



# MỘT SỐ HƯỚNG DẪN MỞ ĐẦU VỀ R/RSTUDIO

**KHÓA BỒI DƯỠNG GIẢNG VIÊN KHU VỰC MIỀN BẮC 2023**

**“THỐNG KÊ HIỆN ĐẠI VỚI PHẦN MỀM THỐNG KÊ R”  
(TS. Nguyễn Thị Nhung Đại học Thăng Long – biên soạn  
TS. Ngô Thị Thanh Nga – trình bày)**

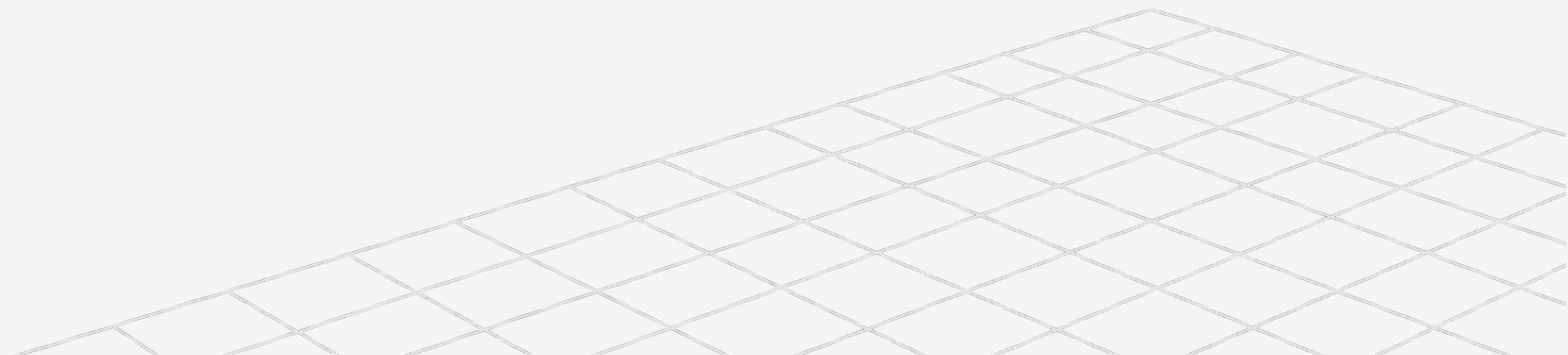
**Hà Nội, tháng 12/2023**



# NỘI DUNG

---

- 1 Thống kê mô tả trên R
- 2 Khoảng tin cậy và kiểm định trên R
- 3 Tương quan và hồi quy trên R



# Thống kê mô tả trên R

# Bảng tần số

---

x là tập dữ liệu	
Hàm	Công dụng
<code>table(x)</code>	Tính tần số của các phần tử trong x
<code>prop.table(table(x))</code>	Tính tần suất của các phần tử trong x
<code>cumsum(table(x))</code>	Tính tần số tích lũy của các phần tử trong x
<code>cumsum(prop.table(table(x)))</code>	Tính tần suất tích lũy của các phần tử trong x

# Phân tổ dữ liệu

Trong R, Hàm `cut` được sử dụng để phân tổ dữ liệu:

● Rectangular Snip

**`cut(x, breaks, labels, right, include.lowest, dig.lab)`**

trong đó

<code>x</code>	véc tơ dữ liệu dạng số cần được phân tổ.
<code>breaks</code>	véc tơ số (ít nhất hai tọa độ) gồm các điểm chia hoặc là một số nguyên dương (lớn hơn hoặc bằng 2) chỉ số tổ.
<code>labels</code>	nhãn của các tổ, theo mặc định <code>labels = NULL</code> , các nhãn được xây dựng dưới dạng nửa khoảng $(a, b]$ .
<code>right</code>	dạng logic, nếu <code>right = TRUE</code> tổ có dạng $(a, b]$ , nếu <code>right = FALSE</code> tổ có dạng $[a, b)$ , mặc định <code>right = TRUE</code> .
<code>include.lowest</code>	dạng logic, nếu <code>include.lowest = TRUE</code> thì tổ đầu chứa giá trị nhỏ nhất của các điểm chia (khi <code>right = TRUE</code> ) hoặc tổ cuối chứa giá trị lớn nhất của các điểm chia (khi <code>right = FALSE</code> ), mặc định <code>include.lowest = FALSE</code> .
<code>dig.lab</code>	số nguyên dương chỉ số chữ số trong điểm chia (trong trường hợp không gán nhãn cho các khoảng chia), mặc định <code>dig.lab = 3</code> .

