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

CẤU TRÚC RỜI RẠC CHO KHMT (CO1007)

Ứng dụng thống kê khảo sát kết quả của kiểm tra môn Cấu trúc rời rạc

GVHD: Huỳnh Tường Nguyên
Nguyễn Tiến Thịnh
Nguyễn Ngọc Lễ
SV thực hiện: Nguyễn Đức An – 2010102
Phạm Quốc An – 2010829
Trần Thị Thu Thảo – 2010629
Dương Huỳnh Anh Đức – 2010226
Nguyễn Quang Huy – 1916081
Trần Thiện Nhân – 2010481

Tp. Hồ Chí Minh, 24/04/2021

Mục lục

1 Động cơ nghiên cứu

Trong mùa dịch Covid-19, trường Đại học Bách Khoa, ĐHQG-HCM đã triển khai giảng dạy trực tuyến và yêu cầu sinh viên thực hiện các bài tập nhỏ để thu nhận phản hồi về việc học tập và hiểu biết của các bạn thông qua các tài nguyên online được cung cấp.
Phân tích & thống kê dữ liệu qua các lần nộp bài của sinh viên không những giúp giáo viên có những hướng đúng trong việc phát hiện ra những kiến thức mà sinh viên chưa chắc chắn, cũng như có hướng để cải thiện bổ sung phần học liệu trong tương lai để phù hợp với hơn người học.

2 Mục tiêu

Khai phá dữ liệu từ các bài thi giữa kỳ, cuối kỳ có ý nghĩa quan trọng trong việc đánh giá chất lượng của sinh viên. Ngoài ra, những đánh giá kết quả thi cử của từng sinh viên, hay từng câu hỏi sẽ góp phần xác định những điểm mạnh, điểm yếu của sinh viên để giáo viên có phương pháp phù hợp trong việc cải thiện kỹ năng của sinh viên.
Trong bài tập lớn này, các sinh viên sẽ bắt đầu với các bài toán thống kê đơn giản từ những dữ liệu được cung cấp. Qua đó, các em sẽ tìm ra những con số thú vị, có ý nghĩa đối với các dữ liệu thực tế trong quá khứ của hệ thống chấm bài online. Những kết quả mà các em tìm ra sẽ là bước khởi đầu cho việc khai phá nguồn dữ liệu của hệ thống sau này, nhằm đạt tới mục tiêu nâng cao kỹ năng lập trình, kỹ năng giải quyết vấn đề cho người học cũng như hướng tới mục tiêu cao hơn khi tích hợp với các hệ thống quản lý và cải thiện chất lượng dạy và học.

3 Mô tả dữ liệu

Đính kèm đề bài tập lớn là 2 files **"192_CO1007.xlsx", "201_CO1007.xlsx"** trong đó chứa thông tin về kết quả trả lời các câu hỏi kỳ thi giữa kỳ và cuối kỳ gồm có các sheets:

1. Sheet CDR: chứa thông tin các chuẩn đầu ra môn học, mỗi câu hỏi trong đề thi có 1 chuẩn tương ứng.
2. Sheet GK, CK: thể hiện cho giữa kỳ, cuối kỳ
   - { A, B, C, D}: Các câu trả lời của sinh viên tương ứng cho các câu hỏi
   - {0, 1}: Kết quả đúng sai tương ứng cho mỗi câu hỏi. Nó được dẫn ra từ đáp án được cung cấp trong sheet.
3. Sheet GK_0, CK_0 cung cấp thông tin về lời giải, chuẩn đầu ra, chương liên quan cho mỗi câu hỏi trong từng mã đề thi.

4 Kiến thức và kết quả nghiên cứu

4.1 Tổng quan về R

R là một ngôn ngữ lập trình hàm cấp cao, cũng là một môi trường dành cho tính toán thống kê. R hỗ trợ rất nhiều công cụ cho phân tích dữ liệu, khám phá tri thức và khai thác dữ liệu.
R rất dễ học và có thể phát triển nhanh các ứng dụng tính toán xác suất thống kê, phân tích dữ liệu trong thời gian ngắn nhờ nhiều công cụ tích hợp sẵn dùng, như khả năng lập trình, kiểu dữ liệu phong phú, các hàm thống kê, giải thuật học tự động và các giao diện truy vấn dữ liệu, hiển thị dữ liệu. Đồng thời, R có thể tích hợp được với ngôn ngữ khác (C, C++) và tương tác với nhiều nguồn dữ liệu và các gói thống kê (SAS, SPSS).
Ngôn ngữ R được biết đến là một công cụ rất mạnh cho machine learning, thống kê và phân tích dữ liệu. R có thể chạy code mà không cần đến bất cứ compiler nào, cũng có thể thực hiện bất kỳ một phép tính, sơ đồ và công thức nào trên vectors... khi cần thiết.

4.2 Thống kê mô tả

Một thống kê mô tả là một thống kê tóm tắt rằng số lượng mô tả hoặc tóm tắt các tính năng từ một tập hợp các thông tin. Một số thước đo thường được sử dụng để mô tả một tập dữ liệu là các thước đo về xu hướng trung tâm và các thước đo về sự biến thiên hoặc phân tán .
Các phép đo xu hướng trung tâm bao gồm giá trị trung bình , giá trị trung bình và phương thức , trong khi các phép đo độ biến thiên bao gồm độ lệch chuẩn (hoặc phương sai), giá trị tối thiểu và tối đa của các biến số, kurtosis và độ lệch.

4.3 Tổng thể và mẫu

Khi nghiên cứu một vấn đề người ta thường khảo sát trên một dấu hiệu nào đó, các dấu hiệu này thể hiện trên nhiều phần tử. Tập hợp tất cả các phần tử mang dấu hiệu này được gọi là tổng thể (population).
Mẫu là một phần của tổng thể được chọn ra theo những cách thức nhất định và với một dung lượng hợp lý.
Mẫu phải có tính đại diện cao.

4.4 Phân tích một số thông tin mẫu

4.4.1 Trung bình - Trung vị - Tứ phân vị

- **Trung bình**

  Trung bình (**Medium**) là một số duy nhất được lấy làm đại diện cho một danh sách các số.

  Thông thường "trung bình" chỉ số trung bình số học bằng tổng của các số chia cho số lượng đang được tính trung bình.

  Công thức tính trung bình là:

$$\overline{x} = \frac{1}{N}\sum_{k=1}^{N} f_i x_i$$

- **Trung vị**

  Trong lý thuyết xác suất và thống kê, số trung vị (**median**) là một số tách giữa nửa lớn hơn và nửa bé hơn của một mẫu, một quần thể, hay một phân bố xác suất. Nó là giá trị giữa trong một phân bố, mà số các số nằm trên hay dưới con số đó là bằng nhau.

  Để tìm số trung vị của một danh sách hữu hạn các số, ta xếp tăng dần tất cả các quan sát, rồi lấy giá trị nằm giữa danh sách. Nếu số quan sát là số chẵn, người ta thường lấy trung bình của hai giá trị nằm giữa.

  Công thức tính số trung vị:

  Nếu có n số liệu, n lẻ ($\mathbf{n = 2k + 1}$) thì

$$M_e = x_{k+1}$$

  Nếu có n số liệu, n chẵn ($\mathbf{n = 2k}$) thì

$$M_e = \frac{x_k + x_{k+1}}{2}$$

- **Tứ phân vị**

  Tứ phân vị (**quartile**) là giá trị bằng số phân chia một nhóm các kết quả quan sát bằng số thành bốn phần, mỗi phần có số liệu quan sát bằng nhau (=25% số kết quả quan sát).

  Tứ phân vị 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 (Q1), thứ nhì (Q2) và thứ ba (Q3). Ba giá trị này chia một tập hợp dữ liệu (đã sắp xếp dữ liệu theo trật từ từ bé đến lớn) thành 4 phần có số lượng quan sát đều nhau.

  - 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
  - Giá trị tứ phân vị thứ ba Q3 bằng trung vị phần trên

4.4.2 Cực đại mẫu - Cực tiểu mẫu

Cực đại mẫu là giá trị lớn nhất trong mẫu
Cực tiểu mẫu là giá trị nhỏ nhất trong mẫu

4.4.3 Một số đại lượng đo lường trong thống kê mẫu

- **Độ lệch chuẩn**

  Độ lệch chuẩn(**standard deviation**) là một đại lượng thống kê mô tả dùng để đo mức độ phân tán của một tập dữ liệu đã được lập thành bảng tần số.

  Độ lệch chuẩn đo tính biến động của giá trị mang tính thống kê. Nó cho thấy sự chênh lệch về giá trị của từng thời điểm đánh giá so với giá trị trung bình.

  Có thể tính ra độ lệch chuẩn bằng cách lấy căn bậc hai của phương sai.

  Công thức độ lệch chuẩn :

$$s = \sqrt{s^2} = \sqrt{\frac{1}{N-1}\sum_{k=1}^{N}(x_i - \overline{x})^2}$$

- **Phương sai**

  Trong lý thuyết xác suất và thống kê, phương sai của một biến ngẫu nhiên là một độ 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.

  Công thức phương sai:

$$\sigma^2 = \frac{1}{N-1}\sum_{k=1}^{N}(x_i - \overline{x})^2$$

  Phương sai của một biến ngẫu nhiên là bình phương của độ lệch chuẩn.

- **Độ méo lệch(skewness)**

  Độ méo lệch (**skewness**) của một phân phối xác suất đo lường sự **bất đối xứng** của phân phối đó. 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.

  Công thức độ méo lệch:

$$skewness = \frac{1}{n}\frac{\sum_{k=1}^{n}(x_i - \overline{x})^3}{s^3}$$

- **Độ nhọn(kurtosis)**

  Độ nhọn (**kurtosis**) là một chỉ số để đo lường về đặc điểm hình dáng của một phân phối xác suất. Cụ thể hơn, nó so sánh độ cao phần trung tâm của một phân phối so sánh với một phân phối chuẩn. Phần trung tâm càng cao và nhọn, chỉ số Kurtosis của phân phối đó càng lớn.

  Công thức độ nhọn :

$$kurtosis = \frac{1}{n}\frac{\sum_{k=1}^{n}(x_i - \overline{x})^4}{s^4}$$

- **Độ phân tán**

  Độ phân tán là một thuật ngữ thống kê mô tả kích thước của phân phối các giá trị dự kiến cho một biến cụ thể. Độ phân tán có thể được đo bằng các chỉ số thống kê khác nhau, chẳng hạn như khoảng giá trị, phương sai và độ lệch chuẩn.

4.5 Tần suất

- Tần suất: là số lượng giá trị dữ liệu rơi vào một lớp nhất định, trong đó các lớp có tần số lớn hơn có thanh cao hơn và lớp có tần số nhỏ hơn có thanh thấp hơn.
- Tần suất tương đối: là thước đo tỷ lệ hoặc phần trăm giá trị dữ liệu rơi vào một lớp cụ thể.
- Tần suất tương đối tích lũy : là tổng các tần số khi ta cộng dồn từ trên xuống.

