Buổi 0. Thực hành XSTK cơ bản

Ho Huu Binh

2025-10-13

1 Giới thiệu R và RStudio

1.1 Giới thiệu

R là một gói phần mềm thống kê có nhiều điểm tương đồng với ngôn ngữ lập trình thống kê S. Phiên bản sơ bộ của S được Bell Labs tạo ra vào những năm 1970, được thiết kế để trở thành một ngôn ngữ lập trình tương tự C nhưng dành cho thống kê. John Chambers là một trong những người phát minh ra ngôn ngữ này, và ông đã giành được Giải thưởng của Hiệp hội Máy tính năm 1999 cho ngôn ngữ này. R là một phần mềm hoàn toàn miễn phí. R và RStudio có mối liên hệ chặt chẽ - chúng không phải là cùng một phần mềm, mà là hai lớp khác nhau của cùng một hệ thống làm việc. R là ngôn ngữ lập trình và môi trường tính toán thống kê. Nó được phát triển để:

- Xửlý dữ liệu,
- Phân tích thống kê,
- Môphỏng ngẫu nhiên,
- Vẽ đồ thị và trực quan hóa dữ liệu.

Khi cài R, người dùng đã có thể chạy lệnh trong giao diện dòng lệnh (console) đơn giản của R, nhưng giao diện này rất thô sơ. Mở R (không dùng RStudio), sẽ thấy một cửa sổ đen hoặc trắng, nơi chỉ có thể gõ lệnh kiểu:

```
x <- rnorm(10)
mean(x)
plot(x)
```

RStudio là một môi trường phát triển tích hợp (IDE- Integrated Development Environment) dành riêng cho R. Nó không thay thế R, mà chạy dựa trên R. RStudio giúp người dùng làm việc với R dễ dàng và trực quan hơn thông qua:

- Source Editor: viết, lưu và chạy script .R.
- Console: chạy lệnh trực tiếp.
- Environment: xem toàn bộ biến, dataset đang tồn tại.

2 Một số lưu ý ngoài lề

2.1 Đặt tên biến

Trong R, khi chúng ta tạo một **object** (biến), ta cần tuân thủ một số quy tắc đặt tên:

1. Tên biến **bắt đầu bằng chữ cái** (không được bắt đầu bằng số).
2. Chỉ được chứa: **chữ cái (a–z, A–Z)**, **chữ số (0–9)**, dấu gạch dưới `\_`, hoặc dấu chấm `.`.
3. **Phân biệt chữ hoa và chữ thường** (case-sensitive).
    - Ví dụ: `Data` và `data` là hai biến khác nhau.

2.1.1 Ví dụ về các phong cách đặt tên

Trong thực tế, có nhiều phong cách (convention) để đặt tên biến:

```
i_use_snake_case      # snake_case: dùng dấu gạch dưới, dễ đọc, thường dùng nhất
otherPeopleUseCamelCase   # camelCase: viết hoa chữ cái đầu mỗi từ (trừ từ đầu)
some.people.use.periods   # dùng dấu chấm (ít dùng trong R hiện nay)
some.people.use.periods   # dùng dấu chấm (ít dùng trong R hiện nay)
And_aFew.People_RENOUNCEconvention  # phong cách hỗn hợp, khó đọc, KHÔNG nên dùng
```

Trong thực tế, **snake_case (my_variable_name)** được khuyên dùng vì dễ đọc, dễ thống nhất.Trong RStudio, khi ta đặt tên quá dài, hãy tận dụng tính năng gợi ý. Ví dụ:

```
this_is_a_really_long_name <- 2.5
```

Khi gõ this rồi nhấn TAB, RStudio sẽ tự động hoàn thành.Nếu chúng ta gõ nhầm, R sẽ báo lỗi ngay:

```
r_rocks <- 2^3
r_rock   # lỗi: object 'r_rock' not found
```

2.2 Khoảng trắng (Spaces)

Khoảng trắng giúp code thoáng, dễ đọc. Một số quy tắc:

- Luôn đặt khoảng trắng xung quanh toán tử: ‘+‘ ‘-‘ ‘*‘ ‘ / ‘ ‘==‘ ‘<-‘.
- Không đặt khoảng trắng ngay sau (hoặc trước).
- Luôn đặt khoảng trắng sau dấu phẩy.

```
# Đúng
z <- (a + b)^2 / d
mean(x, na.rm = TRUE)
# Sai
z<-( a + b ) ^ 2/d
mean (x ,na.rm=TRUE)
```

Đồng thời ta cũng có thể dùng canh lề bằng khoảng trắng để dễ đọc hơn khi tạo nhiều biến:

```
flights |>
  mutate(
    speed      = distance / air_time,
    dep_hour   = dep_time %/% 100,
    dep_minute = dep_time %%  100
  )
```

2.3 Sử dụng pipes

Trong R, luôn có khoảng trắng trước |>. Mỗi |> thường đặt ở cuối dòng để dễ thêm bước mới.

```
# Đúng
flights |>
  filter(!is.na(arr_delay), !is.na(tailnum)) |>
  count(dest)
# Sai
flights|>filter(!is.na(arr_delay), !is.na(tailnum))|>count(dest)
```

- Nếu hàm có nhiều tham số đặt tên (như `mutate()`, `summarize()`), viết mỗi tham số trên một dòng.
- Nếu hàm không có tham số đặt tên (như `select()`, `filter()`), có thể viết trên 1 dòng nếu ngắn gọn.

Ví dụ dưới đây để hiểu rõ hơn:

```
# Đúng
flights |>
  group_by(tailnum) |>
  summarize(
    delay = mean(arr_delay, na.rm = TRUE),
    n = n()
  )
```

Sau bước đầu tiên, mỗi dòng thụt vào 2 dấu cách. Khi liệt kê nhiều tham số, thụt vào thêm 2 dấu cách nữa. Không nên viết pipeline quá dài (>10-15 dòng). Thay vào đó, hãy chia nhỏ thành nhiều bước và gán tên biến trung gian.

2.4 Chia đoạn bằng comment

Khi script dài, hãy chia thành các phần bằng comment có gạch ngang:

```
# Load data --------------------------------------

# Clean data -------------------------------------

# Plot results -----------------------------------
```

2.5 Lưu và đặt tên

Trong RStudio, Script sẽ được tự động lưu khi thoát. Tuy nhiên, ta nên **chủ động đặt tên file rõ ràng** thay vì để mặc định như `Untitled1.R`. Ở đây có 3 quy tắc quan trọng đặt tên mà ta cần chú ý:

- Máy đọc được: không dùng khoảng trắng, ký hiệu đặc biệt.
- Người đọc được: tên file gợi tả nội dung.
- Thứ tự rõ ràng: đánh số thứ tự ở đầu tên file.

Ví dụ giả sử ta có thư mục chứa các file sau:

```
alternative model.R
code for exploratory analysis.r
finalreport.qmd
FinalReport.qmd
fig 1.png
Figure_02.png
model_first_try.R
run-first.r
temp.txt
```

Nếu dự án lớn hơn, hãy **phân loại thêm thư mục con** để quản lý tốt hơn, ví dụ: `scripts/`, `figures/`, `data/`...