# Hàm vẽ hình bậc cao

## Hàm vẽ hình bậc cao

Dưới đây là một số hàm vẽ hình bậc cao thường gặp trong R:

<code>plot(x)</code>	tạo ra các điểm có tọa độ $(i, x_i), i = \overline{1, n}$ với $x = (x_1, x_2, \dots, x_n)$
<code>plot(x, y)</code>	tạo ra các điểm có tọa độ $(x_i, y_i), i = \overline{1, n}$ , với $x = (x_1, x_2, \dots, x_n), y = (y_1, y_2, \dots, y_n)$
<code>hist(x)</code>	vẽ biểu đồ phân phối tần số của véc tơ dữ liệu $x$
<code>barplot(x)</code>	vẽ biểu đồ thanh với cao độ của các phần tử cho trong véc tơ $x$
<code>pie(x)</code>	vẽ biểu đồ hình tròn với cao độ của các phần tử cho trong véc tơ $x$
<code>boxplot(x)</code>	vẽ biểu đồ hộp và râu của véc tơ dữ liệu $x$
<code>symbols(x, y)</code>	vẽ tại tọa độ $x, y$ một số hình như: hình tròn, hình chữ nhật, hình vuông, hình sao, hình nhiệt kế hoặc dạng biểu đồ hộp và râu

# Tham số trong hàm vẽ hình

Cách sử dụng chi tiết những hàm này để vẽ hình, ta có thể đọc trong phần trợ giúp (help) tương ứng với mỗi hàm. Từng hàm có những tham số riêng, tuy nhiên, có một số tham số sau (với giá trị mặc định cho tương ứng) được dùng chung cho nhiều hàm:

`add=FALSE`

nếu `add=TRUE` thì hình đang vẽ sẽ chèn thêm vào hình vẽ trước (nếu có)

`axes=TRUE`

nếu `axes=FALSE` thì không vẽ các trục và hộp bao quanh

`type="p"`

miêu tả kiểu vẽ: "p" (points) dạng điểm; "l" (lines) dạng đoạn thẳng; "b" (both points and lines) dạng các điểm được nối bởi đoạn thẳng; "o" (overstruck) dạng các điểm được nối bởi đoạn thẳng nhưng đoạn thẳng đi qua các điểm; "h" (histogram) dạng thẳng đứng; "s" (stair steps) dạng bậc thang; "n" (no plot) không có kiểu gì cả

giới hạn của trục nằm ngang và trục thẳng đứng

tên của trục nằm ngang và trục thẳng đứng (kiểu kí tự)

tiêu đề của hình vẽ (kiểu kí tự)

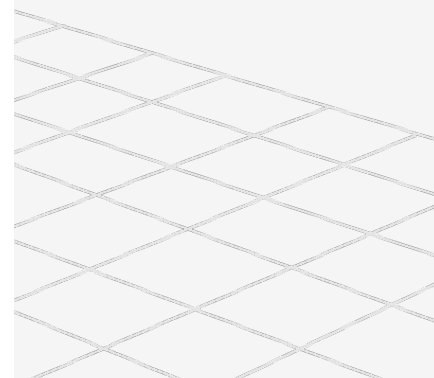
tiêu đề phụ của hình vẽ (kiểu kí tự)