5 Bài tập áp dụng

i) Xác định số lượng sinh viên trong tập mẫu

- **Giải truyền thống**:
  Ta thống kê được trong danh sách GK có 366 sinh viên, CK có 361 sinh viên.
- **Source:**
  - Dùng hàm **nrow()** để đếm số hàng của data.

```
#i
sosvGK = nrow(GK)
sosvCK = nrow(CK)
```

- **Output:**

sosvCK	361L
sosvGK	366L

ii) Nhóm câu hỏi liên quan đến số câu của các sinh viên

1) Tính tổng các câu đúng của mỗi sinh viên trong tập mẫu

- **Giải truyền thống**:
  Tính tổng các số trong mỗi hàng sẽ được tổng số câu đúng của sinh viên ở hàng đó.
- **Source:**
  - Tạo các **vector()** rightGK và rightCK rỗng mà mỗi phần tử của vector có giá trị là 0.

```
rightGK = vector(mode = "numeric", length = sosvGK)
rightCK = vector(mode = "numeric", length = sosvCK)
```

  - Dùng hàm **rowSums()** để tính tổng các phần tử trong mỗi hàng từ cột m đến cột n (tổng giá trị của mỗi hàng là tổng số câu đúng của mỗi sinh viên) xong gán lại vào rightGK và rightCK.

```
rightGK = rowSums(GK[4:28])
rightCK = rowSums(CK[4:32])
```

- **Output:**
  - **GK** : Số câu đúng là :

```
> rightGK
  [1]  7 12 16 13 13 11 20 13 12 15 15 13 10  8 16 12 12 19 13 19 15 15 15 10 19 20
 [27] 14 14 19 13 17 16 15 18 19 13 19 17 18 18 17 19 18 14 18 20 15 19 19 17 15 16
 [53] 17 17 16 16 16 20 17 18 13 16 17 15 16 16 18 16 13 17 19 20 16 15 17 16 16 15
 [79] 12 18  9 15 17 16 13 16 12 17 13 13 19 15 18 16 13 11 16 12 20 15 19 15 20 17
[105] 15 18 21 17 15 18 16 14 17 14 16 16 14 14 18 13 13 15 16 16 16 14 19 14 16 17
[131] 20 14 14 15 18 14 18 12 17 16 16 18 15 10 18 16 18 13 16 16 16 18 16 14 12 11
[157] 15 15 12 13 17 15 19 15 14 18 17 15 11 14 16 17 15 15 15 16 13 12 15 17 12 12
[183] 16 14 21 15 11 15 18 17 16 14  9 15 17 15 12 15 14  9 13 13 13 16 13 10 15 13
[209] 15 16 13 16 18 16 13 13 13 15 15 14 16 13  9 13 19 13 17 15 15 17 17 15 11 14
[235] 13 18 18 15  9 10 19 10 13 15 11 18 15 12 16 15 18 14 18 16 20 17 13 14 15 18
[261] 14 12 15 11 18 18 13 10 14 20 17 13 16 19 15 15 14 12 15 12 15 18 15 15 17 14
[287] 17 15 14 18 12 16 18 17 10 12 16 15 16 14 18 11 18 17 13 14 13 18 12 17 18 15
[313] 15 12 14 20 19 17 15 18 20 11 15 18 17 14 16 15 14 19 14 14 14 19 15 17 17 19
[339] 15 14 16 16 11 14 16 19 16 14 12 16 14 10 15 18 13 11 19 16 13 11 13 18  6  6
[365] 10 15
```

  - **CK** : Số câu đúng là :

```
> rightCK
  [1] 14 20 23 11 15 18 13 13 16  9 22 16 21 20 23 16 25 20 22 25 24 25 24 19 23 20
 [27] 16 23 16 20 23 21 19 26 23 25 23 24 24 21 21 24 24 23 20 21 24 13 19 20 15 18
 [53] 20 23 20 25 22  8  9 17 10 24 24 17 23 17 24 23 17 21 19 16 17 21 17 18 17 18
 [79] 17 20 18 17 16 21 23 19 19 24 10 25 24 22 21 18 21 22 22 22 12 24 19 23 19 24
[105] 23 22 25 24 19 23 12 18 10 15 24 25 15 23 10 24 24 18 24 17 15 23 23 24 16 21
[131] 23 24 22 18 17 23 22 17 22 18 15 17 14 19 20 19 24 23 14 24 15 21 15 21 22 21
[157] 23 23 23 24 23 15 10 25 22 19 18 20 22 23 21 17 18 11 23 24 23 16 25 16 21 24
[183] 23 16 15 22 21 23 23 22 22 19 25 16 20 23 14 10 22 16 21 13 16 20 23 21 16 15
[209] 25 19 21 14 13 11 13 22 23 22 17 22 19 15 23 20 22 22 21 15 19 18 19 26 24 21
[235] 17 23 24 22 23 18 15 19 20 11 21 26 23 15 20 22 26 23 19 20 13 23 22 15 22 24
[261] 21 20 14 18 25 22 19 20 17 21 20 21 16 18 18 20 18 24 17 23 23 20 24 21 12 24
[287] 22 25 20 16 20 20 22 24 14 23 15 22 25 21 15 19 17 21 21 21 21 15 14 22 19 19
[313] 24 25 23 20 18 23 24 23 20 24 24 20 20 19 16 20 19 19 21 19 25 14 21 13 18 20
[339] 12 22 26 25 20 23 25 18 23 21 24 11 21 24 19 22 20 15 20 14  7  7 14
```

2) Tính tổng các câu sai của mỗi sinh viên trong tập mẫu

- **Giải truyền thống**:
  Lấy tổng số câu trừ số câu đúng ta được số câu sai
- **Source:**
  - Số câu sai bằng tổng số câu - số câu đúng.

```
1       wrongGK = 25 - rightGK
2       wrongCK = 29 - rightCK
```

- **Output:**
  - **GK** : Số câu sai là :

```
> wrongGK
  [1] 18 13  9 12 12 14  5 12 13 10 10 12 15 17  9 13 13  6 12  6 10 10 10 15  6  5
 [27] 11 11  6 12  8  9 10  7  6 12  6  8  7  7  8  6  7 11  7  5 10  6  6  8 10  9
 [53]  8  8  9  9  9  5  8  7 12  9  8 10  9  9  7  9 12  8  6  5  9 10  8  9  9 10
 [79] 13  7 16 10  8  9 12  9 13  8 12 12  6 10  7  9 12 14  9 13  5 10  6 10  5  8
[105] 10  7  4  8 10  7  9 11  8 11  9  9 11 11  7 12 12 10  9  9  9 11  6 11  9  8
[131]  5 11 11 10  7 11  7 13  8  9  9  7 10 15  7  9  7 12  9  9  9  7  9 11 13 14
[157] 10 10 13 12  8 10  6 10 11  7  8 10 14 11  9  8 10 10 10  9 12 13 10  8 13 13
[183]  9 11  4 10 14 10  7  8  9 11 16 10  8 10 13 10 11 16 12 12 12  9 12 15 10 12
[209] 10  9 12  9  7  9 12 12 12 10 10 11  9 12 16 12  6 12  8 10 10  8  8 10 14 11
[235] 12  7  7 10 16 15  6 15 12 10 14  7 10 13  9 10  7 11  7  9  5  8 12 11 10  7
[261] 11 13 10 14  7  7 12 15 11  5  8 12  9  6 10 10 11 13 10 13 10  7 10 10  8 11
[287]  8 10 11  7 13  9  7  8 15 13  9 10  9 11  7 14  7  8 12 11 12  7 13  8  7 10
[313] 10 13 11  5  6  8 10  7  5 14 10  7  8 11  9 10 11  6 11 11 11  6 10  8  8  6
[339] 10 11  9  9 14 11  9  6  9 11 13  9 11 15 10  7 12 14  6  9 12 14 12  7 19 19
[365] 15 10
```

  - **CK** : Số câu sai là :

```
> wrongCK
  [1] 15  9  6 18 14 11 16 16 13 20  7 13  8  9  6 13  4  9  7  4  5  4  5 10  6  9
 [27] 13  6 13  9  6  8 10  3  6  4  6  5  5  8  8  5  5  6  9  8  5 16 10  9 14 11
 [53]  9  6  9  4  7 21 20 12 19  5  5 12  6 12  5  6 12  8 10 13 12  8 12 11 12 11
 [79] 12  9 11 12 13  8  6 10 10  5 19  4  5  7  8 11  8  7  7  7 17  5 10  6 10  5
[105]  6  7  4  5 10  6 17 11 19 14  5  4 14  6 19  5  5 11  5 12 14  6  6  5 13  8
[131]  6  5  7 11 12  6  7 12  7 11 14 12 15 10  9 10  5  6 15  5 14  8 14  8  7  8
[157]  6  6  6  5  6 14 19  4  7 10 11  9  7  6  8 12 11 18  6  5  6 13  4 13  8  5
[183]  6 13 14  7  8  6  6  7  7 10  4 13  9  6 15 19  7 13  8 16 13  9  6  8 13 14
[209]  4 10  8 15 16 18 16  7  6  7 12  7 10 14  6  9  7  7  8 14 10 11 10  3  5  8
[235] 12  6  5  7  6 11 14 10  9 18  8  3  6 14  9  7  3  6 10  9 16  6  7 14  7  5
[261]  8  9 15 11  4  7 10  9 12  8  9  8 13 11 11  9 11  5 12  6  6  9  5  8 17  5
[287]  7  4  9 13  9  9  7  5 15  6 14  7  4  8 14 10 12  8  8  8  8 14 15  7 10 10
[313]  5  4  6  9 11  6  5  6  9  5  5  9  9 10 13  9 10 10  8 10  4 15  8 16 11  9
[339] 17  7  3  4  9  6  4 11  6  8  5 18  8  5 10  7  9 14  9 15 22 22 15
```

3) Xác định số câu đúng nhiều nhất và thấp nhất trong tập mẫu

- **Giải truyền thống**:
  - Số câu đúng nhiều nhất là **số lớn nhất** tìm được trong tập mẫu số câu đúng ở trên tùy theo mỗi sheet GK, CK.
  - Số câu đúng ít nhất là **số nhỏ nhất** tìm được trong tập mẫu số câu đúng ở trên tùy theo mỗi sheet GK, CK.
- **Source:**
  - Dùng hàm **max(), min()** để tìm giá trị lớn nhất, nhỏ nhất
    - Số câu đúng nhiều nhất là:

```
rightGK_max = max(rightGK)
rightCK_max = max(rightCK)
```

– Số câu đúng ít nhất là:

```
rightGK_min = min(rightGK)
rightCK_min = min(rightCK)
```

- **Output:**
  - **GK**:

rightGK_max	21
rightGK_min	6

  - **CK**:

rightCK_max	26
rightCK_min	7

4) Vẽ biểu đồ phổ cho tổng số các câu đúng của sinh viên ứng với từng mã đề lần lượt trong tập mẫu giữa kỳ và cuối kỳ

- **Source**:
  - Sử dụng hàm **frame()** và **subset()** để tách ra hai cột mã đề (MADE) và số câu đúng (rightAns) thành các danh sách mới.

```
rANSGK = data.frame(rightGK, GK$MADE)
rANSCK = data.frame(rightCK, CK$MADE)
```

  - Sử dụng hàm **colnames()** để đặt tên cho dòng [2] là MADE.

```
colnames(rANSGK)[2]<-"MADE"
colnames(rANSCK)[2]<-"MADE"
```

  - Tạo các danh sách mới tương ứng.
    - **Giữa kì:**

```
rANS_1921GK<-subset.data.frame(rANSGK, MADE==1921)
rANS_1922GK<-subset.data.frame(rANSGK, MADE==1922)
rANS_1923GK<-subset.data.frame(rANSGK, MADE==1923)
rANS_1924GK<-subset.data.frame(rANSGK, MADE==1924)
```

    - **Cuối kì:**

```
rANS_1921CK<-subset.data.frame(rANSCK, MADE==1921)
rANS_1922CK<-subset.data.frame(rANSCK, MADE==1922)
rANS_1923CK<-subset.data.frame(rANSCK, MADE==1923)
rANS_1924CK<-subset.data.frame(rANSCK, MADE==1924)
```

