911136 1911185 1911186 1911207
[57] 1911217 1911262 1911283 1911285 1911296 1911314 1911363 1911441 1911478 1911520 1911530 1911565 1911569 1911591
[71] 1911594 1911650 1911704 1911736 1911796 1911837 1911841 1911881 1911908 1911907 1911931 1912041 1912046 1912056
[85] 1912084 1912123 1912184 1912190 1912237 1912288 1912371 1912384 1912386 1912410 1912457 1912463 1912523
[99] 1912539 1912579 1912594 1912602 1912675 1912677 1912683 1912700 1912713 1912749 1912761 1912798 1912811
[113] 1912817 1912912 1912916 1912966 1912980 1913014 1913021 1913026 1913040 1913045 1913048 1913075 1913094 1913102
[127] 1913114 1913123 1913167 1913186 1913218 1913241 1913259 1913261 1913266 1913306 1913334 1913338 1913341 1913354
[141] 1913355 1913356 1913380 1913386 1913396 1913418 1913419 1913424 1913430 1913433 1913446 1913457 1913464 1913467
[155] 1913560 1913566 1913599 1913609 1913621 1913651 1913652 1913678 1913695 1913713 1913729 1913763 1913764 1913775
[169] 1913817 1913828 1913832 1913844 1913918 1913944 1913949 1914003 1914011 1914022 1914038 1914047 1914052 1914054
[183] 1914055 1914078 1914079 1914084 1914093 1914121 1914126 1914210 1914220 1914227 1914232 1914291 1914310 1914316
[197] 1914352 1914384 1914405 1914424 1914472 1914474 1914477 1914479 1914481 19144651 19144659 19144661 19144674 19144677 19144685
[211] 1914698 1914704 1914713 1914720 1914738 1914763 1914768 1914807 1914830 1914845 1914864 1914880 1914881 1914914
[225] 1914979 1915016 1915040 1915063 1915071 1915076 1915083 1915130 1915133 1915146 1915251 1915268 1915275 1915294
[239] 1915323 1915329 1915330 1915351 1915378 1915439 1915442 1915473 1915474 1915482 1915486 1915520 1915540 1915541
[253] 1915551 1915557 1915562 1915570 1915598 1915651 1915656 1915745 1915775 1915787 1915795 1915822 1915865 1915866
[267] 1915882 1915903 1915919 1915953 1915982 1915983 1915988 1915991 1916022 1916061 1927007 1936024 1937024 1937064
```

- (c) Tao vecto solannop để ghi nhận số lần nộp bài của từng ID rồi tạo một data frame có tên là data để chứa ID và số lần nộp bài tương ứng:

```
solannop=rep(0,times=length(ID))
for(i in 1:length(ID)){
  for(j in 1:length(c$ID)){
    if(c$ID[j] == ID[i]) {solannop[i]=solannop[i]+1}
  }
}
data=data.frame(ID,solannop)
```

Kết quả thu được như sau: File 6:

	ID	solannop
1	1511191	1
2	1613010	2
3	1812257	1
4	1812477	2
5	1812478	1
6	1813096	2
7	1813528	1
8	1813681	1
9	1814096	2

Showing 1 to 10 of 322 entries, 2 total columns

File 9:

	ID	solannop
1	15111191	3
2	1613010	2
3	1812257	1
4	1812477	2
5	1812478	1
6	1813096	1
7	1813528	2
8	1813681	1
9	1814096	3

Showing 1 to 10 of 306 entries, 2 total columns

File 10:

	ID	solannop
1	15111191	3
2	1613010	3
3	1812257	1
4	1812477	2
5	1812478	1
6	1813096	2
7	1813681	1
8	1814096	2
9	1814518	2

Showing 1 to 10 of 289 entries, 2 total columns

File 12:

ID	solannop
1 1511191	2
2 1613010	1
3 1812257	1
4 1812477	2
5 1812478	1
6 1813096	1
7 1813681	1
8 1814096	1

Showing 1 to 9 of 280 entries, 2 total columns

- (d) Ta tìm t2 là 2/3 khoảng thời gian giữa thời điểm sinh viên đầu tiên và thời điểm sinh viên cuối cùng nộp bài:

```
tmax=max(c$Etime,na.rm = T)
tmin=min(c$Etime,na.rm = T)
hours=as.period(interval(tmin, tmax), unit = "hours")
sociocongthem=hour(hours)-round(hour(hours)/3)
t2=tmin+hours(sociocongthem)
```

Kết quả thu được như sau: File 6:

```
> t2
[1] "2020-05-01 21:05:00 +07"
```

File 9:

```
> t2
[1] "2020-05-04 02:07:00 +07"
```

File 10:

```
> t2
[1] "2020-05-04 17:38:00 +07"
```

File 12:

```
> t2
[1] "2020-05-06 17:42:00 +07"
```

- (e) Tạo vecto landaunop chứa thời điểm lần đầu nộp bài của từng ID rồi thêm vào data một cột là landaunop với dữ liệu trong vecto landaunop:

```
b=subset(c,Etime>t2)
landaunop=c()
for(i in 1:length(ID)){
  for(j in 1:length(c$ID)){
    if(ID[i]==c$ID[j]){
      landaunop[i]=c$Etime[j]
      break
    }
  }
}
landaunop=as.POSIXct(landaunop,tz = "GMT",origin,format = "%d ")
data=cbind(data,landaunop)
```

Kết quả thu được như sau:

File 6:

	ID	solannop	landaunop
1	1511191	1	2020-04-04 08:19:00
2	1613010	2	2020-04-19 16:32:00
3	1812257	1	2020-04-20 06:19:00
4	1812477	2	2020-04-23 00:44:00
5	1812478	1	2020-04-06 09:59:00
6	1813096	2	2020-04-12 08:17:00
7	1813528	1	2020-05-05 23:14:00
8	1813681	1	2020-05-11 06:57:00

Showing 1 to 10 of 322 entries, 3 total columns

File 9:

	ID	solannop	landaunop
1	1511191	3	2020-04-04 13:16:00
2	1613010	2	2020-04-19 16:38:00
3	1812257	1	2020-04-20 06:52:00
4	1812477	2	2020-04-24 07:17:00
5	1812478	1	2020-04-06 10:41:00
6	1813096	1	2020-04-13 03:08:00
7	1813528	2	2020-05-05 23:41:00
8	1813681	1	2020-05-18 01:37:00

Showing 1 to 10 of 306 entries, 3 total columns

File 10:



	ID	solannop	landaunop
1	1511191	3	2020-04-20 06:06:00
2	1613010	3	2020-04-27 10:09:00
3	1812257	1	2020-04-20 10:32:00
4	1812477	2	2020-04-24 11:51:00
5	1812478	1	2020-04-06 10:43:00
6	1813096	2	2020-05-06 02:50:00
7	1813681	1	2020-05-18 01:52:00
8	1814096	2	2020-04-15 07:41:00

Showing 1 to 10 of 289 entries, 3 total columns

File 12:

	ID	solannop	landaunop
1	1511191	2	2020-04-20 06:25:00
2	1613010	1	2020-04-27 10:18:00
3	1812257	1	2020-04-20 10:45:00
4	1812477	2	2020-04-24 13:05:00
5	1812478	1	2020-04-13 05:33:00
6	1813096	1	2020-05-06 04:38:00
7	1813681	1	2020-05-18 02:42:00

Showing 1 to 9 of 280 entries, 3 total columns

(f) Tạo data7 là một data frame con của data chứa dữ liệu của những ID có thời gian lần đầu nộp sau t2:

```
data7=subset(data, landaunop>t2)
```

Kết quả thu được data7 như sau:

File 6:



	ID	solanop	landaunop
7	1813528	1	2020-05-05 23:14:00
8	1813681	1	2020-05-11 06:57:00
12	1820028	1	2020-05-06 18:34:00
14	1910006	2	2020-05-04 16:27:00
17	1910060	1	2020-05-11 13:48:00
25	1910198	1	2020-05-04 05:42:00
29	1910265	1	2020-05-07 14:54:00
39	1910563	1	2020-05-17 09:48:00

Showing 1 to 10 of 46 entries, 3 total columns

File 9:

	ID	solanop	landaunop
7	1813528	2	2020-05-05 23:41:00
8	1813681	1	2020-05-18 01:37:00
11	1820028	1	2020-05-06 18:52:00
13	1910006	2	2020-05-11 13:45:00
16	1910060	2	2020-05-11 14:16:00
24	1910198	2	2020-05-04 07:24:00
28	1910265	3	2020-05-07 15:22:00
32	1910346	1	2020-05-10 07:45:00

Showing 1 to 10 of 58 entries, 3 total columns

File 10:

	ID	solanop	landaunop
6	1813096	2	2020-05-06 02:50:00
7	1813681	1	2020-05-18 01:52:00
10	1820028	2	2020-05-06 18:55:00
11	1852443	2	2020-05-08 11:42:00
12	1910006	3	2020-05-11 13:48:00
15	1910060	3	2020-05-11 14:19:00
24	1910224	1	2020-05-12 09:27:00
26	1910265	3	2020-05-17 03:23:00

Showing 1 to 10 of 70 entries, 3 total columns

File 12:

	ID	solanop	landaunop
1	1511191	2	2020-04-20 06:25:00
2	1613010	1	2020-04-27 10:18:00
3	1812257	1	2020-04-20 10:45:00
4	1812477	2	2020-04-24 13:05:00
5	1812478	1	2020-04-13 05:33:00
6	1813096	1	2020-05-06 04:38:00
7	1813681	1	2020-05-18 02:42:00

Showing 1 to 9 of 280 entries, 3 total columns

(g) Thêm vào data7 cột điểm của từng ID:



```
for(i in 1:length(c$Diem)){
if(!is.na(c$Diem[i])){
if(c$Diem[i]=='-'){c$Diem[i]=NA}
}
}
c$Diem=scan(text=c$Diem, dec=",", sep=".")
diem=c()
for(i in 1:length(data7$ID)){
k=0
for(j in 1:length(c$ID)){
if(data7$ID[i]==c$ID[j]){
if(!is.na(c$Diem[j])){
if(k==0){
k=k+1
diem[i]=c$Diem[j]
}
else{
if(c$Diem[j]>diem[i]){diem[i]=c$Diem[j]}
}
}
}
}
}
data7=cbind(data7,diem)
data7=filter(data7,!is.na(diem))
```

Kết quả thu được data7 như sau: File 6:

	ID	solannop	landaunop	diem
1	1813528	1	2020-05-05 23:14:00	10.0
2	1813681	1	2020-05-11 06:57:00	9.0
3	1820028	1	2020-05-06 18:34:00	10.0
4	1910006	2	2020-05-04 16:27:00	10.0
5	1910060	1	2020-05-11 13:48:00	10.0
6	1910198	1	2020-05-04 05:42:00	10.0
7	1910265	1	2020-05-07 14:54:00	10.0
8	1910563	1	2020-05-17 09:48:00	10.0

Showing 1 to 10 of 46 entries, 4 total columns

File 9:



	ID	solannop	landaunop	diem
1	1813528	2	2020-05-05 23:41:00	10
2	1813681	1	2020-05-18 01:37:00	8
3	1820028	1	2020-05-06 18:52:00	9
4	1910006	2	2020-05-11 13:45:00	10
5	1910060	2	2020-05-11 14:16:00	10
6	1910198	2	2020-05-04 07:24:00	10
7	1910265	3	2020-05-07 15:22:00	10
8	1910346	1	2020-05-10 07:45:00	10

Showing 1 to 10 of 58 entries, 4 total columns

File 10:

	ID	solannop	landaunop	diem
1	1813096	2	2020-05-06 02:50:00	10.0
2	1813681	1	2020-05-18 01:52:00	10.0
3	1820028	2	2020-05-06 18:55:00	10.0
4	1852443	2	2020-05-08 11:42:00	10.0
5	1910006	3	2020-05-11 13:48:00	10.0
6	1910060	3	2020-05-11 14:19:00	10.0
7	1910224	1	2020-05-12 09:27:00	9.5
8	1910265	3	2020-05-17 03:23:00	10.0

Showing 1 to 10 of 70 entries, 4 total columns

File 12:

	ID	solannop	landaunop	diem
1	1813681	1	2020-05-18 02:42:00	9
2	1820028	1	2020-05-06 18:59:00	10
3	1852443	2	2020-05-08 12:01:00	10
4	1910006	1	2020-05-11 13:59:00	10
5	1910060	1	2020-05-11 14:31:00	10
6	1910076	1	2020-05-08 14:05:00	10
7	1910224	1	2020-05-14 08:05:00	10

Showing 1 to 9 of 57 entries, 4 total columns

- b) Số lượng sinh viên đón phó là số ID có trong data7:

```
sosvdoiphon = length(data7$ID)
```

Kết quả thu được data7 như sau: File 6:

```
> sosvdoiph  
[1] 46
```

File 9:

```
|> sosvdoiph  
|[1] 58
```

File 10:

```
> sosvdoiph  
[1] 70
```

File 12:

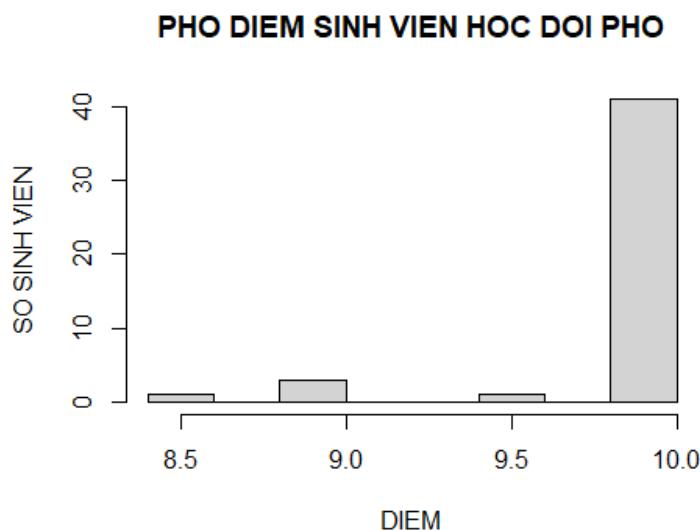
```
|> sosvdoiph  
|[1] 57
```

c) Biểu đồ thể hiện phô điểm của sinh viên học đồi phó với dữ liệu được lấy từ data7 như sau:

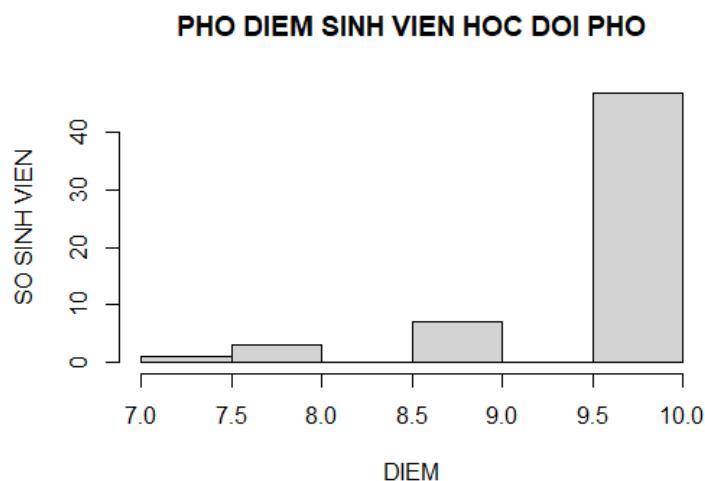
```
hist(data7$diem,main="PHO DIEM SINH VIEN HOC DOI PHO", xlab="DIEM",ylab = "SO SINH VIEN")
```

Kết quả thu được như sau:

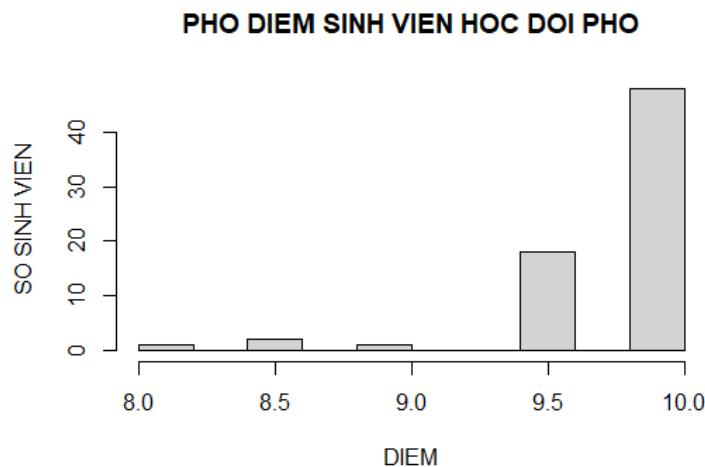
File 6:



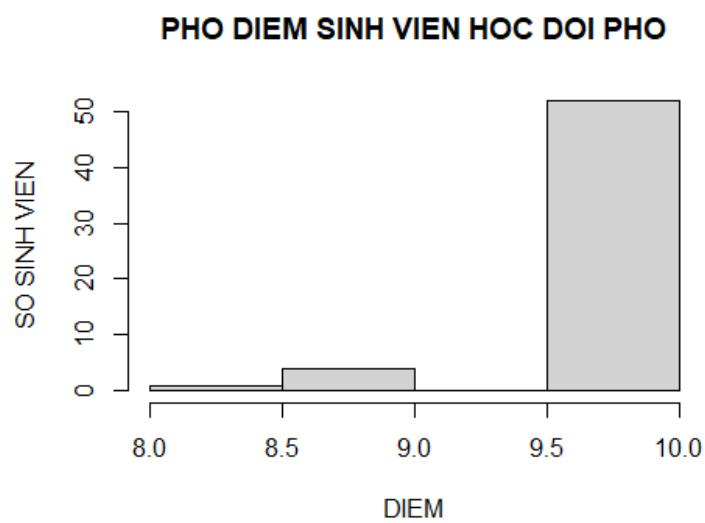
File 9:



File 10:



File 12:



4.7 Câu 9: Nhóm câu hỏi liên quan sinh viên thông minh

a) Sinh viên thông minh sẽ là sinh viên có điểm lớn hơn hoặc bằng 8 ngay từ 2 lần nộp đầu tiên.

(a) Đầu tiên ta tạo một data frame là c để chứa dữ liệu từ file xlsx:

```
library(readxl)
library(utf8)
library(dplyr)
library(lubridate)
library("zoo")
a<-read_xlsx("filename.xlsx")
c=data.frame(a)
names(c)[1]<-"ID"
names(c)[2]<-"Status"
names(c)[3]<-"Stime"
names(c)[4]<-"Etime"
names(c)[5]<-"Time"
names(c)[6]<-"Diem"
names(c)[7]<-"Q1"
names(c)[8]<-"Q2"
names(c)[9]<-"Q3"
names(c)[10]<-"Q4"
names(c)[11]<-"Q5"
names(c)[12]<-"Q6"
names(c)[13]<-"Q7"
names(c)[14]<-"Q8"
names(c)[15]<-"Q9"
names(c)[16]<-"Q10"
c=subset(c,!is.na(c$ID))
c$ID=as.numeric(c$ID)
c$Etime=as.POSIXct(c$Etime, format = "%d %B %Y %I:%M %p")
c$Stime=as.POSIXct(c$Stime, format = "%d %B %Y %I:%M %p")
```

Kết quả ta có bảng c như sau:

File 6:

ID	Status	Stime	Etime	Time	Diem	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
1	Đã hoàn thành	2020-03-29 09:03:00	2020-03-29 09:05:00	2 phút 15 giây	9,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	0,00
2	Đã hoàn thành	2020-03-29 09:25:00	2020-03-29 09:28:00	2 phút 52 giây	10,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
3	Đã hoàn thành	2020-03-29 09:35:00	2020-03-29 09:40:00	4 phút 53 giây	10,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
4	Đã hoàn thành	2020-03-29 10:10:00	2020-03-29 10:13:00	3 phút	10,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
5	Đã hoàn thành	2020-03-29 11:02:00	2020-03-29 11:17:00	14 phút 57 giây	10,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
6	Đã hoàn thành	2020-03-29 11:26:00	2020-03-29 11:29:00	3 phút 10 giây	10,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
7	Đã hoàn thành	2020-03-29 11:51:00	2020-03-29 11:54:00	3 phút 4 giây	10,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
8	Đã hoàn thành	2020-03-29 14:14:00	2020-03-29 14:18:00	3 phút 32 giây	10,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00

Showing 1 to 10 of 441 entries, 16 total columns

File 9:

ID	Status	Stime	Etime	Time	Diem	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
1	Đã hoàn thành	2020-04-04 09:00:00	2020-04-04 09:07:00	6 phút 14 giây	9,00	1,00	0,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
2	Đã hoàn thành	2020-04-04 09:09:00	2020-04-04 09:17:00	8 phút 32 giây	10,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
3	Đã hoàn thành	2020-04-04 09:24:00	2020-04-04 09:32:00	8 phút 31 giây	10,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
4	Đã hoàn thành	2020-04-04 09:32:00	2020-04-04 09:40:00	7 phút 57 giây	9,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	0,00	1,00	1,00
5	Đã hoàn thành	2020-04-04 09:41:00	2020-04-04 09:42:00	1 phút 4 giây	10,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
6	Đã hoàn thành	2020-04-04 09:55:00	2020-04-04 10:02:00	7 phút 11 giây	10,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
7	Đã hoàn thành	2020-04-04 10:21:00	2020-04-04 10:24:00	3 phút 3 giây	10,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00

Showing 1 to 9 of 466 entries, 16 total columns

File 10:

ID	Status	Stime	Etime	Time	Diem	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
1	Đã hoàn thành	2020-04-06 09:36:00	2020-04-06 09:38:00	2 phút 16 giây	8,50	1,00	1,00	1,00	1,00	1,00	1,00	1,00	0,50	1,00	0,00
2	Đã hoàn thành	2020-04-06 09:39:00	2020-04-06 09:40:00	1 giây	9,50	1,00	1,00	1,00	1,00	1,00	1,00	1,00	0,50	1,00	1,00
3	Đã hoàn thành	2020-04-06 09:40:00	2020-04-06 09:44:00	4 phút 10 giây	10,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
4	Đã hoàn thành	2020-04-06 09:40:00	2020-04-06 09:41:00	1 giây	10,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
5	Đã hoàn thành	2020-04-06 09:54:00	2020-04-06 09:58:00	3 phút 35 giây	9,50	1,00	1,00	1,00	1,00	1,00	1,00	1,00	0,50	1,00	1,00
6	Đã hoàn thành	2020-04-06 09:59:00	2020-04-06 10:01:00	2 phút 46 giây	10,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
7	Đã hoàn thành	2020-04-06 10:34:00	2020-04-06 10:39:00	4 phút 16 giây	9,50	1,00	1,00	1,00	1,00	1,00	1,00	1,00	0,50	1,00	1,00
8	Đã hoàn thành	2020-04-06 10:39:00	2020-04-06 10:40:00	1 giây	10,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00

Showing 1 to 10 of 551 entries, 16 total columns

File 12:

ID	Status	Stime	Etime	Time	Diem	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
1	Đã hoàn thành	2020-04-12 08:38:00	2020-04-12 08:42:00	4 phút 38 giây	9,00	1,00	1,00	1,00	0,00	1,00	1,00	1,00	1,00	1,00	1,00
2	Đã hoàn thành	2020-04-12 08:44:00	2020-04-12 08:45:00	1 giây	10,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
3	Đã hoàn thành	2020-04-12 08:52:00	2020-04-12 08:53:00	1 phút 4 giây	9,00	1,00	1,00	1,00	0,00	1,00	1,00	1,00	1,00	1,00	1,00
4	Đã hoàn thành	2020-04-12 09:20:00	2020-04-12 09:23:00	2 phút 54 giây	10,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
5	Đã hoàn thành	2020-04-12 09:31:00	2020-04-12 09:33:00	1 phút 47 giây	9,00	1,00	1,00	1,00	0,00	1,00	1,00	1,00	1,00	1,00	1,00
6	Đã hoàn thành	2020-04-12 09:38:00	2020-04-12 09:39:00	10 giây	10,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
7	Đã hoàn thành	2020-04-12 10:12:00	2020-04-12 10:13:00	1 phút 27 giây	10,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
8	Đã hoàn thành	2020-04-12 10:46:00	2020-04-12 10:50:00	4 phút	10,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
9	Đã hoàn thành	2020-04-12 11:02:00	2020-04-12 11:04:00	2 phút 30 giây	10,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
10	Đã hoàn thành	2020-04-12 12:55:00	2020-04-12 13:05:00	10 phút 17 giây	10,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
11	Đã hoàn thành	2020-04-12 13:06:00	2020-04-12 13:13:00	7 phút 44 giây	10,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00

Showing 1 to 13 of 353 entries, 16 total columns

- (b) Ta tạo vecto ID để lưu lại ID và vecto solannop để lưu lại số lần nộp bài của tất cả các sinh viên có trong c sau đó tạo một data frame với tên là data từ vecto ID và solannop:

```

ID=as.factor(c$ID)
ID <- ID[!is.na(ID)]
ID=levels(ID)
ID=as.numeric(ID)
solannop=rep(0,times=length(ID))
for(i in 1:length(ID)){
  for(j in 1:length(c$ID)){
    if(c$ID[j]== ID[i]) {solannop[i]=solannop[i]+1}
  }
}
data=data.frame(ID,solannop)

```

Kết quả thu được như sau:

File 6:

	ID	solannop
1	1511191	1
2	1613010	2
3	1812257	1
4	1812477	2
5	1812478	1
6	1813096	2
7	1813528	1
8	1813681	1
9	1814096	2

Showing 1 to 10 of 322 entries, 2 total columns

File 9:

	ID	solannop
1	1511191	3
2	1613010	2
3	1812257	1
4	1812477	2
5	1812478	1
6	1813096	1
7	1813528	2
8	1813681	1

Showing 1 to 9 of 306 entries, 2 total columns

File 10:

	ID	solannop
1	15111191	3
2	1613010	3
3	1812257	1
4	1812477	2
5	1812478	1
6	1813096	2
7	1813681	1
8	1814096	2
9	1814518	2

Showing 1 to 10 of 289 entries, 2 total columns

File 12:

	ID	solannop
1	1511191	2
2	1613010	1
3	1812257	1
4	1812477	2
5	1812478	1
6	1813096	1
7	1813681	1
8	1814096	1
9	1814518	1
10	1820028	1
11	1852443	2
12	1910006	1

Showing 1 to 13 of 280 entries, 2 total columns

- (c) Tạo vecto diemlandau và diemlanhai để lưu lại số điểm của từng ID ở từng lần nộp bài rồi thêm vào data hai cột là diemlandau và diemlanhai với dữ liệu từ hai vecto trên (chỉnh sửa các cột điểm sao cho những giá trị như NA hoặc ‘ – ‘ thì trở thành 0):



```
diemlandau=c() for(i in 1:length(data$ID)){
for(j in 1:length(c$ID)){
if(data$ID[i]==c$ID[j]){
diemlandau[i]=c$Diem[j]
break
}
}
}
for(i in 1:length(data$ID)){
for(j in 1:length(c$ID)){
if(data$ID[i]==c$ID[j] && diemlandau[i]=="-"){
diemlandau[i]=c$Diem[j]
break
}
}
}
data=cbind(data,diemlandau)
diemlanhai=c()
for(i in 1:length(data$ID)){
k=0
for (j in 1:length(c$ID)){
if(data$solannop[i]>1){
if(!is.na(c$Diem[j])){
if(data$ID[i]==c$ID[j]){k=k+1}
if(k==2){
if(c$Diem[j]=='-'){diemlandai[i]=0}
else{diemlanhai[i]=c$Diem[j]}
break
}
}
}
else{diemlanhai[i]=0}
}
}
data=cbind(data,diemlanhai)
for(i in 1:length(data$diemlandau)){
if(!is.na(data$diemlandau[i])){
if(data$diemlandau[i]=='-'){data$diemlandau[i]=0}
}
}
data$diemlandau=scan(text=data$diemlandau, dec=",", sep=".")
data$diemlanhai=scan(text=data$diemlanhai, dec=",", sep=".")
```

Kết quả thu được như sau: File 6:

ID	solannop	diemlandau	diemlanhai
1	1511191	1	10.0
2	1613010	2	7.5
3	1812257	1	10.0
4	1812477	2	9.0
5	1812478	1	8.5
6	1813096	2	9.5
7	1813528	1	10.0
8	1813681	1	9.0
9	1814096	2	9.5

Showing 1 to 10 of 322 entries, 4 total columns

File 9:

ID	solannop	diemlandau	diemlanhai
1	1511191	3	8.0
2	1613010	2	7.0
3	1812257	1	10.0
4	1812477	2	7.0
5	1812478	1	7.5
6	1813096	1	9.0
7	1813528	2	7.0
8	1813681	1	8.0

Showing 1 to 9 of 306 entries, 4 total columns

File 10:

ID	solannop	diemlandau	diemlanhai
1	1511191	3	9.0
2	1613010	3	9.0
3	1812257	1	9.0
4	1812477	2	8.0
5	1812478	1	8.5
6	1813096	2	9.0
7	1813681	1	10.0
8	1814096	2	9.5
9	1814518	2	9.5

Showing 1 to 10 of 289 entries, 4 total columns

File 12:

ID	solanhop	diemlandau	diemlanhai
1 1511191	2	9	10
2 1613010	1	10	0
3 1812257	1	10	0
4 1812477	2	7	10
5 1812478	1	10	0
6 1813096	1	10	0
7 1813681	1	9	0
8 1814096	1	10	0
9 1814518	1	10	0
10 1820028	1	10	0
11 1852443	2	9	10
12 1910006	1	10	0

Showing 1 to 13 of 280 entries, 4 total columns

- (d) Tạo ID9 chứa các ID của sinh viên có số điểm lần đầu hoặc số điểm lần hai lớn hơn hoặc bằng 8:

```
ID9=c()
k=0
for(i in 1:length(data$ID)){
  if(data$diemlandau[i]>=8 || data$diemlanhai[i]>=8){
    k=k+1
    ID9[k]=data$ID[i]
  }
}
```

Ta thu được ID9:

File 6:



> ID9

```
[1] 1511191 1613010 1812257 1812477 1812478 1813096 1813528 1813681 1814096
[10] 1814518 1814611 1820028 1852443 1910006 1910032 1910038 1910060 1910076
[19] 1910094 1910101 1910110 1910113 1910123 1910137 1910198 1910202 1910224
[28] 1910238 1910265 1910276 1910298 1910339 1910346 1910347 1910351 1910402
[37] 1910409 1910473 1910563 1910565 1910620 1910643 1910644 1910650 1910663
[46] 1910666 1910735 1910865 1910892 1910916 1910984 1911000 1911015 1911044
[55] 1911056 1911058 1911066 1911105 1911110 1911136 1911185 1911186 1911207
[64] 1911217 1911262 1911285 1911296 1911314 1911363 1911441 1911456 1911478
[73] 1911520 1911530 1911561 1911565 1911569 1911591 1911594 1911650 1911704
[82] 1911736 1911796 1911837 1911841 1911878 1911881 1911900 1911907 1911931
[91] 1911975 1912041 1912046 1912056 1912084 1912123 1912184 1912190 1912237
[100] 1912267 1912288 1912371 1912384 1912386 1912410 1912457 1912463 1912522
[109] 1912523 1912526 1912539 1912579 1912594 1912602 1912675 1912676 1912677
[118] 1912683 1912700 1912705 1912713 1912715 1912749 1912761 1912798 1912811
[127] 1912817 1912912 1912916 1912954 1912958 1912966 1912980 1913014 1913021
[136] 1913026 1913032 1913040 1913043 1913048 1913075 1913094 1913102 1913114
[145] 1913123 1913167 1913186 1913218 1913228 1913241 1913254 1913260 1913261
[154] 1913268 1913306 1913334 1913336 1913341 1913344 1913355 1913356 1913380
[163] 1913386 1913396 1913418 1913419 1913424 1913430 1913433 1913446 1913457
[172] 1913464 1913467 1913560 1913566 1913599 1913609 1913621 1913629 1913651
[181] 1913652 1913678 1913695 1913713 1913729 1913756 1913758 1913763 1913764
[190] 1913775 1913817 1913828 1913832 1913844 1913917 1913918 1913944 1913949
[199] 1913990 1914003 1914011 1914022 1914038 1914047 1914052 1914054 1914055
[208] 1914064 1914078 1914079 1914084 1914093 1914121 1914126 1914210 1914220
[217] 1914227 1914232 1914291 1914310 1914316 1914352 1914384 1914405 1914424
[226] 1914472 1914474 1914477 1914641 1914651 1914659 1914661 1914674 1914677
[235] 1914685 1914697 1914698 1914704 1914713 1914720 1914738 1914763 1914768
[244] 1914802 1914806 1914807 1914830 1914845 1914864 1914878 1914880 1914881
[253] 1914900 1914914 1914979 1915016 1915040 1915063 1915071 1915076 1915083
[262] 1915130 1915133 1915146 1915161 1915251 1915268 1915275 1915294 1915323
[271] 1915329 1915350 1915351 1915378 1915439 1915442 1915470 1915473 1915474
[280] 1915482 1915486 1915520 1915540 1915541 1915551 1915557 1915562 1915570
[289] 1915598 1915650 1915651 1915656 1915667 1915745 1915775 1915787 1915795
[298] 1915822 1915865 1915866 1915882 1915903 1915905 1915919 1915928 1915931
[307] 1915939 1915940 1915953 1915982 1915983 1915988 1915991 1916022 1916061
[316] 1927007 1936024 1937019 1937024 1937064
```

File 9:

> ID9

```
[1] 1511191 1613010 1812257 1812477 1812478 1813096 1813528 1813681 1814096
[10] 1910060 1910076 1910094 1910101 1910110 1910113 1910137 1910198 1910202 1910224
[29] 1910298 1910339 1910346 1910351 1910409 1910473 1910563 1910620 1910643 1910663
[43] 1910666 1910735 1910865 1910894 1910916 1910900 1911015 1911044 1911058 1911066
[57] 1911185 1911186 1911207 1911211 1911262 1911283 1911285 1911316 1911363 1911441
[71] 1911530 1911561 1911565 1911591 1911594 1911650 1911704 1911736 1911796 1911837
[85] 1911900 1911907 1911931 1911975 1912041 1912046 1912056 1912084 1912184 1912190
[99] 1912371 1912384 1912386 1912410 1912457 1912463 1912522 1912523 1912528 1912539
[113] 1912676 1912683 1912700 1912705 1912713 1912715 1912749 1912761 1912811 1912817
[127] 1912966 1912980 1913014 1913021 1913026 1913045 1913048 1913075 1913094 1913102
[141] 1913186 1913218 1913241 1913251 1913260 1913268 1913306 1913334 1913341 1913356
[155] 1913380 1913386 1913396 1913418 1913419 1913424 1913430 1913433 1913446 1913457
[169] 1913599 1913609 1913621 1913629 1913651 1913652 1913678 1913695 1913713 1913729
[183] 1913828 1913832 1913844 1913918 1913944 1913949 1914003 1914011 1914022 1914038
[197] 1914055 1914064 1914078 1914097 1914098 1914099 1914121 1914126 1914210 1914227
[211] 1914136 1914132 1914138 1914140 1914142 1914144 19141472 19141474 19141477 191414641
[225] 19141683 19141698 19141704 1914173 1914173 1914174 19141763 19141768 19141806
[239] 19141878 19141880 19141881 19141911 19141979 1915016 1915040 1915063 1915071
[253] 1915251 1915268 1915275 1915294 1915323 1915350 1915351 1915378 1915439 1915442
[267] 1915482 1915486 1915520 1915541 1915551 1915557 1915562 1915570 1915598 1915651
[281] 1915787 1915795 1915822 1915865 1915866 1915882 1915903 1915905 1915919 1915931
[295] 1915983 1915988 1915991 1916022 1916061 1927007 1936024 1937019 1937024 1937064
```

File 10:

> ID9