`xlim, ylim`

`xlab, ylab`

`main`

`sub`



# Tham số trong hàm vẽ hình

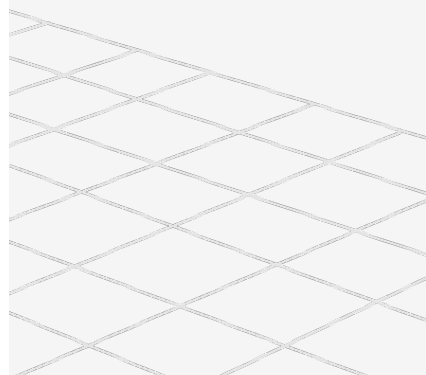
## Các hàm vẽ hình bậc thấp

<code>points(x)</code>	thêm các điểm $(i, x_i), i = \overline{1, n}$ , vào hình vẽ với $x = (x_1, x_2, \dots, x_n)$
<code>points(x, y)</code>	thêm các điểm tọa độ $(x_i, y_i), i = \overline{1, n}$ vào hình vẽ với $x = (x_1, x_2, \dots, x_n), y = (y_1, y_2, \dots, y_n)$
<code>lines(x, y)</code>	thêm các đoạn thẳng nối các điểm $(x_i, y_i)$ với $(x_{i+1}, y_{i+1}), i = \overline{1, n-1}$ vào hình vẽ
<code>text(x, y, labels)</code>	viết đoạn văn bản có nội dung ở <code>labels</code> tại tọa độ $(x, y)$ ; kiểu sử dụng điển hình là <code>plot(x, y, type="n"), text(x, y, name)</code>
<code>mtext(text, side</code>	viết đoạn văn có nội dung ở <code>text</code> bên lề của cạnh thứ $k, k = 1, 2, 3, 4$ của hình
<code>segments(x<sub>0</sub>, y<sub>0</sub>, x<sub>1</sub>, y<sub>1</sub>)</code>	thêm đoạn thẳng nối điểm $(x_0, y_0)$ và $(x_1, y_1)$ vào hình
<code>arrows(x<sub>0</sub>, y<sub>0</sub>, x<sub>1</sub>, y<sub>1</sub>, code)</code>	thêm đoạn thẳng nối điểm $(x_0, y_0)$ và $(x_1, y_1)$ cùng với mũi tên ở $(x_1, y_1)$ nếu <code>code=2</code> và ở $(x_0, y_0)$ nếu <code>code=1</code> và ở cả hai nếu <code>code=3</code>

































# Tham số trong hàm vẽ hình

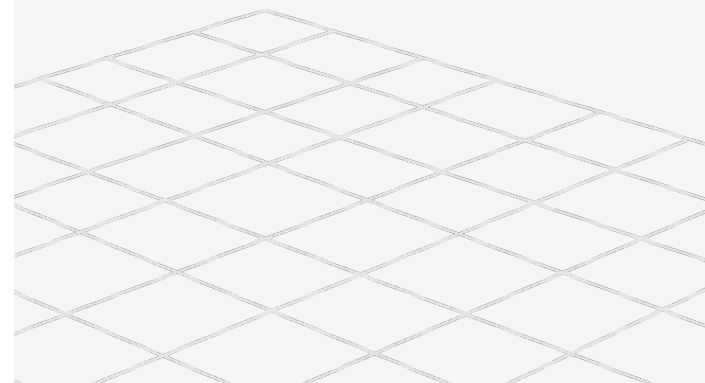
<code>abline(a, b)</code>	thêm đường thẳng với tung độ góc là $a$ và độ dốc là $b$ vào hình
<code>abline(h = y_0)</code>	vẽ thêm đường thẳng $y = y_0$ song song với trục nằm ngang vào hình
<code>abline(v = x_0)</code>	vẽ thêm đường thẳng $x = x_0$ song song với trục thẳng đứng vào hình
<code>abline(lm(y ~ x))</code>	vẽ thêm đường thẳng hồi qui tuyến tính mẫu vào hình
<code>rect(x1, y1, x2, y2)</code>	vẽ thêm hình chữ nhật mà trái, phải, dưới, trên tương ứng giới hạn bởi $x_1, x_2, y_1, y_2$
<code>polygon(x, y)</code>	vẽ thêm một đa giác nối các điểm có tọa độ ở $x, y$
<code>legend(x, y, legend)</code>	điền lời chú thích tại điểm $(x, y)$ với nội dung ở <code>legend</code>
<code>title</code>	điền tiêu đề và tiêu đề phụ của hình vẽ
<code>box</code>	vẽ thêm khung bao quanh hình vẽ
<code>axis(side, vect)</code>	vẽ thêm trục vào hình vẽ, trục dưới nếu <code>side=1</code> , trục trái nếu <code>side=2</code> , trục trên nếu <code>side=3</code> , trục phải nếu <code>side=4</code> , nội dung điền trên mỗi trục qua <code>vect</code>
<code>rug(x)</code>	vẽ tại dữ liệu $x$ trên trục nằm ngang những đoạn ngắn thẳng đứng