  - Sử dụng hàm **hist()** để vẽ đồ thị theo mỗi loại mã đề.
    - **Giữa kì:**

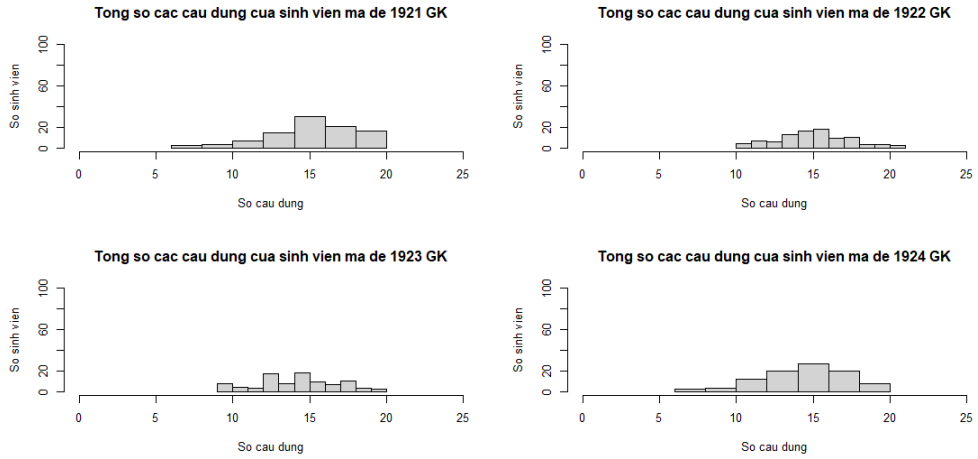
```
par(mfrow=c(2,2))
hist(rANS_1921GK$rightGK, main = "Tong so cac cau dung cua sinh vien ma de 1921 GK",
     xlap = "So cau dung", ylap = "So sinh vien", xlim=c(0,25), ylim=c(0,100))
hist(rANS_1922GK$rightGK, main = "Tong so cac cau dung cua sinh vien ma de 1922 GK",
     xlap = "So cau dung", ylap = "So sinh vien", xlim=c(0,25), ylim=c(0,100))
hist(rANS_1923GK$rightGK, main = "Tong so cac cau dung cua sinh vien ma de 1923 GK",
     xlap = "So cau dung", ylap = "So sinh vien", xlim=c(0,25), ylim=c(0,100))
hist(rANS_1924GK$rightGK, main = "Tong so cac cau dung cua sinh vien ma de 1924 GK",
     xlap = "So cau dung", ylap = "So sinh vien", xlim=c(0,25), ylim=c(0,100))
```

    - **Cuối kì:**

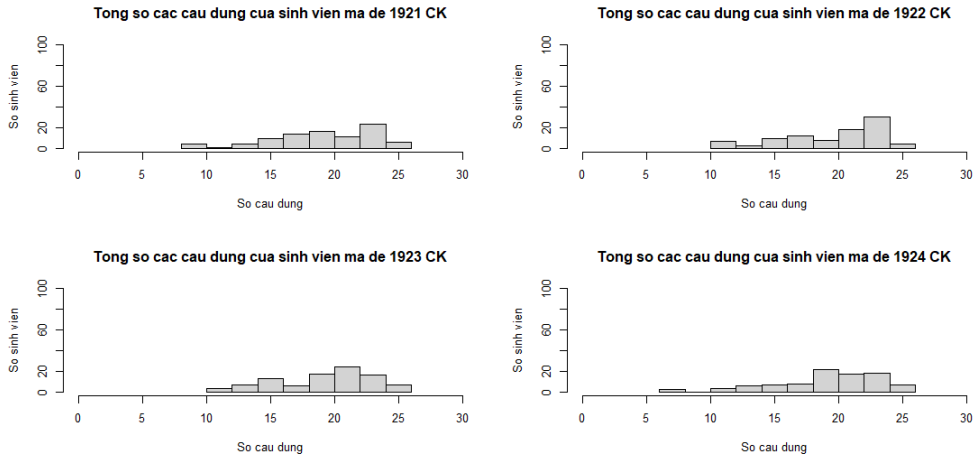
```
par(mfrow=c(2,2)
hist(rANS_1921CK$rightCK, main = "Tong so cau dung cua sinh vien ma de 1921 CK",
     xlap = "So cau dung", ylap = "So sinh vien", xlim=c(0,30), ylim=c(0,100))
hist(rANS_1922CK$rightCK, main = "Tong so cau dung cua sinh vien ma de 1922 CK",
     xlap = "So cau dung", ylap = "So sinh vien", xlim=c(0,30), ylim=c(0,100))
```

```
4   hist(rANS_1923CK$rightCK, main = "Tong so cau dung cua sinh vien ma de 1923 CK",
        xlap = "So cau dung", ylap = "So sinh vien", xlim=c(0,30), ylim=c(0,100))
5   hist(rANS_1924CK$rightCK, main = "Tong so cau dung cua sinh vien ma de 1924 CK",
        xlap = "So cau dung", ylap = "So sinh vien", xlim=c(0,30), ylim=c(0,100))
```

- **Output**:



Hình 1 : Biểu đồ số câu đúng của sinh viên ứng với từng mã đề trong giữa kì



Hình 2 : Biểu đồ số câu đúng của sinh viên ứng với từng mã đề trong cuối kì

5) Vẽ biểu đồ phổ cho tổng số các câu sai của sinh viên ứng với từng mã đề lần lượt trong tập mẫu giữa kỳ và cuối kỳ

- **Source**:
  - Sử dụng hàm **frame()** và **subset()** để tách ra hai cột mã đề (MADE) và số câu đúng (rightAns) thành các danh sách mới.

```
1       wANSGK=data.frame(wrongGK,GK$MADE)
2       wANSCK=data.frame(wrongCK,CK$MADE)
```

  - Sử dụng hàm **colnames()** để đặt tên cho dòng [2] là MADE.

```
1       colnames(wANSGK)[2]<-"MADE"
2       colnames(wANSCK)[2]<-"MADE"
```

  - Tạo các danh sách mới tương ứng.
    - **Giữa kì:**

```
1       wANS_1921GK<- subset.data.frame(wANSGK, MADE==1921)
2       wANS_1922GK<- subset.data.frame(wANSGK, MADE==1922)
3       wANS_1923GK<- subset.data.frame(wANSGK, MADE==1923)
4       wANS_1924GK<- subset.data.frame(wANSGK, MADE==1924)
```

* **Cuối kì:**

```
wANS_1921CK<- subset.data.frame(wANSCK, MADE==1921)
wANS_1922CK<- subset.data.frame(wANSCK, MADE==1922)
wANS_1923CK<- subset.data.frame(wANSCK, MADE==1923)
wANS_1924CK<- subset.data.frame(wANSCK, MADE==1924)
```

– Sử dụng hàm **hist()** để vẽ đồ thị theo mỗi loại mã đề.

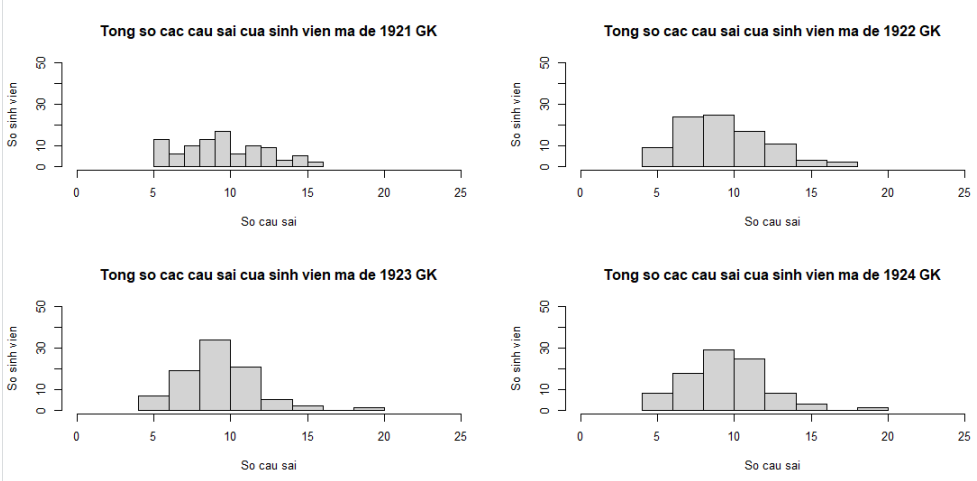
* **Giữa kì:**

```
par(mfrow=c(2,2))
hist(wANS_1921GK$wrongGK, main = "Tong so cac cau sai cua sinh vien ma de 1921 GK",
     xlap = "So cau sai", ylap = "So sinh vien", xlim=c(0,25), ylim=c(0,50))
hist(wANS_1922GK$wrongGK, main = "Tong so cac cau sai cua sinh vien ma de 1922 GK",
     xlap = "So cau sai", ylap = "So sinh vien", xlim=c(0,25), ylim=c(0,50))
hist(wANS_1923GK$wrongGK, main = "Tong so cac cau sai cua sinh vien ma de 1923 GK",
     xlap = "So cau sai", ylap = "So sinh vien", xlim=c(0,25), ylim=c(0,50))
hist(wANS_1924GK$wrongGK, main = "Tong so cac cau sai cua sinh vien ma de 1924 GK",
     xlap = "So cau sai", ylap = "So sinh vien", xlim=c(0,25), ylim=c(0,50))
```

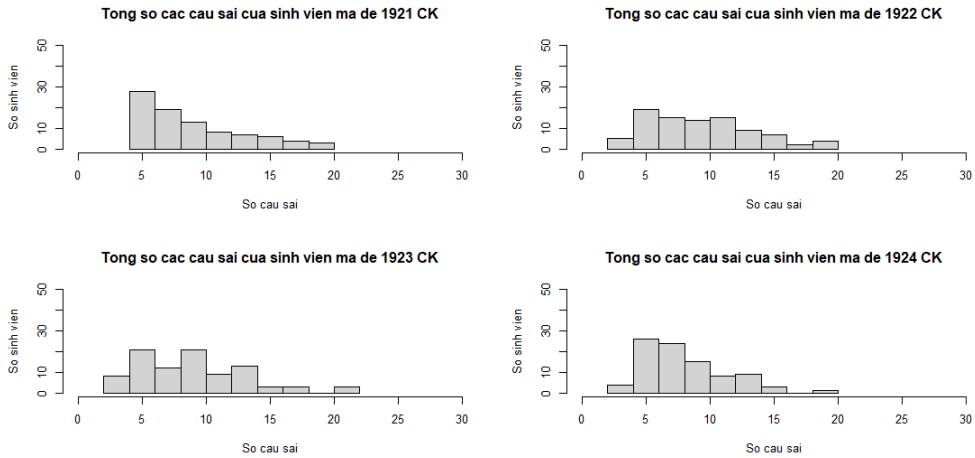
* **Cuối kì:**

```
par(mfrow=c(2,2))
hist(wANS_1921CK$wrongCK, main = "Tong so cac cau sai cua sinh vien ma de 1921 CK",
     xlap = "So cau sai", ylap = "So sinh vien", xlim=c(0,30), ylim=c(0,50))
hist(wANS_1922CK$wrongCK, main = "Tong so cac cau sai cua sinh vien ma de 1922 CK",
     xlap = "So cau sai", ylap = "So sinh vien", xlim=c(0,30), ylim=c(0,50))
hist(wANS_1923CK$wrongCK, main = "Tong so cac cau sai cua sinh vien ma de 1923 CK",
     xlap = "So cau sai", ylap = "So sinh vien", xlim=c(0,30), ylim=c(0,50))
hist(wANS_1924CK$wrongCK, main = "Tong so cac cau sai cua sinh vien ma de 1924 CK",
     xlap = "So cau sai", ylap = "So sinh vien", xlim=c(0,30), ylim=c(0,50))
```

- **Output**:



Hình 3 : Biểu đồ số câu sai của sinh viên ứng với từng mã đề trong giữa kì



Hình 4 : Biểu đồ số câu sai của sinh viên ứng với từng mã đề trong cuối kì

iii) Nhóm câu hỏi liên quan đến điểm của các sinh viên

a. Điểm được tính bằng cách lấy tổng số câu đúng chia cho tổng số câu ứng với mỗi kỳ thi, rồi lấy kết quả nhân 10 và làm tròn đến 2 số thập phân.

- **Giải truyền thống**:

$$\textbf{Điểm giữa kì} = \frac{\textbf{Số câu đúng} * 10}{25}.$$

$$\textbf{Điểm cuối kì} = \frac{\textbf{Số câu đúng} * 10}{29}.$$

Làm tròn lên hai chữ số thập phân.

- **Source**:
- Dùng hàm **round(number, 2)** để làm tròn lên hai chữ số thập phân.

```
DiemGK = round(rightGK * 10 / 25, 2)
DiemCK = round(rightCK * 10 / 29, 2)
```

b. Điểm tổng kết tính bằng cách lấy tổng 40% điểm giữa kỳ và 60% điểm cuối kỳ.