3 Bài tập

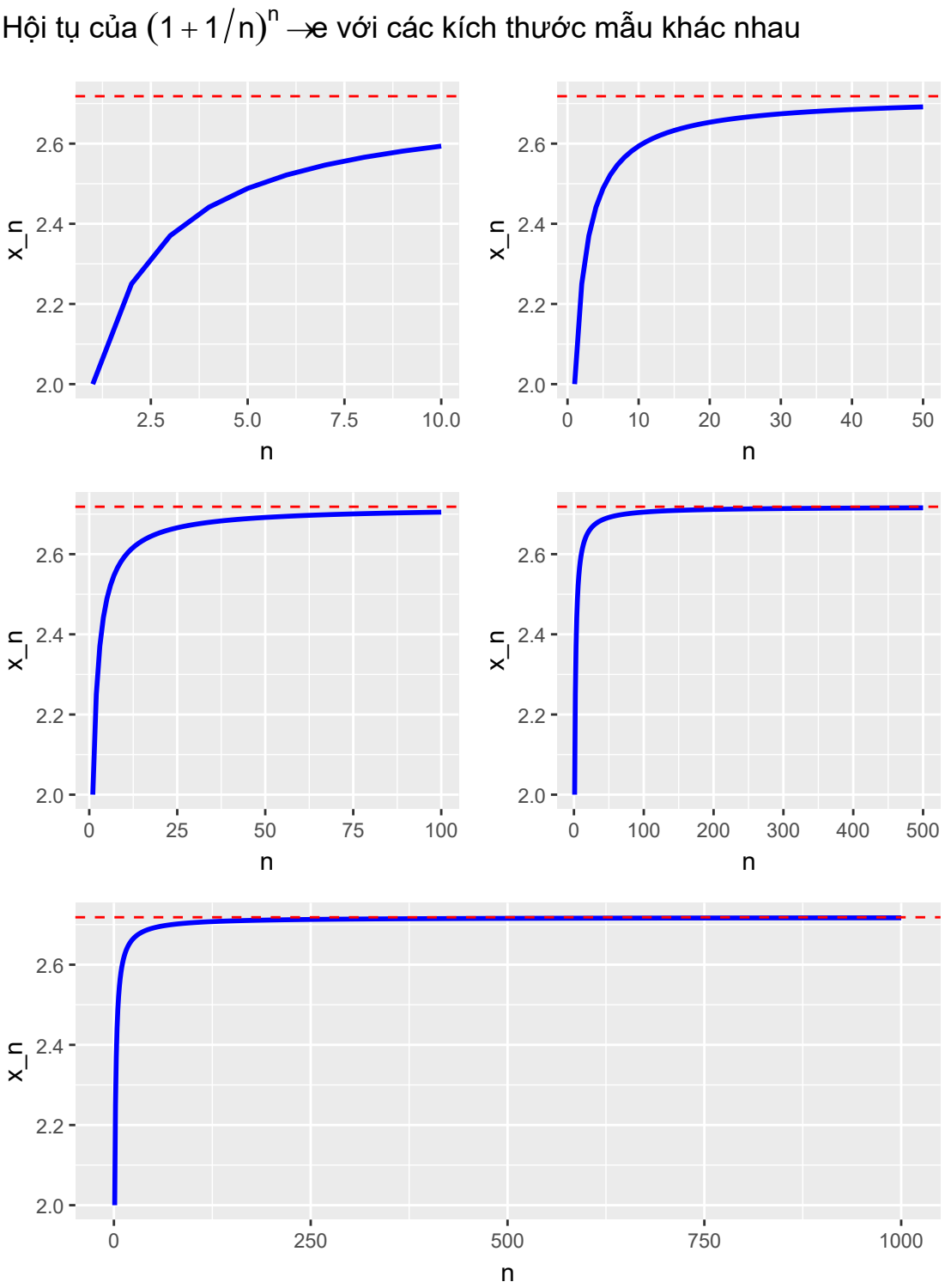
1. Hội tụ dãy số $(1 + 1/n)^n \to e$

Minh họa quá trình hội tụ của dãy $(1 + 1/n)^n$ đến hằng số Euler (e ≈2.71828). Hãy cố gắng viết hàm để lồng vào cho gọn.

1. Viết hàm vẽ đồ thị biểu diễn $x_n = (1 + 1/n)^n$.
2. Hiển thị đường ngang $y = e$ để so sánh.
3. Thử với các kích thước mẫu khác nhau: $n = 10, 50, 100, 500, 1000$.

4. Kết hợp các đồ thị con thành một đồ thị lớn với tiêu đề thích hợp. Có thể tham khảo thư viện `patchwork` để kết hợp nhiều đồ thị con.

Dưới đây là hình vẽ tham khảo:

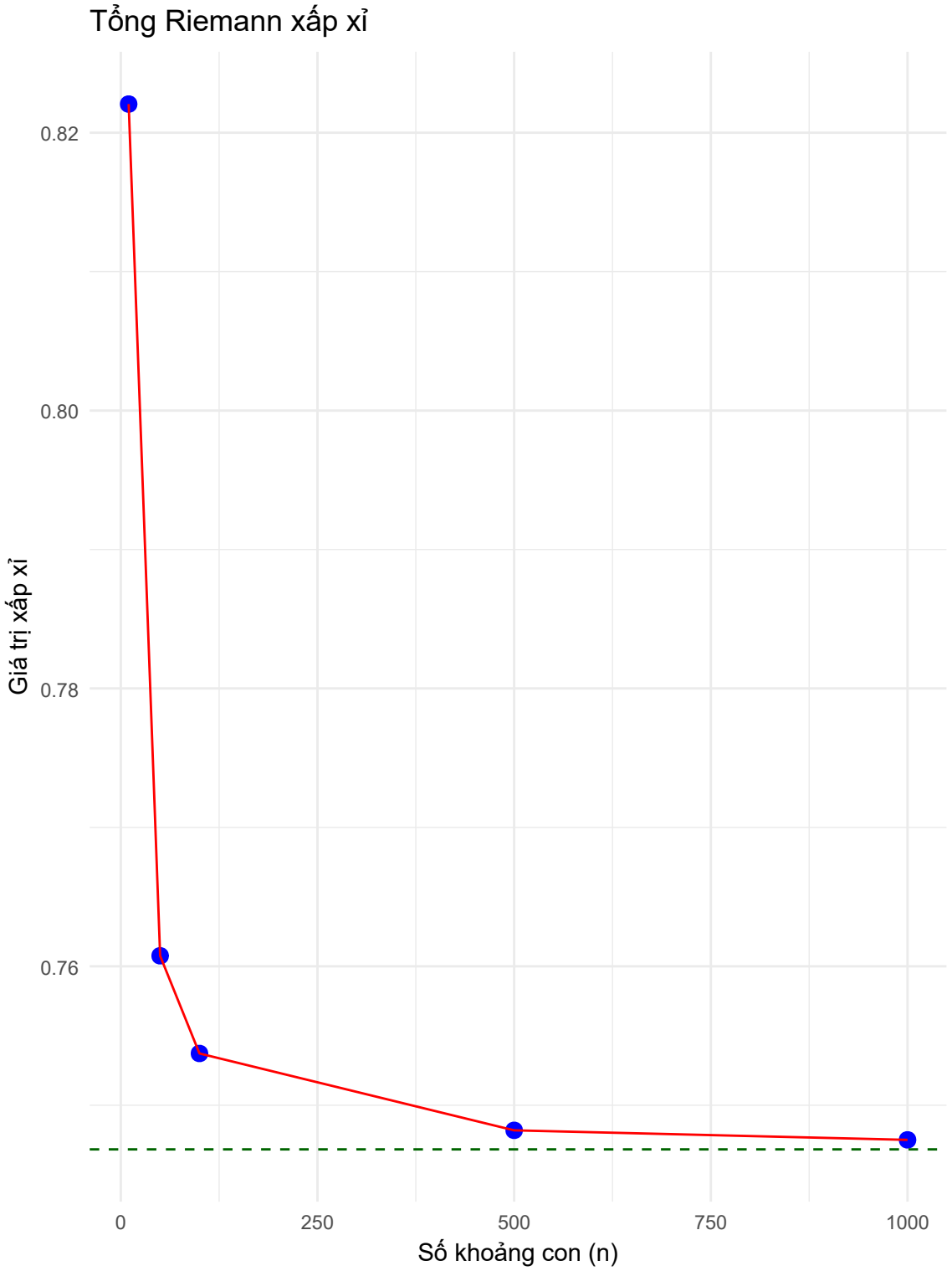


2. Xấp xỉ tích phân bằng tổng Riemann

Xấp xỉ tích phân $\int_0^1 e^{-x^2}\,dx$ bằng tổng Riemann và kiểm tra hội tụ khi tăng số khoảng chia.

Gợi ý. Hãy viết hàm `riemann_sum(f, a, b, n)` để tính tổng Riemann của hàm `f` bất kỳ trên khoảng a, b với số khoảng con n

Dưới đây là hình vẽ tham khảo:



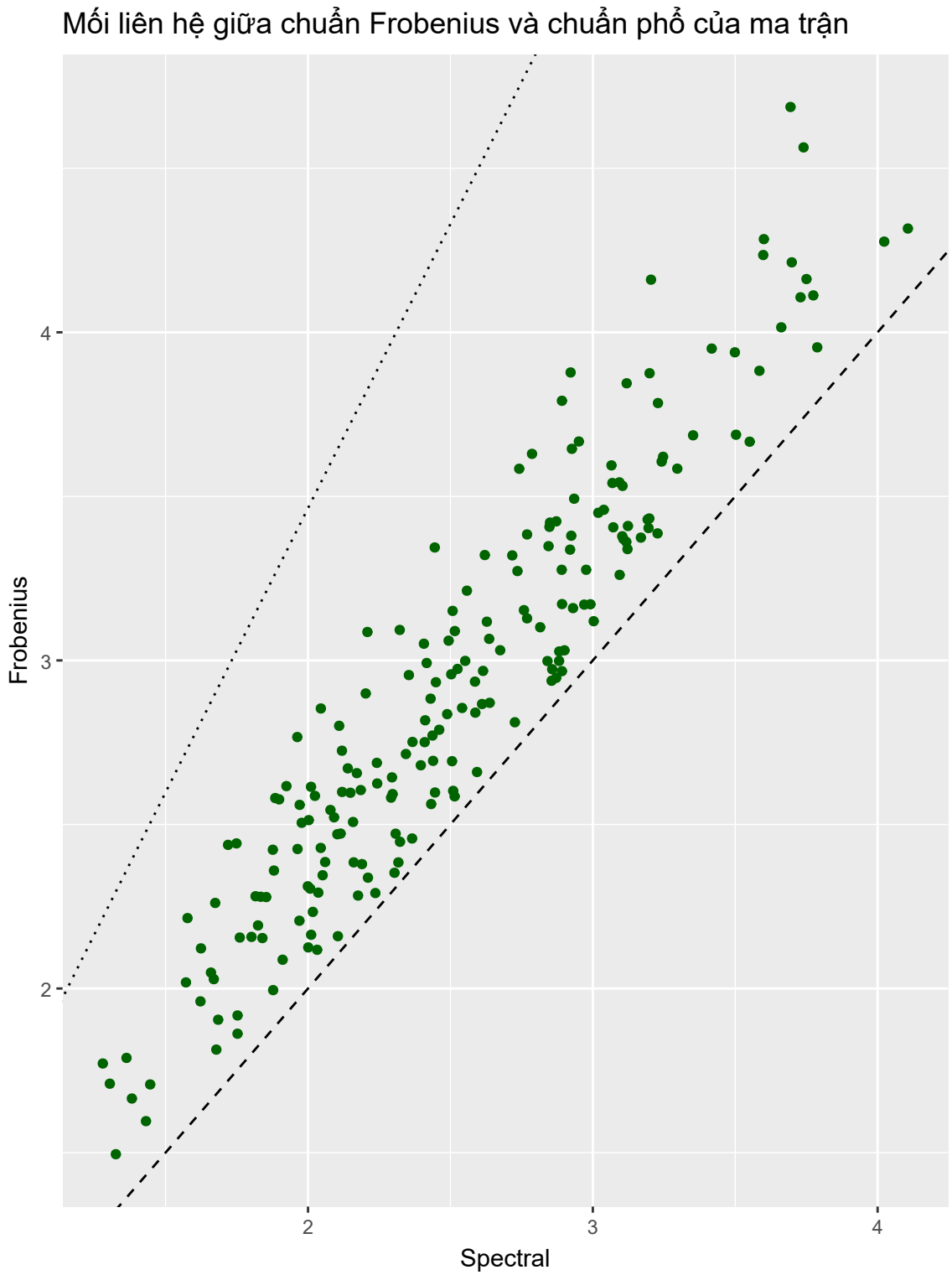
3. Chuẩn Frobenius và chuẩn phổ của ma trận Hãy viết hàm tính chuẩn Frobenius và chuẩn phổ (spectral norm). Kiểm tra bất đẳng thức $\|A\|_2 \leq \|A\|_F \leq \sqrt{r}\|A\|_2$, với $r = \text{rank}(A)$.

Nhớ lại rằng

$$\|A\|_F = \sqrt{\text{trace}(A^TA)} = \sqrt{\sum_{i,j} a_{ij}^2}, \quad \|A\|_2 = \sqrt{\lambda_{\max}(A^TA)}.$$

Gợi ý. Để tính chuẩn phổ, ta chỉ cần tính giá trị kỳ dị lớn nhất của ma trận A^TA. Trong R, hàm `eigen()` có thể giúp ta tìm giá trị riêng của ma trận. Hàm `norm()` với tham số `type=F` hay `type=2` cho ta chuẩn Frobenius và chuẩn phổ. Tuy nhiên, hãy thử tự viết hàm rồi kiểm tra lại nha.

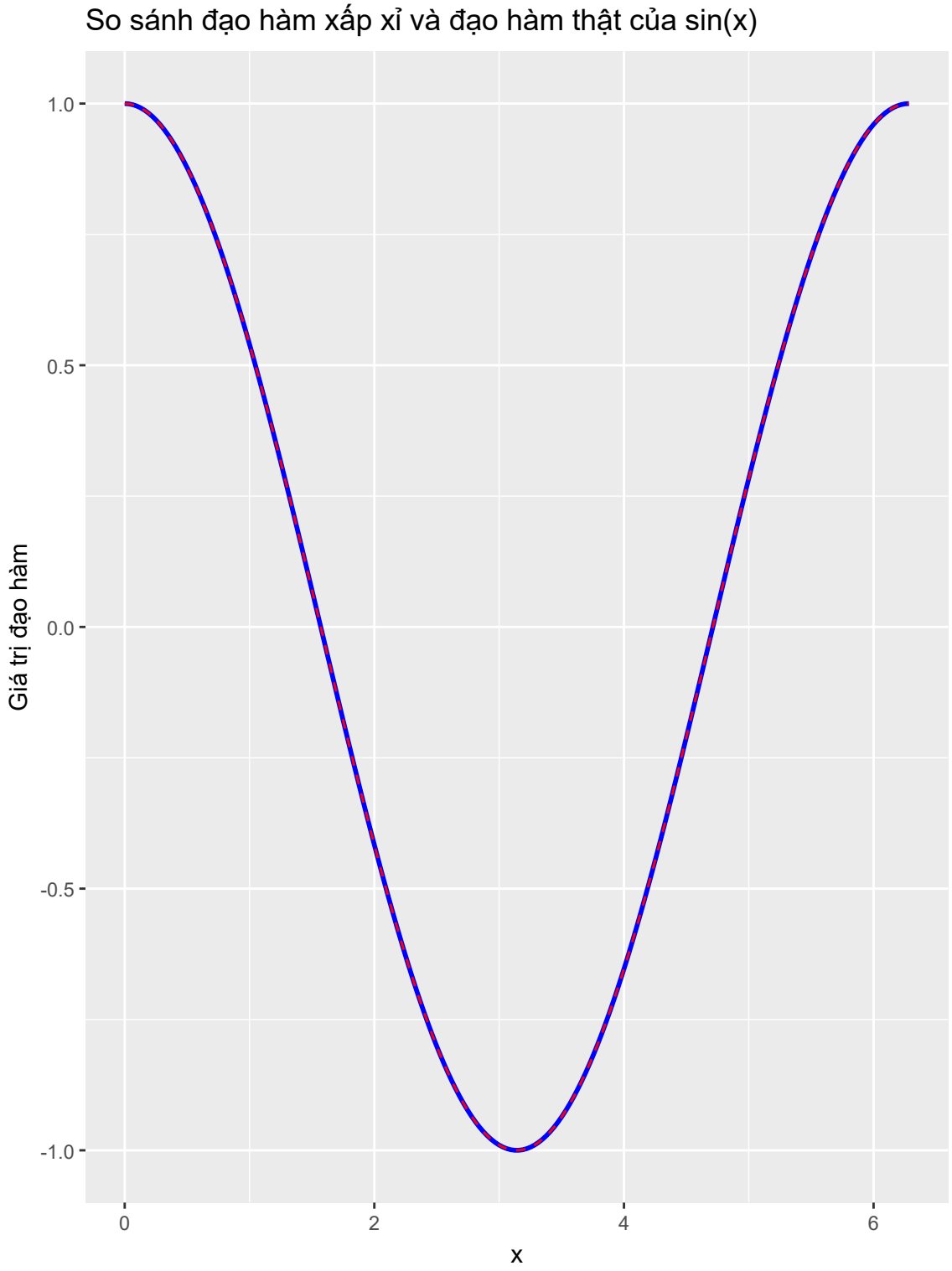
Dưới đây là hình vẽ tham khảo:



4. Đạo hàm số bằng sai phân hữu hạn

So sánh đạo hàm xấp xỉ và đạo hàm thật của hàm $\sin(x)$ trên $[0, 2\pi]$.

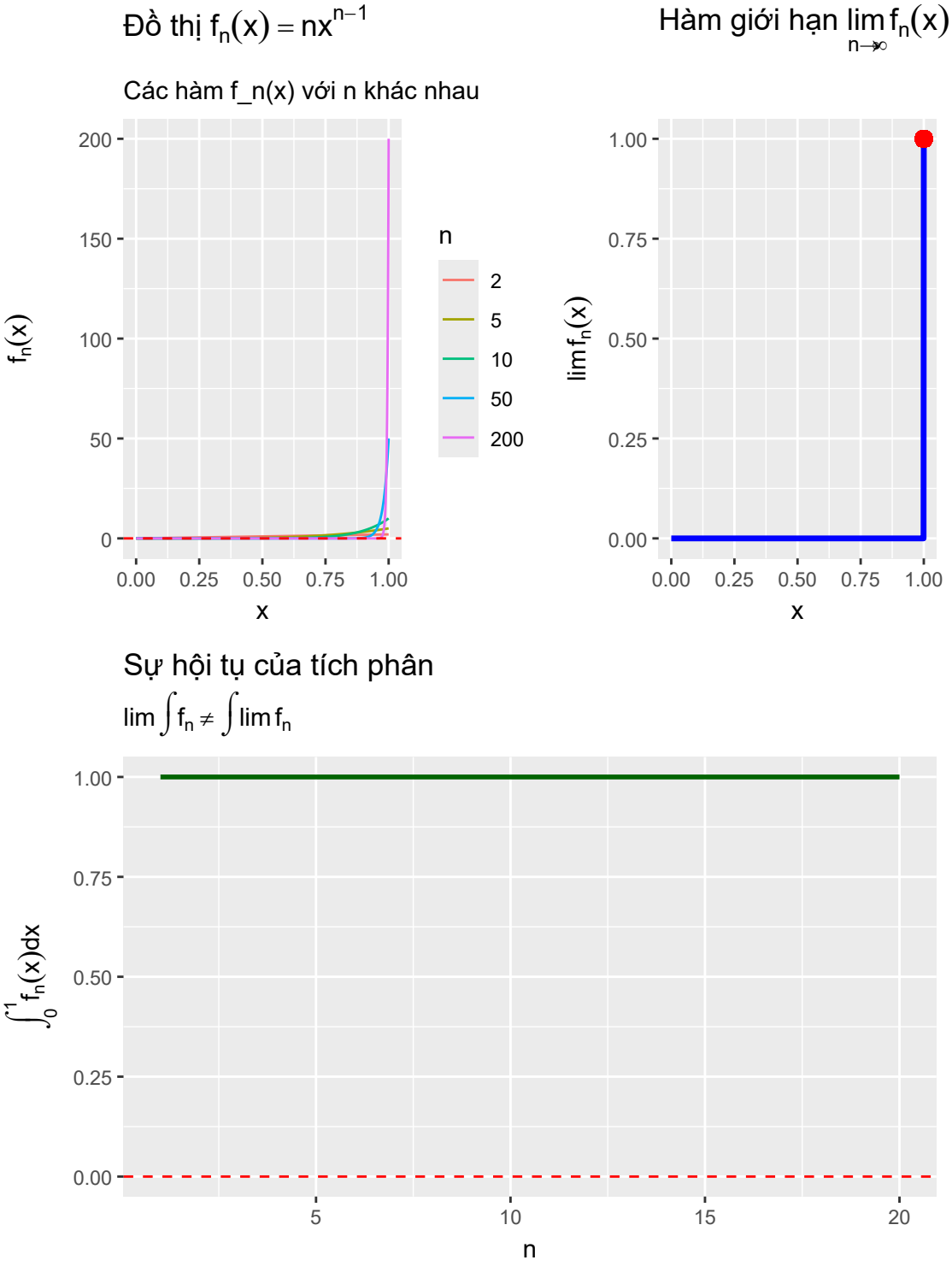
Dưới đây là hình vẽ tham khảo:



5. Hàm $f_n(x) = nx^{n-1}$

So sánh $\int_0^1 \lim f_n$ và $\lim \int f_n$. Từ đó kết luận xem hai đại lượng này bằng nhau hay không?

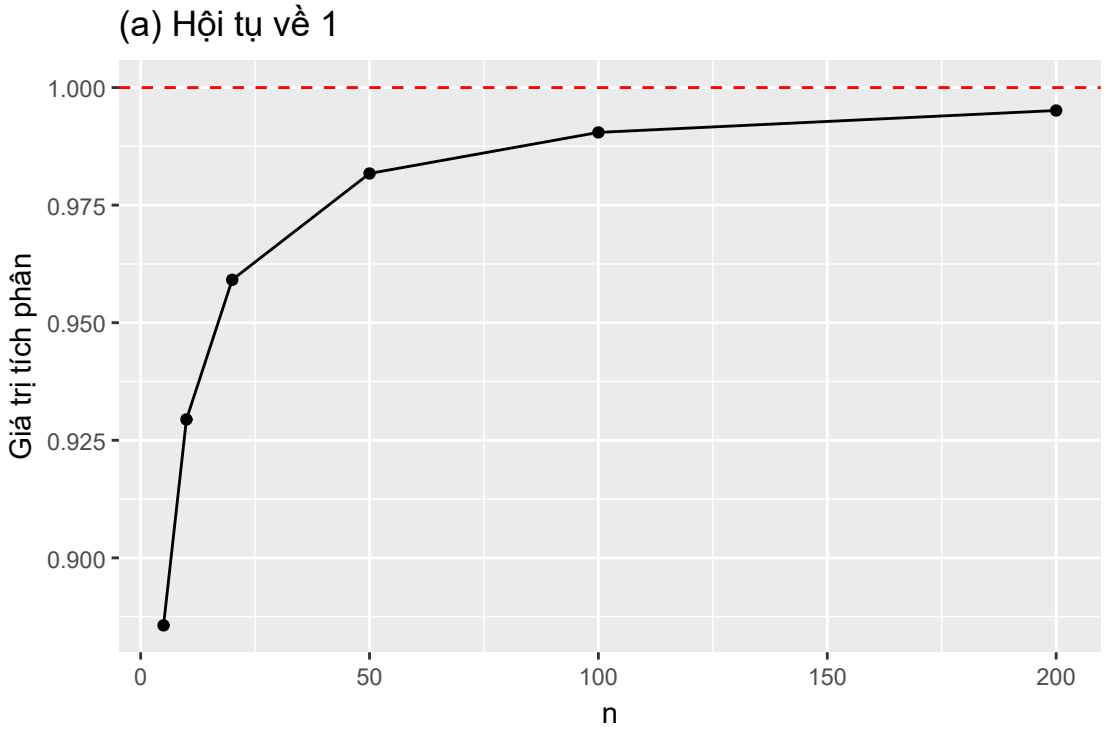
Dưới đây là hình vẽ tham khảo:

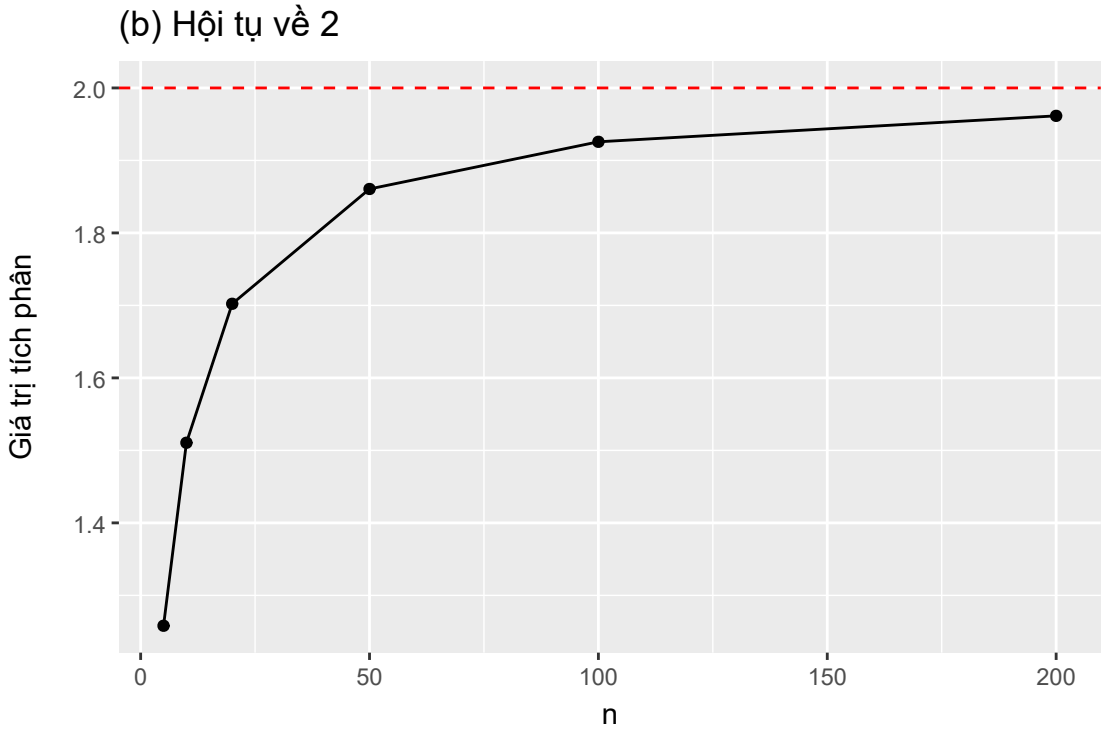


6. Tính hai giới hạn sau:

$$\text{(a)} \lim_{n\to\infty} \int_0^n \left(1+\frac{x}{n}\right)^n e^{-2x}dx, \quad \text{(b)} \lim_{n\to\infty} \int_0^n \left(1-\frac{x}{n}\right)^n e^{x/2}dx.$$

Dưới đây là hình vẽ tham khảo:

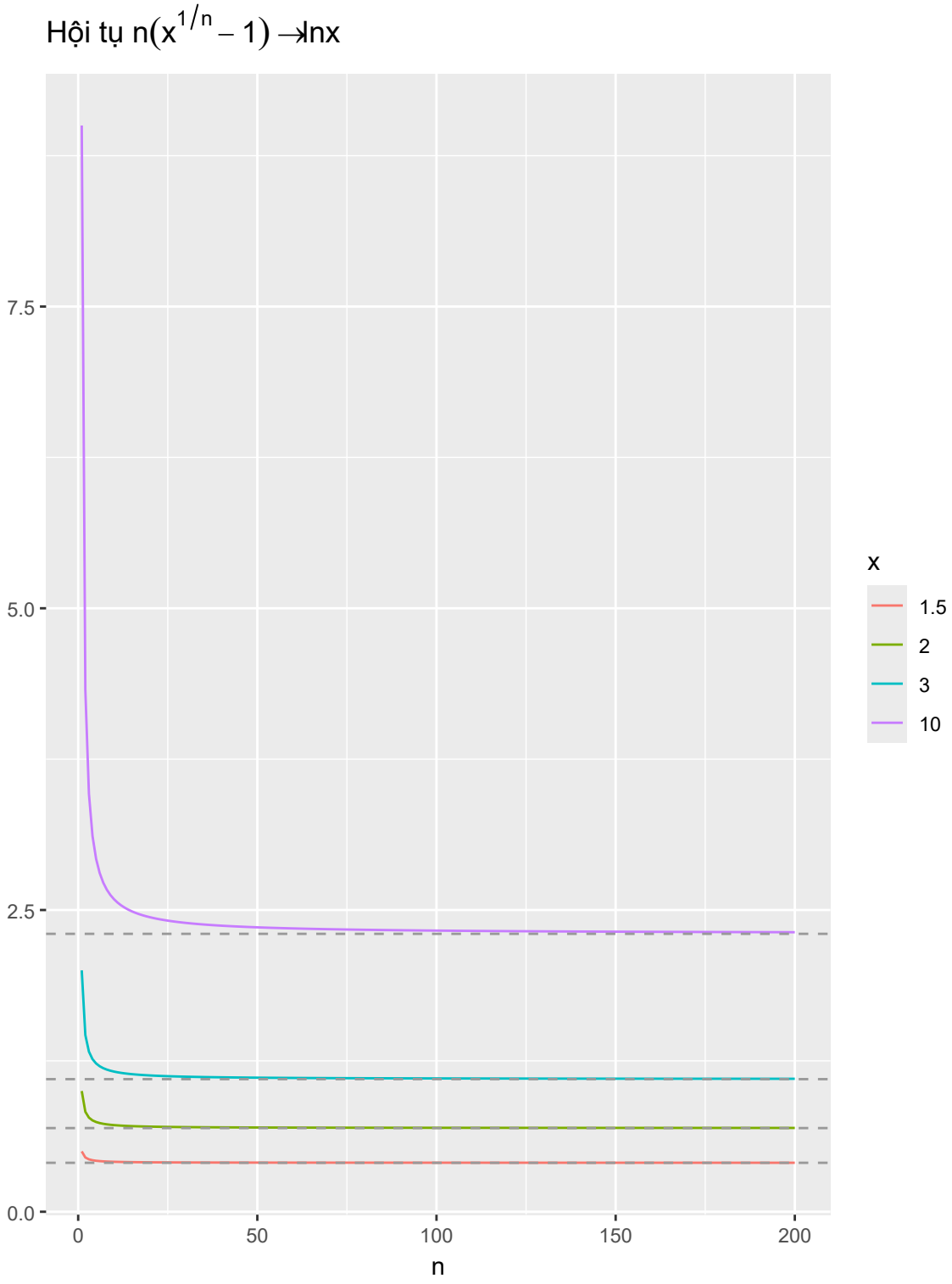




7. Giới hạn $n(x^{1/n} - 1) = \ln x$

Kiểm chứng bằng mô phỏng số đẳng thức trên với $x = \{1.5, 2, 3, 10\}$ và $n = \{50, 100, 150, 200\}$

Dưới đây là hình vẽ tham khảo:



8. Xét hàm số sau:

$$f_n(x) = \begin{cases} 2^n, & x \in (2^{-n}, 2^{1-n}), \\ 0, & \text{khác.} \end{cases}$$

Minh họa hội tụ điểm về 0 nhưng tích phân luôn bằng 1.

Gợi ý.

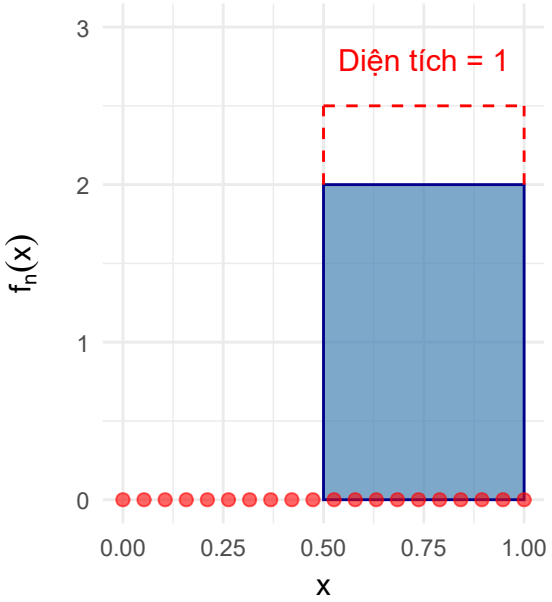
1. Hãy định nghĩa hàm `fn_rect(n)` để tính khoảng $(2^{(-n)}, 2^{(1-n)})$, cùng với diện tích. Mẫu

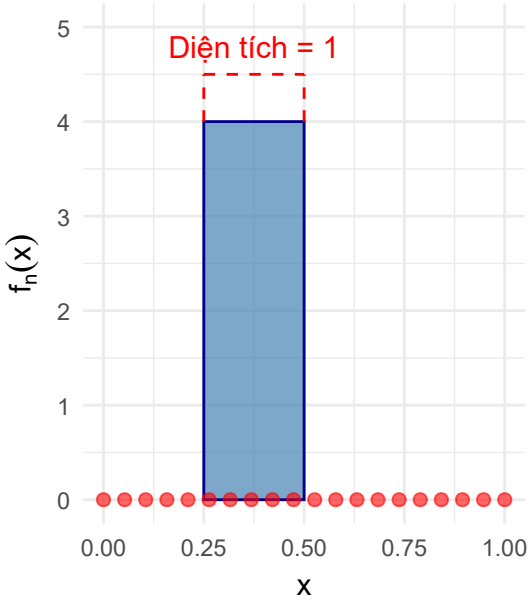
```
fn_rect <- function(n){
  a <-
  b <-
  h <-
  width <- b - a
  area <-

  point_data <- data.frame(
    x = seq(0, 1, length.out = 20),
    y = 0
  )

  p <- ggplot() +
    # Draw rectangle to display area

    # Draw points on x-axis to show convergence using point_data

    # Annotated lines for areas
    geom_segment(aes(x = a, xend = b, y = h + 0.5, yend = h + 0.5),
                 color = "red", linetype = "dashed") +
    geom_segment(aes(x = a, xend = a, y = h, yend = h + 0.5),
                 color = "red", linetype = "dashed") +
    geom_segment(aes(x = b, xend = b, y = h, yend = h + 0.5),
                 color = "red", linetype = "dashed") +
    coord_cartesian(xlim = c(0, 1), ylim = c(0, h + 1)) +
    labs(
      title = TeX(paste0("$f_", n, "(x)$ - Hàm chỉ thị trên $(2^{-", n, "},
                         2^{1-", n, "})$")),
      x = "x",
      y = TeX("$f_n(x)$"),
      subtitle = paste0("Diện tích = ", h, " × ", round(width, 4), " = ", area,
                        "Chiều rộng khoảng = ", round(width, 6))
    ) +
    theme_minimal() +
    annotate("text", x = (a + b)/2, y = h + 0.8,
             label = paste0("Diện tích = 1"), color = "red", size = 4)

  return(p)
```

Dưới đây là hình vẽ tham khảo:

$f_1(x)$ - Hàm chỉ thị trên $(2^{-1}, 2^{1-1})$ $f_2(x)$ - Hàm chỉ thị trên $(2^{-2}, 2^{1}$

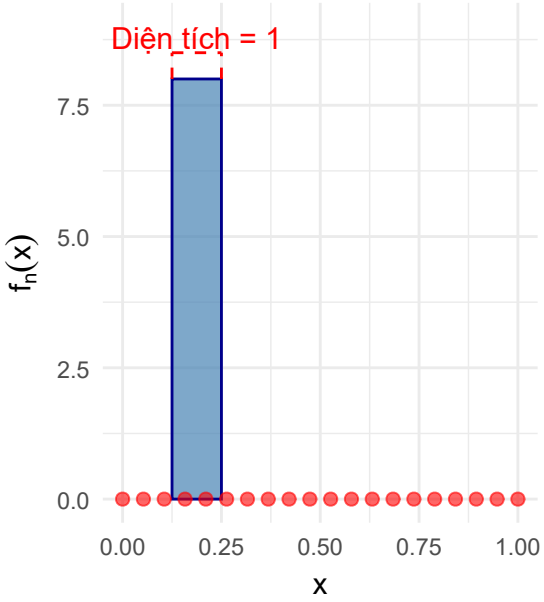
Diện tích = 2 × 0.5 = 1Chiều rộng khoảng = 0.5 Diện tích = 4 × 0.25 = 1Chiều rộng k