# Tham số trong hàm vẽ hình: tham số `pch`

1	2	3	4	5	6	7	8	9	10
									
11	12	13	14	15	16	17	18	19	20
									
21	22	23	24	25	"*"	"?"	"."	"X"	"a"
									

Hình 4.1: Một số kí hiệu diễn hình của tham số `pch`



# Biểu đồ phân phối tần số

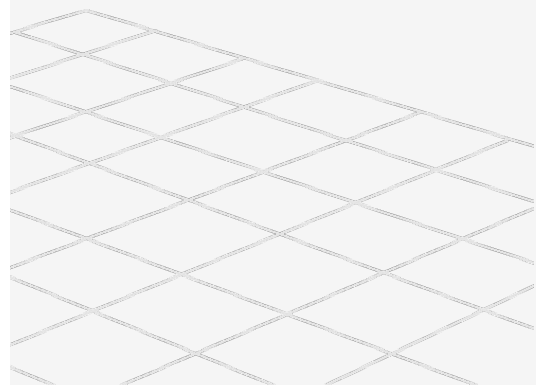
## Biểu đồ phân phối tần số

Trong *R*, để minh họa hình dáng của phân phối của tập dữ liệu bằng biểu đồ phân phối tần số, ta dùng hàm `hist`:

**`hist(x, breaks, freq, include.lowest, right, col, border,  
main, xlim, ylim, xlab, ylab, labels)`**

trong đó,

<code>x</code>	véc tơ dữ liệu dạng số cần vẽ biểu đồ
<code>freq</code>	dạng logic, nếu <code>freq = TRUE</code> các cột của biểu đồ mô tả tần số, nếu <code>freq = FALSE</code> các cột của biểu đồ mô tả tần suất mật độ
<code>breaks</code>	véc tơ số (ít nhất hai tọa độ) gồm các điểm chia giữa các cột hoặc là một số nguyên dương (lớn hơn hoặc bằng 2) chỉ số cột của biểu đồ
<code>right</code>	dạng logic, nếu <code>right = TRUE</code> thì các cột lấy phần tử trong khoảng dạng $(a, b]$ , nếu <code>right = FALSE</code> thì trong khoảng dạng $[a, b)$ , mặc định <code>right = TRUE</code>
<code>include.lowest</code>	dạng logic, nếu <code>include.lowest = TRUE</code> thì cột đầu tiên chứa giá trị nhỏ nhất của các điểm chia (khi <code>right = TRUE</code> ) hoặc cột cuối cùng chứa giá trị lớn nhất của các điểm chia (khi <code>right = FALSE</code> ), mặc định <code>include.lowest = TRUE</code>



# Biểu đồ phân phối tần số

---

**hist(x, breaks, freq, include.lowest, right, col, border,  
main, xlim, ylim, xlab, ylab, labels)**

trong đó,

col

màu của các cột

border

màu đường biên của các cột

main, xlab, ylab

tên của biểu đồ, tên trục  $x, y$

xlim, ylim

giới hạn trên các trục

labels

dạng logic hoặc dạng kí tự điền tên trên đỉnh mỗi cột

# Biểu đồ thân và lá

---

## Biểu đồ thân và lá

Để vẽ biểu đồ thân và lá trong *R*, ta sử dụng hàm `stem`:

**`stem(x, scale, width)`**

trong đó,

<code>x</code>	véc tơ dữ liệu dạng số.
<code>scale</code>	tham số điều chỉnh chiều dài của biểu đồ, mặc định là 1.
<code>width</code>	tham số điều chỉnh chiều dài của biểu đồ theo mong muốn, mặc định là 80.