```
[1] 1511191 1613010 1812257 1812477 1812478 1813096 1813681 1814096 1814518 1820028 1852443 1910006 1910032
[14] 1910038 1910060 1910076 1910094 1910101 1910110 1910113 1910137 1910198 1910202 1910224 1910238 1910265
[27] 1910276 1910298 1910339 1910346 1910351 1910409 1910473 1910565 1910620 1910643 1910644
[40] 1910650 1910663 1910666 1910865 1910892 1910916 1910984 1911000 1911015 1911056 1911066
[53] 1911105 1911110 1911116 1911185 1911186 1911207 1911217 1911219 1911283 1911285 1911296 1911314 1911363
[66] 1911441 1911478 1911520 1911530 1911561 1911565 1911569 1911591 1911594 1911650 1911704 1911736 1911796
[79] 1911837 1911841 1911881 1911900 1911907 1911931 1912041 1912046 1912056 1912084 1912123 1912184 1912190
[92] 1912237 1912267 1912288 1912371 1912384 1912386 1912410 1912457 1912463 1912523 1912526 1912539 1912579
[105] 1912590 1912602 1912675 1912676 1912683 1912700 1912705 1912713 1912715 1912749 1912761 1912798 1912811
[118] 1912817 1912912 1912916 1912954 1912961 1912980 1913014 1913021 1913026 1913040 1913045 1913048 1913075
[131] 1913091 1913102 1913114 1913123 1913167 1913186 1913218 1913241 1913254 1913261 1913268 1913306 1913334
[144] 1913336 1913341 1913354 1913355 1913356 1913380 1913386 1913396 1913418 1913419 1913424 1913430 1913433
[157] 1913446 1913457 1913464 1913467 1913560 1913566 1913599 1913609 1913621 1913629 1913651 1913652 1913678
[170] 1913695 1913713 1913729 1913763 1913764 1913775 1913817 1913828 1913832 1913844 1913918 1913944 1913949
[183] 1914003 1914013 1914022 1914038 1914047 1914052 1914054 1914055 1914078 1914079 1914084 1914093 1914121
[196] 1914120 1914210 1914220 1914227 1914232 1914291 1914310 1914316 1914352 1914384 1914405 1914424 1914472
[209] 1914474 1914477 1914478 1914641 1914651 1914659 1914661 1914674 1914677 1914685 1914689 1914704 1914713 1914720
[222] 1914738 1914763 1914768 1914807 1914813 1914845 1914864 1914880 1914881 1914914 1914979 1915016 1915040
[235] 1915063 1915073 1915076 1915083 1915130 1915133 1915146 1915251 1915268 1915275 1915294 1915323 1915329
[248] 1915350 1915351 1915378 1915439 1915442 1915470 1915473 1915474 1915482 1915486 1915520 1915540 1915541
[261] 1915551 1915557 1915562 1915570 1915598 1915651 1915656 1915745 1915775 1915787 1915795 1915822 1915865
[274] 1915866 1915882 1915903 1915919 1915940 1915953 1915982 1915983 1915988 1915991 1916022 1916061 1927007
```

File 12:



```
> ID9
[1] 1511191 1613010 1812257 1812477 1812478 1813093 1813681 1814096 1814518 1820028 1852443 1910006 1910032 1910038
[15] 1910060 1910076 1910094 1910101 1910113 1910123 1910137 1910202 1910224 1910233 1910265 1910276 1910298
[29] 1910338 1910346 1910347 1910351 1910402 1910407 1910565 1910620 1910643 1910650 1910663 1910666
[43] 1910865 1910892 1910916 1910984 1911000 1911015 1911056 1911058 1911068 1911110 1911130 1911185 1911186 1911207
[57] 1911217 1911262 1911283 1911296 1911313 1911363 1911441 1911478 1911520 1911530 1911565 1911569 1911591
[71] 1911594 1911650 1911704 1911736 1911796 1911837 1911841 1911881 1911900 1911907 1911931 1912041 1912046 1912056
[85] 1912084 1912123 1912184 1912190 1912237 1912267 1912288 1912371 1912384 1912386 1912410 1912457 1912463 1912523
[99] 1912539 1912579 1912594 1912602 1912675 1912676 1912683 1912700 1912713 1912715 1912749 1912761 1912798 1912811
[113] 1912817 1912912 1912916 1912966 1912980 1913014 1913021 1913026 1913044 1913048 1913075 1913094 1913102
[127] 1913114 1913123 1913167 1913188 1913218 1913241 1913254 1913261 1913268 1913306 1913332 1913336 1913341 1913354
[141] 1913355 1913356 1913380 1913386 1913396 1913418 1913419 1913424 1913430 1913433 1913446 1913457 1913464 1913467
[155] 1913560 1913566 1913599 1913601 1913621 1913651 1913652 1913678 1913695 1913713 1913729 1913763 1913764 1913775
[169] 1913817 1913828 1913832 1913844 1913918 1913944 1913949 1914003 1914011 1914022 1914038 1914047 1914052 1914054
[183] 1914055 1914078 1914079 1914084 1914093 1914121 1914126 1914210 1914220 1914227 1914232 1914291 1914310 1914316
[197] 1914352 1914384 1914405 1914424 1914472 1914474 1914477 1914478 1914461 19144651 19144659 19144661 19144674 19144677 19144685
[211] 1914694 1914704 1914713 1914720 1914738 1914763 1914768 1914807 1914830 1914845 1914861 1914880 1914881 1914914
[225] 1914979 1915016 1915040 1915063 1915071 1915076 1915083 1915130 1915133 1915146 1915251 1915268 1915275 1915294
[239] 1915323 1915329 1915350 1915351 1915378 1915439 1915442 1915473 1915474 1915482 1915486 1915520 1915540 1915541
[253] 1915551 1915557 1915562 1915570 1915598 1915651 1915656 1915745 1915775 1915787 1915795 1915822 1915865 1915866
[267] 1915882 1915903 1915919 1915953 1915982 1915988 1915991 1916022 1916061 1927007 1936024 1937024 1937064
```

(e) Tạo vecto diem chứa điểm sau cùng của tất cả các ID trong ID9 và cùng với ID9 tạo ra data9 :

```
for(i in 1:length(c$Diem)){
  if(!is.na(c$Diem[i])){
    if(c$Diem[i]=='-'){c$Diem[i]=0}
  }
}
c$Diem=scan(text=c$Diem, dec=",", sep=".")
diem=c()
for(i in 1:length(ID9)){
  k=0
  for(j in 1:length(c$ID)){
    if(ID9[i]==c$ID[j]){
      if(!is.na(c$Diem[j])){
        if(k==0){
          k=k+1
          diem[i]=c$Diem[j]
        }
        else{
          if(c$Diem[j]>diem[i]){diem[i]=c$Diem[j]}
        }
      }
    }
  }
}
data9=data.frame(ID9,diem)
```

Ta thu được:

File 6:

	ID9	diem
1	15111191	10.0
2	1613010	9.0
3	1812257	10.0
4	1812477	10.0
5	1812478	8.5
6	1813096	10.0
7	1813528	10.0
8	1813681	9.0
9	1814096	10.0
10	1814518	10.0

Showing 1 to 11 of 320 entries, 2 total columns

File 9:

	ID9	diem
1	15111191	10
2	1613010	10
3	1812257	10
4	1812477	10
5	1813096	9
6	1813528	10
7	1813681	8
8	1814096	10
9	1814518	10

Showing 1 to 10 of 304 entries, 2 total columns

File 10:

	ID9	diem
1	15111191	10.0
2	1613010	10.0
3	1812257	9.0
4	1812477	10.0
5	1812478	8.5
6	1813096	10.0
7	1813681	10.0
8	1814096	9.5
9	1814518	10.0
10	1820028	10.0
11	1852443	10.0
12	1910006	10.0

Showing 1 to 13 of 289 entries, 2 total columns

File 12:

	ID9	diem
1	1511191	10
2	1613010	10
3	1812257	10
4	1812477	10
5	1812478	10
6	1813096	10
7	1813681	9
8	1814096	10
9	1814518	10
10	1820000	10

Showing 1 to 10 of 280 entries, 2 total columns

b) Số lượng sinh viên thông minh là số ID có trong data9:

```
sosinhvienthongminh=length(data9$ID9)
```

Kết quả là:

File 6:

```
> sosinhvienthongminh  
[1] 320
```

File 9:

```
> sosinhvienthongminh  
[1] 304
```

File 10:

```
> sosinhvienthongminh  
[1] 289
```

File 12:

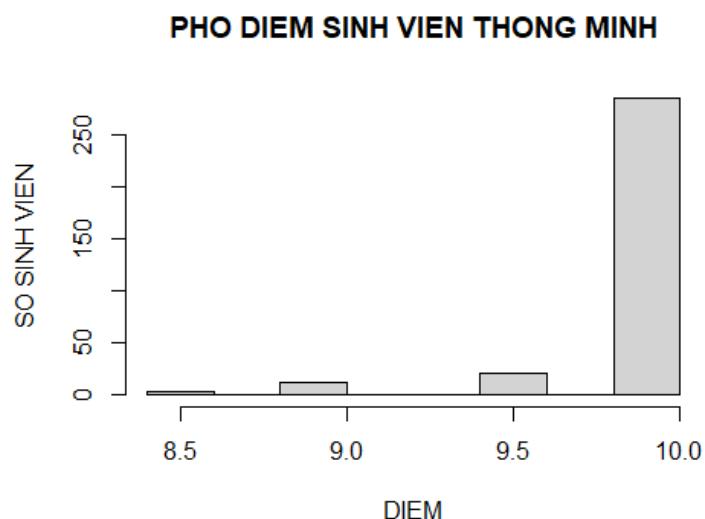
```
> sosinhvienthongminh  
[1] 280
```

c) Biểu đồ thể hiện phổ điểm của sinh viên thông minh với dữ liệu được lấy từ data9 như sau:

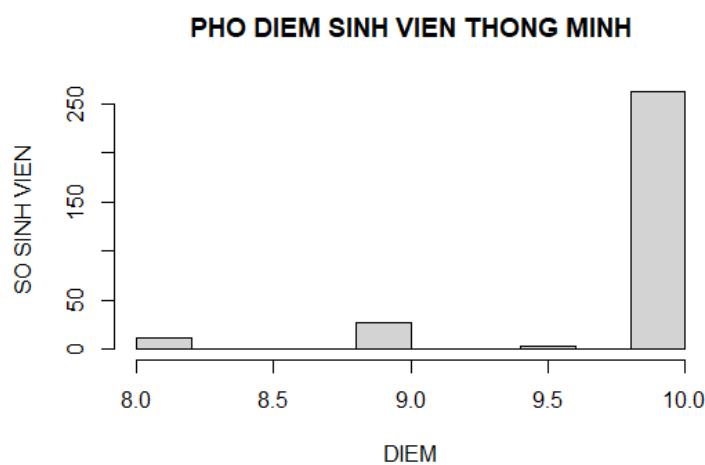
```
hist(data9$diem,main="PHO DIEM SINH VIEN THONG MINH", xlab="DIEM",ylab = "SO SINH VIEN")
```

Kết quả là:

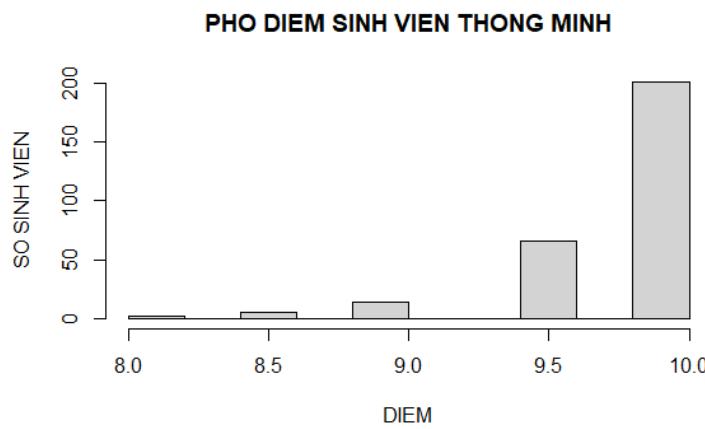
File 6:



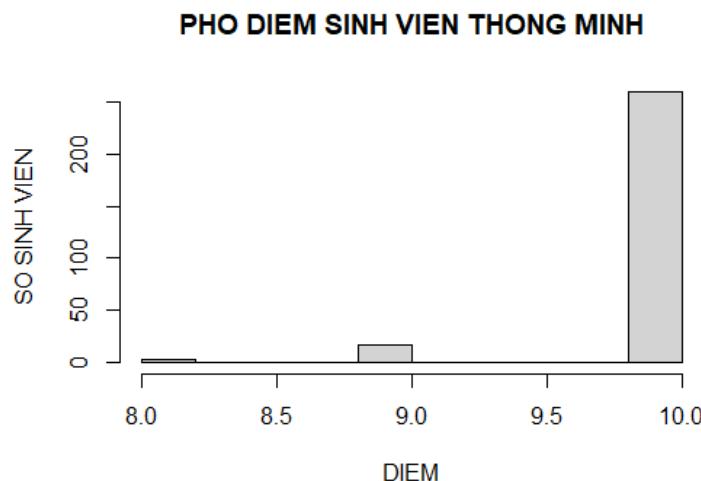
File 9:



File 10:



File 12:



4.8 Câu 10: Nhóm câu hỏi liên quan sinh viên chủ động

- a) Sinh viên chủ động sẽ là sinh viên có thời gian nộp bài lần đầu trước t1 là thời điểm 1/3 quãng thời gian kể từ lúc người đầu tiên nộp bài đến thời điểm người cuối cùng nộp bài, có số lần nộp nhiều hơn 1 và có số điểm nộp lần 1 hoặc lần 2 lớn hơn bằng 8.

(a) Đầu tiên ta tạo một data frame là c để chứa dữ liệu từ file xlsx:

```
library(readxl)
library(utf8)
library(dplyr)
library(lubridate)
library("zoo")
a<-read_xlsx("filename.xlsx")
c=data.frame(a)
names(c)[1]<-"ID"
names(c)[2]<-"Status"
names(c)[3]<-"Stime"
names(c)[4]<-"Etime"
names(c)[5]<-"Time"
names(c)[6]<-"Diem"
names(c)[7]<-"Q1"
names(c)[8]<-"Q2"
names(c)[9]<-"Q3"
names(c)[10]<-"Q4"
names(c)[11]<-"Q5"
names(c)[12]<-"Q6"
names(c)[13]<-"Q7"
names(c)[14]<-"Q8"
names(c)[15]<-"Q9"
names(c)[16]<-"Q10"
c=subset(c,!is.na(c$ID))
c$ID=as.numeric(c$ID)
c$Etime=as.POSIXct(c$Etime, format = "%d %B %Y %I:%M %p")
c$Stime=as.POSIXct(c$Stime, format = "%d %B %Y %I:%M %p")
```

Kết quả ta có như sau: File 6:

ID	Status	Stime	Etime	Time	Diem	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
1	Đã hoàn thành	2020-03-29 09:03:00	2020-03-29 09:05:00	2 phút 15 giây	9,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	0,00
2	Đã hoàn thành	2020-03-29 09:25:00	2020-03-29 09:28:00	2 phút 52 giây	10,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
3	Đã hoàn thành	2020-03-29 09:35:00	2020-03-29 09:40:00	4 phút 53 giây	10,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
4	Đã hoàn thành	2020-03-29 10:10:00	2020-03-29 10:13:00	3 phút	10,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
5	Đã hoàn thành	2020-03-29 11:20:00	2020-03-29 11:37:00	14 phút 57 giây	10,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
6	Đã hoàn thành	2020-03-29 11:26:00	2020-03-29 11:29:00	3 phút 40 giây	10,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
7	Đã hoàn thành	2020-03-29 11:51:00	2020-03-29 11:54:00	3 phút 4 giây	10,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
8	Đã hoàn thành	2020-03-29 14:14:00	2020-03-29 14:18:00	3 phút 32 giây	10,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
9	Đã hoàn thành	2020-03-29 15:39:00	2020-03-29 15:44:00	5 phút 29 giây	10,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00

Showing 1 to 11 of 441 entries, 16 total columns

File 9:

ID	Status	Stime	Etime	Time	Diem	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
1	Đã hoàn thành	2020-04-04 09:00:00	2020-04-04 09:07:00	6 phút 14 giây	9,00	1,00	0,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
2	Đã hoàn thành	2020-04-04 09:09:00	2020-04-04 09:17:00	8 phút 32 giây	10,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
3	Đã hoàn thành	2020-04-04 09:42:00	2020-04-04 09:32:00	8 phút 31 giây	10,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
4	Đã hoàn thành	2020-04-04 09:32:00	2020-04-04 09:40:00	7 phút 57 giây	9,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	0,00	1,00
5	Đã hoàn thành	2020-04-04 09:41:00	2020-04-04 09:42:00	1 phút 4 giây	10,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
6	Đã hoàn thành	2020-04-04 09:55:00	2020-04-04 10:02:00	7 phút 11 giây	10,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
7	Đã hoàn thành	2020-04-04 10:21:00	2020-04-04 10:24:00	3 phút 8 giây	10,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
8	Đã hoàn thành	2020-04-04 10:42:00	2020-04-04 10:45:00	3 phút 24 giây	10,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00

Showing 1 to 10 of 466 entries, 16 total columns

File 10:

ID	Status	Stime	Etime	Time	Diem	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
1	Đã hoàn thành	2020-04-06 09:36:00	2020-04-06 09:38:00	2 phút 16 giây	8,50	1,00	1,00	1,00	1,00	1,00	1,00	1,00	0,50	1,00	0,00
2	Đã hoàn thành	2020-04-06 09:39:00	2020-04-06 09:40:00	45 giây	9,50	1,00	1,00	1,00	1,00	1,00	1,00	1,00	0,50	1,00	1,00
3	Đã hoàn thành	2020-04-06 09:40:00	2020-04-06 09:44:00	4 phút 10 giây	10,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
4	Đã hoàn thành	2020-04-06 09:40:00	2020-04-06 09:41:00	42 giây	10,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
5	Đã hoàn thành	2020-04-06 09:54:00	2020-04-06 09:58:00	3 phút 35 giây	9,50	1,00	1,00	1,00	1,00	1,00	1,00	1,00	0,50	1,00	1,00
6	Đã hoàn thành	2020-04-06 09:59:00	2020-04-06 10:01:00	2 phút 48 giây	10,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
7	Đã hoàn thành	2020-04-06 10:34:00	2020-04-06 10:39:00	4 phút 16 giây	9,50	1,00	1,00	1,00	1,00	1,00	1,00	1,00	0,50	1,00	1,00
8	Đã hoàn thành	2020-04-06 10:39:00	2020-04-06 10:40:00	38 giây	10,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00

Showing 1 to 10 of 551 entries, 16 total columns

File 12:

ID	Status	Stime	Etime	Time	Diem	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
1	Đã hoàn thành	2020-04-12 08:38:00	2020-04-12 08:42:00	4 phút 38 giây	9,00	1,00	1,00	1,00	0,00	1,00	1,00	1,00	1,00	1,00	1,00
2	Đã hoàn thành	2020-04-12 08:44:00	2020-04-12 08:45:00	39 giây	10,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
3	Đã hoàn thành	2020-04-12 08:52:00	2020-04-12 08:53:00	1 phút 4 giây	9,00	1,00	1,00	1,00	0,00	1,00	1,00	1,00	1,00	1,00	1,00
4	Đã hoàn thành	2020-04-12 09:20:00	2020-04-12 09:23:00	2 phút 54 giây	10,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
5	Đã hoàn thành	2020-04-12 09:31:00	2020-04-12 09:33:00	1 phút 47 giây	9,00	1,00	1,00	1,00	0,00	1,00	1,00	1,00	1,00	1,00	1,00
6	Đã hoàn thành	2020-04-12 09:38:00	2020-04-12 09:39:00	40 giây	10,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
7	Đã hoàn thành	2020-04-12 10:12:00	2020-04-12 10:13:00	1 phút 27 giây	10,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
8	Đã hoàn thành	2020-04-12 10:46:00	2020-04-12 10:50:00	4 phút	10,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00

Showing 1 to 10 of 353 entries, 16 total columns

- (b) Ta tạo vecto ID để lưu lại ID và vecto solannop để lưu lại số lần nộp bài của tất cả các sinh viên có trong c sau đó tạo một data frame với tên là data từ vecto ID và solannop:

```

ID=as.factor(c$ID)
ID <- ID[!is.na(ID)]
ID=levels(ID)
ID=as.numeric(ID)
solannop=rep(0,times=length(ID))
for(i in 1:length(ID)){
  for(j in 1:length(c$ID)){
    if(c$ID[j]== ID[i]) {solannop[i]=solannop[i]+1}
  }
}
data=data.frame(ID,solannop)

```

Kết quả ta được data như sau:

File 6:

	ID	solannop
1	15111191	1
2	1613010	2
3	1812257	1
4	1812477	2
5	1812478	1
6	1813096	2
7	1813528	1
8	1813681	1
9	1814096	2
10	1814518	1

Showing 1 to 11 of 322 entries, 2 total columns

File 9:

	ID	solannop
1	15111191	3
2	1613010	2
3	1812257	1
4	1812477	2
5	1812478	1
6	1813096	1
7	1813528	2
8	1813681	1
9	1814096	3
10	1814518	2

Showing 1 to 10 of 306 entries, 2 total columns

File 10:

	ID	solannop
1	15111191	3
2	1613010	3
3	1812257	1
4	1812477	2
5	1812478	1
6	1813096	2
7	1813681	1
8	1814096	2
9	1814518	2
10	18200000	2

Showing 1 to 10 of 289 entries, 2 total columns

File 12:

	ID	solannop
1	15111191	2
2	1613010	1
3	1812257	1
4	1812477	2
5	1812478	1
6	1813096	1
7	1813681	1
8	1814096	1
9	1814518	1
10	18200000	1

Showing 1 to 10 of 280 entries, 2 total columns

(c) Tìm t1 là biến lưu thời điểm t1:



```
tmax=max(c$Etime,na.rm = T)
tmin=min(c$Etime,na.rm = T)
hours=as.period(interval(tmin, tmax), unit = "hours")
sociocongthem=hour(hours)-round(hour(hours)*2/3)
t1=tmin+hours(sociocongthem)
```

Kết quả t1: File 6:

```
> t1
[1] "2020-04-15 03:05:00 +07"
```

File 9:

```
> t1
[1] "2020-04-19 06:07:00 +07"
```

File 10:

```
> t1
[1] "2020-04-20 13:38:00 +07"
```

File 12:

```
> t1
[1] "2020-04-24 12:42:00 +07"
```

- (d) Thêm vào data các cột landaunop, diemlandau và diemlanhai là các cột chứa thời điểm lần đầu nộp, điểm lần đầu và lần hai của từng ID (chỉnh sửa các cột điểm sao cho những giá trị như NA hoặc ‘-’ thì trở thành 0):



```
landaunop=c()
for(i in 1:length(ID)){
for(j in 1:length(c$ID)){
if(ID[i]==c$ID[j]){
landaunop[i]=c$Etime[j]
j=length(c$ID)
}}}
landaunop=as.POSIXct(landaunop,tz = "GMT",origin, format = "%d %B %Y %I:%M %p")
data=cbind(data,landaunop)
diemlandau=c()
for(i in 1:length(data$ID)){
for(j in 1:length(c$ID)){
if(data$ID[i]==c$ID[j]){
diemlandau[i]=c$Diem[j]
break
}}}
for(i in 1:length(data$ID)){
for(j in 1:length(c$ID)){
if(data$ID[i]==c$ID[j] && diemlandau[i]=="-"){
diemlandau[i]=c$Diem[j]
break
}}}
data=cbind(data,diemlandau)
diemlanhai=c()
for(i in 1:length(data$ID)){
k=0
for (j in 1:length(c$ID)){
if(data$solannop[i]>1){
if(!is.na(c$Diem[j])){
if(data$ID[i]==c$ID[j]){k=k+1}
if(k==2){
if(c$Diem[j]=='-'){diemlandai[i]=0}
else{diemlanhai[i]=c$Diem[j]}
break
}}}
else{diemlanhai[i]=0}
}}
data=cbind(data,diemlanhai)
for(i in 1:length(data$diemlandau)){
if(!is.na(data$diemlandau[i])){
if(data$diemlandau[i]=='-'){data$diemlandau[i]=0}
}}
for(i in 1:length(data$diemlanhai)){
if(!is.na(data$diemlanhai[i])){
if(data$diemlanhai[i]=='-'){data$diemlanhai[i]=0}
}}
data$diemlandau=scan(text=data$diemlandau, dec=",", sep=". ")
data$diemlanhai=scan(text=data$diemlanhai, dec=",", sep=". ")
```

Kết quả data:

File 6:



ID	solannop	landaunop	diemlandau	diemlanhai	
1	1511191	1	2020-04-04 08:19:00	10.0	0.0
2	1613010	2	2020-04-19 16:33:00	7.5	9.0
3	1812257	1	2020-04-20 06:19:00	10.0	0.0
4	1812477	2	2020-04-23 00:46:00	9.0	10.0
5	1812478	1	2020-04-06 09:59:00	8.5	0.0
6	1813096	2	2020-04-12 08:19:00	9.5	10.0
7	1813528	1	2020-05-05 23:14:00	10.0	0.0
8	1813681	1	2020-05-11 06:57:00	9.0	0.0
9	1814096	2	2020-03-30 09:10:00	9.5	10.0

Showing 1 to 11 of 322 entries, 5 total columns

File 9:

ID	solannop	landaunop	diemlandau	diemlanhai	
1	1511191	3	2020-04-04 13:19:00	8.0	9.0
2	1613010	2	2020-04-19 16:40:00	7.0	10.0
3	1812257	1	2020-04-20 06:52:00	10.0	0.0
4	1812477	2	2020-04-24 07:19:00	7.0	10.0
5	1812478	1	2020-04-06 10:41:00	7.5	0.0
6	1813096	1	2020-04-13 03:08:00	9.0	0.0
7	1813528	2	2020-05-05 23:42:00	7.0	10.0
8	1813681	1	2020-05-18 01:37:00	8.0	0.0
9	1814096	2	2020-04-12 00:04:00	9.0	0.0

Showing 1 to 10 of 306 entries, 5 total columns

File 10:

ID	solannop	landaunop	diemlandau	diemlanhai	
1	1511191	3	2020-04-20 06:08:00	9.0	9.5
2	1613010	3	2020-04-27 10:11:00	9.0	9.5
3	1812257	1	2020-04-20 10:32:00	9.0	0.0
4	1812477	2	2020-04-24 11:54:00	8.0	10.0
5	1812478	1	2020-04-06 10:43:00	8.5	0.0
6	1813096	2	2020-05-06 02:51:00	9.0	10.0
7	1813681	1	2020-05-18 01:52:00	10.0	0.0
8	1814096	2	2020-04-15 07:42:00	9.5	9.5
9	1814510	2	2020-04-28 01:29:00	9.5	10.0

Showing 1 to 10 of 289 entries, 5 total columns

File 12:

	ID	solanop	landaunop	diemlandau	diemlanhai
1	1511191	2	2020-04-20 06:26:00	9	10
2	1613010	1	2020-04-27 10:18:00	10	0
3	1812257	1	2020-04-20 10:45:00	10	0
4	1812477	2	2020-04-25 08:53:00	7	10
5	1812478	1	2020-04-13 05:33:00	10	0
6	1813096	1	2020-05-06 04:38:00	10	0
7	1813681	1	2020-05-18 02:42:00	9	0
8	1814096	1	2020-04-15 07:56:00	10	0
9	18141510	1	2020-04-28 01:24:00	10	0

Showing 1 to 10 of 280 entries, 5 total columns

- (e) Tạo vecto ID10 chứa tất cả ID của sinh viên chủ động và vecto diem chứa điểm của các ID đó rồi tạo thành data frame data10 từ hai vecto trên:

```
ID10=c()
k=0
for(i in 1:length(data$ID)){
  if((data$landaunop[i]<t1 || data$diemlandau[i]>=8 || data$diemlanhai[i]>=8) && data$solanop[i]>1){
    k=k+1
    ID10[k]=data$ID[i]
  }
}
for(i in 1:length(c$Diem)){
  if(!is.na(c$Diem[i])){
    if(c$Diem[i]=='-') {c$Diem[i]=0}
  }
}
c$Diem=scan(text=c$Diem, dec=",", sep=".")
diem=c()
for(i in 1:length(ID10)){
  k=0
  for(j in 1:length(c$ID)){
    if(ID10[i]==c$ID[j]){
      if(!is.na(c$Diem[j])){
        if(k==0){
          k=k+1
          diem[i]=c$Diem[j]
        }
        else{
          if(c$Diem[j]>diem[i]) {diem[i]=c$Diem[j]}
        }
      }
    }
  }
}
data10=data.frame(ID10,diem)
```

Kết quả data10:

File 6:

	ID10	diem
1	1613010	9.0
2	1812477	10.0
3	1813096	10.0
4	1814096	10.0
5	1910006	10.0
6	1910038	10.0
7	1910276	10.0
8	1910339	10.0
9	1910351	10.0
10	1910650	10.0

Showing 1 to 11 of 96 entries, 2 total columns

File 9:

	ID10	diem
1	1511191	10
2	1613010	10
3	1812477	10
4	1813528	10
5	1814096	10
6	1814518	10
7	1852443	10
8	1910006	10
9	1910060	10
10	1910112	10

Showing 1 to 10 of 143 entries, 2 total columns

File 10:

	ID10	diem
1	1511191	10.0
2	1613010	10.0
3	1812477	10.0
4	1813096	10.0
5	1814096	9.5
6	1814518	10.0
7	1820028	10.0
8	1852443	10.0
9	1910006	10.0
10	1910060	10.0

Showing 1 to 10 of 177 entries, 2 total columns

File 12:

	ID10	diem
1	1511191	10
2	1812477	10
3	1852443	10
4	1910123	10
5	1910409	10
6	1910892	10
7	1911066	10
8	1911262	10
9	1911363	10
10	1911444	10

Showing 1 to 10 of 70 entries, 2 total columns

b) Số lượng sinh viên chủ động là số ID có trong data10:

```
sosinhvienchudong=length(data10$ID10)
```

Kết quả thu được như sau:

File 6:

```
> sosinhvienchudong  
[1] 96
```

File 9:

```
> sosinhvienchudong  
[1] 143
```

File 10:

```
> sosinhvienchudong  
[1] 177
```

File 12:

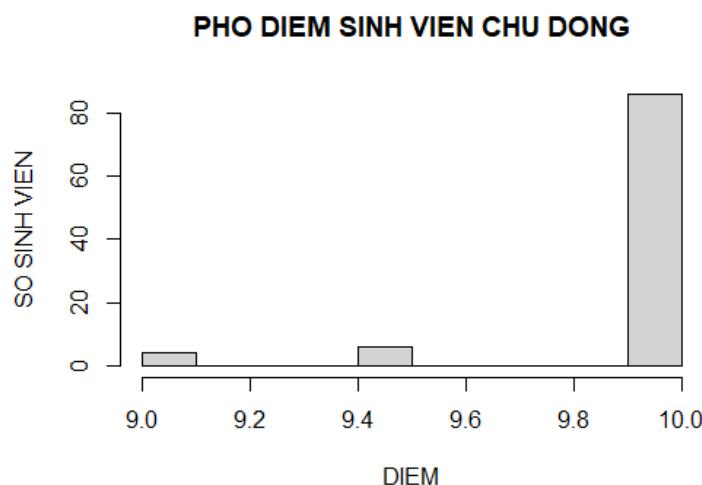
```
> sosinhvienchudong  
[1] 70
```

c) Biểu đồ thể hiện phô điểm của sinh viên chủ động với dữ liệu được lấy từ data10 như sau:

```
hist(data10$diem,main="PHO DIEM SINH VIEN CHU DONG", xlabel="DIEM",ylab = "SO SINH VIEN")
```

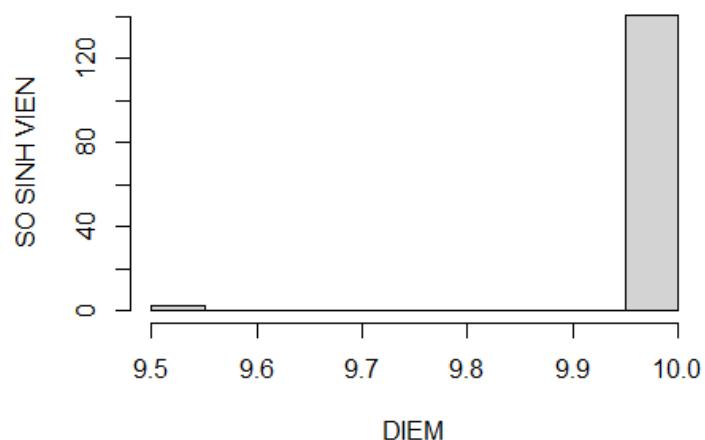
Kết quả thu được như sau:

File 6:



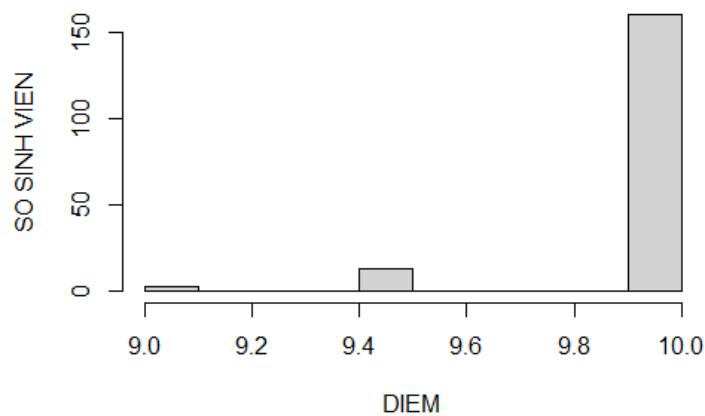
File 9:

PHO DIEM SINH VIEN CHU DONG



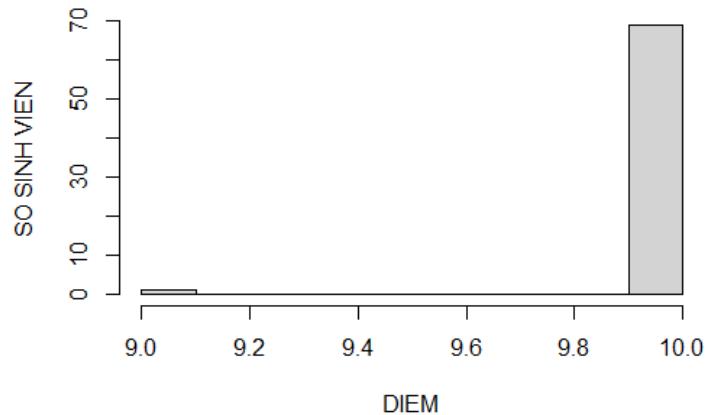
File 10:

PHO DIEM SINH VIEN CHU DONG



File 12:

PHO DIEM SINH VIEN CHU DONG



4.9 Câu 11: Phản giao của các loại sinh viên được đánh giá

- Đầu tiên ta cần cài đặt gói VennDiagram để hỗ trợ vẽ biểu đồ ven:

```
install.packages('VennDiagram')
```

- Ta tạo một data frame là c để chứa dữ liệu từ file xlsx:

```
library(VennDiagram)
library(readxl)
library(utf8)
library(dplyr)
library(lubridate)
library("zoo")
a<-read_xlsx("filename.xlsx")
c=data.frame(a)
names(c)[1]<-"ID"
names(c)[2]<-"Status"
names(c)[3]<-"Stime"
names(c)[4]<-"Etime"
names(c)[5]<-"Time"
names(c)[6]<-"Diem"
names(c)[7]<-"Q1"
names(c)[8]<-"Q2"
names(c)[9]<-"Q3"
names(c)[10]<-"Q4"
names(c)[11]<-"Q5"
names(c)[12]<-"Q6"
names(c)[13]<-"Q7"
names(c)[14]<-"Q8"
names(c)[15]<-"Q9"
names(c)[16]<-"Q10"
c=subset(c,!is.na(c$ID))
c$ID=as.numeric(c$ID)
c$Etime=as.POSIXct(c$Etime, format = "%d %B %Y %I:%M %p")
c$Stime=as.POSIXct(c$Stime, format = "%d %B %Y %I:%M %p")
```

Ta thu được bảng c như sau:

File 6:

ID	Status	Stime	Etime	Time	Diem	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
1	Đã hoàn thành	2020-03-29 09:03:00	2020-03-29 09:05:00	2 phút 15 giây	9,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	0,00
2	Đã hoàn thành	2020-03-29 09:25:00	2020-03-29 09:28:00	2 phút 52 giây	10,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
3	Đã hoàn thành	2020-03-29 09:35:00	2020-03-29 09:40:00	4 phút 53 giây	10,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
4	Đã hoàn thành	2020-03-29 10:10:00	2020-03-29 10:13:00	3 phút	10,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
5	Đã hoàn thành	2020-03-29 11:02:00	2020-03-29 11:17:00	14 phút 57 giây	10,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
6	Đã hoàn thành	2020-03-29 11:26:00	2020-03-29 11:29:00	3 phút 10 giây	10,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
7	Đã hoàn thành	2020-03-29 11:51:00	2020-03-29 11:54:00	3 phút 4 giây	10,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
8	Đã hoàn thành	2020-03-29 14:14:00	2020-03-29 14:18:00	3 phút 32 giây	10,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
9	Đã hoàn thành	2020-03-29 15:39:00	2020-03-29 15:44:00	5 phút 29 giây	10,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00

Showing 1 to 11 of 441 entries, 16 total columns

File 9:



ID	Status	Stime	Etime	Time	Diem	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
1	Đã hoàn thành	2020-04-04 09:00:00	2020-04-04 09:07:00	6 phút 14 giây	9,00	1,00	0,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
2	Đã hoàn thành	2020-04-04 09:09:00	2020-04-04 09:17:00	8 phút 32 giây	10,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
3	Đã hoàn thành	2020-04-04 09:24:00	2020-04-04 09:32:00	8 phút 31 giây	10,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
4	Đã hoàn thành	2020-04-04 09:32:00	2020-04-04 09:40:00	7 phút 57 giây	9,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	0,00	1,00	1,00
5	Đã hoàn thành	2020-04-04 09:41:00	2020-04-04 09:42:00	1 phút 4 giây	10,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
6	Đã hoàn thành	2020-04-04 09:55:00	2020-04-04 10:02:00	7 phút 11 giây	10,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
7	Đã hoàn thành	2020-04-04 10:21:00	2020-04-04 10:24:00	3 phút 8 giây	10,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
8	Đã hoàn thành	2020-04-04 10:42:00	2020-04-04 10:45:00	3 phút 24 giây	10,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
...

Showing 1 to 10 of 466 entries, 16 total columns

File 10:

ID	Status	Stime	Etime	Time	Diem	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
1	Đã hoàn thành	2020-04-06 09:36:00	2020-04-06 09:38:00	2 phút 16 giây	8,50	1,00	1,00	1,00	1,00	1,00	1,00	0,50	1,00	0,00	1,00
2	Đã hoàn thành	2020-04-06 09:39:00	2020-04-06 09:40:00	45 giây	9,50	1,00	1,00	1,00	1,00	1,00	1,00	0,50	1,00	1,00	1,00
3	Đã hoàn thành	2020-04-06 09:40:00	2020-04-06 09:44:00	4 phút 10 giây	10,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
4	Đã hoàn thành	2020-04-06 09:40:00	2020-04-06 09:41:00	42 giây	10,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
5	Đã hoàn thành	2020-04-06 09:54:00	2020-04-06 09:56:00	3 phút 35 giây	9,50	1,00	1,00	1,00	1,00	1,00	1,00	0,50	1,00	1,00	1,00
6	Đã hoàn thành	2020-04-06 09:59:00	2020-04-06 10:01:00	2 phút 48 giây	10,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
7	Đã hoàn thành	2020-04-06 10:34:00	2020-04-06 10:39:00	4 phút 16 giây	9,50	1,00	1,00	1,00	1,00	1,00	1,00	0,50	1,00	1,00	1,00
8	Đã hoàn thành	2020-04-06 10:39:00	2020-04-06 10:40:00	38 giây	10,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
...

Showing 1 to 10 of 551 entries, 16 total columns

File 12:

ID	Status	Stime	Etime	Time	Diem	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
1	Đã hoàn thành	2020-04-12 08:38:00	2020-04-12 08:42:00	4 phút 38 giây	9,00	1,00	1,00	1,00	0,00	1,00	1,00	1,00	1,00	1,00	1,00
2	Đã hoàn thành	2020-04-12 08:44:00	2020-04-12 08:45:00	39 giây	10,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
3	Đã hoàn thành	2020-04-12 08:52:00	2020-04-12 08:53:00	1 phút 4 giây	9,00	1,00	1,00	1,00	0,00	1,00	1,00	1,00	1,00	1,00	1,00
4	Đã hoàn thành	2020-04-12 09:20:00	2020-04-12 09:23:00	2 phút 54 giây	10,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
5	Đã hoàn thành	2020-04-12 09:31:00	2020-04-12 09:33:00	1 phút 47 giây	9,00	1,00	1,00	1,00	0,00	1,00	1,00	1,00	1,00	1,00	1,00
6	Đã hoàn thành	2020-04-12 09:38:00	2020-04-12 09:39:00	40 giây	10,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
7	Đã hoàn thành	2020-04-12 10:12:00	2020-04-12 10:13:00	1 phút 27 giây	10,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
8	Đã hoàn thành	2020-04-12 10:46:00	2020-04-12 10:50:00	4 phút	10,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
...

Showing 1 to 10 of 353 entries, 16 total columns

3. Ta tạo vector ID để lưu lại ID của tất cả sinh viên có trong c:

```
ID=as.factor(c$ID)
ID <- ID[!is.na(ID)]
ID=levels(ID)
ID=as.numeric(ID)
```

Ta thu được vector ID:

File 6:

```
> ID
 [1] 1511191 1613010 1812257 1812477 1812478 1813096 1813528 1813681 1814096 1814518 1814611 1820028 1852443 1910006 1910032
 [16] 1910038 1910060 1910076 1910094 1910101 1910110 1910113 1910123 1910137 1910198 1910202 1910224 1910238 1910265 1910276
 [31] 1910298 1910339 1910346 1910347 1910351 1910402 1910409 1910473 1910563 1910565 1910620 1910643 1910644 1910650 1910663
 [46] 1910666 1910735 1910865 1910892 1910916 1910984 1911000 1911015 1911044 1911056 1911058 1911066 1911105 1911110 1911136
 [61] 1911185 1911186 1911207 1911217 1911262 1911283 1911285 1911296 1911314 1911363 1911441 1911456 1911478 1911520
 [76] 1911561 1911565 1911569 1911591 1911594 1911650 1911704 1911736 1911796 1911837 1911841 1911878 1911881 1911900 1911907
 [91] 1911931 1911975 1912041 1912046 1912056 1912084 1912123 1912184 1912190 1912237 1912267 1912288 1912371 1912384 1912386
 [106] 1912410 1912457 1912463 1912522 1912523 1912526 1912539 1912579 1912594 1912602 1912675 1912676 1912677 1912683 1912700
 [121] 1912705 1912713 1912715 1912749 1912761 1912798 1912811 1912817 1912912 1912916 1912954 1912958 1912966 1912980 1913014
 [136] 1913021 1913026 1913032 1913040 1913045 1913048 1913075 1913094 1913102 1913114 1913123 1913167 1913186 1913218 1913228
 [151] 1913241 1913254 1913260 1913261 1913268 1913306 1913334 1913336 1913341 1913354 1913355 1913356 1913380 1913386 1913396
 [166] 1913418 1913419 1913424 1913430 1913433 1913446 1913457 1913464 1913467 1913560 1913566 1913598 1913609 1913621 1913629
 [181] 1913651 1913652 1913678 1913695 1913713 1913729 1913756 1913758 1913763 1913764 1913775 1913817 1913828 1913832 1913844
 [196] 1913917 1913918 1913944 1913949 1913990 1914003 1914011 1914022 1914038 1914047 1914052 1914054 1914055 1914064 1914078
 [211] 1914079 1914084 1914093 1914121 1914126 1914210 1914220 1914227 1914232 1914291 1914310 1914316 1914352 1914384 1914405
 [226] 1914424 1914472 1914474 1914477 1914641 1914651 1914659 1914661 1914674 1914677 1914685 1914698 1914697 1914704 1914713
 [241] 1914720 1914738 1914763 1914768 1914802 1914806 1914807 1914830 1914845 1914864 1914878 1914880 1914881 1914900 1914914
 [256] 1914979 1915016 1915040 1915063 1915071 1915076 1915083 1915094 1915130 1915133 1915146 1915161 1915251 1915268 1915275
 [271] 1915294 1915323 1915329 1915350 1915351 1915378 1915439 1915442 1915470 1915473 1915474 1915482 1915486 1915520 1915540
 [286] 1915541 1915551 1915557 1915562 1915570 1915578 1915598 1915650 1915651 1915656 1915667 1915745 1915775 1915787 1915795 1915822
 [301] 1915865 1915866 1915882 1915903 1915905 1915919 1915928 1915931 1915939 1915940 1915953 1915982 1915983 1915988 1915991
 [316] 1916022 1916061 1927007 1936024 1937019 1937024 1937064
```

File 9:

```
> ID
[1] 1511191 1613010 1812257 1812477 1812478 1813096 1813528 1813681 1814096 1814518 1820028 1852443 1910006 1910032 1910038
[16] 1910060 1910076 1910094 1910101 1910110 1910113 1910123 1910137 1910198 1910202 1910224 1910238 1910265 1910276 1910298
[31] 1910339 1910346 1910347 1910351 1910402 1910409 1910473 1910565 1910620 1910643 1910644 1910650 1910663 1910666 1910735
[46] 1910865 1910892 1910916 1910984 1911000 1911015 1911044 1911056 1911058 1911066 1911105 1911110 1911136 1911185 1911186
[61] 1911207 1911217 1911262 1911283 1911285 1911296 1911314 1911363 1911441 1911456 1911478 1911520 1911530 1911561 1911565
[76] 1911569 1911591 1911594 1911650 1911704 1911736 1911796 1911837 1911841 1911878 1911881 1911900 1911907 1911931 1911975
[91] 1912041 1912046 1912056 1912084 1912123 1912184 1912190 1912237 1912267 1912288 1912371 1912384 1912386 1912410 1912457
[106] 1912463 1912522 1912523 1912526 1912539 1912579 1912594 1912602 1912675 1912676 1912683 1912700 1912705 1912713 1912715
[121] 1912749 1912761 1912798 1912811 1912817 1912912 1912916 1912954 1912966 1912980 1913014 1913021 1913026 1913040 1913045
[136] 1913048 1913075 1913094 1913102 1913114 1913123 1913167 1913186 1913218 1913241 1913254 1913260 1913261 1913268 1913306
[151] 1913334 1913336 1913341 1913354 1913355 1913356 1913380 1913386 1913396 1913418 1913419 1913424 1913430 1913433 1913446
[166] 1913457 1913464 1913467 1913560 1913566 1913599 1913609 1913621 1913629 1913651 1913652 1913678 1913695 1913713 1913729
[181] 1913763 1913775 1913817 1913828 1913830 1913844 1913944 1913949 1913950 1913959 1913960 1913961 1913962 1913963 1914003
[196] 1914047 1914052 1914054 1914055 1914064 1914078 1914079 1914084 1914093 1914121 1914126 1914210 1914220 1914227 1914232
[211] 1914291 1914310 1914316 1914352 1914384 1914405 1914424 1914472 1914474 1914477 1914641 1914651 1914659 1914661 1914674
[226] 1914677 1914685 1914698 1914704 1914713 1914720 1914738 1914763 1914768 1914802 1914806 1914807 1914830 1914845 1914864
[241] 1914878 1914880 1914881 1914914 1914979 1915016 1915040 1915063 1915071 1915076 1915083 1915130 1915133 1915146 1915251
[256] 1915268 1915275 1915294 1915323 1915329 1915350 1915351 1915378 1915439 1915442 1915470 1915473 1915474 1915482 1915486
[271] 1915520 1915540 1915541 1915551 1915557 1915562 1915570 1915578 1915598 1915651 1915656 1915656 1915745 1915775 1915795 1915822
[286] 1915865 1915866 1915882 1915903 1915905 1915919 1915931 1915939 1915940 1915953 1915982 1915983 1915988 1915991 1916022
[301] 1916061 1927007 1936024 1937019 1937024 1937064
```

File 10:

```
> ID
[1] 1511191 1613010 1812257 1812477 1812478 1813096 1813681 1814096 1814518 1820028 1852443 1910006 1910032 1910038
[14] 1910038 1910060 1910076 1910094 1910101 1910110 1910113 1910123 1910137 1910198 1910202 1910224 1910238 1910265
[27] 1910276 1910298 1910339 1910346 1910347 1910351 1910402 1910409 1910473 1910565 1910620 1910643 1910644
[40] 1910650 1910663 1910666 1910865 1910892 1910916 1910984 1911000 1911015 1911044 1911056 1911058 1911066
[53] 1911105 1911110 1911136 1911185 1911186 1911207 1911217 1911262 1911283 1911285 1911296 1911314 1911363
[66] 1911441 1911478 1911520 1911530 1911561 1911565 1911569 1911591 1911594 1911650 1911704 1911736 1911796
[79] 1911837 1911841 1911881 1911900 1911907 1911931 1912041 1912046 1912056 1912084 1912123 1912184 1912190
[92] 1912237 1912267 1912288 1912371 1912384 1912386 1912410 1912457 1912463 1912523 1912526 1912539 1912579
[105] 1912594 1912602 1912675 1912676 1912683 1912700 1912705 1912713 1912715 1912749 1912761 1912798 1912811
[118] 1912817 1912912 1912916 1912954 1912966 1912980 1913014 1913021 1913026 1913040 1913045 1913048 1913075
[131] 1913094 1913102 1913114 1913123 1913167 1913186 1913218 1913241 1913254 1913261 1913268 1913306 1913334
[144] 1913336 1913341 1913354 1913355 1913356 1913380 1913386 1913396 1913418 1913419 1913424 1913430 1913433
[157] 1913446 1913457 1913464 1913467 1913560 1913566 1913599 1913609 1913621 1913629 1913651 1913652 1913678
[170] 1913695 1913713 1913729 1913763 1913764 1913775 1913817 1913828 1913832 1913844 1913918 1913944 1913949
[183] 1914003 1914011 1914022 1914038 1914047 1914052 1914054 1914055 1914078 1914079 1914084 1914093 1914121
[196] 1914126 1914210 1914220 1914227 1914232 1914291 1914310 1914316 1914352 1914384 1914405 1914424 1914472
[209] 1914474 1914477 1914641 1914651 1914659 1914661 1914674 1914677 1914685 1914698 1914704 1914713 1914720
[222] 1914738 1914763 1914768 1914807 1914830 1914845 1914864 1914880 1914881 1914914 1914979 1915016 1915040
[235] 1915063 1915071 1915076 1915083 1915130 1915133 1915146 1915251 1915268 1915275 1915294 1915323 1915329
[248] 1915350 1915351 1915378 1915439 1915442 1915470 1915473 1915474 1915482 1915486 1915520 1915540 1915541
[261] 1915551 1915557 1915562 1915570 1915598 1915651 1915656 1915745 1915775 1915787 1915795 1915822 1915865
[274] 1915866 1915886 1915903 1915905 1915919 1915940 1915953 1915982 1915983 1915988 1915991 1916022 1927007
[287] 1936024 1937024 1937064
```

File 12:

```
> ID
[1] 1511191 1613010 1812257 1812477 1812478 1813096 1813681 1814096 1814518 1820028 1852443 1910006 1910032 1910038
[14] 1910038 1910060 1910076 1910094 1910101 1910110 1910113 1910123 1910137 1910198 1910202 1910224 1910238 1910265
[27] 1910276 1910298 1910339 1910346 1910347 1910351 1910402 1910409 1910473 1910565 1910620 1910643 1910644
[40] 1910650 1910663 1910666 1910865 1910892 1910916 1910984 1911000 1911015 1911056 1911058 1911110
[53] 1911136 1911185 1911186 1911207 1911217 1911262 1911283 1911285 1911296 1911314 1911363 1911441 1911478
[66] 1911520 1911530 1911565 1911569 1911591 1911594 1911650 1911704 1911736 1911796 1911837 1911841 1911881
[79] 1911900 1911907 1911931 1912041 1912046 1912056 1912084 1912123 1912184 1912190 1912237 1912267 1912288
[92] 1912371 1912384 1912386 1912410 1912457 1912463 1912523 1912579 1912594 1912602 1912675 1912676 1912683
[105] 1912683 1912700 1912713 1912715 1912749 1912761 1912798 1912811 1912817 1912912 1912916 1912966 1912980
[118] 1913014 1913021 1913026 1913040 1913045 1913048 1913075 1913094 1913102 1913114 1913123 1913167 1913186
[131] 1913218 1913241 1913254 1913261 1913268 1913306 1913334 1913341 1913354 1913356 1913380 1913386
[144] 1913386 1913396 1913418 1913419 1913424 1913430 1913433 1913446 1913457 1913464 1913467 1913476
[157] 1913599 1913609 1913621 1913651 1913652 1913678 1913695 1913713 1913729 1913763 1913764 1913775 1913817
[170] 1913828 1913832 1913844 1913918 1913944 1913949 1914003 1914011 1914022 1914038 1914047 1914052 1914054
[183] 1914055 1914078 1914084 1914093 1914121 1914126 1914210 1914220 1914227 1914232 1914291 1914310
[196] 1914316 1914352 1914384 1914405 1914424 1914472 1914474 1914477 1914641 1914651 1914659 1914661 1914674
[209] 1914677 1914685 1914698 1914704 1914713 1914720 1914738 1914763 1914768 1914807 1914830 1914845 1914864
[222] 1914880 1914881 1914914 1914979 1915016 1915040 1915063 1915071 1915076 1915083 1915130 1915133 1915146
[235] 1915251 1915268 1915275 1915294 1915323 1915329 1915350 1915351 1915378 1915439 1915442 1915473 1915474
[248] 1915482 1915486 1915520 1915540 1915541 1915551 1915557 1915562 1915570 1915598 1915651 1915656 1915745
[261] 1915775 1915787 1915795 1915822 1915865 1915866 1915882 1915903 1915919 1915953 1915982 1915983 1915988
[274] 1915991 1916022 1916061 1927007 1936024 1937024 1937064
```

4. Tạo vecto solannop để ghi nhận số lần nộp bài của từng ID rồi tạo một data frame có tên là data để chứa ID và số lần nộp bài tương ứng:

```
solannop=rep(0,times=length(ID))
for(i in 1:length(ID)){
for(j in 1:length(c$ID)){
if(c$ID[j]== ID[i]) {solannop[i]=solannop[i]+1}
}
}
data=data.frame(ID,solannop)
```

Kết quả được data như sau:

File 6:

	ID	solannop
1	1511191	1
2	1613010	2
3	1812257	1
4	1812477	2
5	1812478	1
6	1813096	2
7	1813528	1
8	1813681	1
9	1814096	2
10	1814518	1
11	1814611	1

Showing 1 to 12 of 322 entries, 2 total columns

File 9:

	ID	solannop
1	1511191	3
2	1613010	2
3	1812257	1
4	1812477	2
5	1812478	1
6	1813096	1
7	1813528	2
8	1813681	1
9	1814096	3
10	1814518	2

Showing 1 to 11 of 306 entries, 2 total columns

File 10:

	ID	solannop
1	1511191	3
2	1613010	3
3	1812257	1
4	1812477	2
5	1812478	1
6	1813096	2
7	1813681	1
8	1814096	2

Showing 1 to 9 of 289 entries, 2 total columns

File 12:

	ID	solannop
1	1511191	2
2	1613010	1
3	1812257	1
4	1812477	2
5	1812478	1
6	1813096	1

Showing 1 to 7 of 280 entries, 2 total columns

5. Tạo vecto landaunop chứa thời điểm lần đầu nộp bài của từng ID rồi thêm vào data một cột là landaunop với dữ liệu trong vecto landaunop:

```
landaunop=c()
for(i in 1:length(ID)){
  for(j in 1:length(c$ID)){
    if(ID[i]==c$ID[j]){
      landaunop[i]=c$Etime[j]
      break
    }
  }
}
landaunop=as.POSIXct(landaunop,tz = "GMT",origin, format = "%d %B %Y %I:%M %p")
data=cbind(data,landaunop)
```

Kết quả data sau khi thêm cột landaunop:

File 6:

ID	solannop	landaunop
1	1511191	1 2020-04-04 08:19:00
2	1613010	2 2020-04-19 16:32:00
3	1812257	1 2020-04-20 06:19:00
4	1812477	2 2020-04-23 00:44:00
5	1812478	1 2020-04-06 09:59:00
6	1813096	2 2020-04-12 06:17:00
7	1813528	1 2020-05-05 23:14:00
8	1813681	1 2020-05-11 06:57:00
9	1814096	2 2020-03-30 09:09:00
10	1814510	1 2020-04-28 01:17:00

Showing 1 to 12 of 322 entries, 3 total columns

File 9:

ID	solannop	landaunop
1	1511191	3 2020-04-04 13:16:00
2	1613010	2 2020-04-19 16:38:00
3	1812257	1 2020-04-20 06:52:00
4	1812477	2 2020-04-24 07:17:00
5	1812478	1 2020-04-06 10:41:00
6	1813096	1 2020-04-13 03:08:00
7	1813528	2 2020-05-05 23:41:00
8	1813681	1 2020-05-18 01:37:00
9	1814096	3 2020-04-12 07:13:00

Showing 1 to 11 of 306 entries, 3 total columns

File 10:

ID	solannop	landaunop
1	1511191	3 2020-04-20 06:06:00
2	1613010	3 2020-04-27 10:09:00
3	1812257	1 2020-04-20 10:32:00
4	1812477	2 2020-04-24 11:51:00
5	1812478	1 2020-04-06 10:43:00
6	1813096	2 2020-05-06 02:50:00
7	1813681	1 2020-05-18 01:52:00

Showing 1 to 9 of 289 entries, 3 total columns

File 12:

ID	solannop	landaunop
1	1511191	2 2020-04-20 06:25:00
2	1613010	1 2020-04-27 10:18:00
3	1812257	1 2020-04-20 10:45:00
4	1812477	2 2020-04-24 13:05:00
5	1812478	1 2020-04-13 05:33:00

Showing 1 to 7 of 280 entries, 3 total columns



6. Tạo vecto diemlandau và diemlanhai để lưu lại số điểm của từng ID ở từng lần nộp bài rồi thêm vào data hai cột là diemlandau và diemlanhai với dữ liệu từ hai vecto trên (chỉnh sửa các cột điểm sao cho những giá trị như NA hoặc ‘-’ thì trở thành 0):

```
diemlandau=c()
for(i in 1:length(data$ID)){
  for(j in 1:length(c$ID)){
    if(data$ID[i]==c$ID[j]){
      diemlandau[i]=c$Diem[j]
      break
    }
  }
  for(i in 1:length(data$ID)){
    for(j in 1:length(c$ID)){
      if(data$ID[i]==c$ID[j] && diemlandau[i]=="-"){
        diemlandau[i]=c$Diem[j]
        break
      }
    }
  }
  data=cbind(data,diemlandau)
  diemlanhai=c()
  for(i in 1:length(data$ID)){
    k=0
    for (j in 1:length(c$ID)){
      if(data$solannop[i]>1){
        if(!is.na(c$Diem[j])){
          if(data$ID[i]==c$ID[j]){k=k+1}
          if(k==2){
            if(c$Diem[j]=='-'){diemlandai[i]=0}
            else{diemlanhai[i]=c$Diem[j]}
            break
          }
        }
      }
    }
    else{diemlanhai[i]=0}
  }
  data=cbind(data,diemlanhai)
  for(i in 1:length(data$diemlandau)){
    if(!is.na(data$diemlandau[i])){
      if(data$diemlandau[i]=="-"){data$diemlandau[i]=0}
    }
  }
  for(i in 1:length(data$diemlanhai)){
    if(!is.na(data$diemlanhai[i])){
      if(data$diemlanhai[i]=="-")data$diemlanhai[i]=0
    }
  }
  data$diemlandau=scan(text=data$diemlandau, dec=",", sep=".")
  data$diemlanhai=scan(text=data$diemlanhai, dec=",", sep=".")
}
```

Kết quả data:

File 6:



	ID	solannop	landaunop	diemlandau	diemlanhai	
1	1511191		1	2020-04-04 08:19:00	10.0	0.0
2	1613010		2	2020-04-19 16:32:00	7.5	9.0
3	1812257		1	2020-04-20 06:19:00	10.0	0.0
4	1812477		2	2020-04-23 00:44:00	9.0	10.0
5	1812478		1	2020-04-06 09:59:00	8.5	0.0
6	1813096		2	2020-04-12 08:17:00	9.5	10.0
7	1813528		1	2020-05-05 23:14:00	10.0	0.0
8	1813681		1	2020-05-11 06:57:00	9.0	0.0
9	1814096		2	2020-03-30 09:09:00	9.5	10.0
10	1814510		1	2020-04-28 01:17:00	10.0	0.0

Showing 1 to 12 of 322 entries, 5 total columns

File 9:

	ID	solannop	landaunop	diemlandau	diemlanhai	
1	1511191		3	2020-04-04 13:16:00	8.0	9.0
2	1613010		2	2020-04-19 16:38:00	7.0	10.0
3	1812257		1	2020-04-20 06:52:00	10.0	0.0
4	1812477		2	2020-04-24 07:17:00	7.0	10.0
5	1812478		1	2020-04-06 10:41:00	7.5	0.0
6	1813096		1	2020-04-13 03:08:00	9.0	0.0
7	1813528		2	2020-05-05 23:41:00	7.0	10.0
8	1813681		1	2020-05-18 01:37:00	8.0	0.0
9	1814096		3	2020-04-12 07:13:00	9.0	9.0

Showing 1 to 11 of 306 entries, 5 total columns

File 10:

	ID	solannop	landaunop	diemlandau	diemlanhai	
1	1511191		3	2020-04-20 06:06:00	9.0	9.5
2	1613010		3	2020-04-27 10:09:00	9.0	9.5
3	1812257		1	2020-04-20 10:32:00	9.0	0.0
4	1812477		2	2020-04-24 11:51:00	8.0	10.0
5	1812478		1	2020-04-06 10:43:00	8.5	0.0
6	1813096		2	2020-05-06 02:50:00	9.0	10.0
7	1813681		1	2020-05-18 01:52:00	10.0	0.0

Showing 1 to 9 of 289 entries, 5 total columns

File 12:



ID	solanop	landaunop	diemlandau	diemlanhai
1	1511191	2	2020-04-20 06:25:00	9
2	1613010	1	2020-04-27 10:18:00	10
3	1812257	1	2020-04-20 10:45:00	10
4	1812477	2	2020-04-24 13:05:00	7
5	1812478	1	2020-04-13 05:33:00	10
6	1813096	1	2020-05-06 04:38:00	10
7	1813681	1	2020-05-18 02:42:00	9

Showing 1 to 9 of 280 entries, 5 total columns

7. Ta tìm t2 là 2/3 khoảng thời gian giữa thời điểm sinh viên đầu tiên và thời điểm sinh viên cuối nộp bài:

```
tmax=max(c$Etime,na.rm = T)
tmin=min(c$Etime,na.rm = T)
hours=as.period(interval(tmin, tmax), unit = "hours")
sogiocongthem=hour(hours)-round(hour(hours)/3)
t2=tmin+hours(sogiocongthem)
```

Ta tìm được t2:

File 6:

```
> t2
[1] "2020-05-01 21:05:00 +07"
```

File 9:

```
> t2
[1] "2020-05-04 02:07:00 +07"
```

File 10:

```
> t2
[1] "2020-05-04 17:38:00 +07"
```

File 12:

```
> t2
[1] "2020-05-06 17:42:00 +07"
```

8. Tạo data7 là một data frame con của data chứa dữ liệu của những ID có thời gian lần đầu nộp sau và đổi tên cột ID của data7 thành ID7::

```
data7=subset(data, landaunop>t2)
names(data7)[1]<- "ID7"
```

Ta được data7 như sau:

File 6:



	ID7	solannop	landaunop	diemlandau	diemlanhai
7	1813528	1	2020-05-05 23:14:00	10.0	0.0
8	1813681	1	2020-05-11 06:57:00	9.0	0.0
12	1820028	1	2020-05-06 18:34:00	10.0	0.0
14	1910006	2	2020-05-04 16:27:00	9.5	10.0
17	1910060	1	2020-05-11 13:48:00	10.0	0.0
25	1910198	1	2020-05-04 05:42:00	10.0	0.0
29	1910265	1	2020-05-07 14:54:00	10.0	0.0
39	1910563	1	2020-05-17 09:48:00	10.0	0.0
45	1910663	1	2020-05-12 08:26:00	10.0	0.0
47	1910725	1	2020-05-13 10:13:00	10.0	0.0

Showing 1 to 12 of 46 entries, 5 total columns

File 9:

	ID7	solannop	landaunop	diemlandau	diemlanhai
7	1813528	2	2020-05-05 23:41:00	7.0	10.0
8	1813681	1	2020-05-18 01:37:00	8.0	0.0
11	1820028	1	2020-05-06 18:52:00	9.0	0.0
13	1910006	2	2020-05-11 13:45:00	9.0	10.0
16	1910060	2	2020-05-11 14:16:00	9.0	10.0
24	1910198	2	2020-05-04 07:24:00	8.0	10.0
28	1910265	3	2020-05-07 15:22:00	9.0	9.5
32	1910346	1	2020-05-10 07:45:00	10.0	0.0
35	1910402	2	2020-05-04 10:49:00	9.0	10.0

Showing 1 to 11 of 58 entries, 5 total columns

File 10:

	ID7	solannop	landaunop	diemlandau	diemlanhai
6	1813096	2	2020-05-06 02:50:00	9.0	10.0
7	1813681	1	2020-05-18 01:52:00	10.0	0.0
10	1820028	2	2020-05-06 18:55:00	8.0	10.0
11	1852443	2	2020-05-06 11:42:00	9.5	10.0
12	1910006	3	2020-05-11 13:48:00	8.5	9.5
15	1910060	3	2020-05-11 14:19:00	9.5	9.5
24	1910224	1	2020-05-12 09:27:00	9.5	0.0

Showing 1 to 9 of 70 entries, 5 total columns

File 12:



	ID7	solanop	landaunop	diemlandau	diemlanhai
7	1813681	1	2020-05-18 02:42:00	9	0
10	1820028	1	2020-05-06 18:59:00	10	0
11	1852443	2	2020-05-08 12:01:00	9	10
12	1910006	1	2020-05-11 13:59:00	10	0
15	1910060	1	2020-05-11 14:31:00	10	0
16	1910076	1	2020-05-08 14:05:00	10	0
24	1910224	1	2020-05-14 08:05:00	10	0

Showing 1 to 9 of 57 entries, 5 total columns

9. Số lượng sinh viên đồi phó là số ID có trong data7::

```
sosvdoiph0=length(data7$ID7)
```

Kết quả được như sau:

File 6:

```
> sosvdoiph0  
[1] 46
```

File 9:

```
> sosvdoiph0  
[1] 58
```

File 10:

```
> sosvdoiph0  
[1] 70
```

File 12:

```
> sosvdoiph0  
[1] 57
```

10. Chuyển cột Diem trong c từ class character về class numeric để tiện tính toán:

```
for(i in 1:length(c$Diem))  
if(!is.na(c$Diem[i]))  
if(c$Diem[i]=='-')c$Diem[i]=0  
  
c$Diem=scan(text=c$Diem, dec=",", sep=".")
```

11. Tạo ID9 chứa các ID của sinh viên có số điểm lần đầu hoặc số điểm lần hai lớn hơn hoặc bằng 8:

```
ID9=c()
k=0
for(i in 1:length(data$ID)){
  if(data$diemlandau[i]>=8 || data$diemlanhai[i]>=8){
    k=k+1
    ID9[k]=data$ID[i]
  }
}
```

Vector ID9 như sau:

File 6:

```
> ID9
[1] 1511191 1613010 1812257 1812477 1812478 1813096 1813528 1813681 1814096 1814518 1814611 1820028 1852443 1910006 1910032
[16] 1910038 1910060 1910076 1910094 1910101 1910110 1910113 1910123 1910137 1910198 1910202 1910224 1910238 1910265 1910276
[31] 1910298 1910339 1910346 1910347 1910351 1910402 1910409 1910473 1910563 1910565 1910620 1910643 1910644 1910650 1910663
[46] 1910666 1910735 1910865 1910892 1910916 1910984 1911000 1911015 1911044 1911056 1911058 1911066 1911103 1911110 1911136
[61] 1911185 1911186 1911207 1911217 1911262 1911285 1911296 1911314 1911363 1911441 1911456 1911478 1911520 1911530 1911561
[76] 1911565 1911569 1911591 1911594 1911650 1911704 1911736 1911796 1911837 1911841 1911878 1911881 1911900 1911907 1911931
[91] 1911975 1912041 1912046 1912056 1912084 1912123 1912184 1912190 1912237 1912267 1912288 1912371 1912384 1912386 1912410
[106] 1912457 1912463 1912522 1912523 1912526 1912539 1912559 1912594 1912602 1912675 1912676 1912683 1912700 1912705
[121] 1912713 1912715 1912749 1912761 1912798 1912811 1912817 1912912 1912916 1912954 1912958 1912966 1912980 1913014 1913021
[136] 1913026 1913034 1913040 1913045 1913048 1913075 1913094 1913102 1913114 1913123 1913167 1913186 1913218 1913228 1913241
[151] 1913254 1913260 1913261 1913268 1913306 1913334 1913336 1913341 1913354 1913355 1913356 1913380 1913386 1913396 1913418
[166] 1913419 1913424 1913430 1913433 1913446 1913457 1913464 1913467 1913560 1913566 1913599 1913609 1913621 1913651 1913652
[181] 1913652 1913678 1913695 1913713 1913729 1913756 1913758 1913763 1913764 1913775 1913817 1913828 1913832 1913844 1913917
[196] 1913918 1913944 1913949 1913990 1914003 1914011 1914022 1914038 1914047 1914052 1914054 1914064 1914078 1914079
[211] 1914084 1914093 1914121 1914126 1914210 1914220 1914227 1914232 1914291 1914310 1914316 1914352 1914384 1914405 1914424
[226] 1914472 1914474 1914477 1914641 1914651 1914659 1914661 1914674 1914677 1914685 1914697 1914698 1914704 1914713 1914720
[241] 1914738 1914763 1914768 1914802 1914806 1914807 1914830 1914845 1914864 1914876 1914880 1914881 1914900 1914914 1914979
[256] 1915016 1915040 1915063 1915065 1915071 1915076 1915130 1915133 1915140 1915161 1915251 1915268 1915275 1915294 1915323
[271] 1915329 1915350 1915351 1915378 1915439 1915442 1915470 1915473 1915474 1915478 1915482 1915520 1915540 1915551
[286] 1915557 1915562 1915570 1915598 1915650 1915651 1915667 1915745 1915775 1915787 1915795 1915822 1915865 1915866
[301] 1915882 1915903 1915905 1915919 1915928 1915931 1915939 1915940 1915953 1915982 1915983 1915988 1915991 1916022 1916061
[316] 1927007 1936024 1937019 1937024 1937064
```

File 9:

```
> ID9
[1] 1511191 1613010 1812257 1812477 1812478 1813096 1813528 1813681 1814096 1814518 1814611 1820028 1852443 1910006 1910032 1910060
[16] 1910038 1910060 1910076 1910094 1910101 1910110 1910113 1910123 1910137 1910198 1910202 1910224 1910238 1910265 1910276 1910298 1910339
[31] 1910346 1910347 1910351 1910402 1910409 1910473 1910563 1910620 1910643 1910644 1910650 1910663 1910666 1910735 1910865
[46] 1910892 1910916 1911000 1911015 1911044 1911056 1911058 1911066 1911105 1911130 1911136 1911185 1911186 1911207 1911217
[61] 1911262 1911283 1911285 1911296 1911314 1911363 1911441 1911456 1911478 1911520 1911530 1911561 1911565 1911569 1911591
[76] 1911594 1911650 1911704 1911736 1911796 1911837 1911841 1911878 1911900 1911907 1911931 1911975 1912041 1912046
[91] 1912056 1912084 1912123 1912184 1912190 1912237 1912267 1912288 1912371 1912384 1912386 1912410 1912457 1912463 1912522
[106] 1912523 1912526 1912539 1912559 1912594 1912602 1912676 1912683 1912700 1912705 1912713 1912749 1912761 1912761
[121] 1912798 1912811 1912817 1912912 1912916 1912954 1912966 1912980 1913014 1913021 1913026 1913045 1913048 1913075
[136] 1913094 1913102 1913114 1913123 1913167 1913186 1913218 1913241 1913254 1913260 1913263 1913268 1913303 1913334 1913336
[151] 1913341 1913354 1913355 1913356 1913380 1913386 1913396 1913418 1913419 1913424 1913430 1913433 1913446 1913457 1913464
[166] 1913467 1913560 1913566 1913599 1913609 1913621 1913629 1913651 1913652 1913678 1913695 1913713 1913729 1913763 1913764
[181] 1913751 1913817 1913828 1913832 1913832 1913844 1913918 1913944 1913949 1913990 1914003 1914011 1914022 1914038 1914052
[196] 1914054 1914055 1914064 1914078 1914079 1914084 1914093 1914121 1914126 1914210 1914220 1914227 1914232 1914291 1914310
[211] 1914316 1914352 1914384 1914405 1914424 1914427 1914474 1914477 1914478 1914481 1914486 1914486 1914487 1914488 1914489
[226] 1914698 1914704 1914713 1914720 1914738 1914763 1914768 1914768 1914780 1914786 1914786 1914786 1914786 1914786 1914786
[241] 1914881 1914914 1914979 1915016 1915040 1915063 1915073 1915076 1915076 1915130 1915133 1915146 1915251 1915268 1915275
[256] 1915294 1915323 1915350 1915351 1915378 1915439 1915442 1915470 1915473 1915474 1915478 1915482 1915520 1915540 1915540
[271] 1915541 1915551 1915557 1915562 1915570 1915598 1915651 1915656 1915667 1915745 1915775 1915787 1915795 1915822 1915865 1915866
[286] 1915882 1915903 1915905 1915919 1915928 1915931 1915939 1915940 1915953 1915982 1915983 1915988 1915991 1916022 1927007
[301] 1936024 1937019 1937024 1937064
```

File 10:

```
> ID9
[1] 1511191 1613010 1812257 1812477 1812478 1813096 1813528 1813681 1814096 1814518 1814611 1820028 1852443 1910006 1910032
[14] 1910038 1910060 1910076 1910094 1910101 1910110 1910113 1910123 1910137 1910198 1910202 1910224 1910238 1910265
[27] 1910276 1910298 1910339 1910346 1910347 1910351 1910402 1910409 1910473 1910478 1910563 1910565 1910620 1910643 1910644 1910644
[40] 1910650 1910663 1910666 1910865 1910892 1910916 1910984 1911000 1911015 1911044 1911044 1911056 1911058 1911066
[53] 1911105 1911110 1911136 1911185 1911186 1911207 1911217 1911262 1911283 1911285 1911296 1911314 1911363
[66] 1911441 1911478 1911520 1911530 1911561 1911565 1911569 1911591 1911594 1911650 1911704 1911736 1911796
[79] 1911837 1911841 1911881 1911900 1911907 1911931 1912041 1912046 1912056 1912084 1912123 1912184 1912190
[92] 1912227 1912267 1912288 1912371 1912384 1912386 1912410 1912457 1912463 1912523 1912526 1912539 1912579
[105] 1912594 1912602 1912675 1912683 1912700 1912705 1912713 1912749 1912761 1912761 1912798 1912811
[118] 1912817 1912912 1912954 1912966 1912980 1913014 1913021 1913026 1913040 1913045 1913048 1913075
[131] 1913094 1913102 1913114 1913123 1913167 1913186 1913218 1913241 1913254 1913261 1913268 1913306 1913334
[144] 1913336 1913341 1913354 1913355 1913356 1913386 1913386 1913396 1913418 1913419 1913424 1913430 1913433
[157] 1913446 1913446 1913457 1913464 1913467 1913467 1913468 1913468 1913469 1913469 1913469 1913469 1913469 1913469
[170] 1913695 1913713 1913729 1913763 1913764 1913775 1913817 1913828 1913832 1913844 1913844 1913844 1913844 1913849
[183] 1914003 1914011 1914022 1914038 1914047 1914052 1914054 1914055 1914078 1914079 1914084 1914084 1914093
[196] 1914126 1914210 1914220 1914227 1914232 1914291 1914310 1914316 1914352 1914384 1914405 1914424 1914472
[209] 19144474 1914477 1914461 19144651 19144659 19144661 19144674 19144677 19144685 19144685 19144698 19144704 19144713 1914720
[222] 1914738 1914763 1914768 1914807 1914830 1914836 1914845 1914846 1914880 1914881 1914914 1914914 1914979 1915016
[235] 1915063 1915071 1915076 1915083 1915130 1915133 1915146 1915251 1915268 1915275 1915294 1915323 1915329
[248] 1915350 1915351 1915378 1915439 1915442 1915470 1915473 1915474 1915474 1915482 1915486 1915520 1915540 1915541
[261] 1915551 1915557 1915562 1915570 1915598 1915651 1915656 1915745 1915775 1915787 1915795 1915822 1915865
[274] 1915866 1915882 1915903 1915905 1915919 1915940 1915953 1915982 1915983 1915988 1915991 1916022 1927007
[287] 1936024 1937024 1937064
```

File 12:

```
> ID9
[1] 1511191 1613010 1812257 1812477 1812478 1813096 1813681 1814096 1814518 1820028 1852443 1910006 1910032
[14] 1910038 1910060 1910076 1910094 1910101 1910110 1910113 1910123 1910137 1910202 1910224 1910238 1910265
[27] 1910276 1910298 1910339 1910346 1910347 1910351 1910402 1910409 1910473 1910563 1910620 1910643 1910644
[40] 1910650 1910663 1910666 1910865 1910892 1910916 1910984 1911000 1911015 1911056 1911058 1911066 1911110
[53] 1911136 1911185 1911186 1911207 1911217 1911262 1911283 1911296 1911314 1911363 1911441 1911478
[66] 1911520 1911530 1911565 1911569 1911591 1911594 1911650 1911704 1911736 1911796 1911837 1911841 1911881
[79] 1911900 1911907 1911931 1912041 1912046 1912056 1912084 1912123 1912184 1912190 1912237 1912267 1912288
[92] 1912371 1912384 1912386 1912410 1912457 1912463 1912523 1912539 1912579 1912594 1912602 1912675 1912676
[105] 1912683 1912700 1912713 1912715 1912749 1912761 1912798 1912811 1912817 1912912 1912916 1912966 1912980
[118] 1913014 1913021 1913026 1913040 1913045 1913048 1913075 1913094 1913102 1913114 1913123 1913167 1913186
[131] 1913218 1913241 1913254 1913261 1913268 1913306 1913334 1913341 1913354 1913355 1913356 1913380
[144] 1913386 1913396 1913418 1913419 1913424 1913430 1913433 1913446 1913457 1913464 1913467 1913560 1913566
[157] 1913599 1913609 1913621 1913651 1913652 1913678 1913695 1913713 1913729 1913763 1913764 1913775 1913817
[170] 1913828 1913832 1913844 1913918 1913944 1913949 1914003 1914011 1914022 1914038 1914047 1914052 1914054
[183] 1914055 1914078 1914079 1914084 1914093 1914121 1914126 1914210 1914220 1914227 1914232 1914291 1914310
[196] 1914316 1914352 1914384 1914405 1914424 1914472 1914474 1914477 1914641 1914651 1914659 1914661 1914674
[209] 1914677 1914685 1914698 1914704 1914713 1914720 1914738 1914763 1914768 1914807 1914830 1914845 1914864
[222] 1914880 1914881 1914914 1914979 1915016 1915040 1915063 1915071 1915076 1915083 1915130 1915133 1915146
[235] 1915251 1915268 1915275 1915294 1915323 1915329 1915350 1915351 1915378 1915439 1915442 1915473 1915474
[248] 1915482 1915486 1915520 1915540 1915541 1915551 1915557 1915562 1915570 1915598 1915651 1915656 1915745
[261] 1915775 1915787 1915795 1915822 1915865 1915866 1915882 1915903 1915919 1915953 1915982 1915983 1915988
[274] 1915991 1916022 1916061 1927007 1936024 1937024 1937064
```

12. Số lượng sinh viên thông minh là độ dài ID9:

```
sosinhvienthongminh=length(ID9)
```

Kết quả được như sau:

File 6:

```
> sosinhvienthongminh
[1] 320
```

File 9:

```
> sosinhvienthongminh
[1] 304
```

File 10:

```
> sosinhvienthongminh
[1] 289
```

File 12:

```
> sosinhvienthongminh
[1] 280
```

13. Tìm t1 là biến lưu thời điểm t1:

```
hours=as.period(interval(tmin, tmax), unit = "hours")
sogiocongthem=hour(hours)-round(hour(hours)*2/3)
t1=tmin+hours(sogiocongthem)
```

Vậy t1 :

File 6:

```
> t1
[1] "2020-04-15 03:05:00 +07"
```

File 9:

```
> t1  
[1] "2020-04-19 06:07:00 +07"
```

File 10:

```
> t1  
[1] "2020-04-20 13:38:00 +07"
```

File 12:

```
> t1  
[1] "2020-04-24 12:42:00 +07"
```

14. Tạo vecto ID10 chứa tất cả ID của sinh viên chủ động:

```
ID10=c()  
k=0  
for(i in 1:length(data$ID)){  
if((data$landaunop[i]<t1||data$diemlandau[i]>=8||data$diemlanhai[i]>=8)&&data$solannop[i]>1){  
k=k+1  
ID10[k]=data$ID[i]  
}  
}
```

ID10 như sau:

File 6:

```
> ID10  
[1] 1613010 1812477 1813096 1814096 1910006 1910038 1910276 1910339 1910351 1910650 1910865 1911105 1911110 1911186  
[15] 1911363 1911561 1911796 1911881 1911900 1911907 1911975 1912046 1912056 1912237 1912386 1912410 1912579 1912676  
[29] 1912700 1912761 1912916 1912954 1912966 1913014 1913032 1913045 1913048 1913123 1913306 1913341 1913418 1913424  
[43] 1913433 1913560 1913609 1913651 1913756 1913775 1913817 1913844 1913917 1913918 1913944 1913949 1913990 1914003  
[57] 1914047 1914052 1914210 1914227 1914352 1914384 1914474 1914661 1914674 1914685 1914704 1914713 1914720 1914768  
[71] 1914802 1914979 1915040 1915083 1915133 1915268 1915350 1915378 1915439 1915470 1915562 1915570 1915598 1915775  
[85] 1915882 1915903 1915905 1915919 1915928 1915939 1915940 1915983 1916022 1927007 1936024 1937019
```

File 9:

```
> ID10  
[1] 1511191 1613010 1812477 1813528 1814096 1814518 1852443 1910006 1910060 1910113 1910123 1910137 1910198 1910224 1910238  
[16] 1910265 1910276 1910298 1910347 1910402 1910409 1910620 1910643 1910644 1910735 1910865 1911066 1911105 1911110 1911136  
[31] 1911185 1911217 1911285 1911363 1911520 1911561 1911565 1911650 1911736 1911841 1911907 1911931 1912046 1912237  
[46] 1912371 1912539 1912579 1912602 1912713 1912749 1912761 1912916 1912954 1912980 1913040 1913045 1913048 1913094 1913123  
[61] 1913167 1913186 1913260 1913306 1913341 1913356 1913396 1913418 1913424 1913430 1913467 1913560 1913609 1913678  
[76] 1913695 1913713 1913828 1913949 1914003 1914011 1914038 1914047 1914064 1914084 1914093 1914210 1914227 1914316 1914352  
[91] 1914405 1914424 1914472 1914477 1914641 1914659 1914661 1914674 1914685 1914698 1914704 1914720 1914738 1914763 1914802  
[106] 1914830 1914878 1914880 1914881 1914979 1915016 1915040 1915083 1915130 1915133 1915268 1915323 1915350 1915378  
[121] 1915470 1915520 1915551 1915562 1915570 1915598 1915651 1915775 1915795 1915822 1915866 1915903 1915931 1915939  
[136] 1915940 1915982 1915983 1916061 1927007 1936024 1937019 1937064
```

File 10:

```
> ID10  
[1] 1511191 1613010 1812477 1813096 1814096 1814518 1820028 1852443 1910006 1910060 1910101 1910110 1910113  
[14] 1910137 1910202 1910238 1910265 1910276 1910298 1910339 1910347 1910351 1910402 1910409 1910620 1910643  
[27] 1910644 1910650 1910663 1910666 1910865 1910892 1910916 1911015 1911044 1911056 1911066 1911105 1911136  
[40] 1911110 1911185 1911186 1911217 1911262 1911296 1911314 1911363 1911478 1911520 1911561 1911591 1911650  
[53] 1911736 1911796 1911881 1911900 1911907 1912046 1912056 1912084 1912184 1912237 1912267 1912288 1912371  
[66] 1912384 1912410 1912457 1912594 1912602 1912676 1912700 1912713 1912749 1912798 1912811 1912916  
[79] 1912954 1912980 1913014 1913021 1913040 1913048 1913075 1913094 1913114 1913123 1913167 1913186 1913254  
[92] 1913341 1913355 1913396 1913418 1913424 1913430 1913433 1913464 1913560 1913566 1913609 1913621 1913651  
[105] 1913713 1913763 1913764 1913775 1913828 1913832 1913844 1913918 1913949 1914003 1914038 1914047 1914052  
[118] 1914055 1914079 1914093 1914126 1914210 1914310 1914316 1914352 1914384 1914405 1914474 1914659 1914661  
[131] 1914674 1914677 1914685 1914698 1914704 1914713 1914720 1914738 1914763 1914768 1914830 1914845 1914864  
[144] 1914881 1914914 1914979 1915016 1915040 1915063 1915071 1915083 1915130 1915268 1915323 1915329 1915378  
[157] 1915442 1915470 1915520 1915540 1915541 1915570 1915598 1915651 1915656 1915745 1915775 1915822 1915865  
[170] 1915882 1915919 1915940 1915982 1915991 1916061 1927007 1936024
```

File 12:



```
> ID10
[1] 1511191 1812477 1852443 1910123 1910409 1910892 1911066 1911262 1911363 1911441 1912123 1912288 1912371
[14] 1912410 1912457 1912594 1912602 1912676 1912713 1912761 1912798 1913014 1913021 1913045 1913048 1913075
[27] 1913094 1913380 1913418 1913424 1913430 1913464 1913560 1913599 1913609 1913621 1913918 1914047 1914093
[40] 1914126 1914220 1914227 1914352 1914477 1914659 1914661 1914677 1914685 1914704 1914713 1914830 1915016
[53] 1915040 1915083 1915130 1915133 1915146 1915268 1915350 1915378 1915442 1915520 1915541 1915775 1915795
[66] 1915919 1915983 1915988 1927007 1936024
```

15. Số lượng sinh viên chủ động là độ dài vecto ID10:

```
sosinhvienchudong=length(ID10)
```

Kết quả được như sau:

File 6:

```
> sosinhvienchudong
[1] 96
```

File 9:

```
> sosinhvienchudong
[1] 143
```

File 10:

```
> sosinhvienchudong
[1] 177
```

File 12:

```
> sosinhvienchudong
[1] 70
```

16. Tạo g79 chứa ID của các sinh viên vừa thông minh vừa đồi phó và n79 là số lượng sinh viên vừa thông minh vừa đồi phó:

```
g79=c()
k=0
for(i in 1:length(data7$ID7)){
  for(j in 1:length(ID9)){
    if(data7$ID7[i]==ID9[j]){
      k=k+1
      g79[k]=ID9[j]
    }
  }
}
n79=length(g79)
```

Kết quả được như sau:

File 6:

```
> g79
[1] 1813528 1813681 1820028 1910006 1910060 1910198 1910265 1910563 1910663 1910735 1911110 1911285 1911530 1911565
[15] 1911594 1911975 1912384 1912526 1912705 1912713 1913040 1913045 1913167 1913419 1913430 1913446 1913457 1913560
[29] 1913629 1913678 1913990 1914011 1914232 1914477 1914651 1914661 1914685 1914713 1914830 1915161 1915650 1915667
[43] 1915787 1915866 1915931 1915983
> n79
[1] 46
```

File 9:



```
> g79
[1] 1813528 1813681 1820028 1910006 1910060 1910198 1910265 1910346 1910402 1910663 1910735 1911015 1911044 1911110 1911363
[16] 1911530 1911565 1911594 1911878 1911975 1912384 1912457 1912522 1912526 1912705 1912713 1913040 1913045 1913167 1913419
[31] 1913430 1913446 1913457 1913560 1913599 1913629 1913678 1913990 1914011 1914079 1914232 1914477 1914651 1914661 1914685
[46] 1914713 1914830 1915130 1915133 1915294 1915470 1915570 1915787 1915866 1915931 1915983 1916022
> n79
[1] 57
```

File 10:

```
> g79
[1] 1813096 1813681 1820028 1852443 1910006 1910060 1910224 1910265 1910346 1910347 1910402 1910984 1911015
[14] 1911044 1911110 1911217 1911285 1911296 1911363 1911530 1911565 1911594 1911900 1912267 1912384 1912457
[27] 1912526 1912705 1912713 1912954 1912966 1913040 1913045 1913167 1913354 1913419 1913424 1913430 1913446
[40] 1913457 1913560 1913599 1913629 1913678 1913828 1914047 1914232 1914477 1914651 1914661 1914685 1914713
[53] 1914763 1914830 1914914 1915016 1915130 1915133 1915294 1915350 1915470 1915570 1915598 1915656
[66] 1915787 1915866 1915940 1915983 1916022
> n79
[1] 70
```

File 12:

```
> g79
[1] 1813681 1820028 1852443 1910006 1910060 1910076 1910224 1910265 1910346 1910347 1910402 1910663 1911110
[14] 1911217 1911285 1911296 1911363 1911530 1911900 1912267 1912384 1912675 1912713 1912966 1913040 1913045
[27] 1913167 1913354 1913419 1913424 1913430 1913446 1913457 1913599 1913678 1913763 1913817 1913828 1914047
[40] 1914232 1914651 1914661 1914685 1914763 1914914 1915016 1915063 1915133 1915350 1915570 1915598 1915656
[53] 1915787 1915866 1915982 1915983 1916022
> n79
[1] 57
```

17. Tạo g710 chứa ID của các sinh viên vừa chủ động vừa đổi phó và n710 là số lượng sinh viên vừa chủ động vừa đổi phó:

```
g710=c()
k=0
for(i in 1:length(data7$ID7)){
  for(j in 1:length(ID10)){
    if(data7$ID7[i]==ID10[j]){
      k=k+1
      g710[k]=ID10[j]
    }
  }
}
n710=length(g710)
```

Kết quả như sau:

File 6:

```
> g710
[1] 1910006 1911110 1911975 1913045 1913560 1913990 1914661 1914685 1914713 1915983
> n710
[1] 10
```

File 9:

```
> g710
[1] 1813528 1910006 1910060 1910198 1910265 1910402 1910735 1911110 1911363 1911565 1912713 1913040 1913045 1913167 1913430
[16] 1913560 1913678 1914011 1914477 1914661 1914685 1914830 1915130 1915133 1915470 1915570 1915866 1915931 1915983
> n710
[1] 29
```

File 10:

```
> g710
[1] 1813096 1820028 1852443 1910006 1910060 1910265 1910347 1910402 1911015 1911044 1911110 1911217 1911296
[14] 1911363 1911900 1912267 1912384 1912457 1912713 1912954 1913040 1913167 1913424 1913430 1913560 1913828
[27] 1914047 1914661 1914685 1914713 1914763 1914830 1914914 1915016 1915130 1915470 1915570 1915598 1915656
[40] 1915940
> n710
[1] 40
```

File 12:



```
> g710
[1] 1852443 1911363 1912713 1913045 1913424 1913430 1913599 1914047 1914661 1914685 1915016 1915133 1915350
[14] 1915983
> n710
[1] 14
```

18. Tạo g910 chứa ID của các sinh viên vừa chủ động vừa thông minh và n910 là số lượng sinh viên vừa chủ động vừa thông minh:

```
g910=c()
k=0
for(i in 1:length(ID9)){
  for(j in 1:length(ID10)){
    if(ID9[i]==ID10[j]){
      k=k+1
      g910[k]=ID10[j]
    }
  }
}
n910=length(g910)
```

Kết quả thu được như sau:

File 6:

```
> g910
[1] 1613010 1812477 1813096 1814096 1910006 1910038 1910276 1910339 1910351 1910650 1910865 1911105 1911110 1911186
[15] 1911363 1911561 1911796 1911881 1911900 1911907 1911975 1912046 1912056 1912237 1912386 1912410 1912579 1912676
[29] 1912700 1912761 1912916 1912954 1912966 1913014 1913032 1913045 1913048 1913123 1913306 1913341 1913418 1913424
[43] 1913433 1913560 1913609 1913651 1913756 1913775 1913817 1913844 1913917 1913918 1913944 1913949 1913990 1914003
[57] 1914047 1914052 1914210 1914227 1914352 1914384 1914474 1914661 1914674 1914685 1914704 1914713 1914720 1914768
[71] 1914802 1914979 1915040 1915083 1915133 1915268 1915350 1915378 1915439 1915470 1915562 1915570 1915598 1915775
[85] 1915882 1915903 1915905 1915919 1915928 1915939 1915940 1915983 1916022 1927007 1936024 1937019
> n910
[1] 96
```

File 9:

```
> g910
[1] 1511191 1613010 1812477 1813528 1814096 1814518 1852443 1910006 1910060 1910113 1910123 1910137 1910198 1910224 1910238
[16] 1910265 1910276 1910298 1910347 1910402 1910409 1910620 1910643 1910644 1910735 1910865 1911066 1911105 1911110 1911136
[31] 1911185 1911217 1911262 1911285 1911363 1911520 1911561 1911565 1911650 1911736 1911841 1911907 1911931 1912046 1912237
[46] 1912371 1912539 1912579 1912602 1912713 1912749 1912761 1912916 1912954 1912980 1913040 1913045 1913048 1913094 1913123
[61] 1913167 1913186 1913260 1913306 1913341 1913356 1913396 1913418 1913424 1913430 1913433 1913467 1913560 1913609 1913678
[76] 1913695 1913713 1913828 1913949 1914003 1914011 1914038 1914047 1914064 1914084 1914093 1914210 1914227 1914316 1914352
[91] 1914405 1914424 1914472 1914477 1914641 1914659 1914661 1914674 1914685 1914698 1914704 1914720 1914738 1914763 1914802
[106] 1914830 1914878 1914880 1914881 1914979 1915016 1915040 1915083 1915130 1915133 1915268 1915323 1915329 1915350 1915378
[121] 1915470 1915520 1915551 1915562 1915570 1915598 1915651 1915775 1915795 1915822 1915866 1915882 1915903 1915931 1915939
[136] 1915940 1915982 1915983 1916061 1927007 1936024 1937019 1937064
> n910
[1] 143
```

File 10:

```
> g910
[1] 1511191 1613010 1812477 1813096 1814096 1814518 1820028 1852443 1910006 1910060 1910101 1910110 1910113
[14] 1910137 1910202 1910238 1910265 1910276 1910298 1910339 1910347 1910351 1910402 1910409 1910620 1910643
[27] 1910644 1910650 1910663 1910666 1910865 1910892 1910916 1911015 1911044 1911056 1911058 1911066 1911105
[40] 1911110 1911185 1911186 1911217 1911262 1911296 1911314 1911363 1911478 1911520 1911561 1911591 1911650
[53] 1911736 1911796 1911881 1911900 1911907 1912046 1912056 1912084 1912184 1912237 1912288 1912371
[66] 1912384 1912410 1912457 1912594 1912602 1912676 1912700 1912713 1912715 1912749 1912798 1912811 1912916
[79] 1912954 1912980 1913014 1913021 1913040 1913048 1913075 1913094 1913114 1913123 1913167 1913186 1913254
[92] 1913341 1913355 1913396 1913418 1913424 1913430 1913433 1913464 1913560 1913566 1913609 1913621 1913651
[105] 1913713 1913763 1913764 1913775 1913828 1913832 1913844 1913918 1913949 1914003 1914038 1914047 1914052
[118] 1914055 1914079 1914093 1914126 19141210 1914130 19141316 19141316 1914352 1914384 1914405 1914474 1914659 1914661
[131] 1914674 1914677 1914685 1914698 1914704 1914713 1914720 1914738 1914763 1914768 1914830 1914845 1914864
[144] 1914881 1914914 1914979 1915016 1915040 1915063 1915071 1915083 1915130 1915268 1915323 1915329 1915378
[157] 1915442 1915470 1915520 1915540 1915541 1915570 1915598 1915651 1915656 1915745 1915775 1915822 1915865
[170] 1915882 1915919 1915940 1915982 1915991 1916061 1927007 1936024
> n910
[1] 177
```

File 12:

```
> g910
[1] 1511191 1812477 1852443 1910123 1910409 1910892 1911066 1911262 1911363 1911441 1912123 1912288 1912371
[14] 1912410 1912457 1912594 1912602 1912676 1912713 1912761 1912798 1913014 1913021 1913048 1913075
[27] 1913094 1913380 1913418 1913424 1913430 1913464 1913560 1913599 1913609 1913621 1913918 1914047 1914093
[40] 1914126 1914220 1914227 1914352 1914477 1914659 1914661 1914677 1914704 1914713 1914830 1915016
[53] 1915040 1915083 1915130 1915133 1915146 1915268 1915350 1915378 1915442 1915520 1915541 1915775 1915795
[66] 1915919 1915983 1915988 1927007 1936024
> n910
[1] 70
```



19. Tạo g7910 chứa ID của các sinh viên vừa chủ động vừa thông minh vừa đổi phó và n7910 là số lượng sinh viên vừa chủ động vừa thông minh vừa đổi phó:

```
g7910=c()
k=0
for(i in 1:length(data7$ID7)){
  for(j in 1:length(g910)){
    if(data7$ID7[i]==g910[j]){
      k=k+1
      g7910[k]=data7$ID7[i]
    }
  }
}
n7910=length(g7910)
```

Ta được kết quả:

File 6:

```
> g7910
[1] 1910006 1911110 1911975 1913045 1913560 1913990 1914661 1914685 1914713 1915983
> n7910
[1] 10
```

File 9:

```
> g7910
[1] 1813528 1910006 1910060 1910198 1910265 1910402 1910735 1911110 1911363 1911565 1912713 1913040 1913045 1913167 1913430
[16] 1913560 1913678 1914011 1914477 1914661 1914685 1914830 1915130 1915133 1915470 1915570 1915866 1915931 1915983
> n7910
[1] 29
```

File 10:

```
> g7910
[1] 1813096 1820028 1852443 1910006 1910060 1910265 1910347 1910402 1911015 1911044 1911110 1911217 1911296
[14] 1911363 1911900 1912267 1912384 1912457 1912713 1912954 1913040 1913167 1913424 1913430 1913560 1913828
[27] 1914047 1914661 1914685 1914713 1914763 1914830 1914914 1915016 1915130 1915470 1915570 1915598 1915656
[40] 1915940
> n7910
[1] 40
```

File 12:

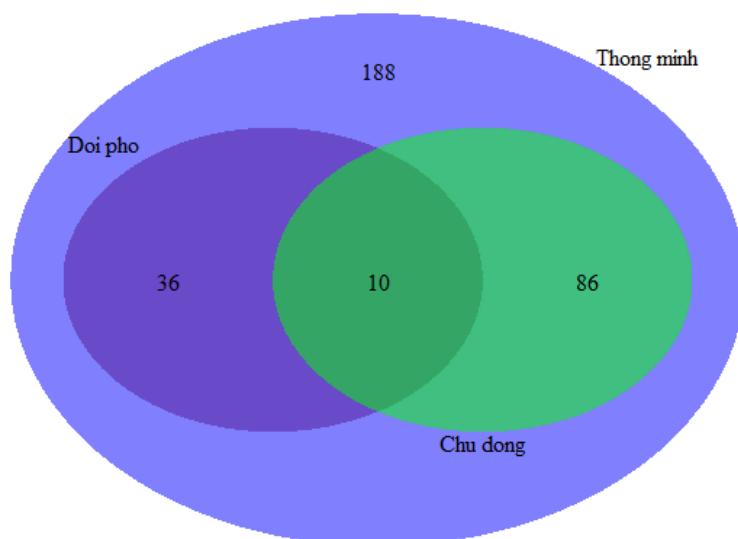
```
> g7910
[1] 1852443 1911363 1912713 1913045 1913424 1913430 1913599 1914047 1914661 1914685 1915016 1915133 1915350
[14] 1915983
> n7910
[1] 14
```

20. Vẽ biểu đồven minh họa phần giao của các loại sinh viên như sau:

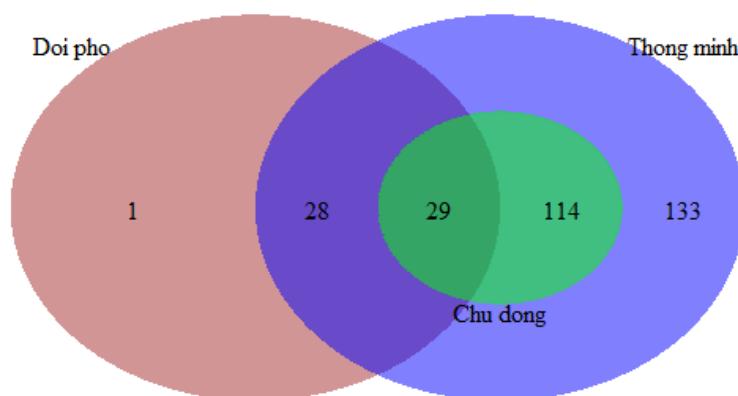
```
grid.newpage()
draw.triple.venn(area1=sosvdoiphopho,area2=sosinhvienthongminh,area3=sosinhvienchudong,n12=n79,n23=n98)
pho","Thong minh","Chu dong"))
```

Ta được các biểu đồ:

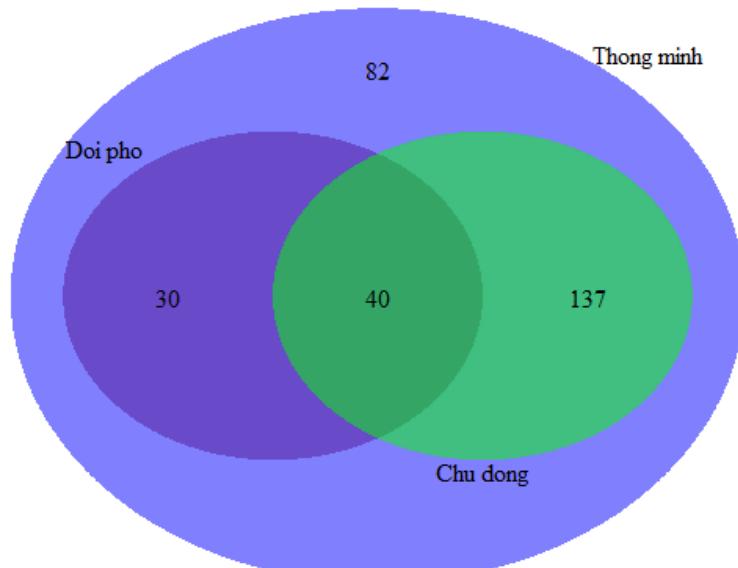
File 6:



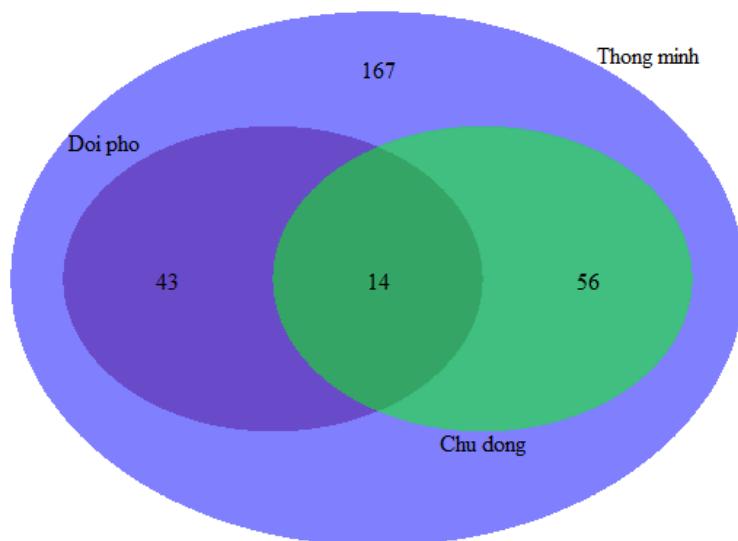
File 9:



File 10:



File 12:





Tài liệu tham khảo

- [1] Dalgaard, P. *Introductory Statistics with R*. Springer 2008.
- [2] Kenett, R. S. and Zacks, S. *Modern Industrial Statistics: with applications in R, MINITAB and JMP*, 2nd ed., John Wiley and Sons, 2014.
- [3] Kerns, G. J. *Introduction to Probability and Statistics Using R*, 2nd ed., CRC 2015.