- **Giải truyền thống**:
Điểm tổng kết = (Điểm giữa kì) * 0.4 + (Điểm cuối kì) * 0.6
Làm tròn lên hai chữ số thập phân.
- **Source:**
Do số thí sinh thi giữa kì là 366 lớn hơn số thi sinh thi cuối kì là 361, nên ta chạy vòng lặp để gán cho những thí sinh không thi cuối kì thì đạt 0 điểm.

```
for ( i in (length(DiemCK) + 1) : length(DiemGK)) { DiemCK <- append(DiemCK, 0)}
DiemTK = round(DiemGK * 0.4 + DiemCK * 0.6 , 2)
```

1) Tính trung vị mẫu, cực đại mẫu, cực tiểu mẫu giữa kỳ và mẫu cuối kỳ

- **Giải truyền thống**:
Trung vị mẫu:
Để tính trung vị của mẫu, ta sắp xếp các giá trị của dữ liệu theo thứ tự **tăng dần**. Gọi n là số phần tử của mẫu.
  - **Nếu n chẵn:**

$$\textbf{Số trung vị} = \frac{\textbf{Số thứ}\frac{n}{2} + \textbf{Số thứ}\frac{n+1}{2}}{2}$$

  - **Nếu n lẻ:**

$$\textbf{Số trung vị} = \textbf{Số thứ}\frac{n+1}{2}$$

Trong bài ta có :

– **GK** :

$$\textbf{Số trung vị} = \frac{\textbf{Số thứ 183} + \textbf{Số thứ 184}}{2}$$

– **CK** :

$$\textbf{Số trung vị} = \textbf{Số thứ 181}.$$

Cực đại mẫu:
Cực đại mẫu là giá trị lớn nhất trong mẫu.
Cực tiểu mẫu:
Cực tiểu mẫu là giá trị nhỏ nhất trong mẫu.

- **Source:**
  – **Trung vị:** Sử dụng hàm **median()** để tính **trung vị** của mẫu.

```
median(DiemGK)
median(DiemCK)
```

  – **Cực đại mẫu:** Sử dụng hàm **max()** để tính cực đại của mẫu.

```
max(DiemGK)
max(DiemCK)
```

  – **Cực tiểu mẫu:** Sử dụng hàm **min()** để tính cực tiểu của mẫu.

```
min(DiemGK)
min(DiemCK)
```

- **Output:**
  – **Trung vị:**

```
> median(DiemGK)
[1] 6
> median(DiemCK)
[1] 6.9
```

  – **Cực đại mẫu:**

```
> max(DiemGK)
[1] 8.4
> max(DiemCK)
[1] 8.97
```

  – **Cực tiểu mẫu:**

```
> min(DiemGK)
[1] 2.4
> min(DiemCK)
[1] 0
```

2) Đếm các sinh viên mà điểm của mỗi sinh viên trong tập mẫu giữa kỳ và mẫu cuối kỳ lớn hơn hoặc bằng 9

- **Giải truyền thống**:
  – Số sinh viên có điểm giữa kì >= 9 là : 0.
  – Số sinh viên có điểm cuối kì >= 9 là : 0.
- **Source:**
  Sử dụng cấu trúc rẽ nhánh **which()** để kiểm tra xem điểm có >=9 và dùng hàm **length()** để đếm số lượng sinh viên.

```
GK9 = length(which(DiemGK >=9))
CK9 = length(which(DiemCK >=9))
```

- **Output:**

GK9	0L
CK9	0L

3) Đếm các sinh viên mà điểm của mỗi sinh viên trong tập mẫu giữa kỳ và mẫu cuối kỳ lớn hơn hoặc bằng 7

- **Giải truyền thống**:
  - Số sinh viên có điểm giữa kì >= 7 là : 77.
  - Số sinh viên có điểm cuối kì >= 7 là : 181.
- **Source:**
  Sử dụng cấu trúc rẽ nhánh **which()** để kiểm tra xem điểm có >=7 và dùng hàm **length()** để đếm số lượng sinh viên.

```
1       GK7 = length(which(DiemGK >=7))
2       CK7 = length(which(DiemCK >=7))
```

- **Output:**

GK7	77L
CK7	181L

4) Đếm các sinh viên mà điểm của mỗi sinh viên trong tập mẫu giữa kỳ và mẫu cuối kỳ lớn hơn hoặc bằng 5
   - **Giải truyền thống**:
     - Số sinh viên có điểm giữa kì >= 5 là : 311.
     - Số sinh viên có điểm cuối kì >= 5 là : 322.
   - **Source:**
     Sử dụng cấu trúc rẽ nhánh **which()** để kiểm tra xem điểm có >=5 và dùng hàm **length()** để đếm số lượng sinh viên.

```
1       GK5 = length(which(DiemGK >=5))
2       CK5 = length(which(DiemCK >=5))
```

   - **Output:**

GK5	311L
CK5	322L

5) Đếm các sinh viên mà điểm của mỗi sinh viên trong tập mẫu nhỏ hơn 5
   - **Giải truyền thống**:
     - Số sinh viên có điểm giữa kì < 5 là : 55.
     - Số sinh viên có điểm cuối kì < 5 là : 44.
   - **Source:**
     Sử dụng cấu trúc rẽ nhánh **which()** để kiểm tra xem điểm có < 5 và dùng hàm **length()** để đếm số lượng sinh viên.

```
1       GK5 = length(which(DiemGK < 5))
2       CK5 = length(which(DiemCK < 5))
```

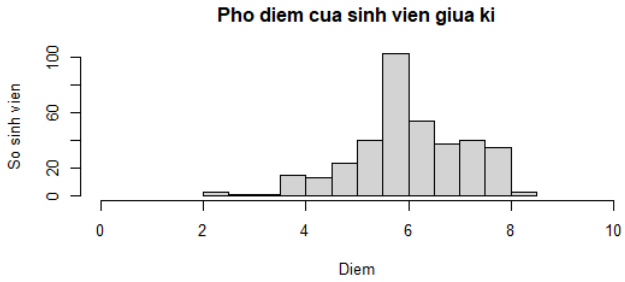
   - **Output:**

GKless5	55L
CKless5	44L

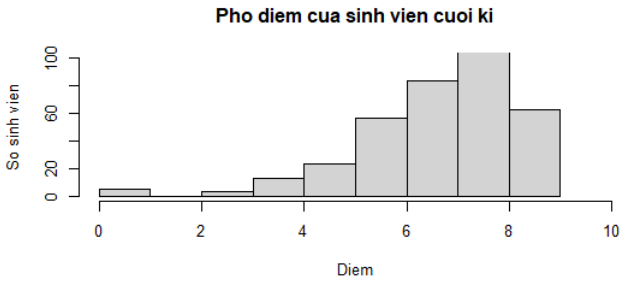
6) Vẽ biểu đồ phổ điểm của sinh viên trong tập mẫu giữa kỳ và mẫu cuối kỳ
   - **Source:**
     - Sử dụng hàm **c()** để tạo các vector.
     - Trục Ox biểu diễn số điểm **xlim()** có giá trị từ 0 đến 10, trục Oy biểu diễn số lượng học sinh tương ứng **ylim()**(chạy từ 0 đến 100).
     - Sử dụng hàm **hist()** để vẽ biểu đồ.

```
1     par(mfrow=c(2,2))
2     hist(DiemGK, main = "Pho diem cua sinh vien giua ki", xlap = "Diem", ylap = "So sinh
          vien", xlim=c(0,10) , ylim=c(0,100))
3     hist(DiemCK, main = "Pho diem cua sinh vien cuoi ki", xlap = "Diem", ylap = "So sinh
          vien", xlim = c(0,10) , ylim = c(0,100))
```

   - **Output:**



Hình 5 : Phổ điểm sinh viên giữa kì



Hình 6 : Phổ điểm sinh viên cuối kì

7) Xác định danh sách sinh viên gồm số thứ tự (No), mã nhóm và tổ có điểm số lớn nhất trong tập mẫu giữa kỳ và mẫu cuối kỳ

- **Source:**
  - Sử dụng hàm **cbind()** để tổ hợp lại các vector cũng là các cột No, MANH, TO của sheet GK, CK thành một ma trận mới.
  - Sử dụng hàm **colnames()** để đặt tên cho các title tại các vị trí [2],[3],[4] lần lượt là "No","MANH", "TO" để tạo thành một ma trận mới hoàn chỉnh.

```
SvGK<-cbind(DiemGK, GK$No, GK$MANH, GK$TO)
colnames(SvGK)[2]<-"No"
colnames(SvGK)[3]<-"MANH"
colnames(SvGK)[4]<-"TO"

SvCK<-cbind(DiemCK,CK$No,CK$MANH,CK$TO)
colnames(SvCK)[2]<-"No"
colnames(SvCK)[3]<-"MANH"
colnames(SvCK)[4]<-"TO"
```

  - Sử dụng hàm **subset()** để lọc ra và tổ hợp lại các phần tử thỏa mãn yêu cầu (điểm lớn nhất) thành một danh sách mới.

```
SvmaxGK<-subset(SvGK,DiemGK == max(DiemGK))
SvmaxCK<-subset(SvCK,DiemCK == max(DiemCK))
```

- **Output:**

	DiemGK	No	MANH	TO
1	8.4	107	L03	C
2	8.4	185	L01	A

Hình 7 : Danh sách sinh viên có điểm lớn nhất ở giữa kì

	DiemCK	No	MANH	TO
1	8.97	34	L03	A
2	8.97	232	L02	B
3	8.97	246	L01	B
4	8.97	251	L02	B
5	8.97	341	L01	C

Hình 8 : Danh sách sinh viên có điểm lớn nhất ở cuối kì

8) Xác định danh sách sinh viên gồm số thứ tự (No), mã nhóm và tổ có điểm số nhỏ nhất trong tập mẫu giữa kỳ và mẫu cuối kỳ

- **Source:**
  - Sử dụng hàm subset() để lọc ra và tổ hợp lại các phần tử thỏa mãn yêu cầu(điểm nhỏ nhất) thành một danh sách mới.

```
SvminGK<-subset(SvGK,DiemGK == min(DiemGK))
SvminCK<-subset(SvCK,DiemCK == min(DiemCK))
```

- **Output:**

	DiemGK	No	MANH	TO
1	2.4	363	L03	D
2	2.4	364	L01	C

Hình 9 : Danh sách sinh viên có điểm thấp nhất ở giữa kì

	DiemCK	No	MANH	TO
1	0	1	L03	A
2	0	2	L03	A
3	0	3	L01	A
4	0	4	L02	A
5	0	5	L02	A

Hình 10 : Danh sách sinh viên có điểm thấp nhất ở cuối kì

9) Xác định điểm số trung bình của của các sinh viên trong mẫu trong tập mẫu giữa kỳ và mẫu cuối kỳ

- **Giải truyền thống**:

  $$\textbf{Điểm trung bình giữa kì} = \frac{\Sigma\textbf{Điểm giữa kì}}{366}.$$

  $$\textbf{Điểm trung bình cuối kì} = \frac{\Sigma\textbf{Điểm cuối kì}}{361}.$$

- **Source:**
  Sử dụng hàm mean() để tính giá trị trung bình trong cột chứa giá trị điểm và dùng hàm round() làm tròn điểm trung bình lên hai chữ số thập phân.

```
TBGK=round(mean(DiemGK),2)
TBCK=round(mean(DiemCK),2)
TBTK=round(mean(DiemTK),2)
```

- **Output:**

TBCK	6.72
TBGK	6.06
TBTK	6.46

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

- **Giải truyền thống**:
  Đếm xem có bao nhiêu sinh viên có số điểm bằng số điểm trung bình.
- **Source:** Dùng hàm **which()** để xét điều kiện và dùng hàm **length()** để đếm số sinh viên.

```
SVTBGK=length(which(DiemGK==TBGK))
SVTBCK=length(which(DiemCK==TBCK))
SVTBTK=length(which(DiemTK==TBTK))
```

- **Output:**

SVTBCK	0L
SVTBGK	0L
SVTBTK	1L

11) Hãy đo mức độ phân tán của điểm số (xung quanh giá trị trung bình) của mẫu giữa kỳ và mẫu cuối kỳ.