# Biểu đồ thanh

## Biểu đồ thanh

Trong *R*, để vẽ biểu đồ thanh, ta dùng hàm `barplot`:

**`barplot(height, names.arg, legend.text, beside, horiz,  
col, border, main, sub, xlab, ylab, xlim, ylim)`**

trong đó,

<code>height</code>	véc tơ hoặc ma trận dữ liệu dùng để vẽ biểu đồ.
<code>names.arg</code>	tên viết dưới mỗi thanh hoặc nhóm các thanh trong biểu đồ.
<code>legend.text</code>	véc tơ gồm các kí tự hoặc dạng logic dùng để ghi chú thích trong biểu đồ.
<code>beside</code>	dạng logic, nếu <code>beside = FALSE</code> thì các thanh của biểu đồ được vẽ chồng lên nhau, nếu <code>beside = TRUE</code> thì các thanh được vẽ cạnh nhau.
<code>horiz</code>	dạng logic, nếu <code>horiz = FALSE</code> thì các thanh được vẽ vuông góc với trục nằm ngang với thanh đầu tiên nằm ở bên trái, nếu <code>horiz = TRUE</code> thì các thanh được vẽ song song với trục nằm ngang với thanh đầu tiên nằm ở dưới cùng.
<code>col</code>	màu của các cột.
<code>border</code>	màu đường biên của các cột.
<code>main, sub</code>	tên của biểu đồ.
<code>xlab, ylab</code>	tên trục <i>x</i> , <i>y</i> .
<code>xlim, ylim</code>	giới hạn trên các trục.

# Biểu đồ tròn

---

## Biểu đồ hình tròn (hình bánh)

Trong R để vẽ biểu đồ hình tròn ta dùng hàm `pie`:

**`pie(x, labels, col, border, lty, main, sub)`**

<code>x</code>	véc tơ dạng số thể hiện giá trị của mỗi hình quạt trong biểu đồ.
<code>labels</code>	tên của những hình quạt trong biểu đồ.
<code>col</code>	màu của các hình quạt.
<code>border</code>	màu của đường ranh giới giữa các hình quạt.
<code>lty</code>	kiểu nét vẽ của đường ranh giới giữa các hình quạt.
<code>main, sub</code>	tiêu đề và tiêu đề phụ của biểu đồ.



# Biểu đồ tán xạ

## 4.2.5 Biểu đồ tán xạ (phân tán)

Biểu đồ tán xạ hay biểu đồ phân tán là một trong những công cụ đồ thị quan trọng cho ta cái nhìn tương đối về dạng liên hệ của hai biến định lượng. Trong R biểu đồ tán xạ được vẽ qua hàm `plot` với tham số `type="p"`. Một số tham số cần thiết của hàm `plot` trong trường hợp này được cho như sau:

```
plot(x, y, type = "p", main = "",  
      xlab = "", ylab = "", bty = "l")
```

trong đó

<code>x, y</code>	tương ứng là véc tơ tọa độ của biến định lượng thứ nhất, thứ hai;
<code>main, xlab, ylab</code>	tên của biểu đồ, tên biến thứ nhất, biến thứ hai;
<code>type = "p"</code>	kiểu vẽ gồm các điểm trên mặt phẳng;
<code>bty = "l"</code>	kiểu bao quanh biểu đồ giống hai trục tọa độ đề các <i>Oxy</i> .