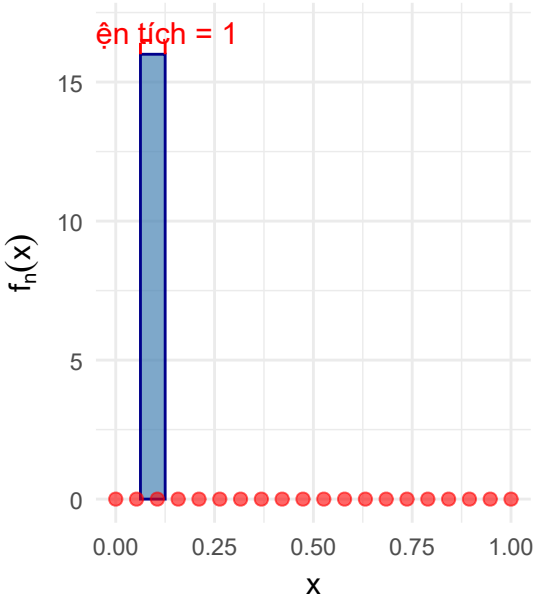


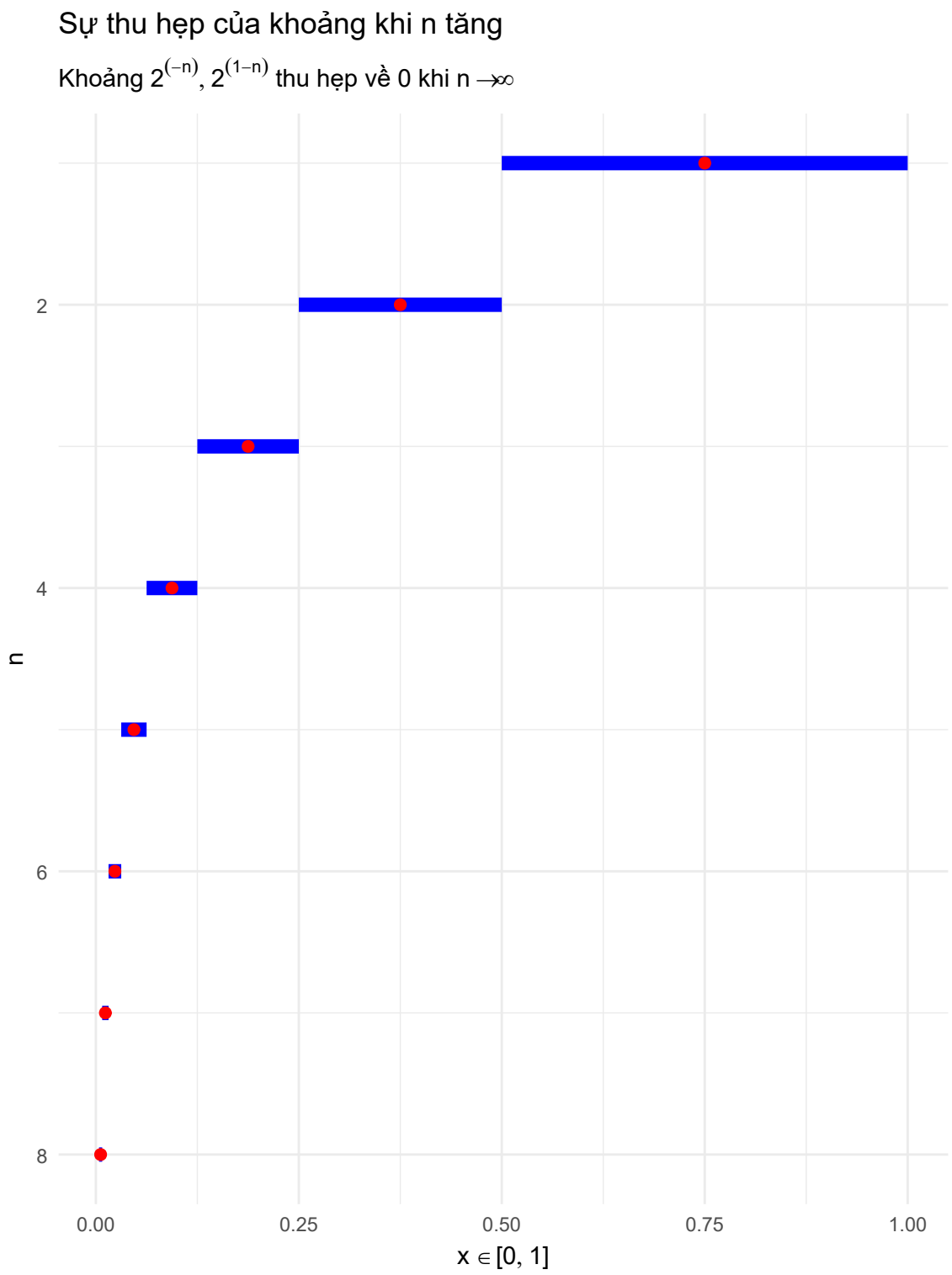


$f_3(x)$ - Hàm chỉ thị trên $(2^{-3}, 2^{1-3})$ $f_4(x)$ - Hàm chỉ thị trên $(2^{-4}, 2^{1}$

Diện tích = 8 × 0.125 = 1Chiều rộng khoảng = 0.125 Diện tích = 16 × 0.0625 = 1Chiều rộ







9. So sánh các đại lượng đo xu hướng trung tâm cho dữ liệu định tính và thứ bậc 9.1. Dữ liệu định danh Trong R không có hàm có sẵn để tính *mode* của một tập dữ liệu, hãy viết một hàm để tính nha.

Một cuộc khảo sát về màu sắc yêu thích được thực hiện trên 30 sinh viên:

```r
# Dữ liệu màu sắc yêu thích
colour <- c("Đỏ", "Xanh", "Xanh", "Vàng", "Đỏ", "Tím", "Xanh", "Đỏ", "Đỏ", "Cam",
            "Vàng", "Xanh", "Đỏ", "Tím", "Xanh", "Đỏ", "Xanh", "Xanh", "Đỏ", "Cam",
            "Đỏ", "Xanh", "Vàng", "Đỏ", "Xanh", "Tím", "Đỏ", "Xanh", "Đỏ", "Xanh")

print(table(colour))
```

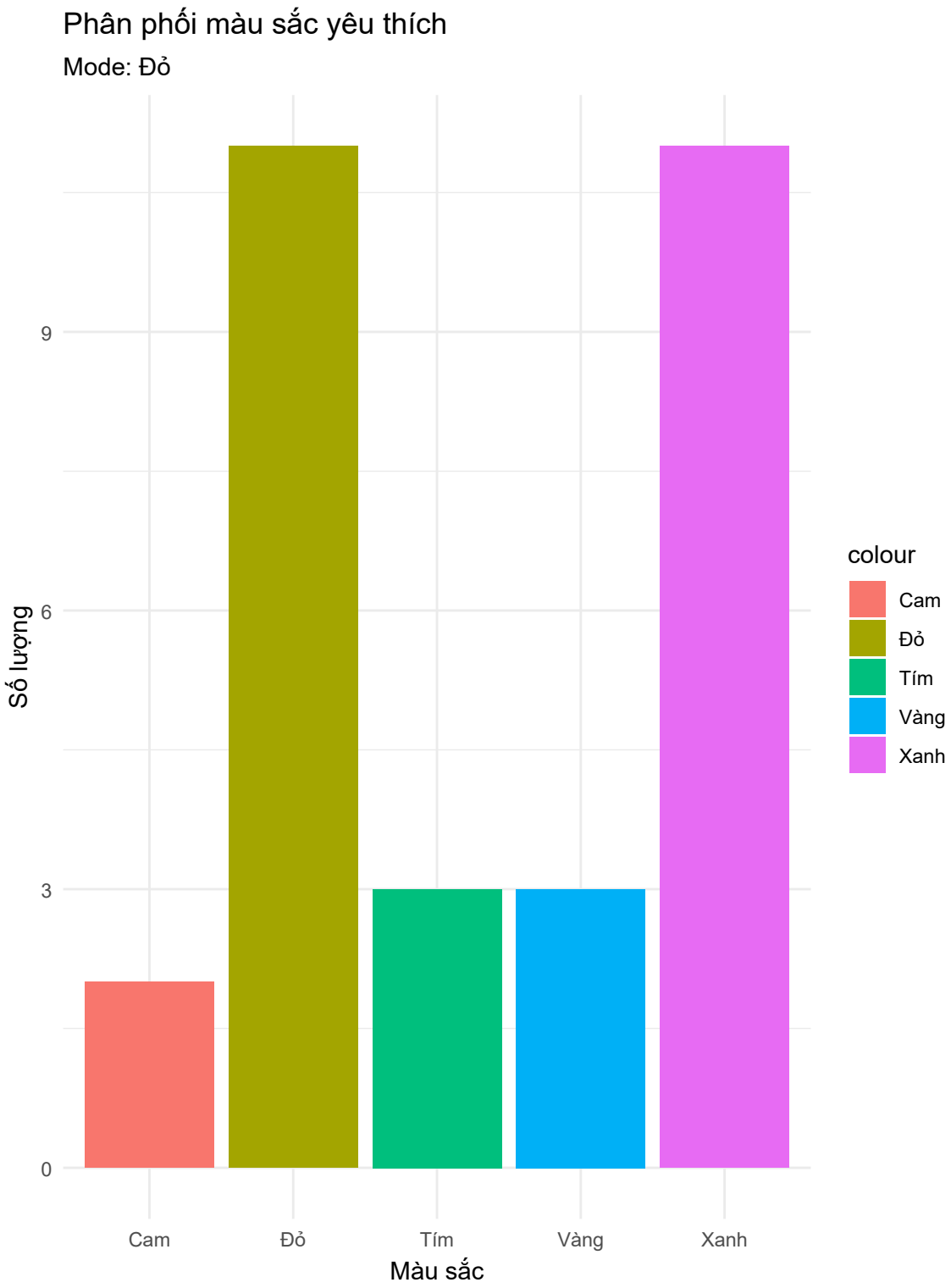
```
## colour
##  Cam   Đỏ  Tím Vàng Xanh
##    2   11    3    3   11
```

a) Tính trung bình cho dữ liệu này có ý nghĩa không? Tại sao?

b) Hãy tính mode cho dữ liệu màu sắc. Màu nào là phổ biến nhất?

c) Trình bày kết quả bằng biểu đồ thích hợp và giải thích ý nghĩa của mode trong trường hợp này.