- **Giải truyền thống**: Công thức độ lệch chuẩn :

$$s = \sqrt{\frac{1}{N}\sum_{k=1}^{n}(x_i - \overline{x})^2)}$$

- **Source:**
  Sử dụng hàm **sd()** để tính mức độ phân tán (độ lệch chuẩn) của mẫu.

```
sd(DiemGK)
sd(DiemCK)
```

- **Output:**

```
> sd(DiemGK)
[1] 1.069936
> sd(DiemCK)
[1] 1.577202
```

12) Tính độ méo lệch (skewness), và độ nhọn (kurtosis) của dữ liệu trong mẫu giữa kỳ và mẫu cuối kỳ.

- **Giải truyền thống**:
  1. **Độ méo lệch(skewness)**
     - Độ méo lệch (**skewness**) của một phân phối xác suất đo lường sự **bất đối xứng** của phân phối đó. 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.
     - Công thức độ méo lệch:

$$skewness = \frac{1}{n}\frac{\sum_{k=1}^{n}(x_i - \overline{x})^3}{s^3}$$

  2. **Độ nhọn(kurosis)**
     - Độ nhọn (**kurtosis**) là một chỉ số để đo lường về đặc điểm hình dáng của một phân phối xác suất. Cụ thể hơn, nó so sánh độ cao phần trung tâm của một phân phối so sánh với một phân phối chuẩn. Phần trung tâm càng cao và nhọn, chỉ số Kurtosis của phân phối đó càng lớn.
     - Công thức độ nhọn :

$$kurtosis = \frac{1}{n}\frac{\sum_{k=1}^{n}(x_i - \overline{x})^4}{s^4}$$

- **Source:**
  - Sử dụng hàm **skewness()** để tính **độ méo lệch** của mẫu.

```
skewness(DiemGK)
skewness(DiemCK)
```

– Sử dụng hàm kurtosis() để tính **độ nhọn** của mẫu.

```
kurtosis(DiemGK)
kurtosis(DiemCK)
```

- **Output:**

```
> skewness(DiemGK)
[1] -0.4136416
> skewness(DiemCK)
[1] -1.44991
> kurtosis(DiemGK)
[1] 3.195214
> kurtosis(DiemCK)
[1] 6.18879
```

13) Tính tứ phân vị (quartile) thứ nhất (Q_1) và thứ ba (Q_3) của giữa kỳ và mẫu cuối kỳ.

- **Giải truyền thống**:
  – **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.
  – 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.
  – Giá trị tứ phân vị thứ ba **Q3** bằng trung vị phần trên.
  Ví dụ: Tập dữ liệu bao gồm {5, 7, 9, 14, 25, 34, 48}
  - Tập dữ liệu trên đã được sắp xếp theo thứ tự tăng dần, dễ dàng nhận thấy giá trị trung vị nằm giữa chính là 14.
  - Trung vị của tập dữ liệu phần dưới {5, 7, 9} là 7.
  - Và trung vị của tập dữ liệu phần trên {25, 34, 48} là 34.
  - Vậy Q1 = 7, Q2 = 14, Q3 = 34
- **Source:**
  Sử dụng hàm quantile() để tính tứ phân vị.

```
quantile(DiemGK,0.25)
quantile(DiemCK,0.25)
quantile(DiemGK,0.75)
quantile(DiemCK,0.75)
```

- **Output:**

```
> quantile(DiemGK,0.25)
25%
5.2
> quantile(DiemCK,0.25)
 25%
5.86
> quantile(DiemGK,0.75)
75%
6.8
> quantile(DiemCK,0.75)
 75%
7.93
```

14) Xác định số lượng sinh viên có điểm số nằm trong 2 mức điểm cao nhất trong tập mẫu giữa kỳ và mẫu cuối kỳ

- **Giải truyền thống**:
  Tìm điểm số cao thứ hai trong các tập mẫu. Xong đếm số lượng sinh viên có điểm số >= se_point đó.
- **Source:**
  – Tìm điểm lớn thứ hai trong bảng bằng cách dùng hàm max() để tìm giá trị lớn nhất với điều kiện **se_point** != **max_point**.

```
se_pointGK = max(DiemGK[DiemGK != max(DiemGK)])
se_pointCK = max(DiemCK[DiemCK != max(DiemCK)])
```

– Lọc ra và tổ hợp lại thành các **subset()** bao gồm các điểm số >= **se_point**.

```
sv_sepGK<-subset(DiemGK,DiemGK >= se_pointGK)
sv_sepCK<-subset(DiemCK,DiemCK >= se_pointCK)
```

– Dùng hàm **length()** để đếm số lượng sinh viên có điểm >= **se_point** trong subset.

```
length(sv_sepGK)
length(sv_sepCK)
```

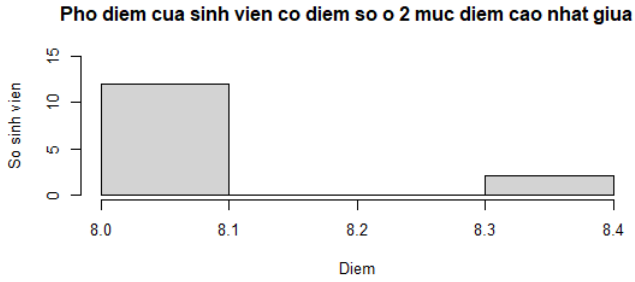
- **Output:**
  – **GK:**
  ```
  > length(sv_sepGK)
  [1] 14
  ```
  – **CK:**
  ```
  > length(sv_sepCK)
  [1] 24
  ```

15) Vẽ phổ điểm của các sinh viên có điểm số ở 2 mức điểm cao nhất trong tập mẫu giữa kỳ và mẫu cuối kỳ

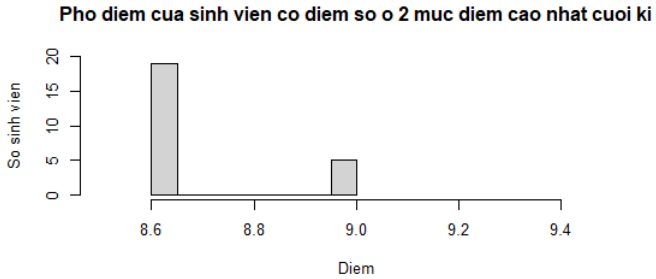
- **Source:**
  - Sử dụng hàm **hist()** để vẽ đồ thị và các hàm **xlim()**, **ylim()** để xét các khoảng cho x, y.

```
par(mfrow=c(2,2))
hist(sv_sepGK,main = "Pho diem cua sinh vien co diem so o 2 muc diem cao nhat giua ki",xlap
    ="Diem", ylap="So sinh vien", xlim=c(se\_pointGK,max(DiemGK)), ylim=c(0,15))
hist(sv_sepCK,main = "Pho diem cua sinh vien co diem so o 2 muc diem cao nhat cuoi ki",xlap
    ="Diem", ylap="So sinh vien",xlim=c(8.5,9.5), ylim=c(0,20))
```

- **Output:**



Hình 11 : Phổ điểm của sinh viên có điểm số ở 2 mức điểm cao nhất trong giữa kì



Hình 12 : Phổ điểm của sinh viên có điểm số ở 2 mức điểm cao nhất trong cuối kì

16) Xác định số lượng sinh viên có điểm số ở mức điểm cao thứ k với k cho trước trong tập mẫu giữa kỳ và mẫu cuối kỳ

- **Giải truyền thống**:
  - Sắp xếp điểm số các sinh viên theo thứ tự tăng dần, và đếm số lượng sinh viên có điểm số cao thứ **k** trong mẫu.
- **Source:**

- **Lấy k = 4 làm ví dụ.**
- Sử dụng hàm **sort()** để sắp xếp điểm lại theo thứ tự tăng dần.

```
uni_GK<-sort(unique(DiemGK))
```

- Sau đó dùng hàm **subset()** để lọc và tổ hợp lại các điểm có giá trị lớn thứ **k** trong danh sách (ở đây k = 4).

```
k_pointGK<-subset(DiemGK,DiemGK >=sort(uni_GK)[length(uni_GK)- 3])
k_pointCK<-subset(DiemCK,DiemCK >=sort(uni_CK)[length(uni_CK)- 3])
```

- Sau đó sử dụng hàm **length()** để đếm số lượng sinh viên thỏa yêu cầu.

```
length(k_pointGK)
length(k_pointCK)
```

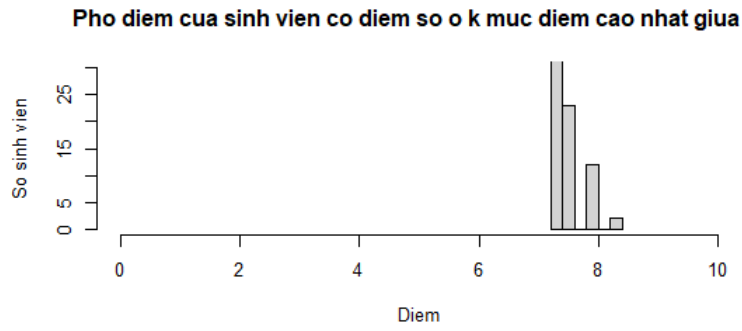
- **Output:**
  - **Lấy k = 4 làm ví dụ.**
  - **GK:**
    ```
    > length(k_pointGK)
    [1] 77
    ```
  - **CK:**
    ```
    > length(k_pointCK)
    [1] 111
    ```

17) Vẽ phổ điểm của các sinh viên có điểm số với k mức điểm cao với k cho trước trong tập mẫu giữa kỳ và mẫu cuối kỳ

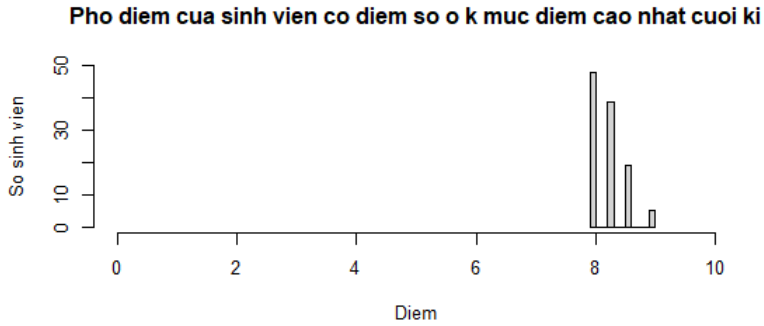
- **Source:**
  - Sử dụng hàm **hist()** để vẽ đồ thị và các hàm **xlim()**, **ylim()** để xét các khoảng cho x, y.

```
par(mfrow=c(2,2))
hist(k_pointGK, main ="Pho diem cua sinh vien co diem so o k muc diem cao nhat giua ki",
     xlap="Diem", ylap="So sinh vien",xlim=c(0,10), ylim=c(0,30))
hist(k_pointCK ,main ="Pho diem cua sinh vien co diem so o k muc diem cao nhat cuoi ki",
     xlap="Diem", ylap="So sinh vien",xlim=c(0,10), ylim=c(0,50))
```

- **Output:**
  Lấy k = 4 làm ví dụ.