# Biểu đồ hộp và râu

---

## 4.3.2 Biểu đồ hộp và râu

Biểu đồ hộp và râu giúp ta minh họa các tham số thống kê như trung vị, tứ phân vị và các giá trị ngoại biên trên cùng một hình vẽ. Để vẽ biểu đồ hộp và râu trong R, ta dùng hàm `boxplot`

**`boxplot(x, names, border, col, horizontal)`**

trong đó,

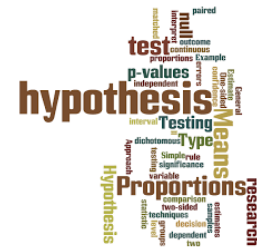
<code>x</code>	véc tơ dữ liệu dạng số.
<code>names</code>	tham số ghi chú thích tên dưới mỗi biểu đồ.
<code>border</code>	màu của râu, đường biên của hộp và giá trị ngoại biên.
<code>col</code>	màu của hộp.
<code>horizontal</code>	tham số logic chỉ cách vẽ biểu đồ, nếu <code>horizontal=FALSE</code> thì biểu đồ được vẽ đứng, nếu <code>horizontal=TRUE</code> thì biểu đồ được vẽ ngang.

# Đại lượng thống kê mô tả

R hỗ trợ nhiều hàm để tính các tham số thống kê mô tả, bảng sau cho ta một số hàm phổ biến:

Hàm	Công dụng
<code>mean(x)</code>	tính trung bình cộng của các giá trị cho trong véc tơ <code>x</code>
<code>median(x)</code>	tính trung vị của các giá trị cho trong véc tơ <code>x</code>
<code>which(table(x) == max(table(x)))</code>	cho các giá trị của mode của các giá trị cho trong véc tơ <code>x</code> và vị trí theo <code>table(x)</code> của những giá trị mode này
<code>summary(x)</code>	cho các giá trị lớn nhất, nhỏ nhất, tứ phân vị thứ nhất, thứ hai, thứ ba và trung bình của các giá trị cho trong véc tơ <code>x</code>
<code>quantile(x)</code>	tính phân vị tùy ý của dữ liệu cho trong véc tơ <code>x</code>
<code>range(x)</code>	cho giá trị nhỏ nhất và lớn nhất của dữ liệu cho trong véc tơ <code>x</code>
<code>var(x)</code>	tính phương sai của các giá trị cho trong véc tơ <code>x</code>
<code>sd(x)</code>	tính độ lệch chuẩn của các giá trị cho trong véc tơ <code>x</code>
<code>skewness(x)</code>	tính hệ số bất đối xứng skewness của các giá trị cho trong véc tơ <code>x</code> ở giới $e1071$
<code>kurtosis(x)</code>	tính hệ số độ nhọn của các giá trị cho trong véc tơ <code>x</code> ở giới $e1071$

# Khoảng tin cậy – Kiểm định



# HÀM KHOẢNG TIN CẬY – KIỂM ĐỊNH THAM SỐ

Tham số	Hàm	Trường hợp	Gói
Trung bình	<code>z.test</code>	Phương sai đã biết	BSDA
	<code>t.test</code>	Phương sai chưa biết	
Tỷ lệ	<code>prop.test</code>		
Phương sai	<code>var.test</code>		

Khi dữ liệu thứ cấp, có thể dùng hàm **`zsum.test`** và **`tsum.test`** trong gói BSDA để tìm khoảng tin cậy/kiểm định cho trung bình.

# z.test và t.test trong khoảng tin cậy cho trung bình

- **z.test (x, sigma.x, conf.level)**

- `x`: véc tơ dữ liệu mẫu;
- `sigma.x`: độ lệch chuẩn của tổng thể;
- `conf.level`: số thuộc  $[0; 1]$  chỉ độ tin cậy của khoảng ước lượng, mặc định là 0.95

## **t.test (x, conf.level)**

- `x`: véc tơ dữ liệu mẫu;
- `conf.level`: số thuộc  $[0; 1]$  chỉ độ tin cậy của khoảng ước lượng, mặc định là 0.95

# prop.test trong khoảng tin cậy cho tỷ lệ

- **prop.test(x, n, conf.level, correct)**
  - x: số lần "thành công";
  - số lần thử nghiệm;
  - `conf.level`: số thuộc  $[0; 1]$  chỉ độ tin cậy của khoảng ước lượng, mặc định là 0.95;
  - tham số dạng logic chỉ xem có hay không sự điều chỉnh liên tục Yate, mặc định là `correct=TRUE`.