Dưới đây là hình vẽ tham khảo:



9.2. Dữ liệu thứ bậc

Khảo sát mức độ hài lòng của khách hàng về một sản phẩm (thang điểm 5 mức):

1: Rất không hài lòng

2: Không hài lòng

3: Bình thường

4: Hài lòng

5: Rất hài lòng

```
satisfaction_lvl <- c(4, 5, 3, 4, 5, 2, 4, 5, 4, 3,
                      5, 4, 4, 3, 5, 4, 4, 5, 3, 4,
                      5, 4, 2, 4, 5, 4, 3, 4, 5, 4)

satisfaction_lvl <- factor(satisfaction_lvl,
                           levels = 1:5,
                           labels = c("Rất không hài lòng", "Không hài lòng",
                                      "Bình thường", "Hài lòng", "Rất hài lòng"),
                           ordered = TRUE)

print(table(satisfaction_lvl))
```

```
## satisfaction_lvl
## Rất không hài lòng     Không hài lòng        Bình thường            Hài lòng
##                  0                  2                  5                  14
##       Rất hài lòng
##                  9
```

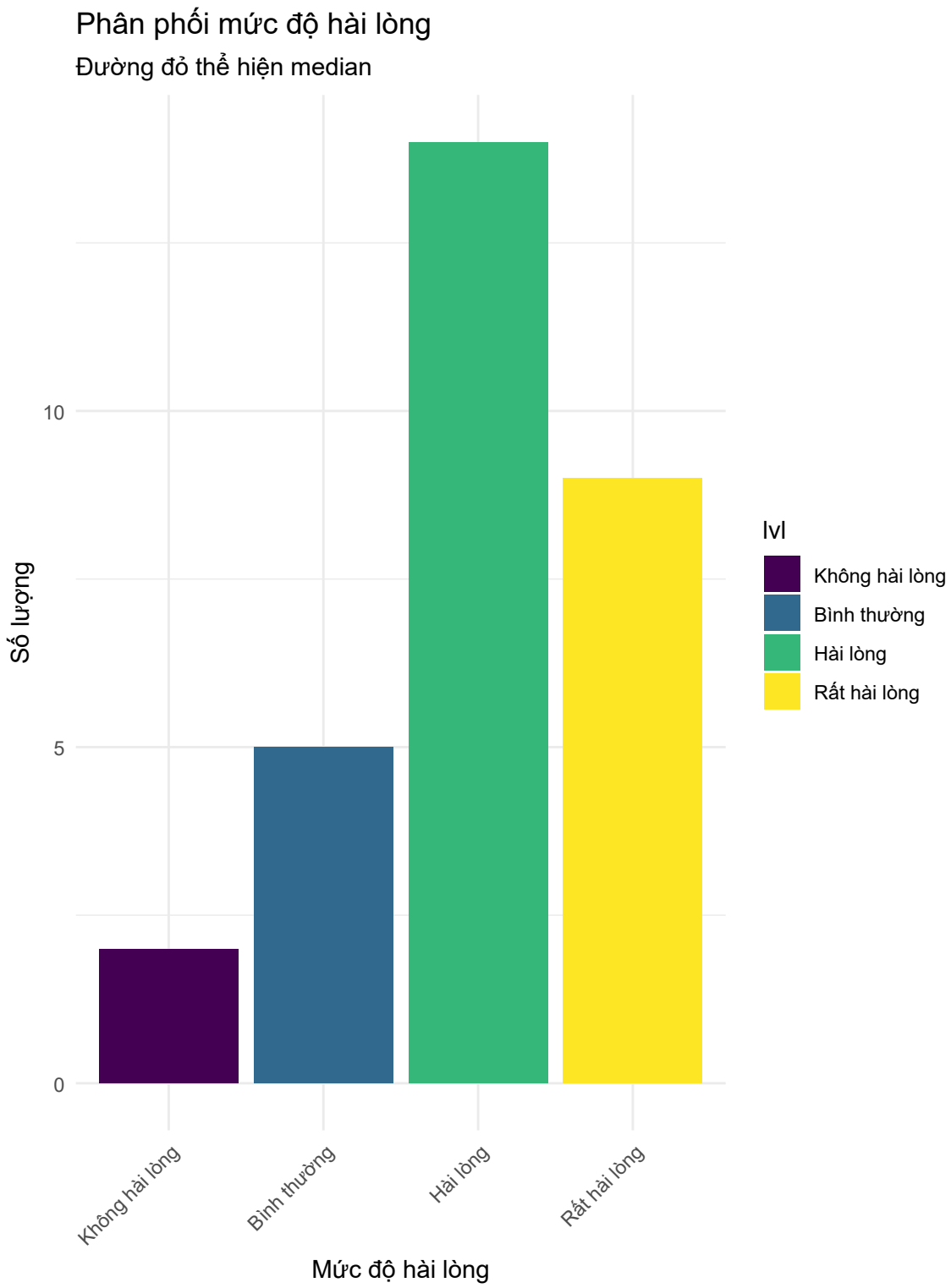
a) Tính trung bình số học cho dữ liệu này. Kết quả có ý nghĩa không?

b) Tính trung vị cho dữ liệu thứ bậc. Giải thích ý nghĩa của kết quả.

c) Tính mode cho dữ liệu này. So sánh median và mode, chỉ ra giá trị nào phản ánh tốt hơn "xu hướng trung tâm" của dữ liệu.

Dưới đây là hình vẽ tham khảo:

```
## Median của 1 3 5 7 9 là: 5
## Median của 1 3 5 7 là: 4
## Kiểm tra với hàm có sẵn: 5 và 4
## Kết quả:
## - Mean (dạng số): 4
## - Median: Hài lòng
## - Mode: Hài lòng
```



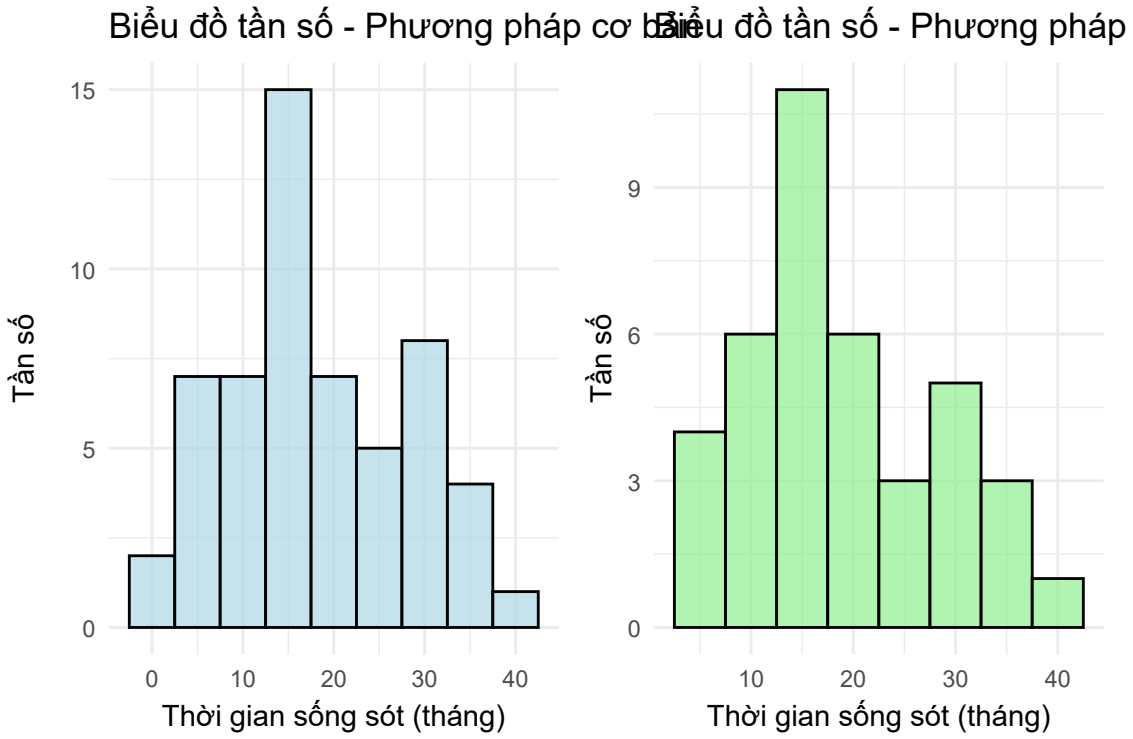
10. Dữ liệu thời gian sống sót của bệnh nhân (tháng):