Hình 13: Phổ điểm của sinh viên có điểm số ở k mức điểm cao nhất trong giữa kì



Hình 14: Phổ điểm của sinh viên có điểm số ở k mức điểm cao nhất trong cuối kì

iv) Nhóm câu hỏi liên quan đến chuẩn đầu ra môn học

1 Xác định số chuẩn đầu ra trong tập mẫu

2 Liệt kê các chuẩn đầu ra của môn học

3 Xác định tần xuất của các chuẩn đầu ra trong kỳ thi giữa kỳ và vẽ biểu đồ cho nó

4 Xác định tần xuất của các chuẩn đầu ra trong kỳ thi cuối kỳ và vẽ biểu đồ cho nó

5 Vẽ biểu đồ theo phân nhóm outcome các sinh viên có câu trả lời đúng dựa theo điểm giữa kỳ của từng nhóm

6 Vẽ biểu đồ theo phân nhóm outcome các sinh viên có câu trả lời đúng dựa theo điểm cuối kỳ của từng nhóm

7 Vẽ biểu đồ thể hiện số lượng câu hỏi của từng outcome dựa theo tập mẫu

8 Vẽ biểu đồ tần xuất tương đối tích lũy số sinh viên theo phân nhóm outcome các sinh viên có câu trả lời sai trong kỳ thi giữa kỳ trong các nhóm

9 Vẽ biểu đồ tần xuất tương đối tích lũy số sinh viên theo phân nhóm outcome các sinh viên có câu trả lời sai trong kỳ thi cuối kỳ trong các nhóm

10 Vẽ biểu đồ tần xuất tương đối tích lũy theo phân nhóm outcome các sinh viên có câu trả lời sai trong tập mẫu

v) Nhóm câu hỏi liên quan đến từng chương trong học phần

1) Xác định số chương liên quan trong tập mẫu

- **Giải truyền thống:** Số chương liên quan là 11.
- **Source**:
  - Sử dụng hàm **cbind()** để lọc ra và tổ hợp lại các cột chứa các chương liên quan.

```
testGK=cbind(GK1[12,3:27],GK1[13,3:27],GK1[14, 3:27], GK1[15, 3:27])
colnames(testGK)<- NULL

testCK=cbind(CK1[12,3:31],CK1[13,3:31],CK1[14, 3:31], CK1[15, 3:31])
colnames(testCK)<- NULL
```

  - Sử dụng hàm **strsplit()** để lưu giá trị ở các cột chứa cả 2 chương vào một biến **spt**. Sau đó, xóa các cột đó đi và thay bằng hai cột mới, mỗi cột chứa một giá trị. Các cột chứa hai giá trị lần lượt là: **15 16 30 55 59 66 88 91**.

```
spt=strsplit(testCK[,15],split=';',fixed=TRUE)[[1]]
testCK= cbind(testCK,spt[1])
testCK= cbind(testCK,spt[2])
testCK<-testCK[,-15]

spt=strsplit(testCK[,15],split=';',fixed=TRUE)[[1]]
testCK= cbind(testCK,spt[1])
testCK= cbind(testCK,spt[2])
```

```
testCK<-testCK[,-15]

spt=strsplit(testCK[,28],split=';',fixed=TRUE)[[1]]
testCK= cbind(testCK,spt[1])
testCK= cbind(testCK,spt[2])
testCK<-testCK[,-28]

spt=strsplit(testCK[,52],split=';',fixed=TRUE)[[1]]
testCK= cbind(testCK,spt[1])
testCK= cbind(testCK,spt[2])
testCK<-testCK[,-52]

spt=strsplit(testCK[,55],split=';',fixed=TRUE)[[1]]
testCK= cbind(testCK,spt[1])
testCK= cbind(testCK,spt[2])
testCK<-testCK[,-55]

spt=strsplit(testCK[,61],split=';',fixed=TRUE)[[1]]
testCK= cbind(testCK,spt[1])
testCK= cbind(testCK,spt[2])
testCK<-testCK[,-61]

spt=strsplit(testCK[,82],split=';',fixed=TRUE)[[1]]
testCK= cbind(testCK,spt[1])
testCK= cbind(testCK,spt[2])
testCK<-testCK[,-82]

spt=strsplit(testCK[,84],split=';',fixed=TRUE)[[1]]
testCK= cbind(testCK,spt[1])
testCK= cbind(testCK,spt[2])
testCK<-testCK[,-84]
```

- Sau đó gộp các danh sách GK, CK lại thành một danh sách chung là final, và sử dụng hàm **sort(unique())** để lọc ra các giá trị khác nhau của mảng.
Cuối cùng, sử dụng hàm **length()** để đếm số lượng chương.

```
final=cbind(testGK,testCK)
final<- as.numeric(final)
final_1=sort(unique(final))
num_chuong=length(final_1)
```

- **Output**:

num_chuong	11L

2) Liệt kê các chương liên quan trong tập mẫu
   - **Giải truyền thống:** Các chương trong tập mẫu lần lượt là:
     - Giữa kì : 1,2,3,4,5,6,7.
     - Cuối kì : 6,8,9,10,11.
   - **Source:**

```
    final=cbind(testGK,testCK)
    final<- as.numeric(final)
    final_1=sort(unique(final))
```

   - **Output:**

```
1 2 3 4 5 6 7 8 9 10 11
```

3) Xác định tần xuất tương đối của các chương trong kỳ thi giữa kỳ và vẽ biểu đồ cho nó
   - **Giải truyền thống:**
     Trong cả bốn mã đề, số lượng câu hỏi của từng chương xuất hiện là:
     - Chương 1 : 8 câu
     - Chương 2 : 20 câu
     - Chương 3 : 8 câu
     - Chương 4 : 12 câu

- Chương 5 : 16 câu
- Chương 6 : 20 câu
- Chương 7 : 16 câu
- Tổng số câu trong tập mẫu là: 100 câu

Tần suất tương đối của các chương xuất hiện trong kì thi giữa kì là:

$$\textbf{Tần suất tương đối của các chương} = \frac{\textbf{Số câu mỗi chương}}{100}.$$

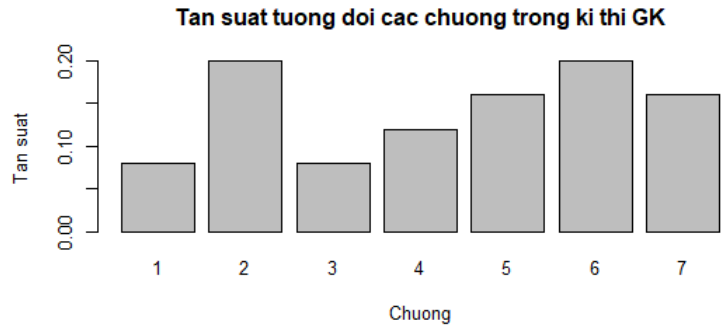
- Chương 1 : 0.08
- Chương 2 : 0.20
- Chương 3 : 0.08
- Chương 4 : 0.12
- Chương 5 : 0.16
- Chương 6 : 0.20
- Chương 7 : 0.16

- **Source:**

```
ts_GK=as.numeric(testGK)
a=length(ts_GK)
ts_GK=table(ts_GK)
ts_GK=ts_GK/a
barplot(ts_GK,main = "Tan suat tuong doi cua cac chuong trong ky thi giua ki", xlap = "
    Chuong", ylap = "Tan suat")
```

- Ta dùng lệnh **length()** để đếm số lượng chương. Tiếp đó, ta sử dụng hàm **table()** để thống kê lại số câu trong từng chương. Cuối cùng, ta dùng bảng thống kê số câu từng chương chia cho tổng số câu, ta được tần suất tương đối của các chương và lệnh **barplot()** để xuất ra đồ thị.

- **Output:**



Hình 15: Tần suất tương đối của các chương trong kỳ thi giữa kì

4) Xác định tần xuất tương đối của các chương trong kỳ thi cuối kỳ và vẽ biểu đồ cho nó

- **Giải truyền thống:**

Số lượng câu hỏi từng của từng chương trong đề thi cuối kì:

- Chương 6: 4 câu
- Chương 8: 12 câu
- Chương 9: 24 câu
- Chương 10: 60 câu
- Chương 11: 24 câu
- Tổng số câu trong tập mẫu là: 124 câu

Tần suất tương đối của các chương trong đề thi cuối kì:

$$\textbf{Tần suất tương đối của các chương} = \frac{\textbf{Số câu mỗi chương}}{124}.$$

- Chương 6: 0.03

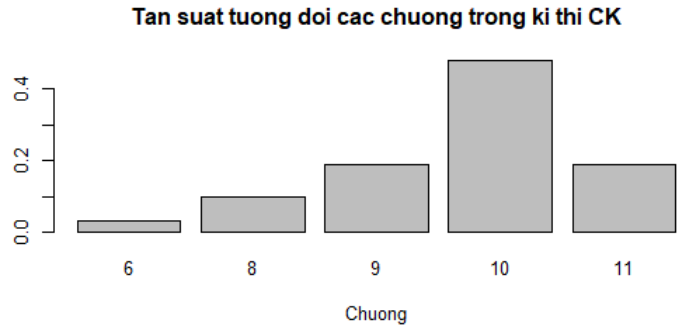
- Chương 8: 0.10
- Chương 9: 0.19
- Chương 10: 0.48
- Chương 11: 0.19

- Source:

```
ts_CK=as.numeric(testCK)
b=length(ts_CK)
ts_CK=table(ts_CK)
ts_CK=ts_CK/b
barplot(ts_CK,main = "Tan suat tuong doi cua cac chuong trong ky thi cuoi ki", xlap = "
    Chuong", ylap = "Tan suat")
```

- Ta dùng lệnh **length()** để đếm số lượng chương. Tiếp đó, ta sử dụng hàm **table()** để thống kê lại số câu trong từng chương. Cuối cùng, ta dùng bảng thống kê số câu từng chương chia cho tổng số câu, ta được tần suất tương đối của các chương và lệnh **barplot()** để xuất ra đồ thị.

- Output:



Hình 16: Tần suất tương đối của các chương trong kỳ thi cuối kì

5) Vẽ biểu đồ theo phân nhóm chương các sinh viên có câu trả lời đúng dựa theo điểm giữa kỳ của từng nhóm

- Source:

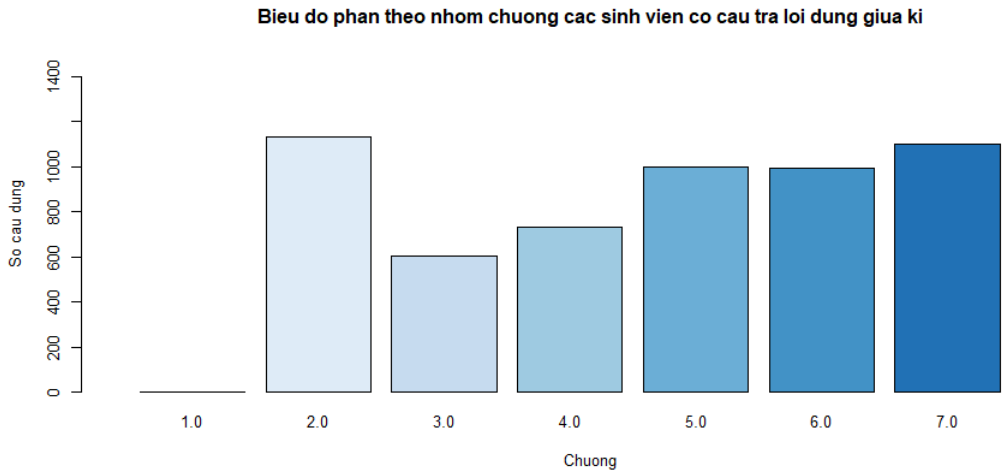
```
GK=GK[with(GK, order(MADE)), ]
gk1921=subset(GK, MADE==1921)
gk1921=gk1921[4:28]

gk1922=subset(GK, MADE==1922)
gk1922=gk1922[4:28]
gk1923=subset(GK, MADE==1923)
gk1923=gk1923[4:28]
gk1924=subset(GK, MADE==1924)
gk1924=gk1924[4:28]

gk_ans=data.frame(colSums(gk1921[1:25]),colSums(gk1922[1:25]),colSums(gk1923[1:25]),colSums
    (gk1924[1:25]))
gk_k=data.frame(GK1[12:15,3:27])
gk_k=t(gk_k)
rownames(gk_ans)<-NULL
rownames(gk_k)<-NULL
s=data.frame(gk_ans,gk_k)
colnames(s)[5]<- "X1"
colnames(s)[6]<- "X2"
colnames(s)[7]<- "X3"
colnames(s)[8]<- "X4"
tableGK=tapply(s$colSums.gk1921.1.25..,s$X1,sum) +tapply(s$colSums.gk1922.1.25..,s$X2,sum)
tableGK[[8]]<-0
names(tableGK)[8]<-c(8)
tableGK=tableGK+tapply(s$colSums.gk1923.1.25..,s$X3,sum)+tapply(s$colSums.gk1924.1.25..,s$
    X4,sum)
barplot(tableGK,main="Bieu do phan theo nhom chuong cac sinh vien co cau tra loi dung giua
    ki",xlab="Chuong",ylab="So cau dung",col=blues9, xlim=c(0,9), ylim=c(0,1500))
```

- Ta dùng lệnh order() để sắp xếp bảng số liệu giữa kỳ theo hướng tăng dần. Tiếp đó, ta sử dụng hàm subset() để lọc bảng theo điều kiện. Tiếp đến, ta dùng lệnh rownames() và colnames() để chuyển đổi các giá trị trong cột thành hàng và ngược lại. Sau đó, ta dùng hàm tapply() để tính toán giá trị cần tìm.