# Kiểm định giả thuyết trung bình: **z . test**

- **z.test (x, y = NULL, alternative = "two.sided", mu = 0, sigma.x = NULL, sigma.y = NULL)**
- **x**: véc tơ dữ liệu mẫu thứ nhất.
  - **y**: véc tơ dữ liệu mẫu thứ hai/ NULL nếu chỉ có một tổng thể
  - **sigma.x**: độ lệch chuẩn của tổng thể thứ nhất.
  - **sigma.y**: độ lệch chuẩn của tổng thể thứ hai/ NULL nếu chỉ có một tổng thể
  - **mu**: hiệu chênh lệch của hai giá trị trung bình, xác định theo giả thuyết không/trung bình theo giả thuyết không, mặc định bằng 0
  - **alternative**: chuỗi kí tự chỉ giả thuyết đối, **alternative = c("two.sided",**
  - **"less", "greater")** tương ứng chỉ giả thuyết đối là hai bên, bên trái, bên phải, mặc định là **"two.sided"**.

## Kiểm định giả thuyết về trung bình: `t.test`

- `t.test (x, y = NULL, alternative = "two.sided", mu = 0, paired = FALSE, var.equal = FALSE)`
- `x`: véc tơ dữ liệu mẫu thứ nhất;
  - `y`: véc tơ dữ liệu mẫu thứ hai/ `NULL` nếu chỉ có một tổng thể;
  - `mu`: giống `z.test`
  - `alternative`: giống `z.test`
  - `var.equal`: dạng logic chỉ phương sai hai tổng thể bằng nhau, mặc định là `FALSE`
  - `paired`: dạng logic chỉ chọn mẫu theo đôi, mặc định là `FALSE`



# Kiểm định giả thuyết về tỷ lệ: `prop.test`

`prop.test(x, n, p = NULL, alternative = c("two.sided", "less", "greater"), correct = TRUE)`

`x`: véc tơ chỉ số lần "thành công" trong mỗi mẫu;

- `n`: véc tơ chỉ số lần thử nghiệm trong mỗi mẫu;
- `p`: véc tơ chỉ xác suất thành công;
- `alternative`: xem trong hàm `z.test`;
- `correct`: tham số dạng logic chỉ xem có hay không sự điều chỉnh liên tục Yates,
- mặc định `correct=TRUE`.

# Kiểm định giả thuyết về phương sai: `var.test`

`var.test(x, y, ratio = 1, alternative = c("two.sided", "less", "greater"))`

- `x`: véc tơ dữ liệu mẫu thứ nhất;
- `y`: véc tơ dữ liệu mẫu thứ hai;
- `ratio`: tỉ số được giả thuyết của phương sai hai tổng thể , mặc định là 1, tức là
- phương sai hai tổng thể bằng nhau;
- `alternative`: xem trong hàm `z.test`;

# `chisq.test` trong kiểm chi-bình phương

- Kiểm chứng tính độc lập:

`chisq.test(A)`

– `A`: ma trận dữ liệu

- Kiểm định mức phù hợp của một phân phối

`chisq.test(x, p)`

- `x`: véc tơ dữ liệu;

- `p`: véc tơ có cùng chiều dài với `x` chỉ xác suất của phân phối cần kiểm chứng.

# Kiểm định phân phối chuẩn

- Hàm `shapiro.test` dùng kiểm định một tập dữ liệu có tuân theo phân phối chuẩn hay không dựa trên kiểm định Shapiro–Wilk được đưa ra bởi Samuel Shapiro và Martin Wilk vào năm 1965, chi tiết có thể tham khảo tại:

*[http://en.wikipedia.org/wiki/Shapiro%E2%80%93Wilk\\_test](http://en.wikipedia.org/wiki/Shapiro%E2%80%93Wilk_test)*

**`shapiro.test(x)`**

x là véc tơ dữ liệu cần kiểm định.