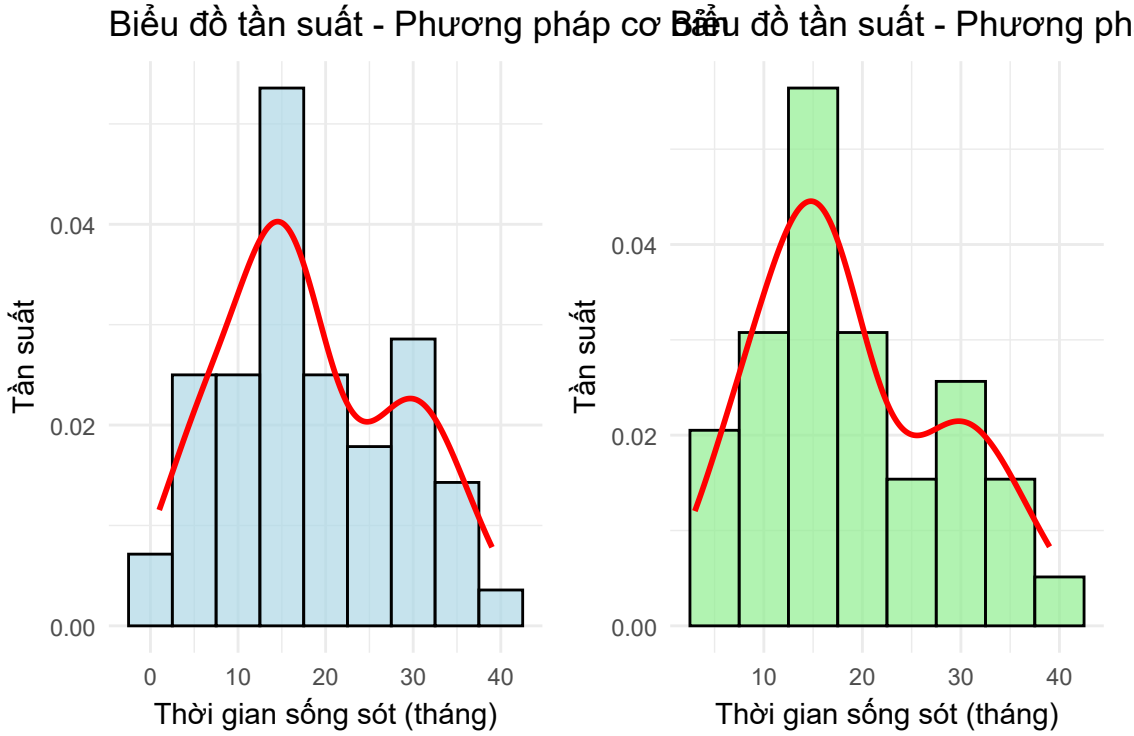
```
Phương pháp cơ bản: 4, 15, 24, 10, 1, 27, 31, 5, 20, 29, 15, 7, 32, 36, 14, 2, 16, 32, 7
Phương pháp mới: 15, 7, 32, 36, 17, 15, 19, 35, 10, 16, 39, 29, 6, 12, 18, 14, 15, 18, 2
```

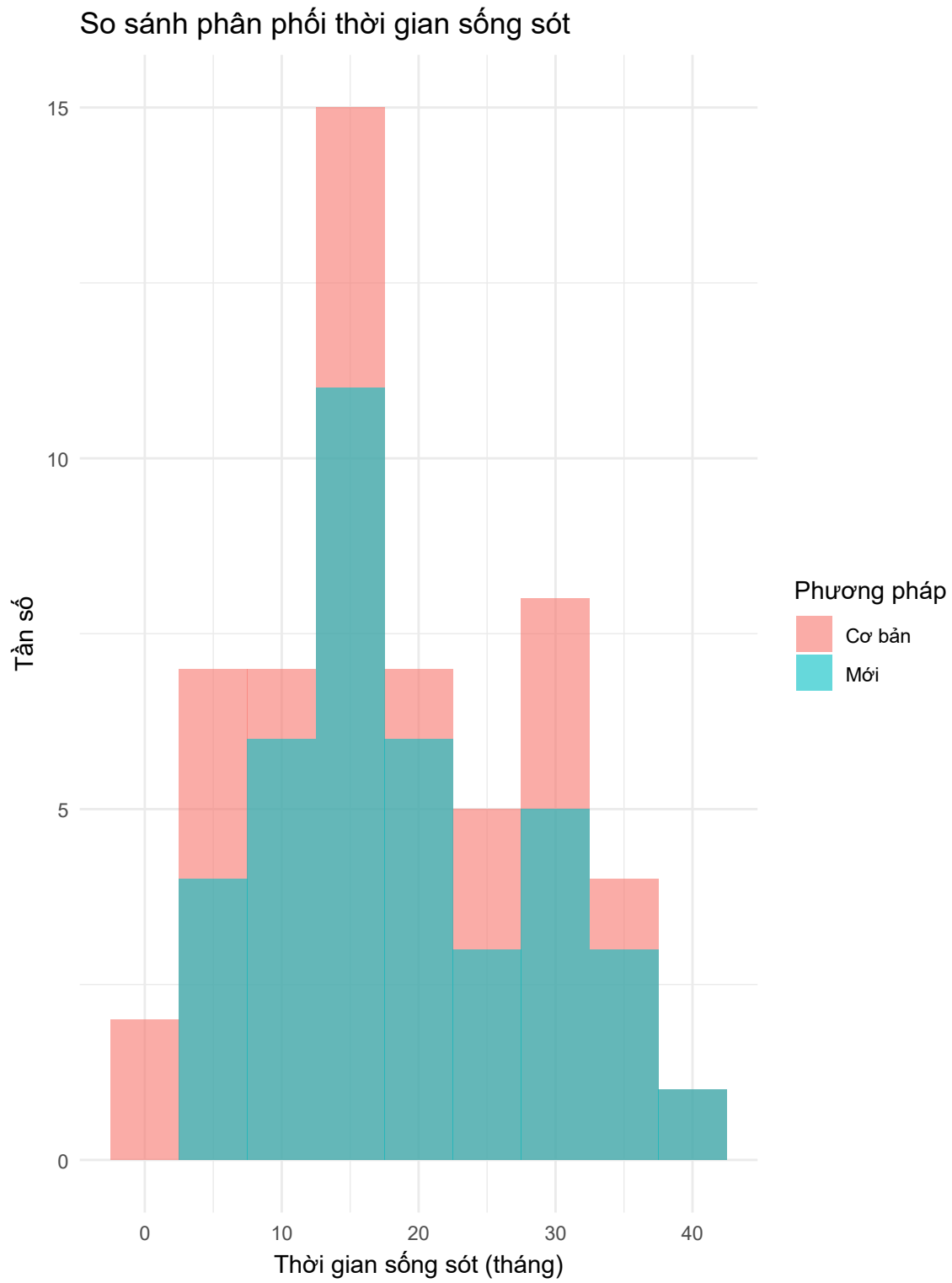
a) Dùng hàm `c()`, nhập hai vector `coBan` và `Moi` với dữ liệu đã cho. Sau đó vẽ đồ thị histogram biểu thị tần số cũng như tần suất của hai phương pháp trên. Theo bạn, liệu phương pháp mới sẽ cho thời gian sống sót của bệnh nhân sau khi điều trị lớn hơn hay không?

b) Kết hợp hai dữ liệu (vector) này thành một, gọi là vector `all` và vẽ lại biểu đồ tần số và tần suất. Theo bạn, liệu phương pháp mới sẽ cho thời gian sống sót của bệnh nhân sau khi điều trị lớn hơn hay không? Giải thích.

Dưới đây là hình vẽ tham khảo:


Biểu đồ tần số - Phương pháp cơ bản
Tần số
Thời gian sống sót (tháng)
Biểu đồ tần số - Phương pháp
Tần số
Thời gian sống sót (tháng)


Biểu đồ tần suất - Phương pháp cơ bản
Tần suất
Thời gian sống sót (tháng)
Biểu đồ tần suất - Phương ph
Tần suất
Thời gian sống sót (tháng)



11. Dùng hàm `c()` nhập số liệu cho vector `data` cho bởi bảng sau:

55	85	90	50	110	115	75	85	8	23
70	65	50	60	90	90	55	70	5	31

a) Không dùng hàm `mean()` và `median()`, hãy xây dựng một hàm để tính các thống kê mô tả mà ta hay sử dụng. Sau đó dùng hàm đó để tính trung bình và trung vị của `data`.

b) Lập bảng tần số, `freq_data`, của `data` và tính mode của `data`.

c) Thay số liệu 110 và 115 bằng 345 và 467. Tính lại trung bình, trung vị và mode. Nhận xét ảnh hưởng của các số liệu có giá trị bất thường tới các thông số đo xu hướng trung tâm