- Output:



Hình 17: Biểu đồ theo phân nhóm chương các sinh viên có câu trả lời đúng dựa theo điểm giữa kỳ của từng nhóm

6) Vẽ biểu đồ theo phân nhóm chương các sinh viên có câu trả lời đúng dựa theo điểm cuối kỳ của từng nhóm

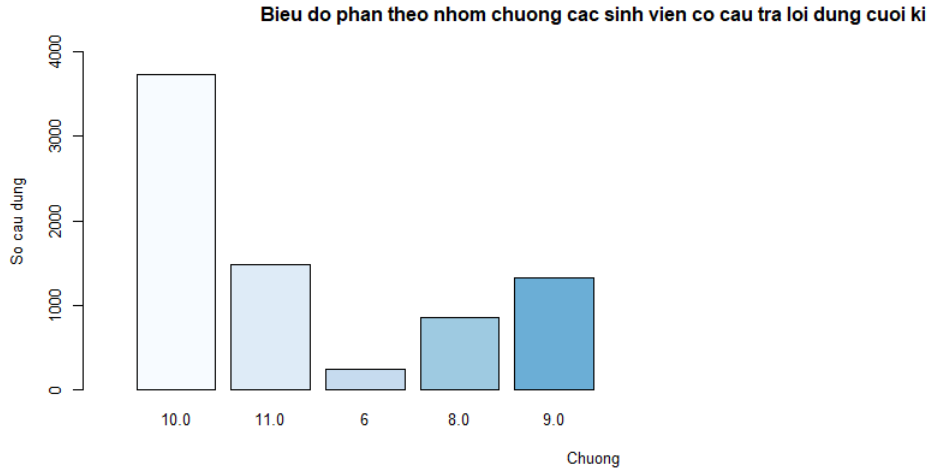
- Source:

```
CK=CK[with(CK, order(MADE)), ]
ck1921=subset(CK, MADE==1921)
ck1921=ck1921[4:32]
ck1922=subset(CK, MADE==1922)
ck1922=ck1922[4:32]
ck1923=subset(CK, MADE==1923)
ck1923=ck1923[4:32]
ck1924=subset(CK, MADE==1924)
ck1924=ck1924[4:32]

ck_ans=data.frame(colSums(ck1921[1:29]),colSums(ck1922[1:29]),colSums(ck1923[1:29]),colSums
    (ck1924[1:29]))
ck_k=data.frame(CK1[12:15,3:31])
ck_k=t(ck_k)
u=data.frame(ck_ans,ck_k)
colnames(u)[5]<- "X1"
colnames(u)[6]<- "X2"
colnames(u)[7]<- "X3"
colnames(u)[8]<- "X4"
tableCK=tapply(u$colSums.ck1921.1.29..,u$X1,sum)+tapply(u$colSums.ck1922.1.29..,u$X2,sum)+
    tapply(u$colSums.ck1923.1.29..,u$X3,sum)+tapply(u$colSums.ck1924.1.29..,u$X4,sum)
names(tableCK)[3]<-c(6)
tableCK[[1]]<-tableCK[[1]]+244
tableCK[[5]]=tableCK[[5]]+tableCK[[3]]+tableCK[[6]]
tableCK=tableCK[-6]
barplot(tableCK,main="Bieu do phan theo nhom chuong cac sinh vien co cau tra loi dung cuoi
    ki",xlab="Chuong",ylab="So cau dung",col=blues9, xlim=c(0,12), ylim=c(0,4000))
```

- Ta dùng lệnh order() để sắp xếp bảng số liệu giữa kỳ theo hướng tăng dần. Tiếp đó, ta sử dụng hàm subset() để lọc bảng theo điều kiện. Tiếp đến, ta dùng lệnh rownames() và colnames() để chuyển đổi các giá trị trong cột thành hàng và ngược lại. Sau đó, ta dùng hàm tapply() để tính toán giá trị cần tìm.

- Output:



Hình 18: Biểu đồ theo phân nhóm chương các sinh viên có câu trả lời đúng dựa theo điểm cuối kỳ của từng nhóm

7) Vẽ biểu đồ thể hiện số lượng câu hỏi cửa từng chương dựa theo tập mẫu

- Source:

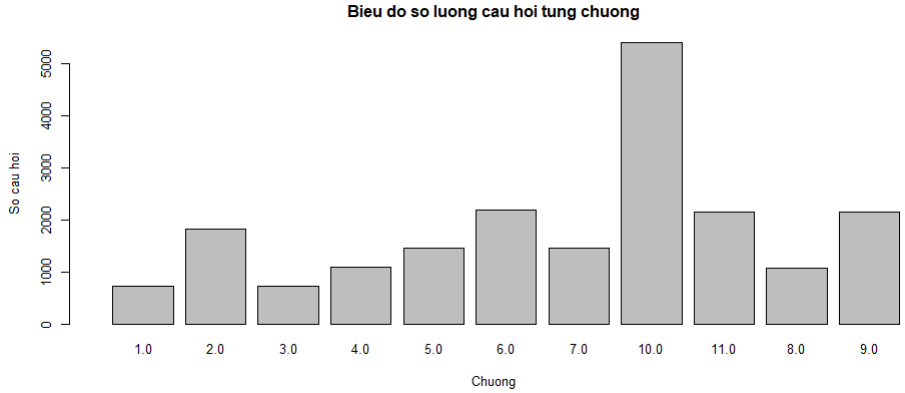
```
num_quesGK=table(gk_k[1:25,1])*nrow(gk1921)+table(gk_k[1:25,2])*nrow(gk1922)
num_quesGK[[8]]<-0
names(num_quesGK)[8]<-c(8)
num_quesGK=num_quesGK+table(gk_k[1:25,3])*nrow(gk1923)+table(gk_k[1:25,4])*nrow(gk1924)

num_quesCK=table(ck_k[1:29,1])*nrow(ck1921)+table(ck_k[1:29,2])*nrow(ck1922)+table(ck_k
    [1:29,3])*nrow(ck1923)+table(ck_k[1:29,4])*nrow(ck1924)
names(num_quesCK)[3]<-c(6)
num_quesCK[[1]]<-num_quesCK[[1]]+num_quesCK[[6]]
num_quesCK[[5]]<-num_quesCK[[5]]+num_quesCK[[3]]+num_quesCK[[6]]
num_quesCK=num_quesCK[-6]

final=c(num_quesGK,num_quesCK)
final[[6]]<-final[[6]]+final[[11]]
final=final[-11]
final[[8]]<-final[[8]]+final[[11]]
final=final[-11]
barplot(final,main="Bieu do so luong cau hoi tung chuong", xlap="Chuong", ylap="So cau hoi"
    , xlim=c(0,13), ylim=c(0,max(final)))
```

- Ta dùng lệnh **table()** để thống kê lại số lượng câu hỏi trong tập mẫu và dùng lệnh **barplot()** để vẽ đồ thị.

- Output:



Hình 19: Biểu đồ thể hiện số lượng câu hỏi cửa từng chương dựa theo tập mẫu

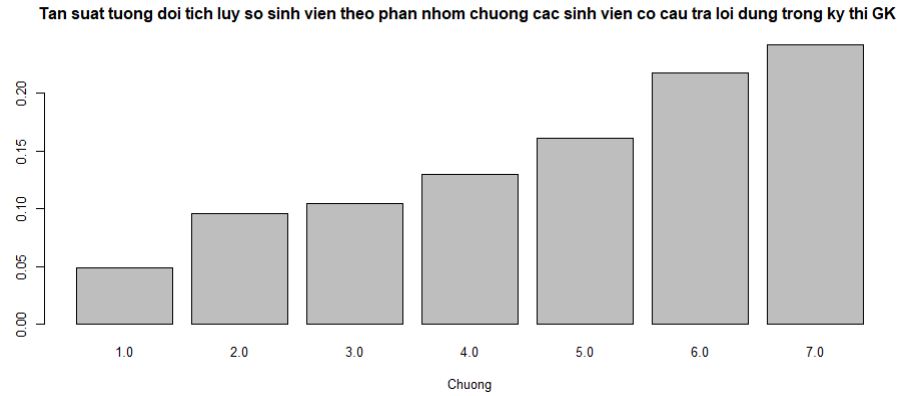
8) Vẽ biểu đồ tần xuất tương đối tích lũy số sinh viên theo phân nhóm chương các sinh viên có câu trả lời sai trong kỳ thi giữa kỳ trong các nhóm

- Source:

```
wrongGK=num_quesGK-tableGK
ts_wrongGK=cumsum(wrongGK)
barplot(ts_wrongGK/sum(ts_wrongGK),main="Tan suat tuong doi tich luy so sinh vien theo phan
      nhom chuong cac sinh vien co cau tra loi sai trong ky thi giua ki trong cac nhom"
     ,xlab="Chuong",ylab="Tan suat")
```

- Hàm **cumsum()** trả về một vectơ có các phần tử là tổng tích lũy, tích, cực tiểu hoặc cực đại của các phần tử của đối số. Tiếp đến, ta chia tổng tích lũy cho **num**, thu được tần suất tương đối tích lũy và cuối cùng, ta dùng hàm **barplot()** để vẽ đồ thị.

- Output:



Hình 20: Biểu đồ tần xuất tương đối tích lũy số sinh viên theo phân nhóm chương các sinh viên có câu trả lời sai trong kỳ thi giữa kỳ trong các nhóm

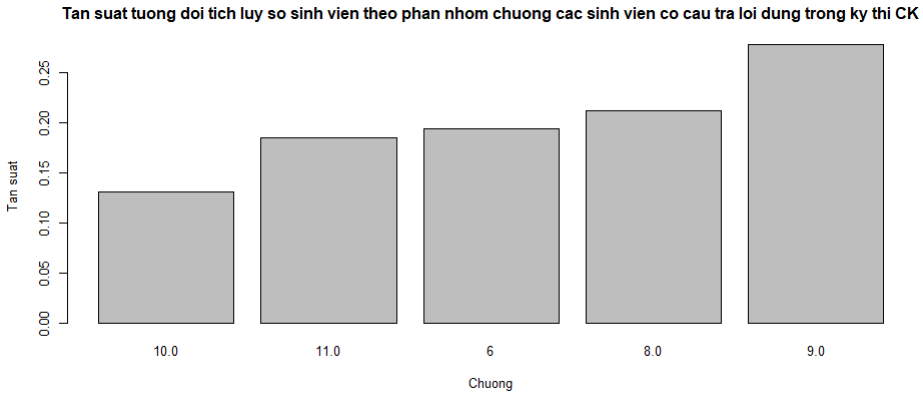
9) Vẽ biểu đồ tần xuất tương đối tích lũy số sinh viên theo phân nhóm chương các sinh viên có câu trả lời sai trong kỳ thi cuối kỳ trong các nhóm

- Source:

```
wrongCK=num_quesCK-tableCK
ts_wrongCK=cumsum(wrongCK)
barplot(ts_wrongCK/sum(ts_wrongCK),main="Tan suat tuong doi tich luy so sinh vien theo phan
      nhom chuong cac sinh vien co cau tra loi sai trong ky thi cuoi ki trong cac nhom"
     ,xlab="Chuong",ylab="Tan suat")
```

- Hàm **cumsum()** trả về một vectơ có các phần tử là tổng tích lũy, tích, cực tiểu hoặc cực đại của các phần tử của đối số. Tiếp đến, ta chia tổng tích lũy cho **num**, thu được tần suất tương đối tích lũy và cuối cùng, ta dùng hàm **barplot()** để vẽ đồ thị.