- Hàm `jarque.bera.test` trong gói `tseries` theo kiểm định Jarque–Bera được giới thiệu bởi Carlos Jarque và Anil K. Bera. Kiểm định Jarque–Bera kiểm định phân phối chuẩn dựa trên hệ số bất đối xứng skewness và hệ số độ nhọn kurtosis, chi tiết có thể tham khảo tại:

*[http://en.wikipedia.org/wiki/Jarque%E2%80%93Bera\\_test](http://en.wikipedia.org/wiki/Jarque%E2%80%93Bera_test)*

**`jarque.bera.test(x)`**

x là véc tơ dữ liệu cần kiểm định.

# Tương quan và hồi quy

# Hàm trong hồi quy tuyến tính/tương quan

Hàm	Kết quả về	Ghi chú
cor	Hệ số tương quan tuyến tính mẫu	
cor.test	Kiểm định hệ số tương quan tuyến tính	
plot	Biểu đồ tán xạ	
lm	Phương trình (đường) hồi qui mẫu	
abline	Vẽ thêm đường hồi quy mẫu vào tập điểm	
summary	Tổng hợp những kết quả trong MH hồi quy	
confint	Khoảng tin cậy cho hệ số hồi quy tổng thể	
predict	Dự báo trong hồi quy	
glht	Kiểm định tổ hợp tuyến tính hệ số hồi quy	Gói multcomp

# Hệ số tương quan

- Tính hệ số tương quan tuyến tính mẫu:  
`cor(x, y)`
  - `x, y`: véc tơ dữ liệu về hai biến.
- Kiểm định hệ số tương quan tổng thể  
`cor.test(x, y, alternative)`
  - `x, y`: véc tơ dữ liệu về hai biến;
  - `alternative` : giống `z.test`.

# Hàm hồi quy

- Vẽ biểu đồ tán xạ:

`plot(x, y)` hoặc `plot(y~x)`

– `x`, `y`: véc tơ dữ liệu về hai biến.

- Ước lượng hệ số trong mô hình hồi quy theo OLS

`lm(y ~ x, data)`

- `x`, `y`: hai véc tơ dữ liệu của biến độc lập và biến phụ thuộc tương ứng;

- `data` : bảng dữ liệu chứa dữ liệu của hai biến.



# Hàm hồi quy

- Tổng hợp những thông tin trong mô hình hồi quy  
`summary(lm(y~x))`
- Khoảng tin cậy hệ số hồi quy  
`confint(object, level = 0.95)`
  - object: mô hình hồi quy;
  - level: độ tin cậy.

# Kiểm định tổ hợp tuyến tính hệ số hồi quy

**glht(model, linfct, rhs, alt)**

- `model` : mô hình hồi qui;
- `linfct` : ma trận A trong giả thuyết  $H_0: AB = b$ ;
- `rhs` : véc tơ cột  $b$  trong giả thuyết  $H_0: AB = b$ ;
- `alt` : xâu kí tự chỉ giả thuyết đối, có thể là  
"less", "greater", "two.sided".

# Hàm dự báo trong hồi quy

**predict(object, newdata, interval, level)**

- `object` : mô hình hồi qui.
- `newdata` : giá trị của biến độc lập (dạng bảng) để dự báo cho biến phụ thuộc;
- `interval` : xâu kí tự chỉ kiểu dự báo,
  - "none" : ước lượng điểm cho Y, mặc định;
  - "confidence" : UL cho KTC cho kỳ vọng của Y;
  - "prediction" : UL cho khoảng dự báo của Y;
- `level` : độ tin cậy của khoảng ước lượng, mặc định là 0.95.

***CẢM ƠN QUÝ VỊ ĐÃ  
LẮNG NGHE!***