- Output:



Hình 21: Biểu đồ tần xuất tương đối tích lũy số sinh viên theo phân nhóm chương các sinh viên có câu trả lời sai trong kỳ thi cuối kỳ trong các nhóm

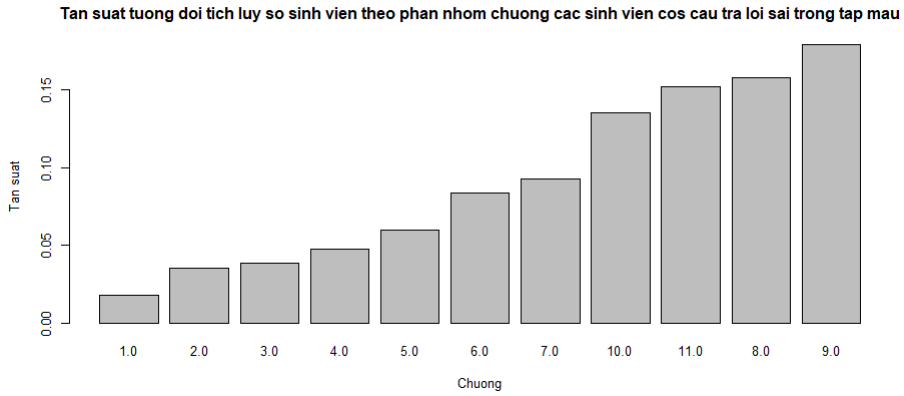
10) Vẽ biểu đồ tần xuất tương đối tích lũy theo phân nhóm chương các sinh viên có câu trả lời sai trong tập mẫu

- Source:

```
wrongfinal=c(num_quesGK-tableGK,num_quesCK-tableCK)
ts_wrongfinal=cumsum(wrongfinal)
barplot(ts_wrongfinal/sum(ts_wrongfinal),main="tTan suat tuong doi tich luy so sinh vien
    theo phan nhom chuong cac sinh vien co cau tra loi sai trong tap mau ",xlab="Chuong"
    ,ylab="Tan suat")
```

- Hàm **cumsum()** trả về một vectơ có các phần tử là tổng tích lũy, tích, cực tiểu hoặc cực đại của các phần tử của đối số. Tiếp đến, ta chia tổng tích lũy cho **num**, thu được tần suất tương đối tích lũy và cuối cùng, ta dùng hàm **barplot()** để vẽ đồ thị.

- Output:



Hình 22: Biểu đồ tần xuất tương đối tích lũy số sinh viên theo phân nhóm chương các sinh viên có câu trả lời sai trong tập mẫu

vi) Đánh giá và suy luận, phân nhóm sinh viên, về độ khó của chương hoặc về độ khó của outcome
Mỗi nhóm sẽ thực hiện các câu tương ứng theo công thức tính:

1) Số lượng sinh viên có điểm số tốt ở giữa kỳ (điểm lớn hơn hay bằng g) có làm bài tốt ở cuối kỳ không? Nếu có hãy thực hiện:

   a. Hãy xác định g

   b. Liệt kê danh sách các sinh viên phù hợp đó

   - Source:

```
g=7.5
Diem=cbind(GK$No,DiemGK,DiemCK)
colnames(Diem)[1]<-"No"
good <- subset(Diem,DiemGK >=g & DiemCK >=g)
```

   - Lệnh **cbind()** dùng để kết hợp nhiều ma trận lại theo chiều ngang, sau đó chuyển ma trận về thành chiều dọc. Cuối cùng, dùng lệnh **subset()** để lấy mảng con có điều kiện với điều kiện là g cho trước.

   - Output:

```
      No DiemGK DiemCK
[1,]  72    7.6   8.62
[2,]  91    7.6   7.93
[3,] 135    7.6   7.93
[4,] 140    7.6   7.93
[5,] 160    7.6   8.28
[6,] 350    7.6   8.28
[7,]  48    8.4   8.62
[8,] 125    7.6   7.93
[9,] 134    8.0   7.93
10,] 162    7.6   7.59
```

Hình 23: Kết quả số lượng sinh viên có điểm số tốt ở giữa kì và điểm số của họ trong kỳ thi cuối kì

2) Số lượng sinh viên làm đúng nhiều câu (q) theo nhóm outcome hay chương ở giữa kỳ có làm tốt các câu thuộc nhóm tương ứng ở cuối kỳ không? Nếu có hãy thực hiện:
   a. Hãy xác định q
   b. Liệt kê danh sách các sinh viên phù hợp đó
3) Các nhóm làm bài tốt như nhau hay một nhóm làm bài tốt hơn ở giữa kỳ? Hãy đưa ra lý do cho câu trả lời của bạn.
4) Các nhóm làm bài tốt như nhau hay một nhóm làm bài tốt hơn ở cuối kỳ? Hãy đưa ra lý do cho câu trả lời của bạn.
   - Source:

```
last_test = data.frame(DiemCK[1:361], CK$MANH)
colnames(last_test)[1]<-"DiemCK"
colnames(last_test)[2]<-"MANH"
print(tapply(last_test$DiemCK,last_test$MANH, mean))
```

   - Output:

```
      L01       L02       L03
6.646154 6.719355 7.026316
```

Hình 24: Kết quả trung bình từng nhóm

5) Chương được xem là khó khi số lượng sinh viên (s) trả lời số câu đúng thấp.
   a. Hãy xác định s
   b. Liệt kê danh sách các sinh viên đó ở giữa kỳ và cuối kỳ
6) Xác định những câu hỏi cần giảng viên khảo sát kỹ lại (đề khó hay sinh viên hiểu nhầm kiến thức hoặc hiểu nhầm đề„..); sau đó, xác định
   a. các chương có nhiều câu hỏi cần kháo sát kỹ lại;
   b. các outcome nào có nhiều câu hỏi cần kháo sát kỹ lại.
   - Source:

```
dif_GK=round(tableGK/num_quesGK,2) #ti le cau tra loi dung trong tong so cau hoi o ki thi GK
partGK=names(which(dif_GK==min(dif_GK)))
related_GK=GK1[12:15,3:27]
related_GK=t(related_GK)
numGK1921=names(which(related_GK[,1]==partGK))
numGK1922=names(which(related_GK[,2]==partGK))
numGK1923=names(which(related_GK[,3]==partGK))
numGK1924=names(which(related_GK[,4]==partGK))
svGK1921=subset(GK[,1:3],GK$MADE==1921 & GK[numGK1921[1]]==1 & GK[numGK1921[2]]==1)
svGK1922=subset(GK[,1:3],GK$MADE==1922 & GK[numGK1922[1]]==1 & GK[numGK1922[2]]==1)
svGK1923=subset(GK[,1:3],GK$MADE==1923 & GK[numGK1923[1]]==1 & GK[numGK1923[2]]==1)
svGK1924=subset(GK[,1:3],GK$MADE==1924 & GK[numGK1924[1]]==1 & GK[numGK1924[2]]==1)
sv__GK=rbind(svGK1921,svGK1922,svGK1923,svGK1924)

dif_CK=round(tableCK/num_quesCK,2)
partCK=names(which(dif_CK==min(dif_CK)))
related_CK=CK1[12:15,3:31]
related_CK=t(related_CK)
numCK1921=names(which(related_CK[,1]==partCK | related_CK[,1]=="6;9" |related_CK[,1]=="9;10" ))
numCK1922=names(which(related_CK[,2]==partCK | related_CK[,2]=="6;9" |related_CK[,2]=="9;10" ))
numCK1923=names(which(related_CK[,3]==partCK | related_CK[,3]=="6;9" |related_CK[,3]=="9;10" ))
numCK1924=names(which(related_CK[,4]==partCK | related_CK[,4]=="6;9" |related_CK[,4]=="9;10" ))

svCK1921=subset(CK,CK$MADE==1921 & (CK[numCK1921[1]]==1 & CK[numCK1921[2]]==1 & CK[numCK1921[3]]==1 & CK[numCK1921[4]]==1 & CK[numCK1921[5]]==1 & CK[numCK1921[6]]==1))
```

```
svCK1922=subset(CK,CK$MADE==1922 & (CK[numCK1922[1]]==1 & CK[numCK1922[2]]==1 & CK[
    numCK1922[3]]==1 & CK[numCK1922[4]]==1 & CK[numCK1922[5]]==1 & CK[numCK1922[6]]==1))
svCK1923=subset(CK,CK$MADE==1923 & (CK[numCK1923[1]]==1 & CK[numCK1923[2]]==1 & CK[
    numCK1923[3]]==1 & CK[numCK1923[4]]==1 & CK[numCK1923[5]]==1 & CK[numCK1923[6]]==1))
svCK1924=subset(CK,CK$MADE==1924 & (CK[numCK1924[1]]==1 & CK[numCK1924[2]]==1 & CK[
    numCK1924[3]]==1 & CK[numCK1924[4]]==1 & CK[numCK1924[5]]==1 & CK[numCK1924[6]]==1))

sv__CK=rbind(svCK1921,svCK1922,svCK1923,svCK1924)
```

- Đầu tiên, ta tính tỉ lệ câu trả lời đúng bằng cách lấy số liệu **tableGk** sau đó chia cho tổng số câu hỏi và dùng hàm làm tròn **round()** để làm tròn tỉ lệ đó. - Lệnh **names()** dùng để đặt hoặc gán tên cho một đối tượng, lệnh **which()** sẽ trả về vị trí của phần tử, lệnh **min** để lấy phần tử có giá trị nhỏ nhất,lệnh **subset()** dùng để lấy chuỗi con có điều kiện, lệnh **rbind()** hay **cbind()** dùng để lấy ma trận theo chiều dọc / chiều ngang.

- Output:

14.0	15.0	16.0	17.0	18.0	19.0	20.0	21.0	22.0	23.0	24.0	25.0	26.0	27.0	28.0	29.0	MADE
No data available in table																

Hình 25: Các outcome cần kiểm tra lại.

7) Nếu xét kết quả làm bài của sinh viên là một chuỗi giá trị, hãy xác định
   a. "các chuỗi con dài nhất có độ tương tự cao" (độ tương tự được mô tả bởi một ngưỡng)
   b. "các chuỗi con dài có độ tương tự cao nhất" (trong các chuỗi có độ dài dài hơn một ngưỡng cho trước, chọn chuỗi có độ dài dài nhất);
   c. "các chuỗi con dài nhất có độ tương tự cao nhất" (lưu ý, nếu chuôi "ABCD" là chuỗi con có độ tương tự cao nhất thì chuỗi "ABC" cũng là chuỗi con có độ tương tự cao nhất do chuỗi "ABC" luôn có độ tương tự cao hơn hoặc bằng độ tương tự của chuỗi "ABCD"; nhưng trong trường hợp này, ta chỉ xét chọn chuỗi "ABCD").

6 Hướng dẫn và yêu cầu

6.1 Hướng dẫn

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

6.2 Yêu cầu

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

- Thời gian làm bài: Từ ngày 29/03/2021 – 18g00 ngày 24/04/2021.

  Đối với mỗi bài toán, yêu cầu sinh viên trình bày lời giải theo lối truyền thống, sử dụng các công thức, kết quả lý thuyết trong phần kiến thức chuẩn bị. Đồng thời, sau đó trình bày kết quả tính toán và biểu đồ minh họa bằng R.
- Trình bày cả code R và kết quả tính toán trong R giống như file mẫu.
- Viết báo cáo theo đúng **bố cục như trong file mẫu** bằng LaTeX.

- Mỗi nhóm khi nộp bài **cần phải nộp theo file log (nhật ký)** ghi rõ: tiến độ công việc, phân công nhiệm vụ, trao đổi của các thành viên,...

6.3 Nộp bài

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

7 Cách đánh giá và xử lý gian lận

7.1 Đánh giá

Mỗi bài làm sẽ được đánh giá như sau.

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

7.2 Xử lý gian lận

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

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

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

Tài liệu

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

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

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